UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ENGENHARIA DE SÃO CARLOS

LUCAS FERREIRA LOPES

# Classificação de projetos sob ótica de risco e de retorno financeiro: uma abordagem baseada em análise de sentimento e árvore de decisão

São Carlos
2023

LUCAS FERREIRA LOPES

# Classificação de projetos sob ótica de risco e de retorno financeiro: uma abordagem baseada em análise de sentimento e árvore de decisão

Monografia apresentada ao Curso de Engenharia de Produção, da Escola de Engenharia de São Carlos da Universidade de São Paulo, como parte dos requisitos para obtenção do título de Engenheiro de Produção.

Orientador: Prof. Lucas Gabriel Zanon

São Carlos

2023

Ficha catalográfica elaborada pela Biblioteca Prof. Dr. Sérgio Rodrigues Fontes da EESC/USP com os dados inseridos pelo(a) autor(a).

Lopes, Lucas
L864c Classificação de projetos sob ótica de risco e de retorno financeiro: uma abordagem baseada em análise de sentimento e árvore de decisão / Lucas Lopes; orientador Lucas Zanon. São Carlos, 2023.

Monografia (Graduação em Engenharia de Produção) -- Escola de Engenharia de São Carlos da Universidade de São Paulo, 2023.

1. Classificação de projetos. 2. Seleção de portfólio de projetos. 3. Análise de sentimento. 4. Árvore de decisão. I. Título.

Eduardo Graziosi Silva - CRB - 8/8907

0

# FOLHA DE APROVAÇÃO

| |
|---|
| **Candidato:** Lucas Ferreira Lopes |
| **Título do TCC:** Classificação de projetos sob ótica de risco e de retorno financeiro: uma abordagem baseada em análise de sentimentos e árvore de decisão |
| **Data de defesa:** 13/12/2023 |

| **Comissão Julgadora** | **Resultado** |
|---|---|
| Professor Doutor Lucas Gabriel Zanon (orientador) | Aprovado |
| Instituição: EESC - SEP | |
| Professor Titular Luiz Cesar Ribeiro Carpinetti | APROVADO |
| Instituição: EESC - SEP | |
| Doutorando Rafael Ferro Munhoz Arantes | APROVADO |
| Instituição: EESC - SEP | |

Presidente da Banca: **Professor Doutor Lucas Gabriel Zanon**

# RESUMO

LOPES, L. F. **Classificação de projetos sob ótica de risco e de retorno financeiro: uma abordagem baseada em análise de sentimentos e árvore de decisão**. 2023. Monografia (Trabalho de Conclusão de Curso) – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2023.

Em um mercado competitivo, a seleção de um portifólio de projetos adequado se torna um problema mais relevante para as organizações, tendo em vista a alocação otimizada de recursos. Dessa forma, as decisões sobre a classificação de projetos quanto à sua importância de execução devem ser assertivas, fazendo com que sejam selecionados os projetos financeiramente viáveis e alinhados aos objetivos organizacionais. O aprendizado de máquina, como um ramo da inteligência artificial, pode ser utilizado para a construção de algoritmos capazes de gerar classificações para um conjunto de dados a partir de algumas técnicas, como a árvore de decisão, que representa um conjunto de regras a serem consideradas para a geração de classificações. Além disso, a análise de sentimento se destaca como uma técnica que busca atribuir um rótulo emocional para um texto em forma de polaridade, podendo ser utilizada para avaliação de riscos. Diante disso, o objetivo deste trabalho é desenvolver um modelo capaz de combinar as técnicas de análise de sentimento e árvore de decisão para aplicação na seleção de portfólio de projetos. Esse modelo é proposto para projetos com dados financeiros e riscos definidos e é baseado na atribuição de polaridade aos riscos tendo em vista os objetivos estratégicos da organização, de forma que seja possível induzir uma árvore de decisão para classificar os projetos de acordo com o ganho esperado, a despesa esperada e a polaridade associada aos riscos. A partir disso, foi realizada uma aplicação piloto do modelo em questão com intuito de avaliar a aderência e a acurácia dos resultados do modelo em um contexto definido. Portanto, como resultado dessa aplicação foi possível obter uma árvore de decisão que possibilita a classificação direta e ágil de projetos com a minimização de avaliações subjetivas dos mesmos, permitindo a alocação otimizada de recursos através da indicação da importância de execução dos projetos e consequente composição do portfólio de projetos.

**Palavras-chave:** Classificação de projetos. Seleção de portfólio de projetos. Análise de sentimento. Árvore de decisão.

# ABSTRACT

LOPES, L. F. **Project classification from the perspective of risk and financial return: an approach based on sentiment analysis and decision tree**. 2023. Monografia (Trabalho de Conclusão de Curso) – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2023.

In a competitive market, the selection of an adequate project portfolio becomes a more relevant problem for organizations, considering the optimized allocation of resources. Thus, decisions regarding the project classification in terms of their importance of execution must be assertive, ensuring the selection of financially viable projects aligned with organizational objectives. Machine learning, as a branch of artificial intelligence, can be used to build algorithms capable of generating classifications for a dataset using techniques such as the decision tree, which represents a set of rules to be considered for generating classifications. Furthermore, sentiment analysis stands out as a technique that seeks to assign an emotional label to a text in the form of polarity, which can be used for risk assessment. Therefore, the aim of this work is to develop a model capable of combining sentiment analysis and decision tree techniques for application in project portfolio selection. This model is proposed for projects with defined financial data and risks and is based on assigning polarity to risks, considering the organization's strategic objectives, allowing the induction of a decision tree to classify projects according to expected gain, expected expense, and the polarity associated with the risks. From this, a pilot application of the model was carried out to evaluate the adherence and accuracy of the model's results in a defined context. Therefore, as a result of this application, it was possible to obtain a decision tree that enables the direct and agile classification of projects with minimization of subjective evaluations, allowing the optimized allocation of resources by indicating the importance of project execution and the consequent composition of the project portfolio.

**Keywords:** Project classification. Project portfolio selection. Sentiment analysis. Decision tree.

# LISTA DE ILUSTRAÇÕES

# LISTA DE TABELAS

# SUMÁRIO

# 1. INTRODUÇÃO

Busca-se, nesta seção, apresentar a contextualização e os objetivos que direcionaram o presente trabalho.

## 1.1. CONTEXTUALIZAÇÃO

Em um cenário de grandes e repentinas mudanças no mercado, as organizações vêm se transformando e têm se preocupado cada vez mais com a sua competitividade (MARINO, 2006). No entanto, essa preocupação com a competitividade, faz com que, naturalmente, as empresas busquem ser mais eficientes, já que competitividade e eficiência estão indissociavelmente conectadas (GOLOVCHENKO et al., 2022).

Vale ressaltar que a eficiência de uma organização pode estar vinculada a três aspectos: 1) operacional, voltado para custos e lucros operacionais; 2) investimento, voltado para os ativos de uma organização; 3) financeiro, voltado para custos de capital. Assim, o controle desses aspectos da eficiência permite que sejam tomadas decisões assertivas (GOLOVCHENKO et al., 2022).

Com isso, em meio ao processo de globalização e o consequente aumento da concorrência global, surge a necessidade de desenvolver estratégias econômicas destinadas a melhorar a eficiência e a competitividade das empresas (GOLOVCHENKO et al., 2022).

Bernardino, Peixoto e Ferreira (2015) ainda afirmam que a sobrevivência, ou não, das organizações em mercados competitivos pode ser resultado de três pilares: da capacidade de gerenciar eficientemente seus recursos, da atratividade que o investimento proporciona e da segurança transmitida ao investidor. Vale ressaltar que, para além da sobrevivência no mercado suportada pela eficiência, Saurin, Lopes e da Costa Júnior (2010) sustentam que existe uma correlação positiva entre a eficiência de uma empresa e a medida financeira vinculada ao valor criado no curto prazo.

Dessa forma, as empresas tendem a adotar duas formas para buscar eficiência: 1) a partir da redução de custos e, consequentemente, reduzir os preços de venda; e 2) a partir de investimentos em qualidade e tecnologias, por exemplo (MARINO, 2006).

Diante do fato de que é imprescindível que empresas se tornem mais eficientes para que sejam cada vez mais competitivas em um mercado concorrido, é fundamental que as decisões sejam tomadas de forma assertiva para que os recursos sejam alocados corretamente e produzam o retorno esperado. Elonen e Artto (2003) argumentam que um ambiente de negócios complexo exige decisões rápidas, alocação eficiente de recursos e foco claro.

Tendo isso em vista, vale destacar o problema da seleção de um portfólio de projetos (ou *project portfolio selection problem*), que se destaca em função da importância de uma alocação otimizada de recursos, principalmente em cenários em que são apresentados recursos limitados. Segundo Mira et al. (2013), comumente os tomadores de decisão são confrontados com esse problema visando a seleção dos projetos que serão executados, que pode ser resumido pela seguinte questão: dado um conjunto de projetos, qual é a seleção que maximiza uma determinada função objetivo sobre todas as outras possíveis seleções?

Nesse sentido, projetos são caracterizados como esforços com resultados claros frequentemente utilizados por organizações como meio para atingir os objetivos estratégicos (PMI, 2008, p. 10-11). Pode-se afirmar, portanto, que um portfólio de projetos se trata de um grupo de projetos que são gerenciados por uma organização e que, com isso, acabam competindo por recursos escassos (como pessoas, capital e tempo) que comumente não são suficientes para executar todos os projetos propostos (ARCHER; GHASEMZADEH, 1999).

Para além da perspectiva estratégica, Moraes e Laurindo (2010) afirmam que a decisão sobre seleção de projetos deve considerar, simultaneamente, aspectos relacionados à eficiência (uso dos recursos), à eficácia (obtenção de resultados para a organização) e aos riscos envolvidos.

Vale destacar, ainda, que o retorno sobre os investimentos vinculados aos projetos pode ter caráter financeiro (na forma de geração de receita ou de redução de custos) ou não financeiro (na forma de desempenho de processos) (SILVA NETO, 2008 apud STRASSMANN, 1990).

Com isso, uma questão fundamental para empresas nos dias atuais é: como classificar corretamente os projetos com o intuito de direcionar a construção do portfólio de projetos alocando os recursos de maneira otimizada?

Para isso, pode ser utilizado o aprendizado de máquina, ou *machine learning* (ML), já que é utilizado em situações em que são buscados padrões com o intuito de prever um resultado. O aprendizado de máquina é um ramo da inteligência artificial (IA) que permite a construção

de algoritmos computacionais a partir do aprendizado oriundo de um conjunto de dados. Sendo assim, o principal objetivo de um modelo de *machine learning* é construir um sistema capaz de aprender com um determinado banco de dados e gerar predições, classificações, detecções, entre outros (PAIXÃO et al, 2022). Dentre as principais técnicas de ML pode-se citar a árvore de decisão, que pode ser utilizada para a realização de classificações (SCHNEIDER, 2016).

Diante disso, propõe-se, no presente trabalho, a utilização da árvore de decisão para a classificação de projetos, de acordo com os respectivos ganhos esperados, despesas esperadas e riscos (ou oportunidades) associados a esses projetos. De acordo com Goebel e Gruenwald (1999), uma árvore de decisão representa um conjunto de regras para classificação de dados de um conjunto, formado por nós que indicam os testes e decisões sobre esse conjunto.

Tendo o risco (ou oportunidade) de um projeto como uma informação qualitativa, propõe-se, ainda, a utilização da análise de sentimento para quantificação desse risco antes que os dados sejam classificados pela árvore de decisão. Segundo Stine (2019), a análise de sentimento busca atribuir um rótulo emocional a determinado texto, a partir da determinação de uma polaridade para o mesmo.

Dessa forma, a partir da combinação entre as técnicas de análise de sentimento e árvore de decisão, pode-se obter a classificação para os projetos propostos para uma organização, tendo em vista os direcionamentos estratégicos da mesma e a alocação otimizada de recursos, sem que as decisões tenham que ser tomadas por indivíduos. Ou seja, a análise de sentimento pode ser utilizada para a avaliação e quantificação dos riscos e oportunidades dos projetos e a árvore de decisão, para classificar os projetos de acordo com os dados referentes aos benefícios, despesas e riscos quantificados para cada um desses projetos.

Portanto, a questão que orienta o presente trabalho é: como aplicar *machine learning* e *text analytics*, a partir das técnicas de árvore de decisão e análise de sentimento, para classificar projetos com base no retorno financeiro e no risco associado?

Levando em consideração os pontos mencionados anteriormente, será desenvolvido, neste trabalho, um modelo composto por duas aplicações computacionais supervisionadas utilizando a linguagem de programação *Python* capaz de integrar as técnicas de análise de sentimento e árvore de decisão, proporcionando decisões mais assertivas sobre a importância de projetos. Diante disso, essas aplicações são: I) uma para aplicar a análise de sentimento e obter a polaridade dos textos referentes aos riscos dos projetos avaliados de acordo com

objetivos estratégicos definidos pela organização; e II) uma para implementar a árvore de decisão e definir a classificação dos projetos em questão. A partir disso, será realizada uma aplicação piloto do modelo com o intuito de avaliar seu desempenho.

## 1.2. OBJETIVOS

### 1.2.1. OBJETIVO GERAL

O objetivo do presente trabalho é apresentar um modelo capaz de combinar as técnicas de análise de sentimento e árvore de decisão, já bastante difundidas na literatura, para que sejam aplicadas no problema de classificação de projetos. Diante disso, busca-se, a partir do retorno financeiro esperado e do risco associado a cada um dos projetos avaliados, determinar a classificação no que diz respeito à importância de execução.

### 1.2.2. OBJETIVOS ESPECÍFICOS

A partir do objetivo geral destacado anteriormente, podem ser atribuídos os seguintes objetivos específicos a este trabalho:

a) Revisar literatura no que diz respeito à análise de sentimento e à árvore de decisão, bem como seus métodos de aplicação;
b) Desenvolver aplicação computacional supervisionada para obtenção dos resultados da análise de sentimento realizada para os riscos dos projetos avaliados;
c) Desenvolver aplicação computacional supervisionada para indução e obtenção dos resultados da árvore de decisão e, consequentemente, a classificação de cada projeto de acordo com sua importância;
d) Analisar desempenho do modelo construído a partir de aplicação piloto.

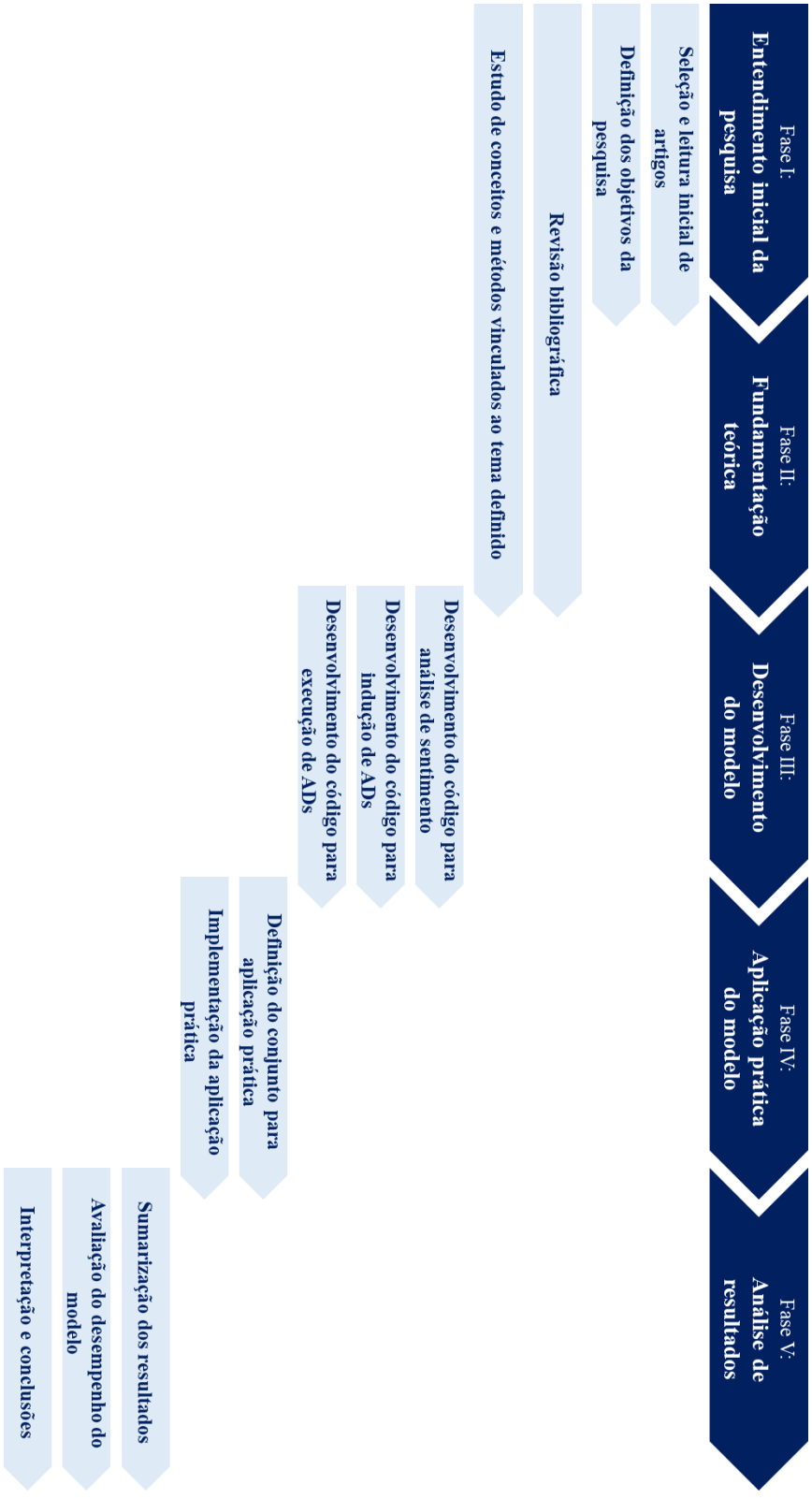
## 2. MÉTODO

O presente trabalho pode ser classificado como uma pesquisa quantitativa axiomática normativa. Segundo Bertrand e Fransoo (2010), um modelo quantitativo é baseado em um conjunto de variáveis que variam em um domínio de acordo com relações quantitativas e causais. Kotronoulas et al. (2023) destacam, ainda, que esse tipo de modelo visa processar dados numéricos para identificar tendências e relações.

