

Amadeu Chacar

**Avaliação de Modelos de Inteligência
Computacional para Classificação de Ativos de
Renda Variável do Mercado Financeiro
Brasileiro Utilizando Análise Fundamentalista**

Campos dos Goytacazes-RJ

Novembro de 2022

Amadeu Chacar

Avaliação de Modelos de Inteligência Computacional para Classificação de Ativos de Renda Variável do Mercado Financeiro Brasileiro Utilizando Análise Fundamentalista

Trabalho de Conclusão apresentado ao curso
Bacharelado em Engenharia de Computa-
ção do Instituto Federal de Educação, Ci-
ência e Tecnologia Fluminense, como parte
dos requisitos para a obtenção do título de
Bacharel em Engenharia de Computação.

Instituto Federal de Educação, Ciência e Tecnologia Fluminense

Orientador: Prof. D.Sc. Fábio Duncan de Souza

Coorientador: Prof. D.Sc. Fernando Luiz de Carvalho e Silva

Campos dos Goytacazes-RJ


Novembro de 2022

Amadeu Chacar


**Avaliação de Modelos de Inteligência Computacional para
Classificação de Ativos de Renda Variável do Mercado
Financeiro Brasileiro Utilizando Análise Fundamentalista**

Trabalho de Conclusão apresentado ao curso
Bacharelado em Engenharia de Computa-
ção do Instituto Federal de Educação, Ci-
ência e Tecnologia Fluminense, como parte
dos requisitos para a obtenção do título de
Bacharel em Engenharia de Computação.


Campos dos Goytacazes-RJ, 11 de Novembro de 2022.



Prof. D.Sc. Fábio Duncan de Souza
Instituto Federal Fluminense (IFF)



Prof. D.Sc. Fernando Luiz de Carvalho e Silva
Instituto Federal Fluminense (IFF)



Prof. M.Sc. Vinicius Barcelos da Silva
Instituto Federal Fluminense (IFF)

Campos dos Goytacazes-RJ
Novembro de 2022

AGRADECIMENTOS

Agradeço à minha família, que desde sempre me apoiou em todas as minhas decisões e me deu o suporte para que pudesse trilhar este caminho. Nada seria possível sem ela.

Aos amigos que fiz durante o curso, que com certeza ficarão para a vida. Fiz amizades incríveis que ajudaram a chegar até aqui.

Aos amigos de longa data, que também sempre me apoiaram em todos os momentos. São uma segunda família para mim.

À minha namorada, que esteve ao meu lado durante boa parte deste processo e que sempre me incentivou a seguir em frente.

Aos professores que tive nesta caminhada, em especial Fábio Duncan de Souza e Fernando Luiz de Carvalho e Silva, que me ajudaram desde o primeiro período e sempre me incentivaram a seguir meu próprio caminho. Desde projetos de monitoria e iniciação científica às orientações de trabalho final, estes se tornaram grandes amigos que levarei para sempre comigo.

RESUMO

Investir no mercado financeiro tem se tornado cada vez mais parte da cultura dos brasileiros, tendo em vista um grande aumento no número de interessados no assunto nos últimos anos. A bolsa de valores é o ambiente onde podem ser negociadas ações de empresas que possuem capital aberto, possibilitando que o investidor realize compra ou venda. Existem diversas estratégias que podem ser utilizadas a fim de obter lucro. Para tomar uma boa decisão de compra a longo prazo, é necessário que se tenha conhecimento sobre a empresa, seu desempenho e saúde financeira. Esse estudo acaba sendo dificultado devido a grande quantidade de ativos disponíveis na bolsa de valores, principalmente para investidores iniciantes. Alguns portais e especialistas fornecem dicas de possíveis boas empresas, porém, em sua grande maioria, são serviços pagos. O presente trabalho propõe uma metodologia que permite pré-selecionar ativos do mercado financeiro para que o investidor analise individualmente, a fim de reduzir seu esforço. Utilizando os balanços patrimoniais trimestrais e demonstrativos de resultados de todas as empresas listadas na bolsa de valores brasileira, foram testados diferentes modelos de inteligência computacional, a fim de entender como se comportam no contexto do trabalho, onde cada um foi preparado visando a escolha de ativos possivelmente lucrativos. Cada modelo foi treinado para gerar uma carteira fictícia de investimento, onde foram avaliados seu tamanho e rendimento no período de um ano. Dessa forma, conclui-se que é possível utilizar inteligência computacional para realizar análises fundamentalistas das empresas e selecionar ativos possivelmente lucrativos para que o investidor tome suas decisões.

Palavras-chaves: Inteligência computacional, mercado financeiro, análise fundamentalista, aprendizado de máquina.

