UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

Prevendo a utilidade de comentários em Português Brasileiro de jogos no site Steam.

Germano Antonio Zani Jorge

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

Prevendo a utilidade de comentários em Português Brasileiro de jogos no site Steam.

Germano Antonio Zani Jorge

USP - São Carlos

2022

Germano Antonio Zani Jorge

Prevendo a utilidade de comentários em Português Brasileiro de jogos no site Steam.

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

Orientador: Prof. Dr. Thiago A. S. Pardo

USP - São Carlos

2022

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

A635p

Antonio Zani Jorge, Germano
Prevendo a utilidade de comentários em Português Brasileiro de jogos no site Steam / Germano Antonio Zani Jorge; orientador Thiago Alexandre Salgueiro Pardo. -- São Carlos, 2022.
65 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. Processamento de Linguagem Natural. 2. Mineração de Textos. 3. Word Embedding. 4. utilidade de comentários. I. Alexandre Salgueiro Pardo, Thiago, 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

RESUMO

JORGE, G. A. Z. **Prevendo a utilidade de comentários em Português Brasileiro de jogos no site Steam**. 2022. 52 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.

Os comentários do produto desejado podem ser um instrumento poderoso na decisão de compra pelo cliente, o que leva à importância da avaliação de utilidade dos comentários dos usuários de um produto. Este trabalho investiga métodos automáticos para previsão da utilidade de comentários de jogos no conhecido site Steam. Foi criada uma grande base de comentários em Português Brasileiro para diferentes gêneros de jogos e investigado um modelo de classificação e um modelo de regressão para prever se estes comentários são úteis ou não. Este trabalho também investiga a importância de diferentes atributos linguísticos e não-linguísticos para as previsões. Os algoritmos foram capazes de prever satisfatoriamente a utilidade baseada em um determinado limiar e os atributos que mais influenciaram foram a recomendação do jogo pelo usuário e o tamanho do comentário.

Palavras-chave: utilidade de comentários; classificação; regressão; *word embeddings*.

ABSTRACT

JORGE, G. A. Z. **Predicting helpfulness of Brazilian Portuguese game reviews on the Steam site**. 2020. 52 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, 2020.

The desired product reviews can be a powerful instrument on the client's buying decision, which leads to the importance of a product user reviews helpfulness evaluation. This paper investigates automatic methods for predicting the helpfulness of game reviews on the well-known Steam site. A large Brazilian Portuguese database was created with different game genres and classification and regression models were investigated to predict if these reviews were helpful or not. This paper also investigates the importance of different linguistic and non-linguistic features for predictions. The algorithms were able to satisfactorily predict the helpfulness based on a determined threshold and the most important features were the recommendation of the game by the user and the size of the review.

Keywords: classification; regression; word embedding.

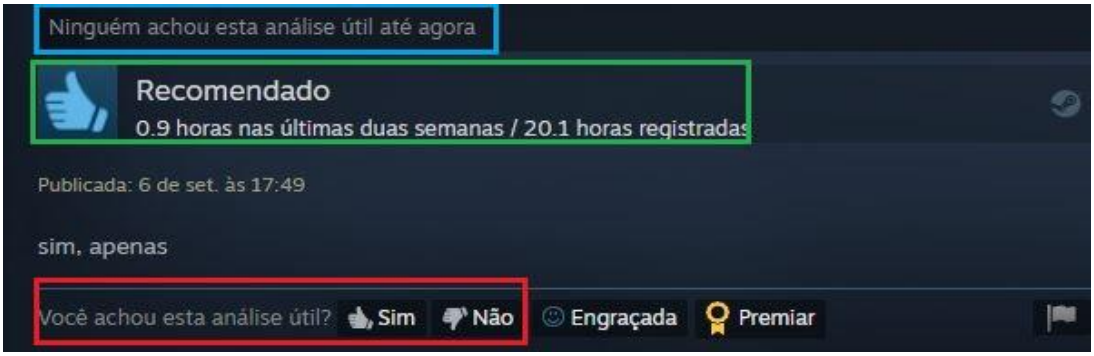
SUMÁRIO

1 INTRODUÇÃO

A grande quantidade e variedade de comentários (*reviews)* disponíveis em sites de *e-commerce*, filmes, viagens ou jogos torna importante a avaliação de utilidade (*helpfulness)* desses comentários (Bertaglia, 2017; Krishanamoorthy, 2015; Sousa, 2019; Zhang et al., 2006), de modo que aqueles considerados mais úteis para o leitor sejam dispostos nas primeiras posições. Demonstra-se significativa a contribuição dos usuários no fornecimento de descrições de qualidade do produto pesquisado pelo consumidor, uma vez que este, através dos comentários avaliados positivamente, pode verificar a presença ou ausência de atributos e características consideradas valorosas na decisão final da compra ou na aquisição de um serviço. Além disso, a classificação da utilidade das *review*s ajuda no combate aos *spams* e *reviews* falsas, já que rebaixa a posição daquelas avaliadas negativamente (Liu, 2012).

Embora diversas empresas já utilizem um sistema de avaliação de comentários, ainda é um processo manual no qual os leitores devem responder a uma pergunta como *"você achou esse comentário útil? "*. O usuário, então, realiza a classificação através de um sistema de votos de *like* ou *dislike*, que são totalizados e dispostos (e.g., "*10 pessoas acharam isso útil"*), como é o caso da famosa loja on-line de jogos Steam[1].

Figura 1 – Exemplo de comentário "não-útil" na Steam. Em vermelho destaca-se a opção de avaliação. O verde demonstra que o autor do comentário recomendou o jogo. Em azul, o total de usuários que marcaram a revisão como útil.

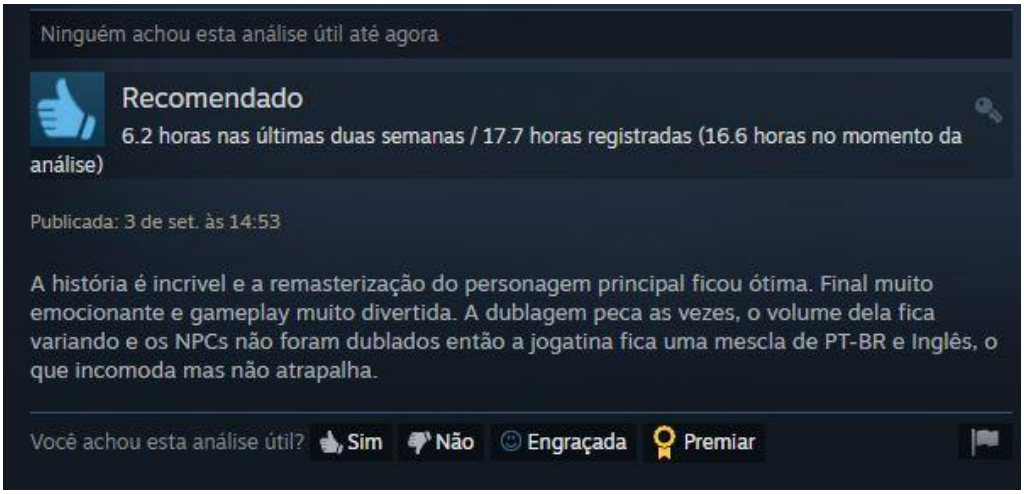


Fonte: https://store.steampowered.com/app/1817070/Marvels_SpiderMan_Remastered/. Acessado em 07 de setembro de 2022.

[1] https://store.steampowered.com/. Acessado em 13 de outubro de 2021.

Esse tipo de sistema é eficaz, porém, são necessários vários votos para que a utilidade do comentário seja classificada apropriadamente. Dessa forma, comentários com poucos votos ou escritos recentemente podem não ter sua utilidade devidamente avaliada, além do que produtos com poucas visualizações podem não conter um número suficiente de votos nas *reviews* para classificá-las como úteis (Kim et al., 2006; Liu, 2012; Sousa, 2019).

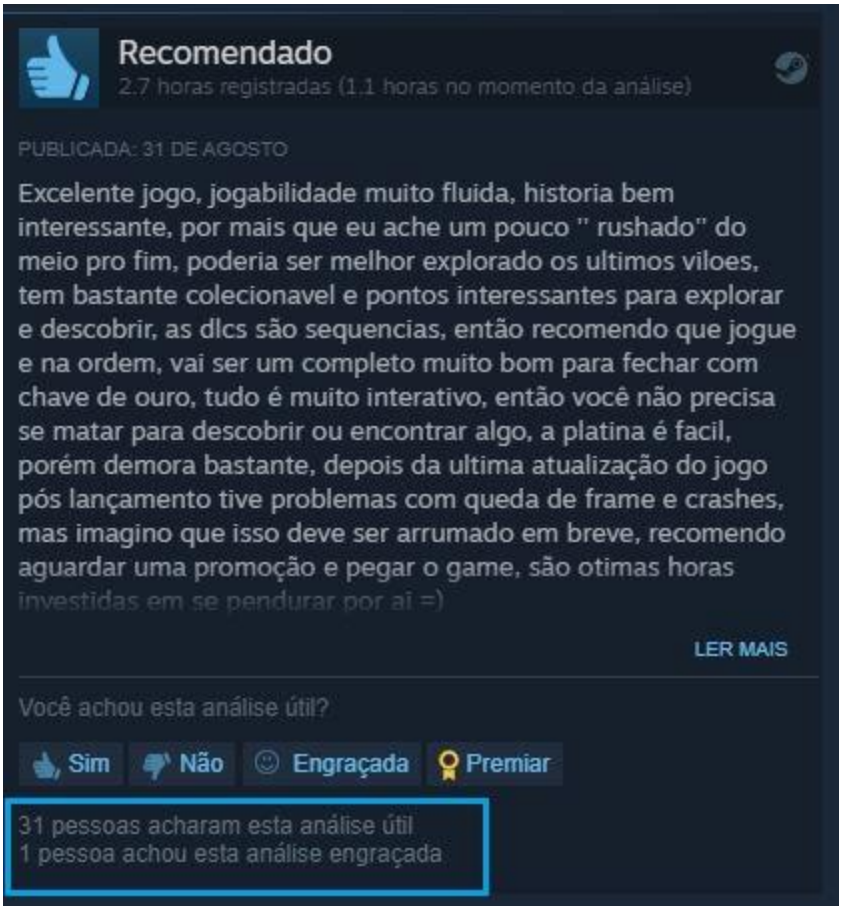
Figura 2 – Exemplo de comentário "não-útil" que contém informações relevantes, mas não possui avaliações de utilidade.



Fonte: https://store.steampowered.com/app/1817070/Marvels_SpiderMan_Remastered/. Acessado em 07 de setembro de 2022.

A automação da avaliação de utilidade de comentários pode acelerar a classificação destes, permitindo que os sites providenciem um feedback rápido para seus autores. Esta tarefa pode ser realizada através de um modelo de aprendizado supervisionado de máquina, resultando em um modelo de Predição de Utilidade (*Helpfulness Prediction)* (Sousa et al., 2019). O modelo se baseia num problema de regressão linear e requer recursos linguísticos que são utilizados para a atribuição de uma pontuação de qualidade para cada comentário (Liu, 2012). Para o Português Brasileiro, nota-se uma escassez nos conjuntos de dados disponíveis e na pesquisa em métodos automáticos para predição utilidade. Pode-se citar o UTLcorpus (Sousa et al., 2019) que é um conjunto recente composto de dois *corpora* anotados automaticamente para essa língua.

Figura 3 – Exemplo de comentário avaliado como útil pela comunidade. Destaque em azul para o número de votos.



Fonte: https://store.steampowered.com/app/1817070/Marvels_SpiderMan_Remastered/. Acessado em 07 de setembro de 2022.

Neste contexto, visando avançar as pesquisas nessa frente, o presente trabalho propõe o desenvolvimento de um sistema para automatizar a avaliação da utilidade de comentários da plataforma de jogos Steam, que será baseado em um modelo de aprendizado supervisionado de máquina com regressão linear e também em um modelo de classificação. Além disso, é proposta a criação de um *corpus* em Português Brasileiro contendo as *reviews* desta loja on-line, contribuindo, juntamente do UTLcorpus, ao aumento dos *corpora* de dados linguísticos da língua portuguesa para as pesquisas em Predição de Utilidade.

O próximo capítulo contém uma breve fundamentação teórica sobre a tarefa de mineração e seus tipos, além dos diferentes modelos de aprendizado de máquina e uma introdução à área de Processamento de Linguagem Natural, seguido de seu atual Estado da Arte.

No capítulo 3 são apresentados os objetivos como criar um modelo para avaliar a utilidade de comentários, descobrir quais atributos podem indicar um comentário útil e construir um corpus em Português Brasileiro, além da justificativa para esta monografia. O capítulo 4 contém a metodologia utilizada. No capítulo 5 são apresentados os experimentos e as análises realizadas. Por fim, são discutidas as conclusões e possíveis trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 TIPOS DE MINERAÇÃO

2.1.1 Mineração de Dados

A enorme quantidade de dados armazenada diariamente por faculdades, supermercados, lojas, hospitais e por sites on-line torna necessário o surgimento de um processo que manipule tais dados adequadamente para que ao menos uma parte de toda essa informação guardada possa transformar-se em conhecimento, e, dessa forma, tornar-se útil a determinados meios e fins. Nesse contexto, surge a mineração de dados, que conforme Kaufman (2005, p.5, tradução minha):

> "(...) é definido como o processo de descobrimento de padrões nos dados. O processo deve ser automático ou (mais comumente) semiautomático. Os padrões descobertos devem ser significantes de modo que levem a alguma vantagem, geralmente uma vantagem econômica".

Dessa forma, nota-se como a mineração de dados pode ser útil para a sociedade de hoje. Uma vez descobertos padrões nos dados acumulados, os últimos se transformam em informação, que se transforma em conhecimento para tomadas de decisão (Rezende, 2003). A mineração é constituída por etapas, conforme mostra a Figura 4.

Figura 4 – Etapas da Mineração de Dados

Fonte: Rezende, S. O. *Sistemas inteligentes: fundamentos e aplicações*. Editora Manole Ltda., 2003.