Para além disso, Cauchick Miguel (2010) afirma que a pesquisa axiomática é destinada, entre outros, à produção de conhecimento sobre o comportamento de variáveis com base em premissas sobre o comportamento de outras variáveis. Ainda segundo o autor, uma pesquisa quantitativa axiomática normativa é aquela que busca, a partir dos conceitos presentes na literatura, encontrar modelos que prescrevam uma decisão para um problema, destacando-se modelos de programação matemática.

No que diz respeito ao método de pesquisa, buscou-se orientar o desenvolvimento do trabalho em questão a partir de cinco fases, a saber: I) entendimento inicial da pesquisa; II) fundamentação teórica; III) desenvolvimento do modelo; IV) aplicação prática do modelo; e V) análise de resultados. Essas fases podem ser observadas na Figura 1 e serão descritas a seguir.

Figura 1 - Método de pesquisa




Fonte: elaboração própria (2023)

## 2.1. ENTENDIMENTO INICIAL DA PESQUISA

A primeira etapa do método que guiou a presente pesquisa teve como objetivo a contextualização inicial do tema relacionado ao trabalho. Diante disso, foram realizadas buscas iniciais de artigos e trabalhos ligados ao tema em questão para que pudesse ser realizada a seleção e leitura dos principais textos.

Com isso, foi possível entender os principais conceitos e métodos envolvidos com o tema e com o modelo a ser desenvolvido, além de definir os objetivos da presente pesquisa.

## 2.2. FUNDAMENTAÇÃO TEÓRICA

O objetivo desta fase foi embasar a revisão bibliográfica realizada sobre o tema e, consequentemente, possibilitar a aplicação prática dos conceitos e métodos presentes na literatura. Sendo assim, foram lidos os textos selecionados na etapa anterior, além de artigos conexos que tivessem vínculo com o tema abordado. Vale ressaltar que foram buscados trabalhos relacionados aos seguintes temas: “projeto”, “árvore de decisão”, “análise de sentimento” e “matriz de confusão”.

Assim, foi realizada uma breve revisão bibliográfica, sendo possível identificar o estado da arte, principalmente, para abordagens relacionadas à árvore de decisão, à análise de sentimento e temas correlatos. Essa revisão embasou todo o modelo e aplicação prática propostos.

## 2.3. DESENVOLVIMENTO DO MODELO

Nesta etapa foram desenvolvidos o modelo teórico proposto neste trabalho e a aplicação computacional responsável por suportá-lo. Dessa forma, o modelo teórico é baseado na integração entre as técnicas de análise de sentimento e árvore de decisão, permitindo que sejam classificados projetos de acordo com as respectivas importâncias a partir de seus benefícios gerados, despesas e riscos. Tendo isso em vista, foi desenvolvido um modelo composto por três etapas, a saber: 1) coleta de dados sobre os projetos; 2) análise de sentimento; 3) árvore de decisão. Essas etapas serão detalhadas na seção 4.1. do presente trabalho.

O código referente à aplicação computacional foi desenvolvido a partir da linguagem *Python*, através da plataforma *Google Colab*, e apresenta três fases principais: a) execução da análise de sentimento para os riscos (ou oportunidades) relacionados a cada projeto avaliado a partir do conjunto de treinamento vinculado às alavancas estratégicas da organização; b) indução e avaliação da árvore de decisão a partir do conjunto de treinamento selecionado composto por dados quantificados sobre retorno financeiro e risco dos projetos; e c) execução da árvore de decisão a partir dos dados desejados. As fases mencionadas serão detalhadas na seção 4.2. do presente trabalho.

## 2.4. APLICAÇÃO PRÁTICA DO MODELO

Com o objetivo de testar o modelo proposto na prática, foi realizada uma aplicação piloto com um conjunto de 40 projetos definidos inicialmente. Para cada um deles foi realizada a análise de sentimento para os respectivos riscos (ou oportunidades) com a quantificação dos mesmos como resultado. Em seguida esse conjunto foi dividido em dois subconjuntos: um de treinamento e um de teste para que fosse possível induzir a árvore de decisão para classificar os projetos e, a partir disso, avaliar o desempenho desta. Essa fase será detalhada na seção 5 do presente trabalho.

## 2.5. ANÁLISE DE RESULTADOS

A partir dos resultados obtidos com a aplicação piloto, buscou-se consolidar as informações geradas, analisá-las sob o aspecto do desempenho do modelo proposto e, com isso, apresentar as conclusões do presente trabalho.

# 3. REVISÃO BIBLIOGRÁFICA

Nesta seção busca-se apresentar os conceitos e definições presentes na literatura relacionados à: Projeto, Portfólio de Projetos, Análise de Sentimento e Árvore de Decisão. Para além disso, tem-se o intuito, também, de apresentar a forma com que esses conceitos podem ser aplicados na prática. Dessa maneira, são apresentados, ainda, os métodos TF-IDF e SVM, que podem ser utilizados para a realização da análise de sentimento e C4.5, que pode ser utilizado para induzir árvores de decisão, e a matriz de confusão, como ferramenta de avaliação de desempenho de modelos supervisionados.

## 3.1. PROJETO

De acordo com o PMI (2008), um projeto é um esforço temporário, frequentemente utilizado por organizações como meio para atingir objetivos estratégicos, empreendido para criar um produto, serviço ou resultado exclusivo e que, naturalmente, deve possuir um início e um término definidos.

Para Archer e Ghasemzadeh (1999), projeto pode ser definido como um esforço complexo, composto de tarefas inter-relacionadas, com objetivo, cronograma e orçamento bem definidos.

Já para Vargas (2005), projeto é um empreendimento não repetitivo, caracterizado por uma sequência clara e lógica de eventos, com início e fim, para atingir um objetivo claro e definido. Além disso, deve ser conduzido por pessoas dentro de parâmetros predefinidos de tempo, custo, recursos envolvidos e qualidade.

Diante das definições mencionadas serão classificados, no presente trabalho, projetos que possuam, necessariamente as seguintes características:

- Objetivo e resultado claros;
- Parâmetros de custo e recursos envolvidos definidos;
- Duração definida.

### 3.1.1. PORTFÓLIO DE PROJETOS

Um portfólio de projetos pode ser caracterizado como um grupo de projetos gerenciados por uma organização que competem por recursos escassos disponíveis (como pessoas, capital e tempo), já que esses recursos comumente não são suficientes para a execução de todos os projetos propostos (ARCHER; GHASEMZADEH, 1999).

Ainda segundo Archer e Ghasemzadeh (1999), o processo de seleção de um portfólio de projetos é uma atividade periódica voltada para a seleção de um portfólio a partir dos projetos disponíveis para execução, de acordo com os objetivos organizacionais e recursos disponíveis. Moraes e Laurindo (2010) destacam outros critérios que devem ser considerados para a seleção de projetos: eficiência (uso de recursos), eficácia (resultados para a organização) e riscos.

Vale ressaltar, ainda, a gestão de um portfólio de projetos possui três objetivos principais, a saber: maximizar o valor do portfólio, conectar o portfólio com a estratégia organizacional e balancear o portfólio (ELONEN; ARTTO, 2003).

Dessa forma, considerando os critérios para seleção de projetos apresentados por Moraes e Laurindo (2010), foram abordados, no presente trabalho, os seguintes critérios:

- Ganho esperado para o projeto: representando o pilar de resultados para a organização;
- Despesa esperada para o projeto: representando o pilar de uso de recursos;
- Riscos (ou oportunidades).

Para além disso, diante da importância de que os projetos selecionados estejam alinhados com os objetivos estratégicos organizacionais, os riscos dos projetos foram avaliados, neste trabalho, de acordo com esses objetivos.

## 3.2. ANÁLISE DE SENTIMENTO

A análise de sentimento (*sentiment analysis*), ou mineração de opinião, tem como objetivo a atribuição de um rótulo emocional a um texto, ou seja, busca determinar uma polaridade (negativa ou positiva) ao texto avaliado (STINE, 2019). Dessa forma, ela pode ser utilizada para classificar pensamentos, percepções e opiniões de indivíduos sobre um determinado assunto (PANDITA; GONDHI, 2021).

Dentre as principais aplicações da análise de sentimento, pode-se citar como exemplos: precificação de produtos, inteligência de mercado, relações entre países e detecção de riscos (SUN; LUO; CHEN, 2017), sendo que essa última aplicação será abordada no presente trabalho.

De forma geral, segundo Pandita e Gondhi (2021) a análise de sentimento pode ser classificada em três níveis, sendo eles: 1) *document level*, que avalia o texto como um todo como positivo ou negativo sem que haja a divisão em partes menores do mesmo; 2) *sentence level*, que também determina uma polaridade do texto, mas em relação às suas sentenças; 3) *feature level*, que aborda a avaliação de polaridade de forma mais granular, para cada um dos aspectos do texto analisado, sendo extremamente vantajoso para avaliação de características de produtos, por exemplo.

Para Sun, Luo e Chen (2017), um sentimento (ou opinião) pode ser representado por cinco parâmetros: entidade, aspecto referente à entidade, locutor da opinião, tempo em que a opinião é emitida e a opinião sobre o aspecto. Para o exemplo “a tela do celular é resistente”, tem-se os seguintes parâmetros:

- Entidade: “celular”;
- Aspecto: “tela”;
- Sentimento/opinião: positivo.

Não é possível determinar os demais parâmetros com as informações disponíveis no exemplo.

Pode-se afirmar, com isso, que o objetivo da análise de sentimento é determinar cada um dos parâmetros citados para um determinado texto. Apesar disso, nem sempre é necessário que sejam identificados os cinco parâmetros, sendo que, para o *document level*, apenas o sentimento é necessário e, quanto mais granular é o nível da análise, mais parâmetros devem ser identificados (SUN; LUO; CHEN, 2017).

Segundo Balazs e Velásquez (2016), os principais passos que devem ser abordados em uma análise de sentimento são: obtenção dos dados que serão analisados, preparação e pré-processamento dos dados, obtenção de resultados (processo de análise) e sumarização e visualização dos resultados.

Em relação à etapa de preparação e pré-processamento dos dados, destacam-se atividades principais que devem ser realizadas para tratamento dos dados obtidos antes da realização da análise de sentimento (BIRJALI; KASRI; BENI-HSSANE, 2021):

- "Tokenização" (t*okenization*): divisão do texto analisado em elementos menores, chamados *tokens*;
- Remoção de palavras irrelevantes (*stop words removal*): remoção de palavras que não contribuem para análise;
- Marcação de classes gramaticais (*part-of-speech tagging*): reconhecimento da classe gramatical dos elementos do texto analisado;
- "Lematização" (*lemmatization*): conversão das palavras do texto à sua forma base, de acordo com o radical.

No que tange à etapa de obtenção dos resultados mencionada anteriormente, ela pode ser realizada a partir de três técnicas, de acordo com Gutierrez-Batista, Vila e Martin-Bautista (2021): a) algoritmos supervisionados; b) dicionários; e c) sistema baseado em regras (ou *rule-based system*). Ainda segundo os autores, os algoritmos supervisionados fornecem bons resultados em termos de acurácia, mas necessitam de dados previamente classificados e, em função disso, se restringem às classes determinadas inicialmente.

No entanto, a abordagem supervisionada é geralmente utilizada quando existe um conjunto específico de classes e quando há dificuldade em se determinar esse conjunto em função da falta de dados rotulados (BIRJALI; KASRI; BENI-HSSANE, 2021). Nesse sentido, essa abordagem foi selecionada para a elaboração do modelo proposto no presente trabalho.

Para a realização da análise de sentimento segundo a abordagem supervisionada foi utilizada uma combinação dos métodos TD-IDF (*term frequency-inverse document frequency*) e SVM (*support vector machine*), que serão detalhados nas seções a seguir.

Para a avaliação do desempenho da análise de sentimento supervisionada, foi utilizada a matriz de confusão, que será detalhada na seção 3.4. deste trabalho.

### 3.2.1. TERM FREQUENCY – INVERSE DOCUMENT FREQUENCY (TF-IDF)

O TD-IDF (*term frequency-inverse document frequency*) é um método estatístico comumente utilizado para o processamento de linguagem natural (PNL) que permite a

determinação do “peso” de cada elemento do texto analisado, fazendo com que seja possível identificar a importância de palavras em um conjunto de documentos (TRSTENJAK; MIKAC; DONKO, 2014).

Com isso, o método em questão tem como objetivo a determinação da frequência relativa de palavras em um determinado conjunto de documentos a partir do inverso da proporção das palavras avaliadas em todo o conjunto. Ou seja, o método é composto por dois elementos: I) TF: frequência da palavra “i” no documento “j”; e II) IDF: inverso da frequência dos documentos que contenham a palavra “i” (TRSTENJAK; MIKAC; DONKO, 2014).

Ainda segundo Trstenjak, Mikac e Donko (2014), TF-IDF pode ser definido por:

$$\mathrm{tf_{ij}idf_i} = \mathrm{tf_{ij}} * \log_2\left(\frac{\mathrm{N}}{\mathrm{df_i}}\right) \quad (1)$$

Em que N é o número de documentos presentes no conjunto analisado, $tf_{ij}$ é a frequência do termo “i” no documento “j” e $df_i$ é a frequência do termo “i” nos documentos do conjunto em questão.

Vale ressaltar que, dentre os vários métodos para atribuição de “pesos” a termos utilizados para a classificação de textos (como binário, TF e TF-IDF), há destaque para o TF-IDF em função da relativização da frequência de um termo em um documento de acordo com a raridade desse termo no conjunto de documentos analisado (CHEN et al, 2016).

Dessa forma, o método TF-IDF foi utilizado, no presente trabalho, com o intuito de atribuir “pesos” a cada um dos elementos textuais de uma frase avaliada, de forma que fosse possível a atribuição de um vetor composto pelos “pesos” em questão, permitindo, assim a classificação realizada pela análise de sentimento.

### 3.2.2. SUPPORT VECTOR MACHINE (SVM)

O SVM (*support vector machine*) é uma das principais técnicas utilizadas para a classificação de dados. O objetivo dessa técnica é produzir um modelo capaz de prever valores-alvo para um conjunto de teste a partir de um conjunto de treinamento inicialmente dado (HSU; CHANG; LIN, 2003).

Para isso, o SVM busca obter um hiperplano otimizado para separar os conjuntos de classes existentes a partir da maior margem possível para os pontos mais próximos do conjunto de treinamento referente a cada uma das classes (BIRJALI; KASRI; BENI-HSSANE, 2021).

Vale ressaltar, ainda, que o SVM é uma técnica de abordagem linear para a aprendizagem supervisionada, que é capaz de lidar com dados discretos ou contínuos e que apresenta uma boa performance quando aplicada à análise de sentimento (BIRJALI; KASRI; BENI-HSSANE, 2021).

Diante disso, a técnica em questão, foi utilizada no presente trabalho como um classificador para a realização da análise de sentimento supervisionada, de forma que fosse possível obter, a partir do conjunto de treinamento, as polaridades para as frases avaliadas.

## 3.3. ÁRVORE DE DECISÃO

Uma árvore de decisão é uma estrutura em que cada nó não terminal representa um teste ou decisão sobre o conjunto de dados considerado, de forma que os testes que a compõe indicam a sequência de ramos e nós a ser seguida até que seja alcançado um nó terminal e, consequentemente, a classificação seja indicada. Ou seja, uma árvore de decisão representa um conjunto de regras para a classificação de dados presentes em um conjunto (GOEBEL; GRUENWALD, 1999).

Qualquer modelo de classificação pode ser representado por uma espécie de "fluxograma", que pode ser induzido a partir de dois métodos: a) entrevistas com especialistas ou b) generalizações de classificações já registradas anteriormente (QUINLAN, 1993).

Dessa forma, a árvore de decisão, sendo um modelo construído a partir de um conjunto já existente, pode ser considerada como um método de aprendizado de máquina supervisionado e preditivo, na medida em que são capazes de predizer classificações a partir de um conjunto de treinamento previamente definido (ARANTES, 2020).

Uma árvore de decisão, portanto, possui as seguintes características (KINGSFORD; SALZBERG, 2008):

- Cada pergunta está contida em um nó e cada nó pai aponta para um nó filho, que representa cada possível resposta à pergunta;

- Um item é classificado em uma classe seguindo o caminho do nó superior até um nó sem filhos, de acordo com as respostas para cada um dos nós.