ABSTRACT

Investing in the stock market has become part of the Brazilian culture, with a large increase in the number of with a large increase in the number of enthusiasts in the last few years. The stock market is the place where people can negotiate with publicly traded companies, which investors can buy or sell. There are several strategies that can be used for profit purposes. To make a good long-term purchase decision, it is necessary to have knowledge about the company, its performance and financial health. This analysis is made difficult due to the large number of stocks available, which makes the life of a novice investor more difficult. Some portals and experts provide tips on possible good companies, however, for the most part, they are paid services. The present work proposes a methodology that allows the pre-selection of financial market assets so that the investor analyze individually in order to reduce his effort. Using quarterly balance sheets of all companies listed on the Brazilian stock market, were tested different models of machine learning in order to understand how they behave in the context the project, where each one was prepared to choose possibly profitable assets. Through comparisons between the results, it was possible to identify which provided more profitable stock sets. In this way, it is concluded that it is possible use computational intelligence to perform fundamental analysis of companies and select potentially profitable assets for the investor to make his decisions.

Keywords: Machine learning, stock market, fundamental analysis.

LISTA DE FIGURAS

| | |
|--|----|
| Figura 1 – Fronteira eficiente de Markowitz | 7 |
| Figura 2 – Exemplo de gráfico de preços utilizando linhas de suporte e resistência | 8 |
| Figura 3 – Exemplo de gráfico de preços utilizando indicadores de tendência | 9 |
| Figura 4 – Etapas do processo de KDD | 11 |
| Figura 5 – Exemplo gráfico da correlação de Pearson | 13 |
| Figura 6 – Funcionamento do algoritmo <i>AdaBoost Classifier</i> | 15 |
| Figura 7 – Funcionamento do algoritmo <i>Decision Tree Classifier</i> | 16 |
| Figura 8 – Funcionamento do algoritmo <i>Random Forest</i> | 17 |
| Figura 9 – Funcionamento do algoritmo <i>KNN Classifier</i> | 20 |
| Figura 10 – Hiperplano ótimo e máxima margem de separação | 22 |
| Figura 11 – Exemplo de neurônio artificial | 23 |
| Figura 12 – Exemplo de camadas de uma rede neural | 24 |
| Figura 13 – Exemplo de camadas de uma rede neural | 25 |
| Figura 14 – Representação do aprendizado <i>backpropagation</i> | 26 |
| Figura 15 – Exemplo de utilização do Jupyter Notebook | 28 |
| Figura 16 – Exemplo de utilização da biblioteca Pandas | 29 |
| Figura 17 – Exemplo de utilização da biblioteca Seaborn | 31 |
| Figura 18 – Etapas de desenvolvimento do projeto | 34 |
| Figura 19 – Etapas de pré-processamento dos dados | 35 |
| Figura 20 – Etapas de aplicação dos modelos | 36 |
| Figura 21 – Estrutura do arquivo de balanços patrimoniais trimestrais e demonstra- tivos de resultados do ativo YDUQ3 | 39 |
| Figura 22 – Estrutura de cotações históricas do ativo ABEV3 | 39 |
| Figura 23 – Tratamento de colunas nos dados de balanços e DREs | 41 |
| Figura 24 – Ilustração da relativização dos dados existentes | 44 |
| Figura 25 – Exemplo de parte dos dados da empresa AMBEV | 46 |
| Figura 26 – Quantidade de cada rótulo realizada na classificação | 47 |
| Figura 27 – Análise de correlação de Pearson entre as colunas | 48 |
| Figura 28 – Colunas mais importantes na classificação | 50 |
| Figura 29 – Dados após o processo de <i>feature selection</i> | 50 |
| Figura 30 – Resultados de uma execução do modelo <i>AdaBoost Classifier</i> | 52 |
| Figura 31 – Resultados da biblioteca PyCaret | 55 |
| Figura 32 – Aplicação da busca randômica para o algoritmo <i>CatBoost</i> | 58 |
| Figura 33 – Aplicação da busca randômica para o algoritmo <i>Extra Trees</i> | 59 |
| Figura 34 – Aplicação da busca randômica para o algoritmo XGBoost | 60 |
| Figura 35 – Aplicação da busca randômica para o algoritmo <i>Random Forest</i> | 61 |

| | |
|--|----|
| Figura 36 – Aplicação da busca randômica para o algoritmo LightGBM | 62 |
|--|----|