De acordo com Rezende (2003), no conhecimento do domínio ocorre a identificação do problema e é definido o objetivo da mineração, verificados quais os dados disponíveis e quais são as metas do processo.

A seguir, no pré-processamento, será decidido como representar os dados para a extração de padrões, realizando seu tratamento através da filtragem e limpeza. Dessa forma, são tratados valores ausentes e inconsistentes, realizando-se a normalização e padronização dos dados. Para isso, é comum a utilização de técnicas como a redução da dimensionalidade, análise de componentes principais (PCA) e o balanceamento de dados.

Para a extração de padrões, por sua vez, são utilizados algoritmos de aprendizado de máquina para que sejam obtidas regularidades nos dados, conforme sugere o nome da etapa. As tarefas realizadas pelos algoritmos podem ser preditivas ou descritivas. Nas do primeiro tipo, utiliza-se o aprendizado supervisionado, que possui métodos como árvores de decisão e modelos de regressão. Já as tarefas descritivas buscam a observação de similaridades nos dados através de seu agrupamento e também de regras de associação, utilizando o aprendizado não supervisionado (o aprendizado de máquina será descrito com mais detalhes posteriormente neste trabalho.).

É conferido, então, no pós-processamento, se as metas estabelecidas anteriormente foram atingidas. Isso irá depender dos critérios de avaliação, que por sua vez dependem do algoritmo de extração. Caso tenha sido utilizado um algoritmo de aprendizado supervisionado, por exemplo, os critérios analisados seriam acurácia, precisão e revocação, possivelmente com validação cruzada.

Por fim, há a utilização do conhecimento, ou seja, sua aplicação para tomadas de decisão em diversas áreas. No *marketing*, por exemplo, o comportamento dos clientes é analisado e propagandas mais adequadas podem ser apresentadas a eles; já na educação, o comportamento dos estudantes pode ser aprendido e a instituição pode refinar seus cursos e materiais.

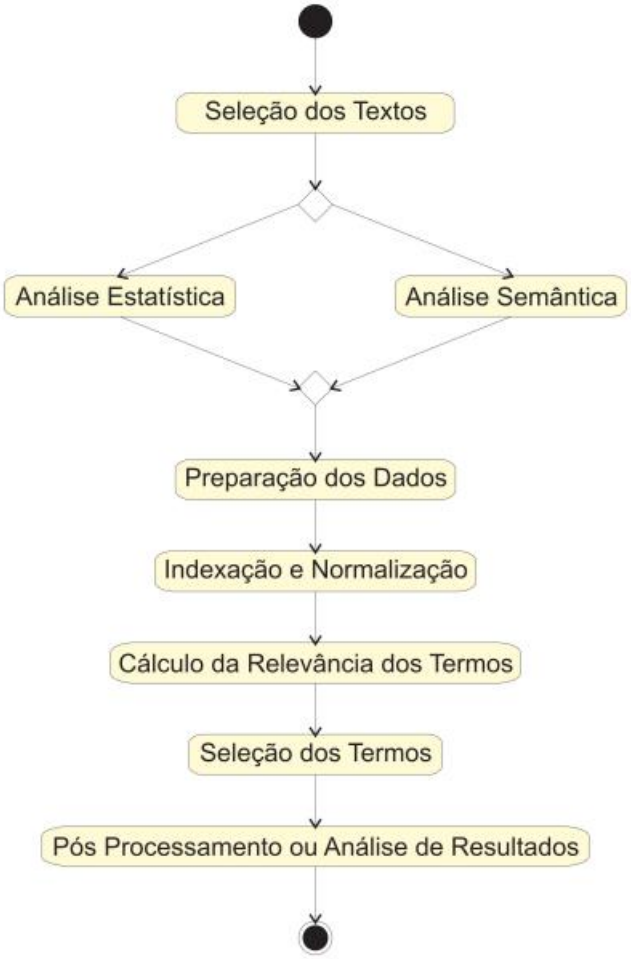
2.1.2 Mineração de Textos

Assim como a mineração de dados, a mineração de textos busca extrair padrões e regularidades de um conjunto de dados para obtenção de conhecimento. Contudo, enquanto a mineração de dados lida com dados estruturados para a construção de análises e modelos, a mineração de textos utiliza dados não estruturados. A diferença entre o primeiro tipo de dados e o último é que os dados estruturados geralmente são compostos

por bases de dados e *data warehouses*, seguindo um formato específico e um padrão, como tabelas com linhas e colunas, já os dados não estruturados são compostos por textos, frases ou palavras em língua natural, não considerando formatos como gráficos e figuras. Isto torna necessário o uso de técnicas específicas, enquadrando esta tecnologia no campo do Processamento de Linguagem Natural (PLN) e Aprendizagem de Máquina.

O processo de mineração pode ser dividido conforme a Figura 5.

Figura 5 – Etapas da Mineração de Textos



Fonte: Morais, E. A. M., & Ambrósio, A. P. L. *Mineração de textos.* Relatório Técnico –Instituto de Informática (UFG), 2007.

Conforme Morais (2007), na seleção de textos, os documentos de interesse são escolhidos.

A seguir, deve ser definido o tipo de abordagem dos dados, que, por sua vez, determinará se a análise será de cunho estatístico ou semântico. A análise semântica tem como intuito identificar a função dos termos no contexto dos textos, procurando identificar a importância das palavras dentro da estrutura de suas orações. Para isso, empregam-se técnicas fundamentadas no Processamento de Linguagem Natural, que avaliam a sequência de termos. Já a análise estatística efetua uma codificação e estimativa dos dados e modelos de representação de documentos para verificar a importância de um termo através do número de vezes que este aparece no texto.

A etapa de preparação dos dados tem como objetivo identificar similaridades morfológicas ou semânticas dos termos nos textos para realizar uma redução dimensional, selecionando um núcleo dentro da base de textos que melhor expressa o conteúdo destes.

Na indexação e normalização, as palavras-chave de um documento são colocadas em um índice a fim de facilitar a identificação de similaridade de significado entre as palavras dos textos. Caso sejam simples, os termos são identificados através de um *parser* (no caso, um analisador léxico). Caso sejam compostos, são identificados através de um dicionário de expressões. Em seguida, são removidas as *Stopwords*: palavras não relevantes para a tarefa, como preposições, pronomes, artigos e palavras cuja frequência é muito alta. Por fim, ocorre a normalização morfológica (*Stemming*), que elimina as variações morfológicas de uma palavra retirando os prefixos e sufixos de seu radical.

O cálculo da relevância atribui um peso às palavras mais importantes de um texto. Geralmente, este cálculo é baseado em fórmulas de frequência da palavra como *frequência relativa, frequência absoluta (TF),* e *frequência inversa de documentos (IDF).*

Após o pré-processamento e cálculo da relevância, as palavras retiradas do texto são selecionadas no processo de seleção de termos de acordo com sua posição sintática em relação ao texto, ou com seu peso calculado anteriormente. Algumas técnicas desse processo são: filtragem baseada no peso do termo, seleção baseada no peso do termo, seleção por análise de co-ocorrência, seleção por *Latent Semantic Indexing* (LSI) e seleção por análise de linguagem natural.

Por fim, o desempenho do sistema de recuperação de informações é avaliado através de métricas como o *recall, precision, fall-out* e *effort*. Com isso, são informados ao usuário quantos e quais documentos são relevantes e sua importância no contexto.

As aplicações da mineração de texto são inúmeras. Através da análise de relatórios médicos e documentos, muitas conclusões podem ser encontradas para o campo da

medicina e saúde pública. A mineração de SMSs (*Short Message Service*), microblogs e outras redes de informações utilizadas diariamente pela população podem ser analisadas por departamentos do governo para investigar a opinião pública sobre determinado assunto. No campo do comércio, pode-se prever a situação econômica e a tendência do mercado da bolsa de valores com a mineração de notícias, relatórios financeiros e *reviews* on-line. Além disso, empresas podem adquirir um *feedback* sobre seus produtos e obter dados para melhoria de qualidade, além de fornecer serviços personalizados ao cliente (Zong et al., 2021).

2.1.3 Mineração de Opiniões

A mineração de opiniões é uma subárea da mineração de textos que avalia computacionalmente opiniões, sentimentos, e atitudes expressas de forma textual. Os termos para análise podem ser retirados de diferentes fontes como sites de *e-commerce*, sites de avaliações, blogs, ou qualquer outro tipo de mídia social. Para analisa-los, este ramo utiliza técnicas das áreas de Processamento de Linguagem Natural, Recuperação de Informações, Estatística, Inteligência Artificial e outras.

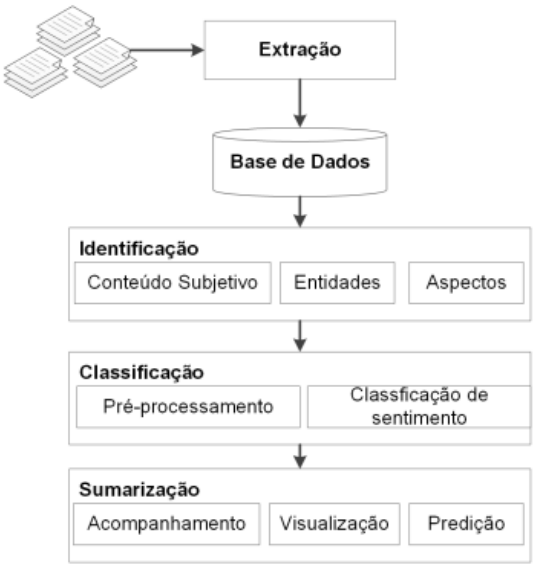
Enquanto a mineração de textos busca estruturar dados não estruturados para mostrar o que foi escrito por clientes sobre um produto ou serviço, por exemplo, a mineração de opinião permite entender se os clientes estão avaliando o produto positiva ou negativamente, ou ainda, permite entender o que torna uma avaliação positiva, negativa, ou neutra, fornecendo as ferramentas necessárias para esse entendimento.

Conforme Liu (2012), uma informação relevante sobre uma opinião pode ser extraída em três níveis textuais. Por exemplo, na análise de polaridade, tem-se o de documento, que avalia o documento como um todo para verificar se este expressa um sentimento positivo ou negativo; o da sentença, que determina o sentimento de uma sentença específica do documento; o da Entidade e Aspecto, que foca na opinião expressa independentemente do documento, sentença ou oração.

De maneira similar aos dois processos de mineração vistos anteriormente, a mineração de opinião também é dividida em etapas. Becker & Tumitan (2013) as separam em: identificação, classificação e sumarização. De acordo com os autores, na primeira etapa identificam-se as entidades e possivelmente seus aspectos e sentimentos. Na segunda, o sentimento é avaliado como positivo ou negativo após serem utilizadas técnicas como a de reconhecimento de n-gramas, a de extração de atributos, a eliminação de termos

irrelevantes e outras. A última etapa, da sumarização, consiste em analisar uma grande quantidade de opiniões com um mesmo alvo através de métricas e sumários para descobrir o sentimento geral do público em relação àquela entidade específica.

Figura 6 – Etapas da Mineração de Opiniões



Fonte: Becker, K., & Tumitan, D. *Introdução à mineração de opiniões: Conceitos, aplicações e desafios*. Simpósio Brasileiro de Banco de dados 75, 2013.

A mineração de opinião pode ser aplicada para prever eleições e aperfeiçoar os resultados de pesquisa de opinião contendo análises automáticas, rastrear tópicos políticos para tomada de decisão, analisar e prever o comportamento da bolsa de valores, prever arrecadação de bilheterias de filmes, e para definição de preços de produtos e serviços para controle de qualidade, em que a opinião do usuário pode ser analisada para aumentar a qualidade do produto, entre muitas outras aplicações.

2.2 PROCESSAMENTO DE LINGUAGEM NATURAL

> "O Processamento de Linguagem Natural emprega técnicas computacionais com o propósito de aprender, entender e produzir conteúdo da língua humana. As primeiras abordagens computacionais para a pesquisa humana focaram em automatizar a análise da estrutura linguística da língua e desenvolver tecnologias básicas como tradução de máquinas, reconhecimento

> de fala, e síntese do discurso. Os pesquisadores de hoje refinam e fazem uso de tais ferramentas em aplicações do mundo real, criando sistemas de diálogos falados e mecanismos de tradução fala-para-fala, mineração de mídias sociais para informações sobre saúde ou finanças, identificando sentimentos e emoções quanto a produtos e serviços. ”
>
> (HIRSCHBERG & MANNING, 2015, p.261-266, tradução minha).

A partir da citação acima, pode-se considerar a área de Processamento de Linguagem Natural como multidisciplinar, perpassando as subáreas da Inteligência Artificial, como Aprendizado de Máquina, Ciência de Dados, Mineração de Dados, Mineração de Textos e, sobretudo, Mineração de opiniões, assim como a Linguística e a Estatística, pois há muito foco hoje nas aplicações que trabalham com dados produzidos por usuários da web.

As opiniões tornam-se cada vez mais úteis para as empresas e para os consumidores, podendo ajudar tanto na melhoria de qualidade e serviço no caso do primeiro, quanto na escolha do produto final pelo segundo.

Ao entrar em um site de compras, por exemplo, é comum haver comentários de usuários avaliando produto. Não só isso, mas os próprios comentários também podem ser avaliados por outros usuários, que os identificam como úteis ou não através de uma opção por botão ou um sistema de ranqueamento por estrelas.

O Processamento de Linguagem Natural, por sua vez, pode ser capaz de prever, com certa eficácia, a utilidade de comentários. Para isso, analisam-se atributos textuais como os derivados de análise de léxico, sintaxe, semântica e de metadados (como a data da escrita do comentário, a reputação do autor no site, e a quantidade de votos de utilidade já feita). Para um site de filmes, por exemplo, o algoritmo de predição de utilidade poderia dispor comentários úteis ao usuário para que este escolhesse se vai assistir ou não.

