UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

Desenvolvimento de um Sistema Automático de Classificação de Documentos com base em modelos de Machine Learning

Marta Pagán Martínez

Trabalho de Conclusão de Curso
MBA em Inteligência Artificial e Big Data

ICMC USP
SÃO CARLOS

UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

Desenvolvimento de um Sistema Automático de Classificação de Documentos com base em modelos de Machine Learning

Marta Pagán Martínez

USP - São Carlos

2022

Marta Pagán Martínez

Desenvolvimento de um Sistema Automático de Classificação de Documentos com base em modelos de Machine Learning

Trabalho de conclusão de curso apresentado ao Departamento de Ciências de Computação do Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo - ICMC/USP, como parte dos requisitos para obtenção do título de Especialista em Inteligência Artificial e Big Data.

Área de concentração: Inteligência Artificial

Orientadora: Prof. Dr. Marcelo G. Manzato

USP - São Carlos

2022

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

P385d

Pagán Martínez, Marta
Desenvolvimento de um sistema automático de classificação de documentos com base em modelos de machine learning / Marta Pagán Martínez; orientador Marcelo G. Manzato. -- São Carlos, 2022.
191 p.

Trabalho de conclusão de curso (MBA em Inteligência Artificial e Big Data) -- Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2022.

1. inteligência analítica. 2. mineração de texto. 3. PLN. 4. recuperação de informações. 5. análise inteligente de texto. I. G. Manzato, Marcelo, orient. II. Título.

Bibliotecários responsáveis pela estrutura de catalogação da publicação de acordo com a AACR2:
Gláucia Maria Saia Cristianini - CRB - 8/4938
Juliana de Souza Moraes - CRB - 8/6176

DEDICATÓRIA

A meu esposo e companheira vital, Adriano, pelo carinho, pelo amor, por renunciar a tantas coisas por mim, por estar nos piores momentos, por compartilhar seus conhecimentos e por me ensinar novas formas de enxergar a realidade.

A meu filho Renuá, de quem bastou sentir o primeiro latido do seu coração para me apaixonar antes de conhece-lo, e que enche de cor, amor e felicidade minha vida.

A minha mãe, Lola, pelo grande apoio e ajuda para poder realizar esse trabalho de pesquisa e quem me ajudou na minha formação sempre.

A todas as pessoas especiais que apareceram na minha vida para iluminar meu caminho, me ajudar e me ensinar.

AGRADECIMENTOS

Agradeço a meu orientador, Prof. Marcelo Manzato pela atenção, respeito, ajuda e ideias compartilhadas.

Agradeço a todos da turma do MBA em Inteligência Artificial do ICMC da USP de São Carlos, especialmente a Nubia Piccirilli e a Jaqueline Laruccia pelas horas dedicadas para me ajudar, pela paciência, pelas dicas e por compartilhar seus conhecimentos fazendo possível o desenvolvimento desse TCC. Cada um, a sua forma, acrescentou um pouco de conhecimento em minha vida, seja este conhecimento acadêmico, profissional ou pessoal. Fui agraciado por conhecer todos vocês mesmo de forma indireta por meio dos comentários em foros e no Discord.

Agradeço ao ICMC da USP de São Carlos, especialmente à coordenação por ter me dado a bolsa integral para cursar o MBA, por acreditar em mim e pela grande oportunidade de abrir novos horizontes de conhecimento.

Aos professores pelo apoio e suporte que foram muito importantes nesta caminhada e pela oportunidade de aprender sobre assuntos e temas tão relevantes. Ao Prof. Dr. Ricardo Sant'Ana por me dar a oportunidade de ir no Brasil em 2015, à Profa. Dra. Zaira Regina Zafalon, que tive à grande ideia de me convidar para dar uma palestra com a impressionante Profa. Dra. Solange que acreditou em min desde o inicio, me abriu as portas da suas magnificas aulas e me animou à cursar esse incrível MBA, abrindo assim novos horizontes à interdisciplinaridade, à inovação e à oportunidade de me adentrar no apaixonante mondo da Ciência de Dados, Inteligência Artificial, processamento da linguagem natural e mineração e análise inteligente de texto, com um MBA que muito me ensinou contribuindo para o meu crescimento científico, professional e pessoal.

Agradeço à minha mãe e minha cunhada Carol, pelo apoio nos momentos de dificuldade e por fazer parte da minha rede de apoio que tem sido imprescindível para a conclusão de mais uma etapa em minha vida, pois sem seu apoio não teria conseguido fazer esse TCC.

Por fim, agradeço à meu esposo e companheiro vital, Adriano, pelo amor, carinho, compreensão, força e encorajamento na construção deste trabalho e da superação em cada curso ou módulo do MBA, pois sem seu apoio e conhecimento não teria conseguido e tudo teria sido mais difícil.

EPÍGRAFE

“O valor de uma ideia está no uso que se faz dela”

Thomas Edison

RESUMO

PAGÁN MARTÍNEZ, M. **Desenvolvimento de um Sistema Automático de Classificação de Documentos com base em modelos de Machine Learning**. 2022. 191 f. Trabalho de conclusão de curso (MBA em Inteligência Artificial e Big Data) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2022.

O principal desafio que tem as entidades públicas e privadas que recebem ou produzem grande quantidade de informação é a sua correta organização e classificação para facilitar a sua posterior recuperação, consulta, pesquisa e gestão, alcançando assim eficiência, inteligência corporativa, transparência e cumprimento da regulamentação associada. Para isso, a aquisição ou aluguel de um sistema de gestão de documentos eletrônicos tem se tornado a solução mais frequente e eficaz. No entanto, a responsabilidade final pela classificação e gestão cabe aos próprios usuários da plataforma, que muitas vezes não investem o tempo necessário, ou não têm conhecimento completo das tabelas de classificação, ignorando se existem documentos, procedimentos e/ou arquivos semelhantes onde localizar os documentos recebidos. Isso leva à retenção de documentos e petições, gerando deficiências na manutenção do sistema documental e riscos com o cumprimento de determinados prazos legais. Nos últimos anos, o avanço dos métodos de Inteligência Artificial e das tecnologias de aprendizado automático, tem sido espetacular para descobrir padrões e para a classificação automática de conteúdo digital. Nesse contexto, pretende-se desenvolver um Sistema Automático de Classificação de Documentos com base em modelos de aprendizado de máquina seguindo uma metodologia que permitirá incorporar os novos modelos aos workflows padrão de gerenciamento de documentos. Com este trabalho, espera-se fornecer aos sistemas de gerenciamento de documentos e informação maior automação, aumentar a produtividade dos usuários, melhorar a experiência e usabilidade dos usuários dos sistemas de gestão de documentos eletrônicos de arquivo, facilitar as tarefas de gestão documental, evitar erros de classificação e reduzir a dependência de pessoas para tarefas susceptíveis de sistematização. Os principais achados mostram a precisão e acurácia de alguns algoritmos ou modelos frente a outros utilizados para gerar o sistema de classificação automática. Os resultados permitem concluir que o uso de aprendizado de maquina e a análise inteligente de texto com mineração de texto é de capital importância para otimizar esses sistemas de classificação, além de para extrair padrões e conhecimento que agilize as tarefas de classificação e recuperação de informação na gestão documental.

Palavras-chave: inteligência analítica; mineração de texto; PLN; recuperação de informações.

ABSTRACT

PAGÁN MARTÍNEZ**,** M. **Development of an Automatic Document Classification System based on Machine Learning models**. 2022. 191p. Master's thesis (MBA in Artificial Intelligence and Big Data) – Institute of Mathematics and Computer Science, University of São Paulo, São Carlos, 2022.

The main challenge faced by public and private entities that receive or produce a large amount of information is its correct organization and classification to facilitate its subsequent retrieval, consultation, research and management, thus achieving efficiency, corporate intelligence, transparency and regulatory compliance. associated. For this, the acquisition or rental of an electronic document management system has become the most frequent and effective solution. However, the final responsibility for classification and management rests with the platform users themselves, who often do not invest the necessary time, or do not have complete knowledge of the classification tables, ignoring whether there are similar documents, procedures and/or files where to locate the received documents. This leads to the retention of documents and petitions, generating deficiencies in the maintenance of the document system and risks with the fulfilment of certain legal deadlines. In recent years, the advancement of Artificial Intelligence methods and Machine Learning technologies has been spectacular for discovering patterns and automatically classifying digital content. In this context, we intend to develop an Automatic File Document Classification System based on machine learning models, following a methodology that will allow the incorporation of the new models to the standard workflows of document management. With this work, it is expected to provide document and information management systems with greater automation, increase user productivity, improve the experience and usability of electronic document management system users, facilitate document management tasks, avoid classification errors and reduce dependency of people for tasks susceptible to systematization. The main findings show the precision and accuracy of some algorithms or models compared to others used to generate the automatic classification system. The results allow us to conclude that the use of machine learning and intelligent text analysis with text mining is of paramount importance to optimize these classification systems, in addition to extracting patterns and knowledge that streamlines the tasks of classification and retrieval of information in document management.

Keywords: analytical intelligence; text mining; NLP; Information Search and Retrieval.

LISTA DE ILUSTRAÇÕES

LISTA DE GRÁFICOS

LISTA DE TABELAS

LISTA DE EQUAÇÕES

LISTA DE ABREVIATURAS E SIGLAS

AM	–	Aprendizado de Máquina
AUC	–	Area Under Curve (ROC)
BoW	–	Bag of Words
BPM	–	Business Process Management
CA	–	Classifier acuracy
DBScan	–	Density-Based Spatial Clustering of Applications with Noise
DM	–	Data Mining
DP	–	Deep Learning
ECM	–	Enterprise Content Management
EDMS	–	Electronic Documents Management System
EDRMS	–	Electronic Documents and Records Management System
EW	–	Emty Word
FN	–	Falsos Negativos
FP	–	Falsos Positivos
FPR	–	Fração de falsos positivos
GB	–	Gradient Boosting
HTML	–	HyperText Markup Language
IA	–	Inteligência Artificial
ISR	–	Information Search and Retrieval
K-NN	–	K-Nearest Neighbour
KDD	–	Knowledge Discovery in Databases
KFCV	–	*k*-fold cross-validation
KM	–	K-Means
LOOCV	–	Leave-one-out cross-validation
LpO	–	Leave-p-out cross-validation
LpOCV	–	Leave-p-out cross-validation
ML	–	Machine Learning
NB	–	Naïve Bayes
NLP	–	Natural Language Processing
PLN	–	Processamento da Linguagem Natural

RF	–	Random Forest
RL	–	Regressão Logística
RN	–	Rede Neural
ROC	–	Receiver Operating Characteristics
SGD	–	Sistema de Gestão Documental
SGDE	–	Sistemas de Gestão de Documentos Eletrônicos
SGDEA	–	Sistemas de Gestão de Documentos Eletrônicos Arquivo
SVC	–	Soft Voting Classifier
SVM	–	Suport Vector Machinenes
TA	–	Text analytics
TF-IDF	–	Term frequency – Inverse document frequency
TIC	–	Tecnologias da Informação e Comunicação
TM	–	Text Mining
TPR	–	Fração de verdadeiros positivos
TRD	–	Tabelas de Retenção Documental
TT	–	Tabelas de Temporalidade
TTD	–	Tabelas de Temporalidade documental
URL	–	Uniform Resource Locator
VN	–	Verdadeiros Negativos
VP	–	Verdadeiros Positivos
WE	–	Word Embedding

SUMÁRIO

1 INTRODUÇÃO

Essa seção introdutória pretende apresentar o trabalho, pelo que contem a delimitação do assunto tratado ou contextualização, a justificativa e motivação, a questão de pesquisa e os objetivos da pesquisa.

1.1 Contextualização

O principal desafio que tem as entidades públicas e privadas que recebem ou produzem grande quantidade de informação é a sua correta organização e classificação para facilitar a sua posterior recuperação, consulta, pesquisa e gestão, alcançando assim eficiência, inteligência corporativa, transparência e cumprimento da regulamentação associada. Para isso, a aquisição ou aluguel de plataformas profissionais de gerenciamento de documentos, isto é, Sistema de Gestão de Documentos Eletrônicos (SGDE) ou Sistemas de Gestão de Documentos Eletrônicos Arquivo (SGDE(A)) tem se tornado a solução mais frequente e eficaz. No entanto, a responsabilidade final pela classificação e gestão cabe aos próprios usuários da plataforma, que muitas vezes não investem o tempo necessário, ou não têm conhecimento completo das tabelas de classificação, ignorando se existem documentos, procedimentos e/ou arquivos semelhantes onde localizar os documentos recebidos. Desta forma, uma infinidade de documentos e petições são retidas, gerando deficiências na manutenção do sistema documental e riscos com o cumprimento de determinados prazos legais.

Nos últimos anos, o avanço dos métodos de Inteligência Artificial (IA) e, em particular, das tecnologias de Aprendizado de Máquina ou *Machine Learning* (ML), tem sido espetacular para a descoberta de padrões e a classificação automática de conteúdo digital (KHAN e MADDEN, 2014; REZENDE, 2003; TAX, 2001).

Diante desse contexto, pretende-se desenvolver um Sistema Automático de Classificação de Documentos com base em modelos de ML seguindo uma metodologia que permitirá incorporar os novos modelos aos *workflows* padrão de gerenciamento de documentos.

Com este trabalho, espera-se fornecer aos sistemas de gerenciamento de documentos e informação maior automação, aumentar a produtividade dos usuários, melhorar a experiência e usabilidade dos usuários dos SGDEA, facilitar as tarefas de gestão documental, evitar erros de classificação e reduzir a dependência de pessoas para tarefas susceptíveis de sistematização.

1.2 Justificativa e Motivação

Devido à importância para empresas e entidades públicas e privadas da organização e disponibilidade de seus documentos para recuperação e consulta, um sistema automático de classificação de Documentos eletrônicos com base em modelos de ML pode resolver problemas de confiabilidade, disponibilidade e custódia de documentos, além de tornar mais eficientes as atividades de acolhimento e atenção de solicitações, reclamações e recursos, estabelecidas à empresa por seus usuários e entidades de controle (HERNÁNDEZ ECHEVERRY, 2017).

A gestão documental está vinculada à atividade administrativa, ao cumprimento de funções e ao desenvolvimento dos processos e procedimentos de todas as entidades. Como contribuição para essa gestão, as organizações podem se apropriar de práticas de gestão de documentos, contando com o uso das tecnologias de informação e comunicação (TIC), levando em conta as recomendações, conceitos e regulamentos emitidos pelo Arquivo Geral do país e as referências internacionais competentes e adequadas no campo, que permitem promover a prestação eficiente de procedimentos, serviços, conteúdos e aplicativos para uma correta gestão de informações institucionais (RANGEL PALENCIA, 2019).

A classificação documental, um método e uma ordem anterior; a falta de representação e uniformidade na nomenclatura de processos administrativos, arquivos e documentos; a falta de um catálogo de políticas de preservação e destinação de documentos; dificuldade em localizar informações; tipologia documental diferente associada ao gerenciador de documentos e registro de diferentes bancos de dados como parte do repositório documental; a duplicação de informações e documentos ou a falta de controle no armazenamento de informações e documentos em um SGDE(A) são alguns dos principais problemas para entidades ou empresas, bem como a necessidade de incorporar informações de diferentes tipos e muitas entradas no sistema. Além disso, estudos recentes apontam a necessidade de soluções que atinjam um bom desempenho da classificação documental (GALLEGO GARCÍA, 2015; HERNÁNDEZ ECHEVERRY, 2017; JOHNSTON e BOWEN, 2005; RANGEL PALENCIA, 2019).

Neste contexto, é proposto o uso de modelos de ML seguindo uma metodologia que permita incorporar os novos modelos aos *workflows* padrão de um SGDE(A) onde se realizará a decomposição de texto em n-gramas, a remoção ou eliminação de palavras vazias (*empty words*), o cálculo da frequência dos n-gramas, a estimativa da estatística tf-idf para os n-gramas e a seleção das variáveis mais importantes.

Alguns dos modelos de ML a ser aplicados nesse projeto, são aqueles que permitam gerar algoritmos de classificação automática de documentos (modelo de classificação);

desenvolver um algoritmo para detectar padrões de gerenciamento de documentos; aplicar um sistema de regras para automatizar tarefas repetitivas; fornecer capacidade de aprendizagem para sistemas.

Dentre os métodos a serem usados, cabe destacar o método *fit* que permitirá ajustar os modelos passando os dados de treinamento e a variável objetivo. Assim, pretende-se desenvolver algoritmos de aprendizado de máquina para realizar reconhecimento automático, classificação e roteamento de documentos eletrônicos processados e armazenados em um SGDE(A).

Espera-se que a aplicação ou sistema atinja um desempenho favorável, diminua erros humanos, automatize tarefas e otimize a gestão de conteúdos e os processos documentais de um SGDE(A).

1.3 Questões de Pesquisa e Objetivos

Neste trabalho, espera-se reduzir as limitações dos Sistemas de Gestão de documentos eletrônicos (SGDE) em termos de agilidade, eficácia, recuperação de informação, independência de pessoas em algumas tarefas que podem ser automatizadas e diminuir problemas de classificação. Para isso, pretende-se desenvolver um Sistema Automático de Classificação de Documentos com base em modelos de *machine learning.*

Perante os desafios e problemas atualmente enfrentados pelos usuários de SGDE e SGDEA, foi elaborada a seguinte questão de pesquisa que norteará esse trabalho:

Q1 *Os Sistemas Automáticos de Classificação de Documentos com base em modelos de machine learning apresentam otimização* nos processos operacionais e *um maior desempenho de organização, classificação e recuperação de informação em relação a sistemas tradicionais?*

Diante dessa questão de pesquisa, são definidos os seguintes objetivos específicos para o desenvolvimento desse trabalho (esses objetivos estão relacionados à questão de pesquisa Q1):

- Coletar informações sobre requisitos legais, arquivamento e parâmetros processuais para a implementação de um modelo eletrônico de gestão de documentos.
- Analisar as informações coletadas para estabelecer os parâmetros de arquivamento e procedimento para a implantação do sistema eletrônico de gestão de documentos, que permite a captura de informações e geração de documentos eletrônicos de arquivamento.

- Preparar um documento com todos os requisitos de *software* e *hardware* necessários para o desenvolvimento de um sistema eletrônico de gerenciamento de documentos baseado em modelos de ML para os processos operacionais de um SGDE(A)
- Mapear algoritmos de aprendizado uniclasse, biclasse e multiclasse disponíveis na literatura, analisando suas lacunas e desempenhos de classificação na detecção de documentos, considerando tanto cenários balanceados quanto cenários desbalanceados.
- Aplicar algoritmos que se destacam na literatura no desenvolvimento de um sistema automático de classificação de documentos com base em modelos de *machine learning*.
- Utilizar pelo menos dois tipos de modelos: um modelo baseado em redes neurais e outro baseado em técnicas de aprendizado de máquina como florestas aleatórias (*Random Forests)* e Xgboost.
- Descrever os pontos fortes e fracos dos diferentes modelos utilizados.
- Avaliar o impacto das diferentes formas de combinação das classificações dos modelos e comparar os desempenhos obtidos a algoritmos de classificação usados.

A partir do modelo proposto, espera-se que os resultados mostrem que os sistemas automáticos de classificação de documentos electrónicos com base em modelos de aprendizado de máquina apresentem maior desempenho e otimização nos processos operacionais em relação a sistemas tradicionais.

2 REVISÃO BIBLIOGRÁFICA

Este capítulo apresenta os principais conceitos de inteligência artificial, *machine learning*, *data mining (DM)*, *text mining (TM)* ou *text analytics (TA)*, sistemas de informação e classificação documental, abrangendo o processamento e preparação dos textos, os algoritmos de classificação utilizados e os métodos de avaliação dos resultados gerados. O levantamento do estado da arte leva em consideração literatura científica como artigos e teses de doutorado de entre outros relacionadas à classificação com grande número de classes, classificação automática de textos y sistemas gestão de documentos eletrônicos (SGDE) e sistemas de documentos eletrônicos de arquivo (SGDEA).

2.1 Inteligência artificial

A Inteligência Artificial (IA) é um campo de conhecimento associado à linguagem e à inteligência, ao raciocínio, ao aprendizado e a resolução de problemas (KAUFMAN, 2019).

Segundo Kaufman (2019) a IA propicia a simbiose entre a máquina e o humano ao acoplar sistemas inteligentes artificiais ao corpo humano (exemplo: células artificiais, prótese cerebral, braço biônico, joelho inteligente, etc.), e a interação entre a máquina e o humano como duas “espécies” diferentes conectadas (exemplo: homem-aplicativos, homem-algoritmos de IA, etc.).

Segundo Ning e Yan (2010) a IA e uma ciência tecnológica, que pesquisa e desenvolve métodos, técnicas e aplicações para simular e expandir a teoria da inteligência humana. IA e um ramo da ciência da computação que tem como objetivo entender a essência da inteligência e produzir uma nova maquina inteligente capaz de ter reações similares à inteligência humana.

A IA é um tema de pesquisa em diversas áreas tais como Computação, Linguística, Filosofia, Matemática, Neurociência, entre outras. De fato, existe uma ampla diversidade de subcampos, atividades, pesquisas e experimentações sobre IA que dificulta a descrição do estado da arte atual sobre o tema. Os estágios de desenvolvimento e as expectativas sobre IA variam entre os campos e suas aplicações, pois incluem: veículos autônomos, reconhecimento de voz, reconhecimentos de imagens, jogos, robótica, tradução de linguagem natural, diagnósticos médicos, entre outros (KAUFMAN, 2019).

Com base no trabalho dos autores Hosea, Harikrishnan e Rajkumar (2011) e Ning e Yan (2011), as pesquisas no campo da Inteligência Artificial incluem: Reconhecimento de fala; Reconhecimento de imagem; Robótica (consiste em programar computadores para perceber e

reagir a outros estímulos sensoriais); Processamento de Linguagem natural (consiste em programar computadores para entender a linguagem natural humana); Sistemas especialistas (consiste em programar computadores para tomar decisões em situações da vida real, por exemplo, alguns sistemas especialistas ajudam médicos a diagnosticarem doenças baseados em sintomas); Redes neurais (que são sistemas que simulam inteligência humana para tentar reproduzir os tipos de conexões físicas que ocorrem em cérebros biológicos humanos ou animais); ou Jogos eletrônicos (consiste em programar computadores para jogar jogos como xadrez e damas, os quais demandam algoritmos baseados em raciocínio logico).

2.2 *Machine Learning*, aprendizado automático ou aprendizado de máquina

O *machine learning* ou Aprendizado de Máquina (AM) é um campo de pesquisa que tem uma interseção com a Estatística, Inteligência Artificial e Ciência da Computação e que também é conhecido como Análise Preditiva ou Aprendizado Estatístico (MULLER e GUIDO, 2017).

Para Mitchell, (1997) o ML é como uma área que pesquisa métodos computacionais relacionados à aquisição automática de novos conhecimentos, novas habilidades e novas formas de organizar a informação já existente. É mais, segundo Mitchell (1997), ML pode ser definido como a capacidade de melhorar o desempenho na realização de alguma tarefa por meio da experiência. Desta forma, no AM, os computadores são programados para aprender com a experiência anterior.

Segundo Monard e Baranauskas (2003), o ML é uma ferramenta poderosa para a aquisição automática de conhecimento. Contudo, deve-se observar que não existe uma única técnica que apresente o melhor desempenho para todos os problemas. Sendo assim, "é importante compreender o poder e a limitação dos diversos algoritmos de AM utilizando alguma metodologia que permita avaliar os conceitos induzidos por esses algoritmos em determinados problemas" (MONARD e BARANAUSKAS, 2003, p. 90)

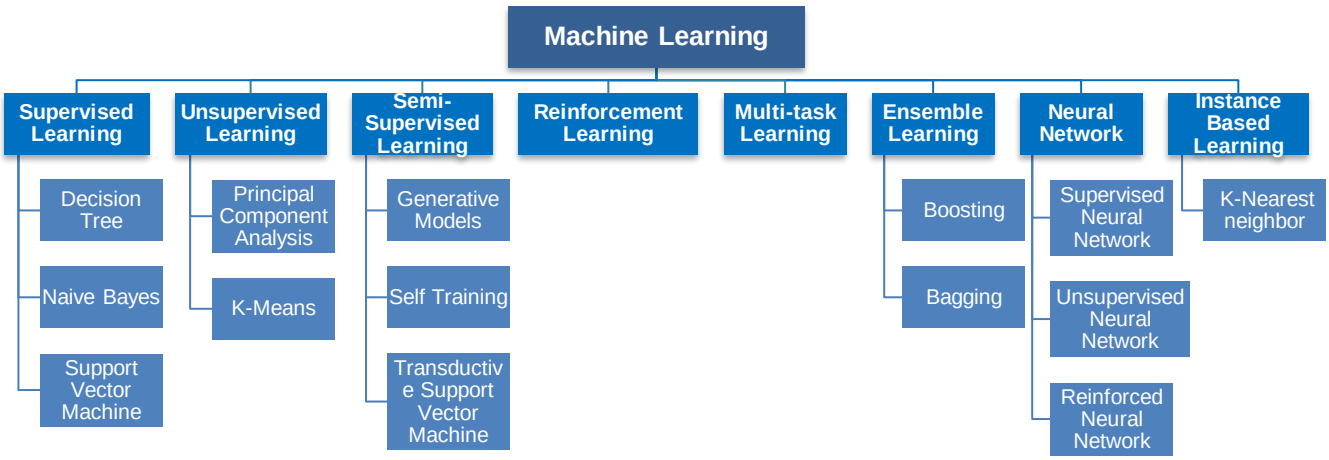
> Aprendizado de Máquina é uma área de Inteligência Artificial (AI) cujo objetivo é o desenvolvimento de técnicas computacionais sobre o aprendizado, bem como a construção de sistemas capazes de adquirir conhecimento de forma automática. Um sistema de aprendizado é um programa de computador que toma decisões com base em experiências acumuladas por meio da solução bem-sucedida de problemas anteriores. (MONARD; BARANAUSKAS, 2003, p. 89).

Baeza-Yates e Ribeiro-Neto (2013) consideram que o AM é uma área ampla da IA que está preocupada com o projeto e o desenvolvimento de algoritmos que aprendem a partir dos dados fornecidos como entrada. Nesse contexto, "os padrões aprendidos, que podem ser bem

complexos, são então usados para fazer predições relativas a dados ainda não vistos e novos." (BAEZA-YATES; RIBEIRO-NETO, 2013, p. 278). Nesse contexto, o AM é semelhante ao aprendizado humano em pelo menos um aspecto: "é um aprendizado baseado em experiências. A máquina aprende através do reconhecimento de padrões. Dessa forma, quanto maior for a quantidade de dados em que a máquina for "alimentada", mais preciso será o reconhecimento desses padrões." (DE MAGALHÃES, 2020, p. 67).

Nas seções subsequentes se detalham os diferentes métodos, técnicas ou ferramentas de *machine learning* existentes. Existem diferentes técnicas de ML como apresenta a Figura 1.

Figura 1 – Técnicas de *Machine Learning*

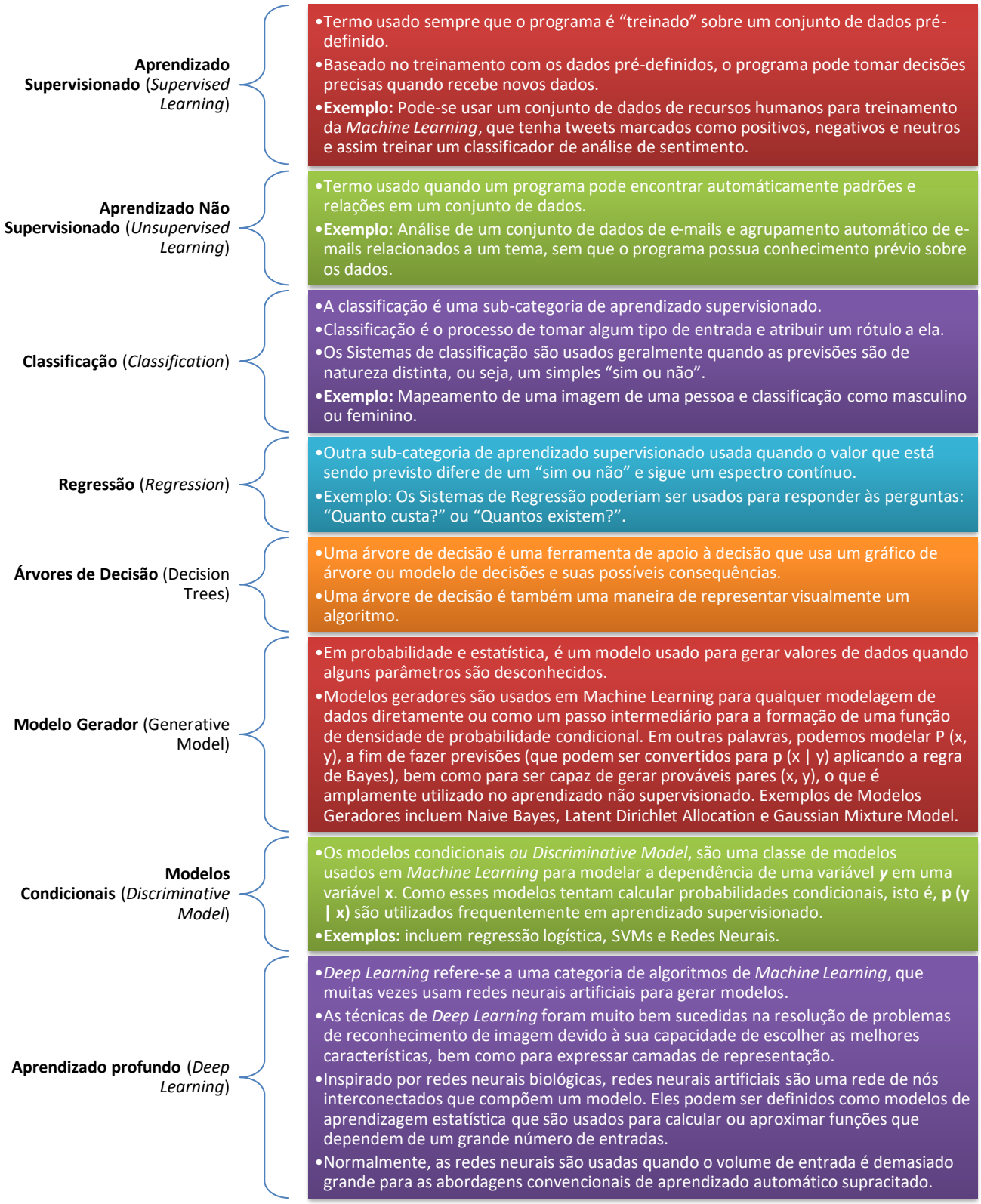


Fonte: Elaborado pela autora.

2.2.1 Métodos, técnicas ou ferramentas de *machine learning*

Os principais métodos, técnicas ou ferramentas de *machine learning* são o aprendizado ou treinamento supervisionado, o aprendizado ou treinamento não supervisionado e o aprendizado profundo ou *Deep Learning (DL)*. A modo de resumo, a Figura 2 apresenta os diferentes métodos ou ferramentas de ML encontradas na literatura.

Figura 2 – Métodos ou ferrramentas de *machine learning*



Fonte: Elaborado pela autora com base em Matos (2015)

2.2.2 Aprendizado Supervisionado

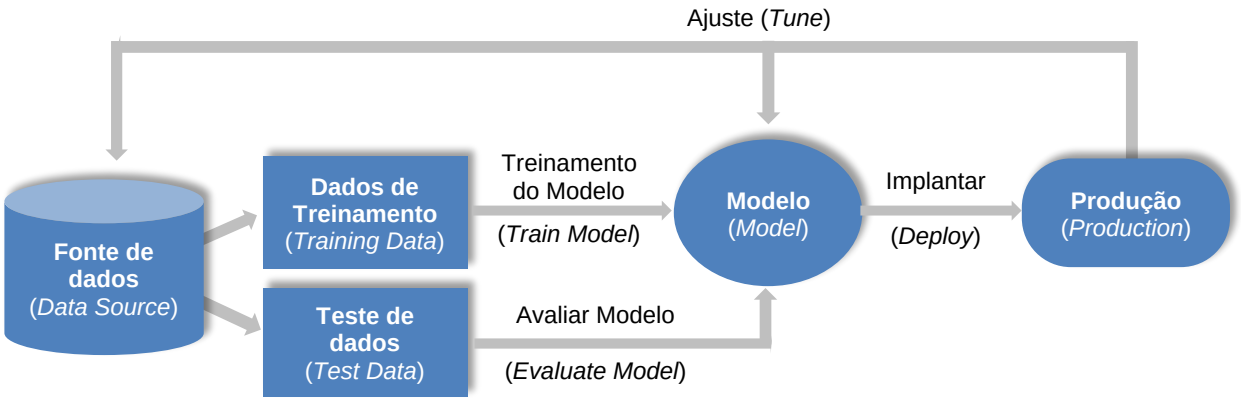
O Aprendizado supervisionado “refere-se à capacidade que determinados algoritmos têm de aprender a partir de exemplos.” (GOLDSCHMIDT; PASSOS; BEZERRA, 2015, p. 73). Nesse tipo de aprendizado, “a máquina é alimentada com uma amostragem, que é rotulada de acordo com suas características dominantes. Assim, o algoritmo vai aprender a partir dessas amostras. Na prática, é como se a máquina fosse instruída com a orientação de um professor, por isso o termo aprendizado supervisionado” (DE MAGALHÃES, 2020, p. 69).

Nesse método é fornecido ao algoritmo de aprendizado ou indutor que é um conjunto de exemplos de treinamento para os quais o rótulo da classe associada é conhecido (CHEESEMAN e STUTZ, 1996).

> Um algoritmo é dito supervisionado quando usa informação fornecida por seres humanos ou obtida por meio de assistência humana como dado de entrada. No caso padrão, um conjunto de classes e exemplos de documentos para cada classe são fornecidos. Os exemplos são determinados por especialistas humanos e constituem o conjunto de treinamento, que é então utilizado para aprender uma função de classificação. Uma vez que essa função for aprendida, é então usada para classificar novos documentos não vistos. (GONÇALVES, 2013, p. 281).

A Figura 3 representa o *workflou* ou fluxograma do algoritmo de aprendizado de maquina supervisionado.

Figura 3 – Fluxograma do algoritmo de aprendizado supervisionado



Fonte: Elaborado pela autora com base em Mahesh (2020)

O aprendizado supervisionado é usada sempre que se pretende prever uma determinada saída a partir de uma determinada entrada, e existem exemplos de pares de entrada / saída, ou seja, tem-se um conjunto de dados cujas saídas são conhecidas previamente (MULLER e GUIDO, 2017).

> O aprendizado supervisionado compreende a abstração de um modelo de conhecimento a partir dos dados apresentados na forma de pares ordenados (entrada, saída desejada). Por entrada entende-se o conjunto de valores das variáveis (atributos) de entrada do algoritmo para um determinado caso. Tais variáveis são denominadas atributos previsores. A saída desejada corresponde ao valor de uma variável (denominada atributo-alvo) que se espera que o algoritmo possa produzir sempre que receber os valores especificados em entrada. (GOLDSCHMIDT, PASSOS e BEZERRA, 2015, p.73).

Em relação ao aprendizado supervisionado, Goldschmidt, Passos e Bezerra (2015) afirmam que:

> O aprendizado supervisionado consiste em, dada uma coleção de exemplos de f, obter uma função h que seja uma aproximação de f. A função h é chamada de hipótese ou modelo de f. [...]. A identificação da função h consiste de um processo de busca, em um espaço de hipótese candidatas H, pela hipótese que mais se aproxime da função original f. Esse processo de busca é o aprendizado supervisionado e a hipótese selecionada é o modelo de conhecimento abstraído a partir dos dados. Todo algoritmo que possa ser utilizado na execução do aprendizado é chamado algoritmo de aprendizado. O conjunto de todas as hipóteses que podem ser obtidas a partir de um algoritmo de aprendizado L é representado por HL. Cada hipótese pertencente a HL é representada por HL. A acurácia de uma hipótese h retrata a capacidade de h em mapear corretamente cada vetor de entradas x em f(x). O conjunto de pares (x, f(x)) utilizados na identificação da função h é denominado conjunto de treinamento. Por outro lado, o conjunto de pares (x, f(x)) utilizados para avaliar a acurácia de h é denominado conjunto de teste. Assim, o algoritmo de aprendizado L pode ser interpretado como uma função L: T → HL, onde T é o espaço composto por todos os conjuntos de treinamento possíveis para L. (GOLDSCHMIDT; PASSOS; BEZERRA, 2015, p. 73).

Por conseguinte, o aprendizado supervisionado usa “padrões para prever os dados do rótulo, também chamado de *labels*, em dados adicionais não rotulados. Esse modelo de aprendizado se divide em duas subcategorias: Classificação e Regressão.” (DE MAGALHÃES, 2020, p. 71).

Segundo Kotsiantis, Zaharakis e Pintelas (2007), a aprendizado de maquina indutiva ou supervisionada consiste na criação de um classificador capaz de aprender a partir de um conjunto de treinamento e generalizar para as novas instancias que aparecem. Durante a classificação supervisionada são apresentados os possíveis resultados recolhidos através da observação, desse modo ele possuir a um escopo limitado e pré-definido de resultados que serão utilizados como parâmetro e, principalmente, como referência durante a classificação.

Classificação e regressão são dois tipos principais de problemas de aprendizado de maquina supervisionados (GUIDO e MULLER, 2016). Uma forma fácil de distinguir tarefas de classificação e regressão é perguntar se existe algum tipo de continuidade na saída, pois se houver continuidade entre os resultados possíveis, é um problema de regressão (SILVA, 2021).

2.2.2.1 Classificação

Os sistemas de classificação são utilizados para organizar o conhecimento (JACOB, 1992). De acordo com Jacob (2004), a classificação é um processo legal e sistemático que envolve a atribuição ordenada e sistemática de cada entidade a uma e apenas uma classe dentro de um sistema de classes reciprocamente exclusivas e não sobrepostas. O autor aponta que é um processo legal porque é realizado de acordo com um conjunto estabelecido de princípios que governam a estrutura de classes e as relações entre elas; e é processo sistemático porque exige a aplicação consistente desses fundamentos dentro da estrutura de uma ordenação prescrita da realidade. Também assinala que o esquema do processo de classificação é arbitrário e artificial: arbitrário, porque os critérios usados para definir classes no esquema refletem uma perspectiva única do domínio, excluindo todas as outras perspectivas; e artificial porque é uma ferramenta criada expressamente para estabelecer uma organização significativa.

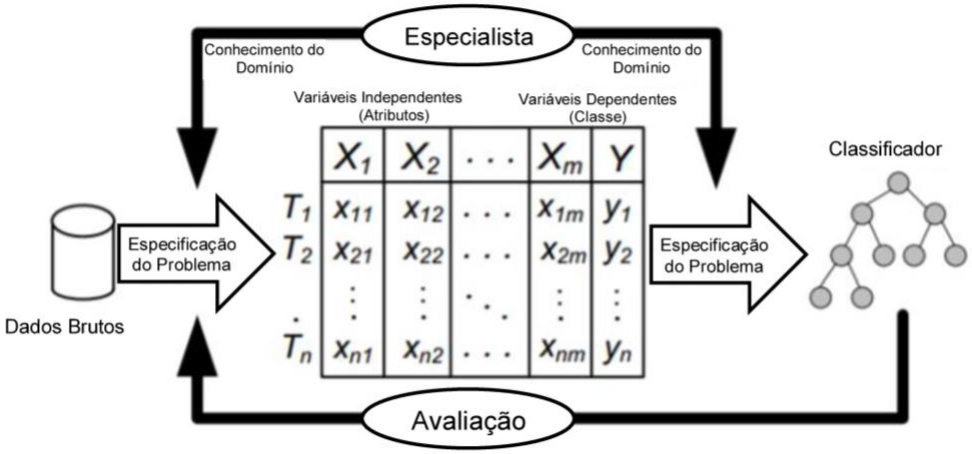
Segundo De Magalhães (2020, p. 71), a classificação é "o processo de adotar algum tipo de entrada e atribuir um rótulo a ela". A autora também aponta que a inserção dos documentos em classes é um processo conhecido como "classificação de textos" e existem diferentes estudos e algoritmos desenvolvidos para aprimorar os recursos que permitam organizar a informação e "induzir automaticamente os sistemas capazes de lidar com problemas de classificação". De fato, desde:

> [...] os primeiros dias da Grande Biblioteca de Alexandria por volta de 300 a. C., bibliotecários tinham que lidar com armazenamento de documentos para recuperação e leitura futura. Com o passar do tempo, o tamanho das coleções cresceu e o problema tornou-se cada vez mais difícil. Procurar por um livro em particular dentre centenas de livros tornou-se uma tarefa tediosa, demorada e impraticável. Para aliviar o problema, bibliotecários começaram a rotular os documentos. Isso forneceu metainformação ao seu conteúdo, permitindo assim que os livros fossem organizados com uma visão que permitia busca rápida e recuperação. Uma das primeiras abordagens para rotular documentos foi atribuir um identificador único para cada documento. Isso resolvia o problema sempre que o usuário soubesse dos identificadores dos livros que eles queriam, mas não resolvia o problema mais genérico de encontrar documentos sobre um assunto ou tópico específico. Nesse caso, a solução natural é agrupar os documentos por tópicos comuns e nomear cada grupo com um ou mais rótulos significativos. Cada grupo rotulado é o que chamamos de uma classe, isto é, um conjunto no qual podemos inserir documentos cujo conteúdo pode ser descrito pelo seu rótulo. (GONÇALVES, 2013, p. 277)

Como mostrado na Figura 4, no processo de classificação, o algoritmo de aprendizado constrói um classificador capaz de determinar corretamente a classe de novos exemplos que ainda não foram etiquetados, dado um conjunto de classes e um conjunto de exemplos de treinamento. Na maioria dos casos, esse processo pode usar o conhecimento de um domínio para fornecer alguma informação previamente conhecida como entrada ao indutor. E, após

induzido, o classificador é normalmente avaliado e o processo de classificação pode ser repetido, se for necessário (MONARD, BARANAUSKAS, 2003; DE MAGALHÃES, 2020).

Figura 4 – Processo de classificação



Fonte: Monard e Baranauskas (2003, p.92)

Goldschmidt, Passos e Bezerra (2015), apontam que na tarefa de classificação os atributos do conjunto de dados são divididos em dois tipos: Atributo previsor e atributo-alvo. De tal modo que:

> Para cada valor distinto do atributo-alvo tem-se uma classe que normalmente corresponde a um rótulo categórico pertencente a um conjunto predefinido. A tarefa de classificação consiste em descobrir uma função que mapeie um conjunto de registros em um conjunto de classes. Uma vez descoberta, tal função pode ser aplicada a novos registros de forma a prever a classe em que tais registros se enquadram. Como exemplo, considere uma financeira que possui o histórico de seus clientes e o comportamento destes em relação ao pagamento de empréstimos contraídos previamente. Considere também dois tipos de clientes: adimplentes e inadimplentes. Essas que são as classes dos problemas (valores do atributo-alvo). Uma aplicação da tarefa de classificação, neste caso, consiste em descobrir uma função que mapeie corretamente os clientes, a partir de seus dados (valores dos atributos previsores). (GOLDSCHMIDT; PASSOS; BEZERRA, 2015, p. 25).

Por conseguinte, nas tarefas de classificação:

> um dos grupos possui somente um atributo, que corresponde ao atributo-alvo, ou seja, a propriedade pela qual se deve fazer a predição de um valor. Nesse caso, o atributo é categórico (domínio composto por categorias/classes) e o outro conjunto contém os atributos a serem utilizados na predição do valor, denominados previsores ou de predição. (DE MAGALHÃES, 2020, p. 72)

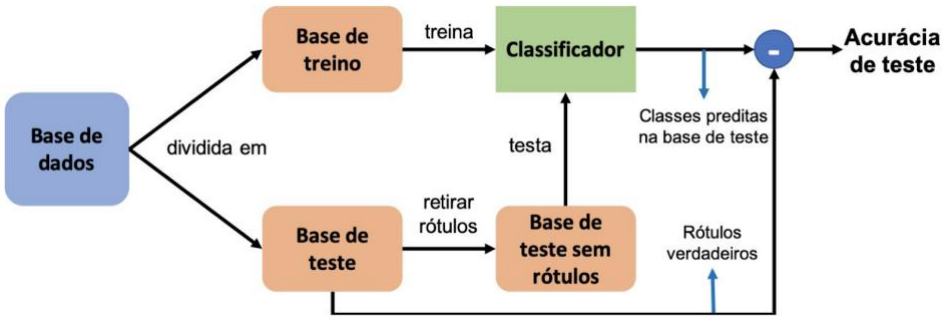
Para Berry e Linoff (2004), a classificação e uma tarefa, na qual as características de um determinado objeto são analisadas e atribuídas a um conjunto predefinido de classes. Consideram que a classificação consiste em rotular os registros em uma determinada classe e nesta tarefa, os objetos a serem classificados são representados por registros em um arquivo ou em uma tabela do banco de dados. Alguns exemplos de classificação são a detecção de sintomas

de doenças, o reconhecimento facial, o reconhecimento de fraudes em sistemas, a classificação de risco de credito, etc.

De acordo com Soares et al. (2008), o processo de classificação é realizado para achar relacionamento entre os objetivo e os atributos preditivos, possuindo, assim, um conhecimento que seja capaz de prever a classe de um registro que ainda não foi classificado.

Segundo Petermann et al. (2006), o processo de classificação tem duas etapas: 1). A primeira etapa é o aprendizado ou treinamento, que é executada com uma base já classificada. Essa primeira etapa é essencial, já que através dele é possível construir um modelo de dados pré-definidos, por meio da analise de tuplas de um banco de dados, no qual cada uma dessas tuplas e pertencente de uma classe; 2). A segunda etapa é o teste, no qual se utilizam registros que não foram utilizados no treinamento. Nessa etapa é feito o mapeamento das entradas e saídas para utilizá-las posteriormente na tomada de decisão em novos registros que ainda não foram classificados. Essa segunda etapa é fundamental para verificar se os o modelo construído na etapa de treinamento foi suficientemente bom para categorizar, com alto índice de precisão, o conjunto de dados ao utilizado na etapa de treinamento. Esta etapa envolve então a predição de classes: os exemplos da base de teste são apresentados para o modelo treinado permitindo, assim, que este realize a predição de suas classes. Ao comparar as classes preditas com as classes verdadeiras da base de teste é possível medir sua capacidade em classificar corretamente exemplos não vistos durante o treinamento. Assim, a Figura 5 resume esse fluxo:

Figura 5 – Fluxograma de problemas de classificação



Fonte: Escovedo et al. (2020) e Silva (2021)

2.2.2.2 Regressão

De acordo com Michie et al. (1994) e Goldschmidt, Passos e Bezerra (2015), a regressão consiste em buscar uma função que mapeie os registros de um banco de dados em um intervalo de valores reais. A tarefa de regressão é parecida à tarefa de classificação, mas com a diferença

de que o atributo-alvo assume valores numéricos. A técnica de regressão é frequentemente usada para modelar relações complexas entre elementos de dados fazendo estimativa de uma variável a partir da outra.

As tarefas de regressão pretendem prever um número contínuo ou um número de pontos flutuantes em termos de programação (chamados de números reais em termos matemáticos). Um exemplo de tarefa de regressão é prever a renda anual da pessoa a partir da idade, local onde mora, profissão e nível escolaridade (SILVA, 2021).

Os algoritmos de aprendizado que relacionam um conjunto de atributos de entrada com uma ou mais saídas contínuas, que podem assumir qualquer valor dentro de um intervalo, são chamados de *algoritmos de regressão*. O exemplo mais básico deste tipo de modelo é o simples ajuste de pontos a uma curva $y = f(x)$, onde pode-se associar um único atributo x com uma saída y. Porem, assim como no caso da classificação, pode ser necessário associar a saída com uma serie de atributos em um vetor x (FONTANA, 2020).

2.2.3 Aprendizado Não Supervisionado

O termo *aprendizado não supervisionado* é usado quando um programa pode automaticamente encontrar padrões e relações em um conjunto de dados, como, por exemplo, analisar e agrupar um conjunto de e-mails, sem que o programa possua qualquer conhecimento prévio sobre os dados (DAS, 2017).

Os casos em que permanece a necessidade de análise dos dados mesmo sem eles estarem rotulados são considerados de aprendizado não supervisionado, onde se aprende sem um professor. Desconhecer a "resposta certa" é o melhor caminho para o aprendizado não supervisionado (DAUMÉ III, 2012).

A classificação realizada através do método não supervisionado não apresenta exemplos da saída desejada, assim o algoritmo deve possuir recursos para superar essa falta de respostas através da apresentação dos resultados classificados de acordo com categorias formuladas por ele próprio (SILVA, 2021).

Segundo Gonçalves (2013), o aprendizado não supervisionado distingue-se do aprendizado supervisionado por não ser fornecido nenhum dado de treinamento. Os algoritmos de aprendizado não supervisionado incluem modelos de redes neurais, *clustering* e análise de componentes independentes. Para categorizar textos, o *clustering* é o tipo de algoritmo de aprendizado não supervisionado de maior interesse.

Segundo Muller e Guido (2017), o aprendizado não supervisionado inclui todos os tipos de Aprendizado de Máquina que não tem saída conhecida, nenhum conhecimento é passado

para o algoritmo de aprendizado, ele somente é alimentado com um conjunto de dados de entrada e solicitado a extrair conhecimento a partir dessas informações.

> No Aprendizado de Máquina não supervisionado, também conhecido como aprendizado por observação e descoberta, a tarefa do algoritmo é agrupar exemplos não rotulados, i.e., exemplos que não possuem o atributo classe especificado. Nesse caso, é possível utilizar algoritmos de aprendizado para descobrir padrões nos dados a partir de alguma caracterização de regularidade, sendo esses padrões denominados clusters. Exemplos contidos em um mesmo cluster são mais similares, segundo alguma medida de similaridade, do que aqueles contidos em clusters diferentes. (MATSUBARA, 2004, p. 41)

Por conseguinte, segundo Cheeseman e Stutz (1996), no aprendizado não supervisionado não há classe associada aos exemplos e o indutor analisa os elementos fornecidos e tenta determinar se alguns deles podem ser agrupados de algum modo, formando grupos de objetos similares). Nesse tipo de aprendizado, o conjunto de treinamento está formado só de exemplos sem nenhum valor associado. O problema está em particionar a amostra de treinamento em *clusters*, através de técnicas de clusterização.

2.2.3.1 Clusterização ou Agrupamento

Segundo Faceli et al. (2017) não existe uma definição formal e única para o termo *cluster*, pelo contrário, existe na literatura uma grande variedade de definições para esse termo. Bárbara (2000, *apud* Faceli et al., 2021, p.178) apresenta algumas definições comuns para *clusters* e as classifica como se detalha a seguir:

> - *Cluster bem separado*: um *cluster* é um conjunto de pontos tal que qualquer ponto em determinado *cluster* está mais próximo (ou é mais similar) a cada outro ponto nesse *cluster* do que a qualquer ponto não pertencente a ele.
> - *Cluster baseado em centro*: um *cluster* é um conjunto de pontos tal que qualquer ponto em dado *cluster* está mais próximo (ou é mais similar) ao centro desse *cluster* do que ao centro de qualquer outro *cluster*. O centro de um *cluster* pode ser um centroide, como a média aritmética dos pontos do *cluster*, ou um medoide (isto é, o ponto mais representativo do *cluster*).
> - *Cluster contínuo ou encadeado* (vizinho mais próximo ou agrupamento transitivo): um *cluster* é um conjunto de pontos tal que qualquer ponto em dado *cluster* está mais próximo (ou é mais similar) a um ou mais pontos nesse *cluster* do que a qualquer ponto que não pertence a ele.
> - *Cluster baseado em densidade*: um *cluster* é uma região densa de pontos, separada de outras regiões de alta densidade por regiões de baixa densidade.
> - *Cluster baseado em similaridade*: um *cluster* é um conjunto de pontos que são similares, enquanto pontos em *clusters* diferentes não são similares. (BÁRBARA, 2000, *apud* FACELI et al., 2021, p.178)

A *Clusterização* tem sido estudado nas áreas de Processamento de Linguagem Natural (PLN), Mineração de Dados (*Data Mining*), Reconhecimento de Padrões e Recuperação de Informações. A Clusterização e um modelo de representação espacial, em que um conjunto de objetos é distribuído em subconjuntos menores, chamados de *clusters* ou agrupamentos

(ZURINI; SBORA, 2011), que usa *machine learning* e PLN para entender e categorizar dados textuais não estruturados.

A *clusterização* ou *agrupamento* é uma técnica "utilizada para segmentar os registros de uma base de dados em subconjuntos ou clusters, de tal forma que os elementos de um cluster compartilhem propriedades comuns que os distingam de elementos nos demais clusters", e seu objetivo principal é maximizar a *similaridade intracluster* e minimizar a *similaridade intercluster*. (GOLDSCHMIDT, PASSOS e BEZERRA, 2015, p. 25). A tarefa de clusterização é diferente da tarefa de classificação já que "cada registro está associado a um ou mais rótulos predefinidos, a clusterização precisa identificar os grupos de dados." (FAYYADD et al., 1996 citado por GOLDSCHMIDT; PASSOS; BEZERRA, 2015, p. 26).

Segundo Jain e Dubes (1988), técnica de agrupamento pretende encontrar uma estrutura nos dados em que os objetos pertencentes a cada *cluster* compartilham alguma característica ou propriedade relevante para o domínio do problema em estudo. Sendo assim, o intuito do agrupamento ou *clustering* é formar grupos diferentes, mas não forçosamente disjuntos, contendo membros muito semelhantes entre eles. Ao contrário do processo de classificação, que segmenta a informação associando-a a grupos já determinados, o *clustering* é uma forma de segmentar informação em grupos não previamente definidos (CAMPOS, 2005).

Markov e Larose (2007) definem o termo *clustering* como um modelo de aprendizado que não usa objetos rotulados e, portanto, não é supervisionado. É uma técnica em que nenhuma suposição é feita sobre os grupos. A diferença da classificação, o processo de clusterização não conta com classes predefinidas e exemplos de treinamento rotulados. Assim, realiza uma forma de aprendizado sem nenhuma forma de supervisão.

Segundo Silva (2021, p. 9), para extrair conhecimento de uma base de dados é essencial "separar estes dados em forma de grupos (*clusters*) que possuam algum significado relevante para a análise". A criação destes grupos é necessária para otimizar a investigação do grande volume de dados. A autora define a *clusterização* como "o método de identificação de grupos de dados semelhantes em um conjunto de dados" e também aponta que a clusterização é uma tarefa que consiste em "dividir a população ou os pontos de dados em vários grupos, de modo que os pontos de dados nos mesmos grupos sejam mais semelhantes a outros pontos de dados no mesmo grupo do que os de outros grupos". Segundo ela, esta abordagem pretende "segregar grupos com traços semelhantes e atribuí-los a *clusters"*.

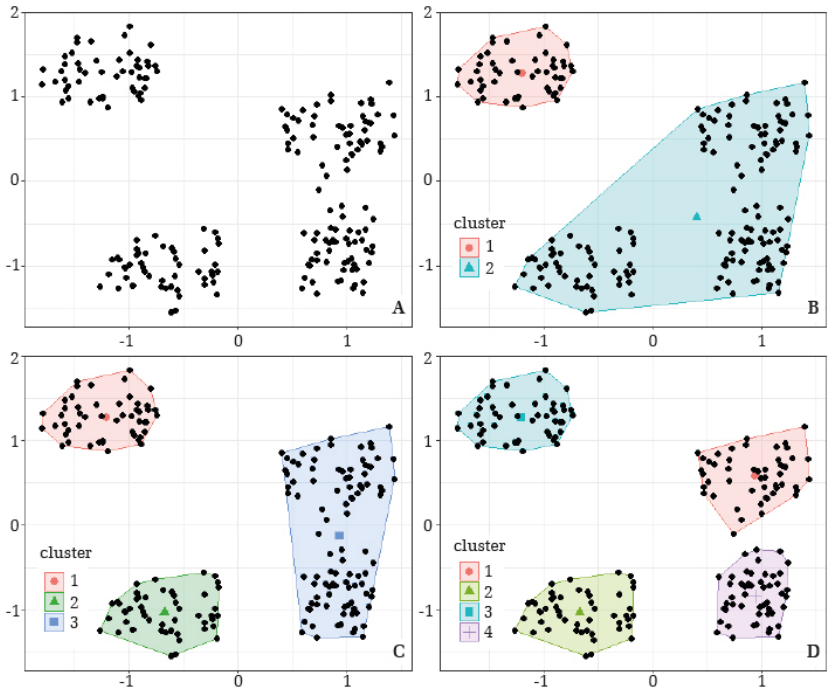
Grosso modo, Silva (2021) divide a clusterização em dois subgrupos: a). *Cluster Rígido*: nesse tipo de cluster cada ponto de dados ou pertence a um cluster completamente ou não; b). *Cluster flexível*: nesse tipo de cluster em vez de colocar cada ponto de dados em um cluster

separado, é atribuída uma probabilidade de que o ponto de dados esteja nesses clusters. Além disso, a autora, assinala que:

> A aplicação da técnica de agrupamento normalmente ocorre quando deseja-se realizar uma análise estatística ou a generalização dos dados de forma exploratória. Logo, esta é uma técnica necessária a analise de informações de mesmas características sem a presença de informações irrelevantes. Neste cenário, tanto a similaridade entre os termos dos grupos quanto a dissimilaridade devem ser investigadas. (SILVA, 2021, p. 9)

Na Figura 6 (A), o grupo apresenta o conjunto de dados originais como entrada. No exemplos (B), (C) e (D) foram formados diferentes grupos de acordo com os critérios estabelecidos e as distâncias euclidianas entre os termos. *K-Means* e *DBScan* são exemplos de algoritmos que utilizam o método de clusterização. Desse modo, no exemplo (B), são formados dois grupos, em (C) três grupos, e em (D) são formados quatro grupos. Os grupos foram formados de acordo com os critérios estabelecidos e as distâncias euclidianas entre os termos.

Figura 6 – (A) Gráfico de dispersão de dados não categorizados. (B) Agrupamento utilizando o algoritmo *K-means* com 2 *clusters* (k=2). (C) Agrupamento utilizando 3 *clusters* (k=3). (D) Agrupamento utilizando 4 *clusters* (k=4)



Fonte: Fernandes e Chiavegatto Filho (2019, p. 5)

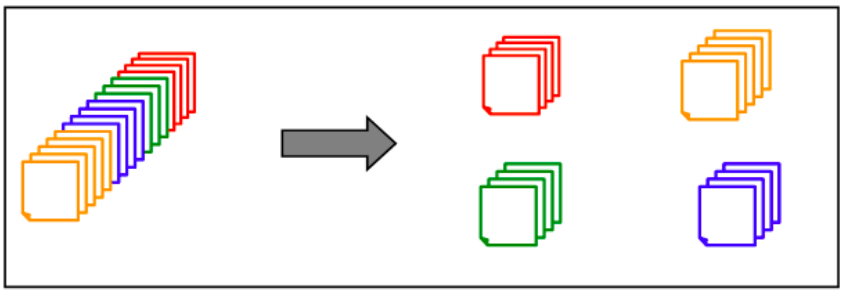
Segundo Ferlin (2008) e Goldschmidt, Passos e Bezerra, (2015, p. 96) “alguns algoritmos de agrupamento requerem que o usuário forneça o número de *k* (número de *clusters* a formar)”. Assim, segundo esse valor, os objetos são separados de forma que os elementos mais similares sejam alocados nos mesmos grupos e os menos similares em grupos diferentes. Portanto, um

sistema de agrupamento pode ser útil no processo de recuperação da informação, pois agrupa os resultados da pesquisa em conjuntos de documentos estreitamente relacionados. O *clustering* pode melhorar o resultado da busca, tornando-a mais relevante, ao se focar em agrupar conjuntos de documentos similares (MARKOV e LAROSE, 2007).

Segundo De Magalhães (2020, p. 81), em uma coleção de documentos ou textos, "o agrupamento também é benéfico, pois facilita o acesso às informações que são interessantes para o usuário, uma vez que o leitor irá acessar somente as notícias dos clusters mais relevantes, ou seja, aquelas que estão de acordo com o interesse do leitor".

Existe um modelo conhecido como *Clustering Model* ou Modelo de Aglomerados (*Clusters*), que usa técnicas de Agrupamento (*Clustering*) de documentos. Seu funcionamento consiste em "identificar documentos de conteúdo similar (que tratem de assuntos parecidos) e armazená-los ou indexá-los em um mesmo grupo ou aglomerado (*cluster*). A identificação de documentos similares em conteúdo dá-se pela quantidade de palavras similares e frequentes que eles contêm. (MORAES e AMBROSIO, 2007, p. 11). A Figura 7 apresenta de forma esquemática o processo de separação de documentos semelhantes em grupos ou *clusters*.

Figura 7 – Separação de documentos semelhantes em *clusters*



Fonte: De Magalhães (2020, p. 81)

A técnica de agrupamento ou clusterização de texto pretende agrupar uma coleção de textos não estruturados em grupos de categorias diferentes para que os documentos no mesmo grupo (*cluster*) descrevam o mesmo assunto (KRISHNA; BHAVANI, 2010). Por conseguinte, a clusterização é um método usado para criar grupos com base em propriedades comuns aos itens, para encontrar padrões, agrupando objetos semelhantes ou organizando-os segundo suas características. Os dados são compilados por similaridades, onde os itens dentro de cada cluster possuem muitas características em comum e muitas diferenças comparadas aos itens de outros grupos (DAS, 2017).

2.2.3.2 Categorização

Segundo Jacob (1992) e De Magalhães (2020), observa-se uma carência de estudos que tratam as diferenças entre os termos *categorização* e *classificação. É mais,* existe uma imprecisão terminológica que eclipsa o entendimento dos pesquisadores e dificulta o aclaração das diferenças entre esses termos no contexto de sistemas de organização da informação. De fato, em repetidas ocasiões, a literatura apresenta as duas palavras como sinônimas. Porém, o processo de agrupamento é uma forma de categorizar, já que na classificação, as classes já existem previamente.

Deepthi e Prasad (2013) consideram que a categorização de texto e o *clustering* de documentos são sinônimos. Pois os autores consideram que essa técnica está relacionada com o raciocínio por trás do agrupamento de dados. De fato Deepthi e Prasad (2013, p. 76) afirmam que "o agrupamento de documentos é basicamente uma técnica mais específica para organização não supervisionada de documentos, extração automática de tópicos e recuperação ou filtragem rápida de informações."

Pelo contrario, Jacob (2004) afirma que embora existam semelhanças óbvias entre *categorização* e *classificação*, as diferenças entre esses termos têm implicações significativas para constituir um ambiente de informação. A autora, considera que a falta de distinção entre esses dois sistemas de organização leva a uma concepção errônea de que eles são sinônimos. Um equívoco que pode ser reforçado pelo fato de que ambos são mecanismos para organizar a informação. Jacob (2004, p. 518), define o termo a *categorização* o processo de dividir o mundo em grupos de entidades cujos membros são semelhantes entre si. Por tanto, "a categorização divide o mundo em grupos ou categorias cujos membros compartilham alguma similaridade perceptível dentro de um dado contexto" (DE MAGALHÃES, 2020, p. 77). Lima (2010, p. 109) aponto que "Categorizar é agrupar entidades (objetos, ideias, ações, etc.) por semelhança."

Apesar de que os sistemas de classificação e categorização sejam mecanismos para estabelecer ordem através do agrupamento de fenômenos relacionados, existem diferenças fundamentais entre eles que influenciam em como esta ordem é efetuada, essas diferenças modificam o contexto de informação estabelecido por cada um desses sistemas (JACOB, 1992).

> Enquanto a classificação tradicional é rigorosa na medida em que determina que uma entidade é ou não é um membro de uma classe particular, o processo de categorização é flexível e criativo e desenha associações não-vinculantes entre entidades - associações que são baseadas não em um conjunto de princípios pré-determinados, mas no simples reconhecimento de similaridades que existem através de um conjunto de entidades. Classificação divide um universo de entidades em um sistema arbitrário de classes mutuamente exclusivas e não sobrepostas que são arranjadas dentro do contexto conceitual estabelecido por um conjunto de princípios estabelecidos. O fato de que nem o contexto nem a composição dessas classes variam é a base para a

> estabilidade de referência fornecida por um sistema de classificação. Ao contrário, categorização divide o mundo da experiência em grupos de categorias nos quais os membros portam alguma similaridade imediata dentro de um dado contexto. Que este contexto pode variar - e com ele a composição da categoria - é a base tanto para a flexibilidade e o poder de categorização cognitiva. (JACOB, 1992 traduzido por GARRIDO, 2011, p. 13).

A teoria clássica das categorias se baseia em três proposições básicas: 1). A intenção de uma categoria é uma representação resumida de uma categoria inteira de entidades; 2). As características essenciais que compõem a intenção de uma categoria são individualmente necessárias e conjuntamente suficientes para determinar a associação dentro do grupo; 3). Se uma categoria (A) estiver aninhada dentro da categoria superordenada (B), os recursos que definem a categoria (B) estão contidos no conjunto de recursos que definem a categoria (A). (SMITH; MEDIN,1981 apud JACOB, 2004, p. 520). Assim, segundo Jacob (2004), a teoria clássica sustenta que pertencer a uma categoria específica (*extensão*) implica a posse do caráter essencial e definidor (*intenção*) da categoria.

2.2.4 Aprendizado por Reforço

O *aprendizado por reforço* é um paradigma computacional de aprendizado em que um agente aprendiz procura maximizar uma medida de desempenho baseada nos reforços (punições ou recompensas) que recebe ao interagir com um ambiente desconhecido (RIBEIRO, 1999).

Segundo Faceli et al. (2021) o aprendizado por reforço da um *feedback* em função do resultado das ações. A autora afirma que:

> O aprendizado por reforço é o que reforça ou recompensa uma ação considerada positiva e pune uma ação considerada negativa. Um exemplo de tarefa de reforço é a de ensinar um robô a encontrar a melhor trajetória entre dois pontos. Algoritmos de AM utilizados nessa tarefa, em geral, punem a passagem por trechos que aumentem o percurso e recompensam a passagem por trechos menores. (FACELI et al., 2021, p.4)

Os métodos de aprendizado por reforço abordam situações onde um agente aprende por tentativa e erro ao atuar sobre um ambiente dinâmico (SUTTON; BARTO, 1998). Sendo assim, não é precisa uma entidade externa que forneça protótipos ou um modelo a respeito da tarefa a ser executada: a única fonte de aprendizado é a própria experiência do agente, cujo objetivo formal é adquirir uma política de ações que maximize seu desempenho geral (MONTEIRO; RIBEIRO, 2004).