Assim, uma árvore de decisão pode ser usada para classificar um caso começando na raiz da árvore e percorrendo-a até encontrar uma folha. Em cada nó de decisão não-folha, o resultado do caso para o teste no nó é determinado e a atenção muda para a raiz da subárvore correspondente a esse resultado. Quando esse processo finalmente leva a uma folha, prevê-se que a classe do caso seja aquela registrada na folha (QUINLAN, 1993).

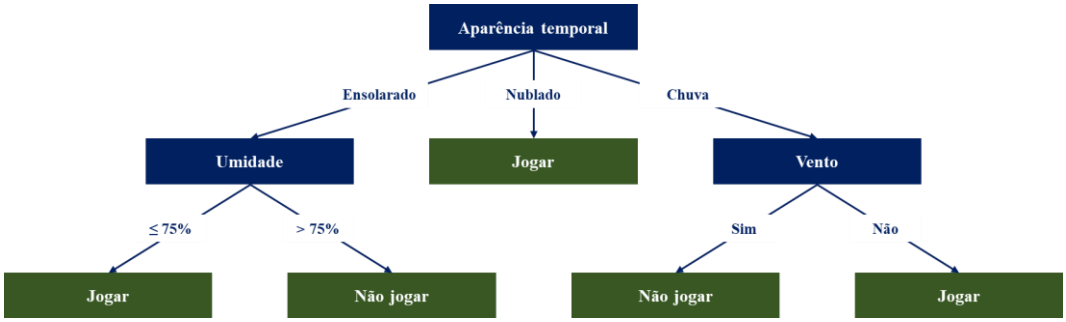
Como exemplo de aplicação de árvores de decisão, tem-se a decisão entre jogar ou não jogar futebol, proposto por Quinlan (1993), a partir das condições climáticas, que consistem em 4 atributos: aparência temporal, temperatura, umidade e vento. A partir disso o conjunto de treino e a sua respectiva árvore de decisão correspondente podem ser observados na Tabela 1 e na Figura 2.

Tabela 1 - Conjunto de treinamento exemplo para indução de árvore de decisão

| # | **Aparência temporal** | **Temperatura (°F)** | **Umidade (%)** | **Vento** | **Classificação (decisão)** |
|---|---|---|---|---|---|
| 1 | Ensolarado | 75 | 70 | Sim | Jogar |
| 2 | Ensolarado | 80 | 90 | Sim | Não jogar |
| 3 | Ensolarado | 85 | 85 | Não | Não jogar |
| 4 | Ensolarado | 72 | 95 | Não | Não jogar |
| 5 | Ensolarado | 69 | 70 | Não | Jogar |
| 6 | Nublado | 72 | 90 | Sim | Jogar |
| 7 | Nublado | 83 | 78 | Não | Jogar |
| 8 | Nublado | 64 | 65 | Sim | Jogar |
| 9 | Nublado | 81 | 75 | Não | Jogar |
| 10 | Chuva | 71 | 80 | Sim | Não jogar |
| 11 | Chuva | 65 | 70 | Sim | Não jogar |
| 12 | Chuva | 75 | 80 | Não | Jogar |
| 13 | Chuva | 68 | 80 | Não | Jogar |
| 14 | Chuva | 70 | 96 | Não | Jogar |

Fonte: Quinlan (1993)

Figura 2 - Exemplo árvore de decisão




Fonte: adaptado de Quinlan (1993)

As árvores de decisão têm algumas vantagens importantes, como interpretabilidade, alta eficiência computacional e capacidade de aprender com poucos dados de treinamento (XU, 2022).

No entanto, podem apresentar problemas no caso da existência de muitas ramificações, na medida que isso pode causar o *overfitting* (ou "superadaptação") que ocorre quando a árvore decisão apresenta uma boa performance com o conjunto de treino e uma performance menos qualificada com o conjunto de dados inserido, ou seja, a árvore acaba classificando corretamente os casos do conjunto de treinamento, mas falhando na classificação de exemplos desconhecidos. Dessa forma, novas ramificações só devem ser feitas em momentos em que o ganho de informação é relevante (KINGSFORD; SALZBERG, 2008).

Existem alguns métodos que podem ser utilizados para a indução de árvores de decisão, como: C4.5, ID3 e CART, sendo o C4.5 uma extensão do ID3. Para além disso, o método C4.5 possui algumas vantagens que podem ser listadas (SHARMA; KUMAR, 2016):

- Lida com atributos discretos e contínuos;
- Lida com valores faltantes no conjunto de dados inicial sem considerá-los nos cálculos de entropia e ganho de informação (exemplo: valores definidos como "?");
- Permite que testes sejam feitos de forma não binária, com dois ou mais resultados (enquanto o método CART permite apenas testes binários).

Dessa forma, para o presente trabalho, foi utilizado o método C4.5, desenvolvido por Quinlan (1993), para a construção do modelo proposto e que será detalhado a seguir. Além disso, para a avaliação de desempenho da árvore de decisão induzida foi utilizada a matriz de confusão, que também será detalhada posteriormente no presente trabalho.

### 3.3.1. C4.5

Assim como diversos algoritmos de indução de árvores de decisão, o método C4.5 se baseia na abordagem de divisão e conquista, a partir de um conjunto de treinamento (QUINLAN, 1993).

Uma das principais etapas do algoritmo vinculado ao método em questão para indução de árvore de decisão diz respeito ao critério de divisão, que representa o momento em que a árvore é dividida em seus nós (LOPES, 2016 apud WITTEN; FRANK; HALL, 2011).

Dessa forma, Quinlan (1993) propõe como critério de divisão, para o método C4.5 o ganho de informação relacionado a um atributo, fazendo com que, a partir da definição do atributo com maior razão de ganho, seja induzida a árvore de decisão.

Quinlan (1993) sugere que a informação gerada por um conjunto S, que é também conhecida por entropia, varia de acordo a probabilidade de que elementos de S pertençam à classe $C_i$ e pode ser calculada a partir do logaritmo de base 2 dessa probabilidade. A partir disso, pode-se concluir que a entropia de um conjunto S com k classes possíveis é definida por:

$$\mathrm{E_{conjunto}}(\mathrm{S}) = -\sum_{\mathrm{i=1}}^{\mathrm{k}} \frac{\mathrm{freq}(\mathrm{C_i}, \mathrm{S})}{|\mathrm{S}|} * \log_2\left(\frac{\mathrm{freq}(\mathrm{C_i}, \mathrm{S})}{|\mathrm{S}|}\right) \quad (2)$$

Ainda segundo Quinlan (1993), ao testar a informação gerada (entropia) por um atributo X com “n” possíveis valores, gera-se um conjunto T, pertencente a S, de forma que, a entropia desse atributo pode ser calculada por:

$$\mathrm{E_{atributo}}(\mathrm{X}) = \sum_{\mathrm{i=1}}^{\mathrm{n}} \frac{|\mathrm{T_i}|}{|\mathrm{S}|} * \mathrm{E_{conjunto}}(\mathrm{T_i}) \quad (3)$$

A medida indicada pela equação acima indica a quantidade de informação necessária para classificar um exemplo após a divisão do conjunto de dados inicial, a partir da seleção do atributo X (QUINLAN, 1993).

Assim, o ganho de informação apresentado na equação abaixo representa o ganho de informação ocorrido quando o conjunto S é particionado segundo um atributo X (QUINLAN, 1993).

$$\mathrm{ganho}(\mathrm{X}) = \mathrm{E_{conjunto}}(\mathrm{S}) - \mathrm{E_{atributo}}(\mathrm{X}) \quad (4)$$

A razão de ganho, que expressa a proporção de informação gerada a partir do teste do atributo X em questão é dada por (QUINLAN, 1993):

$$\text{razão de ganho}(X) = \frac{\text{ganho}(X)}{-\sum_{i=1}^{n} \frac{|T_i|}{|S|} * \log_2\left(\frac{|T_i|}{|S|}\right)} \quad (5)$$

Dessa forma, Quinlan (1993) propõe que o atributo que será atribuído a cada nó durante a indução de uma árvore de decisão seja aquele com maior razão de ganho.

No entanto, os passos apresentados acima são consistentes quando os atributos são discretos, como o atributo "aparência temporal" no exemplo inicial. Para atributos contínuos, como é o caso da "temperatura" nesse mesmo exemplo, devem ser seguidos passos específicos, já que não existem limites claros que podem ser considerados nas equações descritas anteriormente. Assim, Quinlan (1993) propõe um método para que sejam determinados os limites em questão para atributos contínuos.

Seja um atributo A que se deseja testar e que possui os valores ordenados conforme mostrado abaixo:

$$A = \{v_1; v_2; v_3; v_4; v_m\}$$

Para que seja possível determinar a razão de ganho desse atributo A, deve-se encontrar um limite que permita que esse mesmo conjunto seja dividido em dois subconjuntos. Assim, deve-se examinar todas as "m-1" possibilidades de divisão dadas a partir da relação entre dois valores subsequentes. Ou seja, deve-se examinar os pontos médios t de cada intervalo do conjunto, assim como é representado pela equação abaixo (QUINLAN, 1993). Vale ressaltar que pontos subsequentes que respeitem a relação $v_i = v_{i+1}$ não geram um ponto médio t a ser testado, ou seja, o limite a ser definido não deve estar contido no conjunto de treinamento (PAULA, 2002).

$$t = \frac{v_i + v_{i+1}}{2} \quad (6)$$

A partir disso e após testar todos os pontos médios t, deve-se selecionar o limite que maximiza a razão de ganho e, assim, seguir com o processo de indução da árvore de decisão. Esse limite pode ser dado, também, pelo maior valor presente no conjunto de treinamento que não seja superior ao ponto t que maximize a razão de ganho (QUINLAN, 1993).

### 3.3.2. APLICAÇÃO DO MÉTODO C4.5 PARA INDUÇÃO DE ÁRVORE DE DECISÃO

A partir dos passos indicados anteriormente para a indução de árvores de decisão, será retomado o exemplo inicial para uma aplicação prática do método C4.5.

Para calcular a entropia do conjunto de treinamento, deve-se determinar a frequência das classes disponíveis. De acordo com o conjunto apresentado na Tabela 1, existem 9 exemplos que são relacionados à classe “jogar” e 5 exemplos relacionados à classe “não jogar”. Dessa forma, a entropia do conjunto em questão é dada por:

$$\mathrm{E}_{\mathrm{conjunto}}(\mathrm{S}) = -\left(\frac{9}{14}\right) * \log_2\left(\frac{9}{14}\right) - \left(\frac{5}{14}\right) * \log_2\left(\frac{5}{14}\right) = 0{,}940$$

**a) Atributo “aparência temporal”:**

O atributo “aparência temporal” possui três valores distintos: “ensolarado” (5 vezes), “nublado” (4 vezes) e “chuva” (5 vezes), de forma que a entropia para esse atributo é dada por:

$$\mathrm{E}_{\mathrm{atributo}}(\mathrm{X}_1) = \left(\frac{5}{14} * \left(-\left(\frac{2}{5}\right) * \log_2\left(\frac{2}{5}\right) - \left(\frac{3}{5}\right) * \log_2\left(\frac{3}{5}\right)\right)\right) + \left(\frac{4}{14} * \left(-\left(\frac{4}{4}\right) * \log_2\left(\frac{4}{4}\right)\right)\right) + \left(\frac{5}{14} * \left(-\left(\frac{3}{5}\right) * \log_2\left(\frac{3}{5}\right) - \left(\frac{2}{5}\right) * \log_2\left(\frac{2}{5}\right)\right)\right) = 0{,}693$$

Portanto,

$$\text{razão de ganho}(\mathrm{X}_1) = \frac{0{,}940 - 0{,}693}{-\frac{5}{14} * \log_2\left(\frac{5}{14}\right) - \frac{4}{14} * \log_2\left(\frac{4}{14}\right) - \frac{5}{14} * \log_2\left(\frac{5}{14}\right)} = 0{,}156$$

**b) Atributo “temperatura (ºF)”:**

Como indicado anteriormente, para que possa ser realizado o tratamento de atributos contínuos, deve-se, inicialmente, ordenar os valores possíveis do mesmo. Assim, o conjunto do exemplo em questão pode ser observado da seguinte forma:

Tabela 2 - Conjunto de treinamento exemplo ordenado pelo atributo "temperatura (°F)”

| # | Aparência temporal | Temperatura (°F) | Umidade (%) | Vento | Classificação (decisão) |
|---|---|---|---|---|---|
| 8 | Nublado | 64 | 65 | Sim | Jogar |
| 11 | Chuva | 65 | 70 | Sim | Não jogar |

| | | | | | |
|---|---|---|---|---|---|
| 13 | Chuva | 68 | 80 | Não | Jogar |
| 5 | Ensolarado | 69 | 70 | Não | Jogar |
| 14 | Chuva | 70 | 96 | Não | Jogar |
| 10 | Chuva | 71 | 80 | Sim | Não jogar |
| 4 | Ensolarado | 72 | 95 | Não | Não jogar |
| 6 | Nublado | 72 | 90 | Sim | Jogar |
| 1 | Ensolarado | 75 | 70 | Sim | Jogar |
| 12 | Chuva | 75 | 80 | Não | Jogar |
| 2 | Ensolarado | 80 | 90 | Sim | Não jogar |
| 9 | Nublado | 81 | 75 | Não | Jogar |
| 7 | Nublado | 83 | 78 | Não | Jogar |
| 3 | Ensolarado | 85 | 85 | Não | Não jogar |

Fonte: adaptado de Quinlan (1993)

Assim, dado que o conjunto de treinamento possui 14 exemplos, devem ser testadas 13 combinações de valores subsequentes no que tange à sua razão de ganho. Dessa forma, o primeiro valor a ser testado é:

$$t_1 = \frac{v_1 + v_2}{2} = \frac{64 + 65}{2} = 64{,}5$$

Com isso, considerando que o conjunto S inicial deve ser dividido em dois subconjuntos, tendo em vista o valor do atributo “temperatura” e limitados pelo valor 64,5, tem-se:

$$\text{razão de ganho}(X_2) = \frac{0{,}940 - 0{,}892}{-\frac{1}{14} * \log_2(\frac{1}{14}) - \frac{13}{14} * \log_2(\frac{13}{14})} = 0{,}129$$

Aplicando esse mesmo método para as demais combinações possíveis de valores do conjunto de treinamento, com exceção da combinação entre 84 e 85, que deve ser excluída, segundo Quinlan (1993), pelo fato de gerar um conjunto, após a divisão, com apenas um caso de treinamento, tem-se o seguinte resultado:

Tabela 3 - Avaliação de limites para atributo "temperatura (°F)”

| $t_i$ | **Razão de ganho($X_2$)** |
|---|---|
| 64,5 | 0,129 |
| 66,5 | 0,017 |
| 68,5 | 0,001 |
| 69,5 | 0,017 |
| 70,5 | 0,048 |
| 71,5 | 0,001 |

| | |
|---|---|
| 73,5 | 0,001 |
| 77,5 | 0,029 |
| 80,5 | 0,001 |
| 82 | 0,017 |

Fonte: adaptado de Quinlan (1993)

Dessa forma, observa-se que, para i = 1, tem-se o valor máximo para a razão de ganho vinculada ao limite $t_i$, de forma que o limite a ser considerado para o atributo “temperatura” é 65 °F, já que a razão de ganho é para o limite t = 64,5 é máxima.

**c) Atributo “umidade (%)”:**

Tendo, em vista que o atributo “umidade (%)” é contínuo, como o atributo “temperatura (°F)”, deve-se seguir o mesmo método mostrado anteriormente. Diante disso, observa-se o seguinte resultado:

Tabela 4 - Avaliação de limites para atributo "umidade (%)”

| $t_i$ | **Razão de ganho($X_3$)** |
|---|---|
| 67,5 | 0,129 |
| 72,5 | 0,017 |
| 76,5 | 0,048 |
| 79,0 | 0,092 |
| 82,5 | 0,109 |
| 87,5 | 0,029 |
| 92,5 | 0,017 |

Fonte: adaptado de Quinlan (1993)

Dessa forma, observa-se que, para i = 1, tem-se o valor máximo para a razão de ganho vinculada ao limite ti, de forma que o limite a ser considerado para o atributo “umidade” é 70%, já que a razão de ganho é para o limite t = 67,5 é máxima.

**d) Atributo “vento”:**

Tendo em vista que o atributo “vento” é discreto, para determinar sua razão de ganho, deve-se seguir o mesmo método apresentado para o atributo “aparência temporal”. Dessa forma, considerando que esse atributo possui dois valores distintos: “sim” (6 vezes) e “não” (8 vezes), tem-se a seguinte razão de ganho para o atributo “vento”:

$$E(X_4) = \left(\frac{6}{14} * \left(-\left(\frac{3}{6}\right) * \log_2\left(\frac{3}{6}\right) - \left(\frac{3}{6}\right) * \log_2\left(\frac{3}{6}\right)\right)\right) + \left(\frac{8}{14} * \left(-\left(\frac{6}{8}\right) * \log_2\left(\frac{6}{8}\right) - \left(\frac{2}{8}\right) * \log_2\left(\frac{2}{8}\right)\right)\right) = 0{,}892$$

Portanto,

$$\text{razão de ganho}(X_4) = \frac{0{,}940 - 0{,}693}{-\frac{6}{14} * \log_2\left(\frac{6}{14}\right) - \frac{8}{14} * \log_2\left(\frac{8}{14}\right)} = 0{,}049$$

Com a definição das razões de ganho vinculadas aos quatro atributos, deve-se avaliar seus resultados e selecionar a maior razão de ganho para que o atributo referente a ela seja incluído na árvore de decisão em seu processo de indução.