Este tipo de predição pode ser útil uma vez que pode automatizar o sistema de avaliação que hoje é manual. Neste sistema atual, comentários recentes ou com poucas avaliações do usuário podem se perder ou serem deixados de lado mesmo sendo úteis. Este tipo de problema é nomeado como “partida fria” (Fressato, 2019) na área de recomendação de produtos.

2.3 *WORD EMBEDDINGS:* REPRESENTANDO PALAVRAS COM VETORES

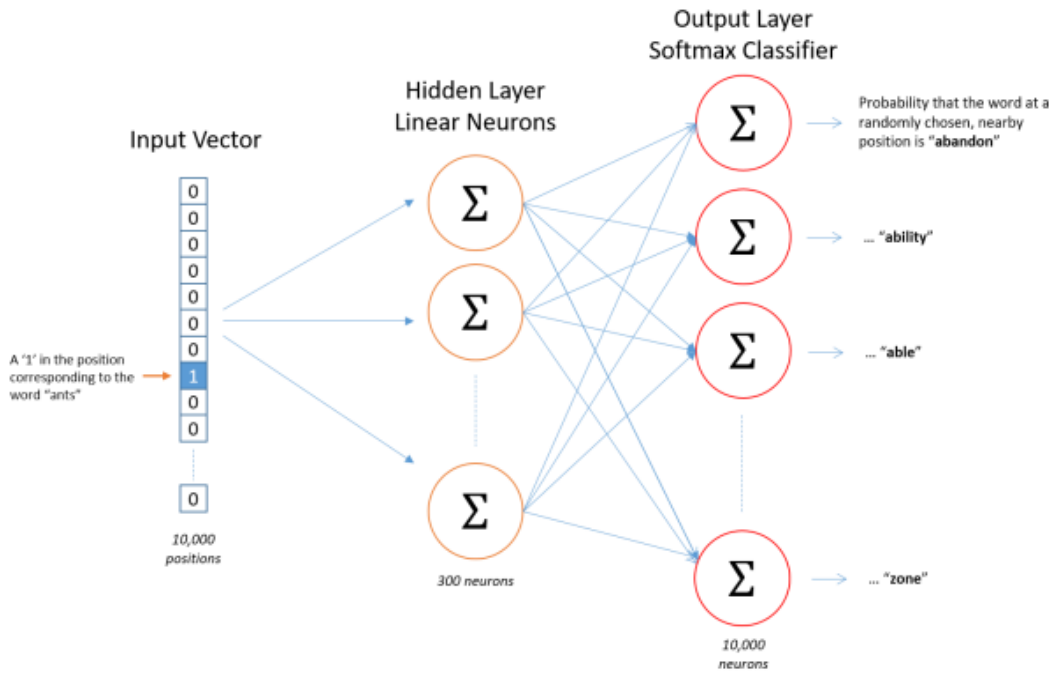
As redes neurais vêm sendo de grande uso para a área do Processamento de Linguagem Natural. É comum nessa área o uso de algoritmos com o MLP e de arquiteturas neurais na criação de *word embeddings*.

McCormick (2019) explica que uma *word embedding* é um vetor, um ponto num espaço dimensional ou simplesmente um arranjo de frações, que "incorpora" o significado da palavra tornando possível calcular uma pontuação de similaridade para qualquer par de palavras.

Há várias formas de aprendizado desses vetores, dentre elas, o Word2Vec (Mikolov et al., 2013) é um algoritmo que treina uma rede neural rasa com apenas uma camada oculta (*hidden layer)* para realizar uma determinada tarefa. Contudo, a rede é utilizada apenas para aprender os pesos da camada oculta. Em razão disso, este processo é nomeado como tarefa falsa (*fake task)*. Dessa forma, treina-se a rede para escolher aleatoriamente uma das palavras próximas da sentença dada a palavra de entrada, o que resultará numa saída com a probabilidade para cada palavra no vocabulário. Por exemplo, se a palavra de entrada for "banana", as probabilidades de saída para palavras como "macaco" e "comi" serão muito maiores do que para "canguru".

Este vocabulário que será utilizado é criado através da representação das palavras como *one-hot-vectors*. Em um vocabulário com 10,000 palavras distintas, esse vetor conterá um "1" na posição da palavra escolhida e 0s no restante das posições. A saída da rede também será apenas um vetor que conterá a probabilidade de cada palavra do vocabulário ser a palavra correta. Este exemplo é demonstrado na figura a seguir.

Figura 7 – Arquitetura de uma rede neural de uma *word embedding*.



Fonte: McCormick, Chris. *The inner workings-of-word2vec*, 2019.

Assim, num vetor com 300 dimensões, a camada ocultada é representada por uma matriz de pesos com 10.000 linhas e 300 colunas (uma para cada neurônio). Multiplicando o vetor *one-hot* por essa matriz é possível verificar qual linha desta corresponde à palavra de entrada.

Figura 8 – Matriz de uma camada ocultada.

$$[0 \quad 0 \quad 0 \quad 1 \quad 0] \times \begin{bmatrix} 17 & 24 & 1 \\ 23 & 5 & 7 \\ 4 & 6 & 13 \\ 10 & 12 & 19 \\ 11 & 18 & 25 \end{bmatrix} = [10 \quad 12 \quad 19]$$

Fonte: McCormick, Chris. *The inner workings-of-word2vec*, 2019.

A figura anterior é um exemplo ilustrativo (com um número diferente de linhas e colunas) de como é possível descobrir a *embedding* desejada (destacada em verde) para a palavra de entrada (marcada como "1" no vetor *one-hot*), através da multiplicação do vetor pela matriz completa da camada oculta. A linha da matriz (em verde) é, portanto, o vetor de palavra (*word vector)*. Como já dito anteriormente, um vetor de palavra nada mais é do que um arranjo de frações, como representado pela próxima figura.

Figura 9 – Representação do vetor da palavra "shop" emitida por uma função "*get*" em um ambiente de programação.

```
embeddings_index.get("shop")
```

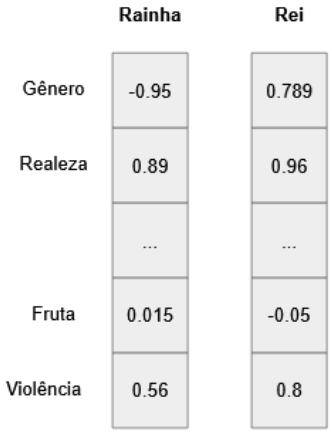
```
array([ 3.0426e-01, -1.4191e-01, -7.9738e-01, -3.5484e-01,  3.0333e-01,
        4.3690e-01, -9.8706e-02,  6.9080e-01,  6.9362e-01,  1.8528e-01,
        1.0648e-01, -4.5209e-01,  8.7568e-01,  1.1414e-01, -2.8514e-01,
        6.0731e-01,  2.7596e-01,  2.3698e-01, -7.1692e-01,  1.6804e-01,
        4.3669e-01,  4.1931e-01,  2.1568e-01, -1.2316e+00,  3.7208e-01,
       -9.0922e-02, -3.8767e-01, -7.0817e-01, -2.4242e-01, -7.2018e-02,
       -3.8969e-01,  5.2464e-01,  2.1317e-01,  8.8327e-02,  6.6017e-04,
        6.7755e-01, -3.3464e-01, -6.1269e-01,  8.2305e-01, -1.4450e+00,
        8.5966e-01, -4.6323e-01, -1.3172e-02, -8.1801e-01,  1.7294e-02,
        1.7025e-01, -6.3946e-01,  4.8516e-01,  6.1706e-01, -3.5333e-01,
       -1.7953e-01,  4.8890e-03, -4.7809e-01,  5.8311e-01, -4.2821e-01,
       -1.7160e+00, -1.3190e+00,  9.0167e-02,  1.3612e+00,  2.2214e-01,
        2.1325e-01,  1.5207e-01,  2.9252e-01,  5.7116e-01, -2.3654e-01,
       -1.4311e-01,  1.2564e+00, -1.6377e-01,  6.9895e-02, -3.2884e-01,
       -4.3554e-01,  4.1309e-01, -1.4767e-01, -4.0058e-01,  2.1931e-01,
        1.9361e-01,  6.5205e-01, -2.0986e-01, -5.8788e-01, -1.4051e-01,
        1.2399e-01, -8.9099e-03, -1.5384e-01, -4.6232e-02, -6.4600e-01,
       -3.1246e-01, -1.4165e-01, -7.6865e-01, -2.7654e-01, -7.6462e-03,
        6.9244e-01,  3.6744e-01,  1.0840e+00, -2.4375e-01, -8.9562e-01,
       -2.3390e-01,  1.4788e-01,  1.3795e-01,  1.2635e+00,  1.0817e-01],
      dtype=float32)
```

Fonte: https://blog.paperspace.com/pre-trained-word-embeddings-natural-language-processing/. Acessado em 07/09/2022

Estes vetores possuem um valor semântico obtido pelo contexto das palavras. Cada valor na matriz é chamado de dimensão e representa uma escala contendo uma informação.

A figura posterior mostra os vetores das palavras "Rainha" e "Rei". As duas contém as mesmas escalas (as dimensões) com valores diferentes, cada uma dessas representando uma informação como gênero ou realeza.

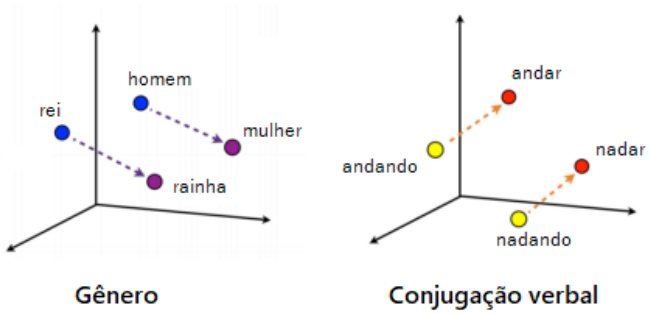
Figura 10 – Dimensões das palavras "Rainha" e "Rei"



Fonte: https://medium.com/turing-talks/word-embedding-fazendo-o-computador-entender-o-significado-das-palavras-92fe22745057. Acessado em 07/09/2022

É possível notar que a dimensão "Realeza" possui valores muito semelhantes, enquanto que em "Gênero" eles se distanciam, como esperado. Esses valores também podem ser colocados em um plano como na Figura 11.

Figura 11 – Representação das dimensões de "Rei", "Rainha" e "Andar", "Nadar".



Fonte: https://medium.com/turing-talks/word-embedding-fazendo-o-computador-entender-o-significado-das-palavras-92fe22745057. Acessado em 07/09/2022

Na figura, nota-se que "rei" aponta para "rainha" assim como homem aponta para mulher, enquanto que "rei" e "homem" ocupam uma posição semelhante a "rainha" e "mulher". Essas distâncias entre vetores podem ser calculadas através da **similaridade de cossenos** e/ou da **distância euclidiana** (McCormick, 2019)

No ano seguinte à publicação do Word2Vec, Mikolov e Le (2014) propuseram o Doc2Vec. Trata-se de uma simples extensão do Word2Vec que aumenta o aprendizado de palavras para, então, sequências de palavras (Lau e Baldwin, 2016), como por exemplo frases e parágrafos. Assim, é possível extrair diferentes dimensões de um comentário na Steam e usá-las para compará-lo com outro, afim de tentar descobrir informações semelhantes que tornam um comentário útil (ou não-útil).

Existem também modelos de *embeddings* mais recentes, que são baseados no *Transformer* (Vaswani et al., 2017), uma arquitetura de modelo capaz de entender as relações entre as palavras em uma frase como um todo, ao invés de uma por uma e em ordem. É o caso BERT (Devlin et al., 2018) e o ELMo (Peters et al., 2018). Enquanto no Word2Vec as *embeddings* são independentes de contexto, a codificação bidirecional permite que o modelo processe a posição de cada palavra em uma sequência para gerar o vetor de palavra, diferentemente do modelo mais antigo (McCormick, 2020). Contudo, o foco desta monografia permanece no Word2Vec, uma vez que visa a reproduzir os métodos de Baowaly et al., (2019) e comparar os resultados.

2.4 APRENDIZADO (S) DE MÁQUINA

Conforme visto anteriormente, o aprendizado de máquina faz parte do processo da extração de padrões na mineração de dados. Utilizando a definição de Alpaydin (2004, p.3, tradução minha): "Aprendizado de máquina é programar computadores para aperfeiçoar o critério de desempenho utilizando dados ou experiência passada". Chollet (2017, p.3, tradução minha) esclarece:

> "Tradicionalmente, a engenharia de software combinou regras criadas por humanos com dados para criar respostas para um problema. O aprendizado de máquina, ao invés, usa os dados e as respostas para descobrir as regras por trás de um problema".

Segundo Alpaydin (2004), a partir de um modelo definido com certos parâmetros, o aprendizado ocorre quando o computador aperfeiçoa estes parâmetros usando os dados de treinamento ou a experiência passada. Esse modelo pode ser preditivo, para prever o futuro, ou descritivo, para ganhar conhecimento a partir dos dados.

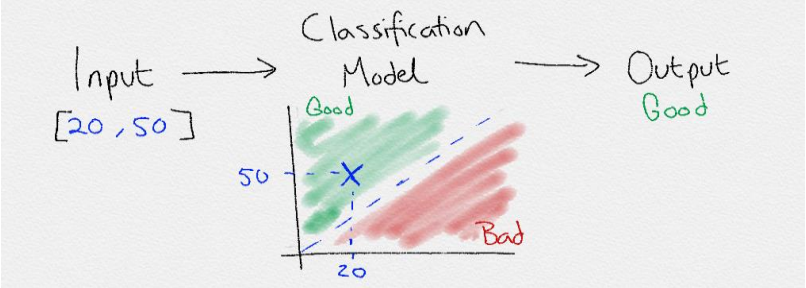
2.4.1 Aprendizado supervisionado

São comuns no modelo descritivo do aprendizado supervisionado os métodos de classificação e regressão. A análise dos dois fornece um bom entendimento para este tipo de aprendizado. Conforme Alpaydin (2004, p.8, tradução minha): "Ambos regressão e classificação são problemas de aprendizado supervisionado, onde há uma entrada, X, e uma saída, Y, e a tarefa é aprender o mapeamento (as regras) da entrada a saída".