2.3 Mineração de dados e mineração de Textos

A mineração de dados (em inglês, *Data Mining)* ou descoberta de conhecimento em bancos de dados, conhecida como *Knowledge Discovery in Databases* (KDD), trata da descoberta de informação útil presente em grandes bases de dados através da identificação de regras e padrões nesses conjuntos. Geralmente, esses conjuntos possuem especificidades, logo, as ferramentas de mineração devem ser utilizadas de forma interativa e não automática gerando diversas fases no processo, tais como: 1). Identificação do Domínio; 2). Preparação dos dados; 3). Mineração; 4). Processamento dos resultados; 5). Utilização dos resultados. (FAYYAD et al., 1996).

O aprendizado de maquina, durante a mineração de dados, geralmente, é combinada com as áreas de bancos de dados e estatística. Embora, existir controvérsias e considerações de similaridade entre as duas áreas, uma das diferenças entre ML e KDD é que na área de estatística há uma busca pelas estruturas que estão por trás dos dados e contribuem para sua formação, enquanto no KDD os dados são a base para o estudo e o interesse nos resultados não depende de estruturas por trás desses dados. Aliás apesar do ML ser parte do núcleo da mineração de dados, elas são coisas diferentes (FAYYAD et al., 1996).

Já o analise inteligente de texto por meio da mineração de texto (em inglês *text mining* ou *text analytics)* é uma tecnologia de inteligência artificial (IA) que usa processamento de linguagem natural para transformar o texto livre (não estruturado) em documentos e bancos de dados em dados normalizados e estruturados adequados para análise ou para acionar os algoritmos de aprendizado de máquina (MILWARD, 2015).

A mineração de texto é amplamente utilizada em organizações orientadas ao conhecimento. O *text mining* é o processo de examinar grandes coleções de documentos para descobrir novas informações ou ajudar a responder a perguntas de pesquisa específicas (MILWARD, 2015).

A mineração de texto identifica fatos, relacionamentos e afirmações que, de outra forma, permaneceriam enterrados na massa de *big data* textual ou do grande volumem de dados textuais. Uma vez extraídas, essas informações são convertidas em um formato estruturado que pode ser analisado posteriormente ou apresentado diretamente usando tabelas HTML agrupadas, mapas mentais, gráficos etc. A mineração de texto utiliza uma variedade de metodologias para processar o texto, sendo uma das mais importantes o Processamento de Linguagem Natural. Os dados estruturados criados pela mineração de texto podem ser

integrados a bancos de dados, data *warehouses* ou painéis de inteligência de negócios e usados para análises descritivas, prescritivas ou preditivas (MILWARD, 2015).

O NLP ajuda às máquinas a "ler" texto (ou outra entrada, como a fala), simulando a capacidade humana de entender uma linguagem natural, como inglês, espanhol ou chinês. O PLN inclui tanto Compreensão de Linguagem Natural quanto Geração de Linguagem Natural, que simula a capacidade humana de criar texto em linguagem natural, por exemplo, para resumir informações ou participar de um diálogo (MILWARD, 2015).

Nesse contexto, igual que a tecnologia, o processamento de linguagem natural amadureceu nos últimos dez anos, com produtos como Siri, Alexa e pesquisa por voz do Google empregando PNL para entender e responder às solicitações dos usuários. Aplicativos sofisticados de mineração de texto também foram desenvolvidos em campos tão diversos como pesquisa médica, gerenciamento de risco, atendimento ao cliente, seguro (detecção de fraude) e publicidade contextual (MILWARD, 2015).

Os sistemas de processamento de linguagem natural de hoje podem analisar quantidades ilimitadas de dados baseados em texto sem fadiga e de maneira consistente e imparcial. Eles podem entender conceitos em contextos complexos e decifrar ambiguidades da linguagem para extrair fatos e relacionamentos importantes ou fornecer resumos. Dada a enorme quantidade de dados não estruturados que são produzidos todos os dias, desde registros eletrônicos de saúde (EHRs) até postagens em mídias sociais, essa forma de automação se tornou fundamental para analisar dados baseados em texto com eficiência (MILWARD, 2015).

2.4 Categorização de Textos

Segundo Silva (2021, p. 11), os textos são uma "forma natural para armazenar informação" e "sua mineração é uma tarefa difícil e multidisciplinar que possui potencial comercial". Tanto os pesquisadores da área de *machine learning* e recuperação de informação como os desenvolvedores de aplicações são "profissionais que precisam trabalhar com grandes quantidades de documentos, por isso a classificação de textos é objeto de interesse para ambas as classes". Essa tendência de foco é "potencializada pela conectividade aumentada e disponibilidade de diversas bases de dados" que contem "documentos de diferentes tipos e com variadas quantidades de informação".

Cabe diferenciar entre as duas variações da categorização de textos que existem: *classificação* e *clusterização*. A categorização de textos por meio da classificação ou categorização de documentos consistem em classificar um documento em um conjunto pré-

especificado de categorias. Assim, dado um conjunto de categorias, assuntos ou tópicos e uma coleção de documentos de texto, a classificação é o processo de encontrar a categoria correta associada a cada documento. Já a categorização por meio de clusterização (*clustering)* é uma das técnicas mais usadas no processo de mineração de dados para descobrir grupos e identificar distribuições de padrões ocultos em uma base de dados. A categorização permite agrupar documentos similares em coleções de texto sem que se tenham informações previas sobre os grupos, portanto não há classes ou rótulos previamente definidos para o treinamento de um modelo. (FELDMAN; SANGER; PRESS, 2007)

Segundo Rajaraman e Ullman (2011), a clusterização *ou clustering* é o processo de examinar uma coleção de "pontos" e agrupar os pontos em "clusters" em função de alguma medida de distância. Os pontos no mesmo *cluster* têm uma pequena distância um do outro e os pontos em diferentes *clusters* estão a uma grande distância um do outro. A clusterização pode ser usada para formar grupos de documentos com conteúdo similar, os chamados *clusters*, com conteúdo mais similar dentro do *cluster* e menos similar entre os diferentes clusters (HOTHO, NÜRNBERGER e PAAβ, 2005).

> Se o objetivo do estudo é determinar em qual categoria os documentos melhor se enquadram, e já existir um conjunto de dados consistente previamente categorizado, costuma-se utilizar a abordagem de classificação, ao invés da clusterização. Há duas abordagens para realizar a classificação de documentos: uma é centrada no conhecimento de especialistas, onde e desenvolvida uma solução em forma de regras de classificação, e outra e pelo aprendizado de máquina, na qual um processo indutivo executa um classificador com base no aprendizado obtido a partir do conjunto de dados previamente categorizado. (SILVA, 2021, p. 11).

Num momento incipiente, em meados da década de 70, as aplicações construídas pretendiam indexar automaticamente os textos para os sistemas de recuperação de informações booleanas. Visando predefinir os termos de indexação, podemos considerar uma forma de classificação onde se use analogamente tais expressões como classes dos sistemas atuais. Com o aumento da relevância e quantidade de documentos nas décadas de 80 e 90, surgiram aplicações de classificação de texto como a classificação de patentes, a classificação de paginas Web e os filtros de notícias (SEBASTIANI, 2005).

Sebastini (2005), também aponta que a recuperação de informação e pesquisa em bases de dados com grande quantidade de textos pode ter duas aproximações: Uma na que se constroem ferramentas robustas para pesquisa nessas bases; e outra onde na solução se constroem ferramentas robustas para estruturar esses dados simplificando a pesquisa.

2.5 Pré-processamento de Textos

O pré-processamento é etapa indispensável para aumentar a precisão dos algoritmos de ML aplicados a categorizações textuais (SILVA, 2021). Concretamente, o pré-processamento de textos é uma etapa que consiste em transformar documentos textuais em um formato estruturado, como por exemplo uma tabela atributo-valor, para poder aplicar algoritmos de aprendizado de maquina que permitam extrair conhecimento dessa informação textual (MITCHELL et al., 1997). Entretanto, essa transformação e um processo demorado e custoso "que deve ser feito com cuidado para que o conhecimento adquirido posteriormente seja útil para o usuário final" (SILVA, 2021, p.13). Por exemplo, a abordagem *bag of words (BOW)* é uma das mais utilizadas em ML para tornar dados estruturados em uma representação atributo-valor, na qual a frequência das palavras (*termos*), independentes do seu significado ou contexto, são contadas. A partir do cômputo é gerada uma tabela cujas entradas contem informações relacionadas à frequência de cada palavra (SILVA, 2021).

> Na transformação de documentos textuais em tabelas atributo-valor, existem alguns métodos para auxiliar na redução do número de atributos (redução de dimensionalidade) visando melhorar a relevância da informação para a classificação do texto. Esses métodos são brevemente descritos a seguir. (SILVA, 2021, p.13).

Os sistemas de ML quase nunca conseguem trabalhar diretamente com informações digitais dispersas em documentos textuais devido ao formato na o estruturado desses textos. Portanto, é preciso desenvolver sistemas capazes de tratar essa informação não estruturada e transformá-las em informação estruturada, especificamente uma tabela atributo-valor, para possibilitar o uso de sistemas de aprendizado já existentes (SILVA, 2021).

2.5.1 Tokenização

Quando é analisada a quantidade de elementos que um texto, é possível identificar durante o processo de tokenização (*tokenizing*) distintos elementos recorrentes tais como palavras, pontuação, ou símbolos. Estes elementos são separados e listados, apenas as palavras são interessantes para os processos posteriores. Portanto, a pontuação, os espaços ou os delimitadores de tabulação servem como separadores durante a formação dos *tokens*, que são instâncias compostas por uma sequência de caracteres em um documento em particular que é agrupada na forma de uma unidade semântica útil para o processamento (DEBARR e WECHSLER, 2009).

2.5.2 Remoção de *stopwords*

Segundo Wilbur e Sirotkin (1992), uma *stopword* é aquela encontrada quando se consulta um documento, mas não é relevante se a consulta está ou não relacionada com o documento. Os autores também destacam que a remoção de *stopwords* permite liberar espaço nos vetores e, as vezes, melhorar a *performance* do classificador ajudando na recuperação de informação. A maioria dos termos utilizados são relativamente óbvios e pertencem a um conjunto reduzido de palavra. Porém, cabe enfatizar que algumas palavras que aparecem nos documentos têm maior valor sintático do que o valor semântico que interessa para o classificador.

Por conseguinte, algumas palavras não são relevantes durante o processo de classificação e devem ser removidas na etapa de pré-processamento. Essas palavras são artigos, conectores, preposições, pronomes e palavras comuns de grande ocorrência e com pouco valor para a classificação que constituem uma lista dos *tokens* com pouca relevância. Esses *tokens* variam segundo a língua do texto e servem como parâmetro para formar o dicionário de palavras definitivas para as demais etapas (SILVA, 2021).

2.5.3 *Stemming*

Silva (2021) o *stemming* é uma técnica que permite reduzir a dimensionalidade sem perder informações importantes do conjunto de dados inicial. A autora também ressalta que, em função do o algoritmo usado para *stemming*, o resultado pode ser um termo sem validade gramatical, pois se pode cortar letras do final ou reduzir as partes da palavra original, tornando-a sem sentido para a leitura. Por exemplo, “ao aplicar *stemming* nas palavras “química”, “químico” ou “químicos”, o algoritmo converte-as para o *stemm* “quimic”.

O algoritmo responsável pelo *stemming* remove as variações de uma palavra através da redução para uma mesma raiz ou uma forma comum. Tal ação é considerada de grande importância pelos pesquisadores da área de recuperação de informações, pois palavras que possuem variações pequenas, como tempo verbal ou mudanças entre plural e singular, ao passarem por essa transformação para que seja reduzida a ocorrências de *tokens* com sentidos similares, aumentam a e ciência da atribuição dos pesos para os termos (LOVINS, 1968).

Na analise morfológica automática, os sufixos da palavra podem ter maior interesse imediato que a raiz da palavra, assim como o cômputo da frequência dos termos pode ser relevante para analise matemática ou estatística de um corpus, pois requerem expressões

idênticas. Porém, alguns problemas linguísticos são comuns para qualquer algoritmo que realize *stemming* não importando sua finalidade (LOVINS, 1968).

2.6 Conversão de valores simbólicos para numéricos

A manipulação de tipos de atributos diferentes, simbólicos e numéricos, depende da capacidade e necessidade da técnica de ML usada, já que algumas técnicas, como *árvores de decisão* podem manipular valores simbólicos enquanto outras, como por exemplos as redes neurais, podem manipular apenas uma representação numérica de um valor simbólico (NEVES, 2003).

Devido a que todas as técnicas de mineração de dados ou de ML, podem manipular dados numéricos, mas algumas não podem manipular dados simbólicos, torna-se preciso aplicar algum método de transformação de valores simbólicos em uma representação numérica apropriada (PYLE, 1999).

A seguir se apresentam as abordagens habitualmente utilizadas para converter palavras em dados numéricos.

2.6.1 TF-IDF

A Frequência de Aparição de um Termo ou TF-IDF (*Term Frequency-Inverse Document Frequency)* proposta por Jones (1972; 1979) é uma das medidas usadas tanto em ML como em mineração de textos para dar peso às palavras presentes nos documentos de uma coleção. Segundo Silva (2021), nesta medida, a importância de uma palavra é mensurada de acordo com a frequência com que essa palavra aparece em um documento, logo quanto mais frequente for uma palavra em um documento, maior sua importância. Entretanto, se na coleção como um todo, aquela palavra for muito frequente, ela não é tão relevante assim. Portanto, essa medida destaca a pouca frequência do termo (IDF) na coleção e a frequência dele (TF) no documento. Dado um termo *t*, a frequência do termo é calculada pela fórmula a seguir:

$$Tf(t) = \frac{Numero\ de\ ocorrências\ do\ termo\ t\ no\ documento}{Quantidade\ de\ termos\ no\ documento} \quad (1)$$

Onde a divisão pela quantidade ou número de termos no documento se deve a que há maior probabilidade de existir um número maior de ocorrências de um termo em um texto grande. Portanto, essa normalização é feita calculando a frequência relativa. A Frequência Inversa do Documento para um Termo (IDF) é uma medida da importância do termo na coleção.

Termos muito frequentes têm seu peso diminuído e o inverso ocorre para termos raros. Assim, o IDF é dado por:

$$idf(t) = loge\ \frac{Número\ total\ de\ documentos}{Quantidade\ de\ termos\ no\ documento} \quad (2)$$

2.6.2 *Bag of Words*

Para McTerar, Callejas e Griol (2016), o modelo de *Bag of Words (BoW)* é uma técnica simples e popular utilizada para representar texto em Processamento de Linguagem Natural (PLN). O modelo *Bag of words* (Saco de Palavras) é comumente usado em métodos de classificação de documentos onde a frequência de ocorrência de cada palavra é usada como um recurso para treinar um classificador. O modelo *Bag of words*:

> utiliza um conjunto de palavras do texto como entrada nos algoritmos classificadores, sendo cada palavra diferente um atributo do conjunto de dados formado. A instância mais simples de BOW é a seleção de todas as palavras do texto, tornando a dimensionalidade do conjunto de dados, e consequentemente do problema, igual à quantidade de palavras diferentes presente no texto. Desconsideram-se do texto a gramatica e até mesmo a ordem das palavras. (SILVA, 2021, p. 15)

Finalmente, cabe destacar que Scheicher et al. (2016) demonstram que em cenários em que a classificação automática é mais complexa o uso da BoW pode apresentar resultados mais baixos.

2.6.3 Atributos *N-Gramas*

Um *n-grama* é uma sequência de *n* itens dentro de uma frase. Os itens podem ser letras, sílabas, palavras, classificação gramatical das palavras ou qualquer outra base. Habitualmente, se chama de unigrama a um *n-grama* de tamanho 1, bigrama quando é de tamanho 2, trigrama se têm tamanho 3 e de 4 em diante é chamado de *n-grama* (VILELA, 2011). Portanto, para uma sequência de palavras, por exemplo "Departamento de Computação da USP", um bigrama de palavras seria: "# Departamento", "Departamento de", "de Computação", "Computação da", "da USP" e "USP#".

Segundo Braga (2010), a obtenção de tabelas atributo-valor baseadas em *bag-of-words* é um dos métodos mais simples de descrição de textos. A maior crítica relacionada a essa representação vem dada pelo fato de ela considerar cada palavra isoladamente, de modo que não consegue capturar bem os conceitos que estão expressos em termos compostos por mais de uma palavra. Assim, foram propostas descrições baseadas em *n-gramas*, ou seja, grupos de *n*

palavras consecutivas que são encontradas nos textos. Os unigramas (n=1) constituem os mesmos atributos obtidos pelo método *bag-of-words*.

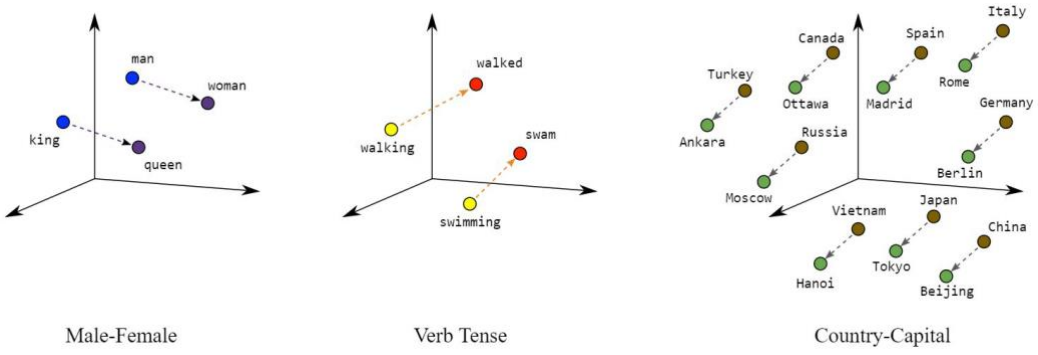
2.6.4 *Word Embedding*

Word Embedding (WE) é um método que consiste em uma abordagem atual usada para representar textos no processamento de linguagem natural. Essa abordagem é capaz de identificar as informações semânticas latentes da linguagem, capturando a concorrência de padrões das palavras. Contudo, no WE cada palavra é representada por um vetor numérico, diferente do BOW que representa cada palavra por um número (MIKOLOV et al., 2013), o que permite reduzir a alta dimensionalidade nos dados e o raciocínio sobre o uso e significado das palavras (SHUANG et al., 2019).

Segundo Silva (2021), WE é um algoritmo semelhante ao *Bag of Words*, no qual as palavras são representadas em vetores grandes e esparsos, só que o WE utiliza vetores densos de tamanho *x* o que são capazes de armazenar informações sobre o significado e contexto dos documentos. Cada palavra é representada por um ponto em um espaço multidimensional chamado de *embedding space*. Durante o treinamento, as palavras são definidas por elas mesmas e pelas palavras que a acompanham.

Na Figura 8, as seguintes visualizações de *embeddings* reais mostram relações geométricas que apresam relações semânticas como a relação entre os gêneros das palavras (feminino e masculino), entre os países, suas capitais e os tempos verbais.

Figura 8 – Exemplos de um enfoque *Word Embedding*



Fonte: GOOGLEDEVELOPERS (2021) e Silva (2021)

Segundo Junior (2007), embora a identificação de *tokens* seja uma tarefa simples para o ser humano, ela pode se tornar complexa quando executada por um computador quando os delimitadores assumirem diferentes papeis. Por exemplo o “ponto”, que é utilizado para marcar

o fim de uma sentença ou na composição de uma abreviatura. Uma sentença pode ter seus vocábulos organizados em diferentes ordens as quais podem inclusive modificar seu significado, e as palavras que apresentam essa relação são consideradas como *colocações*. Por exemplo, “provocar essa destruição” e “essa destruição foi provocada”, por isso é interessante que a tokenização seja composta por todo o conjunto de palavras que podem traduzir uma ideia diferente (SOARES et al., 2008).

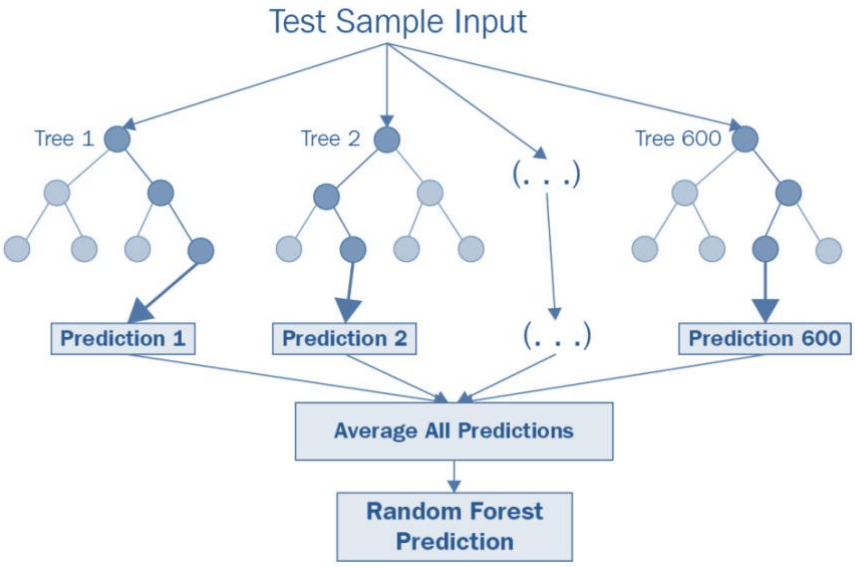
2.6.5 *Random Forest*

Segundo Breiman (2001), o *Random Forest (RF)* é um algoritmo de ML baseado em árvore de decisão. Uma Árvore de Decisão é usada para classificar uma instância baseada em variáveis decisórias seguindo o caminho da raiz até as folhas. Os atributos (variáveis decisórias) são utilizadas para tomar decisões, e suas respostas formam ramificações ou caminhos específicos em uma árvore.

Pacifico, Britto e Ludermir (2020) definem o termo *Árvore de Decisão* como uma estrutura de fluxograma semelhante a uma árvore. A estrutura é usada para aplicar um conjunto de regras, onde cada nó interno é responsável por testar um atributo, cada ramificação representa os resultados do teste de determinado nó e cada folha indica o rótulo da classe. Essa árvore é gerada por meio de um processo de divisão, norteado por uma medida de satisfação conhecida como *entropia*. É utilizada para decidir que atributos devem ser levados em conta na hora de dividir o conjunto de dados em uma determinada iteração do método.

Segundo Britto e Pacco (2020), o RF é um método que combina um conjunto de Árvores de Decisão para evitar o impacto que os ruídos e *outliers* podem ter no resultado de uma única Árvore, o que torna o classificador muito mais robusto. O algoritmo combina diversas Árvores, agregando os votos de diferentes estimadores para decidir a classe final do dado de teste. A Figura 9 apresenta um exemplo do método RF.

Figura 9 – Exemplo de Predição *Random Forest*



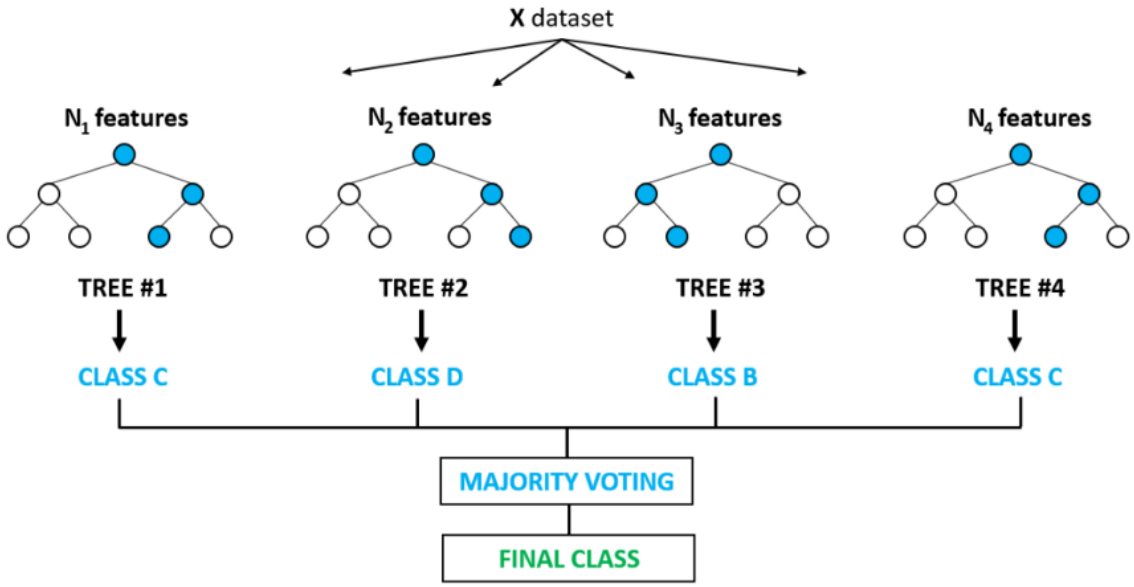
Fonte: Lorena e Carvalho (2007) e Wesner (2020, p. 15)

Segundo Berrondo Urruzola (2020) existem duas maneiras principais de escolher as amostras para o treinamento das múltiplas árvores de decisão:

- *Bagging*, também conhecido como agregação de *Bootstrap* (utilizado en el RF).
- *Boosting* (utilizado no *Gradient Boosting Machine*).

O *Bagging* (Breiman, 1996), um método usado por defeito no RF (Figura 10), ele funciona treinando as árvores de decisão em subconjuntos escolhidos aleatoriamente do conjunto de dados ou *dataset*, usando a substituição para criar novos subconjuntos. Assim, a variância do modelo é significativamente reduzida, uma vez que um grande problema de árvores de decisão simples é sua grande sensibilidade ao ruído de dados. Além disso, segundo Berrondo Urruzola (2020), se as árvores utilizadas no modelo não estiverem correlacionadas umas com as outras, um modelo mais robusto será alcançado diminuindo o viés do modelo. Deve-se notar também que a falta de correlação entre as árvores é de suma importância para que o modelo seja eficaz. Pou o que, além de usar um subconjunto em vez de todos os dados, o RF também realiza o *Feature Bagging*, onde em cada divisão da árvore apenas algumas variáveis são consideradas. Dessa forma, evita-se que o impacto das variáveis mais fortes no modelo seja muito grande, e é possível que as variáveis de menor força possam aparecer mais no modelo.

Figura 10 – Representação gráfica de *Random Forest* para Classificação



Fonte: Berrondo Urruzola (2020, p. 12)

O funcionamento do *Boosting* (Schapire, 1990) é semelhante ao *Bagging*, mas com a diferença de que as amostras ou subconjuntos de dados são atribuídos um peso. A cada vez que um deles é classificado mal, seu peso cresce, e aqueles que têm mais peso são aqueles que terão mais presença nas árvores. Dessa forma, dá mais ênfase a casos difíceis do que em casos fáceis, obtendo um modelo mais robusto. Ao contrário do *Bagging*, onde o treinamento pode ser executado em paralelo, o *Boosting* deve necessariamente ser realizado de forma sequencial.

Por fim, a resposta final do modelo é calculada pela obtenção da média de todas as previsões em problemas de regressão ou utilizando a técnica de votação majoritária em problemas de classificação (Berrondo Urruzola, 2020).

Segundo Silva (2021, p. 19), "para fazer a categorização, pode-se classificar cada atributo como uma *feature* e cada resposta seria uma possível característica dessa *feature"*. Também aponta que uma árvore de decisão poderia ser construída a partir de um subconjunto de dados já classificados. Devido ao dinamismo semântico da língua, a linguagem natural pode variar consideravelmente até quando são utilizadas as mesmas palavras em contextos distintos. Portanto, a construção de uma única árvore não seria suficiente para criar um modelo eficiente de aprendizado de maquina. Por este motivo,

> o RF propõe que sejam criadas diversas árvores de decisão baseadas em subconjuntos aleatórios de uma base de dados. Dessa forma, a classificação passa a ser baseada em uma "floresta" de decisão, e não somente em uma arvore. Se houver atributos nas árvores de decisão, pode ocorrer *overfitting*, tornando o classificador pouco genérico, podendo ocorrer erros maiores que o esperado. Para solucionar esse problema, a

> profundidade de cada árvore é diminuída para maximizar os resultados e generalizar a solução. (SILVA, 2021, p. 19),

2.6.6 Classificação *Naive Bayes*

Segundo Amaral (2016) a técnica de classificação *Naive Bayes* (NB) é um algoritmo de classificação baseado no *Teorema de Bayes*, baseado na teoria das probabilidades, que assume que cada atributo influenciar a de forma independente a classificação de uma nova instância. Na abordagem *Naive Bayes*, pressupõe-se que, dadas as características, existe independência entre os preditores. Portanto, o NB assume que a presença de uma característica particular em uma classe não está relacionada com a presença de qualquer outro atributo. Amaral (2016) também aponta que durante a etapa de treinamento, estabelece-se uma tabela de valores e deve se adjudicar um peso para cada atributo em cada uma das classes de classificação. E quando uma nova instância é submetida à classificação, o modelo somará os pesos de cada atributo em cada uma das classes. Dessa forma, a classe que somar o maior peso, será a classe classificadora do novo item.

A seguir se apresenta a fórmula matemática utilizada pelo NB para cálculo de probabilidades condicionais, descrevendo a probabilidade de um evento com base no conhecimento prévio das condições que podem estar relacionadas com esse evento:

$$P(x/c) = \frac{P(x/c)\,.\,P(c)}{P(x)} \quad (3)$$

Onde:

P(x | c) é a probabilidade posterior da classe alvo (probabilidade de *x* acontecer dado *c*), P(*c*) a probabilidade original da classe, P(x | c) é a possibilidade de que a probabilidade da classe preditora seja dada (probabilidade de c acontecer dado *x*), P(*x*) é a probabilidade original do preditor (probabilidade de *x* acontecer). Segundo, Silva (2021), para a classificação de texto, os *xis* são as palavras individuais do documento e o classificador gerará a classe com a probabilidade posterior máxima.

2.6.7 K-vizinhos mais próximos

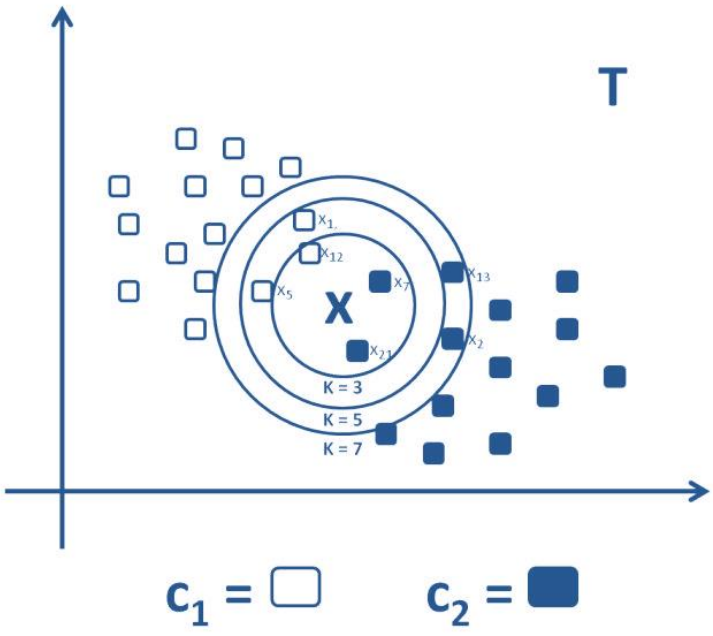
Fukunaga e Narendra (1975) propuseram o K-vizinhos mais próximos ou K *Nearest Neighboors* (k-NN), que definem como um dos classificadores mais simples de ser implementado e que é capaz de obter bons resultados.

> k-NN é um classificador onde o aprendizado é baseado na analogia. O conjunto de treinamento e formado por vetores n-dimensionais e cada elemento deste conjunto representa um ponto no espaço n-dimensional. Para determinar a classe de um elemento que não pertença ao conjunto de treinamento, o classificador k-NN procura *k* elementos do conjunto de treinamento que estejam mais próximos deste elemento desconhecido, ou seja, que tenham a menor distância. (SILVA, 2021, P. 21)

Segundo Faria e Monteiro (2015), os *k* elementos são chamados de k-vizinhos mais próximos. Devem ser verificados quais são as classes desses *k* vizinhos e a classe mais frequente será atribuída à classe do elemento desconhecido. *K* geralmente é um número ímpar quando o número de classes é 2 e quando *K = 1*, o algoritmo é conhecido como o algoritmo do vizinho mais próximo. Supondo que P1 é o ponto para o qual o rótulo precisa prever, deve-se encontrar o ponto mais próximo de P1 e, seguidamente, o rótulo do ponto mais próximo atribuído a P1.

A Figura 11 mostra um modelo de classificação com K-NN e a influência do valor de *k* na classificação de novas instâncias. O conjunto T é usado para classificar a instância desconhecida *x*. Dada a configuração espacial dos exemplos de treinamento, valores diferentes de *k* classificam o novo exemplo com classes diferentes. Para os valores *k = 3* e *k = 7*, a instância é classificada como pertencente à classe c2. Já para *k = 5*, a classe C_1 é atribuída à nova instância. Isso mostra a influência da escolha do valor de *k* no algoritmo K-NN (MOURA, 2015)

Figura 11 – Classificação de uma instância desconhecida com k-NN.



Fonte: Moura (2015, p. 32)

2.6.8 Cálculo da distância

Existem várias métricas para calcular a distância e a escolha de qual usar varia dependendo do problema. Uma das métricas mais utilizadas é a distância Euclidiana, descrita pela equação 4 (SILVA, 2021).

$$D_E(p,q) = \sqrt{(p_1 - q_1)^2} + \cdots + (p_n - q_n)^2 = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2} \quad (4)$$

2.6.9 Distribuição Normal

Segundo Silva (2021), a *distribuição normal* ou *distribuição gaussiana* é uma das mais importantes distribuições contínuas. Sua importância se deve a vários fatores, tais como o *teorema central do limite,* que é o resultado principal em aplicações práticas e teóricas. O *teorema central do limite* garante que inclusive os dados que não possuem distribuição de acordo com uma normal, a média deles converge para uma distribuição normal conforme o número de dados aumenta.

Na prática, diversos estudos têm como resultado uma distribuição normal, como por exemplo, o estudo da altura de uma determinada população, pois, geralmente, a altura segue uma distribuição normal (SILVA, 2021). Assim, a definição é dada pela fórmula 6 onde uma variável aleatória contínua *x* têm distribuição normal se a sua função densidade de probabilidade for dada por:

$$f(x) = \frac{1}{\sqrt{2(\pi\sigma^2)}} exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right], \quad x \in (-\infty, \infty) \quad (5)$$

2.6.10 Testes Paramétricos e Não Paramétricos

Segundo Campos (2002), Os termos paramétrico e não-paramétrico referem-se à média e ao desvio-padrão, que são os parâmetros que definem as populações que apresentam distribuição normal.

Campos (2002) considera que os testes estatísticos se podem classificar em dois grandes grupos, em função de se fundamentam ou não os seus cálculos na premissa de que a distribuição de frequências dos erros amostrais é normal, as variâncias homogêneas, os efeitos dos fatores de variação são aditivos e os erros independentes. Se estas condições ocorrerem, provavelmente a amostra seja simétrica e terá um ponto de máximo (centrado no intervalo de classe onde está a média da distribuição), seu histograma de frequências terá um contorno paramétrico e o desenho terá uma forma de sino da curva normal.

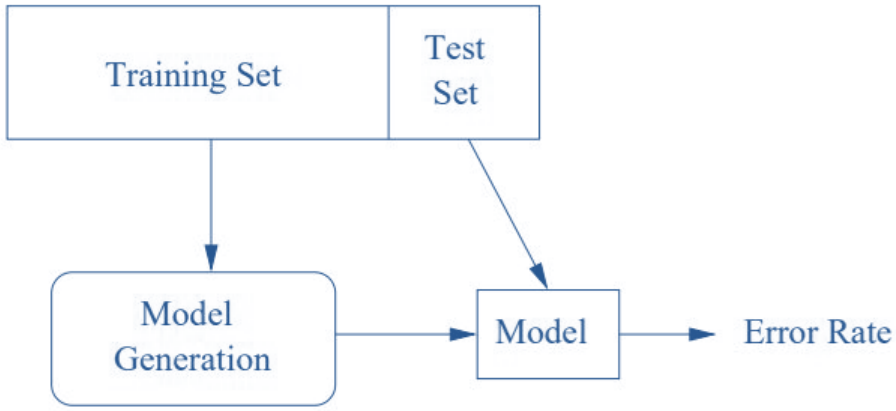
Campos (2002) também destaca que o cumprimento desses requisitos condiciona o pesquisador aos testes paramétricos, uma vez que, se forem preenchidos, ele poderá utilizar a estatística paramétrica, cujos testes são em geral mais poderosos do que os da estatística não - paramétrica, e por conseguinte devem ter a preferência do investigador quando o seu uso for permitido..

2.7 Métodos de Avaliação de modelos de classificação (auditoria de inferências e ações realizadas)

Segundo Silva (2021), as medidas mais comumente utilizadas para avaliar os resultados da classificação de texto são: revocação (*recall*), precisão (*precision)*, acurácia (*accuracy*) e F1-Score. Essas métricas utilizam como base os Verdadeiros Positivos (VP), Verdadeiros Negativos (VN), Falsos Positivos (FP) e Falsos Negativos (FN).

Silva (2021) também aponta que quando o modelo prevê um caso positivo corretamente, é um caso de Verdadeiro Positivo; mas no caso em que o modelo determine que uma classe é verdadeira, quando na verdade não é, estamos diante de um caso de Falso Positivo. Já, quando o modelo prevê um caso negativo corretamente, temos um caso de Verdadeiro Negativo. Mas o caso oposto também pode ocorrer de maneira que quando o modelo diz que a classe não é verdadeira, na verdade é, portanto temos um caso de Falso Negativo. Na Figura 12, observa-se como o conjunto de treinamento ajuda a construir o modelo e o conjunto de testes o valida.

Figura 12 – Arquitetura de treino e teste para construir e validar modelo



Fonte: Rajaraman, Leskovec e Ullman (2014, p. 444)

Na Figura 12, é assumido que todos os dados são adequados para o treino, isto é, as informações de cada classe, por exemplo as informações de cada categoria dos e-mails ou noticias. Segundo Silva (2021), uma fração do conjunto de dados disponível é separada para o

conjunto de testes e os dados restantes devem ser utilizados para verificar se o classificador apresenta resultados afines à proposta. A classe de cada elemento de teste, considera-se quantitativa se o modelo apresenta uma taxa de erro menor que a pré-determinada. Caso a taxa de erro seja maior que a esperada, deve-se utilizar técnicas especificas de cada algoritmo, como, por exemplo, o uso de balanceamento das classes, diminuição nos ruídos dos dados de treinamento ou aumento do conjunto de dados para a etapa de treinamento do modelo.

Nesse contexto, cabe destacar as auditorias já que permitem registrar todas as evidências documentadas das sequências de atividades que afetou em qualquer momento uma operação, processo ou evento específico.

2.7.1 Validação Cruzada

A validação cruzada (*Cross Validation*) é uma técnica para avaliar a capacidade de generalização de um modelo, a partir de um conjunto de dados. A *validação cruzada* é um procedimento estatístico comumente utilizado no âmbito de ML e de mineração de dados. Procedimento que consiste na partição de uma amostra de dados em subconjuntos enquanto os demais são guardados para uma subsequente confirmação e validação da análise inicial (SILVA, 2021; DEVIJVER e KITTLER, 1982). Com esse procedimento, evita-se resultados tendenciosos ao se utilizar apenas um espaço amostral (TAVARES; LOPES; LIMA, 2007).

Pode-se diferenciar entre conjunto de dados inicial que é chamado de *conjunto de treinamento* e os outros conjuntos de dados que são chamados de *conjuntos de validação* ou *teste*. Caso utilizem-se 3 partições para a validação cruzada, significa que o conjunto de dados será separado em 3 conjuntos distintos e no final (após a etapa de treino) deverá ser feita uma avaliação sobre as métricas. O procedimento é repetido de forma circular ou cíclica para todas as partições, onde cada uma é usada uma única vez para o treinamento. O resultado final pode ser a media das avaliações. Com esse procedimento, evita-se resultados tendenciosos ao se utilizar apenas um espaço amostral (TAVARES; LOPES; LIMA, 2007).

Existem dois tipos de validação cruzada: validação cruzada exaustiva e não exaustiva. Os métodos de validação cruzada exaustiva são métodos de validação cruzada que aprendem e testam todas as maneiras possíveis de dividir a amostra original em um conjunto de treinamento e validação. Esses métodos de validação cruzada exaustiva podem ser: Validação cruzada leave-p-out (LpOCV, das siglas em inglês *Leave-p-out cross-validation*) e validação cruzada deixando um fora (LOOCV, as siglas em inglês de *Leave-one-out cross-validation*).

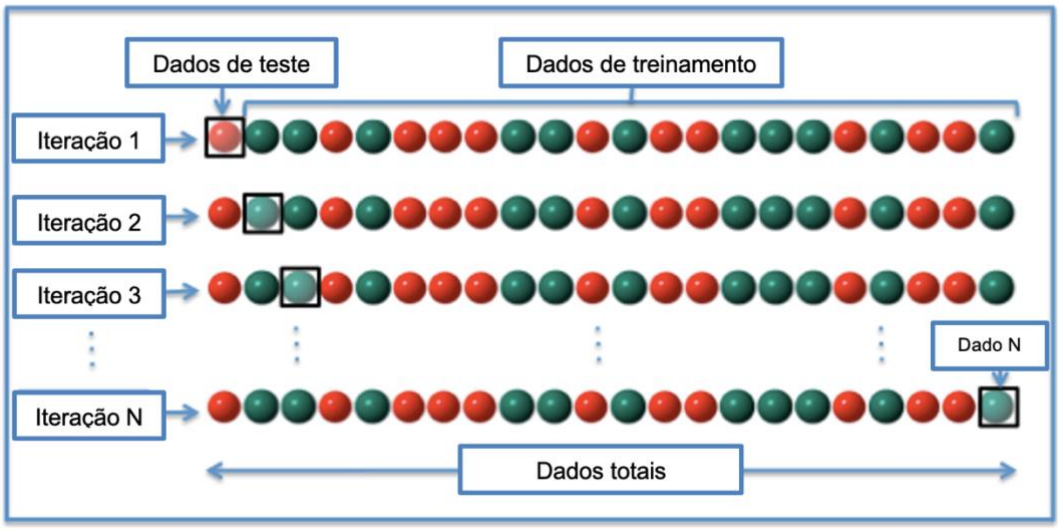
A LpOCV implica o uso de *p* observações como o conjunto de validação e as observações restantes como o conjunto de treinamento. Isso é repetido em todas as maneiras de cortar a amostra original em um conjunto de validação de p observações e um conjunto de treinamento. (CELISSE, 2014). A validação cruzada de LpO requer treinamento e validação do modelo C_p^n vezes, onde n é número de observações na amostra original, e onde C_p^n é o coeficiente binomial[1] . Para p > 1 e para n moderadamente grande, LpO CV pode se tornar computacionalment inviável. Por exemplo, com $n = 100$ observações e $ep = 30$, $C_{30}^{100} \approx 3 \times 10^{25}$. Uma variante da LpOCV com $p = 2$ conhecida como validação cruzada leave-pair-out tem sido recomendada como um método quase imparcial para estimar a área sob a curva ROC de classificadores binários (AIROLA et al., 2011)

A LOOCV é um caso particular de validação cruzada *LpO* (LpOCV) com $p = 1$. O processo é semelhante à re-amostragem de canivete conhecida em inglês como *Jackknife resampling*. No entanto, com validação cruzada, calcula-se uma estatística nas amostras deixadas de fora, enquanto com *jackknifing*, calcula-se uma estatística apenas das amostras mantidas. A LOOCV requer menos tempo de computação do que a LpOCV, porque existem apenas $C_p^n = n$ iterações ou repetições do processo em vez de C_p^n . No entanto, n observações ainda podem exigir um tempo de computação bastante grande, pelo que tem casos em que outras abordagens, como a *validação cruzada k-fold* ou *validação cruzada k-iterações* (*k-fold cross-validation*), podem ser mais apropriadas. (MOLINARO; SIMON; PFEIFFER, 2005).

LOOCV envolve a separação dos dados para que para cada iteração tenhamos uma única amostra para os dados do teste e todos os demais formando os dados de treinamento. A avaliação é dada pelo erro, e nesse tipo de validação cruzada o erro é muito baixo, mas por outro lado, no nível computacional é muito caro, uma vez que um alto número de iterações tem que ser realizada, tanto quanto *N* amostras que temos e para cada um analisar os dados tanto treinamento quanto teste (ELKAN, 2011). A Figura 13 apresenta o funcionamento da LOOCV

[1] Coeficiente binomial: Em matemática , os coeficientes binomiais são os inteiros positivos que ocorrem como coeficientes no teorema binomial.

Figura 13 – Validação cruzada deixando um de fora (*LOOCV - Leave-one-out cross-validation*)



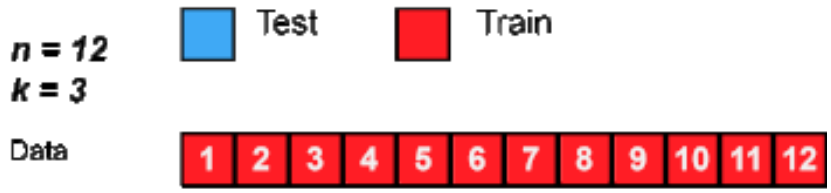
Fonte: Adaptado de https://commons.wikimedia.org/wiki/File:Leave-one-out.jpg

Já os métodos de validação cruzada não exaustivos não computam todas as formas de dividir a amostra original. Esses métodos são aproximações de validação cruzada *leave - p - out.* Na *validação cruzada k-fold,* a amostra original é dividida aleatoriamente em k subamostras de tamanhos iguais. Das k subamostras, uma única subamostra é retida como dados de validação para testar o modelo e as $k - 1$ subamostras restantes são usadas como dados de treinamento. O processo de validação cruzada é então repetido k vezes, com cada uma das k subamostras usadas exatamente uma vez como dados de validação. O $k$os resultados podem então ser calculados para produzir uma única estimativa. A vantagem desse método sobre a subamostragem aleatória repetida (veja abaixo) é que todas as observações são usadas para treinamento e validação, e cada observação é usada para validação exatamente uma vez. A validação cruzada de 10 vezes é comumente usada, mas em geral k permanece um parâmetro não fixo (McLACHLAN; DO; AMBROISE (2004). Por exemplo, definir $k = 2$ resulta em validação cruzada de 2 vezes. Na validação cruzada 2 vezes, o conjunto de dados é embaralhado aleatoriamente em dois conjuntos d_0 e d_1, de forma que ambos os conjuntos tenham o mesmo tamanho (isso geralmente é implementado embaralhando o *array* de dados e depois dividindo-o em dois). Em seguida, são treinados em d_0 e validamos em d_1, seguidos de treinamento em d_1 e validação em d_0. Quando $k = n$ (número de observações), a validação cruzada k -fold é equivalente à validação cruzada sem um. (HASTIE et al., 2009)

Na validação cruzada *k -fold estratificada,* as partições são selecionadas de modo que o valor médio da resposta seja aproximadamente igual em todas as partições. No caso de classificação binária, isso significa que cada partição contém aproximadamente as mesmas proporções dos dois tipos de rótulos de classe. Na validação cruzada *repetida,* os dados são

divididos aleatoriamente em *k* partições várias vezes. O desempenho do modelo pode, assim, ser calculado em várias execuções, mas isso raramente é desejável na prática (VANWINCKELEN, 2019). A Figura 14 apresenta um exemplo de validação cruzada k-fold ou k-iterações onde os dados se dividem em 2 subconjuntos, um conjunto de dados são para teste e outro para treinamento e o processo de validação cruzada é repetido k iterações e são testados e treinados diferentes modelos. Além disso, a validação tem *n = 12* observações e *k = 3*. Após embaralhar os dados, um total de 3 modelos são treinados e testados.

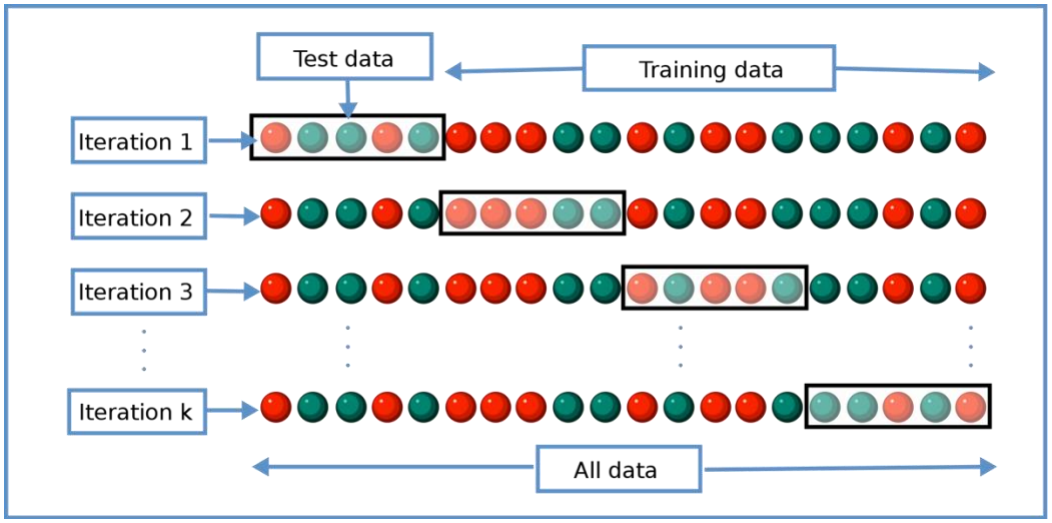
Figura 14 – Exemplo de Validação cruzada k-fold



Fonte: https://upload.wikimedia.org/wikipedia/commons/4/4b/KfoldCV.gif

Na Figura 15, apresenta-se o diagrama de validação cruzada *k-fold* com *k* iterações. Na figura, observa-se à medida que os dados da amostra são divididos em *k* subconjuntos. Um dos subconjuntos é usado como os dados de teste e o resto (*K-1*) são usados como dados de treinamento. O processo de validação cruzada é repetido durante *k* iterações (*folds*) com cada um dos possíveis subconjuntos de dados de teste. Por fim, é realizada a média aritmética dos resultados de cada iteração para obter um único resultado.

Figura 15 – Diagrama de validação cruzada *k-fold* Com k iterações.

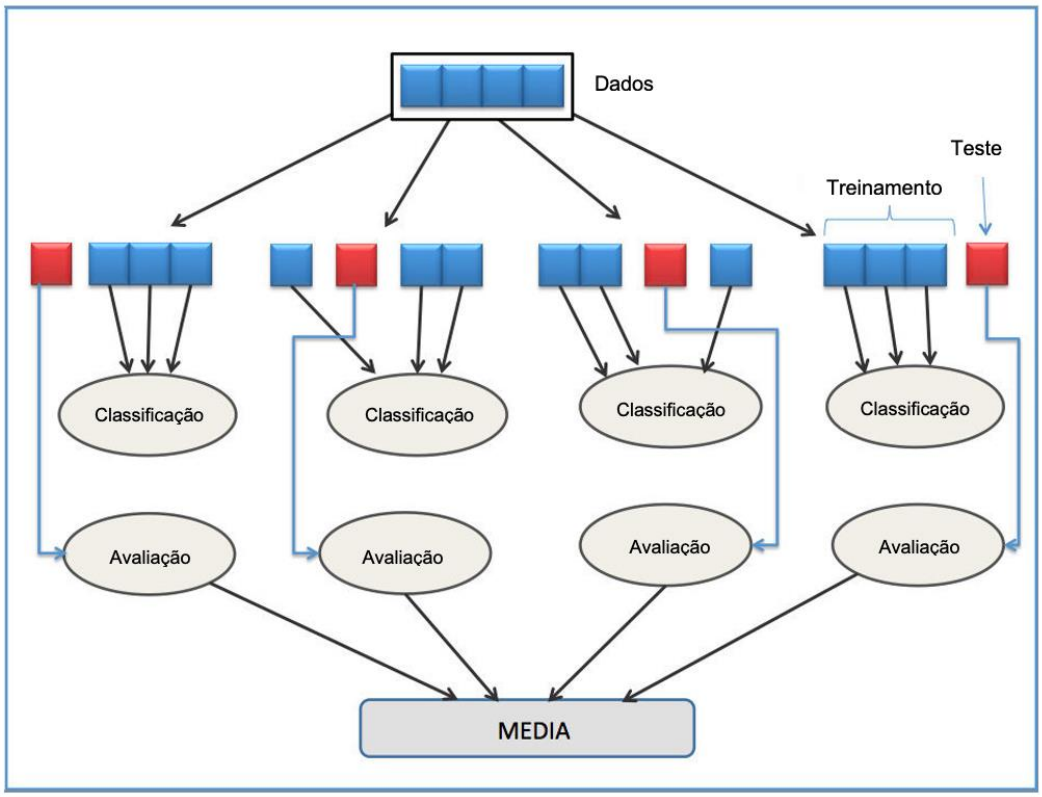


Fonte: Adaptado de Joanneum (2006) e de wikimedia.org.

Segundo Refaeilzadeh, Tang e Liu (2009) e Joanneum (2006), este método de validação é muito preciso, pois permite avaliar a partir de *k* combinações de dados de treinamento e teste, mas ainda tem uma desvantagem, pois ao contrário do método de retenção, é lento do ponto de vista computacional. Na prática, a escolha do número de iterações depende da medição do conjunto de dados. A validação cruzada de 10 iterações é a mais comumente utilizada (*10-fold cross-validation*).

A validação cruzada pode ser usada para comparar os resultados de diferentes procedimentos de classificação preditiva. Por exemplo, suponha que tem um detector que determine se um rosto pertence a uma mulher ou a um homem e consideramos que dois métodos diferentes foram usados, por exemplo, máquinas vetoriais de suporte (SVM) e K-vizinhos mais próximos (kNN), uma vez que ambos nos permitem classificar as imagens. A validação cruzada permite comparar os dois procedimentos e determinar qual dos dois é o mais preciso. Essas informações são fornecidas pela taxa de erro que obtemos ao aplicar a validação cruzada para cada um dos métodos propostos. A Figura 16 apresenta um esquema *k-fold cross validation*, con k=4 y un solo clasificador.

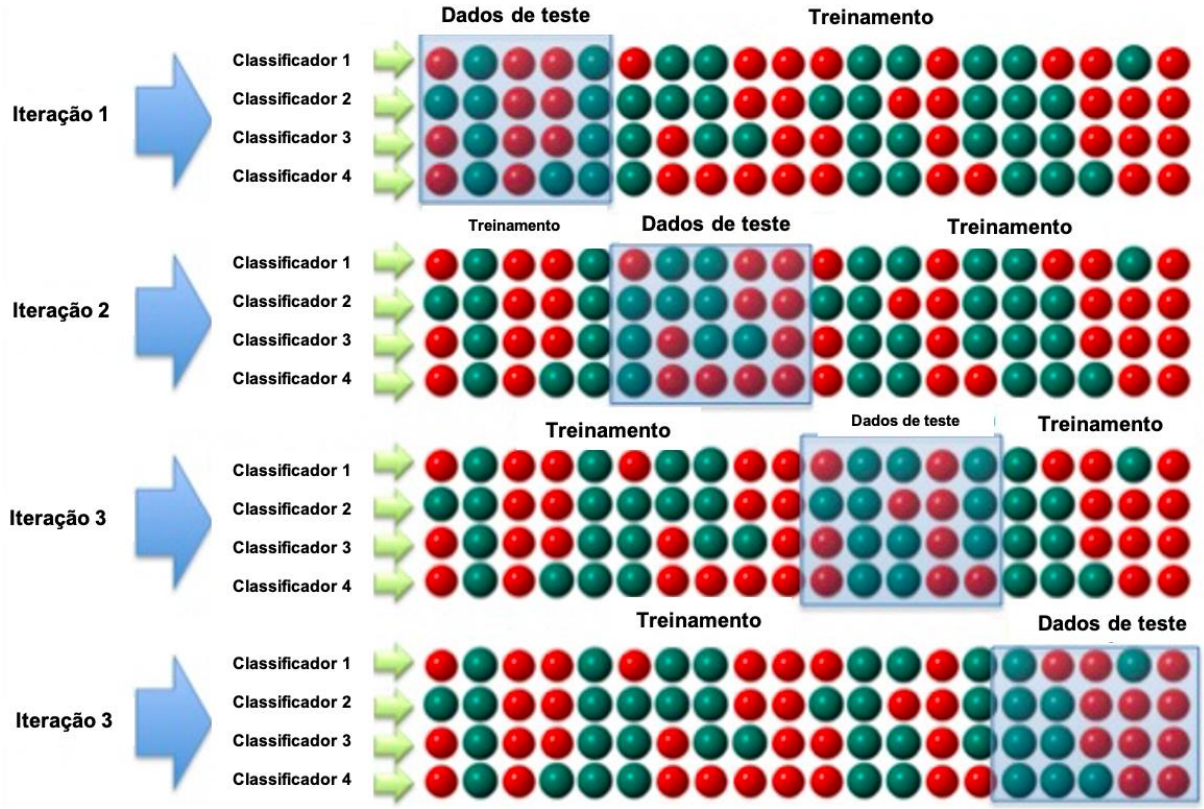
Figura 16 – Esquema k-fold cross validation, con k=4 y un solo clasificador



Fonte: Adpatado de Lang (2014).

A validação cruzada de *k* iterações (*k-fold cross validation*) também permite avaliar modelos em que são usados vários classificadores. Continuando com o exemplo anterior, se tiver um detector que determina se um homem ou uma mulher aparece em uma imagem, e ele usa quatro classificadores binários para detectá-lo, também se poderia usar a validação cruzada para avaliar sua precisão. Se tiver um total de 20 dados (imagens), e usar o método de validação cruzada 4 vezes, serão realizadas quatro iterações e, em cada iteração ou repetição, serão utilizados dados de treinamento diferentes, que serão analisados por quatro classificadores, que posteriormente avaliarão os dados do teste. Desta forma, para cada amostra, seriam otidos quatro resultados, e se for dividida a média entre os resultados de cada classificador e entre as quatro iterações feitas, seria obtido o valor final resultante. Assim, a Figura 17 apresenta um exemplo de aplicação para a classificação preditiva do método de validação cruzada k-fold com k = 4 iterações e com 4 classificadores.

Figura 17 – Exemplo de aplicação para classificação preditiva do método de validação cruzada *k-fold* com *k=4* e com 4 classificadores



Fonte: Adaptado de https://commons.wikimedia.org/wiki/File:4fold3class.jpg

2.7.2 Acurácia

A *acurácia* (exatidão) é a medida de desempenho mais intuitiva (BIRD; KLEIN; LOPER, 2009), pois é uma "proporção do resultado previsto corretamente com o total de observações do conjunto. Com esta medida, pode-se dizer quantas classificações foram de fato classificadas corretamente, independentemente se foram determinadas como positivas ou negativas" (SILVA, 2021, p. 25). Esta métrica é definida pela razão entre o que o modelo acertou e a soma de todas as classificações.

$$Acurácia = \frac{VP+VF}{VP+VF+FP+FN} \quad (6)$$

2.7.3 Precisão

Precisão é "a razão entre as observações VP previstas corretamente e o total de observações julgadas como positivas, ou seja, o divisor é a soma de VP com FP. Pode-se ter verdadeiros positivos" (SILVA, 2021, p. 24).

$$Precisão = \frac{VP}{VP+FP} \quad (7)$$

2.7.4 Revocação

A *Revocação* (*recall*), também chamada de *Sensibilidade* nos casos para classificação binário, é a medição da capacidade do modelo sabendo-se a proporção de Verdadeiros Positivos que se consegue acertar de todas os documentos da classe real, ou seja, mede quão bom um modelo é para prever apenas os positivos. Portanto *Revocação* é definido como a razão entre Verdadeiros Positivos sobre a soma de Verdadeiros Positivos com Negativos Falsos. (SILVA, 2021).

$$Recall = \frac{VP}{VP+FN} \quad (8)$$

2.7.5 F1-Score

F1-Score é uma medida baseada na média harmônica dos valores de precisão e *recall*/ Essa medida é útil nas ocasiões em que é preciso uma medida única para comparar de forma direta dois ou mais classificadores. Sem esquecer que as medidas de precisão e *recall* são de grande utilidade para avaliações de desempenho de classificadores (SILVA, 2021)

$$F1 - Score = 2 * \frac{precisão * recall}{precisão + recall} \quad (9)$$

2.7.6 Matriz de Confusão

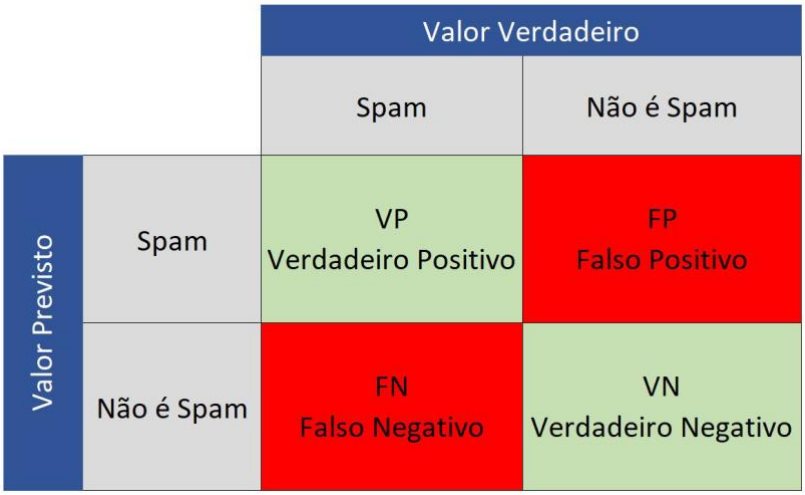
Segundo Chipman et al. (2004), a *matriz de confusão* é um método amplamente utilizado para analisar resultados. Ela separa classe por classe, a relação entre os dados de referência conhecidos e os resultados correspondentes de uma classificação automatizada.

No âmbito de *machine learning*, uma *matriz de confusão* é uma tabela que permite a visualização do desempenho de um algoritmo de classificação (DUDA; HART; STORK, 2001).

A *matriz de confusão* facilita a visualização do número de classificações corretas e do número de classificações preditas para cada classe, de um determinado conjunto de exemplos, segundo o classificador em análise. Por conseguinte, é uma ferramenta útil para analisar a qualidade do classificador no reconhecimento de exemplos de diferentes categorias (HAN; KAMBER, 2006).

A *matriz de confusão* fornece tanto o número de VP, VN, FP e FN, como as métricas de acurácia e revocação. Seguindo o exemplo de Silva (2021), a Figura 18 mostra uma matriz de confusão com classificações de e-mails para determinar se é Spam[2] ou não. As quantidades de cada classificação são apresentadas nos locais de "VP", "VN", "FP" e "FN". Conforme mostrado nas seções anteriores, com estes valores e possível, por exemplo, calcular as métricas de acurácia, precisão, revocação e F1-Score.

Figura 18 – Exemplo de Matriz de Confusão: Classificação de Spams



Fonte: Silva (2021, p. 26)

[2]Spam é uma prática que consiste em utilizar meios eletrônicos para enviar mensagens que não foram solicitadas.

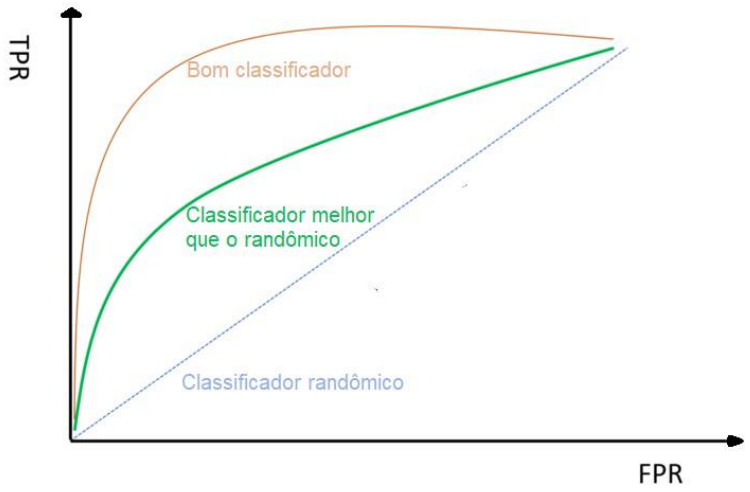
2.7.7 ROC

Receiver Operating Characteristics (ROC) e uma métrica muito usada para modelos de classificação. As denominadas *curvas ROC* são ferramentas que avaliam a sensibilidade da técnica. Concretamente, são baseadas na relação entre a fração de verdadeiros positivos (TPR) e a fração de falsos positivos (FPR).

Na analise do desempenho de modelos classificatórios são aplicados diversos estimadores estatísticos, e um dos mais utilizados é a curva ROC, que consiste em uma representação gráfica da *performance* de um modelo de dados quantitativos segundo sua taxa de sensibilidade (fração dos verdadeiros positivos) e a fração dos falsos positivos (1-especificidade), segundo diferentes valores do teste.

Como observado na Figura 19, a curva ROC é o gráfico traçado com TPR no eixo *y* e FPR no eixo *x* para todos os limites possíveis. Tanto o TPR quanto o FPR variam de 0 a 1. Um bom classificador terá uma curva mais distante da linha do classificador randômico. Para quantificar um bom classificador de um mau usando uma a curva ROC, é feito por AUC (Área sob a Curva).

Figura 19 – Curva ROC



Fonte: Silva (2021, p. 27)

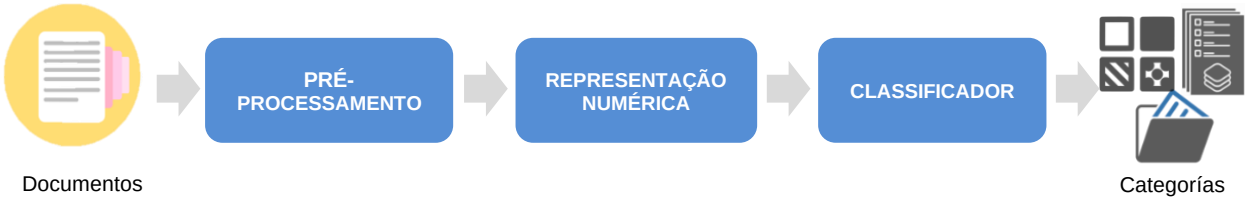
No gráfico da Figura 12 percebe-se que um bom classificador terá AUC maior do que um classificador ruim, pois a área sob a curva será maior para o primeiro.

2.8 Processamento de Linguagem Natural

Segundo Chowdhury (2003a) e Silva (2021) o Processamento de Linguagem Natural (PNL) consiste no desenvolvimento de modelos computacionais para realizar tarefas que dependem de informações expressas em alguma linguagem natural.

Chowdhury (2003b) divide os trabalhos e pesquisas relacionados ao processamento da linguagem natural em três categorias: Analise léxica e morfológica; Semântica e analise do discurso; e Abordagens baseadas no conhecimento e ferramentas para PNL. O autor também destaca que para que os textos possam ser aplicados a um classificador, é necessário submete-los a uma etapa de pré-processamento e depois representa-los numericamente. Com a representação numérica, é possível utilizar algoritmos de aprendizado de máquina para dividir os e-mails em categorias diferentes categorias. A Figura 20 apresenta este processo em um diagrama de blocos.