Assim, tem-se as seguintes razões de ganho para os atributos analisados:

Tabela 5 - Comparação entre razões de ganho para primeira divisão da árvore de decisão

| **Atributo** | **Razão de ganho** |
|---|---|
| Aparência temporal | 0,156 |
| Temperatura (°F) | 0,129 |
| Umidade (%) | 0,129 |
| Vento | 0,049 |

Fonte: adaptado de Quinlan (1993)

Com a definição de que o atributo “aparência temporal” deve ser o primeiro nó da árvore de decisão, já que a razão de ganho vinculada a ele é a maior, a sequência da indução da árvore de decisão em questão se dá a partir da análise dos conjuntos gerados por cada valor do atributo “aparência temporal”.

Assim, avaliando o conjunto de exemplos mostrado abaixo, em que a aparência temporal é “ensolarado”, e seguindo o mesmo método mostrado anteriormente, tem-se o cenário apresentado nas tabelas 6 e 7:

Tabela 6 - Conjunto de treinamento exemplo (aparência temporal "ensolarado")

| # | **Aparência temporal** | **Temperatura (°F)** | **Umidade (%)** | **Vento** | **Classificação (decisão)** |
|---|---|---|---|---|---|
| 1 | Ensolarado | 75 | 70 | Sim | Jogar |
| 2 | Ensolarado | 80 | 90 | Sim | Não jogar |
| 3 | Ensolarado | 85 | 85 | Não | Não jogar |

| | | | | | |
|---|---|---|---|---|---|
| 4 | Ensolarado | 72 | 95 | Não | Não jogar |
| 5 | Ensolarado | 69 | 70 | Não | Jogar |

Fonte: adaptado de Quinlan (1993)

Tabela 7 - Comparação entre razões de ganho (aparência temporal "ensolarado")

| **Atributo** | **t** | **Razão de ganho** |
|---|---|---|
| Temperatura (°F) | 72 | 0,446 |
| Umidade (%) | 75 | 1,000 |
| Vento | - | 0,021 |

Fonte: adaptado de Quinlan (1993)

Com isso, o próximo nó da árvore deve ser representado pelo atributo “umidade”, sendo que, a partir dele já é possível concluir a classificação vinculada ao exemplo, o que indica que não há necessidade da presença de mais nós. Dessa forma, pode-se analisar o conjunto de exemplos mostrado abaixo, em que a aparência temporal é “chuva”, e seguindo o mesmo método mostrado anteriormente, tem-se o cenário apresentado nas tabelas 8 e 9:

Tabela 8 - Conjunto de treinamento exemplo (aparência temporal "chuva")

| **#** | **Aparência temporal** | **Temperatura (°F)** | **Umidade (%)** | **Vento** | **Classificação (decisão)** |
|---|---|---|---|---|---|
| 11 | Chuva | 65 | 70 | Sim | Não jogar |
| 13 | Chuva | 68 | 80 | Não | Jogar |
| 14 | Chuva | 70 | 96 | Não | Jogar |
| 10 | Chuva | 71 | 80 | Sim | Não jogar |
| 12 | Chuva | 75 | 80 | Não | Jogar |

Fonte: adaptado de Quinlan (1993)

Tabela 9 - Comparação entre razões de ganho (aparência temporal "chuva")

| **Atributo** | **t** | **Razão de ganho** |
|---|---|---|
| Temperatura (°F) | 68 | 0,446 |
| Umidade (%) | 80 | 0,446 |
| Vento | - | 1,000 |

Fonte: adaptado de Quinlan (1993)

A partir desse resultado pode-se afirmar que o próximo nó da árvore de decisão em questão deve ser representado pelo atributo “vento”, sendo que, a partir dele já é possível concluir a classificação vinculada ao exemplo, o que indica que não há necessidade da presença de mais nós de decisão.

Para além disso, como para os exemplos vinculados ao conjunto em que os valores do atributo “aparência temporal” são “nublado” não exigem um novo nó de decisão, uma vez que a classificação é a mesma para todos eles, pode-se afirmar que a indução da árvore de decisão está finalizada, sendo possível obter a árvore representada na Figura 2, apresentada anteriormente.

## 3.4. MATRIZ DE CONFUSÃO

Segundo Monard e Baranauskas (2003), um conjunto de exemplos para ser classificado por um modelo de classificação deve ser dividido em dois subconjuntos: um de treinamento e um de teste. O conjunto de treinamento é utilizado para treinar o modelo e, no caso de árvores de decisão, é utilizado para induzi-las, enquanto que o conjunto de teste deve ser utilizado para avaliar o desempenho do modelo, com o cálculo da sua acurácia, por exemplo. Santos, Steiner e Lima (2022) propõem que o conjunto de treinamento represente 70% do conjunto de exemplos e o conjunto de teste, 30%.

Para que se possa avaliar o desempenho de um modelo de classificação, pode-se utilizar a matriz de confusão. Em sua forma genérica, quando existem duas classes possíveis, uma matriz de confusão é uma matriz 2x2 que é construída a partir do cruzamento entre os valores da classe verdadeira e os valores da classe predita, assim como é mostrado abaixo (FAWCETT, 2006).

Tabela 10 - Matriz de confusão 2x2

| | | **Classe verdadeira** | |
|---|---|---|---|
| | | **Negativo** | **Positivo** |
| **Classe predita (modelo)** | **Negativo** | VN | FN |
| | **Positivo** | FP | VP |

Fonte: Fawcett (2006)

Os valores presentes na matriz em questão são classificados, segundo Ramos et al. (2018), como:

- **Verdadeiros negativos (VN):** quando a classificação real é negativa e foi classificada corretamente como negativa pelo modelo;
- **Falsos negativos (FN):** quando a classificação real é positiva e foi classificada incorretamente como negativa pelo modelo;

- **Falsos positivos (FP):** quando a classificação real é negativa e foi classificada incorretamente como positiva pelo modelo;
- **Verdadeiros positivos (VP):** quando a classificação real é positiva e foi classificada corretamente como positiva pelo modelo.

A partir disso, segundo Fawcett (2006), a acurácia A de um modelo de classificação é dada pela porcentagem de exemplos classificados corretamente em relação ao total de exemplos avaliados, assim como é mostrado abaixo para um modelo com duas classes:

$$\mathrm{A(2x2)} = \frac{\mathrm{VN} + \mathrm{VP}}{\mathrm{VN} + \mathrm{FN} + \mathrm{FP} + \mathrm{VP}} \tag{7}$$

No entanto, no presente trabalho, são abordadas mais de duas classificações distintas, tornando-se necessário utilizar uma matriz de confusão de dimensões superiores. A partir do que é proposto por Monard e Baranauskas (2003) a matriz de confusão 3x3 pode ser visualizada da seguinte forma:

Tabela 11 - Matriz de confusão 3x3

| **Classe** | **Verdadeira C1** | **Verdadeira C2** | **Verdadeira C3** |
|---|---|---|---|
| **Predita C1** | M11 | M12 | M13 |
| **Predita C2** | M21 | M22 | M23 |
| **Predita C3** | M31 | M32 | M33 |

Fonte: adaptado de Monard e Baranauskas (2003)

Com isso, segundo Matos et al. (2009) a acurácia do modelo de classificação abordado neste trabalho atrelado à essa matriz é dada por:

$$\mathrm{A(3x3)} = \frac{\mathrm{M}_{11} + \mathrm{M}_{22} + \mathrm{M}_{33}}{\mathrm{M}_{11} + \mathrm{M}_{12} + \mathrm{M}_{13} + \mathrm{M}_{21} + \mathrm{M}_{22} + \mathrm{M}_{23} + \mathrm{M}_{31} + \mathrm{M}_{32} + \mathrm{M}_{33}} \tag{8}$$

Nesse sentido, extrapolando aquilo que é proposto por Matos et al. (2009), tem-se a seguinte equação para a acurácia de uma matriz de confusão:

$$\mathrm{A} = \frac{\text{Total de elementos classificados corretamente}}{\text{Total de elementos classificados}} \tag{9}$$

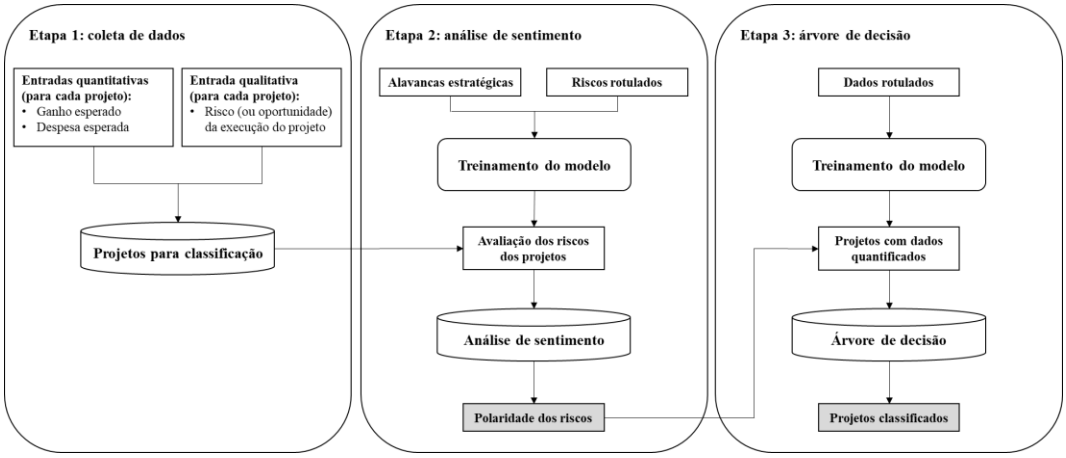
# 4. DESENVOLVIMENTO DO MODELO

Busca-se, nesta seção, apresentar o modelo desenvolvido para integração das técnicas de análise de sentimento e árvore de decisão. Nesse sentido, será detalhado, inicialmente, o modelo teórico para que, em seguida, sejam expostos o modelo computacional e os códigos que suportam o modelo em questão.

## 4.1. MODELO TEÓRICO

De forma geral, o modelo proposto no presente trabalho é composto por três grandes etapas: 1) coleta de dados e definição dos projetos que devem ser classificados; 2) análise de sentimento dos riscos ou oportunidades determinados para os projetos avaliados; e 3) árvore de decisão para classificar os projetos em questão. A relação entre essas etapas pode ser observada na Figura 3.

Figura 3 - Modelo teórico proposto




Fonte: elaboração própria (2023)

Nesse sentido, o modelo tem como objetivo classificar projetos e suas respectivas importâncias de execução de acordo com: o retorno financeiro (representado pelo ganho esperado e pela despesa esperada) e os riscos (ou oportunidades) vinculados à execução desses projetos.

Dessa forma, os projetos para classificação, coletados na primeira etapa do modelo ilustrado na Figura 3, devem conter as informações relacionadas ao retorno financeiro mencionadas acima (entradas quantitativas) e aos riscos associados a cada um (entrada qualitativa).

A etapa de análise de sentimento busca avaliar os riscos associados aos projetos de forma a quantificá-los e torná-los mensuráveis, já que não existe uma quantidade finita de valores que representem o atributo dos riscos dos projetos. Diante dessa necessidade, propõe-se a implementação da análise de sentimento supervisionada, fazendo com que seja necessário um conjunto de treinamento do modelo composto por: exemplos de riscos (ou oportunidades) e polaridades referentes a eles. Além disso, como citado anteriormente neste trabalho, é fundamental que as polaridades atribuídas aos riscos estejam vinculadas aos objetivos estratégicos da organização.

Tendo as informações referentes aos projetos quantificadas, propõe-se a utilização da árvore de decisão para classificação dos projetos de acordo com a importância de execução. A terceira etapa do modelo, portanto, é utilizada para retornar a classificação de cada projeto a partir dos parâmetros iniciais já mencionados. No entanto, assim como a análise de sentimento proposta, sendo a árvore de decisão uma aplicação supervisionada, torna-se necessária a existência de um conjunto de treinamento, que deve apresentar exemplos de projetos com detalhamento de: ganho esperado, despesa esperada, riscos (ou oportunidades) da execução do projeto e classificação desejada.

## 4.2. MODELO COMPUTACIONAL

Esta seção aborda o desenvolvimento do modelo computacional utilizado para a integração entre a análise de sentimento e a árvore de decisão. Como mencionado na seção 2.3. do presente trabalho, o modelo em questão é composto por 3 etapas, a saber:

a) Execução da análise de sentimento para os riscos associados aos projetos com base em dados previamente rotulados;
b) Indução e avaliação de desempenho da árvore de decisão para classificar os projetos com base em dados previamente rotulados;
c) Execução da árvore de decisão a partir dos dados de projetos que se deseja avaliar.

Diante disso, serão detalhados, a seguir, os códigos desenvolvidos para cada uma das etapas mencionadas. Vale ressaltar que esses códigos foram desenvolvidos utilizando a linguagem *Python* e a plataforma *Google Colaboratory*, que se trata de um ambiente de desenvolvimento em nuvem oferecido pela *Google*.

### 4.2.1. EXECUÇÃO DA ANÁLISE DE SENTIMENTO

Para a primeira etapa do modelo computacional desenvolvido, em que se busca executar a análise de sentimento, o usuário deve, inicialmente, apresentar, como entrada, uma tabela com dados já rotulados, no que diz respeito à polaridade esperada para determinados exemplos de riscos (ou oportunidades) de projetos. Tendo em vista que essa classificação deve ser dada de acordo com a estratégia da organização, propõe-se que a tabela a ser apresentada contenha as seguintes informações: “alavanca estratégica”, “risco/oportunidade”, “polaridade”, assim como é mostrado na Tabela 12. Vale ressaltar que os riscos e oportunidades vinculados aos projetos devem estar em inglês para que haja um melhor processamento do modelo.

Tabela 12 - Modelo para dados rotulados - análise de sentimento

| **Alavanca Estratégica** | **Risco/Oportunidade** | **Polaridade** |
|---|---|---|
| - | - | - |
| - | - | - |
| - | - | - |

Fonte: elaboração própria (2023)

A partir disso, pode-se executar a análise de sentimento. Como mencionado anteriormente, foi utilizado um método supervisionado para a realização da análise. Dessa forma, o código elaborado é utilizado para treinar, a partir das informações que constam no conjunto de dados mostrados na Tabela 12, o modelo para a realização da análise de sentimento. O código em questão apresenta como resultados, para além do treinamento do modelo, os conjuntos de treinamento e teste e a acurácia do modelo.

Em um primeiro momento, deve-se realizar a preparação do modelo, de forma que sejam importadas as bibliotecas que serão utilizadas em todo o modelo. Com isso, essas bibliotecas podem ser observadas na Figura 4.

Figura 4 - Preparação do modelo (bibliotecas utilizadas)

```
# Preparação do modelo (bibliotecas)
!pip install scikit-learn
!pip install -U spacy
!python -m spacy download en
!python -m spacy download en_core_web_sm

import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
import spacy
from spacy import displacy
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.pipeline import Pipeline
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
import string
from spacy.lang.en.stop_words import STOP_WORDS
from sklearn.svm import LinearSVC
```

Fonte: elaboração própria (2023)

A Figura 5 apresenta a primeira parte do modelo utilizado para a realização da análise de sentimento. Nesse sentido, são realizados os seguintes passos iniciais: definição do conjunto de dados rotulados para a realização do treinamento do modelo referente à análise de sentimento, tratamento desse conjunto (com o ajuste do tipo de variável relacionada à polaridade para *string*) e pré-processamento dos dados.

A etapa de pré-processamento dos dados, como já mencionada na seção 3.2. do presente trabalho, é composta por alguns passos principais: “tokenização”, remoção de palavras irrelevantes, marcação de classes gramaticais e “lematização”. Para isso, foi utilizada a biblioteca *spaCy*, que é reconhecida por possibilitar o processamento de linguagem natural em *Python* com diversas ferramentas e modelos vinculados a várias línguas.

Vale ressaltar, ainda, que, para a aplicação dos métodos TF-IDF e SVM, também mencionados, respectivamente, nas seções 3.2.1. e 3.2.2. deste trabalho, foi utilizada a biblioteca *scikit-learn*, que oferece ferramentas de aprendizado de máquina em *Python* para análise de dados e modelagem preditiva, podendo ser utilizada, também, para modelos de classificação.

O método TF-IDF, no caso do modelo proposto é utilizado para criar vetores para cada um dos riscos (ou oportunidades) avaliados, de forma que eles sejam constituídos pelos “pesos” de cada uma das palavras que os compõem. A partir desses vetores é possível que seja aplicado o método SVM para determinação dos hiperplanos de separação entre as classes e a consequente obtenção da classificação dos riscos em questão.