Dessa maneira, supondo que as entradas sejam a previsão do tempo e as saídas o número de banhistas na praia; o aprendizado supervisionado visa aprender o mapeamento que "descreve" a relação entre a temperatura e o número de visitantes na praia, daí o nome de modelo "descritivo". Já o nome "supervisionado" é dado pelo fato dos pares de dados entrada e saída serem fornecidos previamente ao modelo para ensiná-lo como se comportar.

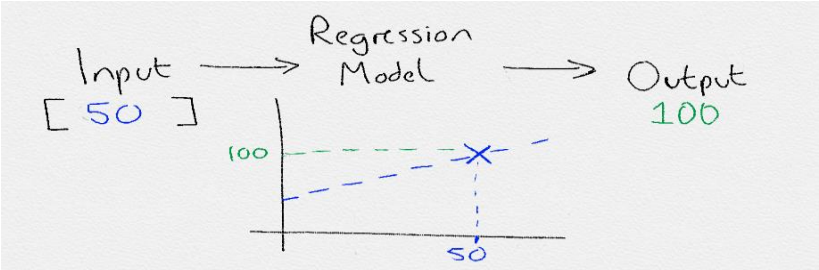
Caso a entrada seja uma **categoria** de um conjunto finito, como [baixo, médio, alto], trata-se de uma classificação (Figura 12). Caso seja um **número escalar,** trata-se de uma regressão (Figura 13).

Figura 12 – Exemplo de Classificação



Fonte: Towards Data Science[2]

Figura 13 – Exemplo de Regressão



Fonte: Towards Data Science[3]

Alguns exemplos de algoritmos muito utilizados para aprendizado de máquina supervisionado incluem *Support Vector Machines* (SVM), Árvores de decisão (Decision Trees) e *Random Forest* (RF).

2.4.2 Aprendizado não-supervisionado

Diferentemente do aprendizado supervisionado, em que são fornecidos a entrada e a saída, no aprendizado não supervisionado são fornecidos apenas os dados de entrada. "O objetivo é encontrar regularidades ou padrões na entrada. Há uma estrutura no espaço da entrada que contém certos padrões que ocorrem mais do que outros" (Alpaydin, 2004, p.10, tradução minha). Para encontrar essas estruturas, utilizam-se técnicas como a Clusterização, Associação, e Redução de Dimensionalidade.

---

[2] Disponível em: <https://towardsdatascience.com/machine-learning-an-introduction-23b84d51e6d0>. Acesso em: 14 fev. 2022.

[3] Disponível em: <https://towardsdatascience.com/machine-learning-an-introduction-23b84d51e6d0>. Acesso em: 14 fev. 2022.

Na Clusterização, identificam-se "subgrupos" (*clusters)* cujas características os diferenciam entre si. Kaufman (2005, p. 136, tradução minha) providencia uma clara definição sobre esse tipo de técnica e indica quando usá-la:

> "As técnicas de agrupamento são aplicadas quando não há classe a ser predita, mas quando as instâncias estão para ser divididas em grupos naturais. Estes clusters provavelmente refletem algum mecanismo que atua no domínio cujas instâncias foram retiradas, um mecanismo que faz com que algumas instâncias possuam uma semelhança mais forte entre si do que as demais instâncias."

A associação, por sua vez, tem como objetivo verificar os dados e descobrir quais regras podem descrevê-los. Conforme Alpaydin (2004, p.3, tradução minha):

> "No descobrimento de uma regra de associação, estamos interessados em aprender uma probabilidade condicional da forma P-(Y|X) onde Y é o produto que gostaríamos de condicionar em X, que é o produto ou conjunto de produtos que sabemos que o cliente já comprou".

Assim, em um exemplo onde P(batata|cerveja)=0,7, significa que 70% de clientes que compram cerveja também compram batata.

Um fator de extrema importância ao se tratar de algoritmos é a questão da dimensionalidade. A complexidade de um algoritmo depende tanto do número de dimensões de entrada quanto do tamanho da amostra de dados. Portanto, a redução de dimensionalidade pode auxiliar na redução desta complexidade ao diminuir a memória e computação necessária e salvar o custo de extração de uma entrada desnecessária, reduzindo o número de atributos e, dessa forma, proporcionando uma ideia melhor do processo que ocorre nos dados para extração do conhecimento e sua visualização para estruturas e *outliers* (ALPAYDIN, 2004, p.106).

2.4.3 Outros tipos de aprendizados

Há ainda mais dois tipos de aprendizado de máquina: o semisupervisionado e o por reforço. O primeiro, como o nome sugere, trata-se de um modelo intermediário entre o aprendizado supervisionado e o não supervisionado. A razão para isso é que utiliza tanto dados anotados quanto dados não anotados. Chama-se de anotação um esquema como o citado anteriormente, no aprendizado supervisionado, em que se deseja obter uma saída *Yi* a partir de uma entrada *Xi*, ou seja, quando há uma resposta desejada. A anotação de dados é um processo caro, que consome bastante tempo e é propenso a erros (Haykin,

2008). Portanto, unir os dois tipos de dados no processo semisupervisionado é uma solução elegante.

O aprendizado por reforço também jaz entre o aprendizado supervisionado e o não supervisionado. Segundo Bhatia (2020, p.68, tradução minha) "os reforços são considerados como uma série de *feedbacks* na qual o algoritmo aprende e recompensas ou punições são dadas conforme o modelo aprende com suas decisões". Haykin (2008) afirma que o aprendizado por reforço acontece quando há contínua interação com o ambiente para que se minimize o custo de ações tomadas com uma sequência de passos. O modelo realiza uma ação e aprende com a resposta do ambiente sobre essa ação. Dessa forma, ao invés de haver um supervisor, é como se houvesse uma crítica para que o sistema seja melhorado.

Estas duas últimas formas de aprendizado, sobretudo a de reforço, são de certa forma parecidas com a dos humanos.

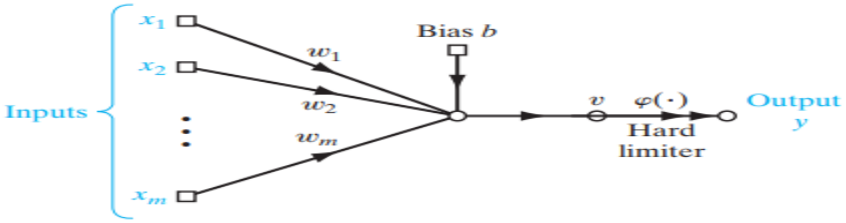
2.5 REDES NEURAIS

Uma área que se baseia no cérebro humano para criação de um processador e vem realizando grandes contribuições para o aprendizado de máquina é a de redes neurais. De acordo com Haykin (2008), uma rede neural é uma máquina que é construída para modelar a maneira na qual o cérebro desempenha uma tarefa em particular ou função de interesse, geralmente através de um processo de aprendizagem. Sobre a motivação de utilizar o cérebro como referência, Haykin (2008, p.1, tradução minha) afirma:

> "O cérebro é um computador extremamente complexo, não linear e paralelo (sistema de processamento de informação). Ele possui a capacidade de organizar seus constituintes originais, conhecidos como neurônios, assim como desempenhar certas computações (e.g., reconhecimento de padrões, percepção e controle motor) muitas vezes mais rápido do que o computador digital mais rápido existente hoje. "

O primeiro modelo consagrado nas redes neurais foi o Perceptron de Rosenblatt (1958)[4], criado com base na estrutura de um neurônio do cérebro humano.

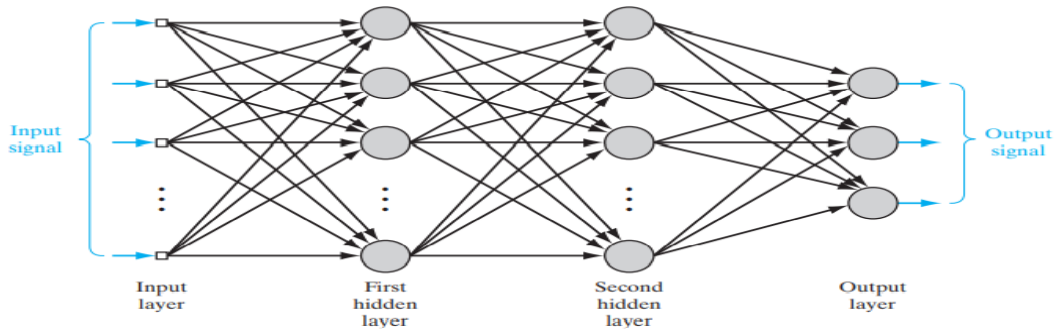
Figura 14 – Perceptron de Rosenblatt (Haykin, 2008)



Fonte: Haykin, Simon. *Neural Networks – A Comprehensive Foundation*. Prentice Hall, 2008.

Este modelo foi criticado por Minsky & Selfridge (1961), onde os autores afirmaram que o perceptron não era capaz de realizar generalizações de pares binários, limitando-se a uma linearização (Haykin, 2008). Contudo, com o surgimento do perceptron de multicamadas (*Multilayer Perceptron)*, doravante MLP, o problema da linearidade foi resolvido e uma nova forma de aprendizado criada.

Figura 15 – Perceptron Multicamadas



Fonte: Haykin, Simon. *Neural Networks – A Comprehensive Foundation*. Prentice Hall, 2008.

[4] Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. Psychological Review, 65(6), 386–408

2.6 ESTADO DA ARTE

Sousa et al. (2019) criaram um corpus em Português Brasileiro, o UTLCorpus, com anotações para predição de utilidade e também para classificação de polaridade. Os dados foram extraídos de comentários de usuários do Filmow, um site brasileiro de revisão de filmes, e de comentários em aplicativos na Google App Store. Para a tarefa de predição de utilidade, utilizaram um método *baseline* e três classificadores de aprendizado de máquina: *Support Vector Machines*, *Multi-layer Perceptron* e *Random Forest*. O *baseline* foi montado com cada sentença representada por um vetor de 2 dimensões com o número de termos positivos e negativos usando um léxico de sentimentos de Português Brasileiro. Em seguida, foi treinado um modelo SVM usando a representação mencionada anteriormente e os conjuntos de dados foram avaliados em um esquema de validação cruzada de 10 pastas (*10-fold cross-validation*). Além disso, foram utilizados mais três classificadores treinados e avaliados no mesmo tipo de validação. Terminado o *baseline*, os outros três classificadores restantes receberam dados representados através de *embeddings* do modelo word2vec com 600 dimensões pré-treinado. A classificação de polaridade, por sua vez, foi feita com o mesmo método, contudo, a diferença principal foi que para o *baseline* foi prevista como positiva qualquer sentença com mais termos positivos do que negativos.

Tabela 1 – Resultados de detecção de Utilidade

Classifier	Helpfulness Prediction			Polarity Classification		
	F1-Help	F1-No-Help	F1-Measure	F1-Pos	F1-Neg	F1-Measure
Baseline	0.6708	0.4583	0.5645	0.1570	0.6179	0.3874
Linear SVM	0.6299	0.6759	0.6529	0.7011	0.7578	0.7294
MLP	0.6383	0.6863	**0.6623**	0.7304	0.7646	0.7475
Random Forest	0.6398	0.6834	0.6616	–	–	–

Fonte: de Sousa, R.F. et.al. *A bunch of helpfulness and sentimento corpora in Brazilian Portuguese*, 2019.

Na Tabela 1 são mostrados os resultados dos corpora mesclados. É possível notar que tanto para a predição de utilidade (*Helpfulness Prediction*) quanto para a classificação de

polaridade (*Polarity Classification*), o maior valor de F1 foi obtido através do modelo *Multi-layer Perceptron* (MLP), atingindo 0.66 e 0.74, respectivamente.

Em Kim et al. (2006), foi proposto um método de predição de utilidade de comentários em Inglês em produtos contidos no site de vendas Amazon.com. Neste estudo, foi utilizado um algoritmo de regressão SVM e foram analisadas diferentes classes de *atributos*: estruturais, sintáticas, semânticas, lexicais e de meta-dados, com o intuito de verificar quais dessas classes são mais importantes para capturar a utilidade de comentários.

Os atributos estruturais dizem respeito ao comprimento *(LEN)* de uma sentença, o número de sentenças *(r)* e à formatação HTML *(HTM)*; os lexicais referem-se ao número de palavras observadas nos comentários, calculado com a fórmula estatística *tf-df* dos *Unigramas* (UGR) e *Bigramas (BGR)*; os sintáticos *(SYN)* incluem a porcentagem de tokens que são substantivos, adjetivos, adjetivos ou advérbios; os semânticos tratam dos atributos do produto *(PRF)* e das palavras com sentimentos positivos ou negativos; os de meta-dados capturam observações que são independentes do texto, como o número de estrelas *(STR)* que um comentário pode receber por usuários na sua avaliação.

Depois de extraídos os atributos, uma função de utilidade *h* é definida a partir da Equação 1, em que *rating+(r)* é o número de usuários que avaliaram o comentário *r* como útil e *rating-(r)* como inútil. Os comentários formam um conjunto de dados padrão-ouro de pares *{review, h(review)}* que podem ser utilizados para treinar um algoritmo de aprendizado de máquina supervisionado. Os autores utilizaram o SVM nos atributos extraídos dos comentários para aprender a função *h*. Depois de o SVM ser treinado para um produto e seu conjunto de comentários *R*, os autores ranqueiam os comentários de *R* em ordem decrescente de $h(r), r \in R$.

(4)

$$h(r \in R) = \frac{rating_{+}(r)}{rating_{+}(r) + rating_{-}(r)}$$

Para o ranqueamento do desempenho do algoritmo, foi adotado o coeficiente de Spearman ρ, de modo que para cada pasta nos experimentos de validação cruzada foi treinado um SVM usando 9 pastas e na restante foram ranqueados os comentários de acordo com a predição de SVM descrita previamente. Por fim, foi computada a correlação ρ entre este ranqueamento e o ranqueamento padrão-ouro da pasta de teste.