LISTA DE TABELAS

| | | |
|-----------|---|----|
| Tabela 1 | – Resultados da primeira etapa de aplicação de modelos | 53 |
| Tabela 2 | – Resultados da segunda etapa de aplicação de modelos | 56 |
| Tabela 3 | – Resultados da busca randômica do modelo <i>CatBoost</i> | 58 |
| Tabela 4 | – Resultados da busca randômica do modelo <i>Extra Trees</i> | 59 |
| Tabela 5 | – Resultados da busca randômica do modelo XGBoost. | 60 |
| Tabela 6 | – Resultados da busca randômica do modelo <i>Random Forest</i> | 61 |
| Tabela 7 | – Resultados da busca randômica do modelo LightGBM. | 62 |
| Tabela 8 | – Resultados da simulação do modelo <i>CatBoost Classifier</i> | 64 |
| Tabela 9 | – Resultados da simulação do modelo <i>Extra Trees Classifier</i> | 65 |
| Tabela 10 | – Resultados da simulação do modelo XGBoost. | 65 |
| Tabela 11 | – Resultados da simulação do modelo <i>Random Forest</i> | 66 |
| Tabela 12 | – Resultados da simulação do modelo LightGBM. | 66 |

LISTA DE SIGLAS

| | |
|----------|--|
| B3 | Brasil, Bolsa, Balcão |
| DRE | Demonstrativo de resultado de exercício |
| DY | <i>Dividend Yield</i> |
| KNN | <i>K-Nearest Neighbors Classifier</i> |
| LDA | <i>Linear Discriminant Analysis</i> |
| LPA | Lucro por ação |
| LightGBM | <i>Light Gradient Boosting Machine</i> |
| NB | <i>Naïve Bayes Classifier</i> |
| P/L | Preço sobre lucro |
| P/VPA | Preço sobre valor patrimonial por ação |
| QDA | <i>Quadratic Discriminant Analysis</i> |
| RNA | Rede neural artificial |
| SVM | <i>Support Vector Machine</i> |
| XGB | <i>Extreme Gradient Boosting Machine</i> |

SUMÁRIO

| | | |
|------------|--|-----------|
| | Sumário | xi |
| 1 | INTRODUÇÃO | 1 |
| 1.1 | Justificativa | 3 |
| 1.2 | Objetivos | 3 |
| 1.2.1 | Objetivos Específicos | 4 |
| 1.3 | Estrutura do Trabalho | 4 |
| 2 | FUNDAMENTAÇÃO TEÓRICA | 5 |
| 2.1 | Introdução ao mercado financeiro | 5 |
| 2.1.1 | Características de uma ação | 5 |
| 2.1.2 | Técnicas para diminuição de risco em investimentos | 6 |
| 2.1.2.1 | Teoria Moderna do Portfólio | 6 |
| 2.1.3 | Métodos de análise financeira | 7 |
| 2.1.3.1 | Análise técnica | 8 |
| 2.1.3.2 | Análise fundamentalista | 9 |
| 2.2 | Tecnologias aplicadas | 10 |
| 2.2.1 | Mineração de dados | 10 |
| 2.2.2 | Correlação de Pearson | 12 |
| 2.2.3 | Inteligência computacional | 13 |
| 2.2.4 | Modelos de inteligência computacional | 14 |
| 2.2.4.1 | AdaBoost Classifier | 14 |
| 2.2.4.2 | Decision Tree Classifier | 15 |
| 2.2.4.3 | Random Forest Classifier | 16 |
| 2.2.4.4 | Extra Trees Classifier | 17 |
| 2.2.4.5 | Gradient Boosting Classifier | 18 |
| 2.2.4.6 | Light Gradient Boosting Machine | 18 |
| 2.2.4.7 | Extreme Gradient Boosting Machine | 18 |
| 2.2.4.8 | CatBoost Classifier | 19 |
| 2.2.4.9 | K-Nearest Neighbors Classifier | 19 |
| 2.2.4.10 | Logistic Regression | 20 |
| 2.2.4.11 | Linear Discriminant Analysis | 20 |