Figura 20 – Diagrama de blocos de um categorizador de documentos



Fonte: Elaborado pela autora.

A classificação automática de documentos consiste em usar técnicas de Inteligência Artificial em um conjunto de elementos para classificá-los por classes ou categorias. No entanto, também podem ser usadas essas técnicas para atribuir um documento a uma determinada classe ou categoria.

Para realizar a classificação automática dos documentos, o primeiro passo é extrair as características (*features*) informativas e não redundantes. Isso facilitará as etapas subsequentes de aprendizado de máquina na classificação automática de documentos. A extração de características é um processo de redução e codificação, onde um conjunto inicial de variáveis brutas – por exemplo, o texto em um documento – é reduzido a características mais gerenciáveis para seu processamento – por exemplo, os números – e que o conjunto de dados original seja descrito com precisão.

Existem várias técnicas para extrair características (*features*) como mostrado na Tabela 1:

Tabela 1 – Técnicas para extrair caraterísticas de documentos

Tipo de técnica	Nome da técnica	Definição e Caraterísticas da técnica
Técnicas clássicas	***Term frequency – Inverse document frequency (TF – IDF)***	É uma técnica amplamente utilizada no m*achine learning* para conceder a relevância de uma palavra em um documento de uma coleção através de uma medida numérica. Por tanto, TF-IDF é a abreviatura de frequência de termo – frequência de documentos inversos, que é uma medida numérica usada para pontuar a importância de uma palavra em um documento com base na frequência com que ela apareceu naquele documento e uma determinada coleta de documentos. Essa técnica baseia-se na ideia de que, se uma palavra aparece com frequência no documento, deve ser importante, por isso deve ser dada uma pontuação alta. No entanto, se uma palavra aparece frequentemente em outros documentos, provavelmente não é um identificador único, por isso deve ser atribuído uma pontuação mais baixa a essa palavra. A fórmula matemática para esta medida[3]: $$tfidf(t,d,D) = tf(t,d) \text{ x } idf(t,D)$$ Onde *t* denota os termos; *d* denota cada documento; *D* denota a coleção de documentos.
Novas técnicas baseadas em *Deep Learning*	***Doc2vec***	O principal objetivo do Doc2Vec é associar documentos arbitrários com rótulos ou etiquetas (*tags)*. Doc2vec é uma extensão do word2vec que aprende a correlacionar rótulos e palavras, em vez de palavras com outras palavras. O primeiro passo é criar um vetor que represente o "significado" de um documento para que depois ele possa ser usado como entrada para um algoritmo de aprendizado de máquina supervisionado e, assim, associar documentos com rótulos.

Fonte: Elaborado pela autora com base em Intelligent (2019)

Após a extração de *features* e com base nas informações anteriores possuídas dos documentos a serem classificados ou categorizados, várias técnicas podem ser realizadas para a classificação automática dos documentos, dentre elas as mais utilizadas são (INTELLIGENT, 2019):

- **Classificação supervisionada e técnicas de aprendizado supervisionado para classificações supervisionadas:** A classificação supervisionada é utilizada quando conhecemos o conjunto de documentos previamente classificados manualmente, estes servirão para treinar o sistema inteligente na classificação automática. As técnicas de aprendizado supervisionado para classificações supervisionadas tentam reduzir uma função que, a partir da coleta de documentos (classificação manual) e de um documento de entrada, seja capaz de prever a classe ou categoria à qual esse documento corresponde. Em outras palavras, essas técnicas

[3] Fórmula matemática tf-idf: http://ethen8181.github.io/machine-learning/clustering_old/tf_idf/tf_idf.html

de classificação supervisionadas partem de um conjunto de documentos já classificados manualmente (conjunto de treinamento) e tentam atribuir uma classificação a um segundo conjunto de documentos. Dependendo do tipo de coleção de documentos ou do tipo de documentos a serem classificados, uma técnica ou outra será utilizada: algoritmos de classificação bayesiana, árvores de decisão, redes neurais, etc.

- **Classificação no supervisionada ou *clustering* de documentos e algoritmos de *clustering* para classificações não supervisionadas**: A classificação não supervisionada é utilizada quando não temos informações a priori sobre o conjunto de documentos ou as categorias em que devem ser classificados. Os algoritmos de agrupamento para classificações não supervisionadas são utilizados quando não temos um conjunto de documentos previamente classificados, por isso partimos das propriedades dos documentos para agrupa-los *(clustering)* de acordo com suas semelhanças entre si.

2.8.1 Modelos de *Machine Learning* adequados para a classificação documental

Os modelos de *machine learning* que permitem gerar algoritmos de classificação automática de documentos (modelo de classificação); desenvolver um algoritmo para detectar padrões de gerenciamento de documentos; aplicar um sistema de regras para automatizar tarefas repetitivas; e fornecer capacidade de aprendizado para sistemas são: Regressão Logística, k-NN, SVM, Naive Bayes (NB), Random Forest (RF), Grandiente Boosting (GB), Classificação em árvore , Redes Neurais (RD), dentre outras.

Dentre os métodos existentes cabe destacar o método *fit* que permite ajustar os modelos passando os dados de treinamento e a variável objetivo.

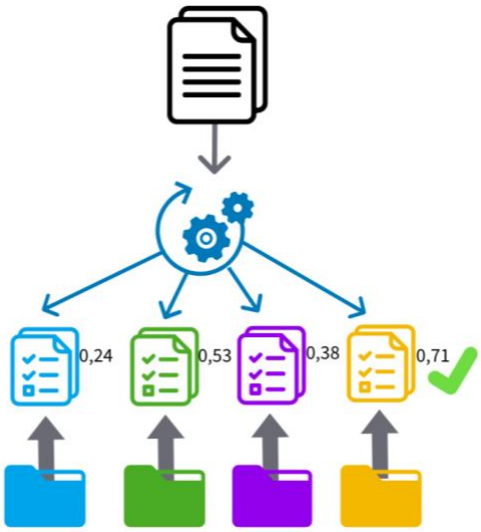
Alguns dos algoritmos de aprendizado de máquina mais destacados que permitem realizar reconhecimento automático, classificação e roteamento de documentos eletrônicos processados e armazenados em um SGDE(A) existentes na literatura são:

- **Algoritmos probabilísticos**: São baseados na teoria probabilística, especialmente o teorema de Bayes. Teorema que permite estimar a probabilidade de um evento a partir da probabilidade de que outro evento ocorra, do qual o primeiro depende. Na verdade, o algoritmo mais conhecido desse tipo, e também o mais simples, é o chamado *Naive Bayes*, proposto por Maron (1961) e van Rijsbergen (1979) que consiste em estimar a

probabilidade de um documento pertencer a uma categoria. Essa pertença depende das possessão de uma série de características, de cada uma das quais sabemos a probabilidade de que eles apareçam nos documentos que pertencem a essa categoria.

- **Algoritmo de Rocchio**: O algoritmo de Rocchio (1971) é conhecido e aplicado para realimentar consultas. O processo é o seguinte: após formular e executa uma primeira consulta, o usuário examina os documentos devolvidos e determina quais são relevantes para ele e quais não são. Com esses dados, o sistema gera automaticamente uma nova consulta, com base nos documentos que o usuário indicou como relevantes ou não relevantes. Nesse contexto, o algoritmo de Rocchio fornece um sistema para construir o vetor da nova consulta, recalculando os pesos dos termos da consulta e aplicando um coeficiente aos pesos da consulta inicial, outro aos dos documentos relevantes e outro diferente aos dos não relevantes. Esse algoritmo, portanto, permite obter os padrões de cada classe, como mostrado na Figura 21. O algoritmo de Rocchio tem sido usado em tarefas de categorização e a literatura mostra resultados ideais como podemos ver nos trabalhos de Lewis et al. (1996), Joachims (1997) e Figuerola, Zazo Rodrígue e Alonso Berrocal (2001).

Figura 21 – Algoritmo de Rocchio



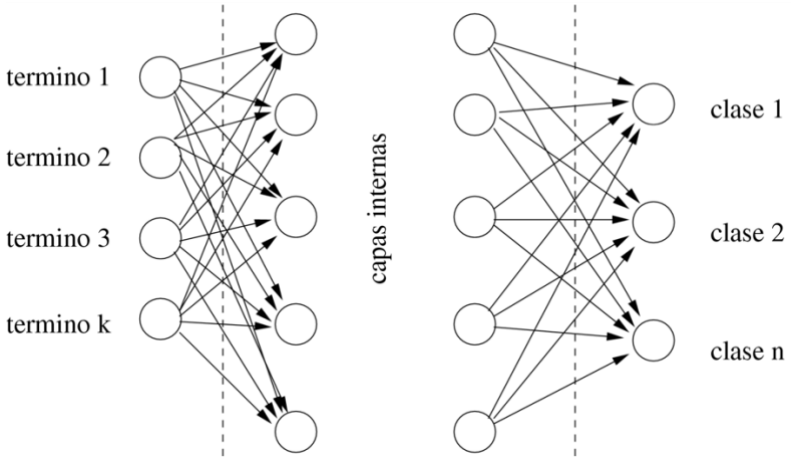
Fuente: Elaborado pela autora com base em Figuerola et al. (2005)

- **Algoritmo vizinho mais próximo e variantes (*Nearest Neighbour*, NN):** Este algoritmo é um dos mais fáceis de implementar. De acordo com Figuerola et al. (2005), se calcularmos a semelhança entre o documento a ser classificado e cada um dos documentos de treinamento, o mais parecido ele estará nos dizendo a qual classe ou categoria devemos atribuir o documento que queremos classificar. O *vizinho mais*

próximo pode ser aplicado a qualquer programa de recuperação do tipo *best macth*. Segundo Figuerola et al. (2005), o mais comum é usar um baseado no modelo vetorial, mas isso não é essencial. Porem, é imprescindível que seja *best macth* e não de comparação exata, como pode ser o caso dos booleanos; o algoritmo baseia-se na localização do documento mais semelhante ou semelhante ao que você deseja classificar. Para isso, você só tem que usar esse documento como se fosse uma consulta sobre a coleção de treinamento. Uma das variantes mais conhecidas deste algoritmo é a de *k-nearest neighbour* (KNN), que consiste em pegar os documentos mais semelhantes, em vez de apenas o primeiro. Como nesses *k* documentos os haverá de várias categorias, serão adicionados os coeficientes dos de cada um delas. O que acumular mais pontos será o candidato ideal.

- **Algoritmos baseados em redes neurais**: Pelo geral, Redes neurais foram propostas em inúmeras ocasiões como instrumentos úteis para recuperação de informações e classificação automática. Durante o *boom* do uso de estas redes em todas as áreas de estudo e aplicação na década dos 90', existem vários pesquisadores que aplicaram redes neurais para a classificação documental, por exemplo cabe destacar ao Oard (1994). Uma das principais aplicações das redes neurais é o reconhecimento de padrões, que também tem sido aplicado para resolver problemas de categorização de documentos. De acordo com Figuerola et al. (2005), uma rede neural consiste em várias camadas de unidades de processamento interconectados ou neurônios; no campo em questão, a camada de entrada recebe termos, enquanto as unidades ou neurônios das classes ou categorias de mapa da camada de saída. Interconexões têm pesos, que é um coeficiente que expressa a maior ou menor força da conexão. É possível treinar uma rede neural para que, dada uma determinada entrada (os termos de um documento), produza a saída desejada (a classe que corresponde a esse documento). Ainda, Figuerola et al. (2005) ressaltam que o processo de treinamento consiste em um ajuste dos pesos das interconexões, para alcançar a saída desejada. Diferentes tipos de redes têm sido aplicadas como podemos ver nos estudos de Schutze, Hull e Pedersen (1995) que usam uma rede de retropropagação aplicada ao problema de *routing* ou filtragem de documentos, algo muito semelhante à categorização; os resultados obtidos são, segundo os autores, superiores aos obtidos com o algoritmo de Rocchio. Já, Orwig, Chen e Schuffels (1996) apresenta um experimento semelhante, mas com um *website*. Na Figura 22 se apresenta o esquema ou modelo de uma rede neural que poderia ser utilizada para a classificação automática documental.

Figura 22 – Rede neuronal para a classificação automática



Fonte: Figuerola et al. (2005, p. 6)

Assim, após revisar alguns dos algoritmos mais frequentemente usado para classificação supervisionada de documentos, pode-se concluir que a classificação automática de documentos pode ser alcançada por diversas técnicas. Além disso, de acordo com Figuerola et al. (2005), indistintamente do grau de complexidade de cada um desses algoritmos habilmente utilizados para a classificação supervisionada de documentos, as pesquisas revelam que são técnicas suficientemente maduras, cujos resultados são comparáveis aos alcançados por classificadores inteligentes, ou seja, pelo ser humano.

De outro lado, Figuerola et al. (2005) identificam algumas questões a serem resolvidas, tais como: as relativas às características ideais das coleções de treinamento, em termos de tamanho e distribuição das diferentes categorias; a capacidade de adaptação às mudanças nos conteúdos das próprias classes; a capacidade de detectar mudanças em documentos e classes; ou o surgimento de novos tipos de documentos que precisam de outro tipos de respostas (por exemplo, os documentos multimídia dificilmente podem ser representados como vetores de termos, uma vez que não consistem em texto ou as páginas de um *website* que, embora sejam documentos textuais em grande parte, também incorporam outros elementos de informação que precisam ser coletados (NOEL; RAGHAVAN; CHU, 2003), não apenas as imagens ou o som, mas também os próprios *links*

2.9 Sistemas de Informação e Gestão documental

Segundo Openkm (2022), os documentos contêm informações que são um recurso valioso e um ativo primário na organização. A adoção de um critério sistemático de gestão documental é essencial para as organizações e a sociedade, a fim de proteger e preservar os

documentos como evidência de suas ações. Um Sistema de Gestão Documental (SGD) torna-se uma fonte de informação sobre as atividades da organização, que pode servir de suporte para as atividades e tomadas de decisão subsequentes, ao mesmo tempo que garante a prestação de contas aos *stakeholders* presentes e futuros. Os documentos permitem às organizações realizar as tarefas que detalhadas na Figura 23:

Figura 23 – Tarefas que os documentos permitem realizar às organizações

Fonte: Elaborado pela autora com base em Openkm (2022)

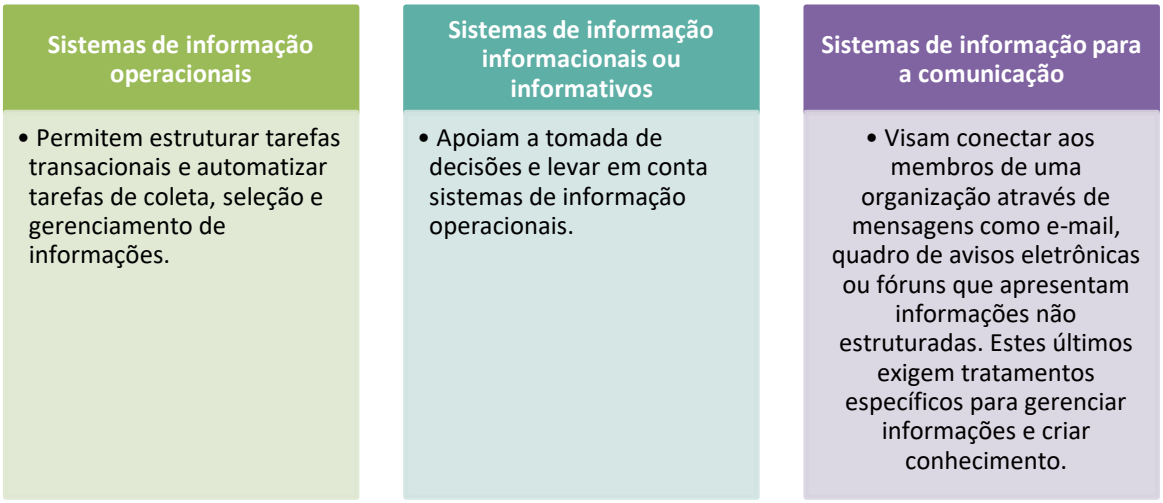
García Alsina (2009) define o sistema de informação como um conjunto de elementos compostos por pessoas, processos de negócios, dados, informações, documentos, ferramentas de computador (*hardware*, *software*, comunicações) e tecnologia de comunicação, visando coletar, armazenar e distribuir informações entre os membros de uma organização, e entre ela e seu ambiente. Também afirma que a efetividade desses processos depende do alinhamento de sistemas com processos de negócios e estratégias e objetivos organizacionais. De fato, tem diferentes ferramentas tecnológicas que permitem gerenciar a informação e recebem diferentes denominações de acordo com o tipo de informação que gerenciam como veremos a seguir nas Figura 21 e 22.

Laudon e Laudon (1016) define o sistema de informação como como:

> “um conjunto de componentes interrelacionados que coletam (ou recuperam), processam, armazenam e distribuem informações para apoiar os processos de tomada de decisão e controle em uma organização. Além de apoiar a tomada de decisão, coordenação e controle, os sistemas de informação também podem ajudar aos gestores e trabalhadores do conhecimento a analisar problemas, visualizar questões complexas e criar novos produtos.” (LAUDON; LAUDON, 2016, p. 16 [tradução nossa do idioma espanhol])

Na Literatura existem, diferentes critérios para classificar sistemas de informação, como o que segue Laudon e Laudon (2016) ou García Alsina (2009), que classifica as ferramentas de acordo com seu propósito como mostrado na figura a seguir:

Figura 24 – Classificação dos sistemas de Informação



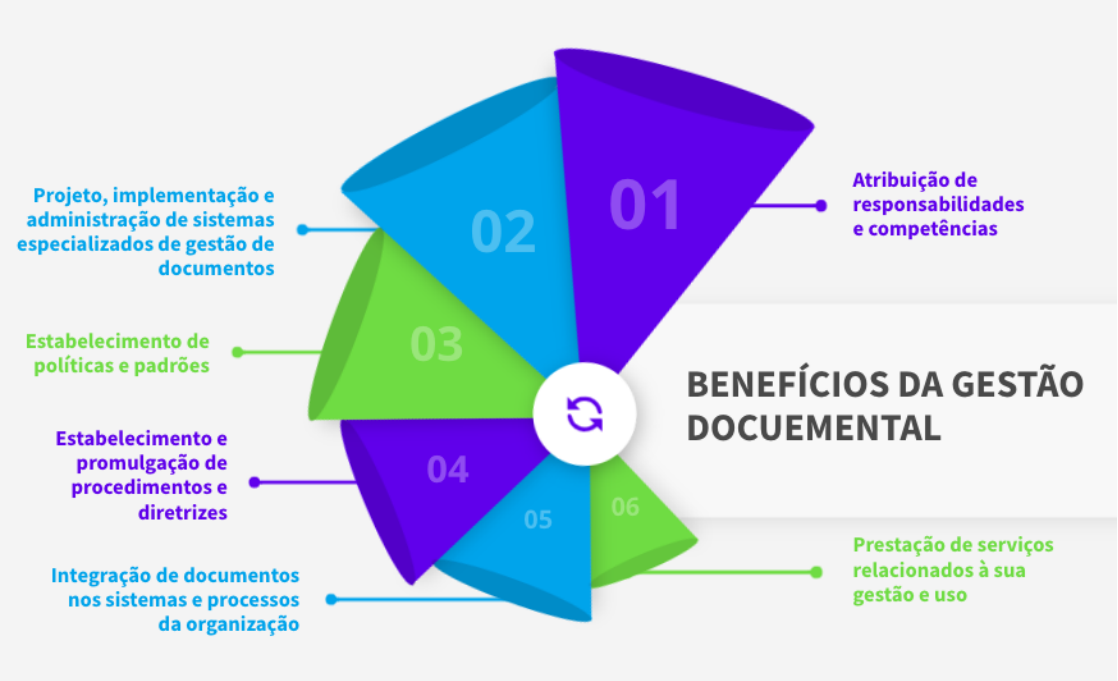
Fonte: Elaborado pela autora com base em García Alsina (2009)

García Alsina (2009) destaca sistemas de informação operacional, que por sua vez atuam como informativos ou informacionais, por sua utilidade para a tomada de decisões. Esses sistemas geram evidências da atividade da organização.

2.9.1 Benefícios da Gestão Documental

A gestão de documentos ajuda a regular as práticas realizadas pelos responsáveis da sua gestão e por qualquer outra pessoa que gere ou utilize documentos no exercício de suas atividades (OPENKM, 2022). Assim, a Figura 25 apresenta uma série de benefícios da gestão documental em uma organização.

Figura 25 – Benefícios da Gestão Documental



Fonte: Elaborado pela autora com base em Openkm (2022)

Por conseguinte, a gestão documental facilita a gestão de conteúdos, simplifica o trabalho e incrementa a eficiência nos processos de trabalho.

2.9.1.1 Sistemas de Gestão de documentos Eletrônicos de Arquivo (SGDE(A))

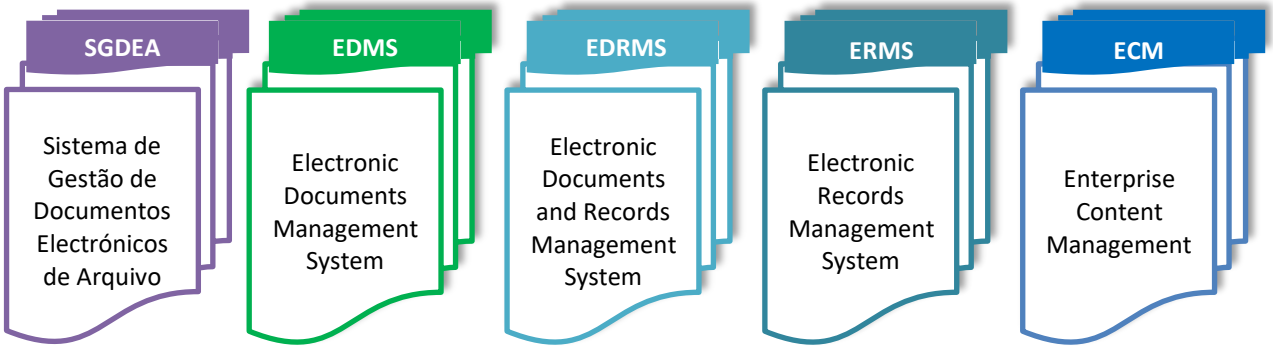
Segundo Rangel (2017), as tecnologias de informação e telecomunicações tem introduzido novas práticas e formas de gestão de documentos, além de ser uma ferramenta fundamental para o acesso, consulta, transparência, otimização e disponibilidade de informações. No entanto, é necessário estabelecer políticas claras sobre a produção, distribuição, consulta, retenção, armazenamento, preservação e destinação final, uma vez que cada decisão associada ao tratamento de documentos terá efeitos sobre o patrimônio documental.

Rangel (2017), destaca também que o princípio fundamental a ser considerado no uso das tecnologias é que, independentemente de a informação eletrônica ser um "documento", um “documento de arquivo”, um "dado" ou um "conteúdo”, a concepção da informação é a mesma, é “eletrônico” e deve ser gerenciado por pelo menos um componente tecnológico.

Rangel (2017) também assinala que existem diferentes termos para se referir a sistemas de gestão de documentos eletrônicos, pois a indústria vem evoluindo e as tecnologias têm

passado por um processo de maturação no qual novos conceitos, siglas e acrónimos tem sido incorporados, cada um com diferentes propósitos e significados. A autora também argumenta que esses termos estão mudando constantemente porque muitos fornecedores, empresas de marketing, a indústria em geral, analistas e consultores acham mais fácil incorporar um novo conceito para descrever variantes emergentes de tecnologias centrais do que modificar as existentes. Assim, dentre os termos referentes aos sistemas de informação associados ao controle e gestão de documentos, foram identificados os apresentados na Figura 26.

Figura 26 – Sistemas de informação associados ao controle e gestão de documentos



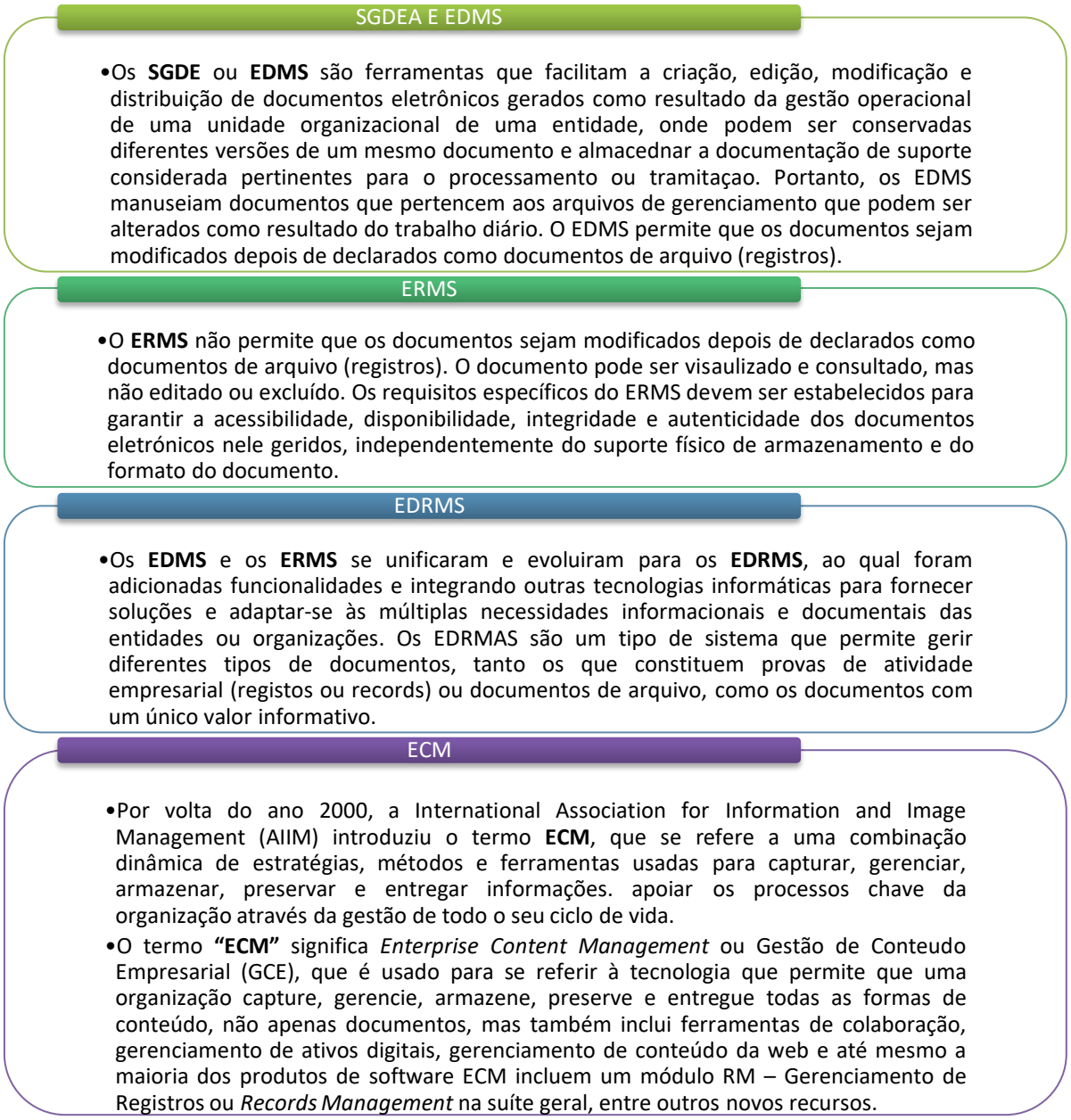
Fonte: Elaborado pela autora.

Noonan (2011) define o **Sistema de Gestão Eletrônico de Documentos** (**SGDE**) ou *Electronic Documents and Records Management System* (**EDMS**) como um sistema de *software* que controla e organiza os documentos em toda a organização, independentemente de terem sido declarados como registros eletrônicos ou não. Um EDMS normalmente inclui:

- Criação e captura de conteúdos e documentos.
- Distribuição de documentos.
- Edição e revisão de conteúdos e documentos.
- Workflow Documental ou fluxo de trabalho de documentos / Gestão de processos empresariais (*Business Process Management* (BPM)).
- Indexação, acesso, armazenamento e recuperação de conteúdo e documentos.
- Processamento de imagem.
- Repositórios de documentos.

A seguir (Figura 27) podem se observar as diferenças e características dos sistemas de informação associados ao controle e gestão documental, sendo eles: SGDEA e EDMS, ERMS, EDRMS e ECM.

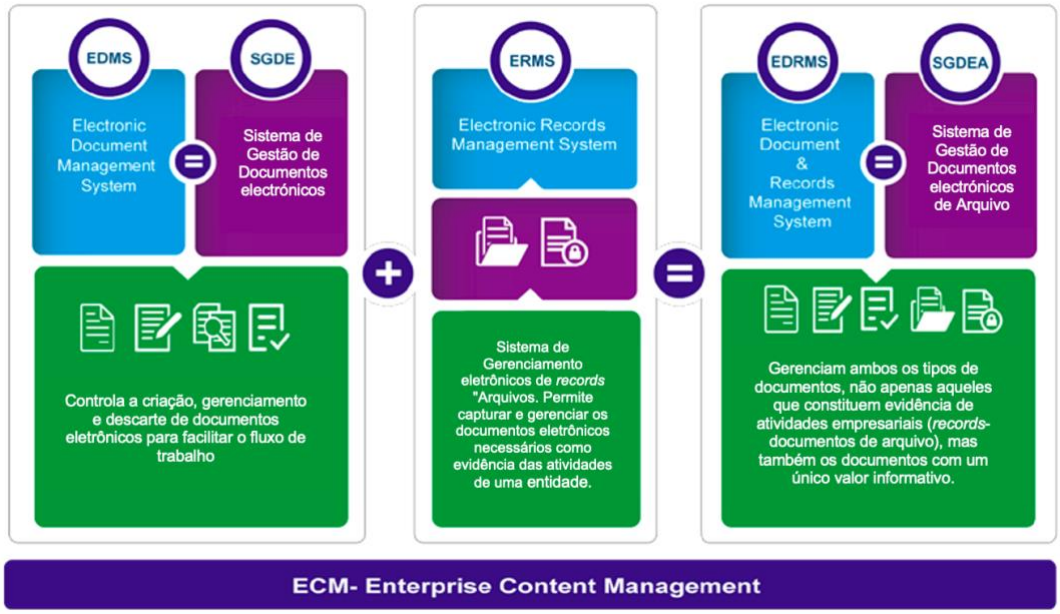
Figura 27 – Diferenças e características dos sistemas de informação associados ao controle e gestão documental



Fonte: Elaborado pela autora com base em AIIM (2022), Noonam (2011) e Rangel (2017).

A Figura 28 mostra uma representação geral dos sistemas supracitados na Figura 19 e 20:

Figura 28– Representação dos SGDE e SGDEA



Fonte: Adaptado e traduzido pela autora de Noonam (2011) e Rangel (2017, p. 19)

Como se pode apreciar na Figura 28, a tradução em português das siglas EDMS (*Electronic Document Management System*) e EDRMS (*Electronic Document and Records Management System*) é SGDE (Sistema de Gestão de Documentos Eletrônicos) e SGDEA (Sistema de Gestão de Documentos Eletrônicos de Arquivo), respectivamente.

Do ponto de vista conceitual de Rangel (2017), existem diferenças entre as características de um SGDE e SGDEA, pois um SGDE é um sistema de informação que integra todas as atividades voltadas para a gestão de documentos eletrônicos que não são necessariamente documentos de arquivo, nem estão harmonizados com os processos de gestão documental (documentos em produção, revisão, gestão ou processamento), embora possam sê-lo num futuro. Já, o SGDEA contém as etapas mencionadas e se foca nos documentos quando são declarados como documentos de arquivo eletrônico e, adicionalmente, tem sido concebidos para integrar o arquivo total independentemente do ciclo de vida em que os documentos se encontram.

A nível tecnológico, existem soluções que permitem a criação, indexação, gestão documental, administração de múltiplas versões e parametrização de fluxos eletrônicos. No entanto, esses documentos não são necessariamente documentos de arquivamento eletrônico e as funcionalidades de classificação são opcionais, esses sistemas são chamados de SGDE. Na Figura 29 são indicadas algumas das características e operações dos SGDE e SGDEA.

Figura 29 – Características, operações e diferenças entre um SGDE e um SGDEA

Fonte: Elaborado pela autora com base em Rangel (2017).

Outros sistemas de informação possuem características de ambos os conceitos (geralmente todos os SGDEA possuem todas as capacidades e funcionalidades de um SGDE), onde os documentos mudam de status ao serem declarados como documentos de arquivo eletrônico, não sendo permitidas adições ou modificações para formar arquivos eletrônicos por meio da incorporação de funcionalidades que permitem a parametrização da estrutura de classificação documental da entidade e a gestão dos tempos de retenção. (RANGEL, 2017)

A nível tecnológico, os fornecedores de sistemas informáticos para gestão do ciclo de vida dos documentos e conteúdos oferecem produtos que dão uma resposta unificada às

necessidades atuais do negócio. Essas soluções geralmente contemplam quatro componentes: captura, gerenciamento, armazenamento e distribuição. Esses tipos de produtos se enquadram na categoria de ECMs. Esses componentes estão relacionados entre si e podem ser usados em combinação ou como módulos independentes. Este tipo de tecnologia permite a gestão de qualquer tipo de ativo digital em todas as fases do seu ciclo de vida, desde a sua criação até à sua disposição final, passando pela automatização de workflows, gestão de regras, revisão partilhada, classificação de informação, gestão de etiquetas e geração de comentários , entre outros. (RANGEL, 2017)

Esse processo requer uma etapa de planejamento e preparação da informação em que os princípios arquivísticos são contemplados e pode incluir o uso de tecnologias de reconhecimento e classificação de documentos, ferramentas de extração automática de dados e armazenamento de metadados. (RANGEL, 2017).

2.9.1.2 Técnicas de aprendizado de máquina como florestas aleatórias e Xgboost

Decomposição de texto em n-gramas, a remoção ou eliminação de palavras vazias (*empty words*), o cálculo da frequência dos n-gramas, a estimativa da estatística TF-IDF para os n-gramas e a seleção das variáveis mais importantes.

2.10 Requisitos legais, arquivamento e parâmetros processuais para implantar um modelo eletrônico de gestão documental

Os objetivos da SGDEA devem ser derivados de uma análise das funções, competências e processos da entidade, levando em consideração aspectos como o tamanho, a natureza de suas atividades, procedimentos, serviços, contexto e ambiente sociocultural e marco legal aplicável e através do qual funciona o referido sistema documental.

Não se trata apenas de buscar uma solução tecnológica, mas também é necessário que a entidade e/ou a organização analisem normas internacionais e/ou nacionais, como políticas, leis e decretos existentes em termos de gestão de documentos, segurança da informação, interoperabilidade, gestão da qualidade, gestão ambiental, etc., o que permitirá uma correta gestão eletrônica dos documentos. Da mesma forma, devem ser levadas em conta as obrigações legais e regulamentares de cada organização e as diretrizes e diretrizes emitidas pelo Arquivo Geral de um país em termos de gestão de documentos. (RANGEL, 2017).

> "O Plano de Classificação Documental é o resultado da análise funcional de uma organização e o instrumento de arquivamento que contém os agrupamentos documentais definidos sob os quais o sistema será parametrizado, sendo eles: fundo(s) seção(s), subseção(s), séries, subséries e tipologias documentais categorizadas e de acordo com os níveis hierárquicos existentes na organização, funções, processos e procedimentos, ajudando a suportar os metadados dos arquivos e de cada um dos documentos eletrônicos para sua captura, gravação, busca, recuperação e preservação." (RANGEL, 2017, p.53-54, [tradução nossa do idioma espanhol])

Rangel (2017) também destaca que o Plano de Classificação Documental (PCD) é o primeiro requisito funcional mínimo a ser suportado pelo SGDEA e possui notável relevância dentro do SGDEA porque a partir dele são parametrizados os aspectos mostrados na Figura 30:

Figura 30 – Aspectos parametrizados a partir de um Plano de Classificação Documental dentro de um Sistema de Gestão de Documentos Electrónicos de Arquivo

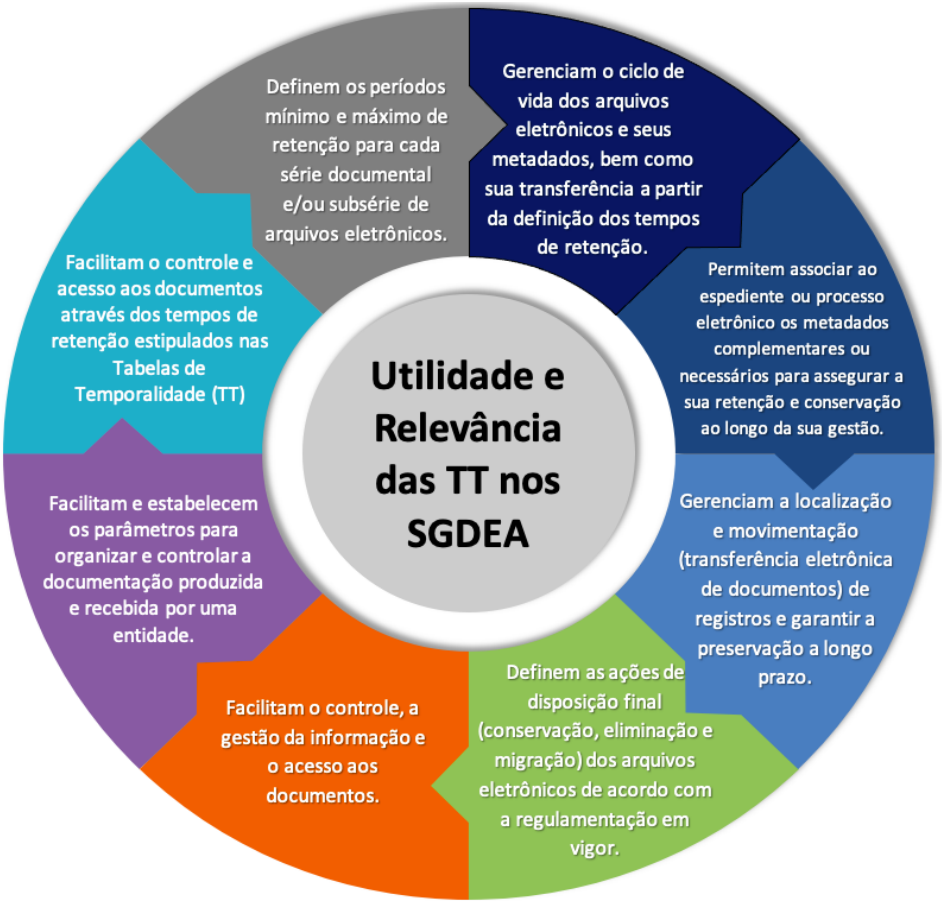
Fonte: Elaborado pela autora com base em Rangel (2017).

Assim, seguindo a Rangel (2017), podemos definir a Tabelas de Temporalidade documental (TTD) como Tabelas de Retenção de Documentos ou Tabelas de Retenção Documental (TRD) que regulam o ciclo de vida dos documentos físicos, cumprem a mesma finalidade com os documentos eletrônicos e permitem a parametrização das regras de disposição final e transações de arquivamento de documentos eletrônicos. As TRD ou TTD alinhados com o PCD, permitem classificar os documentos da organização, de acordo com a sua estrutura orgânico-funcional e gerir o ciclo de vida dos documentos para gerir corretamente as transferências de documentos eletrônicos (primários e secundários) dos ficheiros eletrônicos,

permitindo atribuir tempos de retenção para as séries e/ou subséries documentais em cada fase do arquivo (gestão e central) e sua disposição final: conservação total, eliminação, seleção documental e reprografia, seja para transferência, exportação ou destruição dos documentos do arquivo.

A elaboração de instrumentos de gestão documental, como os códigos de classificação dos documentos arquivísticos e as TTD, são de extrema importância para que seja possível executar as atividades de gestão documental e, consequentemente, gerenciar a informação. As TTD são um instrumento que facilita às entidades a agilização dos processos de organização documental com base nos critérios estabelecidos na Lei Geral de Arquivos e seus decretos normativos. Conseguintemente, as TTD são um instrumento de arquivamento relevante e útil dentro de um SGDEA, como mostrado pela Figura 31.

Figura 31 – Utilidade e relevância das Tabelas de Retenção Documental (TRD) ou Tabelas de Temporalidade (TT) em um Sistema de Gestão de Documentos Eletrônicos de Arquivo (SGDEA)



Fonte: Elaborado pela autora com base em Rangel (2017)

De acordo com García Alsina (2009), a gestão documental deve ser realizada a partir de diferentes pontos de vista, pois através da gestão documental é possível comprovar as atividades realizadas por organizações, públicas ou privadas, onde os documentos atingem um valor primário. Da mesma forma, a gestão documental também deve ser realizada do ponto de vista do conhecimento organizacional, ou seja, levando em consideração o valor secundário dos documentos, uma vez que a documentação gerada na organização contém informações capazes de gerar conhecimento, além de constituir a memória da organização. Sem esquecer que os documentos são um dos instrumentos onde o conhecimento tácito depositado na mente dos membros da organização é coletado.

A autora, García Alsina (2009), destaca ainda que, por meio das ferramentas tecnológicas implementadas na organização, a gestão documental, por sua vez, permite aos membros da organização acessar as informações necessárias para gerar os conhecimentos necessários para tomar decisões ou para elaborar planos estratégicos, por exemplo. Portanto, pode-se concluir que a gestão de documentos facilita a gestão do conhecimento, de modo que todos os projetos de gestão do conhecimento incluam a gestão de documentos e facilitem a tomada de decisões informadas.

Há diferenças e atividades complementares entre gestão do conhecimento e gestão de documentos. De fato, o *State Records Authority of the State of the New South Wales (NSW)* de Australia, destaca que a diferença entre gestão do conhecimento e gestão de documentos está no objetivo de cada disciplina. A gestão do conhecimento primeiro pretende gerar e compartilhar conhecimento, enquanto que a gestão documental se concentra na captura e manutenção de evidências de informações sobre atividades e transações empresariais. Por outro lado, se os documentos são uma importante fonte de conhecimento explícito que pode ser utilizada dentro do ciclo de gestão do conhecimento, o conhecimento criado é documentado e pode ser capturado em um sistema de gestão de documentos, indicando uma relação entre documento e conhecimento (GARCÍA ALSINA, 2009)

Así, de acordo com García Alsina (2009), pode-se concluir que o gerenciamento documental contribui para a inovação e a criação de valor agregado, aporta vantagem competitiva nas organizações, públicas e privadas, e gera riqueza e desenvolvimento econômico e territorial.

3 METODOLOGIA

Nesse capítulo é apresentado o material, os métodos e as técnicas utilizados nesse estudo, assim como os dados e ferramentas usados para a categorização e classificação automática de documentos abordando alguns modelos clássicos de aprendizado de máquina.

Como foi dito anteriormente, o objetivo principal desse trabalho é desenvolver um Sistema Automático de Classificação de Documentos com base em modelos de *machine learning* seguindo uma metodologia que permitirá incorporar os novos modelos aos *workflows* padrão de gerenciamento de documentos electrónicos.

Tendo em conta esse objetivo, essa pesquisa parte de uma revisão bibliográfica como metodologia sobre Sistemas de Informação, SGDEA, classificação de uma única classe e métodos de classificação utilizados para a detecção automática de palavras, aprendizado automático, redes neurais, treinamento supervisionado, e outros conceitos que permitem conformar o referencial teórico do estudo e discutir os resultados obtidos com a literatura.

Como se observa na Tabela 2 da seção 3.1, após **escolher o conjunto de dados** (*datasets)* que é textual, **coletar os dados, escolher a amostra e gerar a base de dados**, foi realizado uma **analise exploratória de dados**, foram decididas as variáveis e as transformações usadas como variáveis preditoras no model e foi feite o **pré-processamento dos dados**. Mas previamente ao pré-processamento foi realizada a **categorização e classificação dos dados textuais**. Seguidamente, foram **escolhidos os modelos a serem implementados para a modelagem** nas fases de **treinamento e teste de modelos**, para depois entrar na fase de **especificação, treinamento e avaliação** desses modelos. Finalmente, forma **calibrados os hiperparámetros**, concluindo com a fase de **implantação do modelo** e **avaliação dos resultados.**

Assim, as fases podem ser resumidos de forma mais visual com a figura a seguir:

Figura 32 – Diagrama das etapas metodológicas para desenvolver e implementar modelos preditivos baseados em técnicas de Inteligência Artificial

Fonte: Elaborado pela autora.

Cada uma dessas etapas metodológicas (Figura 32) que permitem desenvolver esse trabalho é apresentada com maior detalhe nas seções subsequentes. Finalizando esse capítulo com algumas considerações sob a escolha metodológica apresentada.

3.1 Fases metodológicas para o desenvolvimento e implementação dos modelos preditivos

Para a concepção, desenvolvimento e implementação de modelos preditivos baseados em técnicas de Inteligência Artificial, se propõe seguir uma metodologia estruturada em 7 etapas ou fases como se detalha a continuação na Tabela 2:

Tabela 2 – Metodologia para desenvolver e implementar modelos preditivos baseados em técnicas de Inteligência Artificial

Etapa / Fase	Descrição
1. Coleta, amostragem de dados e geração da base de dados	• Nessa primeira fase, que é crítica porque afeta a todas as fases subsequentes, será gerada a base de dados com o *datasets* escolhido. • A correta escolha e construção do *dataset* é essencial nos trabalhos que utilizam *machine learning*. • Esta fase requer a coleta de um grande número de observações representativas do problema. • Também é necessário que o número de amostras em cada classe seja semelhante para evitar problemas de desequilíbrio ou desajustes (*imbalance*) que podem ser difíceis de resolver pela maioria das técnicas de ML.

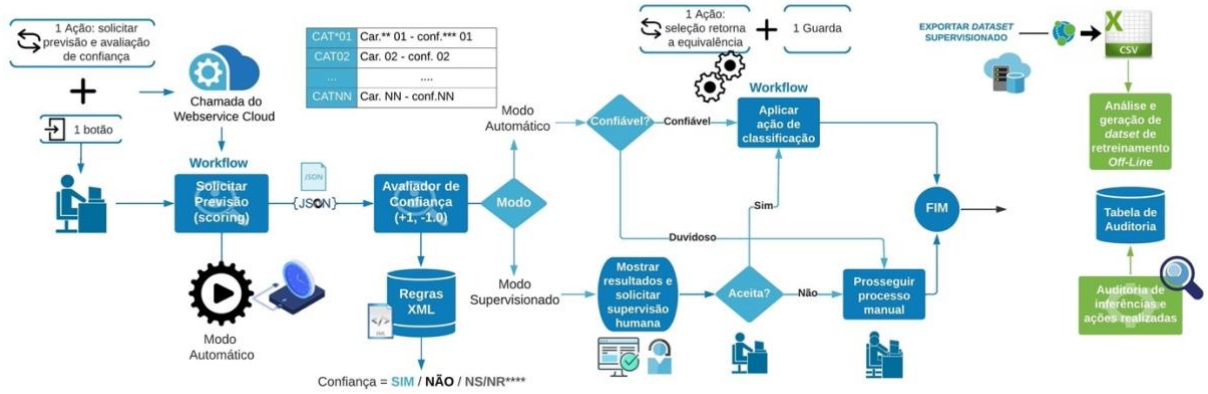
2. Análise exploratória de dados	• Esta etapa é decisiva para entender as informações fornecidas pelos dados e decidir sobre os parâmetros-chave na modelagem dos dados. • Nesta etapa, serão revisados principalmente os seguintes aspectos: o Número total de observações ou amostras e sua distribuição nas diferentes classes; o Presença de valores ausentes e como lidar com eles; o Outras variáveis não textuais em cada observação que podem contribuir com elementos para a correta classificação (chamamos essas variáveis de metadados). Além disso, nesta fase serão determinadas as principais características da distribuição dessas variáveis e sua possível relação com a classe a ser prevista. o Distribuição de frequência de palavras ou *tokens* de texto.
3. Categorização e classificação dos dados textuais ou documentais.	• Nessa etapa são categorizados e classificados os dados textuais ou documentos do *datasets*. • Para a classificação dos textos, foram escolhidas 3 categorias: "Crime" (*Crime*), "Politica" (*Politics*) y "Ciência" (*Science*), pois as entradas ou textos da categoria "Entretenimento" (*Entertainmet*) foram eliminados por estar repetida nas categorias outras três categorias. Ainda na etapa de construção da base de dados, cada texto foi classificado de acordo com a escolha do autor, portanto o estudo propõe uma solução de aprendizado de máquina supervisionado. Cada um dos textos pertence a uma única categoria.
4. Engenharia de variáveis e pré-processamento dos dados	• Com base na etapa anterior, foram decididas as variáveis e suas transformações que serão utilizadas como variáveis preditoras no modelo (as variáveis preditoras são aquelas usadas para predizer alguma outra variável ou resultado). Foram utilizadas técnicas de Análise inteligente de texto ou mineração de texto (*text mining*). Em um modelo de texto, essas etapas incluem, entre outras coisas: o Decomposição de texto ou *tokenização* em n-gramas. o Remoção ou eliminação de palavras vazias (*empty words*): preposições, artigos e outras variáveis comuns, mas sem importância semântica. o Cálculo da frequência dos n-gramas. o Estimativa da estatística TF-IDF para os n-gramas. o Seleção das variáveis mais importantes (por exemplo, os *tokens* de maior frequência ou importância).
5. Especificação, treinamento e avaliação dos modelos	• Após a preparação dos dados, foram escolhidos os modelos selecionados a serem implementados para a modelagem nas fases de treinamento e teste dos modelos. Normalmente, são testados vários tipos de modelos. Neste caso, pretendeu-se utilizar pelo menos dois tipos de modelos: um modelo baseado em Redes Neurais e outro baseado em técnicas de aprendizado de máquina como florestas aleatórias ou *Random Forests*, Xgboost e Gradient Boosting. Em ambos os casos: o Os dados são divididos em um conjunto de treinamento e um conjunto de testes. o O modelo é ajustado aos dados de treinamento. o O modelo é avaliado com os dados de teste utilizando métricas como: precisão, área sobre a curva ROC, matrizes de confusão, etc. • Os algoritmos escolhidos para categorização dos e-mails foram o Máquina de Vetores de Suporte (SVM), Naive Bayes Multilinear, k-Vizinhos Mais Próximos (k-NN), Regressão Logística (RL).

	• Foi sempre levado em consideração que os diferentes modelos podem ter pontos fortes e fracos. • Foi avaliado o impacto das diferentes formas de combinação das classificações dos modelos.
6. Teste e calibração de hiperparâmetros	• Os modelos desenvolvidos na etapa anterior serão testado. • Os modelos usarão os valores padrão ou valores estimados com base nas melhores práticas. • Da mesma forma, os parâmetros foram reavaliados para otimizar o desempenho do modelo.
7. Implantação do modelo	• Após o treinamento do modelo, serão feitas as adaptações necessárias para produzir o modelo.
8. Avaliação dos resultados	• Nessa etapa foi feita a avaliação dos resultados com as métricas escolhidas. • Para medir o desempenho dos classificadores foram utilizadas as métricas a seguir: acurácia (exatidão), precisão, revocação e o F1-Score, por serem métricas simples e amplamente utilizadas em diferentes estudos científicos. • As métricas resultantes das três execuções dos algoritmos de m*achine learning* foram comparadas para verificar a consistência dos resultados. • O cálculo das médias e desvios padrão também foram calculados para analisar o comportamento dos algoritmos utilizados. • Utilizou-se matrizes de confusão normalizadas para verificar visualmente os algoritmos de aprendizado de máquina que apresentaram os melhores resultados. • Finalmente, o tempo de execução de cada algoritmo foi analisado após notar um alto intervalo de execução para o algoritmo de Regressão Logística.

Fonte: Elaborado pela autora com base em Adapting (2021).

Esta metodologia permitirá incorporar os novos modelos aos *workflows* ou fluxo de trabalho padrão de gerenciamento de documentos, conforme mostra a Figura 33:

Figura 33 – Fluxograma de trabalho de um sistema de gestão de documentos eletrônicos de arquivo com previsões inteligentes



Nota: CAT* = Categoria; Car. = Característica ou *feature*; ***conf. = confiança; ****NS/NR = Não Sabe/Não Responde.
Fonte: Elaborado pela autora com base em Adapting (2021).

Mas para desenvolver esse trabalho, foram utilizados diferentes algoritmos e ferramentas como se detalha a seguir.

Com o intuito de facilitar e garantir a reprodutibilidade da metodologia utilizada e avançar no campo objeto do estudo aqui realizado, detalha-se a continuação algumas informações sobre a **identificação do problema de pesquisa e solução proposta** (onde se descreve a base de dados ou *dataset* utilizada nos experimentos, se esclarece a procedência dos dados e quais atributos são importantes para esse trabalho, quais métricas de avaliação experimental, parâmetros, tecnologias, configurações e treinamentos a priori foram utilizados, além do contexto no que foi conduzido o estudo, os materiais, técnicas, bases de dados, procedimentos, ferramentas, *frameworks* teóricos, etc.); sobre o **pré-processamento** (especificando quais recursos foram utilizados para pré-processar a base de dados, que atributos foram selecionados, se foi reutilizado algum procedimento para transformar dados não estruturados em estruturados, o procedimento utilizado para balancear os dados das diferentes classes ou categorias, e porque foi preciso excluir alguns registros devido a informações faltantes ou duplicadas); sobre **a extração de Padrões** (especificando os algoritmos que foram utilizados para extração de padrões, as medidas utilizadas para medir o desempenho do modelo de classificação e os parâmetros que foram considerados na execução dos algoritmos); e sobre os **Experimentos realizados** (especificando qual foi o design do experimento, o objetivo do experimento e as conclusões preliminares).

3.2 Identificação do Problema

3.2.1 Problema de pesquisa e solução proposta

Após aprofundar na literatura científica e técnica foi encontrado um problema e desafio nas entidades públicas e privadas para organizar e classificar o grande volume de informações que recebem ou produzem de forma que facilite a sua posterior recuperação, consulta, pesquisa e gerenciamento, alcançando assim eficiência, inteligência corporativa, transparência e cumprimento da regulamentação associada.

Normalmente, as entidades optam pela aquisição ou aluguel de plataformas profissionais de gerenciamento de documentos como os SGDE(A) que tem se tornado a solução mais frequente e eficaz. Porém, a responsabilidade final pela classificação e gestão documental cabe aos próprios usuários da plataforma, que muitas vezes não investem o tempo necessário, ou não têm conhecimento completo das tabelas de classificação, ignorando se existem documentos, procedimentos e/ou arquivos semelhantes onde localizar

os documentos recebidos. Desta forma, uma infinidade de documentos e petições são retidas, gerando deficiências na manutenção do sistema documental e riscos com o cumprimento de determinados prazos legais.

Diante desse contexto e problema, propõe-se nesse trabalho desenvolver um Sistema Automático de Classificação de Documentos com base em modelos de *machine learning*. Tentando assim responder à questão de pesquisa (**Q1**): *Os Sistemas Automáticos de Classificação de Documentos com base em modelos de machine learning apresentam otimização* nos processos operacionais e *um maior desempenho de organização, classificação e recuperação de informação em relação a sistemas tradicionais?*

Como essa proposta espera-se que os resultados mostrem que os sistemas automáticos de classificação de documentos com base em modelos de aprendizado automático ou ML apresentem maior desempenho e otimização nos processos operacionais em relação a sistemas tradicionais.

3.2.2 Origem, construção e descrição da base de dados utilizada nos experimentos e amostras

Neste trabalho, a coleção de textos ou documentos utilizadas para desenvolver os experimentos é oriunda de *Kaggle* (https://www.kaggle.com). *Kaggle*, é uma plataforma de dados públicos subsidiaria de Google LLC, é uma comunidade online de cientistas de dados e profissionais de *machine learning*. A plataforma permite o acesso gratuito a GPUs e uma grande quantidade de dados e códigos publicados pela comunidade. Na *Kaggle*, você encontrará os códigos e os dados necessários para realizar seus projetos de ciência de dados. Existem mais de 50.000 conjuntos de dados públicos e 400.000 notas públicas disponíveis para todos. O *Kaggle* permite que os usuários encontrar e publicar conjuntos de dados, além de permitir explorar e criar modelos em um ambiente de ciência de dados baseado na Web.

A plataforma tem diferentes serviços que facilita tanto o trabalho colaborativo entre cientistas de dados e engenheiros de *machine learning* quanto a participação através de competições para resolver desafios de ciência de dados. De fato, o *Kaggle* começou em 2010 oferecendo concursos de aprendizado de máquina e agora também oferece uma plataforma pública de dados, uma base de dados baseada em nuvem para ciência de dados e educação em IA. Sua equipe inicial chave foram Anthony Goldbloom e Jeremy Howard. Nicholas Gruen foi o presidente fundador, sucedido por Max Levchin. O patrimônio líquido em 2011 subiu para valorizar a empresa em US$ 25 milhões.

Os temas abordados no Kaggle são muito variado e a plataforma propõe diferentes projetos para estimular o trabalho colaborativo e o aprendizado, oferecendo também a possibilidade de discutir com líderes e especialistas do setor. De fato, essa plataforma web reúne a maior comunidade de *Data Science* do mundo, com mais de 536 mil membros ativos em 194 países , recebe mais de 150 mil publicações por mês, que fornecem todas as ferramentas e recursos mais importantes para progredir ao máximo em ciência de dados. O Kaggle, igual que o *DataScientest,* possui uma interface *Jupyter Notebooks* (https://jupyter.org/) personalizável e que não precisa de configuração.

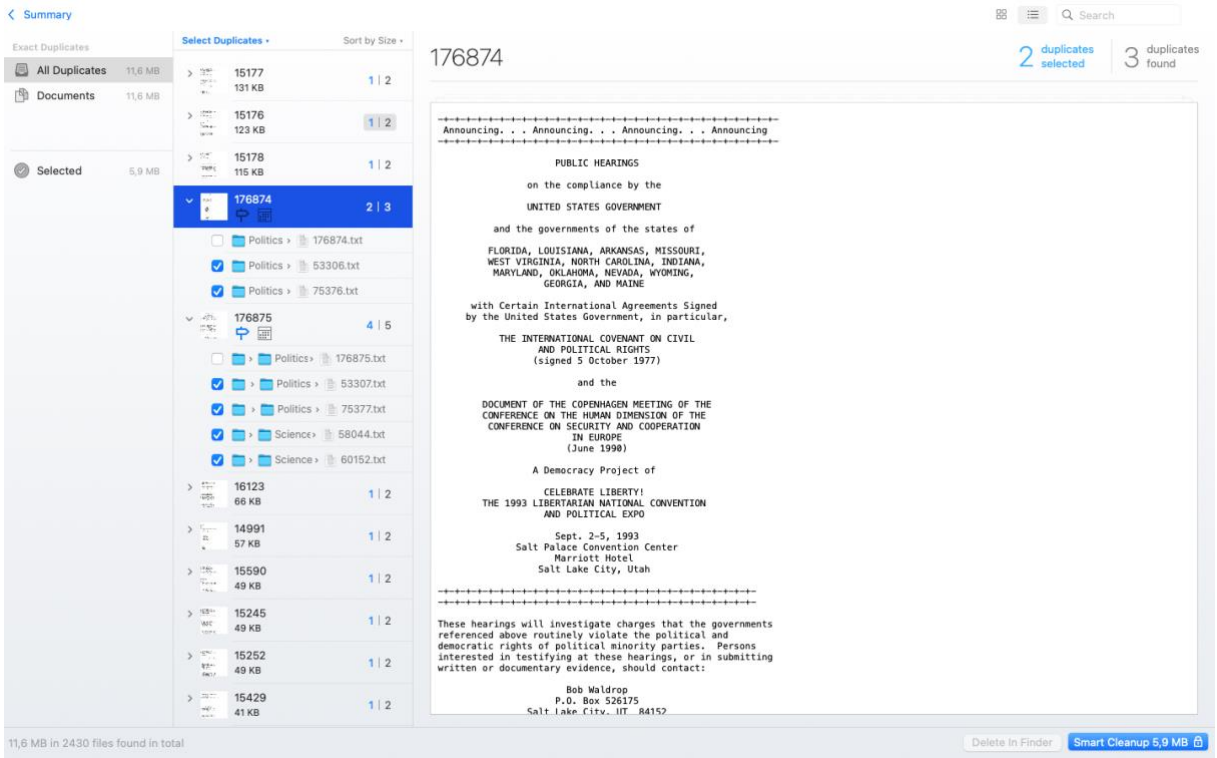
Algumas das maiores empresas de ciência de dados do mundo, como *Walmart* ou *Facebook*, confiam no *Kaggle*. Essa plataforma permite que especialistas em dados e outros desenvolvedores participem de concursos de aprendizado de máquina e desafios de dados, escrevam e compartilhem código e salvem conjuntos de dados.

Assim, da base de dados públicos *Kaggle* foi utilizado um banco de dados composto por um total de 11.309 e-mails que intercambiam os jornalista de um jornal e seu editor chefe também recebe de forma desordenada numa caixa de entrada e os segregou em três categorias com as seguinte quantidade de arquivos textuais em formato ".txt": crime (1100), entretenimento (1053), politica (3001) e ciência (4000). Assim, os dados fornecidos pelo editor contêm pastas rotuladas por suas categorias e dentro delas estão arquivos de texto, que são dados não estruturados. Além disso, os arquivos de texto são nomeados como um identificador exclusivo, exemplo: 52722.txt. Infelizmente, o departamento de Tecnologia da Informação (TI) do jornal não fez um trabalho de classificação adequado mantendo os dados nessa ordem e salvando os arquivos de texto rotulados em várias pastas de forma duplicada, por exemplo: o arquivo chamado ou identificado como "52723.txt" está rotulado de forma repetida nas pastas *Crime*, *Entretenimento* e *Ciência*. Assim, a classe "Entretenimento" (*Entertainment*) foi desconsiderado por não ter dados exclusivos para sua classe. Os dados e os rótulos originais estão escritos em idioma inglês: *Crime, Enternainment, Politcs* e *Science*. Por outro lado, encontrou-se arquivos com diferentes rótulo ou nome mas com o mesmo conteúdo, como por exemplo os arquivos identificados como "176874.txt", o "53306.txt" e "75376.txt" repetido 3 vezes ou os arquivos chamados "176875.txt", "53307.txt", "75377.txt" e "60152.txt" repetido 4 vezes como mostrado na Figura 34. Então, foram eliminados os arquivos duplicados ou repetidos com o *software*

Gemini 2[4] para Mac deixando apenas a versão mais recente do documento. Dessa forma se pretende balancear os dados e evitar problemas na construção dos modelos com ML.

A escolha de e-mail vem justificada, no apenas pela dificuldade de encontrar uma *dataset* público com o que trabalhar, mas também pela importância que tem alcançado nos últimos anos o uso de mensagens de e-mail como meio de comunicação e de recuperação de informação na sociedade, seja no contexto profissional ou pessoal. Além disso, diante do elevado volume de informação recebida via e-mail diariamente, surge a necessidade de organizar essas mensagens para evitar a perda de informação relevante. Nesse contexto, o principal problema é a falta de tempo disponível para organizar manualmente essas mensagens, uma tarefa que é suscetível de automatização para lograr a eficiência na gestão de tempo e outras tarefas como a resolução de problemas, por exemplo, que requerem da inteligência humana e que a máquina não consegue. Surgindo assim, a necessidade de classificar automaticamente o corpus documental que conforma o conjunto de dados desse trabalho de pesquisa.

Figura 34 – Visualização de alguns arquivos duplicados

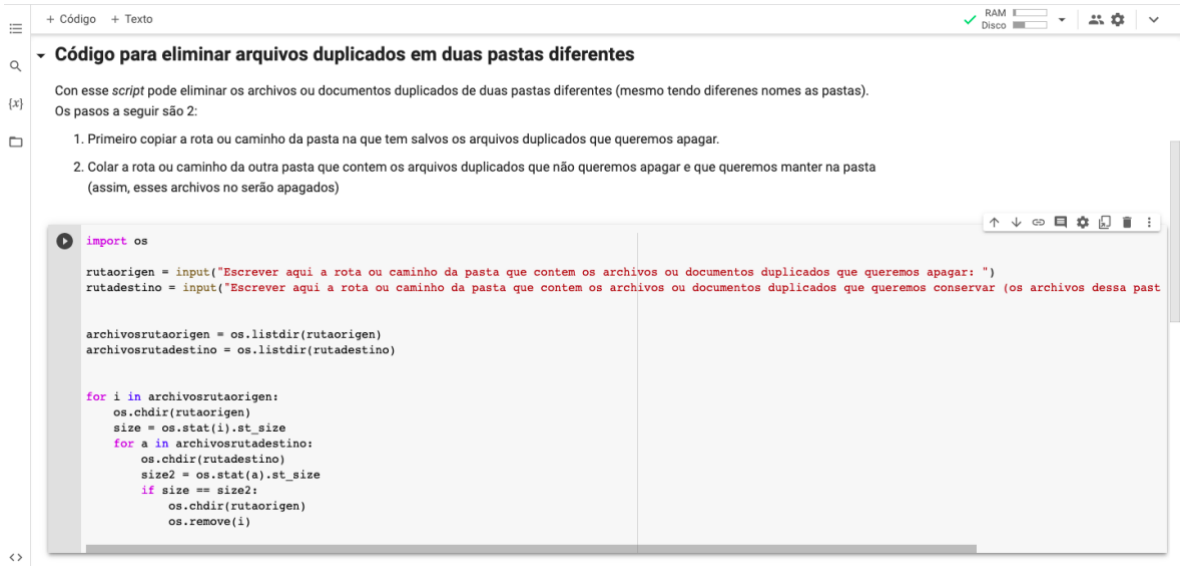


Fonte: Elaborado pela autora com o software Gemini 2.

[4] Gemini 2 é uma solução inteligente, pois é localizador ou buscador de arquivos duplicados inteligente que localizea arquivos duplicados e os remove diretamente facilitando assim a recuperação de espaço na memoria do Mac. Tem versão gratuita. O enlace web desse *software* é: https://macpaw.com/es/gemini.

Fazendo um segundo filtrado para detectar documentos repetidos no *dataset*, utilizou-se o *script* da Figura 35 realizado em P*ython* no *que está disponível no Google Colab*[5].

Figura 35 – *Script* de Google Colab para eliminar arquivos duplicados do *Dataset*



Fonte: Elaborado pela autora com base em Maalfer[6].

Os dados de entrada são compostos por arquivos de texto em formato ".txt" dentro das pastas com rótulos de classe como nome. Os dados textuais são e-mails escritos de um jornalista para outro ou para sua fonte a respeito de um tema. Os e-mails aparecem rotulados em várias classes, o que poderia enganar o modelo. O conteúdo dos documentos foi armazenado em sua forma bruta e limpa (sem *tags* HTML), em arquivos de extensão ".txt". Os arquivos de cada documento foram armazenados em uma pasta separada para facilitar a classificação manual e a verificação dos resultados.