Tabela 2 – Resultados da avaliação de combinação de *features*

Feature Combinations	*MP3 Players*		*Digital Cameras*	
	Spearman†	*Pearson*†	*Spearman*†	*Pearson*†
LEN	0.575 ± 0.037	0.391 ± 0.038	0.521 ± 0.029	0.357 ± 0.029
UGR	0.593 ± 0.036	0.398 ± 0.038	0.499 ± 0.025	0.328 ± 0.029
STR1	0.589 ± 0.034	0.326 ± 0.038	0.507 ± 0.029	0.266 ± 0.030
UGR+STR1	0.644 ± 0.033	0.436 ± 0.038	0.490 ± 0.032	0.324 ± 0.032
LEN+UGR	0.582 ± 0.036	0.401 ± 0.038	0.553 ± 0.028	0.394 ± 0.029
LEN+STR1	0.652 ± 0.033	0.470 ± 0.038	0.577 ± 0.029	0.423 ± 0.031
LEN+UGR+STR1	**0.656 ± 0.033**	**0.476 ± 0.038**	**0.595 ± 0.028**	**0.442 ± 0.031**

LEN=*Length*; UGR=*Unigram*; STR=*Stars*
†95% confidence bounds are calculated using 10-fold cross-validation.

Fonte: Kim, S. M., et.al. *Automatically assessing review helpfulness*. Proceedings of the 2006 Conference on empirical methods in natural language processing, 2006

É possível notar nos resultados exibidos na Tabela 2 que a combinação de *atributos* com melhor desempenho foi a de comprimento, unigramas e estrelas, com 0,656 para MP3 players e 0,595 para Câmeras Digitais.

Barbosa et al. (2016) também propõem um método para automatizar a avaliação de utilidade de comentários. Para isso, coletaram dados em Português Brasileiro de comentários escritos no Steam, um site de vendas e avaliações de jogos amplamente conhecido no meio.

O objetivo principal do estudo foi achar a pontuação *h* da utilidade de comentários, em que *h* é um número real entre 0 e 1, como no trabalho visto anteriormente. Assim, o trabalho desenvolveu um modelo *Multi-Layer Perceptron* para predizer essa pontuação com base num conjunto de *atributos* que foram divididos com base nas características de autoria dos comentários, como a *expertise* e a reputação do autor, elementos estilísticos (comprimento, facilidade de leitura), semânticos (sentimentos positivos e negativos) e de meta-dados (se um comentário foi ou não recomendado pelos usuários, por exemplo).

O modelo é dado por uma função de regressão que tem como entrada um vetor *X* contendo esses atributos e como saída um *h* escalar que é a percepção da pontuação de utilidade de comentário.

Tabela 3 – Peso relativo das variáveis

Variables	Weights
average number of votes per user	0.14272
user average rating	0.06112
amount of timed played	0.02507
number of friends	0.01743
linguistic patterns	-0.00442
readability	0.00789
number of words	0.02794
number of sentences	0.01224
difference between user rating and average rating	0.01302
number of monosyllabic words	-0.00962
difference between product release date and posting date	0.00122

Fonte: Barbosa, J. L., Moura, R. S., & Santos, R. L. D. S. *Predicting Portuguese steam review helpfulness using artificial neural networks*. Proceedings of the 22nd Brazilian Synposium on Multimedia and the Web, 2016.

Os resultados do estudo (reproduzidos na Tabela 3) apontaram para a avaliação do autor do comentário (*user average rating*) e o número de palavras (*number of words*) do comentário como os atributos que mais influenciam para avaliação de utilidade deste, uma vez que os pesos acima de 0,02 são bons indicadores.

Baowaly et al. (2019) descrevem a criação um sistema para ajudar a Steam a priorizar comentários detectando automaticamente sua utilidade. Foi feito não só um modelo preditivo baseado em regressão para prever uma pontuação de utilidade, mas também um modelo preditivo baseado em classificação para identificar automaticamente comentários úteis ou inúteis. Este trabalho foi feito com comentários na língua inglesa dos 10 gêneros de jogos que possuíam a maior quantidade de comentários. O modelo de percepção de utilidade escolhido é o mesmo que o dos trabalhos citados anteriormente, em que *h* é a pontuação de utilidade do comentário. Foram extraídos *atributos* semânticos, lexicais, de *word embeddings* e de meta-dados para o treinamento e teste do modelo.

Para extração de *atributos* semânticos, foi utilizado o LIWC (*Linguistic Inquiry and Word Count*) e a Alocacão de *Dirichlet* Latente (LDA). O primeiro trata-se de um método de dicionário de palavras para avaliar emocional e psicologicamente as palavras num contexto textual. O segundo é um modelo gerativo probabilístico usado para descobrir tópicos semânticos numa coleção de documento. Os *atributos* lexicais foram extraídos e analisados com o método estatístico *tf-idf*, já mencionado anteriormente. Quanto aos *atributos* de *word embeddings*, foi utilizado o Word2Vec como em Sousa et al. (2019).

Os autores aplicaram o algoritmo de aprendizado baseado em árvores GBM (*Gradient Boosting Machine*) para treinar o modelo. Cada conjunto de dados foi particionado em 80% para treinamento e 20% para teste. Os resultados para o classificador alcançaram uma medida-F perto ou acima de 90% na predição de utilidade.

Tabela 4 – Resultado das avaliações por gênero de jogo

Model evaluation results (F-Scores) on test data

Genre	Are the reviews helpful?*		Are the reviews bad?*	
	$\Theta = 0.90$	$\Theta = 0.95$	$\Theta = 0.05$	$\Theta = 0.10$
Action	0.951	0.996	0.910	0.900
Survival	0.937	0.987	0.981	0.923
RPG	0.940	0.991	0.906	0.891
Strategy	0.952	0.990	0.959	0.920
Simulation	0.938	0.983	0.968	0.944
FPS	0.978	0.998	0.980	0.956
Adventure	0.975	0.996	0.955	0.917
Racing	0.988	0.986	0.971	0.956
Horror	0.980	0.980	0.986	0.949
Anime	0.985	0.999	0.987	0.978
Combined dataset	0.740	0.949	0.836	0.818

*Classifying reviews based on different rating thresholds $\Theta \in \{0.90, 0.95, 0.05, 0.10\}$.

Fonte: Baowaly, M. K.., et. al. *Predicting the helpfulness of game reviews: a case study on the Steam store*. Journal of Inteliggent & Fuzzy Systems 36, no. 5, 2019.

Já para a pontuação de utilidade, o modelo GBM foi modificado para adquirir caráter regressivo, calculando a raiz quadrada do erro médio, ou *root mean squared error* (RMSE) de cada gênero específico ou do conjunto de dados combinado. Modelos com gêneros específicos tiveram um desempenho de 13-16%, enquanto que o conjunto de dados apresentou um RMSE de 17%.

Tabela 5 – Resultado das avaliações por modelo de regressão

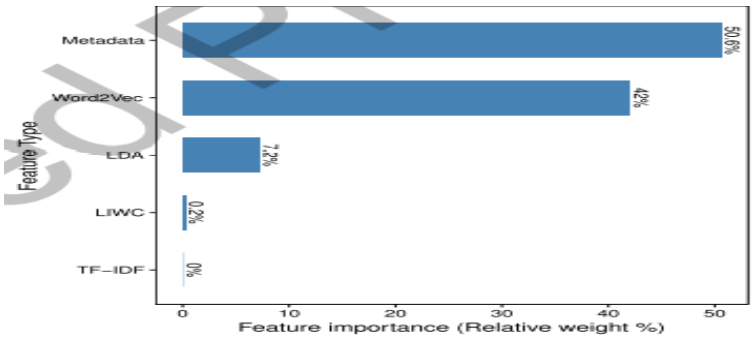
Evaluation results (RMSEs) of regression-based models

Genre	RMSE	Genre	RMSE
Action	0.152	Adventure	0.158
Survival	0.149	Racing	0.157
RPG	0.156	Horror	0.139
Strategy	0.155	Anime	0.139
Simulation	0.158	Combined	0.170
FPS	0.140	dataset	

Fonte: Baowaly, M. K.., et. al. *Predicting the helpfulness of game reviews: a case study on the Steam store*. Journal of Inteliggent & Fuzzy Systems 36, no. 5, 2019.

Além disso, o classificador GBM utilizado possui uma biblioteca que permite avaliar a importância dos *atributos* de acordo com o peso de suas variáveis. Dessa forma, foi constatado que os *atributos* mais importantes são os de meta-dados (50.6%), enquanto os atributos de Word2Vec (42%) e LDA (7.2%) ocupam o segundo e terceiro lugar no grau de importância.

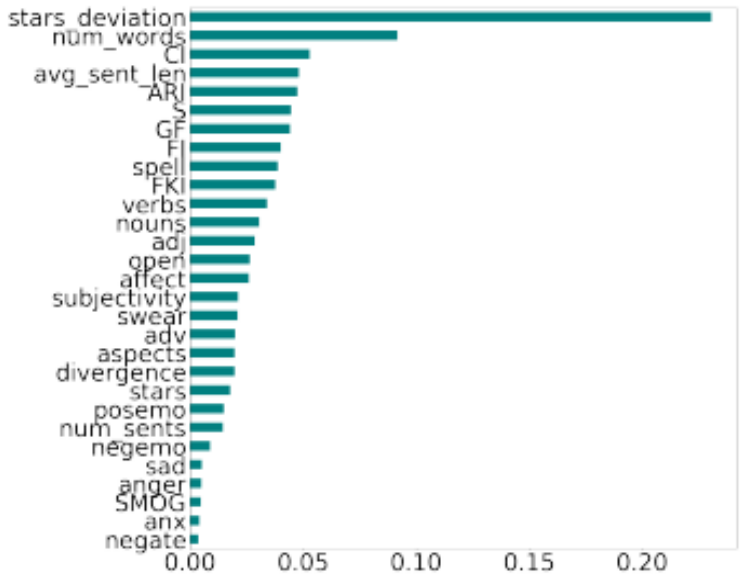
Figura 16 – Importância de *feature* por tipo



Fonte: Baowaly, M. K.., et. al. *Predicting the helpfulness of game reviews: a case study on the Steam store*. Journal of Inteliggent & Fuzzy Systems 36, no. 5, 2019.

Outro trabalho recente de extrema relevância para esta monografia é o de Sousa e Pardo (2022), que investigou diferentes métodos de aprendizado de máquina para classificação de utilidade de comentários e buscou descobrir os atributos mais importantes. Baseado em seu antigo banco de dados que contém comentários de Filmes e Aplicativos, o UTLCorpus, Sousa testou desde métodos mais antigos como o SVM até os mais recentes como um classificador baseado em BERT e outro em redes neurais convolucionais para verificar suas performances.

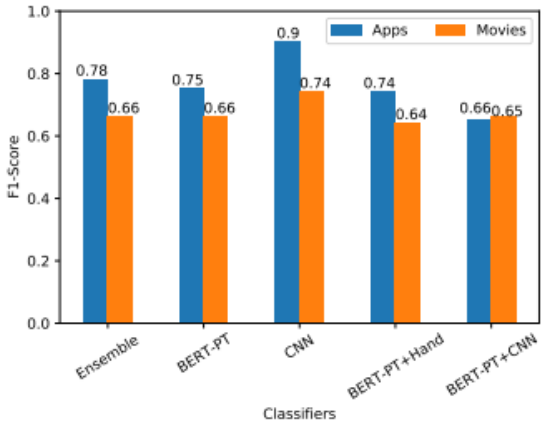
Figura 17 – Resultado da importância de atributos obtida por Sousa e Pardo (2022)



Fonte: Sousa & Pardo, *Evaluating Content Features and Classification Methods for Helpfulness Prediction of Online Reviews: Establishing a Benchmark for Portuguese*, (https://aclanthology.org/2022.wassa-1.19), 2022

O gráfico anterior apresenta *stars_deviation* como o atributo mais importe. Este é calculado através da diferença entre o número de estrelas atributo ao produto pelo autor do comentário e o número total de estrelas que o produto recebeu. Em seguida, destaca-se a grande importância de *num_words*, o número de palavras.

Figura 18 – Medidas F1 obtidas pelos classificadores mais recentes.



Fonte: Sousa & Pardo, *Evaluating Content Features and Classification Methods for Helpfulness Prediction of Online Reviews: Establishing a Benchmark for Portuguese*, (https://aclanthology.org/2022.wassa-1.19), 2022

Sousa e Pardo (2022) demonstram que o classificador que utilizou a rede neural foi o que conseguiu os melhores resultados, apresentando medidas F1 de até 0.9 para aplicativos, enquanto os classificadores mais antigos ficaram com uma média de 0.7

A seguir, com base na revisão relatada, são apresentados os objetivos do trabalho.

3 OBJETIVOS

Embora o consumidor possa buscar nos comentários orientações sobre um jogo para decidir-se na hora de realizar sua compra, os jogos na Steam.com possuem inúmeros comentários com qualidade inconsistente, o que torna a tarefa difícil. Assim, a maioria dos sites de *e-commerce*, como é o caso presente, encoraja os usuários a indicarem a utilidade dos comentários dispostos em uma lista através de um voto e disponibiliza os mais úteis no começo dela. Contudo, os dados coletados neste trabalho mostram que apenas 4% dos comentários possuem três ou mais votos. Logo, pode ser que comentários úteis sejam ignorados pelo usuário por não conterem votos o suficiente para aparecerem em uma posição privilegiada na lista, como é o caso daqueles que foram escritos recentemente ou são de um jogo pouco conhecido e com poucas visualizações.

Para resolver problemas como esse, já é considerável o número de trabalhos em automação da predição de utilidade de comentários. Porém, ainda são escassas as pesquisas e os corpora em Português Brasileiro para este assunto, destacando-se o *Buscapé* (Hartmann et al., 2014) e o *UTLcorpus* (Sousa et al., 2019).