| | | |
|------------|--|-----------|
| 2.2.4.12 | Quadratic Discriminant Analysis | 21 |
| 2.2.4.13 | Naïve Bayes Classifier | 21 |
| 2.2.4.14 | Ridge Classifier | 21 |
| 2.2.4.15 | Support Vector Machine | 21 |
| 2.2.4.16 | Redes neurais artificiais | 22 |
| 2.2.4.16.1 | Arquitetura das redes neurais artificiais | 24 |
| 2.2.4.16.2 | Processo de aprendizagem das redes neurais artificiais | 25 |
| 2.2.4.16.3 | Redes neurais artificiais de múltiplas camadas | 26 |
| 2.3 | Ferramentas de desenvolvimento | 27 |
| 2.3.1 | Jupyter Notebook | 27 |
| 2.3.2 | Python | 28 |
| 2.3.3 | Bibliotecas utilizadas | 29 |
| 2.3.3.1 | Pandas | 29 |
| 2.3.3.2 | NumPy | 30 |
| 2.3.3.3 | Plotly | 30 |
| 2.3.3.4 | Matplotlib | 30 |
| 2.3.3.5 | Seaborn | 30 |
| 2.3.3.6 | Scikit-Learn | 31 |
| 2.3.3.7 | PyCaret | 32 |
| 3 | METODOLOGIA | 33 |
| 3.1 | Coleta de dados | 34 |
| 3.2 | Pré-processamento dos dados | 34 |
| 3.3 | Análise exploratória | 35 |
| 3.4 | Seleção de dados | 35 |
| 3.5 | Aplicação de modelos de inteligência computacional | 36 |
| 3.6 | Simulações com dados reais | 37 |
| 3.7 | Avaliação dos resultados | 37 |
| 4 | DESENVOLVIMENTO | 38 |
| 4.1 | Coleta de dados | 38 |
| 4.2 | Pré-processamento dos dados | 40 |
| 4.2.1 | Definição do intervalo de tempo | 40 |
| 4.2.2 | Limpeza dos dados | 40 |
| 4.2.3 | Inserção dos indicadores fundamentalistas | 42 |
| 4.2.4 | Indicação dos setores de cada empresa | 42 |
| 4.2.5 | Transformação dos dados | 43 |
| 4.2.6 | Criação dos rótulos de classificação | 44 |
| 4.3 | Análise exploratória | 46 |
| 4.4 | Seleção de dados | 49 |

| | | |
|------------|--|-----------|
| 4.5 | Aplicação dos modelos de inteligência computacional | 51 |
| 4.5.1 | Testes com versões genéricas de diversos modelos | 51 |
| 4.5.2 | Testes com <i>Machine Learning</i> Automatizado | 53 |
| 4.5.3 | Seleção dos melhores modelos | 56 |
| 4.5.4 | Otimização de hiperparâmetros dos melhores modelos | 57 |
| 4.5.4.1 | <i>CatBoost Classifier</i> | 58 |
| 4.5.4.2 | <i>Extra Trees Classifier</i> | 58 |
| 4.5.4.3 | <i>Extreme Gradient Boosting Machine</i> (XGBoost) | 59 |
| 4.5.4.4 | <i>Random Forest Classifier</i> | 60 |
| 4.5.4.5 | <i>Light Gradient Boosting Machine</i> (LightGBM) | 61 |
| 4.6 | Simulações com dados reais | 62 |
| 5 | RESULTADOS E DISCUSSÕES | 64 |
| 5.1 | <i>CatBoost Classifier</i> | 64 |
| 5.2 | <i>Extra Trees Classifier</i> | 64 |
| 5.3 | <i>Extreme Gradient Boosting Machine</i> (XGBoost) | 65 |
| 5.4 | <i>Random Forest Classifier</i> | 65 |
| 5.5 | <i>Light Gradient Boosting Machine</i> (LightGBM) | 66 |
| 5.6 | Discussões sobre os resultados | 66 |
| 6 | CONCLUSÃO | 68 |
| 6.1 | Trabalhos futuros | 69 |
| | REFERÊNCIAS | 70 |

1 INTRODUÇÃO

O interesse por parte da população brasileira em conhecer o mercado financeiro e realizar investimentos tem aumentado consideravelmente nos últimos anos. No ano de 2020, a bolsa de valores brasileira registrou cerca de 1,5 milhão de novos investidores, caracterizando um aumento de 92% com relação ao ano anterior (D'ÁVILA, 2021). A pandemia do novo coronavírus e o alto índice de desemprego do país, chegando a 14% em setembro de 2020 (SILVEIRA, 2020), contribuíram para que o cidadão brasileiro buscasse novas fontes de renda.

A bolsa de valores é o ambiente onde ocorrem as negociações de títulos emitidos por empresas de capital aberto, também chamados de ações, que podem ser comprados ou vendidos. No Brasil, o principal meio de negociação de ações é a B3, bolsa de valores oficial do país, fruto da fusão entre as antigas Bovespa, BM&F e Cetip. Nela, além de realizar compra e venda de ativos financeiros, também é possível gerenciá-los e acompanhar os índices de mercado.

Com a grande quantidade de conteúdo disponível gratuitamente na internet, especialmente em redes sociais como o *YouTube*, tornou-se mais simples o processo de busca por conhecimento de forma autônoma. A maior parte dos investidores iniciantes conhecem este universo e aprendem a operar no mercado de ações com influenciadores digitais. Uma pesquisa realizada pela B3 no ano de 2020 mostra que a maioria dos recém-chegados ao mercado, além de estudarem por conta própria, possuem foco nos investimentos a longo prazo (LARGHI, 2020).