O estudo foi realizado desde setembro de 2021 até julho de 2022 seguindo as etapas metodológicas descritas na Tabela 2 dessa seção. A seguir, na Tabela 3, apresentam-se as caraterísticas do conjunto de dados de base:

Tabela 3 – Caraterísticas do *Datasets*

BASE DE DADOS ORIGINAL		**BASE DE DADOS SEM A CLASSE "Entertainment"**		**BASE DE DADOS SEM DUPLICADOS**		**BASE DE DADOS UTILIZAZA**	
Instancias	11.309	**Instancias**	8.101	**Instancias**	6862	**Instancias**	3000
Atributos	3	**Atributos**	3	**Atributos**	3	**Atributos**	3

[5] *Script* para eliminar arquivos duplicados: https://colab.research.google.com/drive/1k58nVmG4nDEt6YoQYfn_wpVXpml-px3g?usp=sharing.
[6] https://github.com/Maalfer/Borrar_Archivos_Duplicados/blob/main/duplicados.py

Tipo de atributo	texto	**Tipo de atributo**	texto	**Tipo de atributo**	texto	**Tipo de atributo**	texto
Classes / Categorias	4	**Classes / Categorias**	3	**Classes / Categorias**	3	**Classes / Categorias**	3
Estado do *Dataset*	Desbalan-ceado	**Estado do *Dataset***	Des-balanceado	**Estado do *Dataset***	Semi-balanceado	**Estado do *Dataset***	Balan-ceado

Fonte: Elaborado pela autora.

3.2.3 Implementação dos algoritmos, ferramentas utilizadas, abordagem e framework

Para a **implementação dos algoritmos** nessa pesquisa foi utilizada a linguagem de programação Python (https://www.python.org/) junto com as bibliotecas do Scikit-Learn ou Sklearn (https://scikit-learn.org/stable/), que, segundo Pedregosa et al. (2011), é um módulo Python que integra uma ampla gama de algoritmos de *machine learning* de última geração para resolver problemas supervisionados e não supervisionados de média escala. Este pacote leva o aprendizado de máquina para não especialistas usando uma linguagem de alto nível de uso geral.

Seguindo a Méndez et al. (2017), dentre as diferentes **ferramentas para processar e classificar documentos textuais** de código aberto que poderiam ser usadas (Lemur, Indri, Weka, Lucene, LingPipe, etc.) foi escolhida a ferramenta Orange 3[7], por ser um *software livre e a*daptado para tarefas relativas ao desenvolvimento de aplicações usadas em ambiente cientifico. Orange 3 é utilizado para o análise inteligente de texto por meio de *data mining*, processamento e classificação também foi utilizado o pacote Orange3-3.32.0-Python3.8.8.dmg[8], que é um pacote universal com tudo embalado e pronto para ser usado. Orange3 é um programa de computador ou *software* para mineração de dados e análise preditiva desenvolvido na Faculdade de Informática da Universidade de Ljubljana. Consiste em uma série de componentes desenvolvidos em C++ que implementam algoritmos de mineração de dados, bem como operações de pré-processamento de dados e representação gráfica. Os componentes de Orange3 podem ser manipulados a partir de programas desenvolvidos em Python ou por meio de um ambiente gráfico e é distribuído sob a licença GPL.

Também foi utilizado o *Google Colab*[9] ou *Colaboratoryte* permite programar e executar Python no navegador com as seguintes vantagens: No requer configuração, da

[7] Orange3: https://orangedatamining.com/
[8] Pacote de Orange: https://download.biolab.si/download/files/Orange3-3.32.0-Python3.8.8.dmg
[9] Google Colab: https://colab.research.google.com/?hl=es

acesso gratuito ao GPUs e permite compartilhar conteúdo facilmente. *Colab* facilita o trabalho do estudante, científico de dados o pesquisador de IA.

Por outra parte, foi usado o Scikit-learn é um *framework open-source de machine learning* para Python acessível para todos e reutilizável em vários contextos. Na atualidade, Scikit-learn é considerada uma das melhores bibliotecas e a mais eficiente para aplicações de *data mining* e de análise de dados. Esta biblioteca tem um amplo conjunto de algoritmos de aprendizado de máquina supervisionados e não-supervisionado para diversas finalidades, a partir de uma interface consistente e de fácil usabilidade (SANTOS, 2015). O Scikit-learn é baseado em outras bibliotecas: NumPy (um pacote para manipulação de arrays N-dimensional), SciPy (um pacote para computação científica), Matplotlib (um pacote para gráficos 2D e 3D), IPython (um pacote para terminal interativo de comandos), Sympy (um Pacote para matemática simbólica), e Pandas (um pacote para a analise de estruturas de dados).

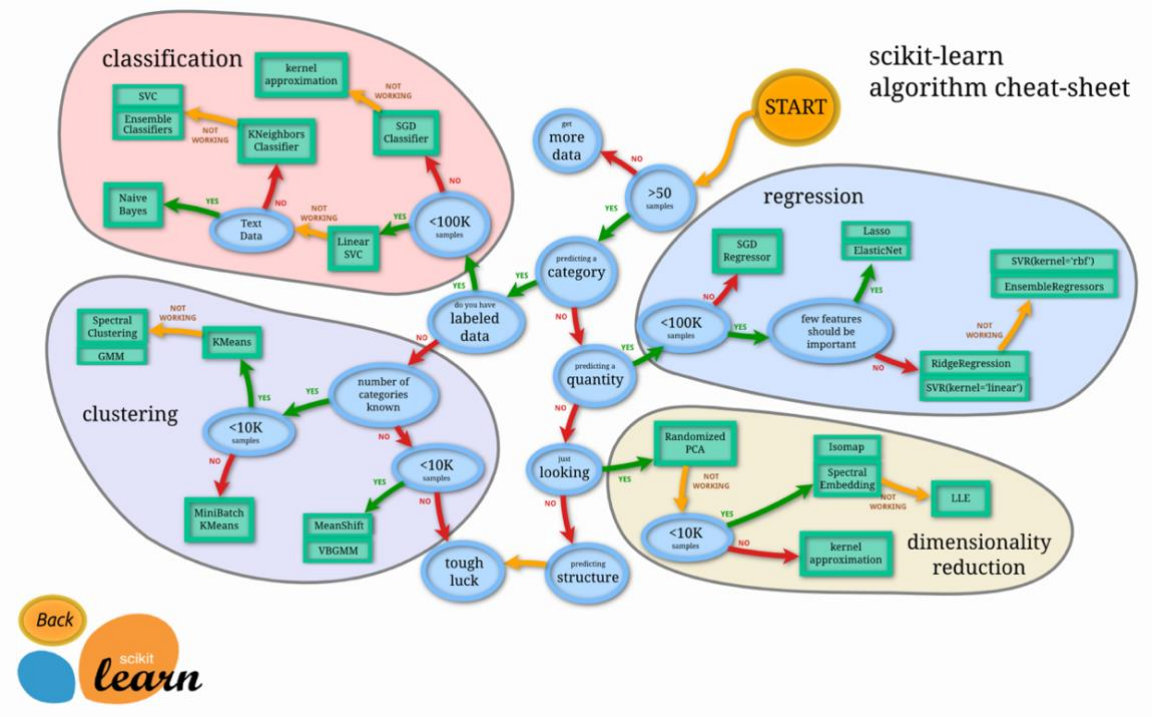
A biblioteca Scikit-learn oferece suporte e robustez para ambientes de produção e também se centra na usabilidade, limpeza de código, colaboração, além de tem bastante documentação e um excelente desempenho. As funções oferecidas utilizam as facilidades da linguagem Python, mas possuem mecanismos implementados em linguagens de baixo nível, como C, para otimizar ao máximo o uso de recursos de *hardware*. O Scikit-learn é utilizado por grandes empresas e tem uma grande comunidade (SANTOS, 2015).

As principais funcionalidades oferecidas pelo Scikit-learn são: Clustering (algoritmos não-supervisionados para agrupar dados não categorizados, como K-means); Cross Validation (técnica para validar modelos preditivos de algoritmos supervisionados contra dados não vistos); Datasets (oferece um conjunto de bases de dados para aplicar as diferentes técnicas de aprendizado); Dimensionality Reduction (para reduzir o número de atributos a serem analisados de forma a otimizar a análise); Ensemble methods (para combinar das predições de diversos modelos); Feature extraction (para definir atributos em dados não estruturados como imagem e texto); Feature selection (para identificar atributos estatisticamente relevantes para a criação de modelos preditivos supervisionados); Parameter Tuning (para otimizar a execução de modelos preditivos supervisionados), e Manifold Learning (para fazer uma análise descritiva de dados complexos e multidimensionais).

Em muitos casos o mais difícil é encontrar ou escolher o modelo ou estimador mais apropriado para resolver um problema de aprendizado de máquina, pois existem diferentes modelos que são mais adequados para determinados tipos de dados e diferentes problemas.

Para auxiliar nessa tarefa, a Figura 36 apresenta um fluxograma com os modelos e funcionalidades preditivos supervisionados e não supervisionados de *Scikit-learn*, fornecendo aos usuários um guia aproximado sobre como abordar problemas em relação a quais modelos devem ser testados em seus dados. Essa imagem (Figura 36) é interativa desde o site da biblioteca[10].

Figura 36 – Modelos Scikit-learn

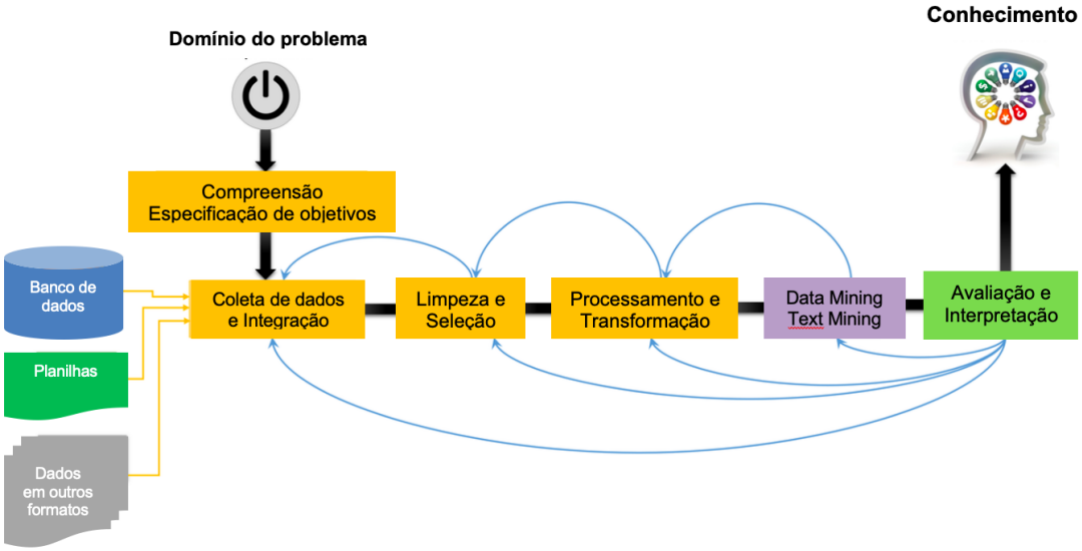


Fonte: http://scikit-learn.org/stable/tutorial/machine_learning_map/

Como base na ideia de extrair conhecimento e não apenar gerar novos dados da base de dados utilizada, os experimentos desse trabalho se basearam no *framework teórico* apresentado na Figura 37 onde a mineração de dados (*data mining*) e mineração de texto (*text mining*) está integrada no processo de descoberta e geração de conhecimento a partir dos bancos de dados, mais conhecido como KDD (*Knowledge Discovery in Databases*):

[10] Movie Review Data. (s.d.). Obtido em abril de 2022, de Cornell University - Department of Computer Science: http://www.cs.cornell.edu/people/pabo/movie-review-data/

Figura 37 – Fases do processo KDD na Meneração de dados e textos



Fonte: Traduzido de Charte (2020).

Assim, o esquema da Figura 37, deve ser lido da esquerda para a direita, começando com a compreensão do domínio a que correspondem os dados a serem utilizados e a especificação dos objetivos a serem perseguidos. A primeira etapa do KDD consiste em coletar os dados de todas essas fontes e homogeneizá-los e integrá-los, produzindo o que é conhecido como conjunto de dados ou *dataset*. Trata-se de uma base de dados com formato unificado e, além disso, com estrutura mais adequada para análise de dados.

Segundo Charter (2020), o processo de KDD está intimamente relacionado ao aprendizado de máquina e também à mineração de dados ou *Data Mining* (DM). Termos que, às vezes, se confundem e se confundem. Para alguns especialistas, KDD e DM são sinônimos, enquanto para outros DM é uma das fases do KDD. Ambos os conceitos existem há quase um século e implicam a intervenção manual do especialista na preparação dos dados em que se espera encontrar o conhecimento mencionado acima. Onde sim concordam os especialistas é que as técnicas de ML são mais tardias do que KDD e DM, e se concentram mais na construção dos algoritmos do que na preparação dos dados em si. O objetivo de um algoritmo de ML é, tomando padrões de dados como entrada, gerar um modelo que contenha o conhecimento extraído.

3.2.4 Principais bibliotecas utilizadas para o desenvolvimento da metodologia

Do mesmo modo, para facilitar a reprodutibilidade, a seguir são apresentadas as bibliotecas utilizadas no Kaggle que coincidem com as mencionadas na figura anterior:

Bibliotecas principais:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import os
```

Processamento de texto:

```
import regex
from wordcloud import WordCloud
from sklearn.feature_extraction.text import TfidfVectorizer
from nltk.corpus import stopwords
from nltk.tokenize import WordPunctTokenizer
from string import punctuation
from nltk.stem import WordNetLemmatizer
```

Métricas e Validação:

```
from sklearn.model_selection import train_test_split, RandomizedSearchCV
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score,
cohen_kappa_score
```

Entrenamento dos modelos com técnicas de ML:

```
from sklearn.ensemble import StackingClassifier, VotingClassifier
from sklearn.linear_model import LogisticRegression, SGDClassifier
from sklearn.naive_bayes import MultinomialNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.cluster import KMeans
import xgboost
```

Assim mesmo, a seguir, na Figura 33 são apresentadas as principais bibliotecas instaladas e importadas no Google Colab:

Figura 38 – Bibliotecas utilizadas no desenvolvimento metodológico

Fonte: Elaborado pela autora.

3.1 Pré-processamento

3.1.1 Recursos e técnicas utilizados para e pré-processar a base de dados

Para limpar nossos dados e obter alguns *insights* significativos, primeiro precisamos garantir que os dados estejam devidamente rotulados e armazenados. Para isso foram seguidos os seguintes passos: 1). Ler o texto de cada arquivo e, ao fazê-lo, certifique-se de que não estamos lendo dados duplicados. 2). Carregar os dados textuais no *DataFrame* para facilitar a análise. 3). Limpar o texto bruto por meio das seguintes técnicas:

- Remoção de caracteres especiais, pontuações, pronomes e palavras irrelevantes;
- Tokenização de cada ponto de dados, ou seja, segmentação de texto em frases e mais em palavras;
- Normalização do texto convertendo-o em minúsculas;
- Extração do lema para cada palavra, por exemplo: *Lemma(swimming)→swim.*

4). Balancear os dados dentro de cada categoria, escolhendo 1000 arquivos de texto de cada categoria ou classe utilizada (*Crime, politica* e *Ciência*), pois a categoria de *Entretenimento* foi removida, porque seus arquivos estavam contidos nas outras três categorias, eliminando, assim, a duplicidade dos dados e possíveis problemas no treinamento dos modelos.

Após limpar, balancear e transformar dados não estruturados em estruturados, primeiro foi realizada a vetorização dos dados limpos; segundo, foram treinados os modelos básicos aplicando as seguintes técnicas:

- Regressão Logística *com RandomizedSearchCV*.
- Multinomial Naive Bayes com *RandomizedSearchCV*.
- *Classificador SDG ou SGD Classifier com squaredhinge loss → SVM (Suport Vector Machinenes).*
- Classificador de Árvore de Decisão (*Decision Tree Classifier*).
- Classificador de K-Vizinhos **(***KNeighbors Classifier*).
- *XGBClassifer*.

Em terceiro lugar, foram combinados os modelos (*Ensembling base models*) por meio do *Hard Voting Classifier*, do empilhamento (*Stacking Classifier*) com XGBClassifier como o "estimador final" ou "meta-aprendiz" e por meio do SVC (*Soft Voting Classifier*). Em quarto lugar, foi realizada a comparação dos Modelos, e finalmente, foram testados os modelos.

3.1.2 Seleção de atributos e procedimento para balancear dados de diferentes classes

Para a seleção dos atributos e balancear dados de diferente classes ou categorias foi utilizado a análise inteligente de texto, ou seja, mineração de texto (*text mining*) com Python, aplicando as técnicas acima descritas. O *text mining* engloba o conjunto de técnicas que permitem estruturar a informação heterogênea presente em textos para identificar padrões como o uso de palavras, com as quais extrair novas informações. Para isso foi feita uma análise exploratória dos dados previamente. Com isso se logrou identificar as palavras mais usadas ou mais frequentes nos textos da base de dados objeto de estudo e criar um modelo de aprendizado de máquina capaz de classificar os textos.

Um dos principais interesses na área de mineração de texto (também conhecida como *text mining* ou *text analytics* ou mineração de texto, processamento de linguagem natural e recuperação de informação é quantificar o tema de um texto, bem como a importância de cada termo que o forma. Uma maneira simples de medir a importância de um termo dentro de um documento é usando a frequência com que ele aparece (*tf* , *term-frequency*). Essa abordagem, embora simples, tem a limitação de atribuir grande importância àquelas palavras que aparecem muitas vezes, embora não forneçam informações seletivas. A estatística *tf-idf* mede o quão informativo é um termo em um documento, levando em consideração a frequência com que esse termo aparece em outros documentos.

Para facilitar a visualização, os resultados foram apresentados nessa fase em forma nuvem de *tags* ou nuvem de palavras chave e histogramas com as palavras mais repetidas por classes ou categorias.

3.2 Extração de Padrões

3.2.1 Algoritmos utilizados para extração de padrões

Os algoritmos utilizados para extrair características o padrões da coleção de documentos utilizados foram: Máquina de Vetores de Suporte (SVM), *Naive Bayes Multilinear* (NBM), k-Vizinhos Mais Próximos (k-NN) e o de Regressão Logística (RL).

3.2.2 Métricas para medir o desempenho do modelo de classificação, métricas de avaliação experimental e parâmetros considerados na execução dos algoritmos

Para medir o desempenho dos classificadores foram utilizadas as métricas a seguir: acurácia, precisão, revocação e o F1-Score, por serem métricas simples e amplamente utilizadas

em diferentes estudos científicos. As métricas resultantes das três execuções dos algoritmos de *machine learning* foram comparadas para verificar a consistência dos resultados.

O cálculo das médias e desvios padrão também foram calculados para analisar o comportamento dos algoritmos utilizados. Utilizou-se matrizes de confusão normalizadas para verificar visualmente os algoritmos de aprendizado de máquina que apresentaram os melhores resultados. Finalmente, o tempo de execução de cada algoritmo foi analisado após notar um alto intervalo de execução para o algoritmo de Regressão Logística.

3.3 Experimentos realizados

Os experimentos foram realizados a fase de modelagem na que são treinados os modelos escolhidos. Assim, após a preparação dos dados, foram escolhidos os modelos selecionados a serem implementados para a modelagem nas fases de treinamento e teste dos modelos. Nessa Normalmente, são testados vários tipos de modelos. Neste caso, utilizaram-se dois tipos de modelos: um modelo baseado em redes neurais e outro baseado em técnicas de aprendizado de máquina como é florestas aleatórias e Xgboost. Em ambos os casos, os dados são divididos em um conjunto de treinamento e um conjunto de teste; o modelo é ajustado aos dados de treinamento; e o modelo é avaliado com os dados de teste utilizando métricas como: precisão, área sob a curva ROC, matrizes de confusão, etc.

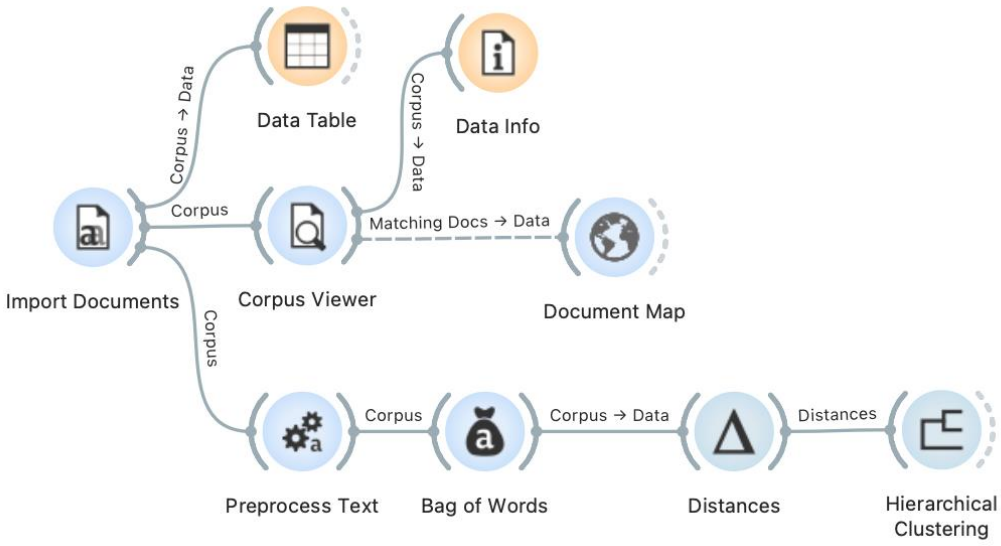
Os algoritmos escolhidos para categorização dos e-mails foram o Máquina de Vetores de Suporte (SVM), Naive Bayes Multilinear (NBM), k-Vizinhos Mais Próximos (k-NN) e o de Regressão Logística (RL), Redes Neurais (RN), Random Forest (RF) e Gradient Boosting (GB).

3.3.1 Design e objetivo do experimentos

Os experimentos foram planejados e executados com o intuito de verificar o impacto da dos modelos em diferentes cenários de classificação documental. Depois forma avaliados os experimentos e nessa avaliação experimental foram executadas as três etapas centrais do processo de Mineração de Textos. Na etapa de Pré-processamento, as coleções de documentos foram representadas como BoW. Na etapa de Extração de Padrões, foram utilizados um conjunto de algoritmos tradicionais na área de Aprendizado de Máquina, presentes na ferramenta Orange data mining 3. Também foram executados experimentos combinando classificadores gerados por diferentes representações. Na etapa de Pós-processamento, os modelos gerados foram avaliados por meio da medida acurácia entre outras já descritas na seção dedicada à metodologia.

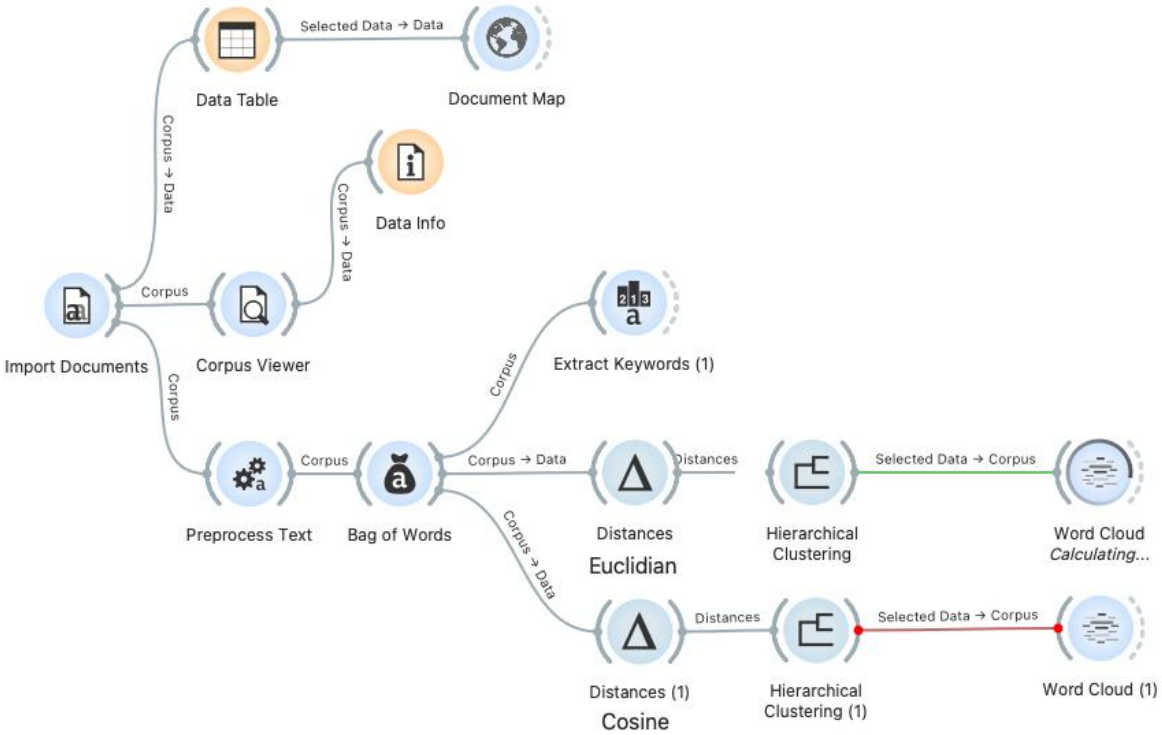
O *design* do experimentos foi realizado no Orange data mining 3 como se mostra na Figura 30, 40, 42 e 43.

Figura 39 – Fluxograma e Framework da fases metodológicas de preparação de dados e pré-processamento no Experimento 1



Fonte: Elaborado pela autora com Orange data mining.

Figura 40 – Fluxograma e Framework da fases metodológicas de preparação de dados e pré-processamento no Experimento 2

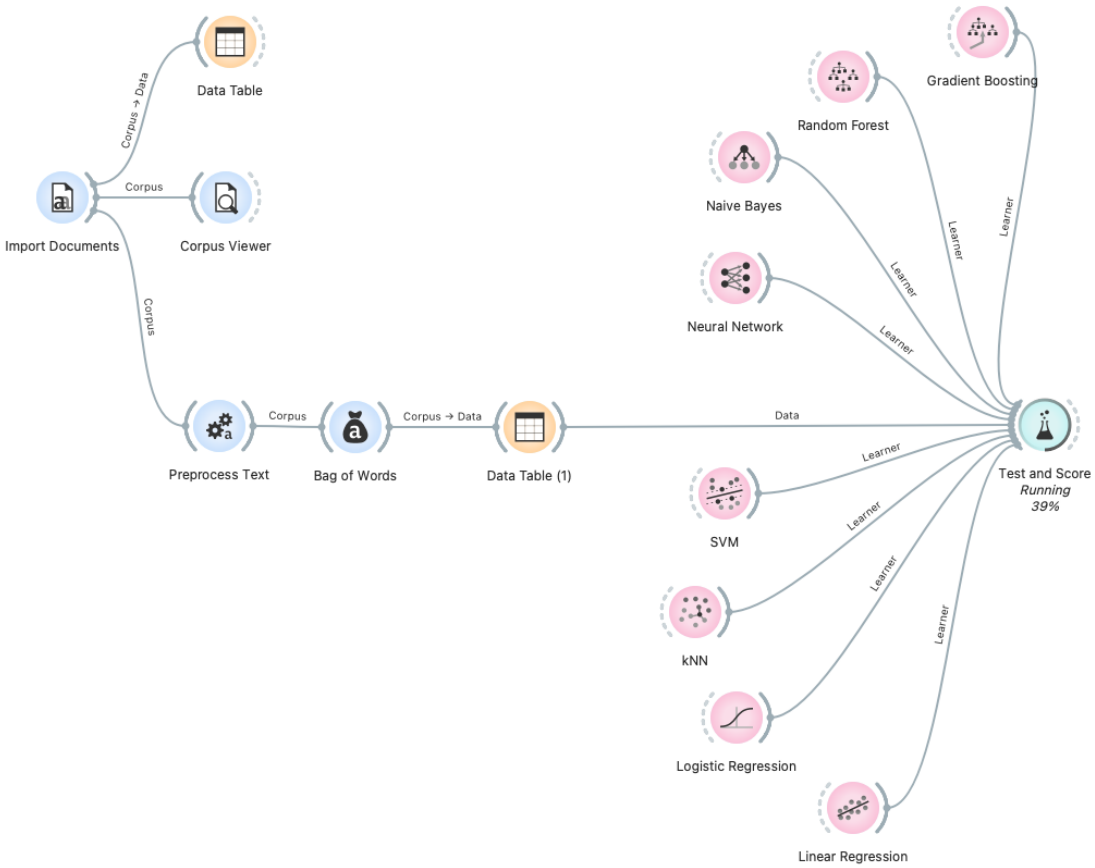


Fonte: Elaborado pela autora com Orange data mining.

Figura 41 – Fluxograma e Framework das fases de preparação de dados e pré-processamento no Experimento 3

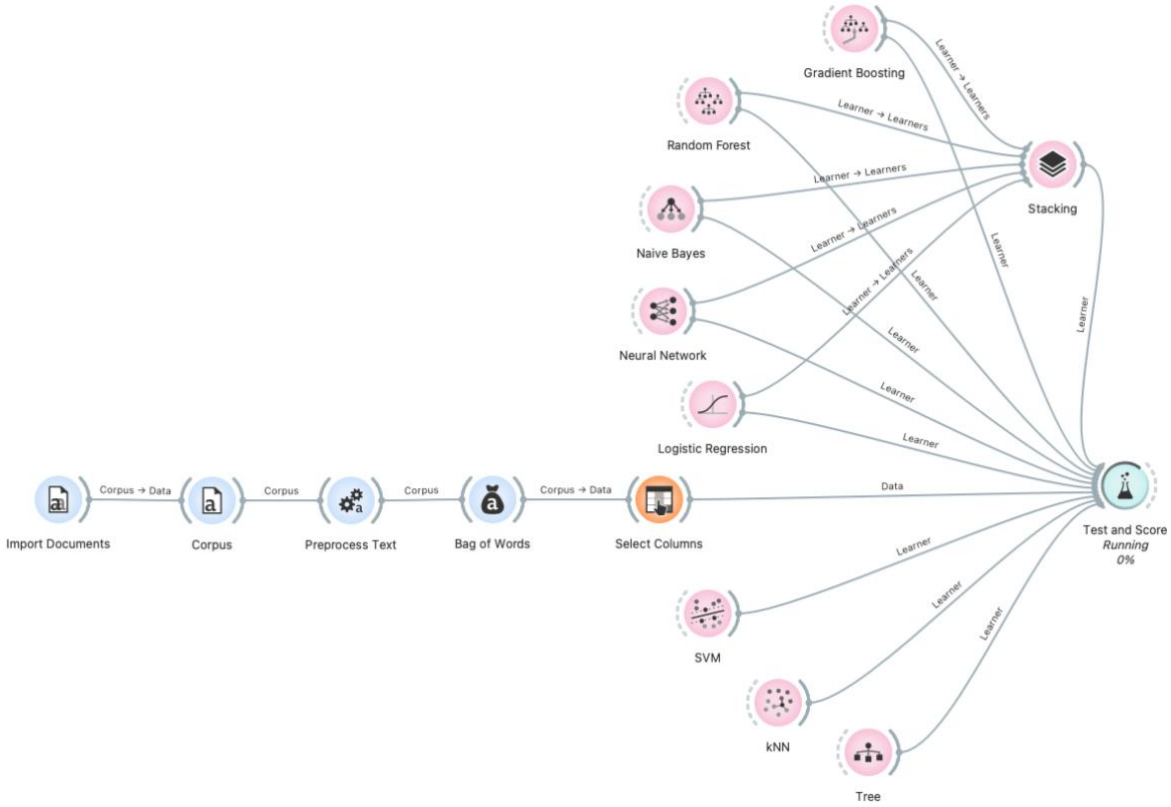
Fonte: Elaborado pela autora com Orange data mining.

Figura 42 – Fluxograma e Framework da fase de Modelagem no Experimento 1



Fonte: Elaborado pela autora com Orange data mining.

Figura 43 – Fluxograma e Framework da fase de Implementação no Experimento 2



Fonte: Elaborado pela autora com Orange data mining.

Os objetivo principal dos experimentos foi poder responder a nossa pergunta de pesquisa, ou seja, saber se os *sistemas automáticos* de classificação documental com base em modelos de ML otimizam os processos operacionais. Assim, buscou-se com os experimentos conseguir um maior desempenho na organização, classificação e recuperação de informação em relação a sistemas tradicionais, pré-avaliar o impacto das diferentes formas de combinação das classificações dos modelos, mapear algoritmos de aprendizado de uma única classe analisando suas lacunas e desempenhos de classificação na detecção de documentos, considerando tanto cenários balanceados quanto cenários desbalanceados.

Utilizou-se pelo um modelo baseado em redes neurais e outro baseado em técnicas de aprendizado de máquina como florestas aleatórias (RF) e Xgboost ou Grandient Boosting. Também se pretendeu com os experimentos descrever os pontos fortes e fracos dos diferentes modelos utilizados e avaliar o impacto das diferentes formas de combinação das classificações dos modelos e comparar os desempenhos obtidos a algoritmos de classificação usados.

3.3.2 Conclusões preliminares

Os resultados permitem concluir de forma preliminar que sistemas automáticos de classificação documental com base em modelos de aprendizado de máquina otimizam os processos operacionais nas fases de organização, classificação e recuperação de informação em relação a sistemas tradicionais. Pois, concordando com Charte (2020), ao contrário dos algoritmos tradicionais de processamento de dados, os algoritmos de aprendizado de máquina processam dados e o que eles geram como saída não são apenas dados novos, mas sim um modelo que representa o conhecimento extraído. De fato, faz parte da Ciência de dados (*Data Science*) descoberta de conhecimento em bases de dados, isso é conhecido com o termo KDD (KDD (*Knowledge Discovery in Databases*).

4 RESULTADOS E DISCUSSÃO

4.1 Coleta e análise dos dados (Fases da I a III da metodologia)

Nessa seção se descreve e mostra os resultados das fases 1 a 3 da metodologia. Na primeira fase que foca na coleta de dados e geração do corpus textual para ser analisado e classificado, após saber quantos arquivos são rotulados em cada classe, foram removidos os pontos de dados rotulados em outras classes. Por exemplo: “52723.txt” está rotulado em “Crime”, “Entretenimento” e “Ciência”, pelo que foi removida as entrada da pasta com textos sobre entretenimento rotulado como “Entertainment”.

O risco aqui é ter removido a entrada de uma classe rotulada corretamente. Por exemplo: o arquivo de texto chamado “52724.txt” pode ter sido rotulado inicialmente como 'Ciência' (“Science”) e ter sido repetido em outras classes. Assim, removendo-o da classe ou categoria inicial “Science” se estaria rotulando erroneamente os dados como “Crime”.

Mas “Science” já contém o maior número de entradas, tornando mais fácil o treinamento do modelo para aquela classe em particular, ou seja, para a classe “Science”. Nesse ponto, cabe destacar que essa abordagem adotada não afeta a análise. Também cabe anotar que fica claramente implícito que a classe “Entretenimento” (“Entertainment”) não tinha dados exclusivos para sua classe. Pelo que ao não considerá-la, também é eliminada qualquer variação que possa ser causada pelos dados duplicados. Mas ainda existiam alguns textos duplicados que podem ter sido resultado de um má gerenciamento de dados ou do envio do mesmo correio eletrônico várias vezes.

Além disso e após gerar uma tabela com os dados do *datasets*, foi observado que tinha alguns e-mails nas categorias ou classes selecionadas que estavam vazios ou apenas incluíam um agradecimento curto, pelo que foram eliminados. Esses documentos removidos foram os seguintes documentos como apresentados na Tabela 4:

Tabela 4 – Documentos removidos do *dataset* balanceado

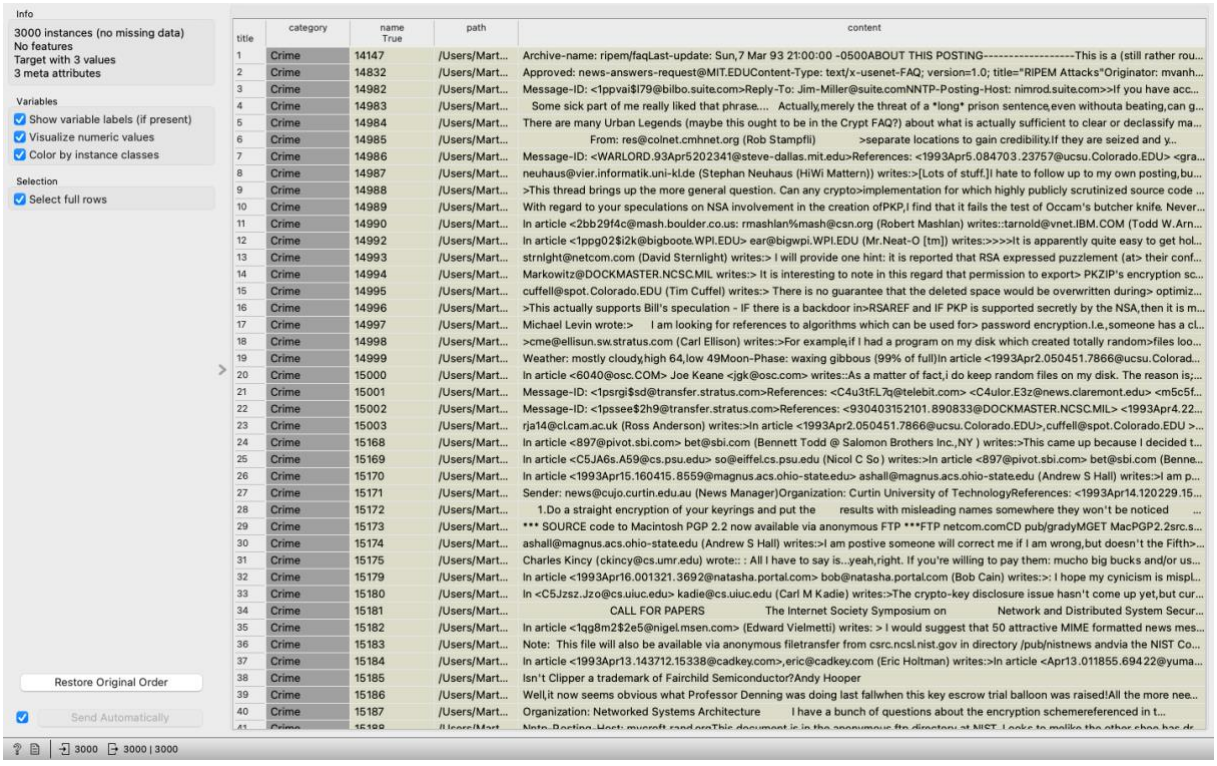
Categoria/Classe	Nome do documento eliminado	Conteúdo do e-mil
“Crime”	15387.txt	Vazio
“Crime”	15822.txt	Vazio
“Crime”	16027.txt	Vazio
“Crime”	16031.txt	Vazio
“Crime”	16141.txt	“Please post to news, too.”
“Politics”	54456.txt	Vazio
“Politics”	54457.txt	Vazio
“Science”	15387.txt	Vazio

“Science”	15822.txt	Vazio
“Science”	53760.txt	“Thanks a lot!”
“Science”	54025.txt	“This is just a test to see if this works.”
“Science”	54149.txt	“Thanks a lot!”
“Science”	54258.txt	Vazio
“Science”	54288.txt	“Thanks a lot!”
“Science”	58063.txt	Vazio
“Science”	58064.txt	Vazio
“Science”	58102.txt	“How about posting one of her replies to your letters? -km.”
Total documentos eliminados	**17**	

Fonte: Elaborado pela autora com os dados da pesquisa.

Pelo que finalmente, com o intuito de balancear os dados e evitar problemas na modelagem dos dados e treinamento dos modelos de ML, foram selecionados apenas 1000 arquivos de cada uma das classes a considerar para o treinamento dos modelos, ou seja, 1000 arquivos ou documentos da classe “Crime”, 1000 da classe “Politics” e 1000 da classe “Science”, tendo um total de 3000 arquivos para montar o *dataset* dos experimentos. Na Figura 44 se mostra a tabela em formato “.csv” com os dados do *dataset*, no que pode se observar o número total de arquivos ou documentos que serão tidos em consideração para conformar o *dataset* balanceado que conforma o corpus textual a ser analisado e classificado.

Figura 44 – Tabela com o *dataset* balanceado

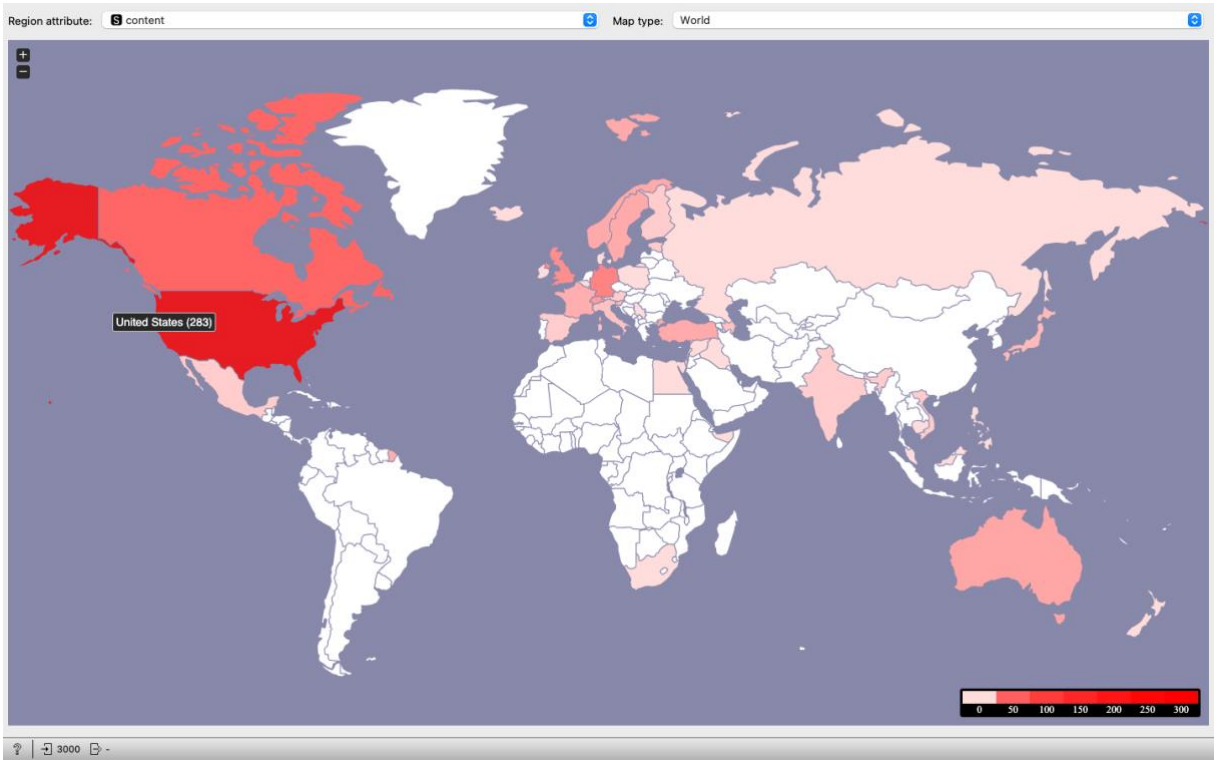


Fonte: Elaborado pela autora no Orange 3 com os dados da pesquisa.

Agora que foi carregada o corpus textual ou *dataset*, a seguinte tarefa é limpar os dados para poder encontrar alguns *insights* úteis sobre o conjunto de dados construindo nuvens de palavras e encontrando frequências de termos em cada classe.

Na Figura 45 é apresentado um mapa dos documentos que mostra a geolocalização dos dados textuais (*string*) dos e-mail que conformam o *dataset*. No mapa aparecem as menções de nomes geográficos (países e capitais) nos e-mails analisados e exibe a distribuição (frequência de menções) desses nomes no mapa.

Figura 45 – Mapa do conteúdo documental por países



Fonte: Elaborado pela autora desde Orange data mining com os dados da pesquisa.

A distribuição do conteúdo dos e-mail (Figura 45) foi a seguinte: 283 entradas de palavras referentes aos Estados Unidos de América (USA), 43 do Canadá, 21 da Alemanha, 16 do Reino Unido (UK), 11 da Suíça, 8 da Australia, 7 da Turquia, 7 da Normandia, 7 da Suécia, 6 da França, 4 da Eslovênia, 4 do Japão, 3 do Azerbaijão, 3 da Itália, 3 de Israel, 2 da Espanha, 2 da Áustria, 2 de Finlândia, 2 de Estônia, 2 da Índia, 2 do Vietnam, 2 de México, 1 da Nova Zelândia, 1de Servia, 1 da Rússia, 1 da Polonia, 1 da Síria, 1 de Iraque, 1 do Egito, 1 da Somália, 1 de Sud África, 1 da Bélgica, 1 da Dinamarca, 1 do Cambaia, 1 de Malásia, 1 de Filipinas.

O mapa é gerado desde a tabela de dados criada previamente que contem atributos de tipo *string*. Assim, foram geradas instâncias dos dados textuais selecionadas, ou seja, de todos

os documentos que contêm menções a um país. Dessa forma, o Mapa documental ou Mapa de documentos apresenta as distribuições de geolocalizações por atributo de país. Um atributo que já contém *tags* de país para cada e-mail.

4.2 Limpeza, engenharia de variáveis e pré-processamento dos dados (Fase IV da metodologia)

Nesse quarta fase de engenharia e pré-processamento, os dados ficam limpos, mas ainda não estruturados. Para estruturar os dados, podemos transforar esses dados textuais em números. Mas agora o foco, nesse ponto dos experimentos, é a fase de pré-processamento e a posterior fase de modelagem para testar os modelos, segundo a metodologia proposta neste trabalho.

A seguir são apresentados as resultado de aplicar as técnicas utilizadas para limpeza e pré-processamento dos documentos nos experimentos.

Após, coletar e analisar os dados nas fases metodológicas da I a III, os documentos selecionados para conformar o corpus balanceados foram pré-processados, os textos foram convertidos em palavras chave e foram removidas as palavras irrelevantes. Durante o pré-processamento foram obtidos 429876 palavras tokens e 23654 tipos de tokens, após transformar todos os textos do corpus de maiúscula para a minúscula, remover as URLs, detectar as etiquetas ou *tags* html e analisar apenas o texto seguindo o seguinte esquema: <a href…>Algum texto → Algum texto.

Após a transformação para pré-processar os textos, foram regularizadas as expressões (aplicando a Regexp (Regular expression), para encontrar uma certa combinação de caracteres dentro de uma mesma *string* de texto. As expressões regulares fornecem uma maneira muito flexível de pesquisar ou reconhecer sequências de texto. Portanto, Regexp permite dividir o texto pela expressão regular fornecida (\w+) que corresponde apenas as palavras exatas sem pontuação, mas foram detectadas letras soltas como “g” ou “c”, então foi aplicada a expressão (\w{2,}) que coincide com palavras que tem mais de 2 caracteres), dividindo o texto por palavras apenas por padrão (foi omitida a pontuação). Dessa forma foi obtido a tokenização. Depois foi executado a marcação ou etiquetado de uma parte da fala ou discurso em tokens por meio do *POS Tagger* calculando marcação ou etiquetado médio *(Averaged Perceptron Tagger)* com o *Perceptron Tagger médio de Matthew Honnibal*. Seguidamente, para a normalização, foi aplicada a Lematização por meio de um modelo pré-treinado para normalizar dados (Lemmagen) para a língua inglesa. Depois, foi a plicado o filtro de *Stopwords* para remover palavras irrelevantes do texto em inglês como por exemplo, remove ‘and’, ‘or’, ‘in’,...,

incluindo números. O rango de n-gramas foi estabelecido no intervalo de 1 e 3 n-gramas para criar n-gramas de tokens (isso significa que incluem unigramas (palavras individuais), bigramas (pares de palavras) e trigramas (três palavras) como tokens).

Assim, os passos a seguir para o pré-processamento dos textos foram: 1) transformar o texto; 2) aplicar tokenização, *tags POS*, normalização, filtragem; e 4) construir n-gramas com base em determinados tokens. Isso é especialmente importante para o WordNet Lemmatizer, pois requer *tags POS* para atingir uma normalização adequada.

Finalmente, foram inspecionadas as principais palavras mostrando nuvem de *tags* ou palavras clave em todos os textos do corpus (Figura 46) e por categorias (Figura 49, 50 e 51). Assim, os resultados foram apresentados nessa fase em forma nuvem de *tags* ou nuvem de palavras mais frequentemente repetidas por classes ou categorias como mostrado na Figura 49, 50 e 51.

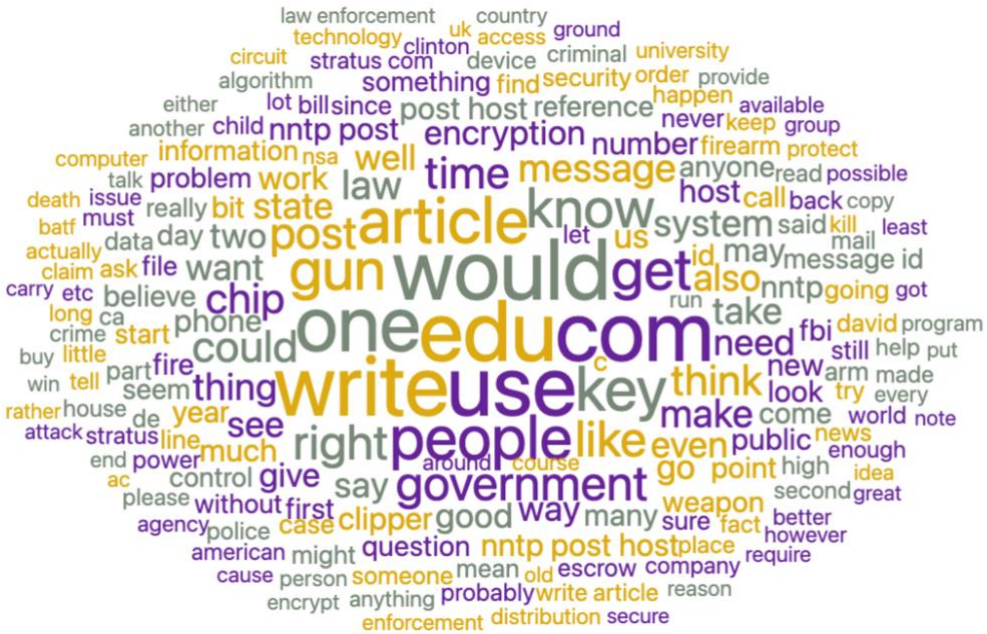
Na Figura 46, pode-se observar a nuvem com as palavras mais frequentes do corpus documental analisado que constava de um total de 3000 documentos. A nuvem de palavras exibe os tokens do corpus e o tamanho dos tokens denota a frequência da palavra no corpus ou uma contagem média do conjunto de palavras. As palavras são listadas por sua frequência (peso). Dessa forma, a Figura 46 apresenta as palavras de maior e menor frequência no corpus documental que forma o *dataset* antes de aplicar os parâmetros de transformação acima mencionados durante o pré-processamento dos textos. Podem-se observar algumas palavras como “ca” que no são significativas para a classificação.

Figura 46 – As palavras mais frequentes do corpus documental analisado antes da transformação no pré-processamento

Fonte: Elaborado pela autora com os dados de pesquisa.

Já, a Figura 47, apresenta as palavras de maior e menor frequência no corpus documental que forma o *dataset* após aplicar os parâmetros de transformação mencionados. Pode-se observar que foram removidas palavras como “ca”. Nas figuras 46 e 47 as palavras que aparecem como maior frequência no corpus tem maior peso e aparecem com um tamanho maior, por exemplo as palavras: *one, write, people, know, would, gun, edu, use, key, article, government, like*, etc. Assim, quanto menor a frequência com a que aparece no *dataset*, menor é o peso e, por tanto, menor o tamanho da palavra, como também se pode observar na Figura 49, 50 e 51.

Figura 47 – As palavras mais frequentes do corpus documental analisado depois da transformação no pré-processamento



Fonte: Elaborado pela autora com os dados de pesquisa

A Figura 48 apresenta as 100 palavras de maior e menor frequência no corpus documental que forma o *dataset*. Dentre as 100 palavras que aparecem como maior frequência no corpus cabe destacar as seguintes: *one, write, people, know, com, edu, would, get gun, use, key, article, government, like*, etc.

Figura 48 – As 100 palavras mais frequentes do corpus documental analisado antes da transformação

Fonte: Elaborado pela autora com os dados de pesquisa.

A Figura 49 apresenta as 30 palavras de maior e menor frequência no corpus documental da categoria Crime (*Crime)*. Dentre as 30 palavras que aparecem como maior frequência na mencionada categoria cabe destacar as seguintes: *encryption, com, edu, would, use, system, write, chip, one, Key, article, clipper*, *know*, *message, article, government,* etc.

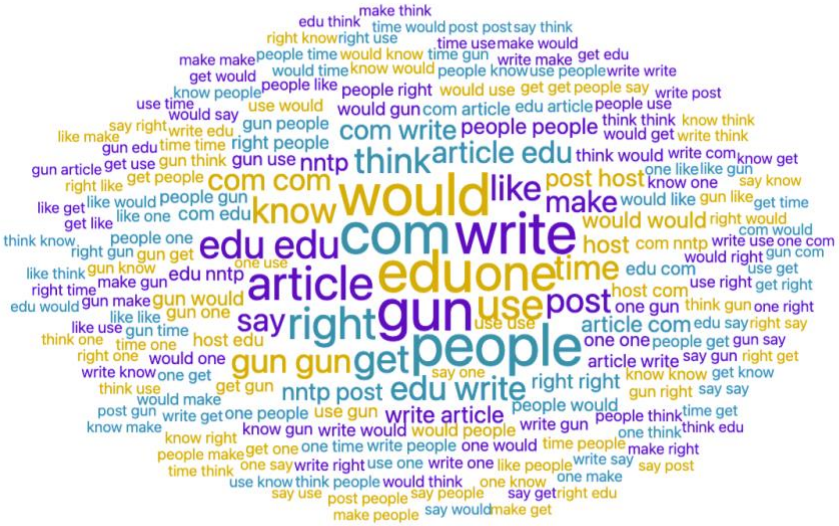
Figura 49 – As 30 palavras mais frequentes na categoria Crime (*Crime)*

Fonte: Elaborado pela autora com os dados de pesquisa.

A Figura 50 apresenta as 30 palavras de maior e menor frequência no corpus documental da categoria Política (*Politics)*. Dentre as 30 palavras que aparecem como maior frequência na

mencionada categoria cabe destacar: *would, com, edu, use, gun, people, one, right, article, know*, etc.

Figura 50 – As 30 palavras mais frequentes na categoria Política (*Politics*)



Fonte: Elaborado pela autora com os dados de pesquisa.

A Figura 51 apresenta as 30 palavras de maior e menor frequência no corpus documental da categoria Ciência (*Science)*. Dentre as 30 palavras que aparecem como maior frequência na mencionada categoria cabe destacar: *would, edu, write, use, one, article, know, work, anyone, need,* etc.

Figura 51 – As 30 palavras mais frequentes na categoria Ciência (*Science*)

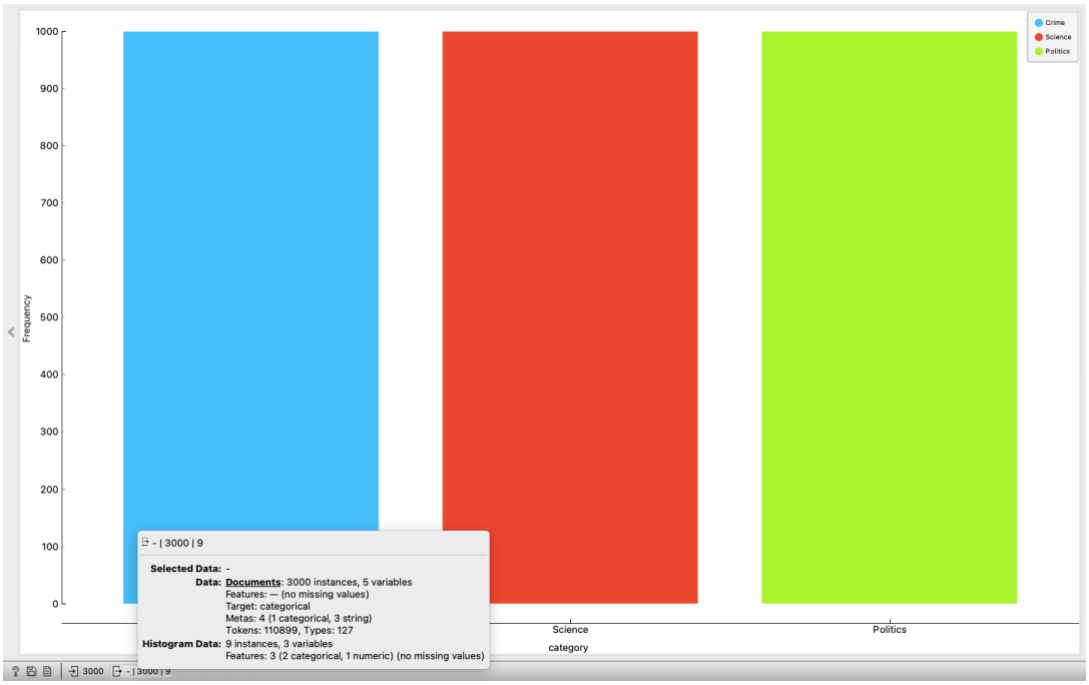
Fonte: Elaborado pela autora com os dados de pesquisa.

O corpus documental analisado constava de um total de 3000 documentos e deu como saída após o pré-processamento acima detalhado um total de 436117 palavras ou tokens e

detectou um total de 23872 tipos de tokens. Foram contabilizadas 250810 instancias de dados (documentos) e 2 variáveis. Mas em todo o corpus, foram achadas 4 variáveis. A distribuição por frequência das categorias após o pré-processamento é apesentado a nos histogramas a seguir.

No Gráfico 1, pode-se observar um histograma com a frequência total na distribuição dos valores para a variável categoria após o pré-processamento do conjunto de dados textuais (*dataset*). Nele se observam os dados documentais de entrada (e-mails) que possuem um total de três categorias, com 1000 instâncias ou entradas (documentos o textos, nesse caso são e-mails) em cada categoria (um total de 3000 instâncias que conformam o corpus documental pré-processado), 110899 palavras (*tokens*) e 127 tipos (*types*) de *tokens*. Apresenta um total de cinco variáveis com marcações (*targets*) categóricas. Mostra um total de 4 metadados (*metas*) composto por três de tipo categóricos e 1 de tipo caracteres ou palavra (*string)*. No presenta nenhuma característica (*features*). Porem, os dados de saída que aparecem no histograma diante desse *dataset* são um total de nove instancias, três variáveis e três características (*features*), das que dois são categóricas e uma numérica. Assim, pode-se observar como o conjunto de dados está balanceado.

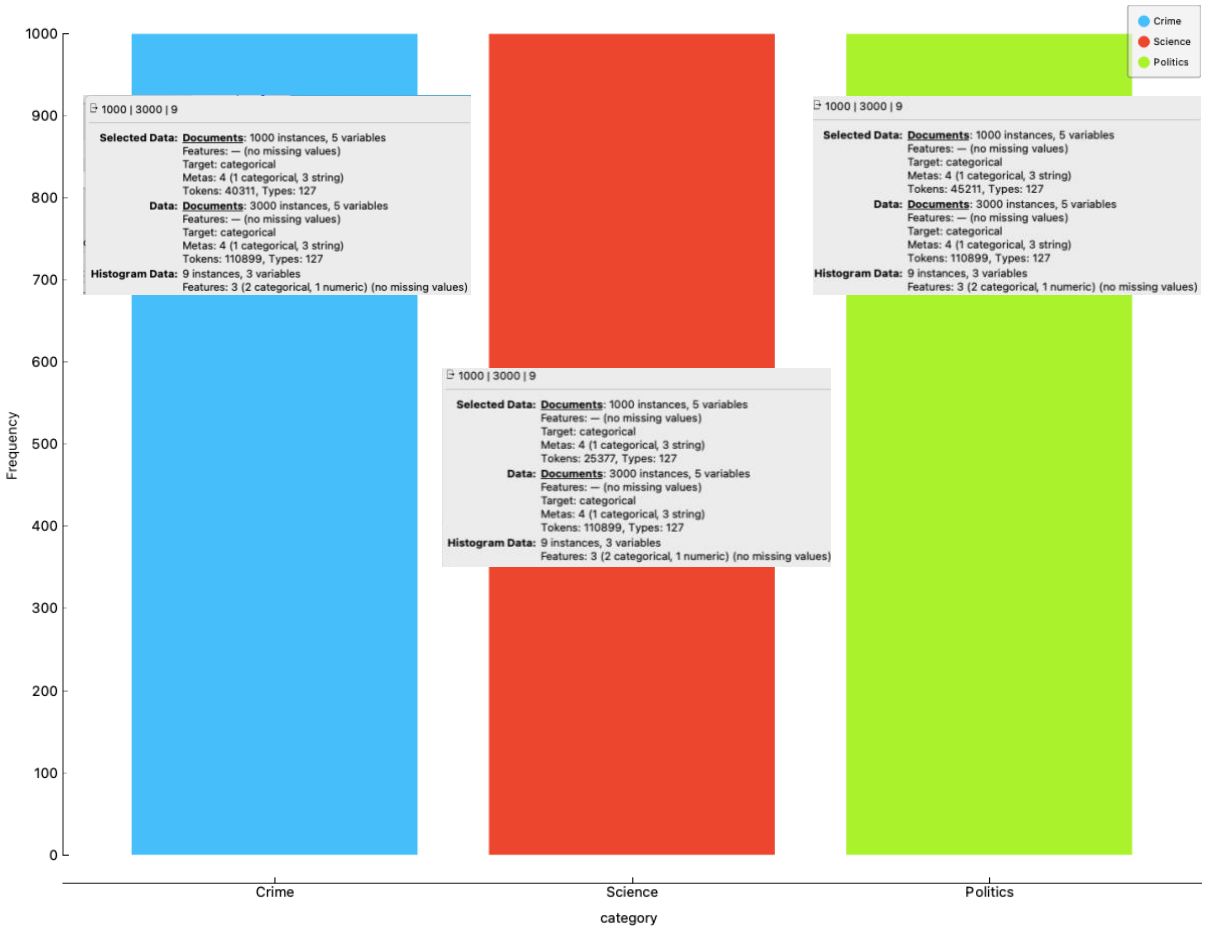
Gráfico 1 – Frequência total na distribuição de valor para a variável categoria do conjunto de dados textuais pré-processado



Fonte: Elaborado pela autora a partir dos dados de pesquisa.

No Gráfico 2, pode-se ver a distribuição da frequência por categorias e as características dos dados de entrada e saída. Tendo em todas em todas as categorias como dados de saídos mostrados no histograma: nove instancias, três variáveis e três caraterísticas: dois categóricas e uma numérica. Pode-se comprovar que para cada categoria forma selecionados 1000 instancias e o sistema detectou cinco variáveis (número de identificação dado a cada documento (id), categoria, nome do documentos, pasta onde foi guardada e conteúdo do e-mail). Em todas as categorias foram encontradas 127 tipos de palavras ou tokens em total. Foram encontrados na a categoria "Crime" (*Crime):* 40311 tokens ou palavras, na categoria "Ciência" (*Science*): 25377 tokens e na categoria "Política" (*Politics*): 45211 tokens.

Gráfico 2 – Frequência na distribuição de valor para cada categoria do corpus documental pré-processado

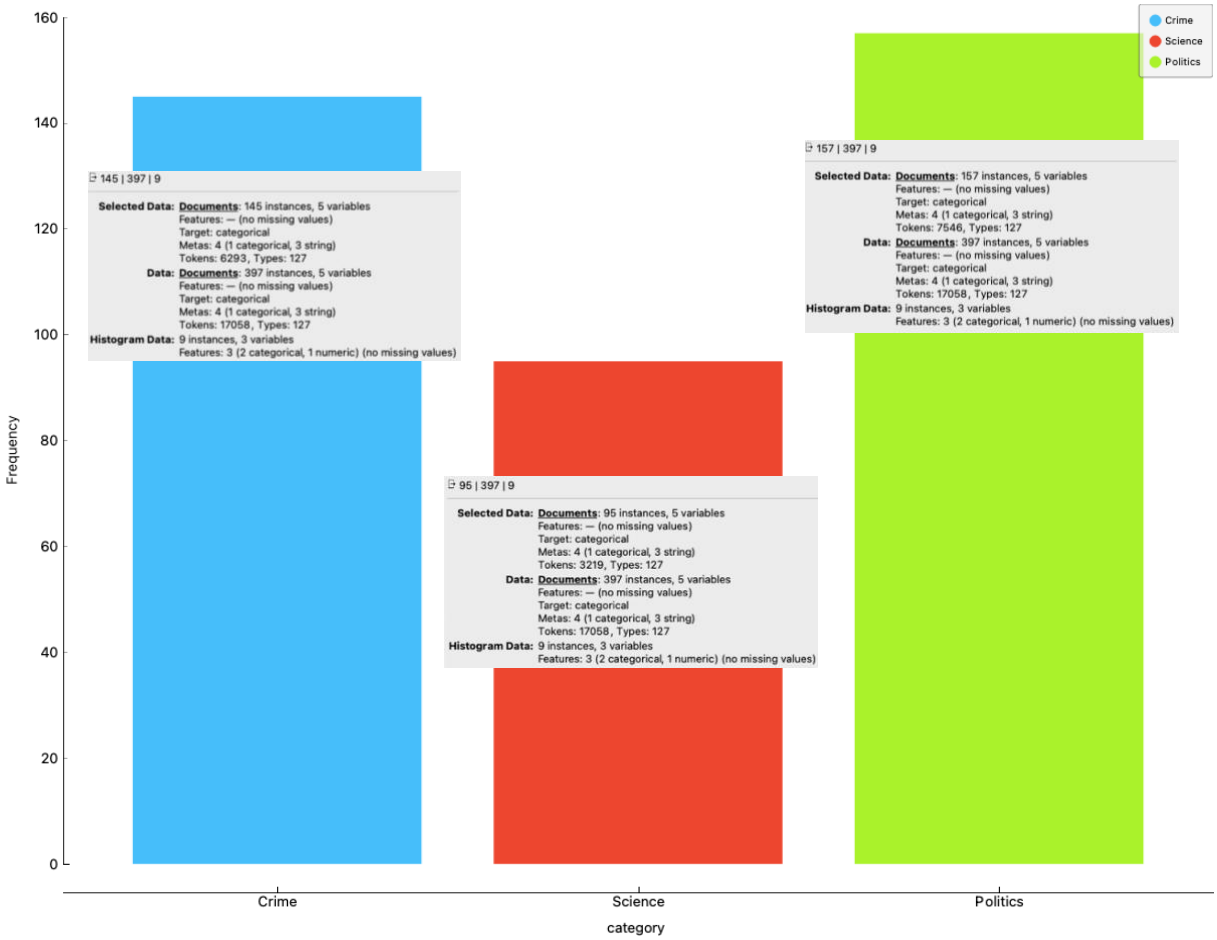


Fonte: Elaborado pela autora a partir dos dados de pesquisa.

No Gráfico 3, pode se observar a distribuição dos tokens por categorias após o pré-processamento do corpus documental e geração da nuvem de palavras. O maior número de palavras é encontrado na categoria "Política" com 7546 tokens, 127 tipos de palavras, 397

instancias o documentos (e-mail) em total, dos quais que foram destacadas 167 instancias em função dos tokens no corpus exibidos na nuvem de palavras, atendendo à frequência (peso) da palavra no corpus ou contagem média do conjunto de palavras, quando os recursos do conjunto de palavras formam parte dos dados de entrada no sistema. Na categoria “Crime”, com uma frequência de 145 instâncias, apresenta 6293 tokens, 127 tipos de palavras, 397 instancias em total. Já a categoria “Ciência” presenta uma frequência de 95 instancias de 397 instancias em total, 3229 tokens ou palavras e 127 tipos de palavras.

Gráfico 3 – Distribuição dos tokens por categorias após o pré-processamento do corpus documental e geração da nuvem de palavras



Fonte: Elaborado pela autora a partir dos dados de pesquisa.

A pós o pré-processamento – no que foi feita 1) a tokenização em n-gramas ou decomposição do texto; 2) a remoção de símbolos e palavras vazias como preposições, artigos, e outras palavras vazias sem relevância semântica; 3) o calculo da frequência dos n-gramas; e 4) a estimativa da estatística TF-IDF para os n-gramas –, foram selecionadas as variáveis mais importantes e os tokens o palavras de maior frequência ou relevância de cada categoria.

Como comentado na literatura mencionada no marco teórico (Jones,1972; Silva, 2021), o TF-IDF é um método confiável para estimar a relevância de um documento para um termo. Um TF-IDF elevado é alcançado, quanto maior a frequência de um termo em uma página, e menor o número de documentos que mencionam esse termo. À medida que o número de documentos que incluem o termo aumenta, o valor do TF-IDF diminui, na medida em que pode chegar a 0, se todos ou quase todos os documentos em uma grande amostra mencioná-lo.

Assim, o TF-IDF mostra com que frequência um termo ou frase aparece dentro de um determinado documento, e o compara ao número de documentos que mencionam esse termo dentro de uma coleção inteira de documentos. Por exemplo, que a palavra "o" ou "a" aparece com muita frequência em um documento específico não significa nada, pois se analisarmos a amostra ou a coleta completa de documentos, veremos que essas palavras são muito comuns. Mas se um documento muitas vezes contém a palavra “sistema”, as chances desse documento ser relevante para a busca por “sistema” crescem, uma vez que o termo é relativamente raro dentro da amostra geral.

Portanto, o cálculo da TF-IDF não é apenas uma contagem das vezes que um termo aparece em um texto. Na primeira parte da fórmula (frequência do termo) uma série de modificadores ou ponderações (*weighting*) são aplicadas aos termos, tais como:

1. Calcular a frequência levando em conta o comprimento do próprio documento
2. Ajustar a frequência do termo de acordo com uma escala logarítmica. A partir de um determinado valor, quase não tem incidência para continuar incluindo esse termo mais.
3. Aplicar uma normalização ou correção, para não recompensar textos longos mais longos. Por exemplo, dividindo a frequência do termo pela frequência do termo que aparece mais vezes no texto.

Os resultados após o pré-processamento da extração de palavras chave por meio do cálculo da média de TF-IDF revelam os termos menos frequentes (IDF) na coleção e a frequência dele (TF) no documento como mostrado na Tabela 5 na que se apresentam os resultados de 30 termos, após calcular a media de repetições de termos do corpus documental. Na Tabela 5, observam-se alguns termos remarcados em azul claro que carecem de valor para a classificação como são: os domínios “.com” e “.edu”; as palavras “could”, “like”, “also”, ou a letra isolada “c”. Assim, foi calculado o TF é o número de vezes que aparece cada termo num documento e o IDF da coleção ou corpus documental foi calculado para ver se os termos são comuns na coleção de documentos.

Tabela 5 – Resultados TF-IDF experimento 1: média de repetições dos 30 termos principais do corpus documental

Palavras/Termos	TF-IDF média de repetições
com_NN	0.00898454151905359
edu_NN	0.008070148037704115
would_MD	0.006731978422307215
people_NNS	0.006371671927573468
government_NN	0.005904934002867694
one_CD	0.005612740408274846
chip_NN	0.005604811951841622
stratus_NN	0.0050207454344231755
encryption_NN	0.004998338098528581
could_MD	0.004819435206523895
key_NN	0.004791712247439311
stratus_NN com_NN	0.004716604250094817
like_IN	0.004626983443150394
article_NN	0.0046269631488685455
gun_NN	0.004605934547132589
get_VB	0.004571051903354902
write_VBZ	0.004506390527795861
message_NN	0.004496268131205678
us_PRP	0.004262304408340026
anyone_NN	0.004245541937117104
phone_NN	0.004245247522678167
time_NN	0.004183693408965189
good_JJ	0.004169708497075746
law_NN	0.004122750868587599
system_NN	0.004086452879345123
post_VBG	0.004062252734034001
also_RB	0.004019186795489684
c_NN	0.004002217573408517
host_NN	0.00399580034353412
even_RB	0.003972265771512895

Fonte: Elaborado pela autora com os dados de pesquisa.

Após o cálculo, é possível modificar os textos com base nos resultados obtidos em cada termo após o cálculo. Pelo que o conhecimento do TF-IDF da informação que permite modificar o texto de tal forma que, neste, os termos e sinônimos usados aparecem um certo número de vezes. Este determinado número de vezes deve estar entre um máximo e um mínimo que foram calculados a partir do número médio de repetições dos conteúdos analisados. Quanto mais textos forem analisados, melhor será a média dos termos a serem utilizados e o número de vezes que devem ser exibidos.

Assim, após a reconfiguração nos parâmetros do pré-processamento dos textos eliminando URLs, desconsiderando *tags* HTML ampliando o rango dos n-gramas de 1 a 3

obtivemos as os resultados do calculo da media de repetições de tokens como mostrado na Tabela 6, na que são apresentados os 30 primeiros termos.

Tabela 6 – Resultados TF-IDF experimento 2: média de repetições dos 30 termos principais do corpus documental

Palavras/Termos	TF-IDF média de repetições
edu_NN	0.00638061
would_MD	0.00576112
people_NNS	0.00523506
government_NN	0.00494133
one_CD	0.00456041
chip_NN	0.00437922
stratus_NN	0.00433901
encryption_NN	0.00399701
stratus_NN com_NN	0.00387842
could_MD	0.00376089
key_NN	0.00374734
article_NN	0.0036985
gun_NN	0.00363612
like_IN	0.00361535
get_VB	0.00360494
message_NN	0.00355724
write_VBZ	0.00355659
anyone_NN	0.00351586
phone_NN	0.00329682
time_NN	0.00328872
good_JJ	0.00326249
law_NN	0.00324319
us_PRP	0.00321557
post_VBG	0.00320417
system_NN	0.00320046
host_NN	0.00318703
also_RB	0.00315337
post_VBG host_NN	0.00314918
even_RB	0.00310641
edu_NN	0.00307786

Os resultados do pré-processamento permitirá selecionar as variáveis mais importantes, tendo em consideração, por exemplo, às palavras de maior frequência e importância.

4.2.1 Pré-processamento de Textos e Representação com *Bag-of-Words*

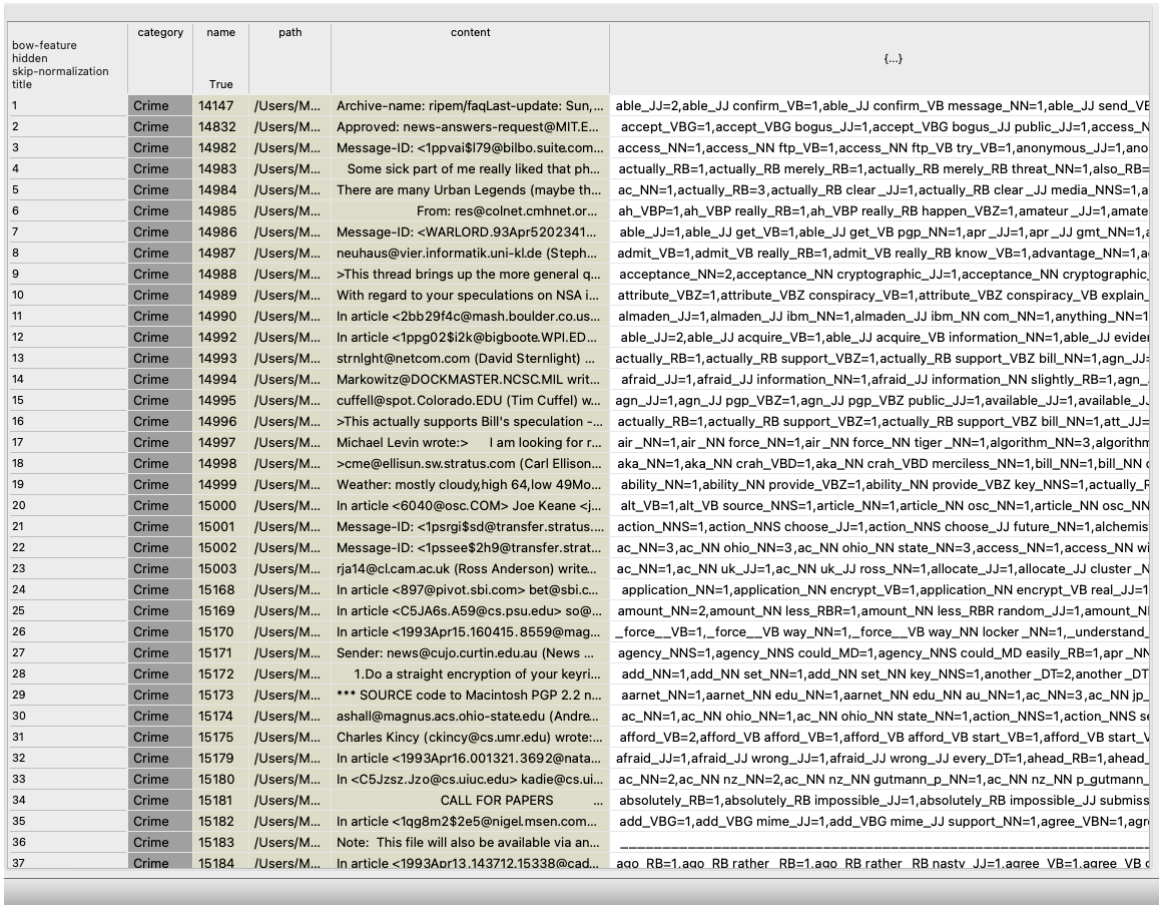
Nessa fase se apresenta a representação do corpus documental pré-processado com *Bag-of-Words (*Saco de palavras*)*. O modelo *Bag-of-Words* cria um corpus com contagens de palavras para cada instância de dados (documento). A contagem pode ser absoluta (contagem

do número de ocorrências de uma palavra em um documento), binária (contém ou não contém uma palavra no documento) ou sublinear (logaritmo do termo frequência). Como parâmetros, nesse experimento foi usada uma contagem absoluta para calcular a frequência do termo e IDF para calcular a frequência inversa do documento.

Foi aplicado o modelo *BoW* com diferentes configurações gerando assim 2 experimentos de classificação:

1) No experimento 1, foi verificado como seria o modelo aplicando uma classificação indutiva supervisionada utilizando a representação BoW, com uma configuração padrão. Para isso, foi carregue o corpus documental e gerado um saco de palavras com uma configuração padrões na que foi feita uma simples contagem de frequências de termos e foram examinados os resultados criando uma tabela com os dados (Figura 52), mostrando as frequências de prazo para cada documento na última coluna. Assim, com uma entrada de 3000 instancias, obtivemos como resultados, 586951 características (*features*), três meta-atributos, e a distinção da classe como categoria.

Figura 52 – Frequências de alguns documentos da categoria “Crime” após aplicar o classificador BoW

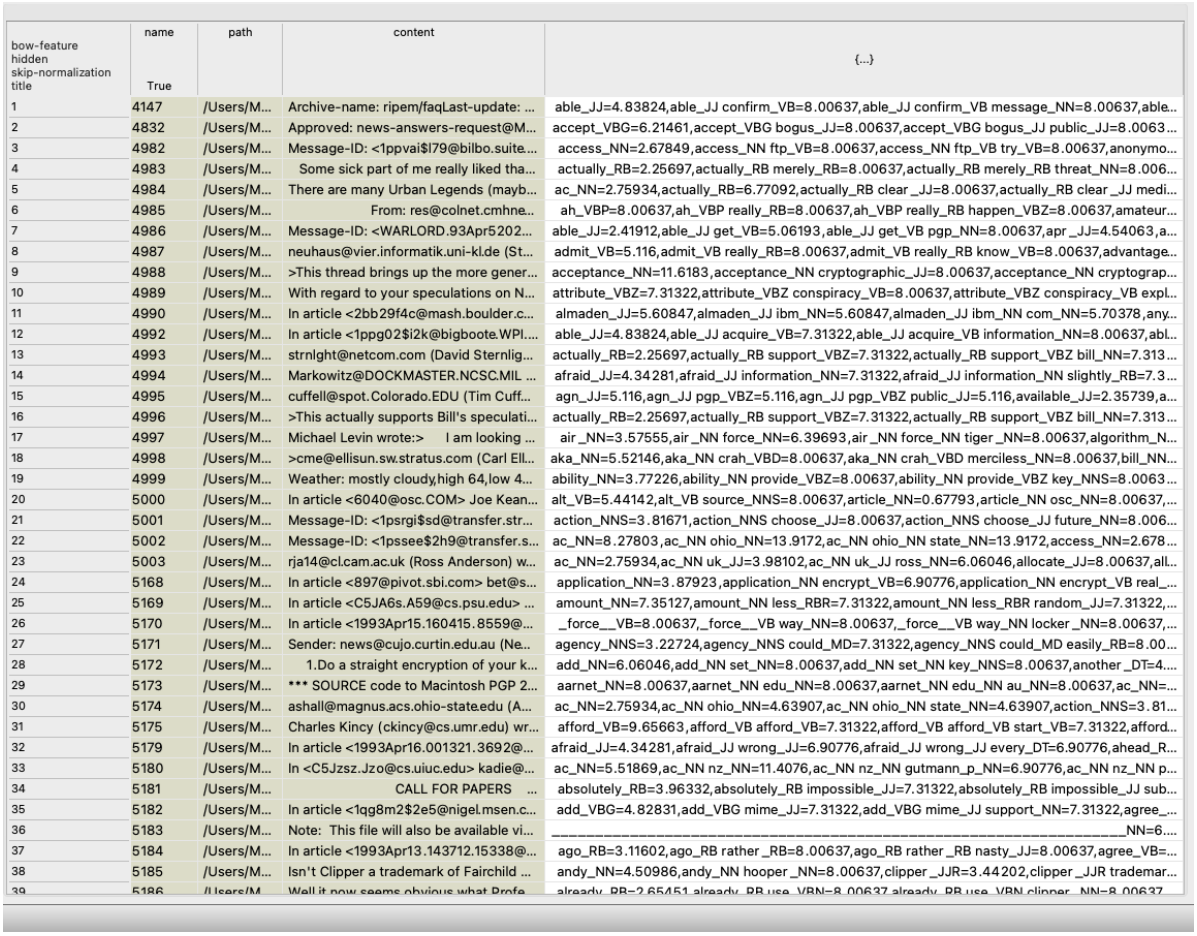


bow-feature hidden skip-normalization title	category	name True	path	content	{...}
1	Crime	14147	/Users/M...	Archive-name: ripem/faqLast-update: Sun,...	able_JJ=2,able_JJ confirm_VB=1,able_JJ confirm_VB message_NN=1,able_JJ send_VB
2	Crime	14832	/Users/M...	Approved: news-answers-request@MIT.E...	accept_VBG=1,accept_VBG bogus_JJ=1,accept_VBG bogus_JJ public_JJ=1,access_N
3	Crime	14982	/Users/M...	Message-ID: <1ppvai$l79@bilbo.suite.com...	access_NN=1,access_NN ftp_VB=1,access_NN ftp_VB try_VB=1,anonymous_JJ=1,ano
4	Crime	14983	/Users/M...	Some sick part of me really liked that ph...	actually_RB=1,actually_RB merely_RB=1,actually_RB merely_RB threat_NN=1,also_RB=
5	Crime	14984	/Users/M...	There are many Urban Legends (maybe th...	ac_NN=1,actually_RB=3,actually_RB clear _JJ=1,actually_RB clear _JJ media_NNS=1,a
6	Crime	14985	/Users/M...	From: res@colnet.cmhnet.or...	ah_VBP=1,ah_VBP really_RB=1,ah_VBP really_RB happen_VBZ=1,amateur _JJ=1,amate
7	Crime	14986	/Users/M...	Message-ID: <WARLORD.93Apr5202341...	able_JJ=1,able_JJ get_VB=1,able_JJ get_VB pgp_NN=1,apr _JJ=1,apr _JJ gmt_NN=1,a
8	Crime	14987	/Users/M...	neuhaus@vier.informatik.uni-kl.de (Steph...	admit_VB=1,admit_VB really_RB=1,admit_VB really_RB know_VB=1,advantage_NN=1,a
9	Crime	14988	/Users/M...	>This thread brings up the more general q...	acceptance_NN=2,acceptance_NN cryptographic_JJ=1,acceptance_NN cryptographic_
10	Crime	14989	/Users/M...	With regard to your speculations on NSA i...	attribute_VBZ=1,attribute_VBZ conspiracy_VB=1,attribute_VBZ conspiracy_VB explain_
11	Crime	14990	/Users/M...	In article <2bb29f4c@mash.boulder.co.us...	almaden_JJ=1,almaden_JJ ibm_NN=1,almaden_JJ ibm_NN com_NN=1,anything_NN=1
12	Crime	14992	/Users/M...	In article <1ppg02$i2k@bigboote.WPI.ED...	able_JJ=2,able_JJ acquire_VB=1,able_JJ acquire_VB information_NN=1,able_JJ evide
13	Crime	14993	/Users/M...	strnlght@netcom.com (David Sternlight) ...	actually_RB=1,actually_RB support_VBZ=1,actually_RB support_VBZ bill_NN=1,agn_JJ=
14	Crime	14994	/Users/M...	Markowitz@DOCKMASTER.NCSC.MIL writ...	afraid_JJ=1,afraid_JJ information_NN=1,afraid_JJ information_NN slightly_RB=1,agn_
15	Crime	14995	/Users/M...	cuffell@spot.Colorado.EDU (Tim Cuffel) w...	agn_JJ=1,agn_JJ pgp_VBZ=1,agn_JJ pgp_VBZ public_JJ=1,available_JJ=1,available_J
16	Crime	14996	/Users/M...	>This actually supports Bill's speculation -...	actually_RB=1,actually_RB support_VBZ=1,actually_RB support_VBZ bill_NN=1,att_JJ=
17	Crime	14997	/Users/M...	Michael Levin wrote:> I am looking for r...	air _NN=1,air _NN force_NN=1,air _NN force_NN tiger _NN=1,algorithm_NN=3,algorithm
18	Crime	14998	/Users/M...	>cme@ellisun.sw.stratus.com (Carl Ellison...	aka_NN=1,aka_NN crah_VBD=1,aka_NN crah_VBD merciless_NN=1,bill_NN=1,bill_NN
19	Crime	14999	/Users/M...	Weather: mostly cloudy,high 64,low 49Mo...	ability_NN=1,ability_NN provide_VBZ=1,ability_NN provide_VBZ key_NNS=1,actually_R
20	Crime	15000	/Users/M...	In article <6040@osc.COM> Joe Keane <j...	alt_VB=1,alt_VB source_NNS=1,article_NN=1,article_NN osc_NN=1,article_NN osc_NN
21	Crime	15001	/Users/M...	Message-ID: <1psrgi$sd@transfer.stratus....	action_NNS=1,action_NNS choose_JJ=1,action_NNS choose_JJ future_NN=1,alchemis
22	Crime	15002	/Users/M...	Message-ID: <1pssee$2h9@transfer.strat...	ac_NN=3,ac_NN ohio_NN=3,ac_NN ohio_NN state_NN=3,access_NN=1,access_NN wi
23	Crime	15003	/Users/M...	rja14@cl.cam.ac.uk (Ross Anderson) write...	ac_NN=1,ac_NN uk_JJ=1,ac_NN uk_JJ ross_NN=1,allocate_JJ=1,allocate_JJ cluster _N
24	Crime	15168	/Users/M...	In article <897@pivot.sbi.com> bet@sbi.c...	application_NN=1,application_NN encrypt_VB=1,application_NN encrypt_VB real_JJ=1
25	Crime	15169	/Users/M...	In article <C5JA6s.A59@cs.psu.edu> so@...	amount_NN=2,amount_NN less_RBR=1,amount_NN less_RBR random_JJ=1,amount_N
26	Crime	15170	/Users/M...	In article <1993Apr15.160415.8559@mag...	_force__VB=1,_force__VB way_NN=1,_force__VB way_NN locker _NN=1,_understand_
27	Crime	15171	/Users/M...	Sender: news@cujo.curtin.edu.au (News ...	agency_NNS=1,agency_NNS could_MD=1,agency_NNS could_MD easily_RB=1,apr _NN
28	Crime	15172	/Users/M...	1.Do a straight encryption of your keyri...	add_NN=1,add_NN set_NN=1,add_NN set_NN key_NNS=1,another _DT=2,another _DT
29	Crime	15173	/Users/M...	*** SOURCE code to Macintosh PGP 2.2 n...	aarnet_NN=1,aarnet_NN edu_NN=1,aarnet_NN edu_NN au_NN=1,ac_NN=3,ac_NN jp_
30	Crime	15174	/Users/M...	ashall@magnus.acs.ohio-state.edu (Andre...	ac_NN=1,ac_NN ohio_NN=1,ac_NN ohio_NN state_NN=1,action_NNS=1,action_NNS s
31	Crime	15175	/Users/M...	Charles Kincy (ckincy@cs.umr.edu) wrote:...	afford_VB=2,afford_VB afford_VB=1,afford_VB afford_VB start_VB=1,afford_VB start_V
32	Crime	15179	/Users/M...	In article <1993Apr16.001321.3692@nata...	afraid_JJ=1,afraid_JJ wrong_JJ=1,afraid_JJ wrong_JJ every_DT=1,ahead_RB=1,ahead_
33	Crime	15180	/Users/M...	In <C5Jzsz.Jzo@cs.uiuc.edu> kadie@cs.ui...	ac_NN=2,ac_NN nz_NN=2,ac_NN nz_NN gutmann_p_NN=1,ac_NN nz_NN p_gutmann_
34	Crime	15181	/Users/M...	CALL FOR PAPERS ...	absolutely_RB=1,absolutely_RB impossible_JJ=1,absolutely_RB impossible_JJ submiss
35	Crime	15182	/Users/M...	In article <1qg8m2$2e5@nigel.msen.com...	add_VBG=1,add_VBG mime_JJ=1,add_VBG mime_JJ support_NN=1,agree_VBN=1,agr
36	Crime	15183	/Users/M...	Note: This file will also be available via an...	_____
37	Crime	15184	/Users/M...	In article <1993Apr13.143712.15338@cad...	ago_RB=1,ago_RB rather_RB=1,ago_RB rather_RB nasty_JJ=1,agree_VB=1,agree_VB

Fonte: Elaborado pela autora com os dados de pesquisa e usando Orange 3.

2) No experimento 2, foi verificado como seria o modelo aplicando uma classificação indutiva supervisionada utilizando a representação BOW, com uma configuração mais elaborada. Para isso, foi carregue o corpus documental e gerado um saco de palavras com uma configuração padrões na que foi feita uma simples contagem de frequências de termos, o cálculo de IDF e se verificaram os resultados criando uma tabela com os dados (Figura 53), mostrando as frequências de prazo para cada documento na última coluna. Assim, com uma entrada de 3000 instancias, foi obtido como resultados, 586951 características (f*eatures*), três meta-atributos, e a distinção da classe como categoria.

Figura 53 – Frequências de alguns documentos da categoria “Crime” após aplicar o classificador BoW



bow-feature hidden skip-normalization title	name True	path	content	{...}
1	4147	/Users/M...	Archive-name: ripem/faqLast-update: ...	able_JJ=4.83824,able_JJ confirm_VB=8.00637,able_JJ confirm_VB message_NN=8.00637,able...
2	4832	/Users/M...	Approved: news-answers-request@M...	accept_VBG=6.21461,accept_VBG bogus_JJ=8.00637,accept_VBG bogus_JJ public_JJ=8.0063...
3	4982	/Users/M...	Message-ID: <1ppvai$l79@bilbo.suite....	access_NN=2.67849,access_NN ftp_VB=8.00637,access_NN ftp_VB try_VB=8.00637,anonymo...
4	4983	/Users/M...	Some sick part of me really liked tha...	actually_RB=2.25697,actually_RB merely_RB=8.00637,actually_RB merely_RB threat_NN=8.006...
5	4984	/Users/M...	There are many Urban Legends (mayb...	ac_NN=2.75934,actually_RB=6.77092,actually_RB clear _JJ=8.00637,actually_RB clear _JJ medi...
6	4985	/Users/M...	From: res@colnet.cmhne...	ah_VBP=8.00637,ah_VBP really_RB=8.00637,ah_VBP really_RB happen_VBZ=8.00637,amateur...
7	4986	/Users/M...	Message-ID: <WARLORD.93Apr5202...	able_JJ=2.41912,able_JJ get_VB=5.06193,able_JJ get_VB pgp_NN=8.00637,apr _JJ=4.54063,a...
8	4987	/Users/M...	neuhaus@vier.informatik.uni-kl.de (St...	admit_VB=5.116,admit_VB really_RB=8.00637,admit_VB really_RB know_VB=8.00637,advantage...
9	4988	/Users/M...	>This thread brings up the more gener...	acceptance_NN=11.6183,acceptance_NN cryptographic_JJ=8.00637,acceptance_NN cryptograp...
10	4989	/Users/M...	With regard to your speculations on N...	attribute_VBZ=7.31322,attribute_VBZ conspiracy_VB=8.00637,attribute_VBZ conspiracy_VB expl...
11	4990	/Users/M...	In article <2bb29f4c@mash.boulder.c...	almaden_JJ=5.60847,almaden_JJ ibm_NN=5.60847,almaden_JJ ibm_NN com_NN=5.70378,any...
12	4992	/Users/M...	In article <1ppg02$i2k@bigboote.WPI....	able_JJ=4.83824,able_JJ acquire_VB=7.31322,able_JJ acquire_VB information_NN=8.00637,abl...
13	4993	/Users/M...	strnlght@netcom.com (David Sternlig...	actually_RB=2.25697,actually_RB support_VBZ=7.31322,actually_RB support_VBZ bill_NN=7.313...
14	4994	/Users/M...	Markowitz@DOCKMASTER.NCSC.MIL ...	afraid_JJ=4.34281,afraid_JJ information_NN=7.31322,afraid_JJ information_NN slightly_RB=7.3...
15	4995	/Users/M...	cuffell@spot.Colorado.EDU (Tim Cuff...	agn_JJ=5.116,agn_JJ pgp_VBZ=5.116,agn_JJ pgp_VBZ public_JJ=5.116,available_JJ=2.35739,a...
16	4996	/Users/M...	>This actually supports Bill's speculati...	actually_RB=2.25697,actually_RB support_VBZ=7.31322,actually_RB support_VBZ bill_NN=7.313...
17	4997	/Users/M...	Michael Levin wrote:> I am looking ...	air _NN=3.57555,air _NN force_NN=6.39693,air _NN force_NN tiger _NN=8.00637,algorithm_N...
18	4998	/Users/M...	>cme@ellisun.sw.stratus.com (Carl Ell...	aka_NN=5.52146,aka_NN crah_VBD=8.00637,aka_NN crah_VBD merciless_NN=8.00637,bill_NN...
19	4999	/Users/M...	Weather: mostly cloudy,high 64,low 4...	ability_NN=3.77226,ability_NN provide_VBZ=8.00637,ability_NN provide_VBZ key_NNS=8.0063...
20	5000	/Users/M...	In article <6040@osc.COM> Joe Kean...	alt_VB=5.44142,alt_VB source_NNS=8.00637,article_NN=0.67793,article_NN osc_NN=8.00637,...
21	5001	/Users/M...	Message-ID: <1psrgi$sd@transfer.str...	action_NNS=3.81671,action_NNS choose_JJ=8.00637,action_NNS choose_JJ future_NN=8.006...
22	5002	/Users/M...	Message-ID: <1pssee$2h9@transfer.s...	ac_NN=8.27803,ac_NN ohio_NN=13.9172,ac_NN ohio_NN state_NN=13.9172,access_NN=2.678...
23	5003	/Users/M...	rja14@cl.cam.ac.uk (Ross Anderson) w...	ac_NN=2.75934,ac_NN uk_JJ=3.98102,ac_NN uk_JJ ross_NN=6.06046,allocate_JJ=8.00637,all...
24	5168	/Users/M...	In article <897@pivot.sbi.com> bet@s...	application_NN=3.87923,application_NN encrypt_VB=6.90776,application_NN encrypt_VB real_...
25	5169	/Users/M...	In article <C5JA6s.A59@cs.psu.edu> ...	amount_NN=7.35127,amount_NN less_RBR=7.31322,amount_NN less_RBR random_JJ=7.31322,...
26	5170	/Users/M...	In article <1993Apr15.160415.8559@...	_force__VB=8.00637,_force__VB way_NN=8.00637,_force__VB way_NN locker _NN=8.00637,...
27	5171	/Users/M...	Sender: news@cujo.curtin.edu.au (Ne...	agency_NNS=3.22724,agency_NNS could_MD=7.31322,agency_NNS could_MD easily_RB=8.00...
28	5172	/Users/M...	1.Do a straight encryption of your k...	add_NN=6.06046,add_NN set_NN=8.00637,add_NN set_NN key_NNS=8.00637,another _DT=4....
29	5173	/Users/M...	*** SOURCE code to Macintosh PGP 2...	aarnet_NN=8.00637,aarnet_NN edu_NN=8.00637,aarnet_NN edu_NN au_NN=8.00637,ac_NN=...
30	5174	/Users/M...	ashall@magnus.acs.ohio-state.edu (A...	ac_NN=2.75934,ac_NN ohio_NN=4.63907,ac_NN ohio_NN state_NN=4.63907,action_NNS=3.81...
31	5175	/Users/M...	Charles Kincy (ckincy@cs.umr.edu) wr...	afford_VB=9.65663,afford_VB afford_VB=7.31322,afford_VB afford_VB start_VB=7.31322,afford...
32	5179	/Users/M...	In article <1993Apr16.001321.3692@...	afraid_JJ=4.34281,afraid_JJ wrong_JJ=6.90776,afraid_JJ wrong_JJ every_DT=6.90776,ahead_R...
33	5180	/Users/M...	In <C5Jzsz.Jzo@cs.uiuc.edu> kadie@...	ac_NN=5.51869,ac_NN nz_NN=11.4076,ac_NN nz_NN gutmann_p_NN=6.90776,ac_NN nz_NN p...
34	5181	/Users/M...	CALL FOR PAPERS ...	absolutely_RB=3.96332,absolutely_RB impossible_JJ=7.31322,absolutely_RB impossible_JJ sub...
35	5182	/Users/M...	In article <1qg8m2$2e5@nigel.msen.c...	add_VBG=4.82831,add_VBG mime_JJ=7.31322,add_VBG mime_JJ support_NN=7.31322,agree_...
36	5183	/Users/M...	Note: This file will also be available vi...	_____NN=6....
37	5184	/Users/M...	In article <1993Apr13.143712.15338@...	ago_RB=3.11602,ago_RB rather _RB=8.00637,ago_RB rather _RB nasty_JJ=8.00637,agree_VB=...
38	5185	/Users/M...	Isn't Clipper a trademark of Fairchild ...	andy_NN=4.50986,andy_NN hooper _NN=8.00637,clipper _JJR=3.44202,clipper _JJR trademar...
39	5186	/Users/M...	Well it now seems obvious what Profe...	already_RB=2.65451,already_RB use_VBN=8.00637,already_RB use_VBN clipper _NN=8.00637...

Fonte: Elaborado pela autora com os dados de pesquisa e usando Orange 3.

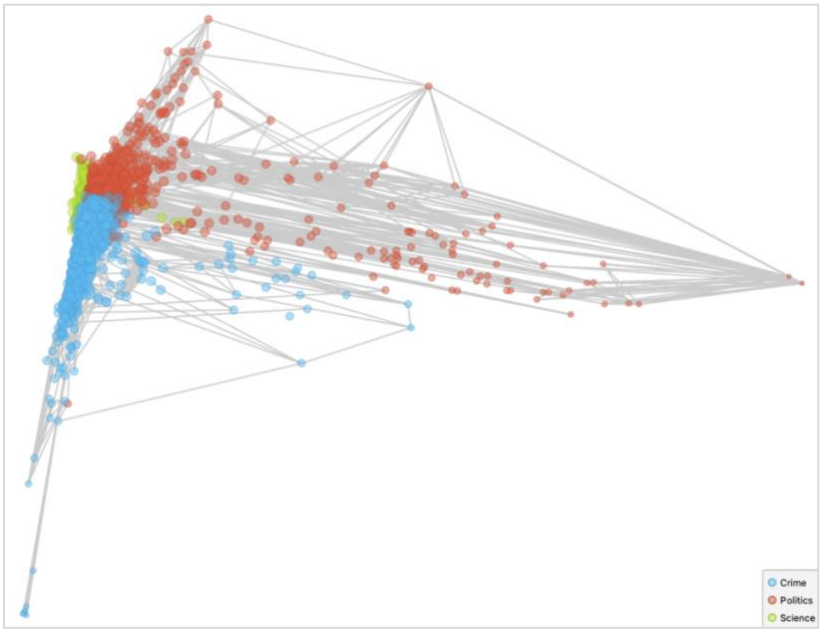