Dessa forma, este trabalho possui os seguintes objetivos:

- criar um modelo capaz de avaliar automaticamente a utilidade de comentários e contribuir para a pesquisa na área;

- descobrir quais atributos (metadados, semânticos, distribucionais) podem indicar um comentário útil;

- ajudar na construção de mais um corpus em Português Brasileiro para predição de utilidade.

4 METODOLOGIA

4.1 COLETA DE DADOS

Os dados foram coletados por meio de um *web crawler* (Kausar et al., 2013)[5], que analisou 12.872 jogos diferentes e criou um arquivo para cada um deles, contendo o seu nome e os comentários, além de informações de metadados essenciais, como o número de votos e se o autor do comentário havia recomendado ou não aquele jogo.

Cada um desses arquivos foi nomeado automaticamente com um ID[6] pelo programa e, depois, foram renomeados/anotados manualmente com o nome do jogo e seu gênero. A decisão de categorizar os jogos pelos seus gêneros foi feita conforme Baowaly et al. (2019), que afirmam que este é um processo fundamental, tratando-se do treinamento de algoritmos de predição de utilidade.

A Tabela 6 mostra o número de comentários por gênero. No total, foram obtidos 2.789.893 comentários. Então, foram filtrados apenas aqueles com número de votos >=3, resultando em 233.824 divididos em 10 gêneros, como Ação, Indie, RPG, etc.

Tabela 6 – Número de comentários por gênero de jogo.

Gênero	Base de dados sem filtro	Base de dados filtrada (pelo menos 3 votos)
	n. de comentários	n. de comentários
Ação	734.894	65.164
Indie	504.648	39.442
RPG	469.548	38.658
Aventura	366.078	33.088
Estratégia	189.073	17.842
Simulação	157.536	11.164
Terror	148.510	12.880
FPS	102.368	7.714
Corrida	93.743	7.166
Esportes	23.495	1.966
Total	2.789.893	233.824

[5] Um programa que busca e recupera de forma automática as informações de um site (Kausar et al., 2013)

[6] Chave de identificação

4.2 ANOTAÇÃO DO CORPUS

O *web crawler* foi programado para percorrer e baixar os dados de comentários a partir de uma lista de jogos no site da Steam obtida com suas respectivas chaves de identificação (ID do jogo). Uma vez baixados os comentários e seus metadados, esses dados eram armazenados em arquivo que continha a ID em seu nome, como por exemplo *review_271590.json*. Embora tenham sido extraídos metadados dos comentários, as informações de gênero e nome do jogo estavam faltando. Dessa maneira, foi necessário digitar manualmente as chaves de identificação dos 12 mil jogos (que estavam contidas no arquivo final do *web crawler* para cada jogo*)* no mecanismo de busca da Steam e assim os jogos foram categorizados por gênero.

4.3 PRÉ-PROCESSAMENTO

Os comentários da base filtrada passaram por um pré-processamento em que tiveram suas letras maiúsculas convertidas para minúsculas, e caracteres especiais, números, e pontuações removidos. Em seguida, foram tokenizados e tiveram suas *stopwords* (como “a”, “o”, “é”) removidas.

A partir disso, a base foi randomizada e dividida ao meio. Uma metade foi utilizada no treinamento do Doc2Vec (Mikolov e Le, 2014) e a outra para o treino e teste do algoritmo. Embora haja repositórios contendo Doc2Vec e Word2Vec pré-treinados de outras bases, julgou-se necessário utilizar vetores de uma base de dados que continha apenas comentários de jogos, ou seja, dados do mesmo gênero com os quais se deseja testar o método.

4.4 ATRIBUTOS

Após a base ter sido dividida e pré-processada, iniciou-se o processo de extração de atributos, que foram categorizados em metadados, semânticos e distribucionais (correspondendo às *word embeddings*). No Quadro 1 é mostrado a quantidade de cada tipo de atributo por comentário, seguida de sua variável e uma breve descrição sobre o que ele representa, assim como o trabalho de referência para o atributo.

Quadro 1– Atributos dos comentários

Atributo	Variável	Explicação
Metadata(8)	Recomendou	Autor recomendou ou não este jogo a outros (Baowaly et al., 2019)
	n.sentences	Número total de sentenças no comentário (Liu et al., 2007, Lu et al., 2010)
	n.words	Número total de palavras no comentário (Kim et al., 2006, Mudambi and Schuff 2010, Baowaly et al., 2019)
	avg.sentence.length	Proporção entre o número de palavras e sentenças do comentário (Liu et al., 2007, Lu et al., 2010)
	n.exclamation	Número total de exclamações no comentário (Baowaly et al., 2019)
	n.question	Número total de interrogações no comentário (Baowaly et al., 2019)
	uppercase.ratio	Proporção entre as letras maiúsculas e minúsculas no comentário (Baowaly et al., 2019)
LIWC(62)	liwc.*(e.g. liwc.leisure, liwc.achieve, liwc.anx)	Contagem de palavras que revelam determinados sentimentos e opiniões(Kim et. al 2006) baseada no dicionário LIWC (Balage Filho et al., 2013, Pennebaker et al., 2001)
LDA(30)	topic.*(e.g. topic.1, topic.2, topic.3)	Tópicos inferidos a partir do corpus do comentário (Baowaly et al., 2019) a partir da técnica LDA (Blei et al., 2017)
Doc2Vec(1000)	wv.*(e.g. wv.1, wv.2, wv.3, wv.4)	Conjunto de vetores Doc2Vec (Mikolov et al., 2013a) que representam o comentário

4.4.1 Atributos de metadados

São os dados sobre os dados, fornecendo informações sobre o autor do comentário como se ele recomendou o jogo ou não e sobre o comentário em si, como o número de sentenças e número de palavras. Todos os atributos e suas devidas explicações podem ser encontrados na tabela anterior.

4.4.2 Atributos semânticos

Dizem respeito aos significados contidos no comentário, que podem remeter a opiniões, emoções e características psicológicas das palavras em um contexto. A partir de um dicionário chamado LIWC (Balage Filho et al., 2013; Pennebaker et al., 2001) o programa compara a palavra alvo do comentário com uma palavra do dicionário e atribui a ela uma categoria: por exemplo, uma emoção positiva, uma expressão de lazer, conquista, etc. (Baowaly et al., 2019).

4.4.3 Atributos distribucionais

Como já melhor explicado na fundamentação teórica deste trabalho, as *word embeddings* são representações vetoriais das palavras do texto. Neste caso, foi utilizado o Doc2Vec (Mikolov et al., 2013a), que recebeu todo o comentário como entrada e forneceu como saída um vetor de 1000 dimensões, treinado a partir de uma rede neural de 2 camadas para aprender contextos linguísticos das palavras.

4.5 EXTRAÇÃO DE PADRÕES

Foi utilizado o algoritmo de *Gradient Boosting Machine (GBM)* (Friedman, 2011), uma ferramenta de aprendizagem de máquina supervisionada já mencionada na fundamentação teórica. Embora o trabalho de Sousa e Pardo (2022) tenha mostrado que o algoritmo de classificação com melhores resultados seja baseado em redes neuras convolucionais, o motivo da escolha do GBM em particular foi devido à intenção desta monografia em replicar o trabalho de Baowaly et al., (2019) e comparar os resultados.

Os dados foram divididos em 80% para treino e 20% para teste. As classes (útil e não útil) foram balanceadas realizando *oversampling* para a classe minoritária e *undersampling* para a majoritária com base em outros estudos (Kotsantis et al., 2006; Lemnaru et al., 2011) utilizando a técnica SMOTENN (Batista et al., 2004), que preenche a classe minoritária com classes sintéticas, ou seja, criadas artificialmente, e também limpa o espaço resultante do oversampling.

Dividindo o número de votos úteis pelo total de total de votos conforme a equação 4, é obtida uma pontuação (ou limiar) de utilidade, doravante *threshold* Θ que vai de 0 a 1, sendo 1 o máximo de utilidade. Este trabalho baseou-se em Sousa et al., (2019) e estabeleceu um *threshold* de > 0,5 para definir um comentário como útil.

As métricas utilizadas foram as medidas F1 no caso da classificação e o a Raiz do Erro Médio Quadrático ou *Root Mean Squared Error (RMSE)* para o algoritmo de regressão, chamado de *Gradient Boosting Regressor (GRE)*. Quanto maior a medida F1, melhores os resultados; quanto menor o RMSE, melhor.

5 EXPERIMENTOS E ANÁLISES

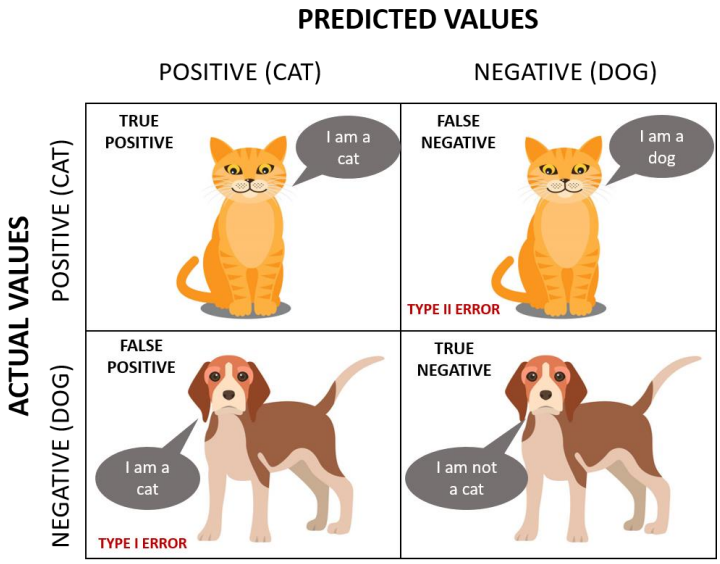
Foram conduzidos experimentos para construir um modelo preditivo de utilidade utilizando os algoritmos *Gradient Boosting Machine (GBM)* (Friedman, 2011) para a tarefa de classificação e *Gradient Boosting Regressor (GRE)* (Friedman, 2011) para a tarefa de regressão. Ambas são técnicas poderosas de aprendizagem de máquina supervisionada. Além disso, buscou-se descobrir a importância de cada atributo para tornar um comentário útil. Para isso, os próprios algoritmos continham uma ferramenta de avaliação de peso de atributos.

5.1 MÉTRICAS PARA AVALIAÇÃO DOS MODELOS

Atualmente é extenso o número de métricas possíveis para avaliar o desempenho dos modelos de aprendizado de máquina. O presente trabalho apresenta as mais relevantes levando em consideração os experimentos feitos e a literatura de estado da arte revisada.

Para o contexto de métodos de classificação, a matriz de confusão é uma medida efetiva das quais podem ser extraídas outras. Esta matriz mostra o número de classificações corretas e as classificações preditas para cada classe (Matos et al., 2009) e é composta por 4 valores: Verdadeiro Positivo (TP), Verdadeiro Negativo (TN), Falso Positivo (FP) e Falso Negativo (FN).

Figura 19 – Matriz de confusão para um algoritmo de classificação de cães e gatos.



Fonte: https://ml-concepts.com/2022/01/15/accuracy-specificity-precision-recall-and-f1-score-for-model-selection/. Acessado em 07/09/2022/.

A figura anterior mostra os possíveis resultados para um algoritmo de classificação de cães e gatos. No primeiro caso previu um gato como um gato (Verdadeiro Positivo (VP)), no segundo previu um gato como um cão (Falso Negativo (FN)), no terceiro um cão como um gato (Falso Positivo (FP)) e por fim previu corretamente um cão como não sendo um gato (Verdadeiro Negativo (VN)).

A partir da matriz de confusão, é possível extrair métricas como a precisão, revocação, acurácia e medida-f. De acordo com Matos et al., (2009):

A precisão é a quantidade de verdadeiros positivos levando em conta o total de positivos e sem os negativos.

(1)

$$Precisão = \frac{\mathrm{VP}}{\mathrm{VP} + \mathrm{FP}}$$

A revocação apresenta o total de informação relevante levando em conta os positivos e negativos.

(2)

$$Revocação = \frac{\mathrm{VP}}{\mathrm{VP} + \mathrm{FN}}$$

A acurácia é o total de verdadeiros positivos e negativos sobre o total de todas as previsões

(3)

$$Acurácia = \frac{\mathrm{VP} + \mathrm{VN}}{\mathrm{VP} + \mathrm{FP} + \mathrm{VN} + \mathrm{FP}}$$

A Medida-F, talvez a mais mencionada ao decorrer deste trabalho, apresentada como F1, é a média harmônica entre precisão e revocação. Um valor alto desta medida significa que os demais valores não apresentam grandes distorções e, portanto, que a acurácia obtida é relevante.

Já no caso de algoritmos de regressão as medidas mais utilizadas são o Erro Quadrático Médio (MSE) e a Raiz do Erro Quadrático Médio (RMSE). Ambas calculam a média da diferença entre o valor predito com o real e penalizam valores que sejam muito diferentes elevando-a ao quadro. A última se difere da primeira por aplicar a raiz

quadrática para padronizar a escala entre a unidade e o dado original. Assim, quanto maior o valor dessas duas métricas, pior o desempenho do modelo nas previsões.

5.2 ANÁLISE DE CLASSIFICAÇÃO

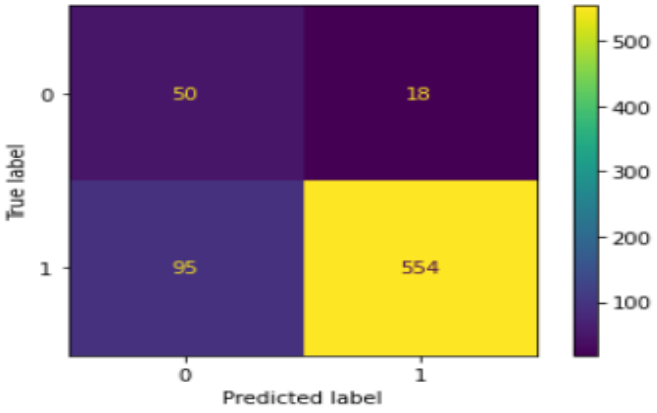
Para a classificação, utilizou-se um *threshold* de utilidade de comentário acima de 0,5. Após o comentário ter seu *score* calculado, é categorizado nas classes útil e não útil para o treino do algoritmo. Essa entrada é fornecida ao algoritmo junto aos demais atributos para o treinamento. Para avaliação de eficácia do algoritmo, foram verificadas as medidas F1. A Tabela 7 demonstra uma média de 90% das medidas F1, sendo os jogos separados por gêneros e o "Combinado", que representa a junção de todos os gêneros.