Figura 5 – Treinamento do modelo utilizado para análise de sentimento (parte 1)

```
# Definição e tratamento do conjunto de dados rotulados para análise de sentimento
data_SA = pd.read_excel('/content/drive/MyDrive/Dados - Modelo TCC/Conjunto de Treinamento - SA.xlsx')
data_SA['Polaridade'] = data_SA['Polaridade'].astype(str)

# Pré-processamento dos dados com definição de pontuações, língua e "stop words" - biblioteca spaCy
punct = string.punctuation
nlp = spacy.load('en_core_web_sm')
stopwords = list(STOP_WORDS)

def text_data_cleaning(sentence):
  doc = nlp(sentence)

  tokens = []
  # Processo de "tokenização"
  for token in doc:
    # Reconhecimento de classe gramatical
    if token.lemma_ != "-PRON-":
      # Processo de "lematização"
      temp = token.lemma_.lower().strip()
    else:
      temp = token.lower_
    tokens.append(temp)

  cleaned_tokens = []
  for token in tokens:
    # Ajuste para considerar somente tokens que não sejam pontuações ou "stop words"
    if token not in stopwords and token not in punct:
      cleaned_tokens.append(token)
  return cleaned_tokens

# Aplicação de técnicas TF-IDF e SVM - biblioteca scikit-learn
tfidf = TfidfVectorizer(tokenizer = text_data_cleaning)
sup_classifier_SA = LinearSVC()

# Definição de atributos e classes
X_SA = data_SA['Risco/Oportunidade']
y_SA = data_SA['Polaridade']
```

Fonte: elaboração própria (2023)

A Figura 6 apresenta a segunda parte do modelo utilizado para a realização da análise de sentimento. Nela é possível observar o treinamento, de fato, do modelo construído a partir dos dados definidos inicialmente. Para isso, foram utilizados o método *fit* e a classe *pipeline* da biblioteca *scikit-learn*, já mencionada anteriormente. A classe *pipeline* permite encadear os passos de pré-processamento e modelagem em um único estimador, formando um fluxo de trabalho para o aprendizado de máquina. Dessa forma, foram utilizados como parâmetros os resultados dos métodos TF-IDF e SVM.

Já o método *fit* é utilizado para o treinamento de um modelo de aprendizado de máquina de acordo com os dados fornecidos inicialmente, permitindo que o modelo “aprenda” as relações existentes entre as entradas e saídas que constam no conjunto de dados. Vale destacar que o tamanho adotado para o conjunto de teste é dado por 30% do tamanho do conjunto de dados inicial.

Figura 6 - Treinamento do modelo utilizado para análise de sentimento (parte 2)

```
# Treinamento do modelo (com tamanho do conjunto de teste equivalente a 30%) - biblioteca scikit-learn
test_size_SA = 0.3
id_random_SA = 1

X_train_SA, X_test_SA, y_train_SA, y_test_SA = train_test_split(X_SA, y_SA, test_size = test_size_SA, random_state = id_random_SA)
classifier_SA = Pipeline([('tfidf', tfidf), ('clf', sup_classifier_SA)])
classifier_SA.fit(X_train_SA, y_train_SA)

# Obtenção da classe predita a partir do conjunto de teste
y_predict_SA = classifier_SA.predict(X_test_SA)
```

Fonte: elaboração própria (2023)

A Figura 7 apresenta o código que permite a visualização dos conjuntos utilizados para treinamento e teste do modelo construído, obtidos a partir da função *train_test_split*, mostrada na Figura 6. Essa função retorna os atributos e classes dos conjuntos de treinamento e teste a partir do conjunto de dados inicial, da proporção entre os conjuntos de treinamento e teste e de um "ID de aleatoriedade" utilizado para identificar aquela divisão específica.

Figura 7 – Conjuntos de treinamento e teste da análise de sentimento

```
# Obtenção do conjunto de treinamento
train_data_SA = pd.DataFrame(X_train_SA)
train_data_SA['Alavanca Estratégica'] = 0

for i in train_data_SA.index:
  line_train = data_train_SA.loc[i]
  train_data_SA.loc[i] = [train_data_SA.loc[i]['Risco/Oportunidade'], line_train['Alavanca Estratégica']]

train_data_SA['Polaridade'] = y_train_SA

train_data_SA.reset_index(drop=True, inplace=True)
new_order = ['Alavanca Estratégica', 'Risco/Oportunidade', 'Polaridade']
train_data_SA = train_data_SA[new_order]

# Obtenção do conjunto de teste
test_data_SA = pd.DataFrame(X_test_SA)
test_data_SA['Alavanca Estratégica'] = 0

for i in test_data_SA.index:
  line_test = data_train_SA.loc[i]
  test_data_SA.loc[i] = [test_data_SA.loc[i]['Risco/Oportunidade'], line_test['Alavanca Estratégica']]

test_data_SA['Polaridade'] = y_test_SA

test_data_SA.reset_index(drop=True, inplace=True)
new_order = ['Alavanca Estratégica', 'Risco/Oportunidade', 'Polaridade']
test_data_SA = test_data_SA[new_order]
```

Fonte: elaboração própria (2023)

A Figura 8 apresenta o código utilizado para avaliação do modelo construído para a realização da análise de sentimento. Sendo assim, a partir do conjunto de teste e do conjunto predito (vinculado ao modelo treinado) foi utilizada a função *confusion_matrix* da biblioteca *scikit-learn* para gerar a matriz de confusão referente ao modelo em questão. Com isso, comparando os "valores corretos" (classificação predita idêntica à classificação verdadeira) com o total de valores do conjunto, foi possível calcular a acurácia do modelo.

Figura 8 – Avaliação do modelo utilizado para análise de sentimento

```
# Obtenção da acurácia do modelo para a análise de sentimento a partir da matriz de confusão
correct_values_SA = 0
total_values_SA = 0

confusion_matrix_SA = confusion_matrix(y_test_SA, y_predict_SA)
for i in range(0, len(data_train_SA['Polaridade'].unique())):
  for j in range(0, len(data_train_SA['Polaridade'].unique())):
    if i == j and confusion_matrix_SA[i,j] > 0:
      correct_values_SA = correct_values_SA + confusion_matrix_SA[i,j]
      total_values_SA = total_values_SA + confusion_matrix_SA[i,j]
    elif i != j and confusion_matrix_SA[i,j] > 0:
      total_values_SA = total_values_SA + confusion_matrix_SA[i,j]

accuracy_SA = (correct_values_SA/total_values_SA)*100
print('acuracia do modelo: ', round(accuracy_SA,1), '%')
```

Fonte: elaboração própria (2023)

Dessa forma, para que se possa obter os resultados da análise de sentimento para o conjunto que se deseja avaliar, o usuário deve incluir, como entrada do código, uma tabela contendo as seguintes informações: “projeto”, “ganho esperado”, “despesa esperada”, “riscos/oportunidades da execução do projeto”, assim como é mostrado na Tabela 13.

Tabela 13 - Modelo para execução da análise de sentimento

| **Projeto** | **Ganho Esperado** | **Despesa Esperada** | **Riscos/Oportunidades da Execução do Projeto** |
|---|---|---|---|
| Projeto A | - | - | - |
| Projeto B | - | - | - |
| Projeto C | - | - | - |

Fonte: elaboração própria (2023)

O resultado da análise, com isso, é dado pela polaridade de cada um dos riscos e oportunidades incluídos na Tabela 13. Portanto, o resultado final da primeira etapa do modelo computacional em questão pode ser representado pela Tabela 14, que é constituída pelas informações de “ganho esperado”, “despesa esperada” e “polaridade dos riscos” para cada um dos projetos avaliados.

Tabela 14 - Resultado final da análise de sentimento

| **Projeto** | **Ganho Esperado** | **Despesa Esperada** | **Polaridade dos Riscos** |
|---|---|---|---|
| Projeto A | - | - | - |
| Projeto B | - | - | - |
| Projeto C | - | - | - |

Fonte: elaboração própria (2023)

O código utilizado para que a tabela ilustrada acima seja obtida pode ser visualizado na Figura 9.

Figura 9 - Obtenção do resultado final da análise de sentimento

```
# Definição do conjunto de projetos que se deseja avaliar
df_SA = pd.read_excel('/content/drive/MyDrive/Dados - Modelo TCC/Conjunto de Treinamento - DT.xlsx')

# Execução da análise de sentimento para os projetos que se deseja avaliar
X_result_SA = df_SA['Riscos/Oportunidades Execução do Projeto']
Y_result_SA = classifier_SA.predict(X_result_SA)

result_SA = df_SA
result_SA['Polaridade dos Riscos'] = Y_result_SA
sup = result_SA['Classificação']
result_SA = result_SA.drop('Classificação', axis=1)
result_SA['Classificação'] = sup
result_SA = result_SA.drop('Riscos/Oportunidades Execução do Projeto', axis=1)
```

Fonte: elaboração própria (2023)

### 4.2.2. INDUÇÃO E AVALIAÇÃO DE DESEMPENHO DA ÁRVORE DE DECISÃO

Como já mencionado em seções anteriores, para que seja possível realizar a indução de árvores de decisão, é fundamental que existam dados previamente rotulados utilizados para treinar e testar a árvore. No caso do presente trabalho, devem existir projetos já classificados no que diz respeito às suas respectivas importâncias.

Dessa forma, espera-se que exista uma base com o formato apresentado na tabela abaixo para que seja possível realizar essa indução, sendo que a polaridade deve ser obtida como resultado da análise de sentimento, apresentada na seção anterior.

Tabela 15 - Modelo conjunto de treinamento - árvore de decisão

| **Projeto** | **Ganho Esperado** | **Despesa Esperada** | **Polaridade** | **Classificação** |
|---|---|---|---|---|
| Projeto A | - | - | - | - |
| Projeto B | - | - | - | - |
| Projeto C | - | - | - | - |

Fonte: elaboração própria (2023)

Com isso torna-se possível atingir um dos objetivos da presente etapa do modelo computacional, ou seja, induzir a árvore de decisão para classificação dos projetos. Para isso, o código foi dividido em 2 fases que serão detalhadas a seguir: tratamento dos dados e execução da indução da árvore de decisão.

A primeira fase diz respeito ao tratamento da base coletada como entrada do modelo. Nesse sentido, busca-se fazer com que a tabela seja composta somente pelas colunas: “ganho esperado”, “despesa esperada”, “polaridade dos riscos” e “classificação”. Afinal, são essas as informações relevantes para a indução da árvore de decisão. Ou seja, considera-se os valores dos atributos (ou *features*) e das classes definidas para cada exemplo. Dessa forma, tem-se o seguinte código utilizado para a realização da presente fase:

Figura 10 - Tratamento de dados para indução árvore de decisão

```
# Base de dados utilizada equivalente ao resultado da análise de sentimento
data = result_SA

# Tratamento de dados e armazenamento do nome das colunas
data = data.drop('Projeto', axis=1)
data['Polaridade dos Riscos'] = data['Polaridade dos Riscos'].astype(float)
data['Despesa Esperada'] = data['Despesa Esperada'].astype(float)
data['Ganho Esperado'] = data['Ganho Esperado'].astype(float)
column0 = data.columns[0]
column1 = data.columns[1]
column2 = data.columns[2]
column3 = data.columns[3]
```

Fonte: elaboração própria (2023)

A segunda fase do código, voltada para a indução, de fato, da árvore de decisão pode ser observada nas figuras a seguir e contempla o processo de indução a partir do método C4.5 apresentado na seção 3.3.1. do presente trabalho.

A Figura 11 mostra a inicialização de duas classes: *Node* e *DecisionTreeClassifier*, que são responsáveis por caracterizar os nós da árvore de decisão e a árvore de decisão em si, que será induzida. Diante disso, define-se os parâmetros que podem ser visualizados na figura em questão para os nós de decisão, para os nós folha e para a árvore de decisão.

Figura 11 - Indução da árvore de decisão (parte 1)

```
# Descrição dos nós da árvore de decisão
class Node():
  def __init__(self, feature_index=None, threshold=None, left=None, right=None, info_gain=None, value=None):

    # Descrição dos nós de decisão
    self.feature_index = feature_index
    self.threshold = threshold
    self.left = left
    self.right = right
    self.info_gain = info_gain

    # Descrição dos nós folha
    self.value = value

# Indução da árvore de decisão
class DecisionTreeClassifier():
  def __init__(self, min_samples_split=2, max_depth=2):

    # Inicialização da raiz da árvore de decisão
    self.root = None

    # Definição das condições de parada da geração da árvore de decisão
    self.min_samples_split = min_samples_split
    self.max_depth = max_depth
```

Fonte: elaboração própria (2023)

A Figura 12 apresenta a função denominada como *build_tree*, que é responsável por induzir a árvore de decisão de acordo com os dados inseridos inicialmente. Nesse sentido, a função é composta pelos passos identificados na seção 3.3.1. do presente trabalho, referente ao método C4.5. Assim, a função em questão depende de outras funções de suporte, mas, de maneira geral, visa definir o critério otimizado de divisão e, com isso, induzir a árvore de decisão desejada.

Figura 12 - Indução da árvore de decisão (parte 2)

```
# Função para construção da árvore de decisão
def build_tree(self, dataset, curr_depth=0):

  X, Y = dataset[:,:-1], dataset[:,-1]
  num_samples, num_features = np.shape(X)

  # Divisões até que as condições de parada sejam atingidas
  if num_samples>=self.min_samples_split and curr_depth<=self.max_depth:
      # Definição o valor orimizado para o critério de divisão
      best_split = self.get_best_split(dataset, num_samples, num_features)
      # Valor da razão de ganho deve ser positivo para que seja necessária uma nova divisão da árvore de decisão
      if best_split["info_gain"]>0:
          # Construção da árvore à esquerda do nó
          left_subtree = self.build_tree(best_split["dataset_left"], curr_depth+1)
          # Construção da árvore à direita do nó
          right_subtree = self.build_tree(best_split["dataset_right"], curr_depth+1)
          # Retorna nó de decisão
          return Node(best_split["feature_index"], best_split["threshold"],
                      left_subtree, right_subtree, best_split["info_gain"])

  # Valor do nó folha
  leaf_value = self.calculate_leaf_value(Y)

  # Retorna nó folha
  return Node(value=leaf_value)
```

Fonte: elaboração própria (2023)

A Figura 13 apresenta a função elaborada para definir o critério de divisão otimizado. Sendo assim, essa função compara os diferentes atributos de um conjunto de dados e, para cada atributo, compara a razão de ganho gerada pelos limites possíveis de um conjunto de um atributo. Dessa forma, a função em questão retorna o critério de divisão que deve direcionar a sequência da indução da árvore de decisão.

Figura 13 - Indução da árvore de decisão (parte 3)

```
# Função para obtenção do critério de divisão otimizado
def get_best_split(self, dataset, num_samples, num_features):

  # Armazenando o valor otimizado para o critério de divisão
  best_split = {}
  max_info_gain = -float('inf')

  # Percorrendo cada um dos atributos e cada um dos possíveis valores de limite para teste
  for feature_index in range(num_features):
      feature_values = dataset[:, feature_index]
      unique_thresholds = np.unique(feature_values)
      possible_thresholds = np.array([])
      for i in range(0, len(unique_thresholds)-1):
        support = np.array([(unique_thresholds[i]+unique_thresholds[i+1])/2])
        possible_thresholds = np.concatenate((possible_thresholds,support))
      for threshold in possible_thresholds:
          dataset_left, dataset_right = self.split(dataset, feature_index, threshold)
          if len(dataset_left)>0 and len(dataset_right)>0:
              y, left_y, right_y = dataset[:, -1], dataset_left[:, -1], dataset_right[:, -1]
              # Determinação do ganho de informação apenas para limites que dividam o conjunto em mais de uma amostra
              if len(dataset_left)>1 and len(dataset_right)>1:
                  curr_info_gain = self.information_gain(y, left_y, right_y, "entropy")
              else:
                  curr_info_gain = float(0)
              # Atualização do valor otimizado para o critério de divisão
              if curr_info_gain>max_info_gain:
                  best_split["feature_index"] = feature_index
                  best_split["threshold"] = threshold
                  best_split["dataset_left"] = dataset_left
                  best_split["dataset_right"] = dataset_right
                  best_split["info_gain"] = curr_info_gain
                  max_info_gain = curr_info_gain

  # Retorna o valor otimizado para o critério de divisão
  return best_split
```

Fonte: elaboração própria (2023)

A Figura 14 apresenta as funções de suporte utilizadas para: a) dividir o conjunto de dados para avaliação da razão de ganho para um determinado limite testado; b) cálculo da

entropia; c) cálculo da razão de ganho de um determinado atributo; d) definição do valor de um nó folha. Nesse sentido, essas funções apoiam as funções apresentadas anteriormente para que a árvore de decisão possa ser induzida de acordo com o método C4.5.

Figura 14 - Indução da árvore de decisão (parte 4)

```
# Função para definição do processo de divisão do conjunto de dados
def split(self, dataset, feature_index, threshold):

  dataset_left = np.array([row for row in dataset if row[feature_index]<=threshold])
  dataset_right = np.array([row for row in dataset if row[feature_index]>threshold])

  return dataset_left, dataset_right

# Função para cálculo da entropia
def entropy(self, y):

  class_labels = np.unique(y)
  entropy = 0
  for cls in class_labels:
      p_cls = len(y[y == cls]) / len(y)
      entropy += -p_cls * np.log2(p_cls)

  return entropy

# Função para cálculo da razão de ganho
def information_gain(self, parent, l_child, r_child, mode="entropy"):
  ''' function to compute information gain '''

  weight_l = len(l_child) / len(parent)
  weight_r = len(r_child) / len(parent)

  gain = self.entropy(parent) - (weight_l*self.entropy(l_child) + weight_r*self.entropy(r_child))
  gain_ratio = gain/(-weight_l*np.log2(weight_l) - weight_r*np.log2(weight_r))

  return gain_ratio

# Função para computação do valor da folha
def calculate_leaf_value(self, Y):

  Y = list(Y)
  return max(Y, key=Y.count)
```

Fonte: elaboração própria (2023)

A Figura 15 apresenta as funções utilizadas para o treinamento da árvore de decisão, ou seja, a indução da mesma a partir das funções mencionadas anteriormente (função *fit*), e para o cálculo da classe predita. Nesse sentido, as funções *predict* e *make_prediction* são voltadas para a definição de classes para um determinado conjunto de acordo com a árvore de decisão induzida.