Analisar o ativo em que se pretende investir é uma das principais tarefas a se realizar quando se trata de atuar no mercado financeiro. A fim de diminuir riscos e elevar a lucratividade, é necessário que o investidor, iniciante ou não, estude sobre a empresa que deseja comprar ações. A análise fundamentalista é uma etapa importante deste processo. Baseada em dados numéricos, este estudo tem como objetivo determinar se a empresa em questão possui uma boa saúde financeira, entendendo se e como ela realiza o pagamento de dividendos, se o preço dela está caro ou barato, dentre outros aspectos do ativo.

Os indicadores fundamentalistas fornecem informações de valor sobre a empresa em questão. REIS (2017) sugere que alguns dos indicadores mais importantes a serem analisados são o Preço Sobre Lucro (P/L), também chamado de Preço Sobre Resultado (PSR), o Preço Sobre Valor Patrimonial (P/VPA) e *Dividend Yield* (DY). Todos estes são obtidos através de cálculos matemáticos, de forma objetiva. Estes, além de outros indicadores e dados disponíveis em balanços patrimoniais e demonstrativos de resultados

de exercício (DRE), evidenciam a saúde financeira da empresa para que o investidor tenha total conhecimento sobre a situação e possa realizar sua tomada de decisão.

Analisar os dados de todas as empresas disponíveis é uma tarefa complexa e com alto custo de tempo e esforço. Em março de 2021, a B3 apresentava mais de 400 empresas listadas para negociação de ativos ([ELIAS, 2021](#)). A utilização de ferramentas computacionais para este processo pode facilitá-lo, podendo realizar uma filtragem de empresas a fim de que o investidor iniciante tenha um número menor de informações para analisar.

Com o avanço da tecnologia, a quantidade de dados gerados e armazenados aumenta diariamente. Segundo [GOLDSCHMIDT e PASSOS \(2005\)](#), a importância destes dados está ligada a capacidade de se extrair valor e conhecimento que sirvam de apoio a tomadas de decisões. A utilização de métodos de mineração de dados e inteligência computacional permitem uma melhor compreensão destas informações, desde o auxílio em estratégias de negócios até a melhoria da experiência de usuários em aplicações, criando ambientes personalizados com base em acessos anteriores, por exemplo.

Segundo a ONU, mais da metade dos pedidos de patentes para produtos baseados em inteligência artificial ocorreu somente entre os anos de 2013 e 2019, desde o surgimento do termo em 1956 ([EXAME, 2019](#)). No campo da economia, por exemplo, existem diversos estudos onde são desenvolvidos modelos de previsão do comportamento de ativos da bolsa de valores. Aplicar técnicas como *machine learning* e *deep learning* pode significar vantagens competitivas para as empresas e investidores.

Existem diversos modelos de inteligência computacional que podem ser aplicados no contexto do mercado financeiro. Estes podem ser preparados para, por exemplo, tentar prever comportamentos futuros baseados em comportamentos passados, ou, também, para realizar classificações com base em rótulos pré-determinados. No contexto econômico, a fim de auxiliar um investidor iniciante, os modelos podem ser desenvolvidos para prever os preços que um determinado ativo pode alcançar baseado em seus valores históricos. Também podem ser utilizados para classificá-los em categorias como “vender” ou “comprar”, além de realizar análises de portfólios e gerar recomendações, baseadas em parâmetros pré-definidos. Alguns destes modelos de inteligência computacional que podem ser utilizados são as redes neurais artificiais e as árvores de decisão.

Para um investidor iniciante, analisar centenas de ativos disponíveis para compor sua carteira de ações pode ser uma tarefa com alto custo de esforço e tempo. Utilizar ferramentas baseadas em inteligência computacional para ajudar neste processo pode diminuir os riscos de uma escolha precipitada e tornar o processo de composição da carteira mais otimizado.

1.1 JUSTIFICATIVA

Dado o cenário de crescimento no número de investidores no mercado brasileiro, cresce também a chance destes iniciantes tomarem decisões precipitadas em relação a quais empresas investir. Isso se deve, além da inexperiência, à grande quantidade de ativos disponíveis na bolsa de valores.

A análise fundamentalista é essencial para um bom desempenho da carteira de ações de um investidor. Principalmente para o investidor iniciante, fazer boas escolhas é essencial para que ele continue investindo futuramente e não abandone o mercado de maneira precoce. No entanto, a grande quantidade de empresas listadas no mercado dificulta um bom estudo de cada ativo de forma individual. O uso de inteligência artificial e mineração de dados em aplicações voltadas para o mercado financeiro pode simplificar este processo de análise e escolha de papéis.

Ao realizar um processo de filtragem dos ativos disponíveis na bolsa de valores brasileira, é possível reduzir o número de empresas a serem analisadas individualmente, evitando o gasto de tempo e esforço desnecessário com ativos que não serão lucrativos. Utilizando modelos de inteligência computacional, é possível realizar uma seleção de empresas que são mais propensas a gerar lucro para que sejam analisadas mais precisamente pelo investidor, aumentando as chances de uma boa tomada de decisão.