Tabela 7– Medidas F1 de comentários por gênero de jogo.

	O comentário é útil?	
Gênero	Medidas F1 Classe 0 (não)	Medidas F1 Classe 1(sim)
	Θ = 0.5	Θ = 0.5
Ação	0.35	0.88
Indie	0.40	0.94
RPG	0.43	0.91
Aventura	0.35	0.93
Estratégia	0.34	0.90
Simulação	0.35	0.93
Terror	0.32	0.92
FPS	0.26	0.90
Corrida	0.44	0.90
Esportes	0.16	0.85
Combinado	0.36	0.90

Nota-se que o gênero com pior F1 foi o de esportes. Este foi um dos gêneros que menos teve comentários para a amostra, o que pode ser a razão deste resultado. Já o gênero Indie foi o segundo com mais comentários, o que pode ter sido um dos motivos para uma medida alta. O gênero combinado, que é a junção de todos os outros, também obteve medidas boas, contudo medianas se comparado com o restante.

Tabela 8– Matriz de confusão para as classes 0 e 1 das medidas F1 do gênero Corrida.



É possível notar na Tabela 8 que a classe mais confundida foi a 0, ou seja, o algoritmo obteve problemas em classificar adequadamente os comentários "não-úteis". Isto se reflete através da proporção entre acertos (50) e erros (18). Interpretando a tabela, tem-se que a verdadeira classe 0 foi prevista como 0 num total de 50 vezes e como 1 num total de 18; já a classe 1 foi prevista corretamente 554 vezes como 1 e erroneamente como 0 em 95 instâncias.

Um possível motivo para isso é a escolha de comentários com *score* maior do que 0,5, deixando pouco treino para o algoritmo descobrir o que faz um comentário não-útil. Além disso, mesmo utilizando técnicas para tentar driblar o desbalanceamento, as classes sintéticas criadas pelo SMOTENN podem não ter sido efetivas, o que também gerou problemas.

Tabela 9– Medidas F1 de comentários por gênero de jogo utilizando apenas a classificação por *Bag of Words* (BoW) como linha de base.

Gênero	Medidas F1 para classe 0 (BoW)	Medidas F1 para classe 1 (BoW)
	Θ = 0.5	Θ = 0.5
Ação	0.18	0.40
Indie	0.14	0.50
RPG	0.19	0.45
Aventura	0.01	0.96
Estratégia	0.17	0.47
Simulação	0.02	0.97
Terror	0.05	0.96
FPS	0.00	0.95
Corrida	0.00	0.95
Esportes	0.19	0.20
Combinado	0.16	0.36

Para estabelecer-se um método baseline, como utilizado na literatura, foi treinado também um algoritmo utilizando como *atributos* de entrada apenas com *Bag of Words* criada com um número de atributos igual ao tamanho do vocabulário (composto por palavras distintas) e com frequência mínima de 1 palavra. Os resultados deste em relação ao anterior (Tabela 10 e Tabela 8) mostram que as medidas F1 são maiores quando foi utilizada uma maior variedade de atributos, chegando a ter um resultado até mesmo dobrado no caso do gênero Ação, por exemplo.

5.3 ANÁLISE DE REGRESSÃO

A regressão busca calcular o *score* de utilidade, isto é, determinar o quanto um comentário é útil. Para isso, recebe o comentário e seus atributos, juntamente da pontuação de utilidade já calculada anteriormente como alvo para prever novos *thresholds* nos comentários de teste. Como medida, é utilizado o RMSE, que, quanto menor, melhor. A Tabela 9 mostra uma média de erros de 10%. Ao contrário do caso de classificação, a regressão não mostrou uma grande diferença no RMSE entre o algoritmo de múltiplos atributos e o *baseline* que foi treinado apenas com *Bag of Words*.

Tabela 10– Medidas RMSE de comentários por gênero de jogo.

Gênero	RMSE
Ação	0.092
Indie	0.091
RPG	0.096
Aventura	0.106
Estratégia	0.093
Simulação	0.099
Terror	0.084
FPS	0.099
Corrida	0.091
Esportes	0.123
Combinado	0.095

Os gêneros com menores erros foram Indie e Ação, coincidentemente os dois com maiores amostras de comentários, estando o Indie novamente entre os melhores previstos como no caso da classificação. O pior também foi o de Esportes, como na

classificação. Novamente, um possível motivo seria a quantidade inferior de comentários disponíveis como entrada para o aprendizado de máquina.

Tabela 11– Medidas RMSE de comentários por gênero de jogo utilizando apenas a *Bag of Words* (BoW) como linha de base.

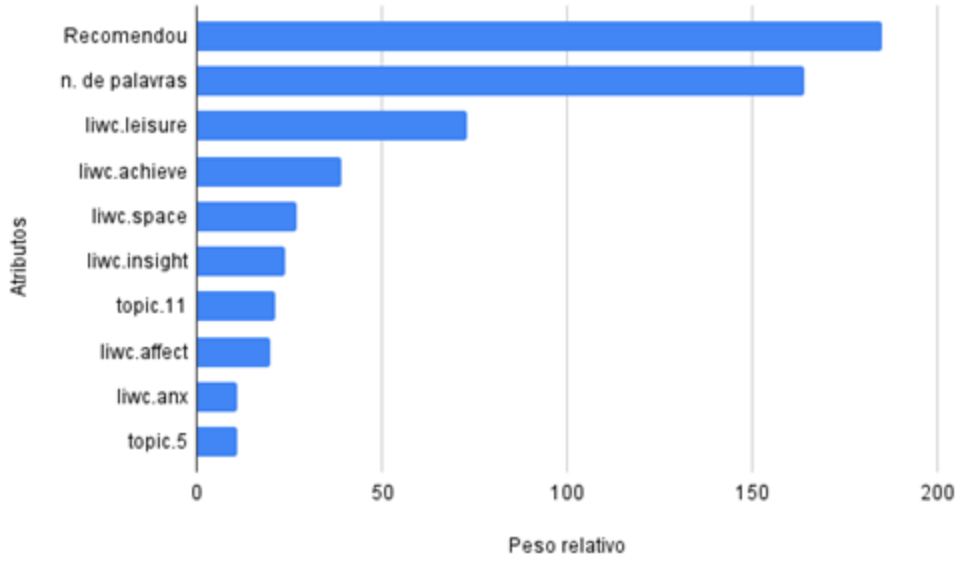
Gênero	RMSE (BoW)
Ação	0.094
Indie	0.095
RPG	0.099
Aventura	0.103
Estratégia	0.096
Simulação	0.099
Terror	0.084
FPS	0.099
Corrida	0.093
Esportes	0.109
Combinado	0.096

A diferença entre o modelo de *Bag of Words* e o original não foi tão discrepante como na previsão por classificação, mantendo o gênero de esportes como o pior e os de ação e aventura como melhores. É importante lembrar que o "Combinado" é a junção de todos os gêneros.

5.4 ANÁLISE DE IMPORTÂNCIA DE ATRIBUTOS

A variável de importância é baseada no número de vezes que uma variável é selecionada para dividir a árvore de decisão do algoritmo e então mensurada pelo aprimoramento quadrático do modelo como resultado de cada divisão (Friedman et al., 2003). Os números maiores indicam uma influência maior na predição. Utilizando este método já contido nos algoritmos, foram calculados os 10 atributos mais importantes para cada gênero. A Figura 20 demonstra os resultados no gênero de Aventura.

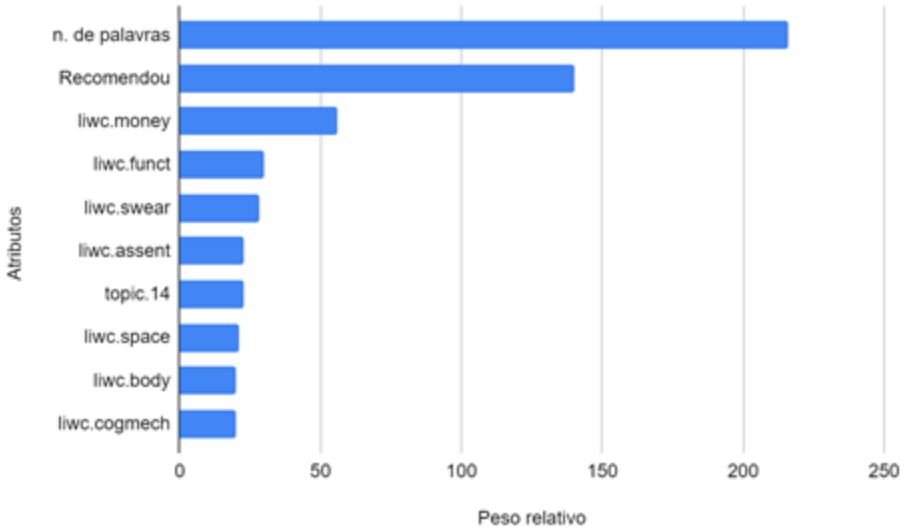
Figura 20 - Gráfico de importância de cada atributo para determinar a utilidade de comentário no gênero Aventura.



Nota-se que a utilidade dos comentários foi determinada principalmente por atributos de metadados, como a recomendação do jogo pelo autor do comentário e também pelo número de palavras deste. Em seguida, os atributos de maior importância foram os semânticos, calculados com base no LIWC. Parece haver forte correlação entre a classe e as categorias de "lazer", "conquista" e "afeição". Por fim, os demais dizem respeito a tópicos calculados com o LDA.

A Figura 21 demonstra resultados semelhantes com o gênero Estratégia. Os dois atributos mais importantes são os mesmos, contudo, o número de palavras está em primeiro lugar. Os demais também são em sua maioria semânticos e fazem parte do LIWC, enquanto que também há a presença de um tópico do LDA.

Figura 21 - Gráfico de importância de cada atributo para determinar a utilidade de comentário no gênero Aventura.



Este foi um padrão que se manteve no restante dos gêneros, sendo os dois atributos mais importantes o número de palavras e a recomendação do autor, seguidos dos atributos semânticos.

Conforme visto no capítulo de Estado da Arte, Sousa e Pardo (2022) também obtiveram o número de palavras como um dos atributos mais importante, geralmente situado em primeiro ou segundo lugar. Um outro atributo no trabalho destes autores foi a diferença de estrelas dadas ao produto pelo autor do comentário com o total de estrelas do produto. Infelizmente, a Steam não possui um sistema de estrelas, mas o sistema de recomendação parece semelhante. Um possível trabalho futuro é calcular a proporção de recomendações totais do jogo pelos usuários com a recomendação ou não-recomendação do autor do comentário.

5.5 ANÁLISE DE ERROS

Os dados a seguir são comentários que fazem parte da amostra balanceada e com sua utilidade baseada num *score* maior que 0,5 para útil, nos jogos do gênero RPG. Foram extraídos, de maneira randômica, 20 comentários que tiveram suas classes classificadas equivocadamente pelo algoritmo. 10 comentários úteis classificados como não-úteis e 10 não-úteis classificados como úteis. Então, foram escolhidos 5 para as análises.

Comentário 1 – útil previsto como não-útil

Depois desses novos nerfs que sairam nao recomendo o jogo, os crafts do jogo são absurdamente caros basicamente vc é obrigado a fazer uma build de gold

farm pq sem ela vc nao faz nada, dps de gastar umas 12h pra montar o boneco de gold farm vc tem que ter a sorte pra consequir os mods corretos e ate entao ate concordo que alguns mods podem estar desbalanceados mas eles nerfaram tanto que tudo que vc fez pro seu boneco ficar forte foi pra vala nao recomendo jogar ate que eles deixem o jogo bem equilibrado pq eu perdi toda a vontade de jogar dps desses nerfs absurdos.

Embora tenha sido bem estruturado, longo e com informações relevantes, o Comentário 1 foi classificado como não-útil. Um possível motivo para isso está em seus metadados e na posição do autor. O autor do comentário não recomendou o jogo, e a recomendação está entre os dois atributos mais importantes conforme a Figura 21. Além disso, o comentário aponta aspectos negativos. Estes dois fatores podem ter influenciado na classificação do Algoritmo.

Comentário 2 – útil previsto como não-útil

QUERO JOGAR ESSA BAGAÇA MAS SIMPLESMENTE O JOGO/LAUCHER APRESENTA CENTENAS DE ERROS SEGUIDOS. VAI TOMAR NO CU

O comentário 2 apresenta alguns aspectos comuns de "não-utilidade", tais como a proporção de letras maiúsculas e o tamanho pequeno. Além disso, trata-se de uma crítica negativa e o autor não recomendou o jogo. Assim, seria lógica a classificação de "não-útil" pelo algoritmo. Contudo, a comunidade o julgou como um comentário útil. Isto mostra um problema a ser resolvido: às vezes a comunidade aprova um comentário somente por concordar com a opinião do autor, ou seja, parece que o restante dos jogadores também encontrou problemas com o jogo e assim julgou o comentário como útil, embora este não contenha atributos de utilidade. Assim, estes casos tornam-se um desafio para o algoritmo.

Comentário 3 – útil previsto como não-útil

Uma bosta

O mesmo acontece com o Comentário 3, que, embora não apresente nenhuma informação relevante além de uma opinião, e seja de tamanho pequeno, foi classificado como não-útil quando na verdade o restante dos usuários votou nele o suficiente para ter um *score* de utilidade acima de 0,5.

Comentário 4 – não-útil previsto como útil

-Jogo com bugs de 10 meses que impedem novos jogadores de terem acesso a sistemas extremamente importantes par aumetnar status.