Figura 15 - Indução da árvore de decisão (parte 5)

```
# Função para treinamento da árvore de decisão
def fit(self, X, Y):

  dataset = np.concatenate((X, Y), axis=1)
  self.root = self.build_tree(dataset)

# Função para predição da classificação com base na árvore de decisão induzida
def predict(self, X):

  preditions = [self.make_prediction(x, self.root) for x in X]
  return preditions

# Função para precição da classificação de uma amostra
def make_prediction(self, x, tree):

  if tree.value != None:
    return tree.value
  feature_val = x[tree.feature_index]
  if feature_val <= tree.threshold:
    return self.make_prediction(x, tree.left)
  else:
    return self.make_prediction(x, tree.right)
```

Fonte: elaboração própria (2023)

É apresentada, na Figura 16, a função utilizada para a visualização da árvore de decisão induzida. Diante disso, a partir do resultado do treinamento da árvore de decisão, a função *print_tree* retorna, visualmente, a estrutura da árvore de decisão como um todo, incluindo os nós e critérios de decisão.

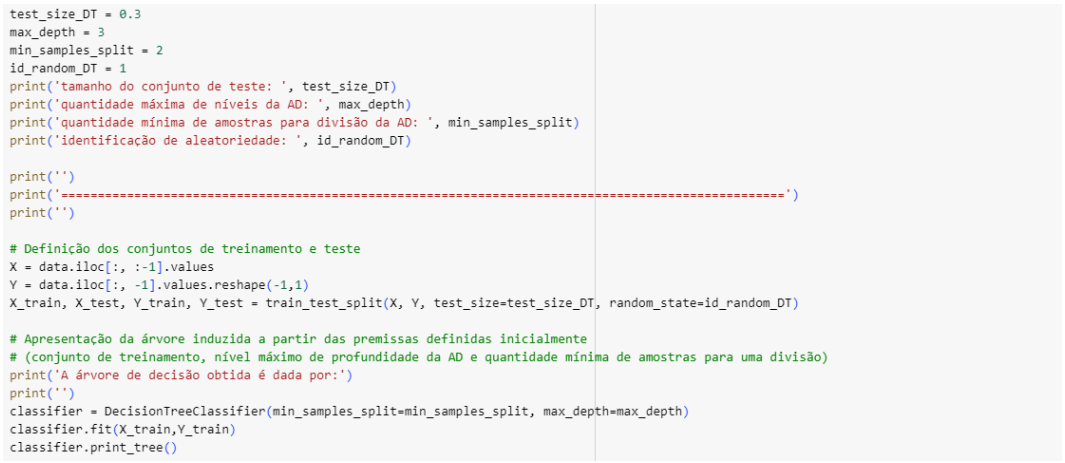
Figura 16 - Indução da árvore de decisão (parte 6)

```
# Função para apresentação da árvore de decisão induzida
def print_tree(self, tree=None, indent=" "):

  if not tree:
      tree = self.root

  if tree.value is not None:
      print(tree.value)

  else:
    # Apresentação caso a decisão seja referente ao Ganho Esperado
    if tree.feature_index == 0:
      print(column0, "<=", tree.threshold, "?", tree.info_gain)
      print("%sSim:" % (indent), end="")
      self.print_tree(tree.left, indent + indent)
      print("%sNão:" % (indent), end="")
      self.print_tree(tree.right, indent + indent)

    # Apresentação caso a decisão seja referente à Despesa Esperada
    elif tree.feature_index == 1:
      print(column1, "<=", tree.threshold, "?", tree.info_gain)
      print("%sSim:" % (indent), end="")
      self.print_tree(tree.left, indent + indent)
      print("%sNão:" % (indent), end="")
      self.print_tree(tree.right, indent + indent)

    # Apresentação caso a decisão seja referente à Polaridade do Risco
    elif tree.feature_index == 2:
      print(column2, "<=", tree.threshold, "?", tree.info_gain)
      print("%sSim:" % (indent), end="")
      self.print_tree(tree.left, indent + indent)
      print("%sNão:" % (indent), end="")
      self.print_tree(tree.right, indent + indent)
```

Fonte: elaboração própria (2023)

Para que possa ser finalizada a indução da árvore de decisão, é apresentado, na Figura 17, o código utilizado para a aplicação prática de todas as funções mencionadas anteriormente. Com isso, são definidos os parâmetros iniciais para a indução da árvore de decisão (no caso do presente trabalho: proporção do conjunto de teste = 30%, nível máximo da árvore de decisão = 3, quantidade mínima de amostras para uma nova divisão = 2 e “ID de aleatoriedade” = 1). A partir disso e da definição dos conjuntos de treinamento e teste pela mesma função já citada anteriormente (*test_train_split*), é possível treinar o modelo de classificação e visualizar o resultado da indução.

Figura 17 - Indução da árvore de decisão (parte 7)

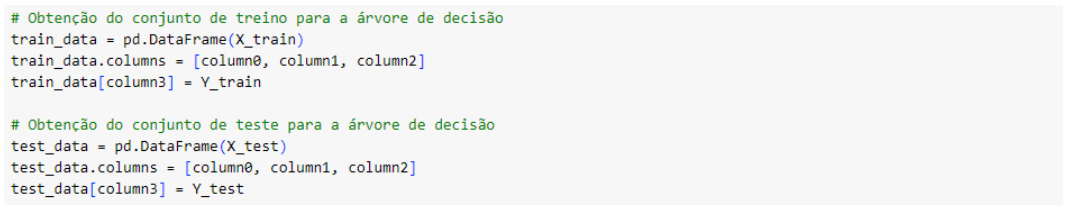


```
test_size_DT = 0.3
max_depth = 3
min_samples_split = 2
id_random_DT = 1
print('tamanho do conjunto de teste: ', test_size_DT)
print('quantidade máxima de níveis da AD: ', max_depth)
print('quantidade mínima de amostras para divisão da AD: ', min_samples_split)
print('identificação de aleatoriedade: ', id_random_DT)

print('')
print('============================================================================================')
print('')

# Definição dos conjuntos de treinamento e teste
X = data.iloc[:, :-1].values
Y = data.iloc[:, -1].values.reshape(-1,1)
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=test_size_DT, random_state=id_random_DT)

# Apresentação da árvore induzida a partir das premissas definidas inicialmente
# (conjunto de treinamento, nível máximo de profundidade da AD e quantidade mínima de amostras para uma divisão)
print('A árvore de decisão obtida é dada por:')
print('')
classifier = DecisionTreeClassifier(min_samples_split=min_samples_split, max_depth=max_depth)
classifier.fit(X_train,Y_train)
classifier.print_tree()
```

Fonte: elaboração própria (2023)

Dessa forma, como resultado dessa fase do código, tem-se a demonstração da forma da árvore de decisão induzida, além do detalhamento dos conjuntos de treinamento e teste considerados para a indução. O código utilizado para o detalhamento dos conjuntos de treinamento e teste é apresentado na Figura 18.

Figura 18 - Obtenção dos conjuntos de treinamento e teste - árvore de decisão



```
# Obtenção do conjunto de treino para a árvore de decisão
train_data = pd.DataFrame(X_train)
train_data.columns = [column0, column1, column2]
train_data[column3] = Y_train

# Obtenção do conjunto de teste para a árvore de decisão
test_data = pd.DataFrame(X_test)
test_data.columns = [column0, column1, column2]
test_data[column3] = Y_test
```

Fonte: elaboração própria (2023)

A partir disso, torna-se possível a avaliação do desempenho da árvore induzida. Para tal, foi elaborado o código mostrado na Figura 19, em que são considerados o conjunto de teste definido e o conjunto real disponibilizado inicialmente pelo usuário e, a partir deles, é calculada a acurácia do modelo construído a partir da matriz de confusão.

Figura 19 - Avaliação do modelo utilizado para árvore de decisão

```
# Obtenção da acurácia do modelo para a árvore de decisão a partir da matriz de confusão
correct_values = 0
total_values = 0

confusion_matrix = confusion_matrix(Y_test, Y_predict)
for i in range(0, len(train_data['Classificação'].unique())):
  for j in range(0, len(train_data['Classificação'].unique())):
    if i == j and confusion_matrix[i,j] > 0:
      correct_values = correct_values + confusion_matrix[i,j]
      total_values = total_values + confusion_matrix[i,j]
    elif i != j and confusion_matrix[i,j] > 0:
      total_values = total_values + confusion_matrix[i,j]

accuracy = (correct_values/total_values)*100
print('acuracia do modelo: ', round(accuracy,1), '%')
```

Fonte: elaboração própria (2023)

## 4.2.3. EXECUÇÃO DA ÁRVORE DE DECISÃO

A terceira etapa do modelo computacional tem como objetivo permitir a execução da árvore induzida com os dados de projetos que se deseja avaliar e, com isso, apresentar, como resultado, os projetos classificados. Diante disso, foi elaborado o código que pode ser observado na Figura 20, que define, o conjunto de atributos que devem ser considerados para a classificação, executa o modelo treinado com os dados desejados e apresenta como resultado uma tabela com o mesmo formato da Tabela 15, apresentada anteriormente.

Figura 20 - Obtenção dos resultados da árvore de decisão

```
# Execução da árvore de decisão a partir dos projetos que se deseja avaliar e obtenção de projetos classificados
# Usuário deve inserir o conjunto de dados com os projetos que se deseja avaliar
aux = data_user
aux = aux.drop('Projeto', axis=1)

# Conjunto de atributos
X_AD = aux.iloc[:, :].values
# Execução da árvore de decisão a partir do modelo treinado (conjunto de classes)
Y_AD = classifier.predict(X_AD)
# Apresentação dos dados classificados após a execução do modelo
classified_projects = data_user
classified_projects['Classificação'] = Y_AD
classified_projects
```

Fonte: elaboração própria (2023)

# 5. APLICAÇÃO PRÁTICA DO MODELO

O objetivo desta seção é apresentar uma aplicação prática do modelo, a partir de um conjunto de dados de 40 projetos já rotulados, com o intuito de realizar a análise de sentimento para obtenção das respectivas polaridades dos riscos e, como objetivo final, a indução de uma árvore de decisão para classificação dos projetos em questão. Busca-se, também, discutir e analisar os resultados obtidos.

A aplicação prática do modelo é dada a partir dos passos indicados na seção 4.2. do presente trabalho. Dessa forma, serão seguidas as seguintes etapas que serão detalhadas a seguir: definição dos conjuntos de treinamento, execução da análise de sentimento e indução da árvore de decisão.

## 5.1. DEFINIÇÃO DOS CONJUNTOS DE TREINAMENTO

Tendo em vista o objetivo final desta aplicação, relacionado à indução de uma árvore de decisão que possa ser utilizada para a classificação de projetos em um determinado contexto, devem ser definidos dois conjuntos de treinamento: I) para a execução da análise de sentimento supervisionada; e II) para a indução da árvore de decisão.

O conjunto de treinamento da análise de sentimento supervisionada deve estar vinculado aos objetivos estratégicos da organização, como já mencionado anteriormente. Dessa forma, para a aplicação em questão, foram adotadas as seguintes alavancas estratégicas: aumento de receita, aumento de eficiência, aumento de satisfação do cliente e garantia de segurança da operação.

Com isso, os riscos (ou oportunidades) de projetos vinculados às alavancas citadas com as respectivas polaridades esperadas (variando, nesse caso, de -1 a 1) podem ser observadas na Tabela 16. Vale ressaltar que foram definidas algumas oportunidades que estão vinculadas ao ambiente externo e que, portanto, não estão relacionadas diretamente às alavancas estratégicas, mas são consideradas relevantes, como: adaptação a leis e sistemas regulatórios e adaptação a sistemas de terceiros.

Tabela 16 - Conjunto de treinamento para análise de sentimento

| Alavanca Estratégica | Risco/Oportunidade | Polaridade |
|---|---|---|
| Receita | Small increase in revenue | 0,25 |
| Receita | Small revenue increase | 0,25 |
| Receita | Small growth in revenue | 0,25 |
| Receita | Small revenue growth | 0,25 |
| Receita | Increase in revenue | 0,5 |
| Receita | Revenue increase | 0,5 |
| Receita | Growth in revenue | 0,5 |
| Receita | Revenue growth | 0,5 |
| Receita | Large increase in revenue | 1,0 |
| Receita | Large revenue increase | 1,0 |
| Receita | Large growth in revenue | 1,0 |
| Receita | Large revenue growth | 1,0 |
| Receita | Small reduction in revenue | -0,25 |
| Receita | Small revenue reduction | -0,25 |
| Receita | Small decrease in revenue | -0,25 |
| Receita | Small revenue decrease | -0,25 |
| Receita | Reduction in revenue | -0,5 |
| Receita | Reduction decrease | -0,5 |
| Receita | Decrease in revenue | -0,5 |
| Receita | Revenue decrease | -0,5 |
| Receita | Large reduction in revenue | -1,0 |
| Receita | Large revenue reduction | -1,0 |
| Receita | Large decrease in revenue | -1,0 |
| Receita | Large revenue decrease | -1,0 |
| Eficiência | Small increase in efficiency | 0,25 |
| Eficiência | Small efficiency increase | 0,25 |
| Eficiência | Increase in efficiency | 0,5 |
| Eficiência | Efficiency increase | 0,5 |
| Eficiência | More efficiency | 0,5 |
| Eficiência | More efficient | 0,5 |
| Eficiência | Large increase in efficiency | 1,0 |
| Eficiência | Large efficiency increase | 1,0 |
| Eficiência | Small reduction in efficiency | -0,25 |
| Eficiência | Small efficiency reduction | -0,25 |
| Eficiência | Reduction in efficiency | -0,5 |
| Eficiência | Efficiency reduction | -0,5 |

| | | |
|---|---|---|
| Eficiência | Less efficiency | -0,5 |
| Eficiência | Less efficient | -0,5 |
| Eficiência | Large reduction in efficiency | -1,0 |
| Eficiência | Large efficiency reduction | -1,0 |
| Satisfação do cliente | Small increase in customer satisfaction | 0,25 |
| Satisfação do cliente | Small increase of customer satisfaction | 0,25 |
| Satisfação do cliente | Small growth in customer satisfaction | 0,25 |
| Satisfação do cliente | Small growth of customer satisfaction | 0,25 |
| Satisfação do cliente | Increase in customer satisfaction | 0,5 |
| Satisfação do cliente | Increase of customer satisfaction | 0,5 |
| Satisfação do cliente | Growth in customer satisfaction | 0,5 |
| Satisfação do cliente | Growth of customer satisfaction | 0,5 |
| Satisfação do cliente | More satisfied customers | 1,0 |
| Satisfação do cliente | More customer satisfaction | 1,0 |
| Satisfação do cliente | Large increase in customer satisfaction | 1,0 |
| Satisfação do cliente | Large increase of customer satisfaction | 1,0 |
| Satisfação do cliente | Large growth in customer satisfaction | 1,0 |
| Satisfação do cliente | Large growth of customer satisfaction | 1,0 |
| Satisfação do cliente | Small reduction in customer satisfaction | -0,25 |
| Satisfação do cliente | Small reduction of customer satisfaction | -0,25 |
| Satisfação do cliente | Small decrease in customer satisfaction | -0,25 |
| Satisfação do cliente | Small decrease of customer satisfaction | -0,25 |
| Satisfação do cliente | Reduction in customer satisfaction | -0,5 |
| Satisfação do cliente | Reduction of customer satisfaction | -0,5 |
| Satisfação do cliente | Decrease in customer satisfaction | -0,5 |
| Satisfação do cliente | Decrease of customer satisfaction | -0,5 |
| Satisfação do cliente | Less satisfied customers | -1,0 |
| Satisfação do cliente | Less customer satisfaction | -1,0 |
| Satisfação do cliente | Large reduction in customer satisfaction | -1,0 |
| Satisfação do cliente | Large reduction of customer satisfaction | -1,0 |
| Satisfação do cliente | Large decrease in customer satisfaction | -1,0 |
| Satisfação do cliente | Large decrease of customer satisfaction | -1,0 |
| Segurança | Small security increase | 0,25 |
| Segurança | Small increase in security | 0,25 |
| Segurança | Small security evolution | 0,25 |
| Segurança | Small evolution in security | 0,25 |
| Segurança | Security increase | 0,5 |
| Segurança | Increase in security | 0,5 |
| Segurança | Security evolution | 0,5 |

| | | |
|---|---|---|
| Segurança | Evolution in security | 0,5 |
| Segurança | More security | 1,0 |
| Segurança | Safer | 1,0 |
| Segurança | Large security increase | 1,0 |
| Segurança | Large increase in security | 1,0 |
| Segurança | Large security evolution | 1,0 |
| Segurança | Large evolution in security | 1,0 |
| Segurança | Small security reduction | -0,25 |
| Segurança | Small reduction in security | -0,25 |
| Segurança | Small security decrease | -0,25 |
| Segurança | Small decrease in security | -0,25 |
| Segurança | Security reduction | -0,5 |
| Segurança | reduction in security | -0,5 |
| Segurança | Security decrease | -0,5 |
| Segurança | decrease in security | -0,5 |
| Segurança | Less security | -1,0 |
| Segurança | Unsafer | -1,0 |
| Segurança | Large security reduction | -1,0 |
| Segurança | Large reduction in security | -1,0 |
| Segurança | Large security decrease | -1,0 |
| Segurança | Large decrease in security | -1,0 |
| Outros | No risks | 1,0 |
| Outros | No verified risks | 1,0 |
| Outros | No risks observed | 1,0 |
| Outros | No risks identified | 1,0 |
| Outros | Low risk | 1,0 |
| Outros | Low risk identified | 1,0 |
| Outros | Low risks observed | 1,0 |
| Outros | Low risks identified | 1,0 |
| Outros | Hard project | 0,0 |
| Outros | Complex project | 0,0 |
| Outros | Great effort allocated | 0,0 |
| Outros | Effort allocated | 0,0 |
| Outros | Much effort allocated | 0,0 |
| Outros | Adaptation to the law | 1,0 |
| Outros | Requirement of the law | 1,0 |
| Outros | Mandatory by law | 1,0 |
| Outros | Adaptation to regulatory systems | 1,0 |
| Outros | Requirement of regulatory systems | 1,0 |

| | | |
|---|---|---|
| Outros | Adaptation to other organizations | 0,5 |
| Outros | Requirement from other organizations | 0,5 |

Fonte: elaboração própria (2023)

O conjunto de treinamento referente à indução da árvore de decisão é composto por uma lista de projetos com os respectivos ganho esperado, despesa esperada, riscos (ou oportunidades) da execução do projeto e classificação esperada. Nesse sentido, para que haja a quantificação dos riscos, esse conjunto deve passar pela análise de sentimento antes que seja utilizado para a indução da árvore de decisão. Dessa forma, o conjunto de treinamento em questão é composto por informações referentes a 40 projetos e pode ser visualizado na Tabela 17. Vale ressaltar que, para essa aplicação, os valores relacionados ao ganho esperado e à despesa esperada dos projetos estão apresentados em "R$ Milhões" e os projetos são classificados de acordo com a sua importância de execução, sendo as possibilidades: prioridade baixa, prioridade média e prioridade alta.