Existem diversas aplicações de inteligência computacional utilizando a análise técnica, que leva em conta apenas o preço dos ativos. Para realizar análises baseadas em indicadores fundamentalistas, porém, existem poucas opções disponíveis. Em uma busca no portal de Periódicos da CAPES, do governo federal brasileiro, não se obteve resultados em buscas de trabalhos brasileiros com os termos "inteligência computacional", "análise fundamentalista", "inteligência artificial", "balanços patrimoniais", "*machine learning*". Já ao combinar estes termos com "mercado financeiro" ou "bolsa de valores", foi possível encontrar trabalhos relacionados, porém, utilizando análise técnica e séries temporais.

Devido a este contexto, é justificável também o estudo e comparação de variados modelos a fim de entender e identificar quais possuem melhor comportamento e, assim, otimizar o resultado no objetivo proposto.

1.2 OBJETIVOS

Este trabalho tem como objetivo principal propor uma avaliação de modelos de aprendizado de máquina que sejam capazes de pré-selecionar empresas do mercado financeiro brasileiro, visando uma posterior utilização por parte do investidor iniciante, utilizando como base dados fundamentalistas, como balanços patrimoniais trimestrais e

demonstrativos de resultados, além de indicadores como preço sobre lucro (P/L). Para que seja atingido, este objetivo geral pode ser desmembrado em metas específicas mostradas na [subseção 1.2.1](#).

1.2.1 Objetivos Específicos

Dentre os objetivos específicos deste trabalho pode-se destacar:

- Desenvolver modelos de inteligência computacional para pré-selecionar empresas do mercado financeiro brasileiro utilizando como dados os balanços patrimoniais trimestrais, demonstrativos de resultados de empresas e indicadores fundamentalistas de cada ativo;
- Comparar o desempenho dos modelos e definir quais possuem melhor performance;
- Realizar simulações com cada modelo a fim de identificar se as empresas selecionadas geraram lucro.

1.3 ESTRUTURA DO TRABALHO

Este trabalho está estruturado em seis capítulos. O [Capítulo 1](#) apresenta uma breve contextualização do estudo, abordagem do problema, objetivo geral, objetivos específicos, justificativa do mesmo e sua organização. No [Capítulo 2](#), são apresentados os conceitos necessários para total entendimento do trabalho, como definições sobre o mercado financeiro, inteligência computacional e mineração de dados. No [Capítulo 3](#), explicita a metodologia proposta para o trabalho. O [Capítulo 4](#) mostra os passos da metodologia devidamente executados. No [Capítulo 5](#) é possível verificar os resultados obtidos e discussões a respeito dos mesmos. Já no [Capítulo 6](#) apresenta a conclusão deste trabalho e etapas futuras que podem ser realizadas.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo serão apresentados os conceitos mais relevantes para o entendimento e desenvolvimento do trabalho proposto. A fundamentação teórica será dividida em três seções: uma introdução ao mercado financeiro, onde serão explicitados conceitos sobre a bolsa de valores no contexto brasileiro, análise fundamentalista e seus indicadores, uma segunda seção, onde serão apresentados conceitos de mineração de dados e inteligência computacional, além de uma terceira, onde serão explicitadas as ferramentas tecnológicas utilizadas durante o desenvolvimento do trabalho.

2.1 INTRODUÇÃO AO MERCADO FINANCEIRO

No mercado nacional, os ativos financeiros, também chamados de ações ou papéis, podem ser negociados na B3, a bolsa de valores brasileira. Estes podem ser comprados ou vendidos, funcionando como uma parte da empresa em questão. As ações representam a menor porcentagem do capital social de uma empresa, e o investidor que realiza compra desses papéis torna-se sócio da mesma ([XPINVESTIMENTOS, 2021](#)). Estas negociações são realizadas diariamente, exceto finais de semana e feriados federais.

2.1.1 Características de uma ação

Uma ação possui diversas características que podem ser analisadas. Ao final de um dia de negociações, são contabilizadas algumas informações sobre um ativo: preço de abertura, de fechamento, preço máximo, mínimo, número de negócios, volume de ações negociado e volume financeiro negociado. Estes dados são armazenados e analisados historicamente, servindo de base para muitos indicadores técnicos e fundamentalistas utilizados por investidores ([BICHARA, 2019](#)).