3) No experimento 3, foi verificado como seria o modelo aplicando uma classificação indutiva supervisionada utilizando a representação BoW, com uma

configuração mais elaborada. Para isso, foi carregado o corpus documental e gerado um saco de palavras com uma configuração padrões na que foi feita uma simples contagem de frequências de termos, o cálculo de IDF. Depois foi calculada a distancia de cosseno para conhecer a similaridade entre os documentos por categorias e os resultados formas verificados criando uma escala multidimensional (*Multidimensional scaling* (MDS)) para projetar os itens em um plano bidimensional com as distancias entre os pontos (Gráfico 4 e 5). Depois foi gerada uma tabela com os dados (Figura 53), mostrando as frequências de prazo para cada documento na última coluna. Assim, a entrada de 3000 instancias, 586409 características (f*eatures*) numéricas, seis meta-atributos (um categóricos, dois numérico e 3 *string*), 31 tokens, 29799 tipos de dados e a distinção da classe como categoria. Foi Tokens: 430534. Os resultados obtidos foram: 3000 instancias; 586409 características numéricas, seis meta-atributos (um categóricos, dois numérico e 3 *string*), 430534 tokens, 29799 tipos de dados, a distinção da classe como categoria.

Segundo Wickelmaier (2003), MSD é uma técnica que encontra uma projeção de pontos de baixa dimensão (nesse caso uma bidimensional), onde tenta encaixar distâncias entre os pontos da melhor forma possível. O ajuste perfeito é tipicamente impossível de obter, uma vez que os dados são de alta dimensão ou as distâncias não são Euclidianas.

O Gráfico 4 apresenta os primeiros resultados após aplicar a técnica MSD. Para obter esse gráfico, foi preciso carregar o conjunto de dados textuais e gerar uma matriz de distância de cosseno.

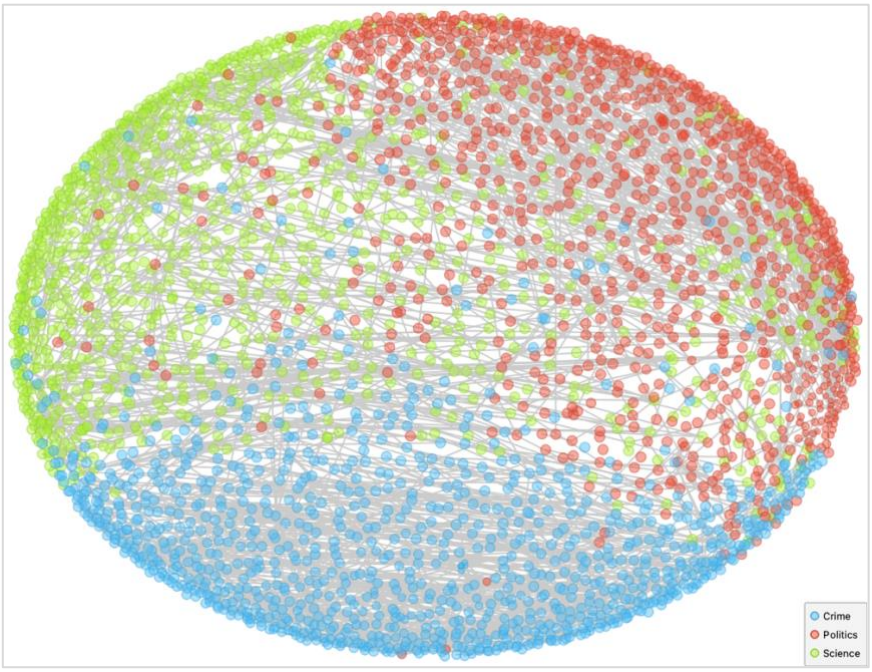
Gráfico 4 – Plano bidimensional por categorias do corpus documental al inicio da geração da matriz de distância de cosseno



Fonte: Elaborado pela autora com os dados de pesquisa.

Já no Gráfico 5 e se observam o resultado final após gerar o saco de palavras (BoW) onde se calculou o IDF, calcular a distância de cosseno e aplicar a técnica MDS. No Gráfico 4 e 5, pode-se visualizar as distâncias entre linhas e como guarda relação temática cada ponto. Observam-se claramente três regiões de cores (azul, vermelho e verde) referentes às diferentes categorias e mesmo assim, tem ocorrências classificadas em categorias diferentes que guardam relação com uma categoria determinada, assim, por exemplo aparecem pontos vermelhos e azul na região onde predominam os pontos verdes e assim acontece com as outras categorias.

Gráfico 5 – Plano bidimensional por categorias com a matriz de distancia de cosseno do corpus documental



Fonte: Elaborado pela autora com os dados de pesquisa.

Os gráficos acima foram desenhados usando o seguinte esquema simples: 1) entrada de dados textuais ou corpus documental; 2) pré-processamento dos dados para obter os tokens; 3) aplicação da técnica de classificação BoW calculando o IDF; 4) cálculo da distância de cosseno por ser a mais apropriada para obter resultados mais precisos com dados textuais; 5) aplicação da técnica MDS e 6) geração de uma tabela na que se visualizam as coordenadas com os dados MDS *x* e *y* (Figura 53).

Podem ser calculadas as distâncias entre linhas ou colunas em um conjunto de dados. No cálculo da Distância Euclidiana, os dados foram normalizados para garantir o tratamento igualitário das características individuais. A normalização é sempre feita em termos de coluna. Dados esparsos só podem ser usados com métrica Euclidiana, Manhattan e Cosseno. A matriz de distância resultante foi alimentada com a Matriz de Distância para visualizar as distâncias e com a aplicação da técnica MDS para mapear as instâncias de dados usando a matriz de distância.

Assim, após escolher como métrica de Distância de Cosseno para conhecer a similaridades entre vetores. Como parâmetros, foram medidas as distâncias entre linhas e o cálculo da distância de cosseno foi 1-similaridade. O a Distância de

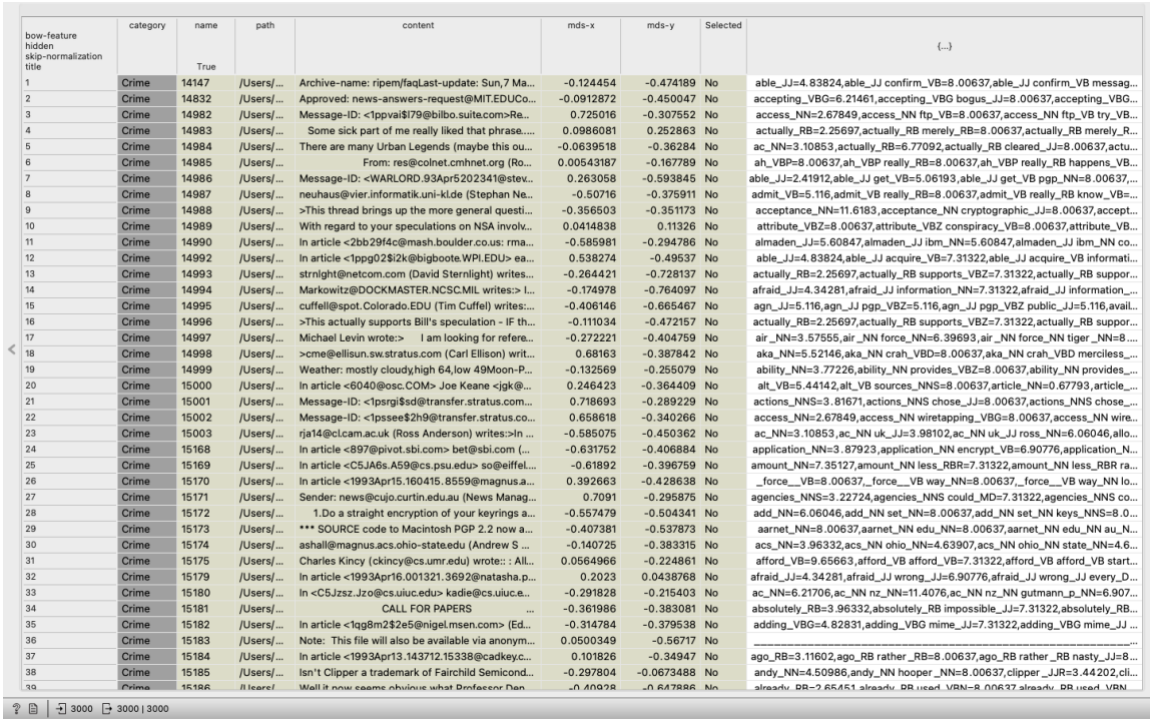
Cosseno permite calcular o cosseno do ângulo entre dois vetores de um espaço interno do produto.

Nesse ponto, cabe destacar que foi considerado como mais apropriado o cálculo da distância de Cosseno que o cálculo da distância Euclidiana por se tratar de um *dataset* textual. Assim, o calculo da similaridade de cosseno é mais adequado para as características do conjunto de dados utilizado nesse trabalho.

Além disso, foi calculada a Distância Euclidiana com normalização das características. A normalização é sempre feita em termos de coluna. Os valores são zero centrados e dimensionados. Em caso de falta de valores, automaticamente se imputa o valor médio da linha ou da coluna. Isso se pode aplicar tanto para dados numéricos quanto categóricos. Em caso de dados categóricos, a distância é 0 se os dois valores forem os mesmos (por exemplo: “verde” e “verde”) e 1 se não forem (“verde” e “azul”). A métrica de distância Euclidiana calcula numa “linha reta” a distância entre dois pontos.

Calcularam-se as distâncias entre instâncias de dados (linhas), gerou-se uma matriz de distâncias, aplicou-se a técnica MDS para observar as semelhanças de dados e plotou-se um gráfico de dispersão para facilitar a visualização e análise dos resultados obtidos nessa fase de pré-processamento. Dessa forma, encontramos grupos de instâncias de dados. Para futuros estúdios, podem-se calcular as a distância entre colunas e descobrir como as características do *dataset* são semelhantes (Figura 54).

Figura 54 – Frequências de alguns documentos da categoria “Crime” após aplicar o classificador BoW, calcular a distância de cosseno e aplicar a técnica MDS



bow-feature hidden skip-normalization title	category	name True	path	content	mds-x	mds-y	Selected	{...}
1	Crime	14147	/Users/...	Archive-name: ripem/faqLast-update: Sun,7 Ma...	-0.124454	-0.474189	No	able_JJ=4.83824,able_JJ confirm_VB=8.00637,able_JJ confirm_VB messag...
2	Crime	14832	/Users/...	Approved: news-answers-request@MIT.EDUCo...	-0.0912872	-0.450047	No	accepting_VBG=6.21461,accepting_VBG bogus_JJ=8.00637,accepting_VBG...
3	Crime	14982	/Users/...	Message-ID: <1ppvai$i79@bilbo.suite.com>Re...	0.725016	-0.307552	No	access_NN=2.67849,access_NN ftp_VB=8.00637,access_NN ftp_VB try_VB...
4	Crime	14983	/Users/...	Some sick part of me really liked that phrase....	0.0986081	0.252863	No	actually_RB=2.25697,actually_RB merely_RB=8.00637,actually_RB merely_R...
5	Crime	14984	/Users/...	There are many Urban Legends (maybe this ou...	-0.0639518	-0.36284	No	ac_NN=3.10853,actually_RB=6.77092,actually_RB cleared_JJ=8.00637,actu...
6	Crime	14985	/Users/...	From: res@colnet.cmhnet.org (Ro...	0.00543187	-0.167789	No	ah_VBP=8.00637,ah_VBP really_RB=8.00637,ah_VBP really_RB happens_VB...
7	Crime	14986	/Users/...	Message-ID: <WARLORD.93Apr5202341@stev...	0.263058	-0.593845	No	able_JJ=2.41912,able_JJ get_VB=5.06193,able_JJ get_VB pgp_NN=8.00637,...
8	Crime	14987	/Users/...	neuhaus@vier.informatik.uni-kl.de (Stephan Ne...	-0.50716	-0.375911	No	admit_VB=5.116,admit_VB really_RB=8.00637,admit_VB really_RB know_VB=...
9	Crime	14988	/Users/...	>This thread brings up the more general questi...	-0.356503	-0.351173	No	acceptance_NN=11.6183,acceptance_NN cryptographic_JJ=8.00637,accept...
10	Crime	14989	/Users/...	With regard to your speculations on NSA involv...	0.0414838	0.11326	No	attribute_VBZ=8.00637,attribute_VBZ conspiracy_VB=8.00637,attribute_VB...
11	Crime	14990	/Users/...	In article <2bb29f4c@mash.boulder.co.us: rma...	-0.585981	-0.294786	No	almaden_JJ=5.60847,almaden_JJ ibm_NN=5.60847,almaden_JJ ibm_NN co...
12	Crime	14992	/Users/...	In article <1ppg02$i2k@bigboote.WPI.EDU> ea...	0.538274	-0.49537	No	able_JJ=4.83824,able_JJ acquire_VB=7.31322,able_JJ acquire_VB informati...
13	Crime	14993	/Users/...	strnlght@netcom.com (David Sternlight) writes...	-0.264421	-0.728137	No	actually_RB=2.25697,actually_RB supports_VBZ=7.31322,actually_RB suppor...
14	Crime	14994	/Users/...	Markowitz@DOCKMASTER.NCSC.MIL writes:> I...	-0.174978	-0.764097	No	afraid_JJ=4.34281,afraid_JJ information_NN=7.31322,afraid_JJ information_...
15	Crime	14995	/Users/...	cuffell@spot.Colorado.EDU (Tim Cuffel) writes:...	-0.406146	-0.665467	No	agn_JJ=5.116,agn_JJ pgp_VBZ=5.116,agn_JJ pgp_VBZ public_JJ=5.116,avail...
16	Crime	14996	/Users/...	>This actually supports Bill's speculation - IF th...	-0.111034	-0.472157	No	actually_RB=2.25697,actually_RB supports_VBZ=7.31322,actually_RB suppor...
17	Crime	14997	/Users/...	Michael Levin wrote:> I am looking for refere...	-0.272221	-0.404759	No	air_NN=3.57555,air_NN force_NN=6.39693,air_NN force_NN tiger_NN=8....
18	Crime	14998	/Users/...	>cme@ellisun.sw.stratus.com (Carl Ellison) writ...	0.68163	-0.387842	No	aka_NN=5.52146,aka_NN crah_VBD=8.00637,aka_NN crah_VBD merciless_...
19	Crime	14999	/Users/...	Weather: mostly cloudy,high 64,low 49Moon-P...	-0.132569	-0.255079	No	ability_NN=3.77226,ability_NN provides_VBZ=8.00637,ability_NN provides_...
20	Crime	15000	/Users/...	In article <6040@osc.COM> Joe Keane <jgk@...	0.246423	-0.364409	No	alt_VB=5.44142,alt_VB sources_NNS=8.00637,article_NN=0.67793,article_...
21	Crime	15001	/Users/...	Message-ID: <1psrgi$sd@transfer.stratus.com...	0.718693	-0.289229	No	actions_NNS=3.81671,actions_NNS chose_JJ=8.00637,actions_NNS chose_...
22	Crime	15002	/Users/...	Message-ID: <1pssee$2h9@transfer.stratus.co...	0.658618	-0.340266	No	access_NN=2.67849,access_NN wiretapping_VBG=8.00637,access_NN wire...
23	Crime	15003	/Users/...	rja14@cl.cam.ac.uk (Ross Anderson) writes:>In ...	-0.585075	-0.450362	No	ac_NN=3.10853,ac_NN uk_JJ=3.98102,ac_NN uk_JJ ross_NN=6.06046,allo...
24	Crime	15168	/Users/...	In article <897@pivot.sbi.com> bet@sbi.com (...	-0.631752	-0.406884	No	application_NN=3.87923,application_NN encrypt_VB=6.90776,application_N...
25	Crime	15169	/Users/...	In article <C5JA6s.A59@cs.psu.edu> so@eiffel....	-0.61892	-0.396759	No	amount_NN=7.35127,amount_NN less_RBR=7.31322,amount_NN less_RBR ra...
26	Crime	15170	/Users/...	In article <1993Apr15.160415.8559@magnus.a...	0.392663	-0.428638	No	_force__VB=8.00637,_force__VB way_NN=8.00637,_force__VB way_NN lo...
27	Crime	15171	/Users/...	Sender: news@cujo.curtin.edu.au (News Manag...	0.7091	-0.295875	No	agencies_NNS=3.22724,agencies_NNS could_MD=7.31322,agencies_NNS co...
28	Crime	15172	/Users/...	1.Do a straight encryption of your keyrings a...	-0.557479	-0.504341	No	add_NN=6.06046,add_NN set_NN=8.00637,add_NN set_NN keys_NNS=8.0...
29	Crime	15173	/Users/...	*** SOURCE code to Macintosh PGP 2.2 now a...	-0.407381	-0.537873	No	aarnet_NN=8.00637,aarnet_NN edu_NN=8.00637,aarnet_NN edu_NN au_N...
30	Crime	15174	/Users/...	ashall@magnus.acs.ohio-state.edu (Andrew S ...	-0.140725	-0.383315	No	acs_NN=3.96332,acs_NN ohio_NN=4.63907,acs_NN ohio_NN state_NN=4.6...
31	Crime	15175	/Users/...	Charles Kincy (ckincy@cs.umr.edu) wrote: : All...	0.0564966	-0.224861	No	afford_VB=9.65663,afford_VB afford_VB=7.31322,afford_VB afford_VB start...
32	Crime	15179	/Users/...	In article <1993Apr16.001321.3692@natasha.p...	0.2023	0.0438768	No	afraid_JJ=4.34281,afraid_JJ wrong_JJ=6.90776,afraid_JJ wrong_JJ every_D...
33	Crime	15180	/Users/...	In <C5Jzsz.Jzo@cs.uiuc.edu> kadie@cs.uiuc.e...	-0.291828	-0.215403	No	ac_NN=6.21706,ac_NN nz_NN=11.4076,ac_NN nz_NN gutmann_p_NN=6.907...
34	Crime	15181	/Users/...	CALL FOR PAPERS ...	-0.361986	-0.383081	No	absolutely_RB=3.96332,absolutely_RB impossible_JJ=7.31322,absolutely_RB...
35	Crime	15182	/Users/...	In article <1qg8m2$2e5@nigel.msen.com> (Ed...	-0.314784	-0.379538	No	adding_VBG=4.82831,adding_VBG mime_JJ=7.31322,adding_VBG mime_JJ ...
36	Crime	15183	/Users/...	Note: This file will also be available via anonym...	0.0500349	-0.56717	No	_____...
37	Crime	15184	/Users/...	In article <1993Apr13.143712.15338@cadkey.c...	0.101826	-0.34947	No	ago_RB=3.11602,ago_RB rather_RB=8.00637,ago_RB rather_RB nasty_JJ=8...
38	Crime	15185	/Users/...	Isn't Clipper a trademark of Fairchild Semicond...	-0.297804	-0.0673488	No	andy_NN=4.50986,andy_NN hooper_NN=8.00637,clipper_JJR=3.44202,cli...
39	Crime	15186	/Users/...	Well it now seems obvious what Professor Den...	-0.40928	-0.647886	No	already_RB=2.65451,already_RB used_VBN=8.00637,already_RB used_VBN...

3000 | 3000 | 3000

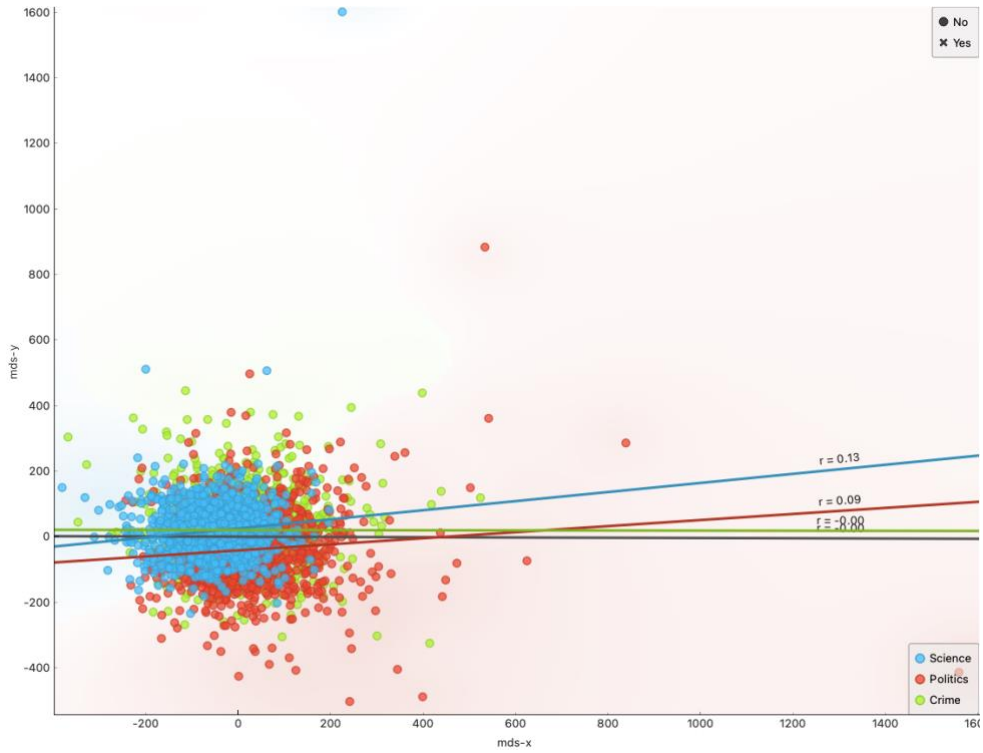
Fonte: Elaborado pela autora com os dados de pesquisa.

O Gráfico 6, 7, 8, 9 e 10 são diagramas de dispersão com as linhas de regressão. Em todos os gráficos foram usadas variáveis categóricas como é a variável categorias para mostrar a cor, dessa forma, a pontuação foi calculada assim: para cada instância de dados, o método encontra 10 vizinhos mais próximos no espaço 2D projetado, ou seja, na combinação de pares de atributos. Em seguida, verifica quantos deles têm a mesma cor. A pontuação total da projeção é então o número médio de vizinhos da mesma cor. Se tiveram sido escolhidos para expor as cores variáveis numéricas, a computação seria semelhante, exceto que o coeficiente de determinação seria usado para medir a homogeneidade local da projeção. Esses gráficos demonstram a utilidade da classificação. A primeira projeção do gráfico de dispersão foi definida como o gráfico padrão da largura da sépala para o comprimento da sépala (usamos o conjunto de dados Iris para simplificar). Ao executar a otimização Localizar Projeções Informativas (LPI), o diagrama de dispersão foi convertido em uma projeção onde se otimizou a largura ou distância entre os tokens dos documentos ou gráfico do comprimento dos tokens dos documentos.

O Gráfico 6 apresenta o diagrama de dispersão gerado durante a fase de pré-processamento, após de ter gerado os tokens, configurar o saco de palavras (BoW), calcular a distância euclidiana, gerar uma matriz de distância e aplicar a técnica MDS. No gráfico, as cores

representam as três categorias utilizadas até agora para classificar o corpus documental formado por e-mails. No eixo *x* aparecem os valores de *MDS-x* como variáveis dependentes e no eixo *y* aparecem os valores do *MDS-y* como vaiáveis independentes. As linhas de cores com o valor *r* representam as linhas de regressão. Assim, temos para a categoria "Ciência" o valor *r* igual a 0.13, para a categoria "Política" o *r* é igual a 0.09, para a categoria "Crime" o valor *r* é igual a 0.00. Na cor preta, aparece a linha de regressão para o valor 0.00.

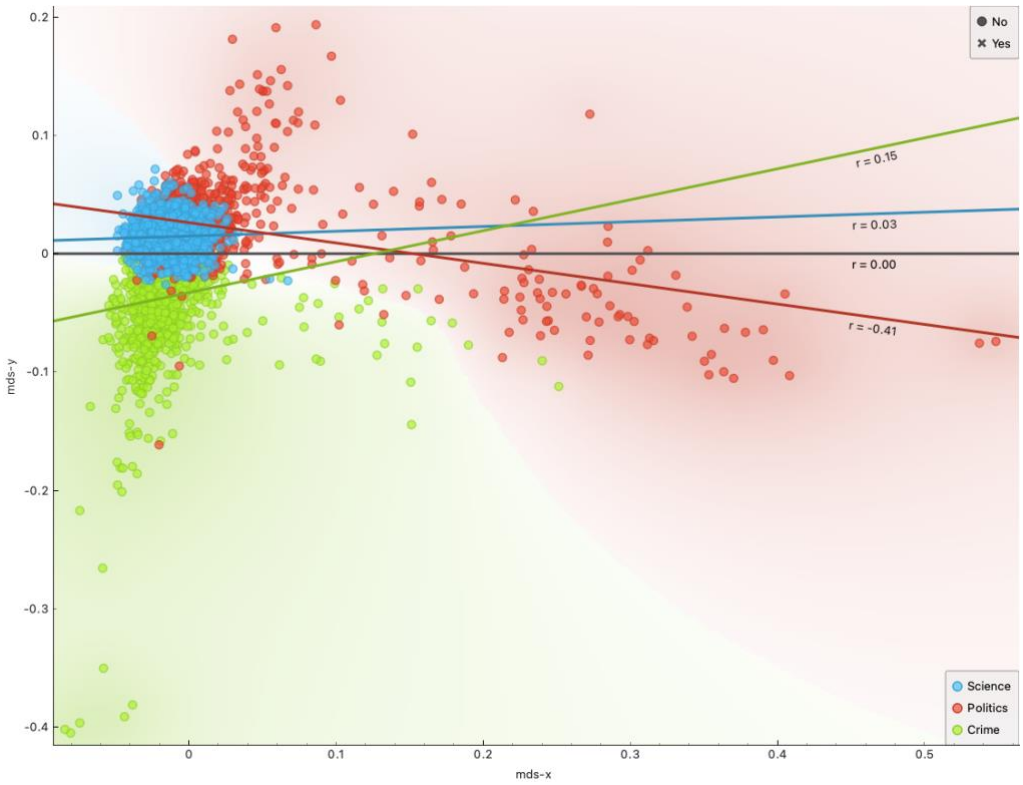
Gráfico 6 – Diagrama de dispersão da distância Euclidiana e MDS



Fonte: Elaborado pela autora com os dados de pesquisa.

O Gráfico 7 apresenta o diagrama de dispersão gerado durante a fase de pré-processamento, após de ter gerado os tokens, configurar o saco de palavras (BoW), calcular a distância de cosseno, gerar uma matriz de distância e aplicar a técnica MDS. No gráfico, as cores representam as três categorias utilizadas até agora para classificar o corpus documental formado por e-mails. No eixo *x* aparecem os valores de *MDS-x* como variáveis dependentes e no eixo *y* aparecem os valores do *MDS-y* como vaiáveis independentes. As linhas de cores com o valor *r* representam as linhas de regressão. Assim, temos para a categoria "Ciência" o valor *r* igual a 0,03, para a categoria "Política" o *r* é igual a -0,41, para a categoria "Crime" o valor *r* é igual a 0,15. Na cor preta, aparece a linha de regressão para o valor 0,00.

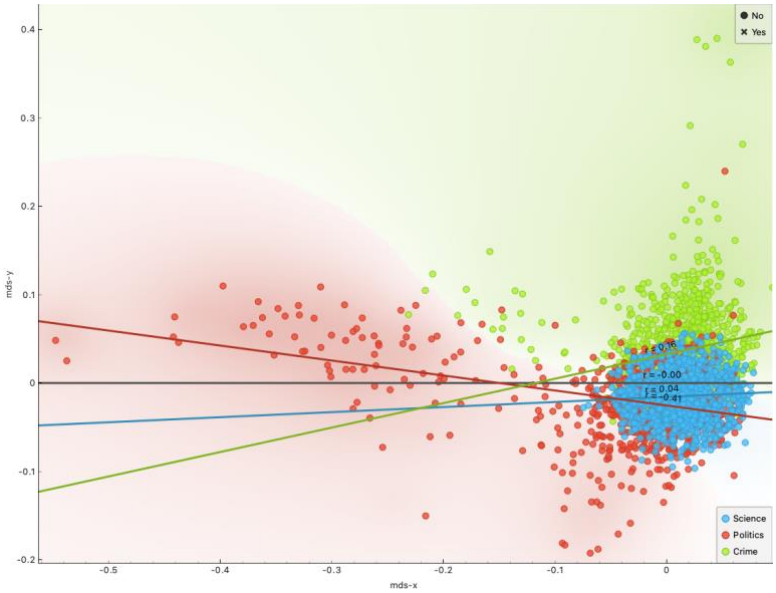
Gráfico 7 – Diagrama de dispersão da distância de cosseno e MDS



Fonte: Elaborado pela autora com os dados de pesquisa.

O Gráfico 8 apresenta o diagrama de dispersão gerado durante a fase de pré-processamento, após de ter gerado os tokens, configurar o saco de palavras (BoW), calcular a Similaridade Hashing, calcular a distância de cosseno, gerar uma matriz de distância e aplicar a técnica MDS. No gráfico, as cores representam as três categorias utilizadas até agora para classificar o corpus documental formado por e-mails. No eixo *x* aparecem os valores de *MDS-x* como variáveis dependentes e no eixo *y* aparecem os valores do *MDS-y* como vaiáveis independentes. As linhas de cores com o valor *r* representam as linhas de regressão. Assim, temos para a categoria “Ciência” o valor *r* igual a 0,04, para a categoria “Política” o *r* é igual a -0,41, para a categoria “Crime” o valor *r* é igual a 0,16. Na cor preta, aparece a linha de regressão para o valor 0,00.

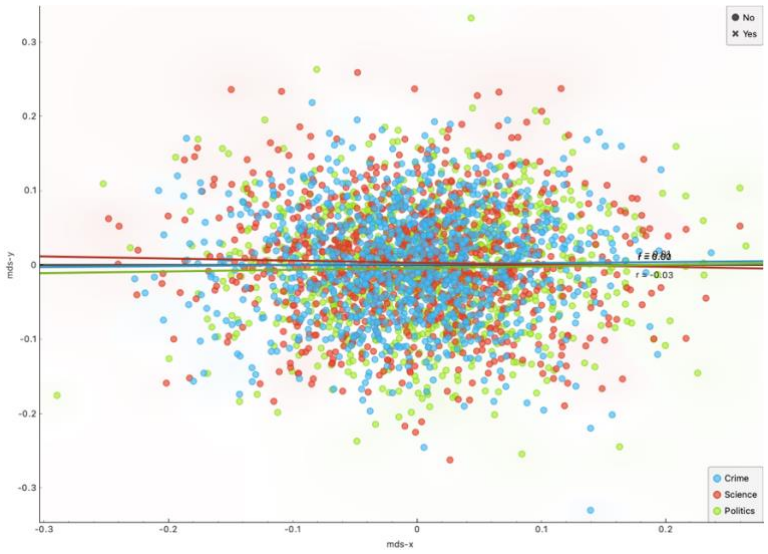
Gráfico 8 – Diagrama de dispersão da distância de cosseno, Similaridade *Hashing* e MDS



Fonte: Elaborado pela autora com os dados de pesquisa.

O Gráfico 9 apresenta o diagrama de dispersão gerado durante a fase de pré-processamento, após de ter gerado os tokens, configurar o saco de palavras (BoW), calcular a Similaridade Hashing, gerar uma matriz de distância e aplicar a técnica MDS. No gráfico, as cores representam as três categorias utilizadas até agora para classificar o corpus documental formado por e-mails. No eixo *x* aparecem os valores de *MDS-x* como variáveis dependentes e no eixo *y* aparecem os valores do *MDS-y* como vaiáveis independentes. As linhas de cores com o valor *r* representam as linhas de regressão. Assim, temos para a categoria “Ciência” o valor *r* igual a -0,03, para a categoria “Política” o *r* é igual a 0,00, para a categoria “Crime” o valor *r* é igual a 0,01. Na cor preta, aparece a linha de regressão para o valor 0,00.

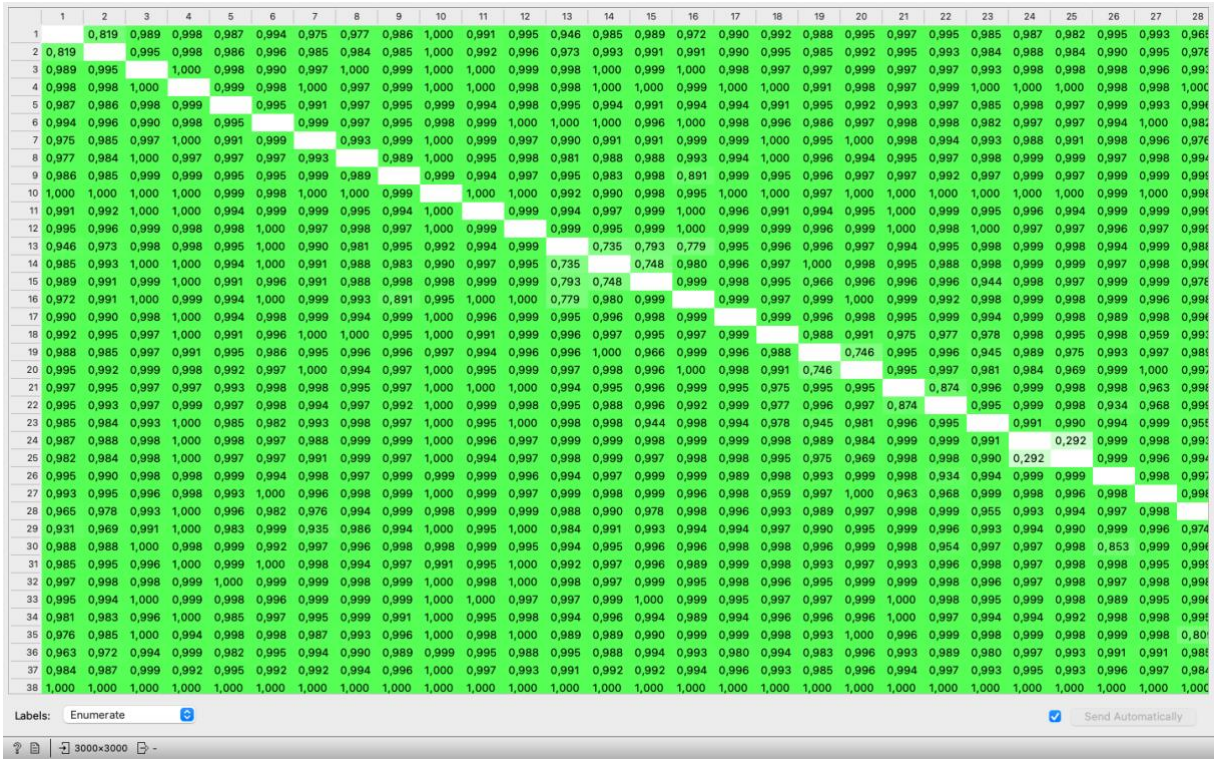
Gráfico 9 – Diagrama de dispersão da Similaridade *Hashing* e MDS



Fonte: Elaborado pela autora com os dados de pesquisa.

A Figura 55 apresenta a Matriz de distância de cosseno, que é uma matriz bidimensional contendo as distâncias, tomadas em pares, entre os elementos de um conjunto, nesse caso, é um conjunto de documentos numerados de 1 a 3000, mesmo que aqui apenas aparecem nas filas os 38 primeiros documentos da categoria “Crime” e nas colunas aparecem os 28 primeiros documentos do corpus pertencentes à categoria “Crime”. Temos valores de entre zero e um (0 e 1), sendo que mais próximo de 1 mais perto estão, portanto, mais semelhante são. Destaca-se, por exemplo, o documentos número 38 que apresenta valores de um (1.00) na distância com os 28 primeiros documentos. Cabe destacar que o número de elementos no conjunto de dados define o tamanho da matriz. As matrizes de dados são essenciais para o agrupamento hierárquico e são extremamente úteis na bioinformática também, onde são usadas para representar estruturas proteicas de forma independente, mas como se pode observar, também pode se aplicar no *text mining* e no PLN. Como entrada, se usaram os valores da distância de cosseno e como saída obtivemos uma tabela de dados contendo a matriz de distância. A matriz apresenta as linhas da amostra a partir do conjunto de dados utilizado nesse estudo e as saídas na Matriz de Distância.

Figura 55 – Matriz de distância de cosseno



Fonte: Elaborado pela autora com os dados de pesquisa.

A Figura 56 apresenta a Matriz de distância euclidiana, que é uma matriz bidimensional contendo as distâncias, tomadas em pares, entre os elementos de um conjunto, nesse caso, é um conjunto de documentos numerados de 1 a 3000, mesmo que aqui apenas aparecem nas filas os 38 primeiros documentos da categoria “Crime” e nas colunas aparecem os 28 primeiros documentos do corpus pertencentes à categoria “Crime”. Igual que na matriz de distancia de cosseno, o número de elementos no conjunto de dados define o tamanho da matriz. Porém, neste caso, a distância é uma medida continua e infinita. Quanto mais o número mais perto estão um do outro, ou seja, mais semelhantes. Observando-se que o menor valor das distâncias é o caso da linha 38. Isso mostra como as duas medidas, euclidiana e de cosseno se correlacionam em muitos casos, porém, não são perfeitamente correlacionadas. Pois, como exemplo, na Figura 55 a distância (de cosseno) entre a coluna 19 e a linha 38 apresentam distância com o máximo valor, igual a 1, significando que os dois estão no mesmo lugar. Já no caso da Figura 56, a distância (euclidiana) entre a coluna 19 e a linha 38 não é a menor encontrada na coluna, significando que a linha 20 é mais perto da coluna 19 do que a linha 38.

Figura 56 – Matriz de distância euclidiana

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1		532,229	470,153	466,393	516,699	471,648	472,631	481,114	490,544	464,661	480,846	481,814	467,319	467,808	472,572	473,568	475,721	471,244	497,263
2	532,229		373,482	367,927	430,555	374,960	377,700	388,289	398,639	365,718	386,475	387,685	373,312	370,717	376,192	380,131	380,003	374,465	406,563
3	470,153	373,482		118,260	257,246	138,479	149,587	176,157	198,190	110,793	169,437	170,722	140,986	127,959	143,933	154,247	154,152	138,102	213,880
4	466,393	367,927	118,260		248,693	122,348	134,404	163,102	186,797	88,869	156,003	157,371	124,531	109,528	127,901	139,300	139,347	121,463	203,039
5	516,699	430,555	257,246	248,693		258,877	264,315	280,618	294,543	245,247	276,061	277,328	259,909	253,024	261,078	267,066	267,134	258,060	305,380
6	471,648	374,960	138,479	122,348	258,877		153,134	178,769	200,399	115,179	172,319	173,752	144,644	131,861	147,188	157,495	157,398	141,623	215,173
7	472,631	377,700	149,587	134,404	264,315	153,134		186,786	208,230	127,911	181,022	182,125	154,118	142,432	156,840	166,856	166,919	152,368	222,944
8	481,114	388,289	176,157	163,102	280,618	178,769	186,786		226,739	157,936	202,925	204,404	178,849	169,514	181,767	190,327	190,396	178,282	241,367
9	490,544	398,639	198,190	186,797	294,543	200,399	208,230	226,739		182,230	222,207	223,560	201,679	191,908	203,936	200,412	211,393	199,791	257,930
10	464,661	365,718	110,793	88,869	245,247	115,179	127,911	157,936	182,230		150,445	151,968	117,169	100,951	120,934	132,804	133,085	114,206	199,243
11	480,846	386,475	169,437	156,003	276,061	172,319	181,022	202,925	222,207	150,445		198,771	173,510	163,277	176,305	184,763	184,437	170,984	236,411
12	481,814	387,685	170,722	157,371	277,328	173,752	182,125	204,404	223,560	151,968	198,771		175,205	164,510	177,555	185,997	185,975	172,948	237,617
13	467,319	373,312	140,986	124,531	259,909	144,644	154,118	178,849	201,679	117,169	173,510	175,205		115,162	132,965	140,733	158,814	143,505	217,359
14	467,808	370,717	127,959	109,528	253,024	131,861	142,432	169,514	191,908	100,951	163,277	164,510	115,162		118,980	146,258	147,398	130,716	209,382
15	472,572	376,192	143,933	127,901	261,078	147,188	156,840	181,767	203,936	120,934	176,305	177,555	132,965	118,980		161,679	161,667	146,289	216,371
16	473,568	380,131	154,247	139,300	267,066	157,495	166,856	190,327	200,412	132,804	184,763	185,997	140,733	146,258	161,679		170,893	156,522	226,333
17	475,721	380,003	154,152	139,347	267,134	157,398	166,919	190,396	211,393	133,085	184,437	185,975	158,814	147,398	161,667	170,893		156,660	226,002
18	471,244	374,465	138,102	121,463	258,060	141,623	152,368	178,282	199,791	114,206	170,984	172,948	143,505	130,716	146,289	156,522	156,660		214,817
19	497,263	406,563	213,880	203,039	305,380	215,173	222,944	241,367	257,930	199,243	236,411	237,617	217,359	209,382	216,371	226,333	226,002	214,817	
20	466,443	367,905	119,430	99,386	248,763	123,512	135,497	163,784	187,483	90,503	156,622	158,365	125,622	110,772	128,840	140,385	140,291	122,124	186,586
21	472,311	375,332	140,313	123,828	259,409	143,968	154,253	179,671	201,472	116,899	173,492	174,825	145,535	132,942	148,471	158,644	158,325	141,425	216,872
22	476,482	380,518	154,577	139,900	267,706	157,903	166,946	191,145	211,119	133,711	185,203	186,355	159,336	147,384	161,979	170,736	171,405	155,495	226,377
23	488,402	395,911	192,337	181,145	289,691	193,959	202,543	223,096	240,783	176,393	217,601	219,118	196,676	187,517	193,858	206,239	205,929	193,001	247,349
24	469,649	372,534	134,363	117,100	256,687	137,987	148,063	175,327	197,502	109,554	168,379	169,785	140,093	126,847	142,905	153,289	153,306	137,180	212,505
25	478,436	383,970	165,766	152,144	274,218	168,707	177,031	200,407	219,874	146,416	194,094	195,559	170,340	159,730	172,707	181,394	181,385	167,835	231,723
26	485,666	391,527	181,195	168,711	284,090	183,580	192,121	213,073	231,851	163,638	207,876	208,662	185,067	175,589	187,733	195,693	194,689	183,322	244,832
27	477,216	382,002	157,357	143,012	268,925	160,854	169,820	193,514	213,916	137,014	187,565	188,581	162,394	151,076	164,964	173,730	173,864	156,882	228,487
28	478,913	386,811	174,098	161,588	279,474	176,008	183,822	207,137	226,672	156,109	202,008	203,116	178,019	168,070	179,387	189,306	189,159	176,227	239,534
29	506,733	429,674	261,126	253,421	338,421	263,717	262,765	283,346	298,367	250,024	280,355	281,764	263,354	257,372	265,765	271,486	271,416	263,090	308,509
30	476,175	380,612	156,261	141,421	268,772	158,871	168,539	192,212	212,744	135,286	186,427	187,215	160,635	149,414	163,402	172,380	172,577	158,525	227,351
31	476,954	382,909	159,716	145,685	270,989	163,134	172,190	195,095	215,422	139,231	189,213	190,819	164,147	153,443	166,960	175,175	176,116	162,287	229,675
32	469,107	371,476	128,858	110,702	254,033	132,795	143,864	171,039	193,775	102,723	164,184	165,701	134,743	120,879	137,808	148,219	148,425	131,657	209,615
33	474,052	377,656	147,101	131,344	263,511	150,237	160,309	185,118	206,229	124,704	178,836	179,926	152,080	140,140	154,935	164,432	164,096	149,428	221,359
34	526,323	442,993	278,974	271,361	352,121	280,829	286,008	300,968	313,231	268,210	296,701	297,854	281,540	275,456	283,001	287,649	288,255	280,355	324,255
35	471,692	376,199	145,799	129,607	262,800	149,125	158,141	183,608	205,033	123,190	177,691	179,076	150,174	138,101	152,953	163,260	163,350	148,306	220,149
36	566,503	493,473	358,750	353,138	417,230	360,243	364,383	375,183	385,723	350,699	372,803	372,480	361,025	355,544	362,038	366,027	364,557	359,820	392,802
37	509,700	422,975	243,815	234,191	326,857	245,174	251,346	267,858	283,079	231,061	263,927	264,272	246,162	239,092	247,929	254,168	254,375	244,832	292,707
38	462,150	362,536	99,739	74,636	240,501	104,697	118,465	150,387	175,756	62,130	142,501	144,107	107,222	89,211	111,019	124,004	124,033	103,517	193,456

Fonte: Elaborado pela autora com os dados de pesquisa.

Após apresentar, observar e analisar os resultados obtidos na fase de pré-processamento, é iniciada a fase de modelagem onde os dados foram divididos em dados de treinamento e dados de teste, foram gerados os modelos e avaliados os modelos, como mostrado a seguir.

4.3 Modelagem: especificação, teste, treinamento, avaliação e calibração hiperparamêtrica dos modelos (Fase V e VI da metodologia)

Nessas fases de modelagem (fase V e VI da metodologia) se descrevem as especificações, os experimentos, os testes, os treinamento, avaliação e calibração hiperparamêtrica dos modelos nos diferentes experimentos para gerar classificadores do corpus documental.

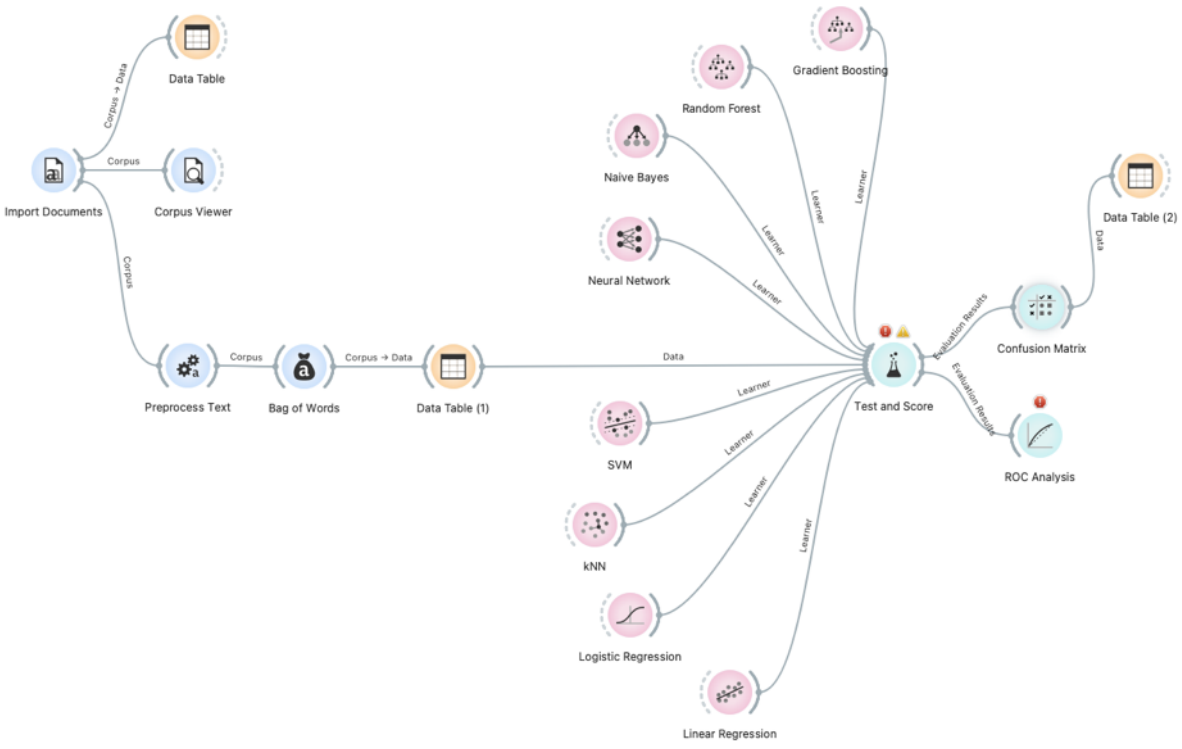
A seguir se apresentam as algoritmos utilizados para categorizar os e-mails (SVM, k-NN, *Naive Bayes Miltilinear* e Regressão Logística).

Nos experimentos, primeiramente, buscou-se prever a categoria do documento do *dataset* usado no pré-processamento, com parâmetros usados para gerar o BoW no Experimento 1 da fase de pré-processamento para obter as frequências de termo pelas quais foram calculados o modelo na seguinte fase da metodologia (fase de modelagem) na que se utilizaram os algoritmos oferecidos pela ferramenta Orange Data Mining para fazer o teste e obter os valores por meio

do widget *Text & Score* conectando o BoW gerado previamente para realizar uma modelagem preditiva. Além disso, foi escolhido o classificador SVM, kNN, Regressão Logística, *Naive Bayes Multilinear*, para categorização, que foram conectado com o *Test & Score* para calcular as pontuações de desempenho dos dados de entrada. Nesse ponto cabe destacar que o teste como Regressão Linear deu erro e não gerou nenhuma pontuação pois esse é um algoritmo que trabalha melhor com biclasses e não com multiclasse (como é o casso desse estudo), pelo que que foi desconsiderado. Também foram treinados e testados um modelo baseado em redes neurais, outro baseado em técnicas de aprendizado de máquina como florestas aleatórias (*Ramdom Forest*) e *Gradient Boosting* (Sci-Learn). Obtendo uns resultados significativos que permitem verificar onde o modelo cometeu um erro e qual foram os melhores acurácias e precisão. Finalmente, foi calculada a Matriz de Confusão a parti dos resultados do *Test & Score* e foram gerados uns gráficos de dispersão. A matriz de confusão exibe os documentos classificados correta e incorretamente como mostrado mais adiante nesse trabalho. A seleção do erro classificado irá produzir documentos classificados inadequadamente que foram analisados para ajustar o modelo.

A seguir se apresenta o esquema realizado em Orange Data Mining para o cenário 1 do Experimento 1 da fase de Modelagem (Figura 57).

Figura 57 –Experimento 1 da fase de Modelagem para treinar e testar modelos de classificação documental



Fonte: Elaborado pela autora desde Orange.

O sistema gerado no Experimento 1 da fase de Modelagem fiz predições com um o conjunto de dados balanceado composto por um total de 3000 instâncias, 283950 variáveis, 283916 características numéricas, um *target* categórica, 33 metadados (9 categóricas, 21 numéricos e 3 *strings*), 435731 tokens e 23396 tipos de tokens. Para obter os resultados da avaliação foram usados 7 métodos ou modelos para classificar 3000 instancias (documentos) testadas.

Na Tabela 7, 8, 9 e 10, são apresentados os resultados após treinar e testar os 7 modelos por meio da validação cruzada k-iterações (*k-fold cross-validation*). As tabelas apresentam o cenário do experimento 1 da fase de modelagem, os modelos de classificação treinados e testados e os resultados obtidos após realizar a validação cruzada estratificada com k=10 iterações (*folds*). A validação cruzada escolhida, divide os dados em um determinado número de repetições ou iterações (geralmente 5 ou 10). Os algoritmos é testado apresentando exemplos de uma iteração de cada vez; o modelo é induzido a partir de outras iterações e os exemplos da iteração estendida são classificados. Isso é repetido para todas as iterações e modelos. Para o caso atual do classificador com 3 classes (crime, politica e ciência), as pontuações são computadas para a classe 1 como classe-alvo, classe 2 como classe-alvo e classe 3 como classe-alvo. Essas pontuações são calculadas com pesos com base no tamanho do modelo de aprendizado para recuperar a pontuação final.

Os resultados de avaliação da estatística de desempenho dos classificadores ou modelos de classificação vem dado, como mostrado nas tabelas, pela média da classificação obtida no cálculo da *área sob a curva ROC (AUC), acurácia (CA), F1 Score, Precisão, Recall, Especificidade e LogLoss.* O significado e uso dessas avaliações estatísticas são as seguintes:

- *Area sob a curva ROC (Area Under Curve -AUC):* é uma métrica robusta muito útil para problemas de classificação binária. A AUC (), representa a área sob a curva. O valor dessa métrica está em uma faixa entre 0 e 1, onde 0 é como se tivéssemos um modelo aleatório, ou seja, se jogássemos os dados no ar teríamos um resultado melhor, e 1 é um ótimo resultado que indica que nosso modelo generaliza muito bem.
- A*curácia (CA):* representa a porcentagem total de valores classificados corretamente, tanto positivos quanto negativos.
- *F1 Score: é a* média harmônica ponderada de precisão e *Recall* e é uma métrica amplamente utilizada em problemas em que o conjunto de dados a ser analisado é desbalanceada.

- *Precisão*: é a proporção de verdadeiros positivos entre os casos classificados como positivos e é uma métrica de precisão usada para saber qual porcentagem de valores que foram classificados como positivos são realmente positivos, por exemplo, a proporção dos documentos da classe ou categoria Crime é corretamente identificada como Crime em alguns modelos como o de Redes Neurais onde te valore de precisão do 0,968, ou seja, o modelo presenta uma precisão para a categoria Crime (Tabela 8) de quase um 97%). Recomenda-se usar essa métrica em problemas em que os dados são balanceados, ou seja, há o mesmo número de valores de cada rótulo (neste caso, o mesmo número de 1s e 0s).
- *Recall*: é a proporção de verdadeiros positivos entre todas as instâncias positivas nos dados e é usada para descobrir quantos valores positivos estão classificados corretamente, por exemplo, o número de documentos da classe "Crime" entre todos os classificados como "Crime".
- *Especificidade* (*Specifity*): é a proporção de verdadeiros negativos entre todas as instâncias negativas, por exemplo, o número de documentos que não pertencem a classe "Crime" entre todos os documentos categorizados fora da classe "Crime").
- *LogLoss*: é a perda de entropia cruzada que leva em consideração a incerteza de sua previsão com base em quanto ela varia do rótulo real. Entendendo entropia como medida de incerteza existente.
- *Tempo de treinamento*: é o tempo cumulativo em segundos usado para modelos de treinamento.
- *Tempo de teste*: é tempo cumulativo em segundos usado para testar modelos.

Essas métricas servem para avaliar modelos de classificação e podem ser utilizadas em tanto em problemas de classificação binária quanto em problemas multiclasse. Se os dados estão desbalanceados, a métrica mais adequada é o F-score e para dados balanceados a métricas mais adequadas são Precisão e AUC, porque dão maior relevância aos dados que aparecem com maior frequência.

Na Tabela 7, é apresentada a média da validação cruzada k-iterações dos classificadores ou modelos de classificação escolhidos, sendo que os modelos de classificação kNN e SVM são os que apresentam os piores resultados apresentas em todas avaliação estatística dos classificadores (AUC, CA, F1, Precisão e Recall). Pelo contrario, a Rede Neural é o modelo que apresenta os melhores resultados, apresentando uma precisão e acurácia de quase o 97% (0,969) e chegando quase ou 100% (0,992 = 99,2%) nos valores de AUC (área sob a curva

ROC), o que supõe o valor um na área sob a curva, ou seja, um diagnóstica perfeito. Para calcular a AUC foi comparada a área sob a curva dos diferentes testes. A área sob a curva ROC tem um valor entre 0,5 e 1, onde 1 representa um valor de diagnóstico perfeito e 0,5 é um teste sem capacidade diagnóstica discriminatória. Os modelos Regressão Logística, *Gradient Boosting* e Florestas Aleatórias (*Random Forest*) também apresentaram ótimos valores como classificadores. Finalmente, cabe destacar que o modelo *Naive Bayes* não apresentou nenhum valor relativo à AUC e apresentando uma Precisão do 88,4% (0,884) e uma acurácia de 83,4% (0,834).

Tabela 7 – Resultados da validação cruzada k-fold dos modelos: média das classes

Experimento	Modelo	AUC	CA	F1	Precisão	Recall
1	kNN	0.698	0.440	0.368	0.744	0.440
1	SVM	0.618	0.408	0.317	0.645	0.408
1	Random Forest	0.967	0.887	0.888	0.898	0.887
1	Neural Network	0.992	0.969	0.969	0.969	0.969
1	Naive Bayes		0.834	0.837	0.884	0.834
1	Regressão Logística	0.994	0.958	0.958	0.961	0.958
1	Gradient Boosting (Sci-Learn)	0.980	0.896	0.897	0.908	0.896

Fonte: Elaborado pela autora com os resultados de pesquisa.

Na Tabela 8, apresenta-se a validação cruzada k-iterações dos classificadores ou modelos de classificação documental escolhidos para a categoria “Crime”. Os modelos kNN e SVM são os que apresentam os piores resultados como classificadores em quase todas avaliação estatística (AUC, CA, F1 e Recall), porém o classificador documental SVM apresenta uma precisão do 100% (1000) e o classificador k-NN tem uma precisão do 97% (0,971). A Rede Neural e Regressão logística são os modelo que apresenta os melhores resultados, superando o 99% (0,992 = 99,2%) nos valores da AUC, pelo que é um modelo quase perfeito por se aproximar no valor 1 dentro da curva. Além disso, a Rede Neural apresenta uma precisão de 96.8% e uma acurácia de quase o 97,4% e a Regressão Logística apresenta o 98,6% de precisão e o 97% de acurácia. Já os modelos *Gradient Boosting* e *Random Forest* também apresentaram ótimos valores como classificadores. Finalmente, cabe destacar, de novo, que o modelo *Naive Bayes* não apresentou nenhum valor relativo à AUC e apresentando uma precisão do 98,6% (0,986) e uma acurácia de 93,6% (0,936).

Tabela 8 – Resultados da validação cruzada k-fold dos modelos para a categoria “Crime”

Experimento	Modelo	AUC	CA	F1	Precisão	Recall
1	kNN	0.725	0.721	0.286	0.971	0.168
1	SVM	0.533	0.675	0.051	1000	0.026
1	Random Forest	0.963	0.934	0.894	0.959	0.837
1	Neural Network	0.990	0.974	0.960	0.968	0.952
1	Naive Bayes		0.936	0.896	0.986	0.821
1	Regressão Logística	0.991	0.970	0.954	0.989	0.921
1	Gradient Boosting (Sci-Learn)	0.974	0.939	0.902	0.968	0.845

Fonte: Elaborado pela autora com os resultados de pesquisa.

A Tabela 9 apresenta a validação cruzada k-iterações dos classificadores ou modelos de classificação documental escolhidos para a categoria “Política”. Os modelos kNN e SVM apresentam, os piores resultados como classificadores em todas avaliação estatística (AUC, CA, F1 e Precisão) menos no modelo k-NN onde Recall diz que o 99% dos valores positivos estão classificados corretamente. A Regressão logística, Rede Neural e *Naive Bayes* são os modelo de classificação documental que apresenta os melhores resultados, superando o 99% nos valores da AUC, pelo que é um modelo quase perfeito por se aproximar no valor 1 dentro da curva. Los classificadores *Gradient Boosting* (com um 98,6% na AUA e um 93,6% de acurácia e precisão) e *Random Forest* (com um 97,4% na AUA, um 94,5% de acurácia e um 95,5% de precisão) também apresentaram ótimos valores.

Tabela 9 – Resultados da validação cruzada k-fold dos modelos para a categoria “Politica”

Experimento	Modelo	AUC	CA	F1	Precisão	Recall
1	kNN	0.734	0.445	0.546	0.374	0.990
1	SVM	0.702	0.687	0.372	0.563	0.278
1	Random Forest	0.974	0.936	0.900	0.936	0.866
1	Neural Network	0.995	0.987	0.980	0.987	0.974
1	Naive Bayes	0.990	0.893	0.810	0.994	0.683
1	Regressão Logística	0.996	0.982	0.973	0.989	0.957
1	Gradient Boosting (Sci-Learn)	0.986	0.945	0.913	0.955	0.875

Fonte: Elaborado pela autora com os resultados de pesquisa.

Na Tabela 10, pode-se observar a validação cruzada k-iterações dos classificadores ou modelos de classificação documental escolhidos para a categoria “Ciência”. Os modelos kNN e SVM são os que apresentam os piores resultados como classificadores em todas avaliação estatística (AUC, CA, F1 e Recall), porém o classificador documental k-NN apresenta uma precisão do 88,5%. A Rede Neural e Regressão logística são os modelo que apresenta os melhores resultados, com um 99,5% nos valores da AUC, pelo que é um modelo quase perfeito

por se aproximar no valor 1 dentro da curva. Além disso, a Rede Neural apresenta uma precisão de 95,1% e uma acurácia do 97,7%. A Regressão Logística apresenta o 90% de precisão e o 96,3% de acurácia. Já os modelos *Gradient Boosting* e *Random Forest* e *Naive Bayes* também apresentaram valores elevados como classificadores.

Tabela 10 – Resultados da validação cruzada k-fold dos modelos para a categoria "Ciência"

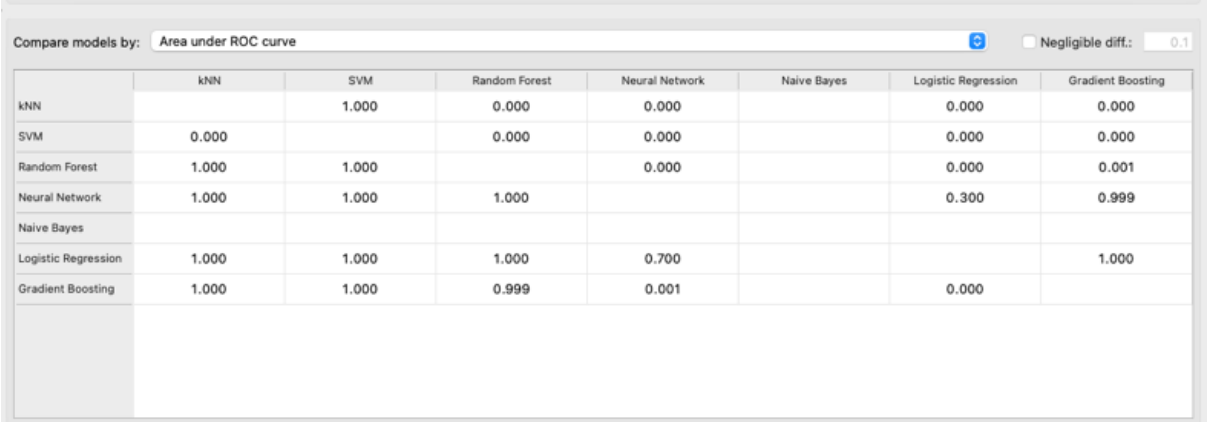
Experimento	Modelo	AUC	CA	F1	Precisão	Recall
1	kNN	0.702	0.714	0.274	0.885	0.162
1	SVM	0.705	0.454	0.529	0.371	0.921
1	Random Forest	0.967	0.905	0.871	0.798	0.959
1	Neural Network	0.995	0.977	0.966	0.951	0.980
1	Naive Bayes	0.937	0.838	0.804	0.674	0.997
1	Regressão Logística	0.995	0.963	0.947	0.904	0.995
1	Gradient Boosting (Sci-Learn)	0.979	0.909	0.877	0.800	0.969

Fonte: Elaborado pela autora com os resultados de pesquisa.

A seguir, apresenta-se uma comparação dos modelos por pares desde as diferentes métricas estatísticas acima descritas (AUC, Acurácia da Classificação, Valores F1, precisão, Recall, LogLoss e especificidades) do Cenário 1 do Experimento1 da fase de Modelagem. Os valores que aparecem nas figuras de comparação de modelos representam a probabilidade de que o modelo correspondente à linha seja melhor que o modelo correspondente à coluna. Sirva como exemplo a Figura 55 onde é possível ver que a probabilidade de que SVM sejam melhor que os outros modelos é de 0,00 (0%), ou seja, é nula. Porem, a probabilidade de que a Rede Neural seja melhor que kNN, SVM e *Randon Forest* é um (100%) e a probabilidade de que Rede Neural seja melhor que *Grandient Boosting* é de quase um (0,999), ou seja, 99,9%.

Assim, a Figura 58 mostra os resultados da comparação dos modelos com base na estatística da área sob a curva ROC (AUC). Os dados revelam que a probabilidade de que o modelo de classificação documental kNN seja melhor que SVM é do 100% (1,000), mas é pior que os demais modelos treinados e testados (*Random Forest*, Rede Neural, Regresão Logística e *Gradient Boosting*), pois os valores são 0.000, ou seja, que a probabilidade de que kNN seja melhor que os outros modelos é do 0%. A probabilidade de que *Random Forest* seja melhor que kNN e SVM é do 100%. Existe uma probabilidade do 100% de que o modelo de Regressão Logística é melhor que kNN, SVM, *Random Forest* e *Grandient Boosting* e que a Rede Neural num 70%. Finalmente a figura demonstra que o modelo *Grandient Boosting* é melhor num 100% que kNN, SVM e que *Random Forest* num 99,9% e é pior que o modelo de Regressão Logística.

Figura 58 – Comparação dos modelos baseada na área sob a curva ROC (AUC) após calcular a media de classes

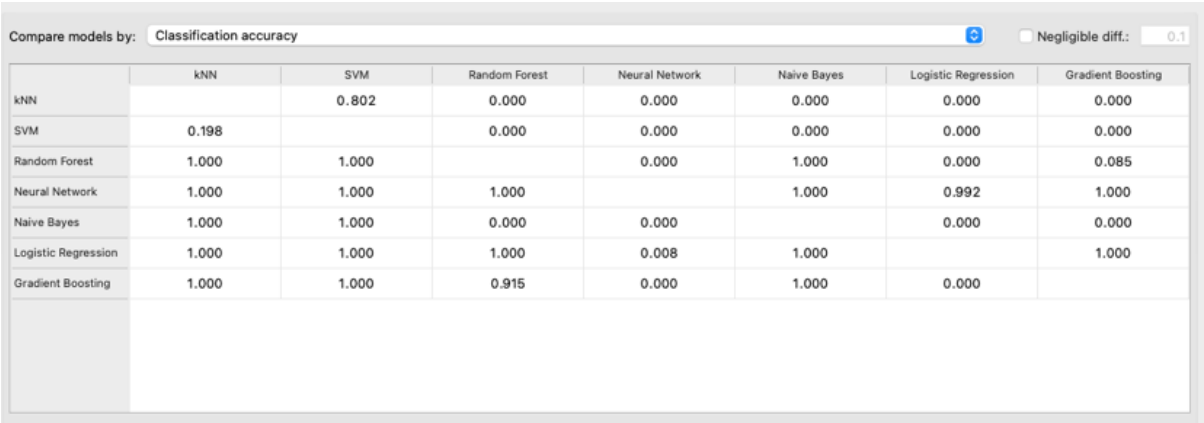


	kNN	SVM	Random Forest	Neural Network	Naive Bayes	Logistic Regression	Gradient Boosting
kNN		1.000	0.000	0.000		0.000	0.000
SVM	0.000		0.000	0.000		0.000	0.000
Random Forest	1.000	1.000		0.000		0.000	0.001
Neural Network	1.000	1.000	1.000			0.300	0.999
Naive Bayes							
Logistic Regression	1.000	1.000	1.000	0.700			1.000
Gradient Boosting	1.000	1.000	0.999	0.001		0.000	

Fonte: Elaborado pela autora com os dados de pesquisa desde Orange.

A Figura 59 mostra os resultados da comparação dos modelos com base na estatística da área sob a curva ROC (AUC). Os dados revelam que a probabilidade de que o modelo de classificação documental kNN seja melhor que SVM é do 100% (1,000), mas é pior que os demais modelos treinados e testados (*Random Forest*, Rede Neural, Rede Neural, Regresão Logística e *Gradient Boosting*), pois os valores são 0,000, ou seja, que a probabilidade de que kNN seja melhor que os outros modelos é do 0%. A probabilidade de que *Random Forest* seja melhor que kNN e SVM é do 100%. Existe uma probabilidade do 100% de que o modelo de Regressão Logística é melhor que kNN, SVM, *Random Forest* e *Grandient Boosting* e que a Rede Neural num 70%. Finalmente, a figura demonstra que o modelo *Grandient Boosting* é melhor num 100% que kNN, SVM e que *Random Forest* num 99,9% e é pior que o modelo de *Regressão Logística.*

Figura 59 – Comparação dos modelos baseada na acurácia da classificação (CA) após calcular a media de classes



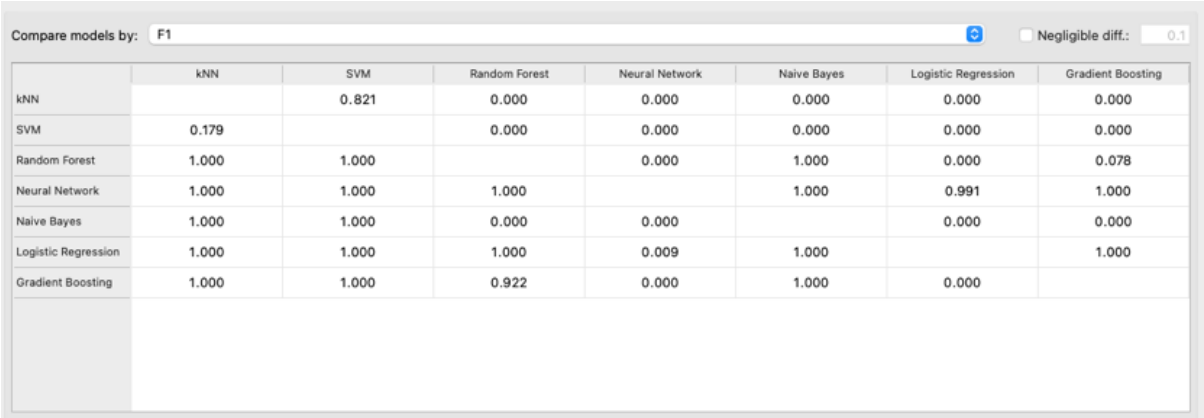
	kNN	SVM	Random Forest	Neural Network	Naive Bayes	Logistic Regression	Gradient Boosting
kNN		0.802	0.000	0.000	0.000	0.000	0.000
SVM	0.198		0.000	0.000	0.000	0.000	0.000
Random Forest	1.000	1.000		0.000	1.000	0.000	0.085
Neural Network	1.000	1.000	1.000		1.000	0.992	1.000
Naive Bayes	1.000	1.000	0.000	0.000		0.000	0.000
Logistic Regression	1.000	1.000	1.000	0.008	1.000		1.000
Gradient Boosting	1.000	1.000	0.915	0.000	1.000	0.000	

Fonte: Elaborado pela autora com os dados de pesquisa desde Orange.

A Figura 60 mostra os resultados da comparação dos modelos com base na estatística dos valores F1. Os dados descobrem que o modelo de classificação documental kNN apenas é

melhor que SVM num 82,1% (0,821), mas é pior que os demais modelos treinados e testados (*Random Forest*, Rede Neural, *Naive Bayes*, Regressão Logística e *Gradient Boosting*), pois os valores são 0,000, ou seja, que a probabilidade de que kNN seja melhor que os outros modelos é do 0%. O modelo SVM é o que apresenta os piores resultados comparativos com os outros modelos. Por outro lado, a probabilidade de que *Random Forest* seja melhor que kNN, SVM e *Naive Bayes* é do 100%. A Rede Neural é o que apresenta os melhores resultados, sendo ele melhor numa probabilidade do 100% que os outros modelos. Existe uma probabilidade do 100% de que o modelo de Regressão Logística é melhor que kNN, SVM, *Random Forest, Naive Bayes* e *Grandient Boosting*. Por último, cabe destacar que o modelo *Grandient Boosting* é melhor num 100% que kNN, SVM e *Naive Bayes*, e que *Random Forest* num 92,2% e é pior que o modelo de Regressão Logística e a Rede Neural.

Figura 60 – Comparação dos modelos baseada nos valores F1 após calcular a media de classes



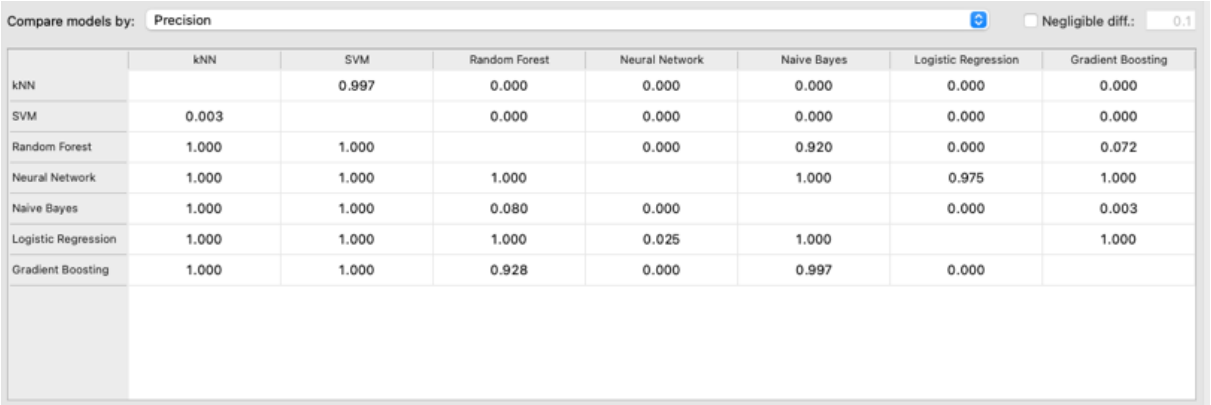
	kNN	SVM	Random Forest	Neural Network	Naive Bayes	Logistic Regression	Gradient Boosting
kNN		0.821	0.000	0.000	0.000	0.000	0.000
SVM	0.179		0.000	0.000	0.000	0.000	0.000
Random Forest	1.000	1.000		0.000	1.000	0.000	0.078
Neural Network	1.000	1.000	1.000		1.000	0.991	1.000
Naive Bayes	1.000	1.000	0.000	0.000		0.000	0.000
Logistic Regression	1.000	1.000	1.000	0.009	1.000		1.000
Gradient Boosting	1.000	1.000	0.922	0.000	1.000	0.000	

Fonte: Elaborado pela autora com os dados de pesquisa desde Orange.

A Figura 61 mostra os resultados da comparação dos modelos com base na estatística da precisão. Os dados mostram que o modelo de classificação documental kNN apenas é melhor que SVM num 99,7% (0,997), quase o 100%, mas é pior que os demais modelos treinados e testados (*Random Forest*, Rede Neural, *Nayve Bayes*, Regressão Logística e *Gradient Boosting*), pois os valores são 0.000, por tanto, a probabilidade de que kNN seja melhor que os outros modelos é do 0%. O modelo SVM é o que apresenta os piores resultados comparativos com os outros modelos. Já a probabilidade de que *Random Forest* seja melhor que kNN e SVM do 100% e que *Naive Bayes* é do 92%. A Rede Neural, neste caso, é o que apresenta os melhores resultados, sendo ele melhor numa probabilidade do 100% que os outros modelos. Existe uma probabilidade do 100% de que o modelo de Regressão Logística é melhor que kNN, SVM, *Random Forest, Naive Bayes* e *Grandient Boosting*, mas é pior que o modelo de Rede Neural. Por último, cabe notar que o modelo *Grandient Boosting* é melhor num 100% que kNN, SVM,