Tabela 17 - Conjunto de treinamento para indução da árvore de decisão

| **Projeto** | **Ganho Esperado** | **Despesa Esperada** | **Riscos/Oportunidades Execução do Projeto** | **Classificação** |
|---|---|---|---|---|
| Projeto 1 | 2,0 | 1,0 | No risks identified | Prioridade alta |
| Projeto 2 | 10,0 | 7,0 | Increase in process efficiency | Prioridade média |
| Projeto 3 | 1,0 | 0,0 | Small reduction in process efficiency | Prioridade média |
| Projeto 4 | 2,5 | 1,5 | Small reduction in revenue over the next cycle | Prioridade média |
| Projeto 5 | 3,0 | 2,0 | Adaptation to other organizations | Prioridade alta |
| Projeto 6 | 2,5 | 1,5 | Efficiency reduction in processes carried out manually | Prioridade baixa |
| Projeto 7 | 0,5 | 0,2 | Less satisfied customers | Prioridade baixa |
| Projeto 8 | 1,5 | 0,0 | Security reduction due to vulnerability for data protection | Prioridade baixa |
| Projeto 9 | 8,0 | 5,0 | Increase in process efficiency | Prioridade média |
| Projeto 10 | 0,7 | 0,0 | No risks identified | Prioridade alta |
| Projeto 11 | 2,5 | 2,0 | Increase in customer satisfaction | Prioridade média |
| Projeto 12 | 3,5 | 2,5 | Small increase in process efficiency | Prioridade média |

| | | | | |
|---|---|---|---|---|
| Projeto 13 | 0,5 | 0,0 | Less satisfied customers | Prioridade baixa |
| Projeto 14 | 5,0 | 3,5 | Small increase in process efficiency | Prioridade média |
| Projeto 15 | 0,5 | 0,0 | No risks identified | Prioridade alta |
| Projeto 16 | 4,0 | 0,0 | Revenue reduction | Prioridade baixa |
| Projeto 17 | 1,5 | 1,0 | Large reduction in security with more vulnerability to external threats | Prioridade baixa |
| Projeto 18 | 15,0 | 10,0 | Increase in process efficiency | Prioridade média |
| Projeto 19 | 10,0 | 2,0 | Large increase in revenue due to high profits | Prioridade alta |
| Projeto 20 | 10,0 | 0,0 | Large revenue reduction | Prioridade baixa |
| Projeto 21 | 5,0 | 0,5 | Increase in revenue | Prioridade alta |
| Projeto 22 | 2,0 | 2,0 | Large reduction in efficiency due to fraud | Prioridade baixa |
| Projeto 23 | 12,0 | 6,0 | Large increase in revenue | Prioridade média |
| Projeto 24 | 7,5 | 5,0 | Increase in revenue | Prioridade média |
| Projeto 25 | 5,0 | 2,0 | New law requirement | Prioridade alta |
| Projeto 26 | 1,5 | 0,8 | Efficiency reduction in processes with increased execution time | Prioridade baixa |
| Projeto 27 | 6,0 | 3,0 | More efficient company due to more reliable data | Prioridade alta |
| Projeto 28 | 1,0 | 0,8 | Reduction in process efficiency | Prioridade baixa |
| Projeto 29 | 5,0 | 1,0 | Increase in revenue | Prioridade alta |
| Projeto 30 | 0,5 | 0,0 | Large efficiency reduction due to legal proceedings | Prioridade baixa |
| Projeto 31 | 1,5 | 1,0 | Large efficiency reduction due to legal proceedings | Prioridade baixa |
| Projeto 32 | 4,0 | 1,0 | Increase in revenue due to new customers | Prioridade alta |
| Projeto 33 | 5,0 | 3,0 | Reduction in efficiency of the company with the end of discounts | Prioridade baixa |
| Projeto 34 | 1,0 | 0,0 | Small reduction in revenue | Prioridade média |
| Projeto 35 | 2,0 | 1,0 | Reduction in customer satisfaction | Prioridade baixa |

| | | | | |
|---|---|---|---|---|
| Projeto 36 | 5,0 | 0,0 | Large revenue reduction | Prioridade baixa |
| Projeto 37 | 5,0 | 2,0 | Increase in customer satisfaction | Prioridade alta |
| Projeto 38 | 2,0 | 0,0 | Revenue decrease | Prioridade baixa |
| Projeto 39 | 0,3 | 0,0 | Less satisfied customers | Prioridade baixa |
| Projeto 40 | 1,5 | 0,5 | Security reduction due to vulnerability for data protection | Prioridade baixa |

Fonte: elaboração própria (2023)

## 5.2. EXECUÇÃO DA ANÁLISE DE SENTIMENTO

A partir do conjunto de treinamento proposto para a análise de sentimento, mostrado anteriormente na Tabela 16, foi realizado treinamento do modelo utilizado para a análise de sentimento utilizando-se o código apresentado na seção 4.2.1. do presente trabalho. Assim, considerando a proporção entre o conjunto de teste e o conjunto inicial equivalente a 30%, o conjunto de teste foi composto por um total de 35 elementos a serem testados.

Nesse sentido, foi obtido o resultado mostrado na Tabela 18, que compara a classificação predita e a classificação verdadeira para o conjunto de teste em questão.

Tabela 18 - Comparação entre polaridade verdadeira e polaridade predita - análise de sentimento

| **Alavanca Estratégica** | **Risco/Oportunidade** | **Polaridade Verdadeira** | **Polaridade Predita** |
|---|---|---|---|
| Segurança | Large decrease in security | -1,0 | -1,0 |
| Satisfação do cliente | Increase in customer satisfaction | 0,5 | 0,5 |
| Satisfação do cliente | Small decrease in customer satisfaction | -0,25 | -0,25 |
| Outros | No verified risks | 1,0 | 1,0 |
| Segurança | Small increase in security | 0,25 | 0,25 |
| Outros | Adaptation to other organizations | 0,5 | 0,5 |
| Segurança | Increase in security | 0,5 | 0,5 |
| Satisfação do cliente | Reduction in customer satisfaction | -0,5 | -0,5 |
| Eficiência | Efficiency reduction | -0,5 | -0,5 |
| Eficiência | Large reduction in efficiency | -1,0 | -1,0 |
| Satisfação do cliente | Large reduction of customer satisfaction | -1,0 | -1,0 |
| Outros | No risks | 1,0 | 1,0 |
| Receita | Small growth in revenue | 0,25 | 0,25 |

| | | | |
|---|---|---|---|
| Segurança | Large security increase | 1,0 | 1,0 |
| Outros | Adaptation to the law | 1,0 | 1,0 |
| Satisfação do cliente | Large growth of customer satisfaction | 1,0 | 1,0 |
| Outros | Low risks observed | 1,0 | 1,0 |
| Outros | No risks identified | 1,0 | 1,0 |
| Segurança | Large security decrease | -1,0 | -1,0 |
| Satisfação do cliente | Small growth in customer satisfaction | 0,25 | 0,25 |
| Outros | Hard project | 0,0 | 0,0 |
| Receita | Reduction decrease | -0,5 | -0,5 |
| Segurança | Large security evolution | 1,0 | 1,0 |
| Satisfação do cliente | Small reduction in customer satisfaction | -0,25 | -0,25 |
| Segurança | Large reduction in security | -1,0 | -1,0 |
| Eficiência | Small efficiency reduction | -0,25 | -0,25 |
| Satisfação do cliente | Large decrease of customer satisfaction | -1,0 | -1,0 |
| Satisfação do cliente | More satisfied customers | 1,0 | -1,0 |
| Satisfação do cliente | Growth in customer satisfaction | 0,5 | 0,5 |
| Segurança | Large security reduction | -1,0 | -1,0 |
| Eficiência | Large efficiency increase | 1,0 | 1,0 |
| Satisfação do cliente | Small reduction of customer satisfaction | -0,25 | -0,25 |
| Satisfação do cliente | Reduction of customer satisfaction | -0,5 | -0,5 |
| Receita | Large growth in revenue | 1,0 | 1,0 |
| Eficiência | Small reduction in efficiency | -0,25 | -0,25 |

Fonte: elaboração própria (2023)

Dessa forma, pode-se concluir que 34 dos 35 elementos testados tiveram a classificação predita pelo modelo igual à sua respectiva classificação verdadeira, fazendo com que a acurácia do modelo construído para a realização da análise de sentimento supervisionada seja de 97,1%, como mostrado abaixo em cálculo realizado com base na equação (8).

$$\text{Acurácia}_{\text{SA}} = \frac{34}{35} = 97{,}1\%$$

Com isso, foi possível executar a análise de sentimento para os riscos/oportunidades presentes no conjunto apresentado anteriormente na Tabela 17 a partir do modelo construído inicialmente. Assim, tem-se a quantificação desses riscos/oportunidades, de acordo com os resultados apresentados na tabela abaixo.

Tabela 19 - Resultado da análise de sentimento para a aplicação prática

| Projeto | Ganho Esperado | Despesa Esperada | Riscos/Oportunidades Execução do Projeto | Polaridade dos Riscos | Classificação |
|---|---|---|---|---|---|
| Projeto 1 | 2,0 | 1,0 | No risks identified | 1,0 | Prioridade alta |
| Projeto 2 | 10,0 | 7,0 | Increase in process efficiency | 0,5 | Prioridade média |
| Projeto 3 | 1,0 | 0,0 | Small reduction in process efficiency | -0,25 | Prioridade média |
| Projeto 4 | 2,5 | 1,5 | Small reduction in revenue over the next cycle | -0,25 | Prioridade média |
| Projeto 5 | 3,0 | 2,0 | Adaptation to other organizations | 0,5 | Prioridade alta |
| Projeto 6 | 2,5 | 1,5 | Efficiency reduction in processes carried out manually | -0,5 | Prioridade baixa |
| Projeto 7 | 0,5 | 0,2 | Less satisfied customers | -1,0 | Prioridade baixa |
| Projeto 8 | 1,5 | 0,0 | Security reduction due to vulnerability for data protection | -0,5 | Prioridade baixa |
| Projeto 9 | 8,0 | 5,0 | Increase in process efficiency | 0,5 | Prioridade média |
| Projeto 10 | 0,7 | 0,0 | No risks identified | 1,0 | Prioridade alta |
| Projeto 11 | 2,5 | 2,0 | Increase in customer satisfaction | 0,5 | Prioridade média |
| Projeto 12 | 3,5 | 2,5 | Small increase in process efficiency | 0,25 | Prioridade média |
| Projeto 13 | 0,5 | 0,0 | Less satisfied customers | -1,0 | Prioridade baixa |
| Projeto 14 | 5,0 | 3,5 | Small increase in process efficiency | 0,25 | Prioridade média |
| Projeto 15 | 0,5 | 0,0 | No risks identified | 1,0 | Prioridade alta |
| Projeto 16 | 4,0 | 0,0 | Revenue reduction | -0,5 | Prioridade baixa |
| Projeto 17 | 1,5 | 1,0 | Large reduction in security with more vulnerability to external threats | -1,0 | Prioridade baixa |
| Projeto 18 | 15,0 | 10,0 | Increase in process efficiency | 0,5 | Prioridade média |
| Projeto 19 | 10,0 | 2,0 | Large increase in revenue due to high profits | 1,0 | Prioridade alta |

| | | | | | |
|---|---|---|---|---|---|
| Projeto 20 | 10,0 | 0,0 | Large revenue reduction | -1,0 | Prioridade baixa |
| Projeto 21 | 5,0 | 0,5 | Increase in revenue | 0,5 | Prioridade alta |
| Projeto 22 | 2,0 | 2,0 | Large reduction in efficiency due to fraud | -1,0 | Prioridade baixa |
| Projeto 23 | 12,0 | 6,0 | Large increase in revenue | 1,0 | Prioridade média |
| Projeto 24 | 7,5 | 5,0 | Increase in revenue | 0,5 | Prioridade média |
| Projeto 25 | 5,0 | 2,0 | New law requirement | 1,0 | Prioridade alta |
| Projeto 26 | 1,5 | 0,8 | Efficiency reduction in processes with increased execution time | -0,5 | Prioridade baixa |
| Projeto 27 | 6,0 | 3,0 | More efficient company due to more reliable data | 0,5 | Prioridade alta |
| Projeto 28 | 1,0 | 0,8 | Reduction in process efficiency | -0,5 | Prioridade baixa |
| Projeto 29 | 5,0 | 1,0 | Increase in revenue | 0,5 | Prioridade alta |
| Projeto 30 | 0,5 | 0,0 | Large efficiency reduction due to legal proceedings | -1,0 | Prioridade baixa |
| Projeto 31 | 1,5 | 1,0 | Large efficiency reduction due to legal proceedings | -1,0 | Prioridade baixa |
| Projeto 32 | 4,0 | 1,0 | Increase in revenue due to new customers | 0,5 | Prioridade alta |
| Projeto 33 | 5,0 | 3,0 | Reduction in efficiency of the company with the end of discounts | -0,5 | Prioridade baixa |
| Projeto 34 | 1,0 | 0,0 | Small reduction in revenue | -0,25 | Prioridade média |
| Projeto 35 | 2,0 | 1,0 | Reduction in customer satisfaction | -0,5 | Prioridade baixa |
| Projeto 36 | 5,0 | 0,0 | Large revenue reduction | -1,0 | Prioridade baixa |
| Projeto 37 | 5,0 | 2,0 | Increase in customer satisfaction | 0,5 | Prioridade alta |
| Projeto 38 | 2,0 | 0,0 | Revenue decrease | -0,5 | Prioridade baixa |
| Projeto 39 | 0,3 | 0,0 | Less satisfied customers | -1,0 | Prioridade baixa |
| Projeto 40 | 1,5 | 0,5 | Security reduction due to vulnerability for data protection | -0,5 | Prioridade baixa |

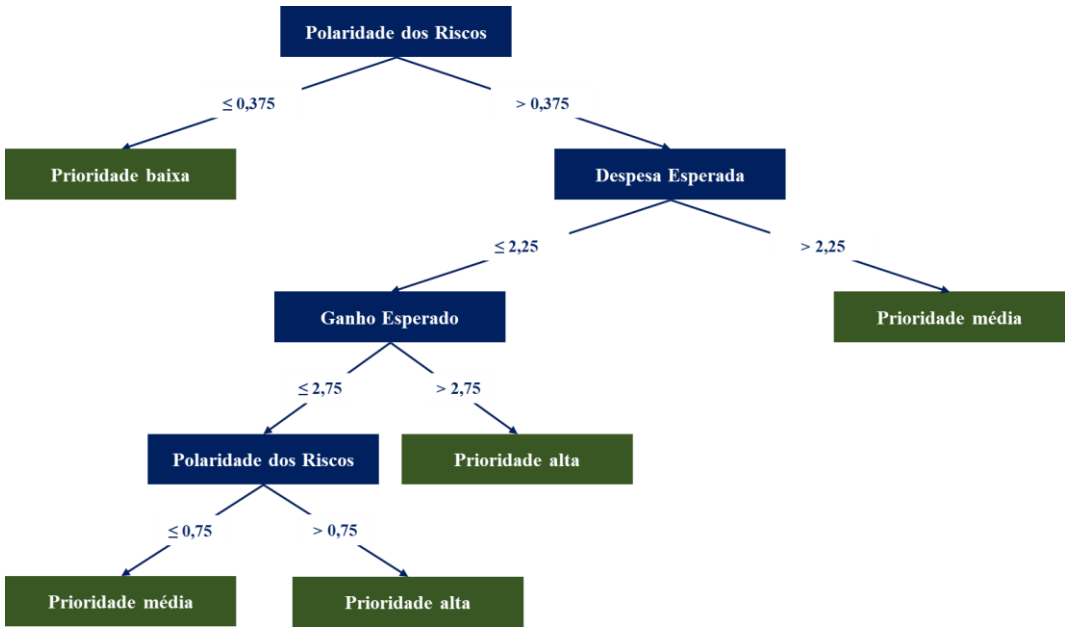
Fonte: elaboração própria (2023)

## 5.3. INDUÇÃO DA ÁRVORE DE DECISÃO

A indução da árvore de decisão para a aplicação prática em questão foi realizada a partir dos resultados obtidos para a análise de sentimento realizada anteriormente e da consequente quantificação dos riscos (ou oportunidades) vinculada aos projetos avaliados. Diante disso, o conjunto utilizado para essa indução mantém os valores exibidos na Tabela 19, com a exclusão da coluna em que os riscos dos projetos são colocados de forma qualitativa, fazendo com que sejam consideradas somente informações quantitativas.