O preço de uma ação pode ser definido como a interseção das curvas de oferta e demanda ([ELDER, 2004](#)). A oscilação de valores de um papel é uma das características do mercado financeiro. A volatilidade é uma variável que representa a frequência e intensidade dessa variação em um determinado período de tempo ([VOGLINO, 2020](#)). Ou seja, quanto maior a volatilidade de um ativo, mais significativa é a oscilação de seu preço. Uma volatilidade alta de uma ação aumenta sua imprevisibilidade, gerando um risco maior para o investidor.

2.1.2 Técnicas para diminuição de risco em investimentos

Existem conceitos comprovados que diminuem riscos quando se trata de investimentos no mercado financeiro e seleção de ativos para compor uma carteira de ações. Um dos mais difundidos é o da diversificação, definida por Markowitz em seus trabalhos publicados em 1952 e 1959, onde definiu a Teoria Moderna do Portfólio, trabalho reconhecido com o Prêmio Nobel de Economia no ano de 1990.

A diversificação da carteira de ações é amplamente aceita no mercado, podendo ser implementada de diversas maneiras. SWENSON (2009) demonstra uma estratégia aplicada pela fundação da Universidade de Yale, Estados Unidos, entre os anos de 1985 e 2008, baseada nas teorias definidas por Markowitz e utilizando uma abordagem de longo prazo.

2.1.2.1 Teoria Moderna do Portfólio

A Teoria Moderna do Portfólio utiliza conceitos como retorno desejado, risco e correlação. O retorno desejado representa o quanto um investidor espera receber de acordo com o nível de risco assumido. Este pode ser calculado como uma soma ponderada dos retornos individuais de cada papel da carteira. O risco é medido por MARKOWITZ (1959) como o desvio padrão de cada ativo da carteira em relação ao seu retorno médio, caracterizando ativos com maior volatilidade como os de maior risco. A correlação mede o comportamento dos ativos, comparando-os, possuindo valores entre -1 e 1. Ativos que possuem correlação de valor -1 possuem comportamentos totalmente opostos, ou seja, enquanto um está valorizado, o outro está desvalorizado. Já os que possuem valor 1 de correlação se movem de maneira semelhante (LEITE, 2020). Esta teoria considera que os investidores são avessos ao risco, onde este somente é assumido se for esperado um maior retorno.

Cada combinação de ativos de uma carteira pode ser plotado em um gráfico, onde o desvio padrão do portfólio se encontra no eixo das abscissas e o retorno esperado no eixo das ordenadas. A partir do gráfico, é possível encontrar todas as combinações que possuem menor risco e maior retorno, caracterizando a carteira eficiente. ALMONACID e SANTOVITO (2010) definem a fronteira eficiente como o conjunto de carteiras onde os ativos, para cada patamar de risco, possui o maior retorno possível, e para cada patamar de retorno esperado, o menor risco possível.

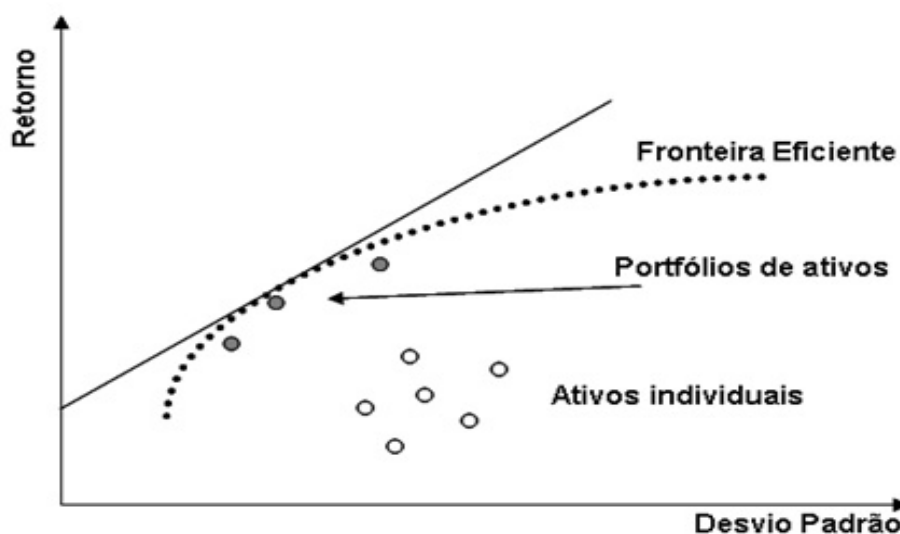


Figura 1 – Fronteira eficiente de Markowitz

Fonte: [InvestidorEmValor \(2020\)](#)

A carteira de ações com maior retorno e menor risco, segundo a Teoria Moderna do Portfólio, está o mais próximo possível desta fronteira eficiente mostrada na [Figura 1](#).

Para que seja aplicada a Teoria Moderna do Portfólio é preciso diversificar de diversas maneiras a carteira de ações a fim de encontrar a que esteja mais otimizada. Existem várias formas de realizar a diversificação, como entre classes de ativos e setores econômicos.