Naive Bayes (com um 99,7%), e que *Random Forest* num 92,8%, mas é pior que o modelo de Regressão Logística e a Rede Neural.

Figura 61 – Comparação dos modelos baseada na precisão após calcular a media de classes

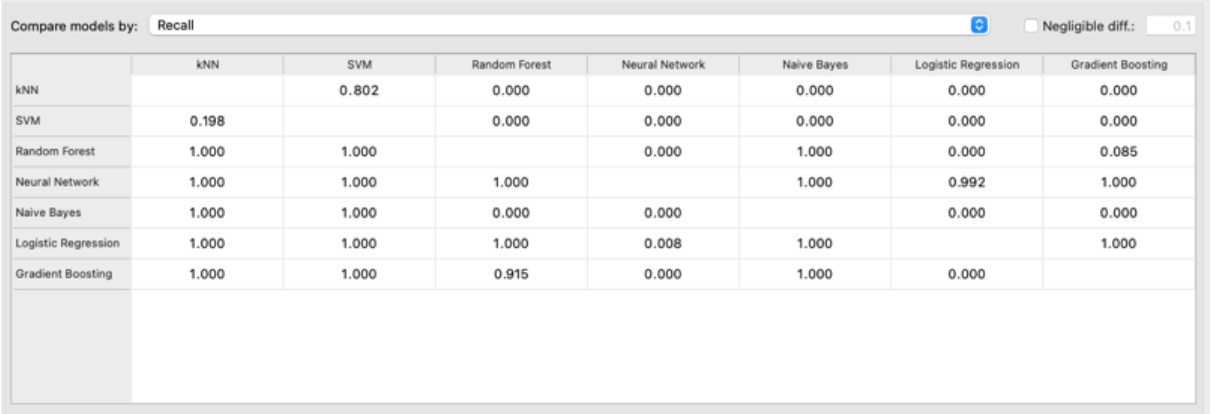


	kNN	SVM	Random Forest	Neural Network	Naive Bayes	Logistic Regression	Gradient Boosting
kNN		0.997	0.000	0.000	0.000	0.000	0.000
SVM	0.003		0.000	0.000	0.000	0.000	0.000
Random Forest	1.000	1.000		0.000	0.920	0.000	0.072
Neural Network	1.000	1.000	1.000		1.000	0.975	1.000
Naive Bayes	1.000	1.000	0.080	0.000		0.000	0.003
Logistic Regression	1.000	1.000	1.000	0.025	1.000		1.000
Gradient Boosting	1.000	1.000	0.928	0.000	0.997	0.000	

Fonte: Elaborado pela autora com os dados de pesquisa desde Orange.

A Figura 62 mostra os resultados da comparação dos modelos com base na estatística de Revocação (Recall). Os dados revelam que o classificador kNN apenas é melhor que SVM num 80% (0,802), mas é pior que os demais modelos de classificação documental treinados e testados (*Random Forest*, Rede Neural, *Naive Bayes*, Regressão Logística e *Gradient Boosting*), pois os valores são 0,000, pelo que a probabilidade de que kNN seja melhor que os outros modelos é do 0%. O modelo SVM, mais uma vez, apresenta os piores resultados comparativos com os outros modelos. Por outro lado, a probabilidade de que *Random Forest* seja melhor que kNN, SVM e *Naive Bayes* é do 100%. A Rede Neural, novamente, apresenta os melhores resultados, sendo ele melhor numa probabilidade do 100% que os outros modelos. Existe uma probabilidade do 100% de que o modelo de Regressão Logística é melhor que kNN, SVM, *Random Forest*, *Naive Bayes* e *Grandient Boosting*. Por último, cabe destacar que o modelo *Grandient Boosting* é melhor num 100% que kNN, SVM, *Naive Bayes* e que *Random Forest* num 91,5% e é pior que o modelo de Regressão Logística e a Rede Neural.

Figura 62 – Comparação dos modelos baseada no Recall após calcular a media de classes

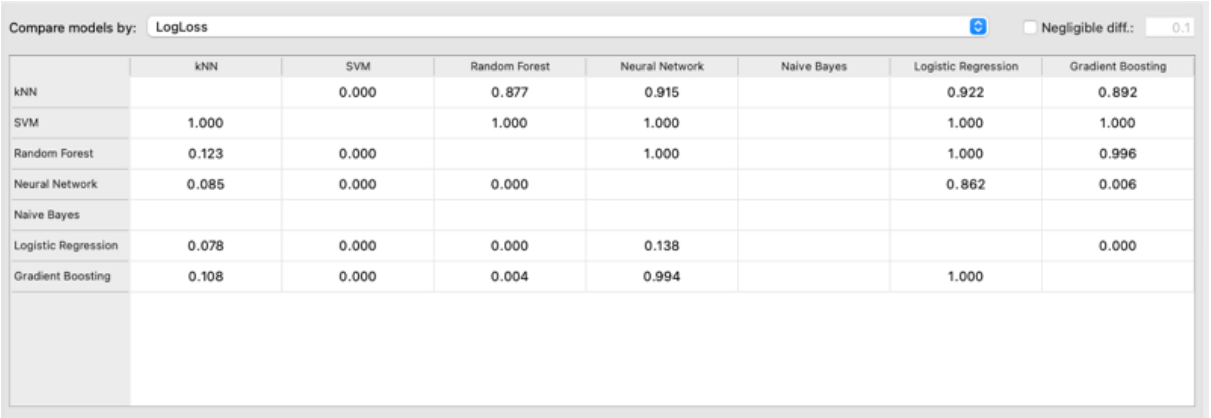


	kNN	SVM	Random Forest	Neural Network	Naive Bayes	Logistic Regression	Gradient Boosting
kNN		0.802	0.000	0.000	0.000	0.000	0.000
SVM	0.198		0.000	0.000	0.000	0.000	0.000
Random Forest	1.000	1.000		0.000	1.000	0.000	0.085
Neural Network	1.000	1.000	1.000		1.000	0.992	1.000
Naive Bayes	1.000	1.000	0.000	0.000		0.000	0.000
Logistic Regression	1.000	1.000	1.000	0.008	1.000		1.000
Gradient Boosting	1.000	1.000	0.915	0.000	1.000	0.000	

Fonte: Elaborado pela autora com os dados de pesquisa desde Orange.

A Figura 63 mostra os resultados da comparação dos modelos com base na estatística dos valores LogLoss (entropia cruzada ou medida de incerteza da precisão do modelo). Os dados mostram que o classificador kNN é pior que SVM e é melhor que *Random Forest* (com um 87,7%), que a Rede Neural (com um 91,5%), que a Regressão Logística (com um 92,2%) e que *Gradient Boosting* (com um 89,2%). Dessa vez, o modelo SVM apresenta melhores resultados que com as validações estatísticas anteriores, pois o classificador é melhor que os outros modelos num 100%. Por outro lado, *Random Forest* é melhor que a Rede Neural, que a Regressão Logística e que *Gradient Boosting* é do 100%. O modelo feito com Rede Neural e o feito com Regressão Logística, nesse caso, apresentam os piores resultados em comparação os outros modelos. Por último, cabe destacar que o modelo *Grandient Boosting* apenar supera ao classificador de Regressão Logística (num 100%) e à Rede Neural (com um 99,4%).

Figura 63 – Comparação dos modelos baseada no LogLoss após calcular a media de classes



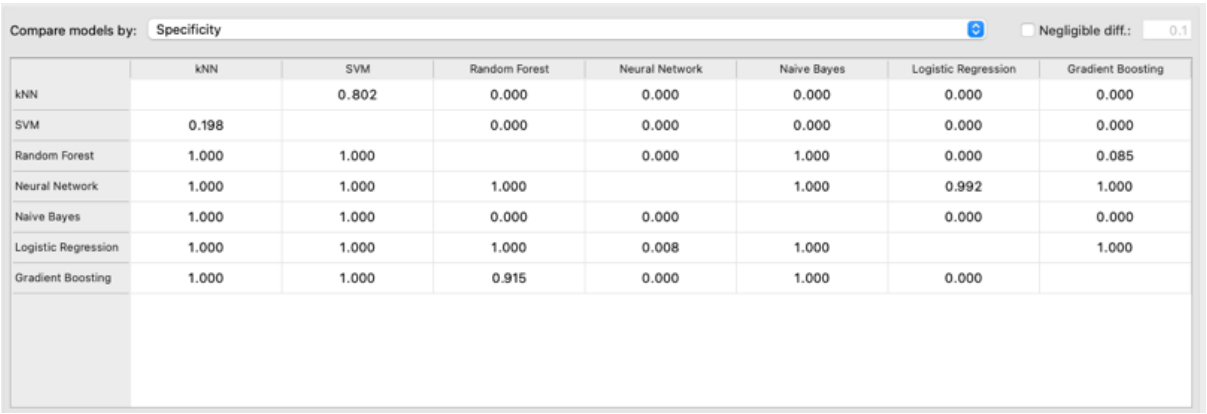
	kNN	SVM	Random Forest	Neural Network	Naive Bayes	Logistic Regression	Gradient Boosting
kNN		0.000	0.877	0.915		0.922	0.892
SVM	1.000		1.000	1.000		1.000	1.000
Random Forest	0.123	0.000		1.000		1.000	0.996
Neural Network	0.085	0.000	0.000			0.862	0.006
Naive Bayes							
Logistic Regression	0.078	0.000	0.000	0.138			0.000
Gradient Boosting	0.108	0.000	0.004	0.994		1.000	

Fonte: Elaborado pela autora com os dados de pesquisa desde Orange.

A Figura 64 mostra os resultados da comparação dos modelos com base na estatística da especificidade do modelo. Os dados revelam que o modelo de classificação documental kNN

apenas é melhor que SVM num 80,2% (0,802), mas é pior que os demais modelos treinados e testados (*Random Forest*, Rede Neural, *Naive Bayes*, Regressão Logística e *Gradient Boosting*), pois os valores são 0,000 (0%). O classificador SVM, apresenta os piores resultados comparativos com os outros modelos. Em contraposição, nenhum dos modelos (kNN, SVM, *Random Forest*, *Naive Bayes*, Regressão Logística e *Gradient Boosting*) é melhor que o modelo feito com a Rede Neural, sendo essa útlima melhor que todos os modelos num 100% em especificidade.

Figura 64 – Comparação dos modelos baseada na especificidade após calcular a media de classes



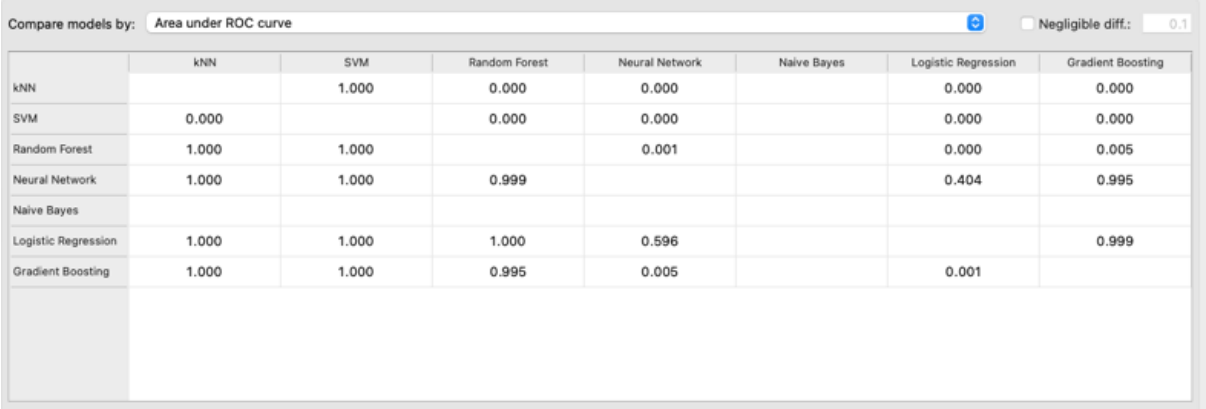
	kNN	SVM	Random Forest	Neural Network	Naive Bayes	Logistic Regression	Gradient Boosting
kNN		0.802	0.000	0.000	0.000	0.000	0.000
SVM	0.198		0.000	0.000	0.000	0.000	0.000
Random Forest	1.000	1.000		0.000	1.000	0.000	0.085
Neural Network	1.000	1.000	1.000		1.000	0.992	1.000
Naive Bayes	1.000	1.000	0.000	0.000		0.000	0.000
Logistic Regression	1.000	1.000	1.000	0.008	1.000		1.000
Gradient Boosting	1.000	1.000	0.915	0.000	1.000	0.000	

Fonte: Elaborado pela autora com os dados de pesquisa desde Orange.

Nas Figuras 65, 66, 67, 68, 69, 70 e 71, apresenta-se uma comparação dos modelos por pares para a categoria “Crime” desde as diferentes métricas estatísticas acima descritas (AUC, Acurácia da Classificação, valores F1, precisão, Recall, LogLoss e especificidade) do Experimento1 da fase de Modelagem. Como comentado anteriormente, os valores que aparecem nas figuras representam a probabilidade de que o modelo correspondente à linha seja melhor que o modelo correspondente à coluna.

Assim, na Figura 65, é possível ver que a probabilidade de que SVM sejam melhor que os outros modelos para a categoria “Crime” é de 0,00 (0%), ou seja, é nula. Acontece a mesma coisa com o classificador kNN que é pior que os outros modelos e melhor que SVM. Porem, a probabilidade de que a Rede Neural seja melhor que kNN, SVM e *Randon Forest* é um (100%) e a probabilidade de que Rede Neural seja melhor que *Grandient Boosting* é do 99,5%. (0,999). A Regressão Logística também é melhor que os outros modelos.

Figura 65 – Comparação dos modelos baseada na área sob a curva ROC (AUC) para a categoria “Crime”

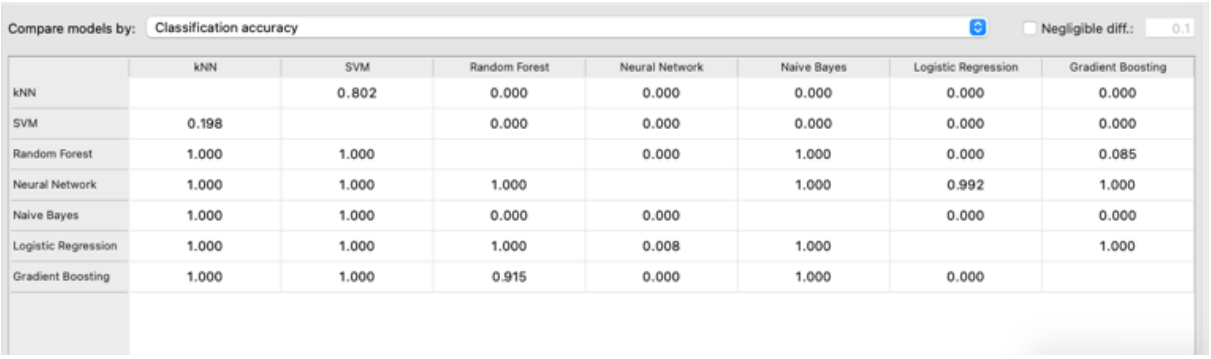


	kNN	SVM	Random Forest	Neural Network	Naive Bayes	Logistic Regression	Gradient Boosting
kNN		1.000	0.000	0.000		0.000	0.000
SVM	0.000		0.000	0.000		0.000	0.000
Random Forest	1.000	1.000		0.001		0.000	0.005
Neural Network	1.000	1.000	0.999			0.404	0.995
Naive Bayes							
Logistic Regression	1.000	1.000	1.000	0.596			0.999
Gradient Boosting	1.000	1.000	0.995	0.005		0.001	

Fonte: Elaborado pela autora com os dados de pesquisa desde Orange.

A Figura 66 mostra os resultados da comparação dos modelos com base na acurácia da classificação do modelo para a categoria “Crime”. Os dados revelam que o modelo de classificação documental kNN apenas é melhor que SVM num 80,2% (0,802), mas é pior que os demais modelos treinados e testados (*Random Forest*, Rede Neural, *Naive Bayes*, Regressão Logística e *Gradient Boosting*), pois os valores são 0,000 (0%). O classificador SVM, apresenta os piores resultados comparativos com os outros modelos novamente. Pelo contrario, nenhum modelo (kNN, SVM, *Random Forest*, *Naive Bayes*, Regressão Logística e *Gradient Boosting*) é melhor que o modelo feito com o modelo de Rede Neural, sendo esse último melhor que todos os modelos num 100% em acurácia de classificação.

Figura 66 – Comparação dos modelos baseada na acurácia da classificação (CA) para a categoria “Crime”



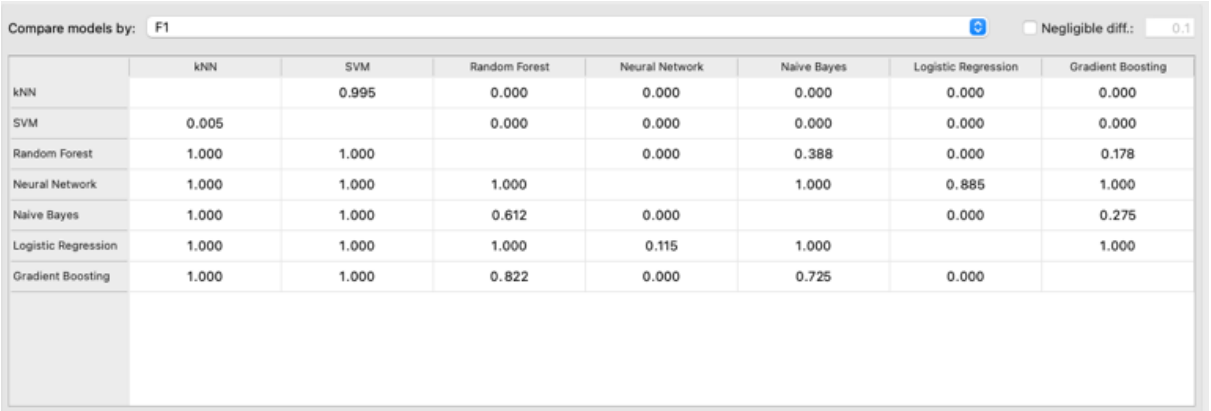
	kNN	SVM	Random Forest	Neural Network	Naive Bayes	Logistic Regression	Gradient Boosting
kNN		0.802	0.000	0.000	0.000	0.000	0.000
SVM	0.198		0.000	0.000	0.000	0.000	0.000
Random Forest	1.000	1.000		0.000	1.000	0.000	0.085
Neural Network	1.000	1.000	1.000		1.000	0.992	1.000
Naive Bayes	1.000	1.000	0.000	0.000		0.000	0.000
Logistic Regression	1.000	1.000	1.000	0.008	1.000		1.000
Gradient Boosting	1.000	1.000	0.915	0.000	1.000	0.000	

Fonte: Elaborado pela autora com os dados de pesquisa desde Orange.

A Figura 67 exibe os resultados da comparação dos modelos com base nos valores de F1 para a categoria “Crime”. Os dados mostram que o modelo de classificação documental kNN só é melhor que SVM num 99,5% (0,995), ou seja, quase o 100%, mas é pior que os demais modelos treinados e testados (*Random Forest*, Rede Neural, *Naive Bayes*, Regressão Logística e *Gradient Boosting*), pois os valores são nulos. De novo, o classificador SVM, que

apresenta os piores resultados comparados com os outros modelos. Pelo contrario, nenhum modelo (kNN, SVM, *Random Forest*, *Naive Bayes*, Regressão Logística e *Gradient Boosting*) é melhor que o modelo feito com o modelo de Rede Neural, sendo esse último melhor que todos os modelos num 100% em termos de valores F1.

Figura 67 – Comparação dos modelos baseada nos valores F1 para a categoria “Crime”



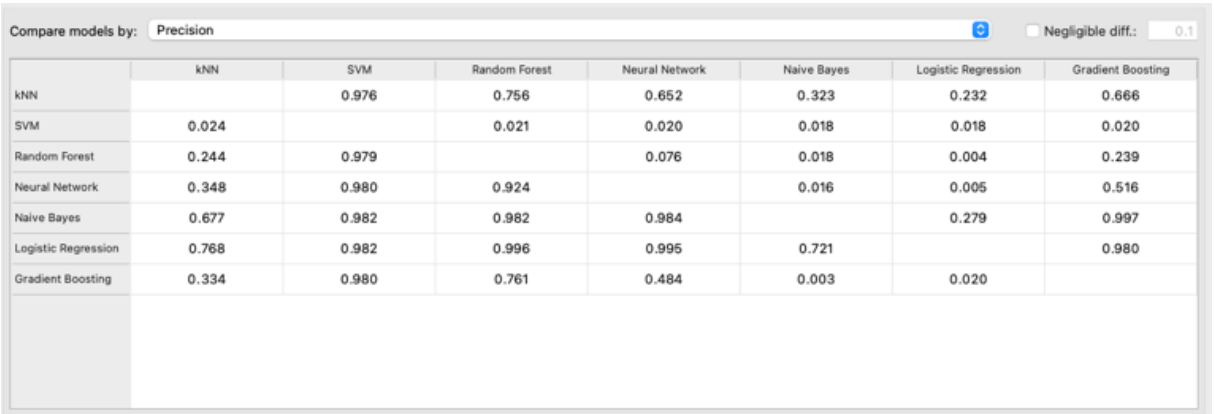
Compare models by: F1 — Negligible diff.: 0.1

	kNN	SVM	Random Forest	Neural Network	Naive Bayes	Logistic Regression	Gradient Boosting
kNN		0.995	0.000	0.000	0.000	0.000	0.000
SVM	0.005		0.000	0.000	0.000	0.000	0.000
Random Forest	1.000	1.000		0.000	0.388	0.000	0.178
Neural Network	1.000	1.000	1.000		1.000	0.885	1.000
Naive Bayes	1.000	1.000	0.612	0.000		0.000	0.275
Logistic Regression	1.000	1.000	1.000	0.115	1.000		1.000
Gradient Boosting	1.000	1.000	0.822	0.000	0.725	0.000	

Fonte: Elaborado pela autora com os dados de pesquisa desde Orange.

A Figura 68 presenta os resultados da comparação dos modelos com base na precisão do modelo para a categoria “Crime”. Nesse caso, o cenário parace mudar um pouco, pois os dados revelam que o modelo de classificação documental kNN é melhor os demais modelos treinados e testados (SVM (97,6%), *Random Forest* (75,6%), Rede Neural (65,2%), *Naive Bayes* (32,2%), Regressão Logística (23,2%) e *Gradient Boosting* (66,6%)). Por outro lado, mas uma vez, o classificador SVM, apresenta os piores resultados comparativos com os outros modelos novamente. Finalmente, cabe destacar que, nesse caso, os modelos *Naive Bayes*, Regressão Logística e *Gradient Boosting* são melhores que o modelo de Rede Neural num quase 100% de precisão.

Figura 68 – Comparação dos modelos baseada na precisão para a categoria “Crime”



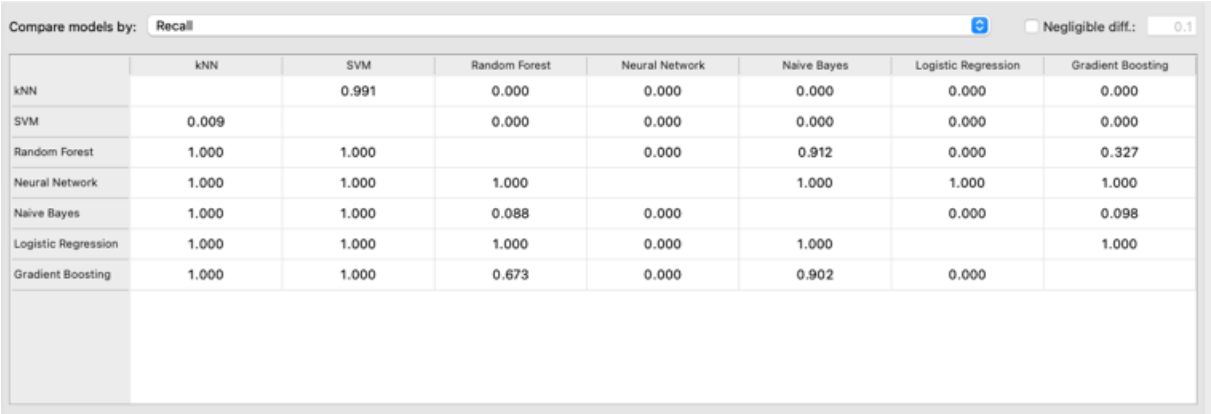
Compare models by: Precision — Negligible diff.: 0.1

	kNN	SVM	Random Forest	Neural Network	Naive Bayes	Logistic Regression	Gradient Boosting
kNN		0.976	0.756	0.652	0.323	0.232	0.666
SVM	0.024		0.021	0.020	0.018	0.018	0.020
Random Forest	0.244	0.979		0.076	0.018	0.004	0.239
Neural Network	0.348	0.980	0.924		0.016	0.005	0.516
Naive Bayes	0.677	0.982	0.982	0.984		0.279	0.997
Logistic Regression	0.768	0.982	0.996	0.995	0.721		0.980
Gradient Boosting	0.334	0.980	0.761	0.484	0.003	0.020	

Fonte: Elaborado pela autora com os dados de pesquisa desde Orange.

A Figura 69 mostra os resultados da comparação dos modelos com base no Recall do modelo para a categoria “Crime”. Os dados exibem o mesmo comportamento que nas figuras anteriores dessa categoria, ou seja, o modelo de classificação documental kNN apenas é melhor que SVM num 99,1%, mas é pior que os demais modelos treinados e testados (*Random Forest*, Rede Neural, *Naive Bayes*, Regressão Logística e *Gradient Boosting*), pois os valores são nulos. O classificador SVM, apresenta os piores resultados comparativos com os outros modelos novamente. Pelo contrario, nenhum modelo (kNN, SVM, *Random Forest*, *Naive Bayes*, Regressão Logística e *Gradient Boosting*) é melhor que o modelo feito com o modelo de Rede Neural, sendo esse último melhor que todos os modelos num 100%.

Figura 69 – Comparação dos modelos baseada no Recall para a categoria “Crime”

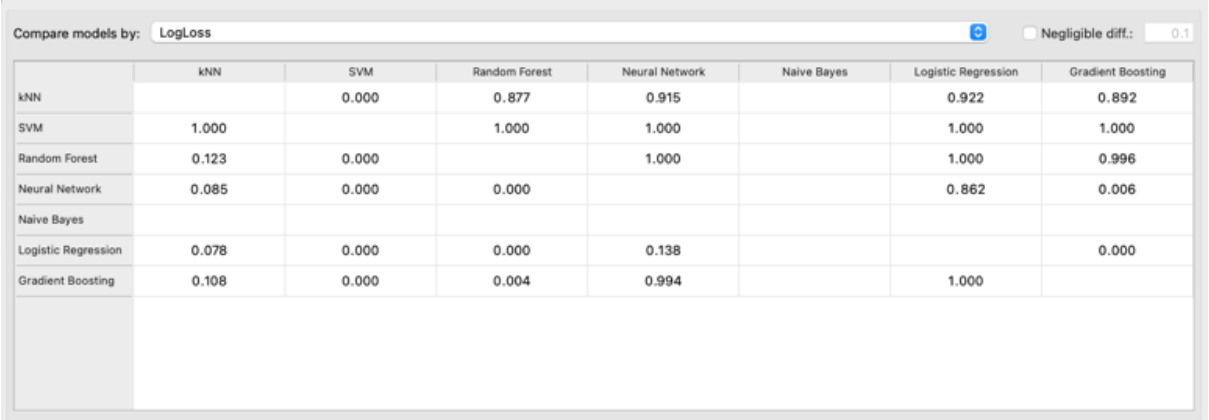


	kNN	SVM	Random Forest	Neural Network	Naive Bayes	Logistic Regression	Gradient Boosting
kNN		0.991	0.000	0.000	0.000	0.000	0.000
SVM	0.009		0.000	0.000	0.000	0.000	0.000
Random Forest	1.000	1.000		0.000	0.912	0.000	0.327
Neural Network	1.000	1.000	1.000		1.000	1.000	1.000
Naive Bayes	1.000	1.000	0.088	0.000		0.000	0.098
Logistic Regression	1.000	1.000	1.000	0.000	1.000		1.000
Gradient Boosting	1.000	1.000	0.673	0.000	0.902	0.000	

Fonte: Elaborado pela autora com os dados de pesquisa desde Orange.

A Figura 70 mostra os resultados da comparação dos modelos com base na LogLoss (entropia cruzada ou medida de incerteza da precisão do modelo) para a categoria “Crime”. Os dados revelam que o modelo de classificação documental kNN apenas é melhor que os classificadores *Random Forest*, Rede Neural, Regressão Logística e *Gradient Boosting* com mais do 87% de probabilidade, mas é pior que kNN. O classificador SVM, dessa vez apresenta excelentes resultados em comparação com os outros modelos novamente (com um 100% de probabilidade de ser melhor). Pelo contrario, os modelos *Random Forest*, Rede Neural e Regressão Logística apesentam os piores resultados, pois na comparação, com algumas exceções, são piores que os outros modelos. *Gradient Boosting* é melhor num Rede Neural (num 99,4%) e que Regressão Logística (num 100%).

Figura 70 – Comparação dos modelos baseada no LogLoss para a categoria “Crime”

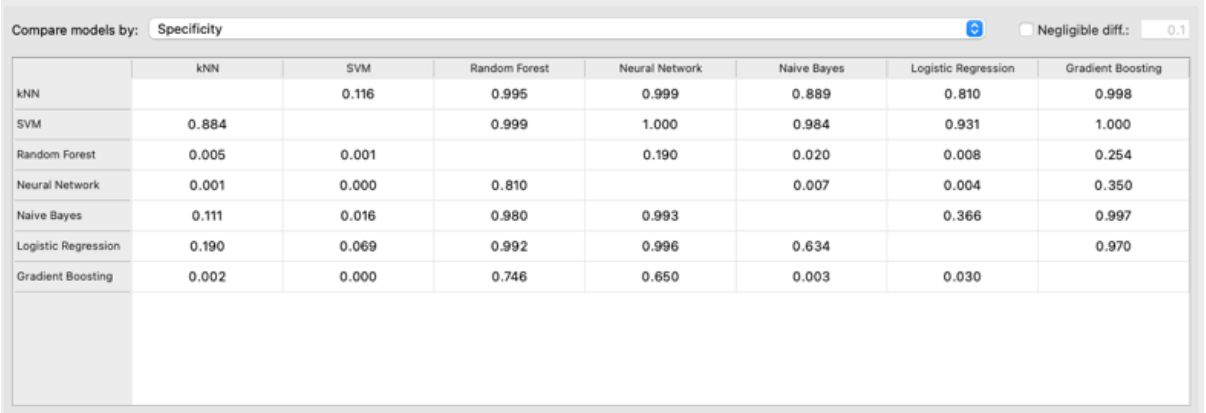


	kNN	SVM	Random Forest	Neural Network	Naive Bayes	Logistic Regression	Gradient Boosting
kNN		0.000	0.877	0.915		0.922	0.892
SVM	1.000		1.000	1.000		1.000	1.000
Random Forest	0.123	0.000		1.000		1.000	0.996
Neural Network	0.085	0.000	0.000			0.862	0.006
Naive Bayes							
Logistic Regression	0.078	0.000	0.000	0.138			0.000
Gradient Boosting	0.108	0.000	0.004	0.994		1.000	

Fonte: Elaborado pela autora com os dados de pesquisa desde Orange.

A Figura 71 mostra os resultados da comparação dos modelos com base na especificidade do modelo para a categoria “Crime”. Assim, como na figura anterior o comportamento é diferente do observado até agora nas outras comparações. Nesse caso os dados revelam que os modelo de classificação documental kNN e SVG apresentam os melhores resultados na comparação como os outros modelos chegando no 100% em muitos casos. Pelo contrario, o pior resultado na comparação é para o classificador *Random Forest* que apresenta valores quase nulos em termos de especificidade.

Figura 71 – Comparação dos modelos baseada na especificidade para a categoria “Crime”



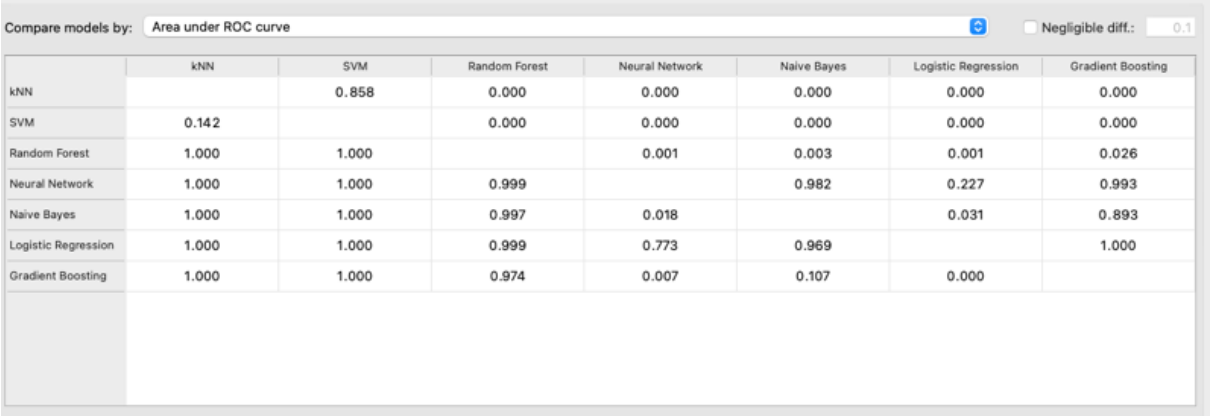
	kNN	SVM	Random Forest	Neural Network	Naive Bayes	Logistic Regression	Gradient Boosting
kNN		0.116	0.995	0.999	0.889	0.810	0.998
SVM	0.884		0.999	1.000	0.984	0.931	1.000
Random Forest	0.005	0.001		0.190	0.020	0.008	0.254
Neural Network	0.001	0.000	0.810		0.007	0.004	0.350
Naive Bayes	0.111	0.016	0.980	0.993		0.366	0.997
Logistic Regression	0.190	0.069	0.992	0.996	0.634		0.970
Gradient Boosting	0.002	0.000	0.746	0.650	0.003	0.030	

Fonte: Elaborado pela autora com os dados de pesquisa desde Orange.

Na Figura 72, 73, 74, 75, 76, 77 e 78, apresenta-se uma comparação dos modelos por pares para a categoria “Politica” desde as diferentes métricas estatísticas acima descritas (AUC, Acurácia da Classificação, Valores F1, precisão, Recall, LogLoss e especificidades) do Experimento1 da fase de Modelagem. Como comentado anteriormente, os valores que aparecem nas figuras representam a probabilidade de que o modelo correspondente à linha seja melhor que o modelo correspondente à coluna.

Na Figura 72 é possível ver que a probabilidade de que SVM sejam melhor que os outros modelos para a categoria “Politica” é de 0,00 (0%), ou seja, é nula. Porem, a probabilidade de que a Rede Neural seja melhor que kNN, SVM e *Randon Forest* é um (100%) e a probabilidade de que Rede Neural seja melhor que Naive Bayes é do 98,2% e que *Grandient Boosting* é do 99,3% com base na área sob a curva ROC. Os poires resultados na comparação entre modelos são, novamente, para os modelos kNN e SVM que apresenta, pelo geral, valores nulos. Pelo contrario, o melhor modelo comparado com os outros classificadores é a Regressão Logística na área sob a curva ROC.

Figura 72 – Comparação dos modelos baseada na área sob a curva ROC (AUC) para a categoria “Política”

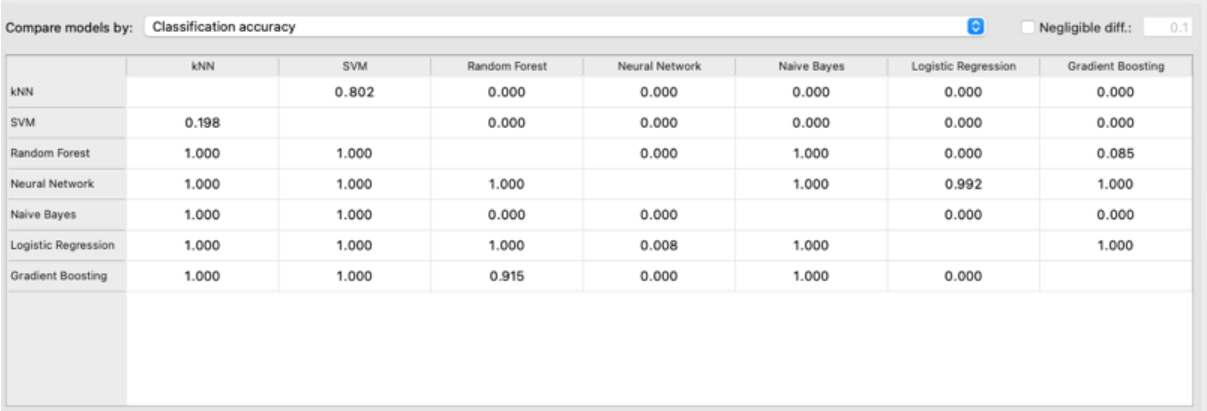


	kNN	SVM	Random Forest	Neural Network	Naive Bayes	Logistic Regression	Gradient Boosting
kNN		0.858	0.000	0.000	0.000	0.000	0.000
SVM	0.142		0.000	0.000	0.000	0.000	0.000
Random Forest	1.000	1.000		0.001	0.003	0.001	0.026
Neural Network	1.000	1.000	0.999		0.982	0.227	0.993
Naive Bayes	1.000	1.000	0.997	0.018		0.031	0.893
Logistic Regression	1.000	1.000	0.999	0.773	0.969		1.000
Gradient Boosting	1.000	1.000	0.974	0.007	0.107	0.000	

Fonte: Elaborado pela autora com os dados de pesquisa desde Orange.

A Figura 73 mostra os resultados da comparação dos modelos com base na acurácia da classificação do modelo para a categoria “Política”. Os dados revelam que o modelo de classificação documental kNN apenas é melhor que SVM num 80,2% (0,802), mas é pior que os demais modelos treinados e testados (*Random Forest*, Rede Neural, *Naive Bayes*, Regressão Logística e *Gradient Boosting*), pois os valores são nulos (0%). O classificador SVM apresenta os piores resultados comparativos com os outros modelos novamente. Pelo contrario, nenhum modelo (kNN, SVM, *Random Forest*, *Naive Bayes*, Regressão Logística e *Gradient Boosting*) é melhor que o modelo feito com o modelo de Rede Neural, sendo esse último melhor que todos os modelos num 100% em acurácia de classificação. Porém, os classificadores *Random Forest*, *Naive Bayes*, Regressão Logística e *Gradient Boosting*, pelo geral são melhores que kNN e SVM num 100% em termos de acurácia.

Figura 73 – Comparação dos modelos baseada na acurácia da classificação (CA) para a categoria “Política”

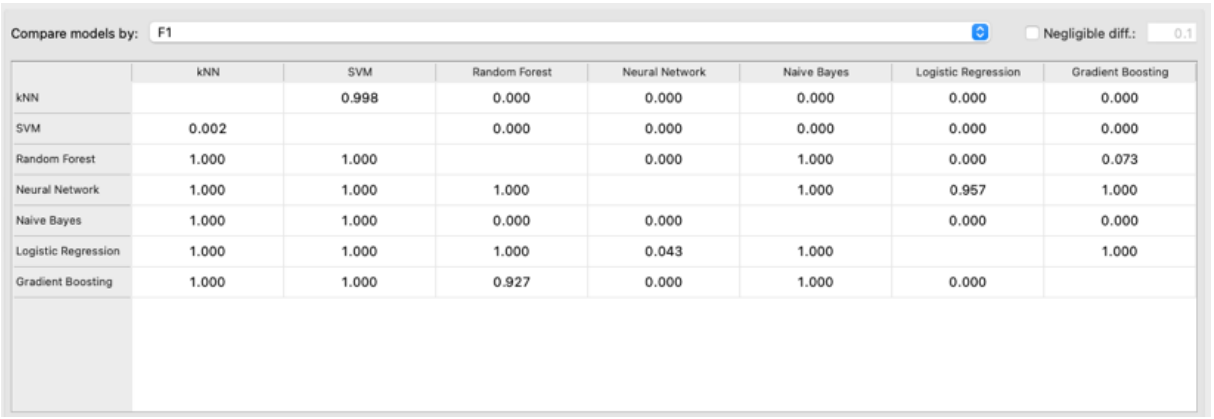


	kNN	SVM	Random Forest	Neural Network	Naive Bayes	Logistic Regression	Gradient Boosting
kNN		0.802	0.000	0.000	0.000	0.000	0.000
SVM	0.198		0.000	0.000	0.000	0.000	0.000
Random Forest	1.000	1.000		0.000	1.000	0.000	0.085
Neural Network	1.000	1.000	1.000		1.000	0.992	1.000
Naive Bayes	1.000	1.000	0.000	0.000		0.000	0.000
Logistic Regression	1.000	1.000	1.000	0.008	1.000		1.000
Gradient Boosting	1.000	1.000	0.915	0.000	1.000	0.000	

Fonte: Elaborado pela autora com os dados de pesquisa desde Orange.

A Figura 74 mostra os resultados da comparação dos modelos com base nos valores F1 do modelo para a categoria “Política”. Os dados revelam que o modelo de classificação documental kNN apenas é melhor que SVM num 99,8%, mas é pior que os demais modelos (*Random Forest*, Rede Neural, *Naive Bayes*, Regressão Logística e *Gradient Boosting*), pois os valores são nulos (0%). O classificador SVM apresenta os piores resultados comparativos com os outros modelos novamente. Pelo contrario, nenhum classificador (kNN, SVM, *Random Forest*, *Naive Bayes*, Regressão Logística e *Gradient Boosting*) é melhor que o modelo de Rede Neural, sendo esse último melhor que todos os modelos num 100% no referente aos valores F1.

Figura 74 – Comparação dos modelos baseada nos valores F1 para a categoria “Política”



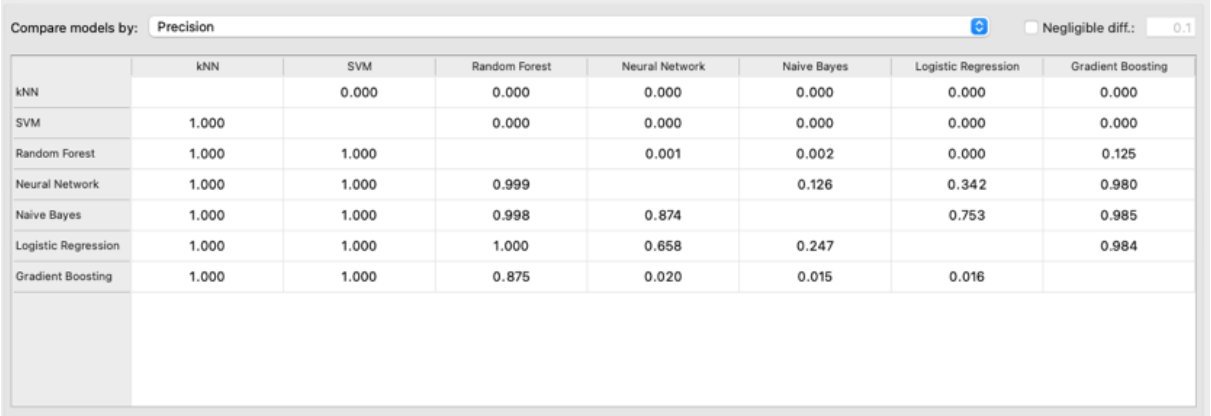
	kNN	SVM	Random Forest	Neural Network	Naive Bayes	Logistic Regression	Gradient Boosting
kNN		0.998	0.000	0.000	0.000	0.000	0.000
SVM	0.002		0.000	0.000	0.000	0.000	0.000
Random Forest	1.000	1.000		0.000	1.000	0.000	0.073
Neural Network	1.000	1.000	1.000		1.000	0.957	1.000
Naive Bayes	1.000	1.000	0.000	0.000		0.000	0.000
Logistic Regression	1.000	1.000	1.000	0.043	1.000		1.000
Gradient Boosting	1.000	1.000	0.927	0.000	1.000	0.000	

Fonte: Elaborado pela autora com os dados de pesquisa desde Orange.

A Figura 75 mostra os resultados da comparação dos modelos com base na precisão do modelo para a categoria “Política”. Os dados demonstram que o modelo SVM apenas é melhor que kNN num 100%, mas é pior que os demais modelos treinados e testados (*Random Forest*, Rede Neural, *Naive Bayes*, Regressão Logística e *Gradient Boosting*), pois os valores são nulos (0%). Nesse caso, o classificador kNN é o que apresenta os piores resultados comparativos com

os outros modelos novamente. Pelo contrario, os modelos SVM, *Random Forest* e *Gradient Boosting*) são piores que o modelo de Rede Neural, sendo esse último melhor que quase todos os modelos num 100% em termos de precisão, com exceção do modelo *Grandient Boosting*.

Figura 75 – Comparação dos modelos baseada na precisão para a categoria “Política”

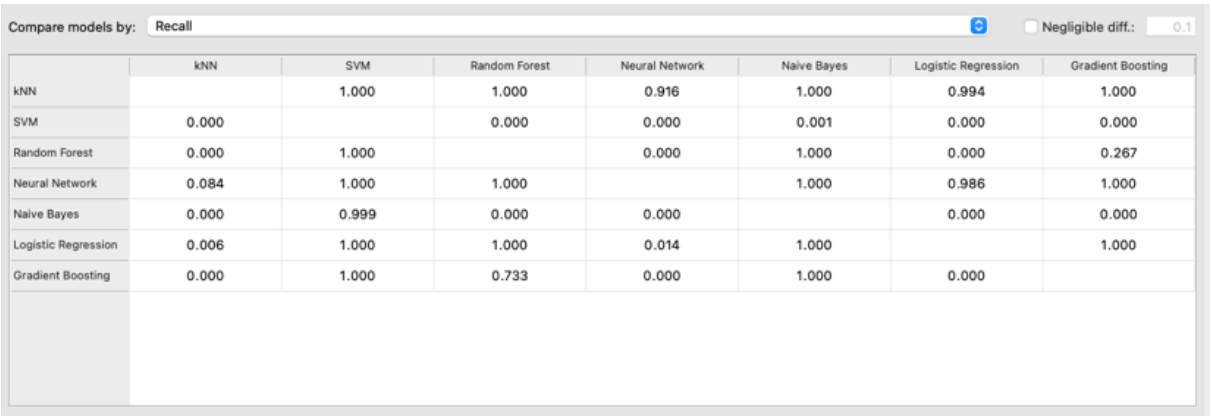


	kNN	SVM	Random Forest	Neural Network	Naive Bayes	Logistic Regression	Gradient Boosting
kNN		0.000	0.000	0.000	0.000	0.000	0.000
SVM	1.000		0.000	0.000	0.000	0.000	0.000
Random Forest	1.000	1.000		0.001	0.002	0.000	0.125
Neural Network	1.000	1.000	0.999		0.126	0.342	0.980
Naive Bayes	1.000	1.000	0.998	0.874		0.753	0.985
Logistic Regression	1.000	1.000	1.000	0.658	0.247		0.984
Gradient Boosting	1.000	1.000	0.875	0.020	0.015	0.016	

Fonte: Elaborado pela autora com os dados de pesquisa desde Orange.

A Figura 76 mostra os resultados da comparação dos modelos com base no Recall do modelo para a categoria “Política”. Como na figura anterior o comportamento é diferente do observado geralmente nas outras comparações. Nesse caso, os dados mostram que os modelo de classificação documental kNN e a Rede Neural apresentam os melhores resultados na comparação como os outros modelos chegando no 100% na maioria dos casos. Pelo contrario, o pior resultado na comparação é para o classificador SVM que apresenta valores quase nulos em termos de revocação.

Figura 76 – Comparação dos modelos baseada no Recall para a categoria “Política”



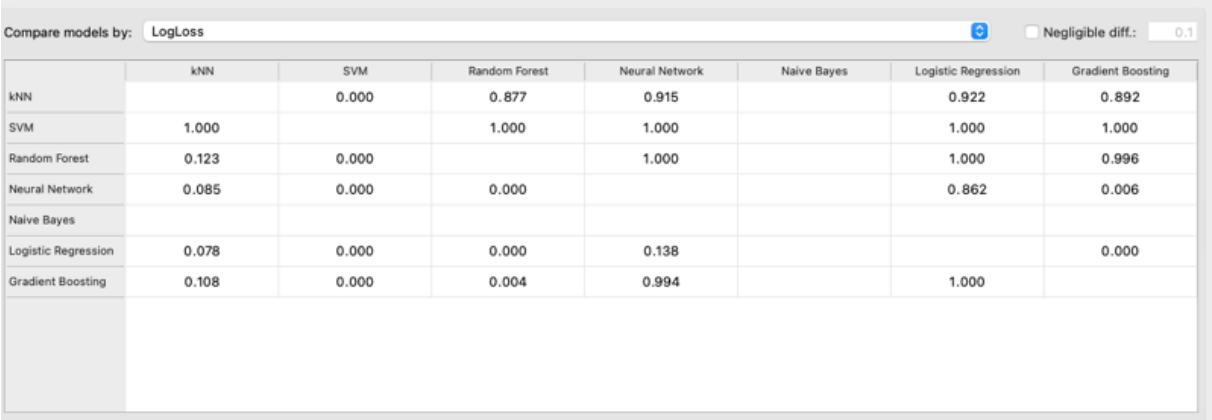
	kNN	SVM	Random Forest	Neural Network	Naive Bayes	Logistic Regression	Gradient Boosting
kNN		1.000	1.000	0.916	1.000	0.994	1.000
SVM	0.000		0.000	0.000	0.001	0.000	0.000
Random Forest	0.000	1.000		0.000	1.000	0.000	0.267
Neural Network	0.084	1.000	1.000		1.000	0.986	1.000
Naive Bayes	0.000	0.999	0.000	0.000		0.000	0.000
Logistic Regression	0.006	1.000	1.000	0.014	1.000		1.000
Gradient Boosting	0.000	1.000	0.733	0.000	1.000	0.000	

Fonte: Elaborado pela autora com os dados de pesquisa desde Orange.

A Figura 77 mostra os resultados da comparação dos modelos com base no LogLoss do modelo para a categoria “Política”. Assim, como na figura anterior o comportamento é diferente do observado até agora nas outras comparações. Nesse caso os dados revelam que os modelo

de classificação documental kNN e SVG apresentam os melhores resultados na comparação como os outros modelos chegando no 100% em muitos casos. Pelo contrario, o pior resultado na comparação é para o modelo de Rede Neural e de Regressão Logística que apresenta valores quase nulos na maioria dos casos em termos de LogLoss (entropia cruzada ou incerteza da precisão do modelo).

Figura 77 – Comparação dos modelos baseado no LogLoss para a categoria “Política”

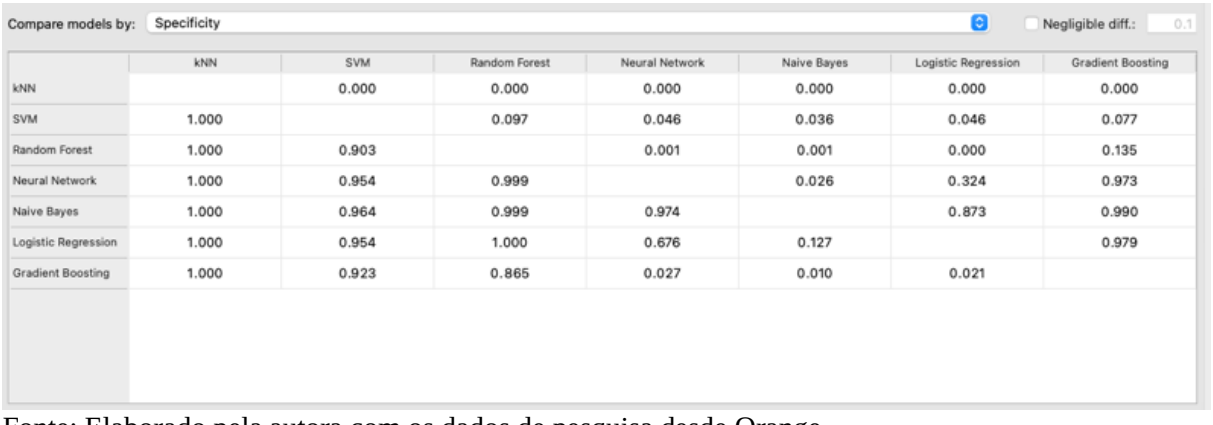


	kNN	SVM	Random Forest	Neural Network	Naive Bayes	Logistic Regression	Gradient Boosting
kNN		0.000	0.877	0.915		0.922	0.892
SVM	1.000		1.000	1.000		1.000	1.000
Random Forest	0.123	0.000		1.000		1.000	0.996
Neural Network	0.085	0.000	0.000			0.862	0.006
Naive Bayes							
Logistic Regression	0.078	0.000	0.000	0.138			0.000
Gradient Boosting	0.108	0.000	0.004	0.994		1.000	

Fonte: Elaborado pela autora com os dados de pesquisa desde Orange.

A Figura 78 mostra os resultados da comparação dos modelos com base na especificidade do modelo para a categoria “Política”. Os dados mostram que os piores classificadores são kNN e SVM com valores nulos. Pelo contrario, todos os modelos são melhores que kNN, SVM e *Random Forest*, na maioria dos casos em termos de especificidade.

Figura 78 – Comparação dos modelos baseada na especificidade para a categoria “Política”



	kNN	SVM	Random Forest	Neural Network	Naive Bayes	Logistic Regression	Gradient Boosting
kNN		0.000	0.000	0.000	0.000	0.000	0.000
SVM	1.000		0.097	0.046	0.036	0.046	0.077
Random Forest	1.000	0.903		0.001	0.001	0.000	0.135
Neural Network	1.000	0.954	0.999		0.026	0.324	0.973
Naive Bayes	1.000	0.964	0.999	0.974		0.873	0.990
Logistic Regression	1.000	0.954	1.000	0.676	0.127		0.979
Gradient Boosting	1.000	0.923	0.865	0.027	0.010	0.021	

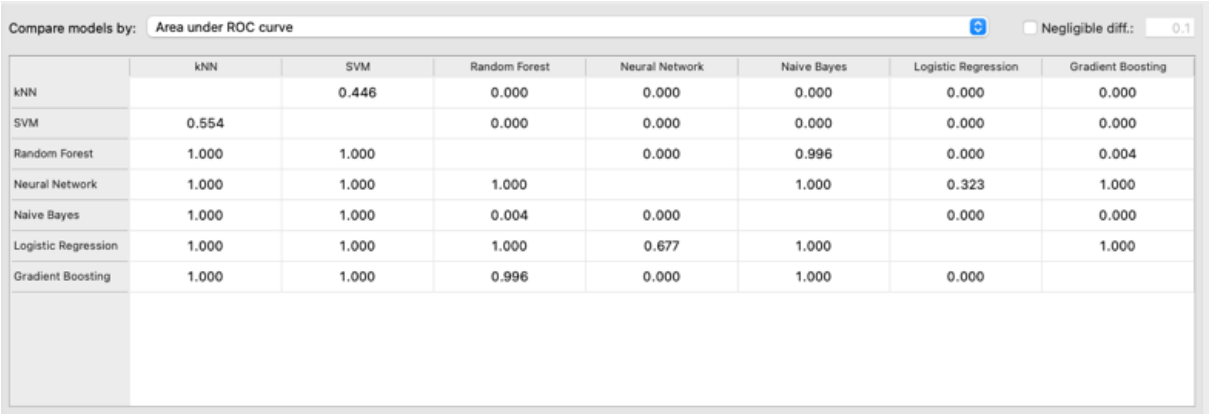
Fonte: Elaborado pela autora com os dados de pesquisa desde Orange.

Na Figura 79, 80, 81, 82, 83, 84 e 85, se apresenta uma comparação dos modelos por pares para a categoria “Ciência” desde as diferentes métricas estatísticas acima descritas (AUC, Acurácia da Classificação, Valores F1, precisão, Recall, LogLoss e especificidades) do Experimento1 da fase de Modelagem. Como comentado anteriormente, os valores que

aparecem nas figuras representam a probabilidade de que o modelo correspondente à linha seja melhor que o modelo correspondente à coluna.

Na Figura 79, é possível ver que a probabilidade de que SVM sejam melhor que os outros modelos para a categoria “Ciência” é nula (0%) em quase todos os casos, com exeçao do kNN que é melhor num 55,4% na área sob a curva ROC. Porem, a probabilidade de que a Rede Neural e a Regressão Logística seja melhor que os outros modelos é um 100% em quase todos os casos.

Figura 79 – Comparação dos modelos baseada na área sob a curva ROC (AUC) para a categoria “Ciência”

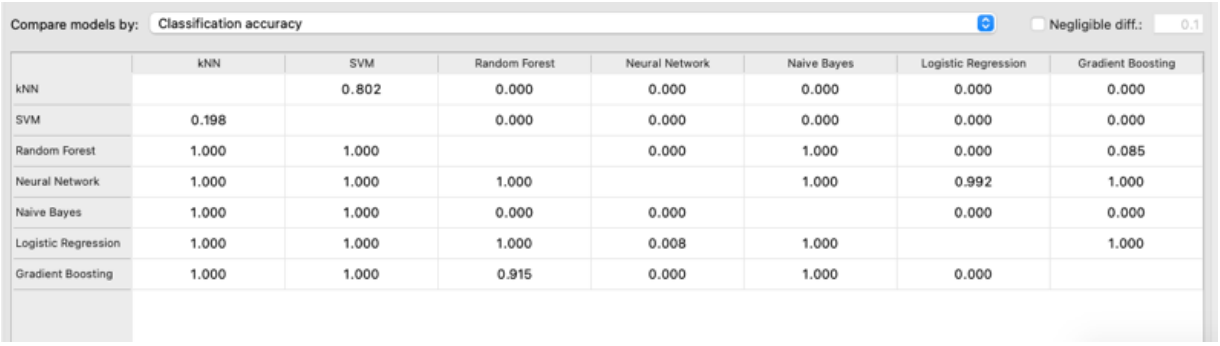


	kNN	SVM	Random Forest	Neural Network	Naive Bayes	Logistic Regression	Gradient Boosting
kNN		0.446	0.000	0.000	0.000	0.000	0.000
SVM	0.554		0.000	0.000	0.000	0.000	0.000
Random Forest	1.000	1.000		0.000	0.996	0.000	0.004
Neural Network	1.000	1.000	1.000		1.000	0.323	1.000
Naive Bayes	1.000	1.000	0.004	0.000		0.000	0.000
Logistic Regression	1.000	1.000	1.000	0.677	1.000		1.000
Gradient Boosting	1.000	1.000	0.996	0.000	1.000	0.000	

Fonte: Elaborado pela autora com os dados de pesquisa desde Orange.

A Figura 80 mostra os resultados da comparação dos modelos com base na acurácia da classificação do modelo para a categoria “Ciência”. Os dados revelam que o modelo de classificação documental kNN apenas é melhor que SVM num 80,2% (0,802), mas é pior que os demais modelos treinados e testados (*Random Forest*, Rede Neural, *Naive Bayes*, Regressão Logística e *Gradient Boosting*), pois os valores são 0,000 (0%). O classificador SVM, que apresenta os piores resultados comparativos com os outros modelos novamente. Pelo contrario, nenhum modelo (kNN, SVM, *Random Forest*, *Naive Bayes*, Regressão Logística e *Gradient Boosting*) é melhor que o modelo feito com o modelo de Rede Neural, sendo esse último melhor que todos os modelos num 100% de acurácia de classificação em quase todos os casos.

Figura 80 – Comparação dos modelos baseada na acurácia da classificação para a categoria “Ciência”



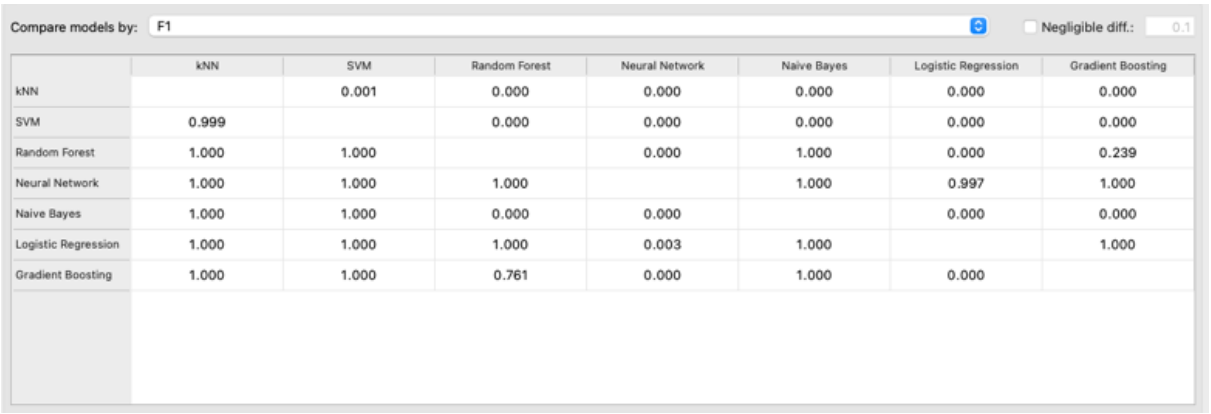
Compare models by: Classification accuracy — Negligible diff.: 0.1

	kNN	SVM	Random Forest	Neural Network	Naive Bayes	Logistic Regression	Gradient Boosting
kNN		0.802	0.000	0.000	0.000	0.000	0.000
SVM	0.198		0.000	0.000	0.000	0.000	0.000
Random Forest	1.000	1.000		0.000	1.000	0.000	0.085
Neural Network	1.000	1.000	1.000		1.000	0.992	1.000
Naive Bayes	1.000	1.000	0.000	0.000		0.000	0.000
Logistic Regression	1.000	1.000	1.000	0.008	1.000		1.000
Gradient Boosting	1.000	1.000	0.915	0.000	1.000	0.000	

Fonte: Elaborado pela autora com os dados de pesquisa desde Orange.

A Figura 81 mostra os resultados da comparação dos modelos com base nos valores F1 do modelo para a categoria “Ciência”. Os dados revelam que o modelo de classificação documental SVM apenas é melhor que kNN num 99,9%, mas é pior que os demais modelos treinados e testados (*Random Forest*, Rede Neural, *Naive Bayes*, Regressão Logística e *Gradient Boosting*), pois os valores são nulos (0%). O classificador kNN, apresenta os piores resultados comparativos com os outros modelos. Pelo contrario, nenhum modelo (kNN, SVM, *Random Forest*, *Naive Bayes*, Regressão Logística e *Gradient Boosting*) é melhor que o modelo feito com o modelo de Rede Neural, sendo esse último melhor que todos os modelos num 100% em quase todos os casos.

Figura 81 – Comparação dos modelos baseada nos valores F1 para a categoria “Ciência”



Compare models by: F1 — Negligible diff.: 0.1

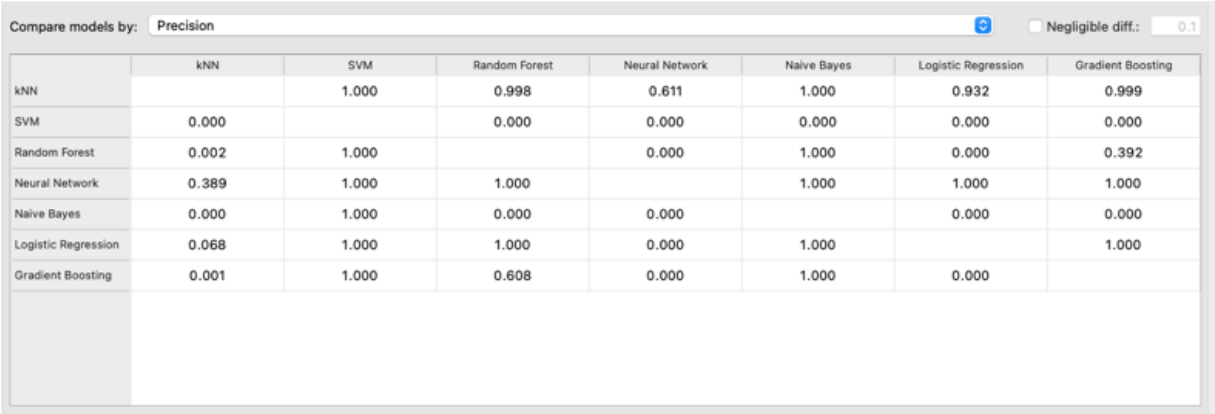
	kNN	SVM	Random Forest	Neural Network	Naive Bayes	Logistic Regression	Gradient Boosting
kNN		0.001	0.000	0.000	0.000	0.000	0.000
SVM	0.999		0.000	0.000	0.000	0.000	0.000
Random Forest	1.000	1.000		0.000	1.000	0.000	0.239
Neural Network	1.000	1.000	1.000		1.000	0.997	1.000
Naive Bayes	1.000	1.000	0.000	0.000		0.000	0.000
Logistic Regression	1.000	1.000	1.000	0.003	1.000		1.000
Gradient Boosting	1.000	1.000	0.761	0.000	1.000	0.000	

Fonte: Elaborado pela autora com os dados de pesquisa desde Orange.

A Figura 82 mostra os resultados da comparação dos modelos com base na precisão do modelo para a categoria “Ciência”. Assim, como em alguns caos anteriores, o comportamento é diferente do observado até agora nas outras comparações. Nesse caso os dados revelam que os modelo de classificação documental kNN e SVG apresentam os melhores resultados na comparação como os outros modelos chegando no 100% em muitos casos. Pelo contrario, o

pior resultado na comparação é para o classificador *Naive Bayes* que apresenta valores nulos em todos os casos, exceto na comparação com o classificador SVM, em termos de precisão.

Figura 82 – Comparação dos modelos baseada na precisão para a categoria “Ciência”

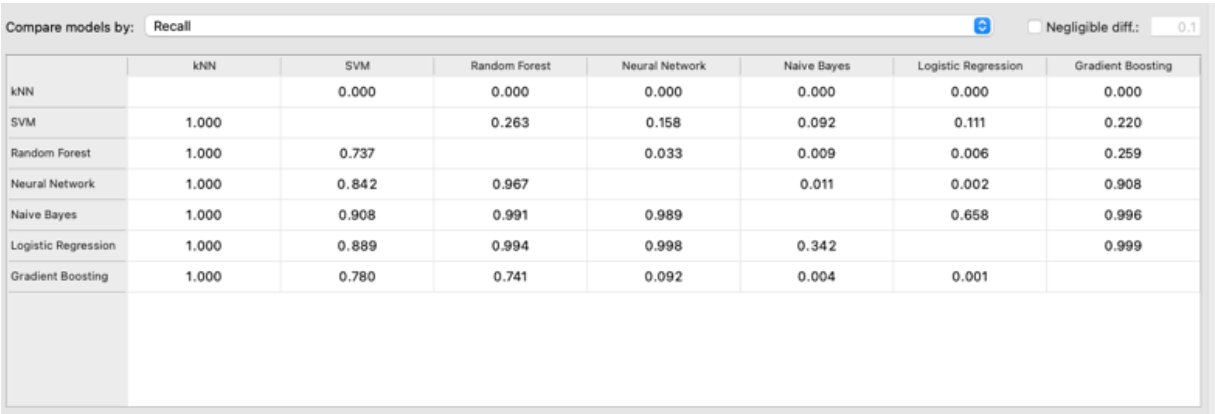


	kNN	SVM	Random Forest	Neural Network	Naive Bayes	Logistic Regression	Gradient Boosting
kNN		1.000	0.998	0.611	1.000	0.932	0.999
SVM	0.000		0.000	0.000	0.000	0.000	0.000
Random Forest	0.002	1.000		0.000	1.000	0.000	0.392
Neural Network	0.389	1.000	1.000		1.000	1.000	1.000
Naive Bayes	0.000	1.000	0.000	0.000		0.000	0.000
Logistic Regression	0.068	1.000	1.000	0.000	1.000		1.000
Gradient Boosting	0.001	1.000	0.608	0.000	1.000	0.000	

Fonte: Elaborado pela autora com os dados de pesquisa desde Orange.

A Figura 83 mostra os resultados da comparação dos modelos com base na revocação (Recall) do modelo para a categoria “Ciência”. Os dados mostram que o modelo de classificação documental kNN apresenta os piores resultados comparativos com os outros modelos com valores nulos. O classificador SVM, apesenta valores poco significativos. Pelo contrario, os outros modelos *Random Forest*, *Naive Bayes*, Regressão Logística e *Gradient Boosting*) apresentan resultados más significativos em comparação com outros modelos em quase todos os casos.

Figura 83 – Comparação dos modelos baseada no Recall para a categoria “Ciência”



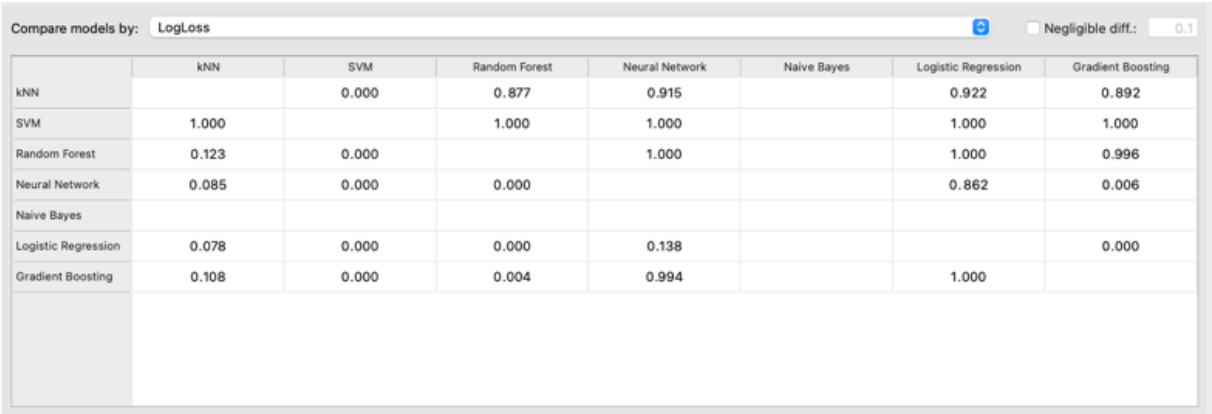
	kNN	SVM	Random Forest	Neural Network	Naive Bayes	Logistic Regression	Gradient Boosting
kNN		0.000	0.000	0.000	0.000	0.000	0.000
SVM	1.000		0.263	0.158	0.092	0.111	0.220
Random Forest	1.000	0.737		0.033	0.009	0.006	0.259
Neural Network	1.000	0.842	0.967		0.011	0.002	0.908
Naive Bayes	1.000	0.908	0.991	0.989		0.658	0.996
Logistic Regression	1.000	0.889	0.994	0.998	0.342		0.999
Gradient Boosting	1.000	0.780	0.741	0.092	0.004	0.001	

Fonte: Elaborado pela autora com os dados de pesquisa desde Orange.

A Figura 84 mostra os resultados da comparação dos modelos com base no LogLoss (entropia cruzada ou medida de incerteza da precisão do model) para a categoria “Ciência”. Nesse caso os dados revelam que os modelo de classificação documental kNN e SVG apresentam os melhores resultados na comparação como os outros modelos chegando no 100%

em muitos casos. Pelo contrario, o pior resultado na comparação é para à Regressão Logística, que apresenta valores nulos em quase todos os casos no que tange à incerteza da precisão.

Figura 84 – Comparação dos modelos baseada no LogLoss para a categoria “Ciência”

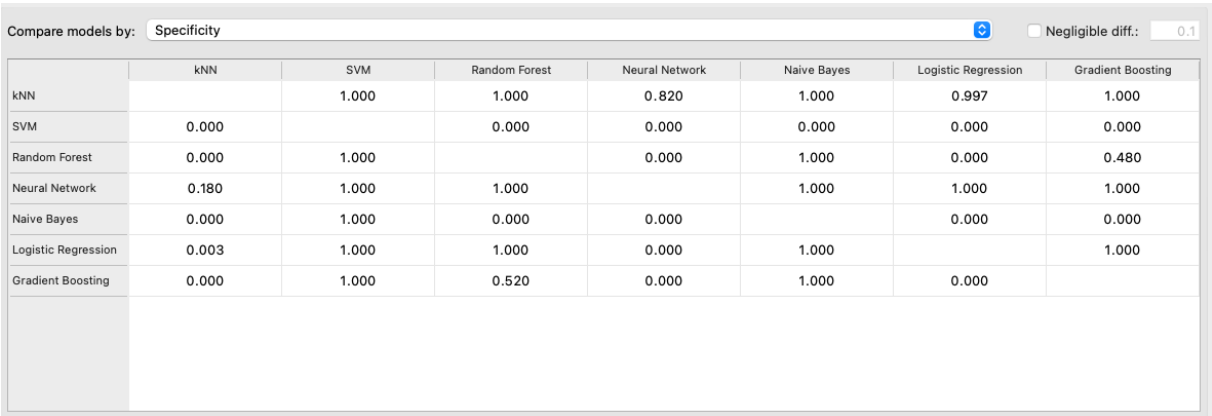


	kNN	SVM	Random Forest	Neural Network	Naive Bayes	Logistic Regression	Gradient Boosting
kNN		0.000	0.877	0.915		0.922	0.892
SVM	1.000		1.000	1.000		1.000	1.000
Random Forest	0.123	0.000		1.000		1.000	0.996
Neural Network	0.085	0.000	0.000			0.862	0.006
Naive Bayes							
Logistic Regression	0.078	0.000	0.000	0.138			0.000
Gradient Boosting	0.108	0.000	0.004	0.994		1.000	

Fonte: Elaborado pela autora com os dados de pesquisa desde Orange.

A Figura 85 mostra os resultados da comparação dos modelos com base na especificidades do modelo para a categoria “Ciência”. Assim, como na figura anterior o comportamento é diferente do observado até agora nas outras comparações. Nesse caso os dados revelam que os modelo de classificação documental kNN e Rede Neural apresentam os melhores resultados na comparação como os outros modelos chegando no 100% em muitos casos. Pelo contrario, o pior resultado na comparação é para o classificador SVM que apresenta valores nulos em termos de especificidade em todos os casos.

Figura 85 – Comparação dos modelos baseada na especificidade para a categoria “Ciência”



	kNN	SVM	Random Forest	Neural Network	Naive Bayes	Logistic Regression	Gradient Boosting
kNN		1.000	1.000	0.820	1.000	0.997	1.000
SVM	0.000		0.000	0.000	0.000	0.000	0.000
Random Forest	0.000	1.000		0.000	1.000	0.000	0.480
Neural Network	0.180	1.000	1.000		1.000	1.000	1.000
Naive Bayes	0.000	1.000	0.000	0.000		0.000	0.000
Logistic Regression	0.003	1.000	1.000	0.000	1.000		1.000
Gradient Boosting	0.000	1.000	0.520	0.000	1.000	0.000	

Fonte: Elaborado pela autora com os dados de pesquisa desde Orange.

Para sintetizar, cabe destacar que na maioria dos cálculos estatísticos os resultados demonstram que os melhores algoritmo ou modelo de classificação documental são Rede Neural e Regressão Logística, salvo algumas exceções já mencionadas.

Com o intuito de avaliar os modelos ou algoritmos de classificação, foi realizada a validação cruzada *k-fold* e as matrizes de confusão. Assim, após executar a validação cruzada na fase de teste e treinamento dos modelos (Teste & Score) para obter previsões de classe pelos algoritmos escolhidos para todas as instâncias dos dados (corpus documental), os resultados obtidos com os testes foram inseridos nas matrizes de confusão, onde podemos observar quantas instâncias foram classificadas incorretamente e de que maneira, a proporção de predição e proporção atual das categorias e algoritmos de classificação. Por tanto, a matriz de confusão é uma tabela que permite avaliar e comparar os resultados gerados pelos modelos de classificação frente (versus) a classificação real dos dados objeto de estudo. Nesse ponto, cabe esclarecer o seguinte:

- O **número de instâncias** mostra as instâncias classificadas de forma correta e incorreta numericamente.
- As **proporções da predição** mostram quantas instâncias classificadas como, "Crime" estão em uma classe verdadeira. Por exemplo, na Figura 83b se pode ler que 27,7% dos documentos da categoria "Crime" não pertencem realmente à categoria "Política", 100% daqueles classificados como "Crime" pertencem à categoria "Crime" e 38,8% não pertencem à categoria "Ciência", pelo que está classificado erroneamente.
- **Proporções atual ou real** mostram a relação oposta. Por exemplo, a Figura 85c mostra que de todos os documentos verdadeiramente classificados dentro da categoria "Crime", apenas o 2,6% foram corretamente classificados dentro da categoria "Crime", 13,7% foi erroneamente classificado na categoria "Política" e 83,7% foi erroneamente classificado dentro da categoria "Ciência".

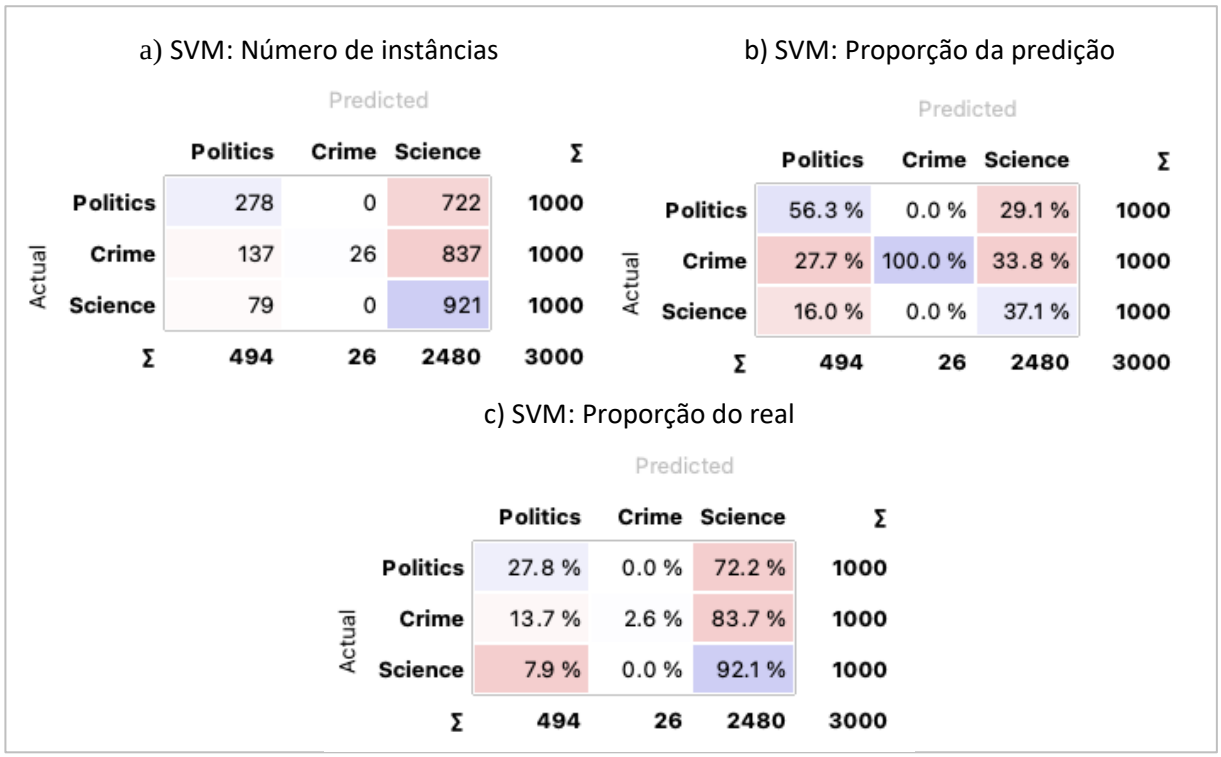
As cores das matrizes de confusão facilitam a compreensão dos resultados fornecidos pela mesma, de forma que a cor **rosa escuro e claro** indica a proporção ou número de instâncias o registros classificadas incorretamente dentro de uma categoria ou classe (ou seja, indica o número ou proporção de erros na classificação feita pelo algoritmo), a cor **violeta escuro ou clara** que indica a proporção ou número de instâncias corretamente classificada dentro dessa classe (ou seja, indica o número ou proporção de acertos na classificação feita pelo algoritmo) e a cor **branca** que indica valores nulos na classificação (ou seja, indica que não existem erros né acertos no número na classificação feita pelo algoritmo para essa categoria ou classe).

Assim, após analisar as tabelas e figuras anteriores, foi gerada a matriz de confusão que mostra as proporções entre a classe prevista e a real. Por conseguinte, os dados de entrada foram

os resultados da avaliação, ou seja, os resultados dos algoritmos de classificação da fase de teste e treinamento dos modelos. Como saída, obtemos a Matriz de Confusão com o número/proporção de instâncias entre a classe prevista e a classe real. Na matriz, cada linha corresponde a uma classe correta, enquanto as colunas representam as classes previstas. Por exemplo, na Figura 83a, podem-se observar 722 (72,2%) instâncias da categoria “Política” que foram classificadas erroneamente dentro da categoria “Ciência”. A coluna mais à direita fornece o número de instâncias reais de cada classe (por exemplo há 1000 instâncias de cada uma das três classes ou categorias) e a linha inferior fornece o número de instâncias classificadas em cada classe (por exemplo, 494 instâncias foram classificadas em “Política”).

Conseguintemente, nas figuras a seguir, encontramos as matrizes de confusão nas que foram adicionadas o número de instâncias, as proporções da predição e a proporção real dos algoritmos de classificação. Na Figura 86, podem-se observar as matrizes de confusão com a predição do algoritmo de classificação SVM. As matrizes mostram que existe um total de 3000 documentos ou instâncias, com uma distribuição real, atual e inicial de 1000 documentos para cada categoria, porem o classificador SVM prediz uma distribuição diferente para cada categoria, sendo essa de 494 documentos (e-mails) classificados dentro da categoria “Política”, 26 documentos incluídos dentro da categoria “Crime” e 2480 documentos incluídos dentro da categoria “Ciência”.

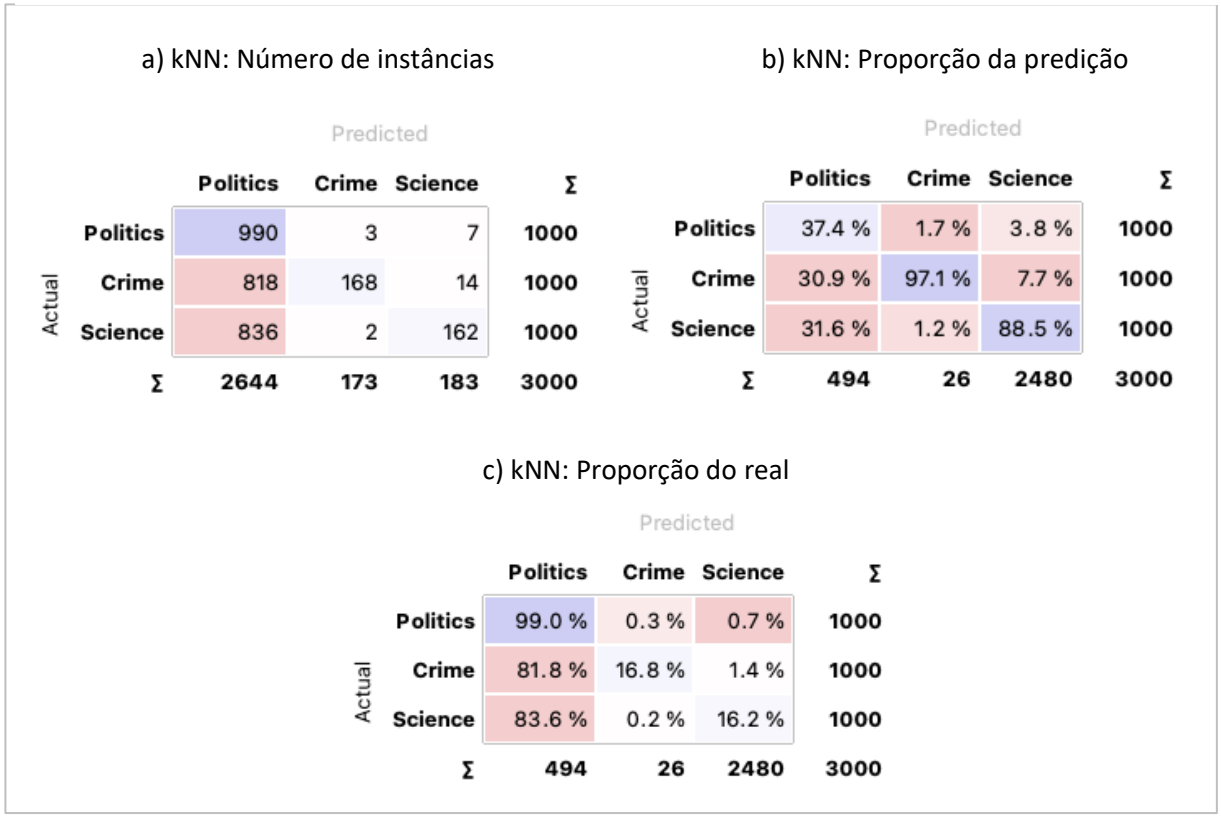
Figura 86 – Matrizes de confusão com a predição do algoritmo de classificação SVM



Fonte: Elaborado pela autora com os dados de pesquisa.

Na Figura 87, pode-se observar as matrizes de confusão com a predição do algoritmo de classificação kNN. A Figura 87a mostra o número de instancias ou documentos e a Figura 87c mostra a proporção real da classificação categórica dos documentos mostrando a porcentagem das instâncias ou documentos pertencentes a uma categoria. Já a Figura 87b apresenta a proporção da predição feita pelo algoritmo kNN para cada categoria. Nas matrizes com o número de instâncias e proporção real, pode-se ler que 99% dos documentos foram corretamente classificados dentro da categoria "Política", 0,3% (3) foi classificado erroneamente na categoria "Crime" e 0,7% (7) dentro da categoria "Ciência". Apenas 16,8% dos documentos foram bem classificados pelo kNN dentro da categoria "Crime", 81,8% dos documentos foram classificados erroneamente dentro da categoria "Política" e 1,4% foi erroneamente classificado dentro da categoria "Ciência. Dentro a categoria "Ciência", apenas o 16,2% dos documentos foram corretamente classificados na categoria "Ciência" e os demais (83,8%) foram mal classificados dentro das categoria "Política" e "Crime". A Figura 87b, mostra que o algoritmo kNN acertou na sua predição de classificação documental dentro da categoria "Política" num 37,4%, num 97,1% na classificação dos documentos dentro da categoria "Crime" e num 88,5% dos documentos classificados dentro da categoria "Ciência". Os demais documentos não pertencem realmente à categoria ás outras categorias.

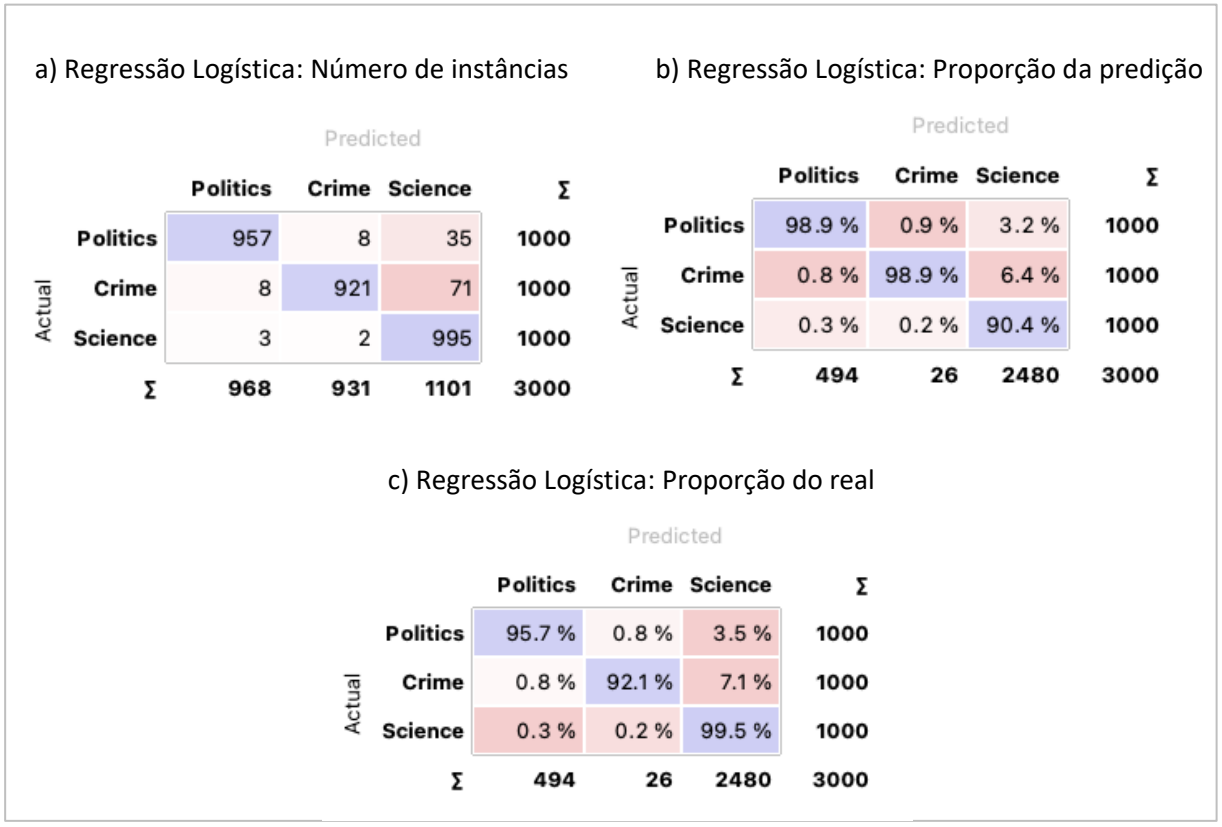
Figura 87 – Matrizes de confusão com a predição do algoritmo de classificação kNN



Fonte: Elaborado pela autora com os dados de pesquisa.

Na Figura 88, pode-se observar as matrizes de confusão com a predição do algoritmo de classificação Regressão Logística. A Figura 88a revela o número de instancias e a Figura 88c mostra a proporção real da classificação categórica dos documentos mostrando a porcentagem das instâncias ou documentos pertencentes a uma categoria. Já a Figura 88b apresenta a proporção da predição feita pelo algoritmo para cada categoria. Nas matrizes com o número de instâncias e proporção real, pode-se ler que 95,7% (957 instâncias) dos documentos pertencentes à categoria "Política" foram corretamente classificados dentro da categoria "Política". Pelo contrario, foram mal classificados 0,8% (8 instâncias) na categoria "Crime" e 3,5% (35 instâncias) dentro da categoria "Ciência". Por outro lado, 92,1% (921 instâncias) dos documentos foram bem classificados pelo algoritmo dentro da categoria "Crime", 0,8% (8 instâncias) dos documentos foram classificados erroneamente dentro da categoria "Política" e 7,1% (71 instâncias) foi mal classificado dentro da categoria "Ciência. Dentro a categoria "Ciência", uma elevada quantidade de documentos foram bem categorizados ou classificados dentro da categoria "Ciência", ou seja, 99,5% (995 instâncias) dos documentos e apenas o 0,5% foram mal classificados dentro das categoria "Política" (3 instâncias) e "Crime" (2 instâncias). A Figura 88b, prova que o modelo Regressão Logística acertou na sua predição de classificação documental dentro da categoria "Política" num 98,9%, num 98,9% na classificação dos documentos dentro da categoria "Crime" e num 90,4% dos documentos classificados dentro da categoria "Ciência". Os demais documentos não pertencem realmente à categoria ás outras categorias.

Figura 88 – Matrizes de confusão com a predição do algoritmo de classificação Regressão Logística

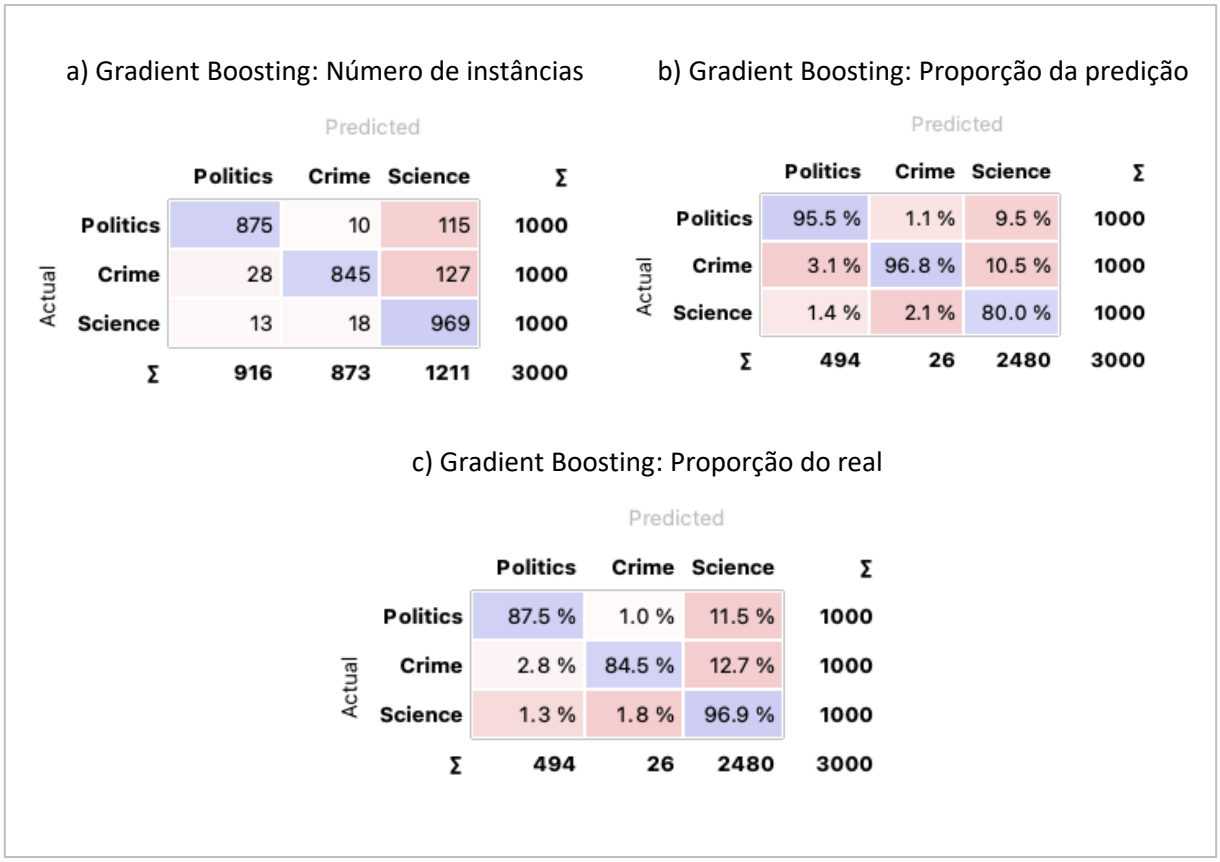


Fonte: Elaborado pela autora com os dados de pesquisa.

Na Figura 89, pode-se observar as matrizes de confusão com a predição do algoritmo de classificação *Gradient Boosting*. Na Figura 89a e 89c, observam-se as matrizes de confusão com o número de instâncias e a proporção real da classificação dos documentos. Essas figuras, mostram que 87,5% dos documentos (875 instâncias) foram corretamente classificados dentro da categoria "Política", 1,0% (10) foi classificado erroneamente dentro da categoria "Crime" e 11,5% (115 instâncias) foi mal classificado dentro da categoria "Ciência". Apenas 84,5% dos documentos, ou seja, 845 instâncias foram bem classificados pelo modelo dentro da categoria "Crime", 2,8% dos documentos (28 instâncias) foram classificados erroneamente dentro da categoria "Política" e 12,7% (127 instâncias) foi classificado erroneamente dentro da categoria "Ciência. Dentro da categoria "Ciência", 96,9% dos documentos (969 instâncias) foram corretamente classificados na categoria "Ciência" e os demais (1,3%) foram mal classificados dentro das categoria "Política" (13 instâncias o documentos) e "Crime" (18 instâncias). A Figura 89b apresenta a proporção da predição feita pelo algoritmo para cada categoria onde se pode ler que o algoritmo *Gradient Boosting* acertou na sua predição de classificação documental dentro da categoria "Política" num 95,5%, num 96,8% na classificação dos documentos dentro da categoria "Crime" e num 80% dos documentos classificados dentro da

categoria “Ciência”. Os demais documentos não pertencem realmente à categoria ás outras categorias.

Figura 89 – Matrizes de confusão com a predição do algoritmo de classificação Gradiente Boosting

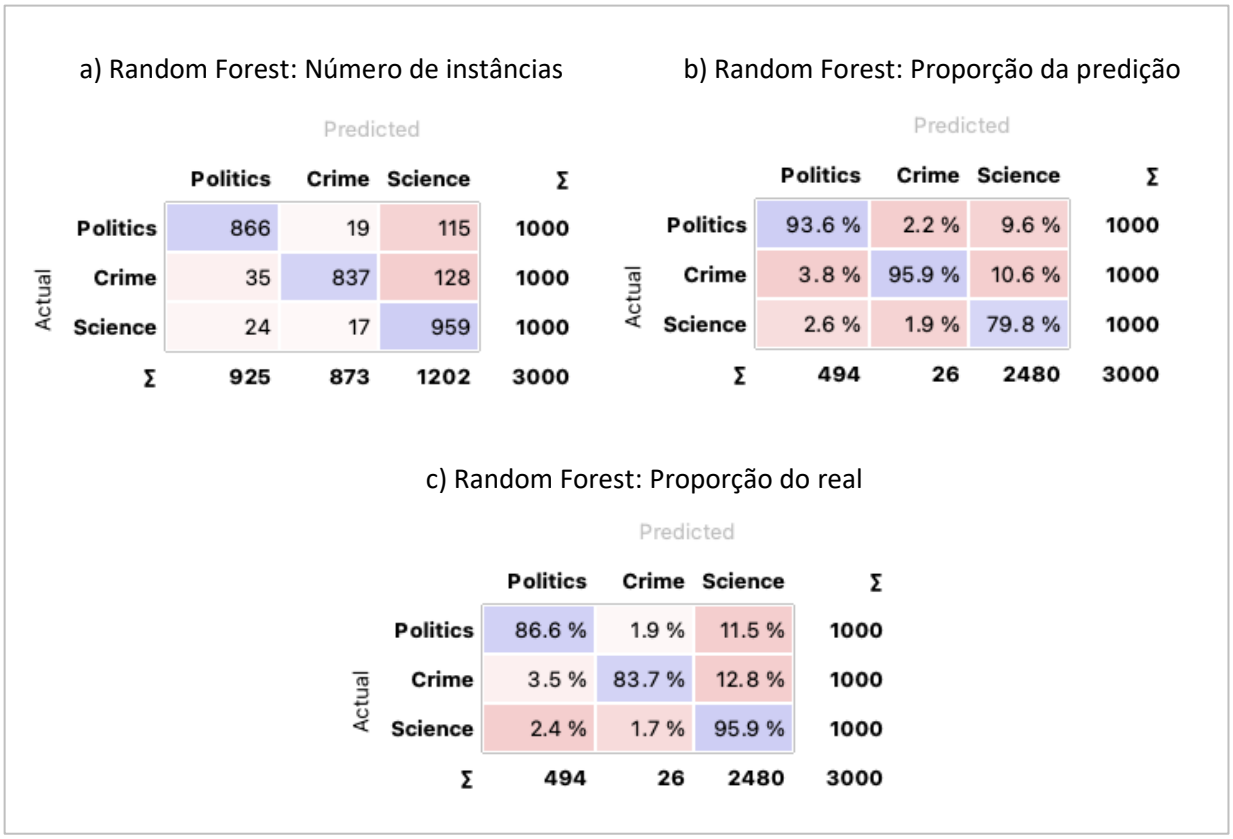


Fonte: Elaborado pela autora com os dados de pesquisa.

Na Figura 90, pode-se observar as matrizes de confusão com a predição do algoritmo de classificação *Random Forest* (Florestas aleatórias). Na Figura 90a e 90c, observam-se as matrizes de confusão com o número de instâncias e a proporção real da classificação dos documentos. Essas figuras, mostram que 86,6% dos documentos foram corretamente classificados dentro da categoria “Política”, 1,9% foi classificado erroneamente dentro da categoria “Crime” e 11,5% (115 instâncias) foi mal classificado dentro da categoria “Ciência”. Um 83,7% dos documentos foi corretamente classificados pelo modelo dentro da categoria “Crime”, 3,5% dos documentos foram classificados erroneamente dentro da categoria “Política” e 12,8% foi classificado erroneamente dentro da categoria “Ciência. Dentro da categoria “Ciência”, 95,9% dos documentos foram corretamente classificados na categoria de “Ciência” e os demais foram mal classificados dentro das categoria “Política” (2,4% instâncias o documentos) e “Crime” (1,7 instâncias). A Figura 90b apresenta a proporção da predição feita pelo algoritmo para cada categoria onde e pode se ler nessa figura que o modelo de *Random*

Forest acertou na sua predição de classificação documental dentro da categoria “Política” num 93,6%, num 95,9% na classificação dos documentos dentro da categoria “Crime” e num 79,8% dos documentos classificados dentro da categoria “Ciência”. Os demais documentos não pertencem realmente à categoria ás outras categoria segundo a predição do algoritmo.

Figura 90 – Matrizes de confusão com a predição do algoritmo de classificação Random Forest

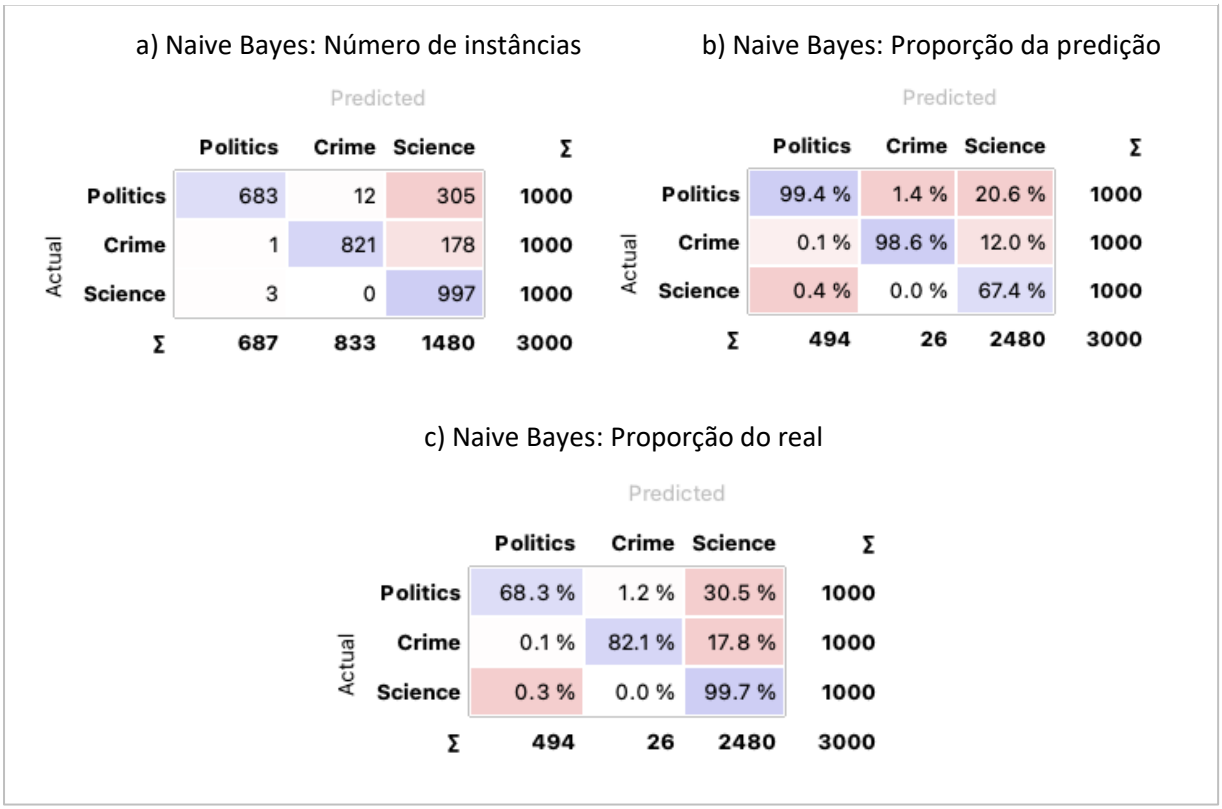


Fonte: Elaborado pela autora com os dados de pesquisa.

Na Figura 91, pode-se observar as matrizes de confusão com a predição do algoritmo de classificação *Naive Bayes* (Bayesiano). Na Figura 91a, pode-se ler que de 3000 instâncias em total, 683 documentos foram corretamente classificados dentro da categoria “Política”, 12 foram classificado erroneamente na categoria “Crime” e 305 documentos foram mal classificados dentro da categoria “Ciência”. Dentro da categoria “Crime”, 821 documentos foram corretamente classificados pelo modelo dentro da categoria “Crime”, apenas 1 documento foi mal classificados dentro da categoria “Política” e 178 documentos forma mal classificados dentro da categoria “Ciência”. No que tange à classificação de documentos da classe “Ciência”, e nenhum documento (0%) foi erroneamente classificado dentro da categoria. Dentro a categoria “Ciência”, 997 documentos foram corretamente classificados na categoria “Ciência” e os demais documentos foram mal classificados pelo modelo dentro das categoria