Tendo esse conjunto em vista e considerando os parâmetros mencionados na seção 4.2.2. do presente relatório, foi obtida, a partir do código apresentado anteriormente, a árvore de decisão mostrada na Figura 21 que considera como critérios de decisão o ganho esperado, a despesa esperada e a polaridade dos riscos.

Figura 21 - Árvore de decisão induzida para a aplicação prática




Fonte: elaboração própria (2023)

Para que a árvore de decisão exposta na Figura 21 fosse induzida, o conjunto de dados inicial (40 projetos) foi dividido entre treinamento e teste, como mencionado anteriormente neste trabalho. Dessa forma, o conjunto de teste foi composto por 12 elementos (30% do conjunto inicial) e a comparação entre a classificação verdadeira desses elementos e a classificação predita pode ser observada na Tabela 20.

Tabela 20 - Comparação entre classificação verdadeira e classificação predita - árvore de decisão

| Ganho Esperado | Despesa Esperada | Polaridade dos Riscos | Classificação Verdadeira | Classificação Predita |
|---|---|---|---|---|
| 1.0 | 0,0 | -0,25 | Prioridade média | Prioridade média |
| 4.0 | 1,0 | 0,50 | Prioridade alta | Prioridade alta |
| 2.5 | 1,5 | -0,25 | Prioridade média | Prioridade média |
| 2.0 | 2,0 | -1,00 | Prioridade baixa | Prioridade baixa |
| 1.0 | 0,8 | -0,50 | Prioridade baixa | Prioridade baixa |
| 0.5 | 0,0 | -1,00 | Prioridade baixa | Prioridade baixa |
| 12.0 | 6,0 | 1,00 | Prioridade média | Prioridade média |
| 1.5 | 0,5 | -0,50 | Prioridade baixa | Prioridade baixa |
| 10.0 | 0,0 | -1,00 | Prioridade baixa | Prioridade baixa |
| 6.0 | 3,0 | 0,50 | Prioridade alta | Prioridade média |
| 5.0 | 3,0 | -0,50 | Prioridade baixa | Prioridade baixa |
| 15.0 | 10,0 | 0,50 | Prioridade média | Prioridade média |

Fonte: elaboração própria (2023)

Diante do conteúdo da tabela acima, pode-se observar que, dentre os 12 elementos avaliados, 11 tiveram a classificação predita pelo modelo equivalente à sua classificação verdadeira, fazendo com que a acurácia do modelo construído para a árvore de decisão em questão seja de 91,7%, assim como é mostrado abaixo.

$$\text{Acurácia}_{\text{DT}} = \frac{11}{12} = 91{,}7\%$$

Assim, a matriz de confusão para a árvore de decisão induzida, de acordo com o conjunto de teste apresentado anteriormente pode ser visualizada na tabela abaixo.

Tabela 21 - Matriz de confusão - árvore de decisão

| Classe | Prioridade baixa verdadeira | Prioridade média verdadeira | Prioridade alta verdadeira |
|---|---|---|---|
| **Prioridade baixa predita** | 6 | 0 | 0 |
| **Prioridade média predita** | 0 | 4 | 1 |
| **Prioridade alta predita** | 0 | 0 | 1 |

Fonte: elaboração própria (2023)

## 5.4. DISCUSSÃO DOS RESULTADOS

Diante da aplicação prática do modelo proposto e dos resultados obtidos, busca-se discutir aspectos vinculados a esses resultados, tendo em vista a interpretação dos mesmos em um contexto de uma organização.

Nesse sentido, observa-se que a árvore de decisão induzida no exemplo proposto, mostrada na Figura 21, apresenta características principais relacionadas aos atributos que servem como referência para a classificação dos projetos. Ou seja, verifica-se que o atributo referente aos riscos (ou oportunidades) dos projetos é o principal direcionador para a classificação dos mesmos, na medida em que é destacado como primeiro nó de decisão e que, para riscos com alto valor negativo, esses projetos são automaticamente classificados com "prioridade baixa".

Para além disso, assim como era esperado inicialmente para o contexto apresentado na aplicação prática em questão, os projetos classificados com "prioridade alta" são aqueles que requerem gastos baixos e apresentam ganhos relevantes ou oportunidades alinhadas com a estratégia da organização.

Vale ressaltar, portanto, que a definição de riscos (ou oportunidades) dos projetos atrelados aos objetivos organizacionais se mostra fundamental para os resultados do modelo proposto, tendo em vista a eficiência da organização. Isto é, considerando que a classificação dos projetos é diretamente influenciada pela polaridade dos riscos, decisões assertivas e alinhadas aos objetivos estratégicos devem ser obtidas a partir de polaridades classificadas "corretamente". Dessa forma, para que seja selecionado um portfólio de projetos adequado esses objetivos devem ser considerados no início da aplicação do modelo em questão.

# 6. CONCLUSÃO

O presente trabalho, conforme mencionado anteriormente, buscou propor um modelo de classificação de projetos a partir da combinação de duas técnicas: análise de sentimento e árvore de decisão. Para isso, considerou-se dados financeiros e riscos para os projetos avaliados.

O modelo em questão, portanto, é composto por duas fases principais: a análise de sentimento supervisionada e a indução da árvore de decisão. Na primeira fase, busca-se, transformar os riscos (ou oportunidades) vinculados aos projetos que se deseja avaliar, inicialmente qualitativos, em informações quantitativas (representadas pela polaridade). A partir disso, a indução da árvore de decisão torna-se possível considerando os dados financeiros dos projetos e as suas respectivas polaridades dos riscos. O modelo oferece, como resultado, os projetos classificados de acordo com a importância de execução.

Diante dos resultados expostos durante a seção anterior deste trabalho, destacam-se alguns pontos relacionados à aplicação prática do modelo elaborado: acurácia, interpretabilidade do modelo e adaptabilidade a novos dados.

No que diz respeito à acurácia do modelo, nota-se que, tanto para a análise de sentimento supervisionada quanto para a árvore de decisão, foi obtido um valor relevante, acima de 90%, indicando a efetividade do modelo como um todo. Em relação à análise de sentimento, pode-se afirmar que houve uma boa performance, ressaltando ainda mais a importância da definição de alavancas estratégicas com o intuito de guiar a determinação da polaridade dos riscos. Essa definição se mostrou fundamental para o desempenho do modelo da análise de sentimento supervisionada.

Já para a acurácia obtida com a indução da árvore de decisão, pode-se afirmar que ela reflete a efetividade da combinação entre os dois métodos propostos, uma vez que os dados passaram tanto pela análise de sentimento quanto pela árvore de decisão, indicando que essa interação entre eles é possível e pode gerar uma boa performance para a classificação de projetos, sem a ocorrência de *overfitting*. Destaca-se, ainda, a efetividade do método C4.5 para a indução de árvores de decisão.

Foi possível obter, também, um modelo cujos resultados são de fácil entendimento e a relação entre as variáveis (atributos) pode ser identificada de forma trivial. Para além disso, pode-se observar, ainda, a ausência de restrições para a adaptação do modelo a novos dados, de

forma que, para que uma nova árvore de decisão possa ser induzida, basta que sejam apresentados conjuntos de treinamentos adequados a um contexto diferente para a realização da análise de sentimento e a indução da árvore de decisão. Diante disso, cita-se como limitação do modelo justamente a questão do contexto, já que, em função da necessidade de ser adequado aos objetivos de uma organização, não há possibilidade de que seja generalizado, fazendo com que o modelo tenha performances efetivas em contextos específicos.

Em relação a oportunidades para estudos futuros, sugere-se: a) a comparação entre classificadores para a análise de sentimento e para a árvore de decisão; b) a utilização de novos critérios para a classificação de projetos (exemplos: ROI, TIR e probabilidade de ocorrência dos riscos); e c) a comparação entre métricas para avaliação de desempenho do modelo (exemplos: precisão, *recall* e *F1 score*).

Vale ressaltar que os objetivos específicos indicados na seção 1.2.2. foram alcançados a partir do desenvolvimento de um modelo capaz de integrar as técnicas de análise de sentimento para avaliação de riscos de projetos e árvore de decisão para classificação de projetos, de acordo com os conceitos presentes na literatura. Ademais, foi possível, também, avaliar o modelo através de uma aplicação piloto.

Por fim, destaca-se que o modelo em questão é um meio de tornar organizações mais eficientes, o que, no mercado atual, trata-se de uma característica fundamental. Essa eficiência é obtida em função da classificação direta de projetos que devem ter a execução considerada importante e da consequente alocação otimizada de recursos. Ou seja, um modelo de classificação de projetos permite que avaliações subjetivas em que os critérios podem não ser claros e adotados de forma inconstante sejam minimizadas. Dessa forma, as decisões sobre classificação de projetos podem ser tomadas de forma ágil considerando critérios objetivos e vinculados à estratégia da organização, como dados financeiros e riscos dos projetos.

## 7. REFERÊNCIAS


ARANTES. R. F. M. **Proposta de avaliação de fornecedores com obtenção de consenso automatizado, por meio do ANFIS, e comparação do uso das técnicas Random Forest e Decision Tree para segmentação de fornecedores**. 2023. 140. Tese - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2020.

ARCHER, Norm P.; GHASEMZADEH, Fereidoun. **An integrated framework for project portfolio selection**. International Journal of Project Management, v. 17, n. 4, p. 207-216, 1999.

BALAZS, Jorge A.; VELÁSQUEZ, Juan D. **Opinion mining and information fusion: a survey**. Information Fusion, v. 27, p. 95-110, 2016.

BERNARDINO, Flávia Ferreira Marques; PEIXOTO, Fernanda Maciel; FERREIRA, Roberto do Nascimento. **Governança e eficiência em empresas do setor elétrico brasileiro**. Revista Pretexto, v. 16, n. 1, p. 36-51, 2015.

BERTRAND, J. Will M.; FRANSOO, Jan C. **Modelling and simulation**. Researching operations management. Routledge, 2010.

BIRJALI, Marouane; KASRI, Mohammed; BENI-HSSANE, Abderrahim. **A comprehensive survey on sentiment analysis: Approaches, challenges and trends**. Knowledge-Based Systems, v. 226, p. 107134, 2021.

CAUCHICK MIGUEL, Paulo Augusto et al. **Metodologia de pesquisa em engenharia de produção e gestão de operações**. Rio de Janeiro: Elsevier, 2010.

CHEN, Kewen et al. **Turning from TF-IDF to TF-IGM for term weighting in text classification**. Expert Systems with Applications, v. 66, p. 245-260, 2016.

ELONEN, Suvi; ARTTO, Karlos A. **Problems in managing internal development projects in multi-project environments**. International journal of project management, v. 21, n. 6, p. 395-402, 2003.

FAWCETT, Tom. **An introduction to ROC analysis**. Pattern recognition letters, v. 27, n. 8, p. 861-874, 2006.

GOEBEL, Michael; GRUENWALD, Le. **A survey of data mining and knowledge discovery software tools**. ACM SIGKDD explorations newsletter, v. 1, n. 1, p. 20-33, 1999.

GOLOVCHENKO, O; SAIENSUS, M.; SOROKOUMOV, G.; ONOFRIICHUK, O; ZUBKO, O; LIU, L. **Management of efficiency and competitiveness of enterprises**. Economic Affairs, Vol. 67, Ed. 3, 2022.

GUTIÉRREZ-BATISTA, Karel; VILA, Maria-Amparo; MARTIN-BAUTISTA, Maria J. **Building a fuzzy sentiment dimension for multidimensional analysis in social networks**. Applied Soft Computing, v. 108, p. 107390, 2021.

HSU, Chih-Wei; CHANG, Chih-Chung; LIN, Chih-Jen. **A practical guide to support vector classification**. 2003.

KINGSFORD, Carl; SALZBERG, Steven L. **What are decision trees?**. Nature biotechnology, v. 26, n. 9, p. 1011-1013, 2008.

KOTRONOULAS, Grigorios et al. **An overview of the fundamentals of data management, analysis, and interpretation in quantitative research**. Seminars in oncology nursing. WB Saunders, 2023.
LOPES, Mariana Vieira Ribeiro. **Tratamento de imprecisão na geração de árvores de decisão**. 2016.

MARINO, L. **Gestão da qualidade e gestão do conhecimento: fatores-chave para produtividade e competitividade empresarial**. XXII SIMPEP – Simpósio de Engenharia de Produção, 2006.

MATOS, Pablo Freire et al. **Relatório técnico "métricas de avaliação"**. Universidade Federal de Sao Carlos, 2009.

MIRA, Cleber et al. **A project portfolio selection decision support system.** 10th International Conference on Service Systems and Service Management. IEEE, p. 725-730, 2013.

MONARD, Maria Carolina; BARANAUSKAS, José Augusto. **Conceitos sobre aprendizado de máquina**. Sistemas inteligentes-Fundamentos e aplicações, v. 1, n. 1, p. 39-56, 2003.

MORAES, Renato de Oliveira; LAURINDO, Fernando José Barbin. **Um estudo de caso de gestão de portfolio de projetos de tecnologia da informação**. Gestão & Produção, v. 10, p. 311-328, 2003.

PAIXÃO, Gabriela Miana de Mattos et al. **Machine Learning na Medicina: Revisão e Aplicabilidade**. Arquivos Brasileiros de Cardiologia, v. 118, p. 95-102, 2022.

PANDITA, Harsheta; GONDHI, Naveen Kumar. **A literature survey of sentiment analysis based on E-commerce reviews**. 5th International Conference on Computing Methodologies and Communication (ICCMC). IEEE, p. 1767-1772, 2021.

PAULA, Maurício Braga. **Indução automática de árvores de decisão**. 2002.

PMI. **Um guia do conhecimento em gerenciamento de projetos (guia PMBOK)**. 2008.
QUINLAN, J. Ross. **C4. 5: programs for machine learning**. Elsevier, 1993.

RAMOS, Jorge Luis Cavalcanti et al. **Um estudo comparativo de classificadores na previsão da evasão de alunos em EAD**. Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE), p.1463-1472, 2018.

SANTOS, Bruno Samways dos; STEINER, Maria Teresinha Arns; LIMA, Rafael Henrique Palma. **Proposal of a method to classify female smokers based on data mining techniques**. Computers & Industrial Engineering, v. 170, p. 1-18, 2022.

SAURIN, Valter; LOPES, Ana Lúcia Miranda; DA COSTA JÚNIOR, Newton C.A. **Eficiência e valor: uma abordagem com base na análise envoltória de dados (DEA) aplicada às empresas do setor elétrico no Brasil**. Revista de Economia e Administração, v. 9, n. 2, p. 170-190, 2010.

SCHNEIDER, Pedro Henrique. **Análise preditiva de Churn com ênfase em técnicas de Machine Learning: uma revisão**. 2016.

SHARMA, Himani; KUMAR, Sunil. **A survey on decision tree algorithms of classification in data mining**. International Journal of Science and Research (IJSR), v. 5, n. 4, p. 2094-2097, 2016.

SILVA NETO, Arlindino Nogueira. **Avaliação de projetos estratégicos de tecnologia da informação**. 2008.

STINE, Robert A. **Sentiment analysis**. Annual review of statistics and its application, v. 6, p. 287-308, 2019.

SUN, Shiliang; LUO, Chen; CHEN, Junyu. **A review of natural language processing techniques for opinion mining systems**. Information fusion, v. 36, p. 10-25, 2017.

TRSTENJAK, Bruno; MIKAC, Sasa; DONKO, Dzenana. **KNN with TF-IDF based framework for text categorization**. Procedia Engineering, v. 69, p. 1356-1364, 2014.

VARGAS, Ricardo Viana. **Gerenciamento de Projetos: estabelecendo diferenciais competitivos**. Brasport, 2005.

XU, Zhuoer et al. **One-Stage Tree: end-to-end tree builder and pruner**. Machine Learning, v. 111, n. 5, p. 1959-1985, 2022.