-Drop rate de itens absurdamente baixa , tão baixa que vc pode passar quase uma semana farmando 5h/dia com todos os buffs de drop de cash

e evento sem dropar um unico item 5stars com bonus de skill, isso tornou o comercio do jogo totalmente morto sendo que vc não dropará os itens que vc precisa e não podera comprar de ngm pq eles tbm n droparão

-Sistema de grind que todo player antigo adora dizer que é perfeito por ser dificil mas em quase dois anos de servidor ngm chegou perto de pegar lv maximo.

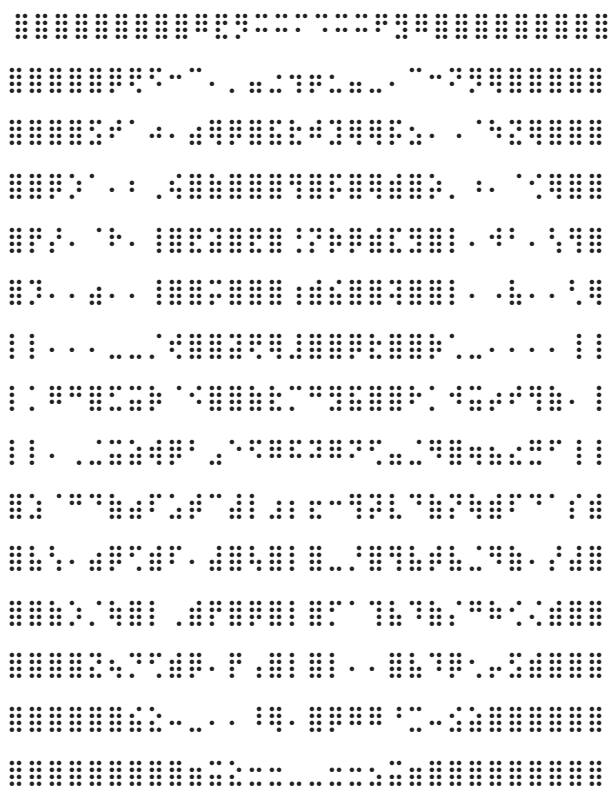
Não importa o tipo do evento ou atualização, o servidor se mantem desde sempre com menos de 30 players ativos diariamente (não contado os alts de quem loga varias contas).

-Nenhuma classe do jogo possui ataques em area então vc tem que matar mobs de um em um usando varias skills em cada podendo passar facilmente de 10 skills em cada mob e vc terá que matar MILHÕES de mobs literalmente até chegar em mid lv.

-Hj, com a loja de cash é possivel comprar de tudo, desde de buffs de dano/def até tokens usados para trocar por box com os melhores equipes do jogo que um player free ou que vai gastar pouco jamais terá.

Já o Comentário 4 apresenta várias características de utilidade que justificam a escolha do algoritmo, como o seu tamanho, estrutura e diversas informações. Contudo, os usuários não o acharam útil o suficiente, novamente, talvez pelo simples ato de discordar do autor.

Comentário 5 – não-útil previsto como útil



Por fim, o Comentário 5 não se trata de um texto propriamente dito, mas de uma imagem formada com texto. Uma hipótese para a classificação deste comentário como

útil pelo comentário é devido ao número de caracteres e à recomendação do jogo pelo autor, dois dos principais atributos, como visto na Figura 21.

6 CONCLUSÕES E TRABALHOS FUTUROS

Esta monografia discutiu a importância da predição de utilidade de comentários para sites de *e-commerce* e a falta de corpora para o assunto em Português Brasileiro. Dessa forma, propôs a criação de um novo corpus de jogos da Steam.com e um modelo que pudesse classificar um comentário como útil e atribuir a ele uma pontuação de utilidade, como também verificar quais atributos contribuem para esta tarefa.

Além disso, foi apresentado um estudo abrangente dos métodos de mineração de dados, textos, opiniões e Processamento de Linguagem Natural e o estado da arte em predição de utilidade, assim como uma breve revisão bibliográfica dos assuntos.

Foi criada uma base de dados contendo 2.789.893 comentários de jogos da Steam em Português Brasileiro e um modelo preditivo que, embora não atingiu os mesmos resultados que o estado da arte, conseguiu obter medidas F1 de 90% na classificação de comentários úteis e medidas RMSE de 10% para pontuação de utilidade.

Também foi demonstrado como os atributos que mais contribuem para tornar um comentário útil são os de metadados, como se o usuário recomendou aquele jogo ou não para outros e o número de palavras de um comentário, seguido dos atributos semânticos obtidos com o LIWC e os de tópico com o LDA.

Este trabalho seguiu as recomendações de Baowaly et al. (2019), que criticou a escassez de dados na tentativa de predição de utilidade em comentários da Steam por Barbosa et al. (2016), sugerindo uma base de dados relativamente grande e também a divisão dos dados por gênero de jogo. Dessa forma, criou-se uma extensa base e realizou-se tal divisão.

No caso dos algoritmos de classificação, foi possível notar um aprimoramento nas medidas F1 quando os gêneros foram divididos. Contudo, na regressão, isto parece não ter tido efeito.

Uma pesquisa futura poderia utilizar este banco de dados criado para observar a importância de outros tipos de atributos como TF-IDF, lexicais e sintáticos, além de utilizar outros algoritmos como o lightGBM e o XGBoost ou algoritmos de redes neurais e comparar seus desempenhos na predição de utilidade de comentários. Outro fator a ser considerado é a representação textual por meio de vetores. Este trabalho utilizou o Doc2Vec, contudo, seria interessante verificar a eficácia de modelos mais novos como o BERT (Devlin et al., 2018) para a vetorização dos comentários.

7 BIBLIOGRAFIA

ABUKAUSAR, MD.; S. DHAKA, V.; KUMAR SINGH, S. Web Crawler: A Review. **International Journal of Computer Applications**, v. 63, n. 2, p. 31–36, 15 fev. 2013.

BAOWALY, M. K.; TU, Y.-P.; CHEN, K.-T. Predicting the helpfulness of game reviews: A case study on the Steam store. **Journal of Intelligent & Fuzzy Systems**, v. 36, n. 5, p. 4731–4742, 14 maio 2019.

BARBOSA, J. L. N.; MOURA, R. S.; SANTOS, R. L. DE S. **Predicting Portuguese Steam Review Helpfulness Using Artificial Neural Networks**. Disponível em: <https://sol.sbc.org.br/index.php/webmedia/article/view/5376>. Acesso em: 3 set. 2022

BATISTA, G.; BAZZAN, A.; MONARD, M. **Balancing Training Data for Automated Annotation of Keywords: a Case Study**. [s.l: s.n.]. Disponível em: <http://www.inf.ufrgs.br/maslab/pergamus/pubs/balancing-training-data-for.pdf>. Acesso em: 7 set. 2022.

BECKER, K.; TUMITAN, D. **Minicurso 2 Introdução à Mineração de Opiniões: Conceitos, Aplicações e Desafios**. [s.l: s.n.]. Disponível em: <https://sbbd2013.cin.ufpe.br/Proceedings/artigos/pdfs/sbbd_min_02.pdf>. Acesso em: 21 jul. 2022.

BERTAGLIA, T. F. C. **Normalização textual de conteúdo gerado por usuário**. Disponível em: <https://www.teses.usp.br/teses/disponiveis/55/55134/tde-10112017-170919/en.php?ref=https://githubhelp.com>. Acesso em: 3 set. 2022.

BHATIA, P. **Data mining and data warehousing : principles and practical techniques**. Cambridge, United Kingdom ; New York, Ny: Cambridge University Press, 2020.

CHOLLET, F. **Deep Learning with Python**. Shelter Island (New York, Estados Unidos): Manning, Cop, 2017.

DEVLIN, J. et al. **BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding**. Disponível em: <https://arxiv.org/abs/1810.04805>.

PETERS, M. et al. **Deep contextualized word representations**. [s.l: s.n.]. Disponível em: <https://arxiv.org/pdf/1802.05365.pdf)%5BELMo%5D%E2%80%8B..>. Acesso em: 7 set. 2022.

FIGUEREDO DE SOUSA, R.; ALEXANDRE, T.; PARDO, S. **Evaluating Content Features and Classification Methods for Helpfulness Prediction of Online Reviews: Establishing a Benchmark for Portuguese**. [s.l: s.n.]. Disponível em: <https://aclanthology.org/2022.wassa-1.19.pdf>. Acesso em: 7 set. 2022.

FILHO, P.; PARDO, T.; ALUÍSIO, S. **An Evaluation of the Brazilian Portuguese LIWC Dictionary for Sentiment Analysis**. [s.l: s.n.]. Disponível em: <https://aclanthology.org/W13-4829.pdf>. Acesso em: 8 maio. 2022.

FRESSATO, E. P. **Incorporação de metadados semânticos para recomendação no cenário de partida fria**. Disponível em: <https://www.teses.usp.br/teses/disponiveis/55/55134/tde-09082019-134753/en.php>. Acesso em: 7 set. 2022.

FRIEDMAN, J. H. Greedy function approximation: A gradient boosting machine. **The Annals of Statistics**, v. 29, n. 5, p. 1189–1232, out. 2001.

FRIEDMAN, J. H.; MEULMAN, J. J. Multiple additive regression trees with application in epidemiology. **Statistics in Medicine**, v. 22, n. 9, p. 1365–1381, 2003.

HARTMANN, N. et al. **A Large Corpus of Product Reviews in Portuguese: Tackling Out-Of-Vocabulary Words**. Disponível em: <https://aclanthology.org/L14-1354/>. Acesso em: 3 set. 2022.

HIRSCHBERG, J.; MANNING, C. D. Advances in natural language processing. **Science**, v. 349, n. 6245, p. 261–266, 16 jul. 2015.

KAUFMAN, L.; ROUSSEEUW, P. J. Finding Groups in Data: An Introduction to Cluster Analysis. 2005.

KHAN, S. Ethem Alpaydin. Introduction to Machine Learning (Adaptive Computation and Machine Learning Series). The MIT Press, 2004. ISBN: 0 262 01211 1 Price £32.95 / $50.00 (hardcover). xxx+415 pages. **Natural Language Engineering**, v. 14, n. 01, 12 dez. 2007.

KIM, S.-M. Et al. **Automatically Assessing Review Helpfulness**. [s.l.] Association for Computational Linguistics, 2006. Disponível em: <https://aclanthology.org/W06-1650.pdf>.

KOTSIANTIS, S.; DIMISTRIS, D.; PINTELAS, P. Handling imbalanced datasets: A review. **GESTS international transactions on computer science and engineering 30,** v. 1, 2006.

KRISHNAMOORTHY, S. Linguistic features for review helpfulness prediction. **Expert Systems with Applications**, v. 42, n. 7, p. 3751–3759, maio 2015.

LE, Q.; MIKOLOV, T. **Distributed Representations of Sentences and Documents**. Disponível em: <http://proceedings.mlr.press/v32/le14.html?ref=https://githubhelp.com>.

LIU, B. Sentiment Analysis and Opinion Mining. **Synthesis Lectures on Human Language Technologies**, v. 5, n. 1, p. 1–167, 23 maio 2012.

LU, Y. et al. **Exploiting Social Context for Review Quality Prediction**. [s.l: s.n.]. Disponível em: <http://www.ra.ethz.ch/CDstore/www2010/www/p691.pdf>. Acesso em: 3 set. 2022.

MATOS, Pablo Freire, L. Lombardi, R. Ciferri, T. Pardo, C. Ciferri, and M. Vieira. **Relatório técnico "métricas de avaliação"**." Universidade Federal de Sao Carlos (2009).

MCCORMIRCK, C. **The Inner Workings of word2vec**. [s.l: s.n.].

MIKOLOV, T. et al. **Distributed Representations of Words and Phrases and their Compositionality**. Disponível em: <https://proceedings.neurips.cc/paper/2013/hash/9aa42b31882ec039965f3c4923ce901b-Abstract.html>.

MINSKY, M. Steps toward Artificial Intelligence. **Proceedings of the IRE**, v. 49, n. 1, p. 8–30, jan. 1961.

MORAIS, E.; AMBRÓSIO, A. **Mineração de Textos**. [s.l: s.n.]. Disponível em: <http://ww2.inf.ufg.br/sites/default/files/uploads/relatorios-tecnicos/RT-INF_005-07.pdf>.

MUDAMBI, S.; SCHUFF, D. Research Note: What Makes a Helpful Online Review? A Study of Customer Reviews on Amazon.com. **MIS Quarterly**, v. 34, n. 1, p. 185, 2010.

PENNEBAKER, J. et al. **Linguistic Inquiry and Word Count: LIWC2015 Operator's Manual**. [s.l: s.n.]. Disponível em: <http://downloads.liwc.net.s3.amazonaws.com/LIWC2015_OperatorManual.pdf>.

REZENDE, S. O. et al. **Mineração de dados, chapter 12.** [s.l: s.n.].

ROSENBLATT, F. The perceptron: A probabilistic model for information storage and organization in the brain. **Psychological Review**, v. 65, n. 6, p. 386–408, 1958.

SIMON, H. **Neural networks : a comprehensive foundation**. New Delhi: Prentice-Hall Of India, 2008.

SOUSA, R. F. DE; BRUM, H. B.; NUNES, M. DAS G. V. A bunch of helpfulness and sentiment corpora in brazilian portuguese. **Proceedings**, 2019.

VASWANI, A. et al. **Attention Is All You Need**. [s.l: s.n.]. Disponível em: <https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf>.

Z, Z. Utility scoring of product reviews. **Proc. 15th ACM International Conference on Information and Knowledge Management, 2006**, p. 51–57, 2006.

ZHANG, R. et al. **Enterprise Information Systems: 13th International Conference, ICEIS 2011, Beijing, China, June 8-11, 2011, Revised Selected Papers**. [s.l.] Springer,

2012.

ZONG, C.; XIA, R.; ZHANG, J. **Text data mining**. Singapore: Springer, 2021.