"Política" (3 instâncias) e "Crime" apresenta o valor nulo, ou seja, nenhuma instância foi classificada corretamente dentro da categoria "Crime". Já a Figura 91b, mostra que o algoritmo *Naive Bayes* acertou na sua predição de classificação documental dentro da categoria "Política" num 99,4%, num 98,61% na classificação dos documentos dentro da categoria "Crime" e num 67,4% dos documentos classificados dentro da categoria "Ciência". Os demais documentos não pertencem realmente à categoria ás outras categorias.

Figura 91 – Matrizes de confusão com a predição do algoritmo de classificação Naive Bayes

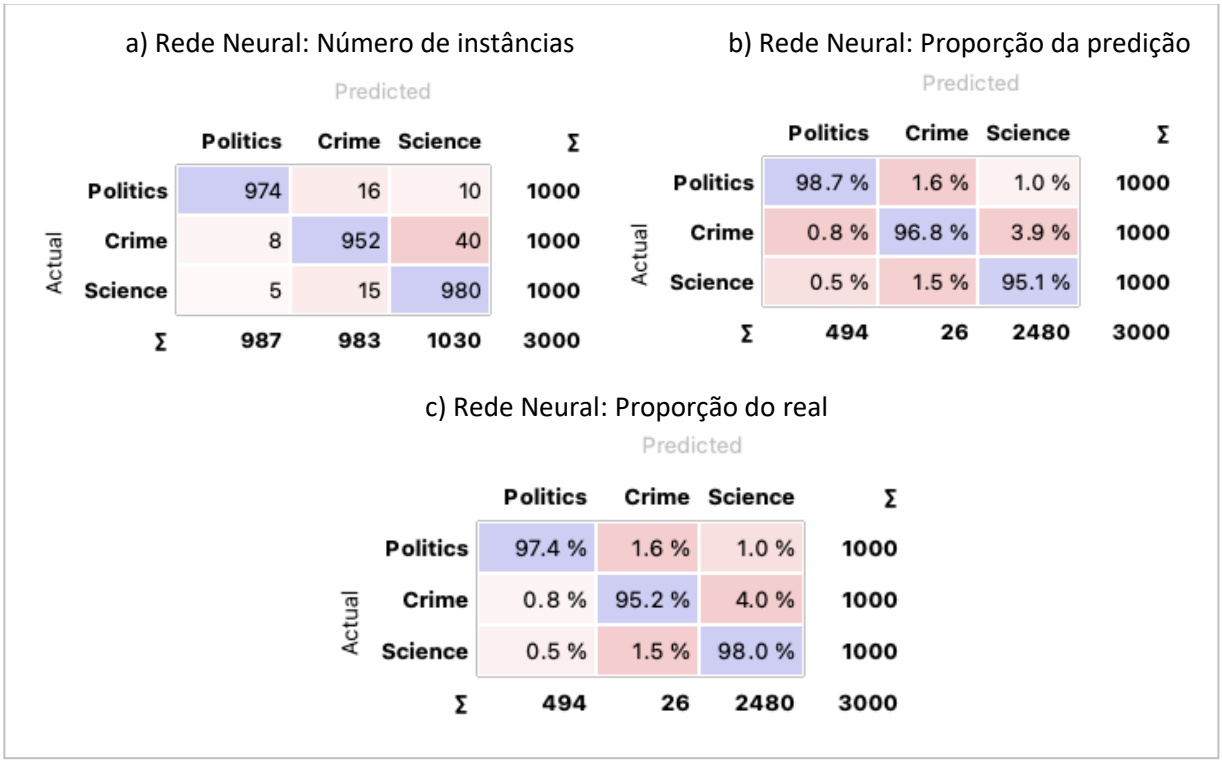


Fonte: Elaborado pela autora com os dados de pesquisa.

Por último, na Figura 92, pode-se observar as matrizes de confusão com a predição do algoritmo de classificação Rede Neural. Na Figura 92a e 92c, observam-se as matrizes de confusão com o número de instâncias e a proporção real da classificação dos documentos. Essas figuras, mostram que 97,4% dos documentos (974 instâncias) foram corretamente classificados dentro da categoria "Política", 1,6% (16 instâncias) foi classificado erroneamente dentro da categoria "Crime" e 1% dos documentos (10 instâncias) foi mal classificado dentro da categoria "Ciência". No que tange à categoria "Crime", um elevado número de documentos (95,2% dos documentos, ou seja, 952 instâncias foram bem classificados pelo modelo dentro da categoria "Crime", 0,8% dos documentos (8 instâncias) foram classificados erroneamente dentro da categoria "Política" e 4% dos documentos (40 instâncias) foi classificado erroneamente dentro

da categoria “Ciência. Dentro da categoria “Ciência”, 98% dos documentos (980 instâncias) foram corretamente classificados na categoria “Ciência” e os demais foram mal classificados dentro das categoria “Política” (0,5% das instancias) e “Crime” (1,5% das instâncias). Finalmente, cabe destacar que na Figura 92b o modelo de Rede Neural apresenta poucos erros de classificação documental para cada categoria, de fato pode se ver que que o algoritmo acertou na sua predição de classificação documental dentro da categoria “Política” num 98,7%, num 96,8% na classificação dos documentos dentro da categoria “Crime” e num 95,1% dos documentos classificados dentro da categoria “Ciência”. Os demais documentos não pertencem realmente ás outras categorias.

Figura 92 – Matrizes de confusão com a predição do algoritmo de classificação Rede Neural



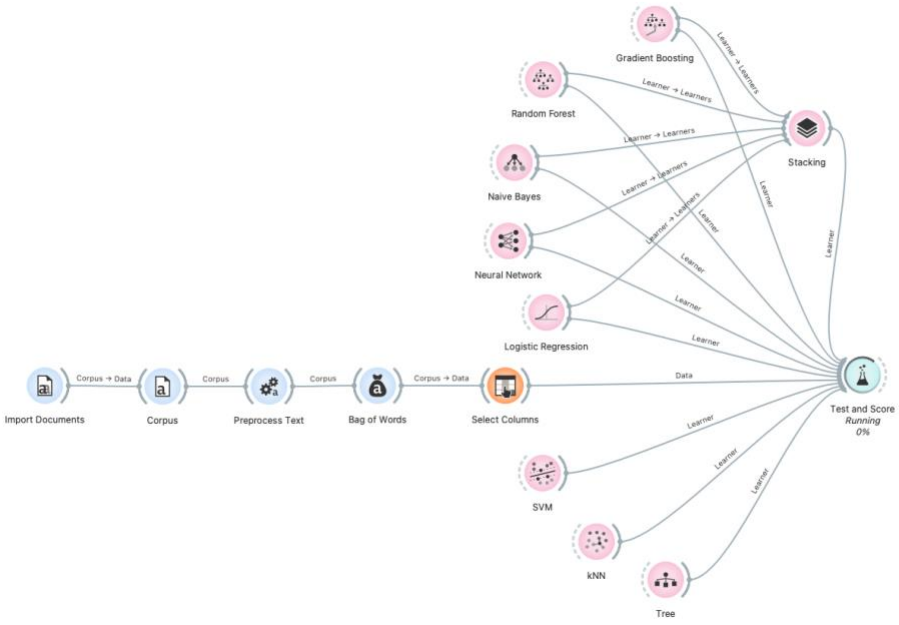
Fonte: Elaborado pela autora com os dados de pesquisa.

Feitas as análises das matrizes de confusão, pode-se comparar a precisão dos modelos e ver que os algoritmos k-NN e SVM são os menos precisos, porque cometeu mais erros e os modelos mais preciso são Regressão Logística, *Naive Bayes, Random Forest, Gandient Boosting*, e Rede Neural que são os que apresentaram um menor número de erros. Essa informação é essencial na tomada de decisões na hora de implementar o modelo e o SGDE(A).

4.4 Implementação do modelo (Fase VII da metodologia)

Após fazer os ajustes nos parâmetros e avaliar os modelos, nessa fase da metodologia se descreve como tentou-se implementar os modelos ou algoritmos de classificação. Para isso, empilharam-se multiplex modelos de classificação usando o método *Stacking* no *software* Orange data mining. Concretamente, foram escolhidos os modelos que tinha dado os melhores resultados nas medições estatísticas realizadas (AUC, acurácia do classificador, precisão, etc.). Porém, não foram obtidos resultados concludentes, devido à grande quantidade de dados o software ainda está processando os textos para empilhar os modelos. O empilhamento (*stack*) é um método de conjunto o método ensamblado que calcula um metamodelo formado por vários modelos básicos. No *software* Orange data mining, o empilhamento requer vários modelos na entrada e um método de agregação. Nesse caso, os modelos escolhidos foram Regressão Logística, *Gradient Boosting* (Sci-learn), *Random Forem*, *Naive Bayes*, Rede Neural e Classificação em árvore (*Tree*). Um meta-aprendiz construído é então enviado para *Test & Score*. Espera-se que os resultados melhoraram os anteriores, mesmo que apenas marginalmente. Cabe destacar que o empilhamento normalmente funciona bem em conjuntos de dados complexos. Así, na Figura 93 se apresenta o fluxograma, feito no Orange data mining 3, do Experimento 2, onde, nesse, caso, partiu-se do corpus, foram pré-processados os textos, aplicou-se o método BoW calculando o TF-IDF e selecionamos e eliminamos aquelas palavras irrelevantes (*would*, abc, *could*, *about* e as etiquetas HTML).

Figura 93 – Experimento 2 da fase de Implementación para implementar modelos de classificação documental



Fonte: Elaborado pela autora com no software Orange data mining.

Além disso, o seguinte passo para implementar os novos modelos de classificação, seria incorporar eles no fluxograma aos *workflows* ou fluxo de trabalho de um SGDE(A) padrão, como mostrado na Figura 33. Porém, isso fica para uma continuidade desse trabalho como pesquisa futura.

A integração do sistema gerado (um sistema automática de classificação de documentos electrónicos baseado em modelos de aprendiza automático) nu fluxograma de um SGDE(A) não só facilitará ao usuário a disponibilidade, recuperação e consulta de documentos, mas também pode resolver problemas de confiabilidade, disponibilidade e custódia de documentos, além de tornar mais eficientes as atividades de acolhimento e atenção de solicitações, reclamações e recursos, estabelecidas à empresa por seus usuários e entidades de controle como dito por Hernández Echeverry (2017). Nesse ponto, cabe recordar que a gestão documental ajuda a regular as práticas realizadas pelos responsáveis da sua gestão e por qualquer outra encarregada de gerar e utilizar documentos no exercício de suas atividades (OPENKM, 2022).

Seguindo a Milward (2015), também cabe destacar que os dados estruturados a partir da mineração de texto poderão ser integrados a bancos de dados, data *warehouses* ou painéis de inteligência de negócios, além de ser usados para análises descritivas, prescritivas ou preditivas.

Após a integração do modelos classificação documental gerado aqui num SGDE(A), cabe comprovar que o sistema diminuirá os erros humanos, automatizará tarefas e otimizará a gestão de conteúdos e os processos documentais dento de um SGDE(A). Pelo que esse trabalho, fica aberto a pesquisas futuras.

Após feitas essas aclarações, cabe fazer uma análise, avaliação e discussão dos resultados.

4.5 Análise, avaliação e discussão dos resultados (Fase VIII da metodologia)

Após analisar e avaliar os resultados obtidos desde diferentes métricas estatística, cabe destacar que os modelos de aprendizado de máquina testado e treinados que maior precisão apresentaram para a classificação do corpus documental foram a Regressão Logística, *Naive Bayes, Random Forest, Gandient Boosting,* e a Rede Neural, por ser os que apresentaram um menor número de erros na classificação. Pelo contrario, os modelos de k-NN e SVM foram os menos preciso, pois apresentaram maior número de erros na classificação documental nos experimentos da fase de modelagem. Porem, devido a que os experimentos foram realizados com um elevado número instâncias documentos (3000 e-mails), o tempo de treinamento e teste

de todos os modelos no software Orange Data Mining foi de 48 horas e 36 minutos. Algo que, ainda, deve ser otimizado usando outras ferramentas para reduzir o tempo, de forma que quando sejam integrados em um sistema de gerenciamento documental alguns dos modelos que apresentaram maior precisão e acurácia nessa pesquisa, agilize o sistema realmente.

Os resultados dessa pesquisa permitem corroborar as afirmações feitas por Khan e Madden (2014), Rezende (2003) e Tax (2001), quando dizem que avanço e aplicação das tecnologias de Aprendizado de Máquina, tem sido essencial para descobrir padrões e para classificação automática de conteúdo digital.

Os resultados da pesquisa também permitem concluir que o sistema de classificação documental gerado nesse trabalho com os modelos que apresentaram melhores resultados em acurácia e precisão são soluções que alcançaram um bom desempenho da classificação documental. Pelo que o desenho e implementação desse modelos permite contribuir diante da necessidade detectado por entendidos na matéria como Gallego García (2015); Hernández Echeverry (2017); Johnston e Bowen (2005) e Rangel Palencia (2019) de gerar esse tipo de soluções no âmbito da classificação de documentos.

Os resultados também demonstram que os modelos de representação adotado para a coleção de documentos objeto desse estudo durante o processo de Mineração de Textos tem grande impacto no resultado final do processo. Seguindo a Milward (2015), podemos concluir que o processo de mineração de texto permitiu examinar uma grande coleção de documentos, descobrir novas informações e ajudar a responder a perguntas de pesquisa feita nesse trabalho (Q1). Além disso, como apontado por Milward (2015), os dados revelam a eficiência e efetividade da mineração de texto para identificar fatos, relacionamentos e afirmações que, de outra forma, permaneceriam ocultos ou passariam desapercebidos. A mineração de texto, tem permitido nesse trabalho converter as informações em um formato estruturado susceptível de ser analisado posteriormente e apresentado diretamente usando tabelas HTML agrupadas, mapas mentais, gráficos, e etc. Esses dados estruturados, foram usados nesse estudo usados para análises descritiva e preditiva.

Por outro lado, cabe destacar que os resultados também revelam um desempenho favorável no que tange à aplicação ou sistema. Além disso, permitirá diminua erros humanos, automatize tarefas e otimize a gestão de conteúdos e os processos documentais de um SGDE(A). De fato, os modelos de ML aplicados nesse trabalho, permitiram: gerar algoritmos de classificação automática de documentos (modelo de classificação), desenvolver algoritmos para detectar padrões de gerenciamento de documentos, aplicar um sistema de regras para automatizar tarefas repetitivas, e fornecer capacidade de aprendizado para o sistema.

Seguindo a Jacob (1992), o sistema de classificação documental gerado como os diferentes modelos facilitou notavelmente a organizar do conhecimento.

O pré-processamento é etapa indispensável para aumentar a precisão dos algoritmos de ML aplicados a categorizações textuais (SILVA, 2021). Como apontado por Silva (2013, p.13), as transformações feitas nesse trabalho durante a fase de pré-processamento, foi um processo demorado e custoso feito com o máximo cuidado "para que o conhecimento adquirido posteriormente seja útil para o usuário final".

Também foi preciso desenvolver um sistema capazes de tratar informação não estruturada e transformá-las em informação estruturada, especificamente uma tabela atributo-valor, para permitir o uso de sistemas de aprendizado. Pois como a pontado pro Silva (2021), os sistemas de ML quase nunca conseguem trabalhar diretamente com informações digitais dispersas em documentos textuais porque esses textos tem um formato não estruturado.

Para a avaliação dos modelos de classificação e extração de conhecimento, foram utilizados o método de validação cruzada *k-fold*, pois como destacam Refaeilzadeh, Tang e Liu (2009) e Joanneum (2006), mesmo sendo um método mais lento computacionalmente falando que o método de retenção, é muito preciso porque que permite avaliar a partir de *k* combinações de dados de treinamento e teste,. Na prática, a escolha do número de iterações depende da medição do conjunto de dados. Utilizando a validação cruzada de 10 iterações por ser a mais comumente utilizada (*10-fold cross-validation*).

Por último, cabe destacar que os resultados corroboram as conclusões feitas por outros pesquisadores já mencionados no marco teórico como Oard (1994) que aplicou o uso de redes neurais para a classificação documental**,** podendo afirmar mais uma vez que os algoritmos baseados em redes neurais são instrumentos úteis para a recuperação de informações e para a classificação automática de documentos em diferentes áreas, nesse caso também na área de ciências da informação, e mais amplamente, na área das Ciências Sociais e Humanas.

Em soma, os resultados mostram que um sistema automáticos de classificação de documentos com base em modelos de aprendizado automático apresentam maior desempenho e otimização nos processos operacionais em relação a sistemas tradicionais.

Após discutir os resultados com a literatura, na seguinte seção são apresentadas as considerações finais do trabalho de pesquisa realizado.

5 CONCLUSÃO

O objetivo principal desse trabalho de pesquisa foi desenvolver um sistema automático de classificação de documentos com base em modelos de aprendizado de máquina seguindo uma metodologia que, posteriormente, permitira incorporar os novos modelos aos fluxo de trabalho padrão de gerenciamento de documentos, com o intuito de automatizar tarefas tediosas dentro da gestão documental como a de classificação de documento e recuperação de informação. Para a consecução desse objetivo, foram delineados cuidadosamente os objetivos específicos dentre os que cabe destacar aqui o mapeamento de algoritmos de aprendizado de máquina disponíveis na literatura; aplicação dos algoritmos mais destacados na literatura no com base em modelos de aprendizado de máquina; uso de diferentes modelos de modelos (Regressão Logística, *Naive Bayes*, *Random Forest, Gandient Boosting*, Rede Neural, k-NN e SVM); Descrição dos pontos fortes e fracos dos diferentes modelos utilizados; avaliação do impacto das diferentes formas de combinação das classificações dos modelos e comparação d o desempenho dos algoritmos de classificação utilizados.

Mas, para atingir esses objetivos, tanto o principal como os específicos, primeiramente foi feita pesquisa e análise bibliográfica dos principais assuntos abordados como inteligência artificial, técnicas e ferrramentas de aprendizado de máquina, análise inteligente de texto, mineração de texto, processamento da linguagem natural, categorização e pré-processamento de textos, técnicas de conversão de valores simbólicos para numéricos, métodos de avaliação de modelos de classificação, sistemas de informação e gestão documental, arquivamento e requisitos legais para implantar um modelo de gerenciamento documental. Esses tópicos foram trazidos com o intuito atingir um alinhamento entre os conhecimentos da área de computação, a área de administração e a aérea de Ciências da informação, além de somar e conectar conhecimentos entre o campo da ciência da computação, das ciências sociais e das ciências humanas. Agregando, assim, valor aos conhecimentos sobre inteligência artificial, big data, ciência de dados, PLN e mineração de textos nas diferentes áreas e campos de estudo.

Com este trabalho foi possível construir abordagens de aprendizado de máquina para classificar o corpus textual formado por 3000 e-mails de forma semiautomática e supervisionada. Assim, a inteligência humana, ainda foi necessária para realizar previamente uma categorização do conjunto de dados textuais objeto desse estudo.

Após estudar e analisar os diversos algoritmos de aprendizado de máquina amplamente utilizados para categorizar textos, foram escolhidos para esse trabalho sete algoritmos, SVM, k-NN, *Random Forest*, *Naive Bayes*, *Grandient Boosting*, Regressão Logística, e Rede Neural,

os quais, grosso modo, mostraram-se eficazes para resolver o problema estudado e alcançar o objetivo geral.

Diante desse contexto, neste trabalho de pesquisa foram aplicados modelos de aprendizado de máquina para desenvolver um sistema automático de classificação de documentos, seguindo uma metodologia que permitirá incorporar os novos modelos aos *workflows* padrão de gerenciamento de documentos. Concretamente, foram aplicadas técnicas de aprendizado supervisionado para a classificação supervisionada (*Naive Bayes*, SVM, Rede Neural) por se tratar de um conjunto de dados formado por documentos previamente classificados manualmente. Classificação que serviu como ponto de partida para treinar e testar o sistema inteligente na classificação automática. Também se usara aprendizado ensamblado (*Grandient Boosting* (sci-learn), e aprendizado baseado em instâncias (k-NN). Também, usaram-se algoritmos de classificação como Regressão logística e florestas aleatórias (*Random Forest*). Foram feitos todos os passos propostos na metodologia e atingidos, o que permitiu atingir todos os objetivos específicos e, por tanto, o objetivo geral.

A técnica de vetorização do Bag of Words aplicando o cálculo do TF-IDF, apesar da sua simplicidade, obteve resultados melhores que aplicando a técnica BoW sem calcular TF-IDF nos algoritmos.

Os resultados desse trabalho, permitem verificar que o uso de técnicas de aprendizado de máquina é tão útil como eficaz para a classificação de e-mails nas diferentes categorias. Para a classificação de uma base de dados que apresenta grande quantidade e variedade de informação, o algoritmo SVM apresentou os piores resultados de desempenho por média de categorias em relação aos demais algoritmos, com uma acurácia de 40,8%, precisão de 64,5%, uma revocação de 40,8%% e F1-Score de 31,7%. O algoritmo k-NN também apresentou péssimos resultados na comparação com os outros classificadores, como uma acurácia de 44,4%, precisão de 74,4%, uma revocação de 44%% e F1-Score de 36,8%. Pelo contrario os melhores resultados foram apresentados pelo algoritmo de Regressão Logística a pesar do seu alto tempo de execução (com uma acurácia de 99,4%, precisão de 96,1%, uma revocação de 95,8% e F1-Score de 95,8%) e pelo modelo de Rede Neural (com uma acurácia de 99,2%, precisão de 96,9%, uma revocação de 96,9% e F1-Score de 96,9%). A pesar dos algoritmos de Regressão Logística e Rede Neural ter obtido as melhores métricas, os algoritmos restantes, *Grandient Boosting* e *Random Forest* também apresentaram uma boa acurácia (95.8%, 85.76% e 96.7% respectivamente). Importante destacar que o algoritmo de *Naive Bayes* não obteve nenhum resultado na média de classes. Os resultados das métricas, obtidos com este estudo,

indica que os sete algoritmos testados podem ser aplicados em casos similares de categorização de mensagens eletrônicos com base no corpo de e-mail.

Foram elaborados diferentes cenários e experimentos para o pré-processamento nos que foi realizada a decomposição de texto em n-gramas, a remoção ou eliminação de palavras vazias (*empty words*), o cálculo da frequência dos n-gramas, a estimativa da estatística TF-IDF para os n-gramas e a seleção das variáveis mais importantes. Os modelos de representação adotado para a coleção de documentos objeto desse estudo durante o processo de Mineração de Textos tem grande impacto no resultado final do processo.

Após análise dos resultados obtidos após o treinamento e teste dos modelos e após a comparação feita entre os modelos, é possível responder à questão de pesquisa de foram afirmativa, pois os achados permitem afirma que os sistemas automáticos de classificação documental baseados em modelos de aprendizado de máquina apresentam maior otimização nos processos operacionais e um maior desempenho de organização, classificação e recuperação de informação em relação a sistemas tradicionais.

Com este trabalho, espera-se fornecer aos sistemas de gerenciamento documental maior automação, aumentar a produtividade dos usuários, melhorar a experiência e usabilidade dos usuários dos SGDEA, facilitar as tarefas de gestão documental, evitar erros de classificação e reduzir a dependência de pessoas para tarefas susceptíveis de sistematização.

Dentre os resultados e impactos esperados desse trabalho, destacam-se: o impacto considerável em um amplo grupo de empresas e entidades que possuem ou desejam adquirir um Sistemas de Gestão de Registros ou *Electronic Records Management Systems* (ERMS), que poderá ser integrado aos serviços de processamento em nuvem, facilitando o processo de transformação digital. Lembrando que os ERMS e os EDRMS (*Electronic Document and Records Management Systems*) são os aplicativos que permitem o gerenciamento automático de documentos (JOHNSTON e BOWEN, 2005); incorporação dos novos modelos aos *workflows* padrões de gerenciamento de documentos; aumento da agilidade e sistematização nas tarefas de gestão documental; e redução de erros de classificação e da dependência de pessoas para tarefas que podem ser sistematizadas.

5.1 Limitações do estúdio e principias problemas de pesquisa

Durante a realização desse trabalho de pesquisa, foram encontradas algumas limitações ou problemas. Uma delas foi relacionada à dificuldade para encontrar bases de dados públicas para obter um conjunto de dados formado por textos.

Outra limitação foi tentar carregar os dados em formato texto em Google Colab e Júpiter, pelo que finalmente foi escolhido o *software* livre Orange Data Mining para poder fazer essa tarefa e converter os dados em tabelas em formato ".CSV", mesmo que outra dificuldade foi encontrada na hora de aplicar o modelo de validação cruzada k-fold com os 3000 documentos que formavam o *dataset*. Além disso, precisou processar muito tempo para obter resultados.

Finalmente, cabe voltar a destacar que após fazer os ajustes nos parâmetros e avaliar os modelos e para implementar os modelos de classificação, aplicou-se o empilhamento multiplex dos modelos melhores modelos de classificação usando o método *Stacking* no Orange, porém, devido à delimitação de tempo em terminar esse trabalho e entregar não foram terminados os processamentos ainda, pelo que não pode-se mostrar resultados concludentes dessa implementação, pois devido à grande quantidade de dados o software ainda está processando os textos para empilhar os modelos. Mas essa limitação se torna um desafio e um ponto de partida para trabalhos de pesquisa futuros junto com outras ideias de trabalhos detalhados a seguir.

5.2 Líneas de pesquisa e trabalhos futuros

Esse trabalho é um ponto de partida para trabalhos futuros e outras líneas de pesquisa, como por exemplo testar diferentes cenários de implementação dos modelos; testar com outros modelos; fazer uma comparação dos resultados obtidos nesse estudo com outros métodos de representação de documentos; utilizar outros dados e metadados que compõem o conjunto de dados textual, como: assunto, destinatário(s) e remetente dos e-mails; gerar novas categorias; ou classificar uma base de dados mais genérica, ou seja, com menos categorias ou sem balancear.

Outra ideia de pesquisa futuras seria usar outras métricas para validar os resultados, como a curva ROC e a média macro, por exemplo.

Além disso, propõe-se como trabalhos futuros replicar esse trabalho para classificar site, analisar sentimentos ou classificar outros conjuntos de dados formado por textos de diferente índole como administrativos, comerciais ou jurídicos.

Um dos principais interesses desse estudo é a evolução do protótipo para um serviço *web*, que permita que esta ferramenta chegue a diferentes profissionais, pois poderia se aplicar na área judiciaria, da saúde, da administração, da educação, e etc. Em trabalhos ou pesquisas futuras, seria relevante a realização de testes do utilizador, para validar a aplicação em termos de usabilidade, tempo e satisfação, e que fosse feita a atualização dos classificadores com a

obtenção de novos registos. Poderia se realizar, inclusive, o analise de sentimento aplicando o analise inteligente de textos ou *text mining*.

Sugere-se, também, para pesquisas futuras, o aprofundamento nas questões que envolvem ética em inteligência artificial e big data, que é um grande desafio.

REFERÊNCIAS

ADAPTING. Grupo Adapting: software for records management. **Página de seção de noticias de I+D+i e IA.** 2021. Disponível em: https://www.adapting.com/como-disenar-y-desplegar-modelos-de-machine-learning-en-flujos-documentales/ Acesso em: 15/09/2021.

AIIM-ASSOCIATION FOR INFORMATION AND IMAGE MANAGEMENT. **What is Enterprise Content Management (ECM)?** [online]. 2022. Disponível em: http://www.aiim.org/What-is-ECM-Enterprise-Content. Acesso em: 04/01/2022.

AIROLA, A.; PAHIKKALA, T.; WAEGEMAN, W.; DE BAETS, B.; e SALAKOSKI, T. (2011-04-01). An experimental comparison of cross-validation techniques for estimating the area under the ROC curve. **Computational Statistics & Data Analysis**, v. 55, n. 4, p. 1828–1844. doi:10.1016/j.csda.2010.11.018.

AMARAL, F. **Aprenda Mineração de Dados Teoria e Práttica**. Rio de Janeiro: Alta Books, 2016.

BAEZA-YATES; e R. RIBEIRO-NETO, B. **Recuperação de Informação: Conceitos e tecnologia das máquinas de busca**. Tradução técnica: Leandro Krug Wives, Viviane Pereira Moreira. 2ª Ed. Porto Alegre: Bookman, 2013.

BARBOSA, B. Minerando informações de textos. *In:* **Trilha de Machine Learning**. 2020. Disponível em: https://www.slideshare.net/BarbaraBarbosaClaudi/minerando-informaes-de-textos. Acesso em: 28/04/2022

BERRONDO URRUZOLA, Ander. **Detección de carreteras en imágenes de reconocimiento remoto mediante Deep Learning**. Trabalho de Conclussão de Curso [Trabajo de Fin de Grado], Universidad del País Vasco, 2020. Disponível em: https://addi.ehu.es/bitstream/handle/10810/48823/Memoria_AnderBerrondo.pdf?sequence=1. Acesso em: 04/05/2022.

BERRY, M. J.; e LINOFF, G. S. **Data mining techniques: for marketing, sales, and customer relationship management.** [S.l.]: John Wiley & Sons, 2004.

BIRD, S.; KLEIN, E.; e LOPER, E. **Natural language processing with Python**. O'Reilly, Beijing; Cambridge [Mass.], 2009.

BREIMAN, L. **Bagging Predictors. Machine Learning**, 1996, p. 123–140.

PACIFICO, L. D.; BRITTO, L. F.; LUDERMIR, T. B. Reconhecimento de plantas medicinais através de características das folhas e aprendizagem de máquina. In: SBC. Anais do XIV Brazilian e-Science Workshop. [S.l.], 2020. p. 17–24.

BRITTO, L.; e PACíFICO, L. Uma abordagem de classificação de sentimentos em revisões de livros em português brasileiro usando diferentes métodos de extração de características. In: SBC. **Anais do XVII Encontro Nacional de Inteligência Artificial e Computacional**. [S.l.], 2020. p. 116–127.

CAMPOS, G. M. Estatística prática para docentes e pós-graduandos. [S.l.]: Faculdade de Odontologia de Ribeirão Preto da Universidade de São Paulo, 2005.

CELISSE, Alain. Optimal cross-validation in density estimation with the LogLoss. **The Annals of Statistics**, v. 42, n. 5, p. 1879–1910, 2014. arXiv:0811.0802. doi:10.1214/14-AOS1240. ISSN 0090-5364. S2CID 17833620.

CHARTE, F. ¿Cómo es el proceso de extraer conocimiento a partir de bases de datos? Recurso *online* de Campus MVP. Publicado em 17 nov. 2020. Disponível em: https://www.campusmvp.es/recursos/post/el-proceso-de-extraccion-de-conocimiento-a-partir-de-bases-de-datos.aspx. Acesso em: 10/04/2022.

CHEESEMAN, P.; e STUTZ, J., Bayesian Classification (AutoClass): **Theory and Results**, In U. M. Fayyad, G. Piatetsky-Shapiro, P. Smith, and R. Uthurusamy, editors, Advances in Knowledge Discovery and Data Mining, pages 153-180, 1996. AAAI/MIT Press. Disponível em: http://ic-www.arc.nasa.gov/ic/projects/bayes- group/people/cheeseman/home.html. Acesso em: 18/05/2022.

CHIPMAN, J. W. et al. Mapping lake water clarity with landsat images in wisconsin, usa. Canadian journal of remote sensing, **Taylor & Francis**, v. 30, n. 1, p. 1–7, 2004.

CHOWDHURY, G. G. Natural language processing. **Annual review of information Science and technology, Wiley Online Library**, v. 37, n. 1, p. 51-89, 2003b.

CHOWDHURY, G. Natural language processing. **Annual Review of Information Science and Technology**, v. 37, n. 1, p. 51-89, jan. 2003a. ISSN 0066-4200.

DAS. **Formação Cientista de Dados**. Curso de Machine Learning ofertado por Data Science Academy. E-book. 2017

DAUMÉ III, H. **A course in machine learning**. Publisher, ciml. info, v. 5, p. 69, 2012.

DE MAGALHÃES, Lúcia Helena, et al. **Agrupamento automático de notícias de jornais on-line usando técnicas de machine learning para clustering de textos no idioma português**. 2020.

DEBARR, D.; e WECHSLER, H. Spam detection using clustering, random forests, and active learning. In: **CITESEER. Sixth Conference on Email and Anti-Spam**. Mountain View, California. [S.l.], 2009. p. 1–6.

DEEPTHI, A. L.; e PRASAD, J.V.D. Hierarchal clustering and similarity measures along with multi representation. **IJRET: International Journal of Research in Engineering and Technology**, v. 2, n. 8, p. 78-79, 2013. Disponível em: https://www.academia.edu/7527139/HIERARCHAL_CLUSTERING_AND_SIMILARITY_MEA SURES_ALONG_WITH_MULTI_REPRESENTATION?auto=download. Acesso em 21/05/2022.

DEVIJVER, P. A.; e KITTLER, J. **Pattern Recognition: A Statistical Approach**. Prentice-Hall: Londres, 1982

DUDA, R. O.; HART, P. E.; e STORK, D. G. Pattern Classification, Hoboken. [S.l.]: NJ: Wiley, 2001.

ELKAN, Charles. **Evaluating Classifiers**. University of California, San Diego, 18 de enero de 2011. Disponível em: https://web.archive.org/web/20111218192652/http://cseweb.ucsd.edu/~elkan/250B/classifiereval.pdf. Acesso em: 25/03/2022.

ESCOVEDO, T. et al. Neuroevolutionary learning in nonstationary environments. **Applied Intelligence, Springer,** v. 50, n. 5, p. 1590–1608, 2020.

FACELI, Katti. et al. **Inteligência Artificial: uma abordagem de aprendizado de máquina**. Rio de Janeiro: LTC, 2011.

FARIA, M. M.; e MONTEIRO, A. M. **Investigação sobre técnicas de detecção de intrusões em redes de computadorescom base nos algoritmos knn e k-means**. 2015.

FAYYAD, U. M. et al. Knowledge discovery and data mining: Towards a unifying framework. In: KDD. [S.l.: s.n.], 1996. v. 96, p. 82–88. C

FELDMAN, S. Nlp meets the jabberwocky: Natural language processing in information retrieval. ONLINE-WESTON THEN WILTON-, Citeseer, v. 23, p. 62–73, 1999.

FERNANDES FERNANDO TIMOTEO; CHIAVEGATTO FILHO, A. D. P. **Perspectivas do uso de mineração de dados e aprendizado de máquina em saúde e segurança no trabalho**. 2019. Disponível em: https://developers.google.com/ machine-learning/crash-course/embeddings/translating-to-a-lower-dimensional-space>. Acesso em: 15/03/2022.

FIGUEROLA, C. G.; ALONSO BERROCAL, J. L.; ZAZO RODRÍGUEZ, Á. F.; e RODRÍGUEZ VÁZQUEZ DE ALDANA, E. Algunas Técnicas de Clasificación Automática de Documentos. Universidad de Salamanca, Grupo REINA. *Cuadernos de Documentación Multimedia*, *15*, 2005. Disponível em: https://reina.usal.es/biblio/files/figuerola2005algunas.pdf. Acesso em: 18/06/2022.

FIGUEROLA, Carlos G.; ZAZO RODRÍGUEZ, Ángel F.; e ALONSO BERROCAL, Jose Luís. Automatic vs. manual categorization of documents in Spanish. **Journal of Documentation**, v. 57, n. 6, p.763–773, 2001.

FONTANA, E. Introdução aos algoritmos de aprendizagem supervisionada. Universidade de Paraná: 2020.

FUKUNAGA, K.; e NARENDRA, P. M. A branch and bound algorithm for computing k-nearest neighbors. **IEEE transactions on computers, IEEE**, v. 100, n. 7, p. 750–753, 1975.

GALLEGO GARCÍA, F. **Auditoría de MOREQ2 en el SGDE del Ayuntamiento de Picanya**. 2015. Tese (Doutorado). Universitat Politècnica de València, 2015.

GARCÍA ALSINA, Montserrat. **Gestión de documentos electrónicos en el contexto de gestión de la información**. Barcelona, UOC, 2009.

GOLDSCHMIDT, Ronaldo; PASSOS, Emmanuel; e BEZERRA, Eduardo. **Data Mining.** Elsevier Brasil, 2015.

GONÇALVES, 2013 GONÇALVES, M. **Classificação de Textos**. *In:* BAEZA-YATES, R. RIBEIRO-NETO, B. Recuperação de Informação: Conceitos e tecnologia das máquinas de busca. Tradução técnica: Leandro Krug Wives, Viviane Pereira Moreira. 2. ed. Porto Alegre: Bookman, 2013. p. 277-338.

GOOGLEDEVELOPERS. 2021. Disponível em: https://developers.google.com/ machine-learning/crash-course/embeddings/translating-to-a-lower-dimensional-space. Acesso em: 15/04/2022.

HAN, J.; KAMBER, M. Data mining: Concepts and techniques, 2nd editionmorgan kaufmann publishers. San Francisco, CA, USA, 2006.

HASTIE, Trevor; TIBSHIRANI, Robert; e FRIEDMAN, Jeronime. **The elements of statistical learning: data mining, inference, and prediction**. New York: springer, 2dn Edition, 2009. https://hastie.su.domains/ElemStatLearn/

HERNÁNDEZ ECHEVERRY, A. **Análisis y diseño de un modelo de gestión electrónica de documentos para los procesos operativos de terreno**. Universidade de La Salle, Bogotá, 2017. Disponível em: https://ciencia.lasalle.edu.co/sistemas_informacion_documentacion/263. Acesso em: 04 Jan. 2022. Acesso em: 25/01/2022

HOSEA, S.; HARIKRISHNAN, V.; e RAJKUMAR, K. Artificial intelligence. In: 2011 3rd International Conference on Electronics Computer Technology, v. 4, p. 124-129, 2011. Disponível em: https://www.researchgate.net/publication/273895978_Classification_and_Categorization_Drawing_the_Line. Acesso em: 10/01/2022.

HOTHO, A; NÜRNBERGER. A.; e PAAβ, G. A brief survey of text mining. In: CITESEER. **Ldv Forum**. [S.l.], v. 20, n. 1, p. 19–62, 2005.

INTELLIGENT. **Machine Learning & NPL: cómo funciona un clasificador de documentos**. **Publicação eletrônica,** publicado em 16/01/2019. Disponível em: https://itelligent.es/es/pln-clasificacion-automatica-documentos/. Acesso em: 12/12/2021.

JACOB, E. K. Classification and categorization: a difference that makes a difference. **Library Trends**, v. 52, n. 3, p. 515-540, 2004. Disponível em: https://www.ideals.illinois.edu/handle/2142/1686. Accesso em: 14/01/2022.

JACOB, E. K. **Classification and Categorization: Drawing the Line**. 2nd ASIS SIG/CR Classification Research Workshop, p. 63-80. doi:10.7152/acro.v2i1.12548, 1992.

JACOB, E. K. **Ontologies and the semantic** web. Bulletin of the American Society for Information Science and Technology, Silver Spring, p. 16-18, Apr./May 2004.

JAIN, A., DUBES, R. **Algorithms for Clustering Data**. Prentice-Hall, Englewood Cliffs: NJ, 1988.

JOACHIMS, Thorsten. A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. In Douglas H. Fisher, editor, **Proceedings of ICML-97, 14th International Conference on Machine Learning**, p. 143–151, Nashville, US, 1997. Morgan Kaufmann Publishers, San Francisco, US.

JOANNEUM, F. H. **Cross-Validation Explained**, Institute for Genomics and Bioinformatics, 2006. http://genome.tugraz.at/proclassify/help/pages/XV.html

JOHNSTON, G. P.; e BOWEN, D. V. The benefits of electronic records management systems. A general review of published and some unpublished cases. **Records Management Journal**, v. 15, n. 3, p. 131-140, 2005.

Jones, S. K. A Statistical Interpretation of Term Specificity and Its Application in Retrieval. **Journal of Documentation**, *v.* 28, n. 1, p. 1-*21, 1972.* CiteSeerX 10.1.1.115.8343. doi*:*10.1108/eb026526

Jones, D. S. **Elementary Information Theory.** Clarendon Press: Oxford, 1979.

KAUFMAN, D. A inteligência artificial irá suplantar a inteligência humana? [S.l.]: ESTACAO DAS LETRAS E CORES EDI, 2019.

KHAN, S.S.; e MADDEN, M.G. One-class classification: taxonomy of study and review of techniques. **The Knowledge Engineering Review**, v. 29, n. 3, p. 345–374, 2014.

LANG, Jean-Philippe. **Predictors tutorial**. Archivado el 3 de enero de 2014 en Wayback Machine, Bioinformatic Department Projects.

LAUDON, K. C.; e LAUDON, J. P. **Sistemas de información gerencial**. México: Pearson (14 Ed.), p. 1–684, 2016.

LEWIS, David D.; SCHAPIRE, Robert E.; e CALLAN, James P.; e PAPKA, Ron. Training algorithms for linear text classifiers. In **Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval**, August 18–22, 1996, Zurich, Switzerland (Special Issue of the SIGIR Forum), p 298–306. ACM, 1996.

LIMA, G. A. B. de O. Modelos de categorização: apresentando o modelo clássico e o modelo de protótipos. **Perspectivas em Ciência da Informação**, v.15, n.2, p.108-122, maio/ago. 2010. Disponível em: http://www.scielo.br/pdf/pci/v15n2/a08v15n2.pdf. Acesso em: 15/04/2022.

LORENA, A. C.; e CARVALHO, A. de. Uma introdução às *support vector machines*. **Revista de Informática Teórica e Aplicada**, v. 14, n. 2, p. 43–67, 2007. ISSN 21752745.

LOVINS, J. B. Development of a stemming algorithm. Mech. Transl. **Comput. Linguistics**, v. 11, n. 1-2, p. 22–31, 1968.

MAHESH, Batta. Machine Learning Algorithms-A Review. **International Journal of Science and Research (IJSR**), v. 9, p. 381-386, 2020. Disponível em: https://www.researchgate.net/profile/Batta-Mahesh/publication/344717762_Machine_Learning_Algorithms_-A_Review/links/5f8b2365299bf1b53e2d243a/Machine-Learning-Algorithms-A-Review.pdf. Acesso em: 23/01/2022.

MARKOV, Z.; e LAROSE, D. T. **Data Mining the Web: Uncovering Patterns in Web Content, Structure, and Usage**. New Britain: Wiley, 2007.

MARON, M. Automatic indexing: an experimental inquiry. **Journal of the ACM**, v. 8, p.404– 417, 1961.

MATOS, D. **Conceitos Fundamentais de Machine Learning**. 22 de setembro de 2015 Disponível em:bhttp://www.cienciaedados.com/conceitos-fundamentais-de-machine-learning/. Acesso em: 25/01/2022.

MATSUBARA, E. T. **O Algoritmo de Aprendizado Semi-Supervisionado CO-TRAINING e sua Aplicação na Rotulação de Documentos**. Dissertação (Mestrado em Ciências de Computação e Matemática Computacional). Instituto de Ciências Matemáticas e de Computação - ICMC-USP. São Carlos, 2004.

McLACHLAN, Geoffrey J.; DO, Kim-Anh; e AMBROISE, Christophe (2004). **Analyzing microarray gene expression data**. Wiley, 2004.

McTEAR, M.; CALLEJAS, Z.; GRIOL, D. Affective conversational interfaces. In: **The Conversational Interface.** [S.l.]: Springer, 2016.

MÉNDEZ, M. S. A. R.; PERDOMO, M. S. E. P.; ISABEL, M. S. M.; DOMÍNGUEZ, C.; e ZALDÍVAR, V. J. G. **Biblioteca Digital con técnicas de clasificación automática de documentos.** VIII Edição da Conferencia Científica Internacional da Universidade de Holguín, 2017.

MICHIE, Donald; SPIEGELHALTER, David J.; e TAYLOR, Charles C. **Machine learning, neural and statistical classification**. 1994.

MIKOLOV, T. et al. **Efficient estimation of word representations in vector space**. Proceedings of Workshop at ICLR, v.1, 2013.

MILWARD, David. **Presentation: Linguamatics I2E and Machine Learning**. Website de Linguamatics an IQVIA Company, 2015. Disponível em: https://www.linguamatics.com/what-text-mining-text-analytics-and-natural-language-processing. Accesso em: 28/01/2022.

MITCHELL, T. M. Machine Learning. McGraw-Hill, 1997.

MOLINARO, A. M.; SIMON, R.; e PFEIFFER, R. M. "Prediction error estimation: a comparison of resampling methods". Bioinformatics. **V. 21, n.** 15, p. 3301–3307, 2005). . doi:10.1093/bioinformatics/bti499. ISSN 1367-4803. PMID 15905277.

MONARD, M. C.; e BARANAUSKAS, J. A. **Sistemas Inteligentes: Fundamentos e Aplicações**, capítulo Conceitos sobre Aprendizado de Máquina, p. 89–114, 2003.

MONTEIRO, S. T.; e RIBEIRO, C. H. Desempenho de algoritmos de aprendizagem por reforço sob condições de ambiguidade sensorial em robótica móvel. Sba: **Controle & Automação Sociedade Brasileira de Automatica, SciELO** Brasil, v. 15, n. 3, p. 320–338, 2004.

MORAIS, E. A. M.; AMBRÓSIO, A. P. L. **Mineração de Textos**. Relatório Técnico - Instituto de Informática, Universidade Federal de Goiás. Goiânia, 2007, 30p.. Disponível em: http://www.inf.ufg.br/sites/default/files/uploads/relatorios-tecnicos/RT-INF_005-07.pdf. Acesso em: 03/04/2022.

MOURA, S. D. O. **Uma abordagem para a escolha do melhor método de seleção de instâncias usando meta-aprendizagem**. Dissertação (Mestrado) — Universidade Federal de Pernambuco, 2015. Disponível em: https://repositorio.ufpe.br/handle/123456789/16311. Accesso em: 15/03/2022

MÜLLER, A. C;. e GUIDO, S. **Introduction to Machine Learning with Python**. O'Reilly Media, 2017.

NEVES, R. d. C. D. d. **Pré-processamento no processo de descoberta de conhecimento em banco de dados.** 2003.

NING, S.; e YAN, M. Discussion on research and development of artificial intelligence. In: **IEEE. 2010 IEEE International Conference on Advanced Management Science (ICAMS 2010).** [S.l.], v. 1, p. 110-112, 2010.

NOEL, S., RAGHAVAN, V., e CHU., C. H. **Document Clustering, visualization and retrieval via link mining**. In: Weili Wu, Hui Xiong, and Shashi Shekhar, editors. Clustering and Information Retrieval. Kluwer, p.161–194, 2003.

NOONAN, Daniel. **EDMS/ERMS/ECM Explained - Ohio State University Libraries. 2011** [en línea] Library.osu.edu. Disponible en: https://library.osu.edu/projects-initiatives/osu-records-management/electronic-records/edms-erms-ecm-explained. Accesso em: 04/09/2021.

OARD, D. W. **Neural networks in information filtering and retrieval**, 1994.

OPENKM, 2022. **Convierta la información en conocimiento**. Recurso online. Disponível em: https://www.openkm.com/es/gestion-documental.html?gclid=Cj0KCQiA_c-OBhDFARIsAIFg3ezT2ryfzQrmXx1XRYMt72funzD2k1f3ylY-ftr4pmjdzHONx6EVlOQaAmfTEALw_wcB. Acesso em: 15/01/2022

ORWIG, C.; CHEN, R.; e SCHUFFELS, H. Internet categorization and search: a machine learning approach. **Journal of Visual Communications and Image Representation**, v. 1, n.7, p.88–102, 1996.

PEDREGOSA, F. et al. Scikit-learn: Machine learning in python. **The Journal of machine Learning research, JMLR. org**, v. 12, p. 2825–2830, 2011.

PETERMANN, R. J. et al. **Modelo de mineração de dados para classificação de clientes em telecomunicação**. Pontifícia Universidade Católica do Rio Grande do Sul, 2006.

PYLE, D. **Data preparation for data mining**. [S.l.]: morgan kaufmann, 1999.

RAJARAMAN, A.; e ULLMAN, J. D. **Mining of massive datasets**. [S.l.]: Cambridge University Press, 2011.

RAJARAMAN, A.; LESKOVEC, J.; e ULLMAN, J. D. **Mining Massive Datasets**. [s.n.], 2014. Disponível em: http://infolab.stanford.edu/˜ullman/mmds/book.pdf. Acesso em: 20/02/2022.

RANGEL PALENCIA, E. L. **Guía de Implementación de un Sistema de Gestión de Documentos Electrónicos de Archivo (SGDEA).** 2017. Disponível em: https://www.archivogeneral.gov.co/caja_de_herramientas/docs/12.%20herramientas/DT%20-%20IMPLEMENTACION%20DE%20UN%20SGDEA.pdf. Acesso em: 20/02/2022.

RANGEL PALENCIA, E. L. **Propuesta de fortalecimiento al proceso de preservación digital en el Archivo General de la Nación de Colombia, solución basada en arquitectura empresarial**. Tese (Doutorado). Bogotá: Universidad Externado de Colombia, 2019.

REFAEILZADEH, Payam; TANG, Lei; e LIU, Huan. Cross-Validation. In: Arizona State University, **Encyclopedia of database systems**, v. 5, p. 532-538, 2009. Disponível em: https://web.archive.org/web/20110905044421/http:/www.public.asu.edu/~ltang9/papers/ency-cross-validation.pdf. Acesso em: 17/07/2022.

REZENDE, S. O. **Sistemas inteligentes**: **fundamentos e aplicações**. São Paulo: Manole, 2003.

ROCCHIO, J. J. **Relevance feedback in information retrieval**. In Gerard Salton, editor, The SMART Retrieval System. Experiments in Automatic Document Processing, p. 313–323. Prentice Hall, Englewoods Cliffs, N. J., 1971.

SANTOS, Cedric Michael dos. **Classificação de documentos com processamento de linguagem natural.** Tese (Mestrado). Coimbra, Portugal: Instituto Superior de Engenharia de Coimbra, 2015. Disponível em: https://comum.rcaap.pt/handle/10400.26/15293 Acesso em: 20/04/2022.

SCHAPIRE, R. **The strength of weak learnability**. Machine Learning. p. 197–227, 1990.

SCHEICHER, R. B.; ROBERTA, A. S.; KOGA, N. J.; e REZENDE, S. O. **Uso de expressões do domínio na classificação automática de documentos**. XIII Encontro Nacional de Inteligência Artificial e Computacional, 625-636, 2016.

SCHNEIDER, Jeff. **Cross Validation.** Disponível em: https://www.cs.cmu.edu/~schneide/tut5/node42.html. Acesso em: *11/04/2022*

SCHUTZE, H.; HULL, D. A.; e PEDERSEN, J. Q. **Uma comparação de classificações e representações documentais para o problema de roteamento**. SIGIR 95, 1995.

SEBASTIANI, F. **Text categorization. In: Encyclopedia of Database Technologies and Applications.** [S.l.]: IGI Global, 2005. p. 683–687.

SHUANG, K. et al. Convolution–deconvolution word embedding: An end-to-end multi-prototype fusion embedding method for natural language processing. **Information Fusion**, v. 53, n. 06, 2019.

SILVA, Bruno Ferreira. **Utilização de aprendizagem de machine para classificação de e-mails em categorias relevantes**. Trabalho de Conclusão de Curso. Universidade Federal de São Carlos Departamento de Computação Engenharia de Computação, 2021.

SOARES, M. V. B. et al. **Pretext ii: descrição da reestruturação da ferramenta de pré-processamento de textos**. São Carlos, SP, Brasil., 2008.

SUTTON, R. S.; e BARTO, A. G. **Introduction to reinforcement learning**. 1998.

TAVARES, L. G.; LOPES, H. S.; LIMA, C. R. E. Estudo comparativo de métodos de aprendizado de máquina na detecção de regiões promotoras de genes de escherichia coli. **Anais do I Simpósio Brasileiro de Inteligência Computacional**, p. 8–11, 2007.

TAX, D. M. J. **One-class classification**: concept-learning in the absence of counter-examples. 2001. Tese (Doutorado) - Delft University of Technology, Holanda, 2001.

VAN RIJSBERGEN, C. J. **Information retrieval**. 1979.

VANWINCKELEN, Gitte. **On Estimating Model Accuracy with Repeated Cross-Validation.** Lirias Kuleuven. p. 39–44, 2019. ISBN 9789461970442.

VILELA, P. DE C. S. **Classificação de sentimento para notícias sobre a Petrobras no mercado financeiro**. Tese (Doutorado), 2011.

WESNER, Francisco Benjamín. **Métodos de Detección Automática de Fraudes Informáticos por Suplantación de Identidad**. Trabalho de Conclussão de Curso (Tablho Final de Graduação). Universidad de Buenos Aires: 2020. Disponível em: http://bibliotecadigital.econ.uba.ar/download/tpos/1502-1712_WesnerFB.pdf. Acesso em: 25/02/2022.

WICKELMAIER, F. **An Introduction to MDS**. Sound Quality Research Unit, Aalborg University, 2003.

ZURINI, M.; e SBORA, C**. Clustering Analysis within Text Classification Techniques. Informática Econômica**. v. 15, n. 4, p. 178-189, 2011. Disponível em: http://revistaie.ase.ro/content/60/14%20-%20Zurini,%20Sbora.pdf. Acesso em: 25/04/2022.