A diversificação entre classes de ativos se dá através do investimento em variados ativos disponíveis no mercado, como ações, títulos de renda fixa e fundos de investimento imobiliário. Já a diversificação entre setores econômicos se dá através do investimento em ações que atuem em áreas diferentes, como, por exemplo, aviação e petróleo, a fim de possuir papéis com baixa correlação entre si ([LEITE, 2020](#)).

2.1.3 Métodos de análise financeira

Para que o investidor escolha as melhores ações para compor sua carteira, é necessário que este analise as empresas em que pretende se tornar sócio. Existem técnicas para estudo de um ativo, a fim de auxiliar na tomada de decisão, como, por exemplo, a análise técnica e a análise fundamentalista.

2.1.3.1 Análise técnica

A análise técnica, também chamada de análise gráfica, consiste no estudo das flutuações do preço de uma ação, a fim de entender e prever seu comportamento futuro (MUPRHY, 1999). Este tipo de análise é mais comum para investidores que buscam o curto prazo, como os *day traders*, que são pessoas que atuam realizando compra e venda de ações com prazo máximo de um dia.

Este processo de estudo de uma ação consiste em utilizar gráficos multiformes junto a fórmulas matemáticas e estatísticas a fim de localizar tendências (MIRANDA et al., 2012). São utilizados conceitos como linhas de suporte e resistência, que marcam topos e fundos do gráfico de preços, indicadores de tendência, média móvel, dentre outros. Um exemplo da utilização de linhas de suporte e resistência pode ser visto na Figura 2.



Figura 2 – Exemplo de gráfico de preços utilizando linhas de suporte e resistência

Fonte: BlogRico (2019)

As linhas de suporte e resistência são extremamente importantes na análise técnica, indicando zonas de preço onde a possibilidade de rejeição é alta. Ou seja, destacam níveis onde, ao serem alcançados pelo preço do ativo, tendem a mudar seu movimento para o sentido contrário (PINHEIRO, 2019). Se um ativo está próximo de sua linha de suporte, portanto, a tendência é que o preço suba em um momento próximo. Por outro lado, se ele se encontra próximo de sua linha de resistência, pode-se prever que seu preço irá cair posteriormente.

Outro aspecto analisado na análise técnica, os indicadores de tendência também possuem uma grande importância neste processo. Uma aplicação é mostrada na Figura 3.



Figura 3 – Exemplo de gráfico de preços utilizando indicadores de tendência

Fonte: [BlogRico \(2019\)](#)

Com a identificação de topos e fundos em um gráfico de preços de um ativo, é possível demarcar a linha de tendência do mesmo, a fim de entender se esta é de alta ou de baixa. Os topos são os pontos mais altos de um preço, onde são encontradas as resistências. Já os fundos são os pontos mais baixos, onde são encontrados os suportes ([BLOGRICO, 2019](#)).

2.1.3.2 Análise fundamentalista

A análise fundamentalista é parte essencial de uma estratégia de investimento que visa o longo prazo, já que é a partir dela que se pode identificar como está a saúde financeira de uma empresa e se o preço de uma ação está supervalorizado ou não.

Essa metodologia de análise considera que o valor justo para uma ação de uma empresa está relacionado à sua capacidade de gerar lucros futuros ([MIRANDA et al., 2012](#)). Essa análise leva em consideração fundamentos econômicos do mercado, balanços patrimoniais da empresa, demonstrativos de resultados de exercício (DRE), indicadores de saúde financeira, métodos de definição de preço justo, dentre outros.

A análise fundamentalista pode ser dividida em duas etapas: a análise quantitativa e a análise qualitativa. A primeira diz respeito aos números da empresa, saúde financeira, que é realizada utilizando os indicadores fundamentalistas. Já a segunda é feita com base em aspectos que medem a qualidade da gestão e governança corporativa, como a experiência dos gestores, por exemplo.

Segundo [MOORE \(2012\)](#), alguns dos principais indicadores quantitativos que podem ser utilizados em uma análise são:

- P/L (Preço / Lucro): Indica a rentabilidade de uma ação. É calculado pela divisão do preço da ação pelo lucro líquido da empresa por ação;
- P/VPA (Preço / Valor patrimonial por ação): Compara o valor de mercado da empresa com seu valor contábil ([INFOMONEY, 2020](#)). É calculado pela divisão do preço da ação pelo valor patrimonial por ação da empresa;
- Dividend Yield: É a taxa de retorno de dividendos de uma empresa, sendo calculado pelo valor esperado em proventos, por ação, pelo preço atual do papel ([INFOMONEY, 2020](#));
- EBITDA (*Earnings before interest, taxes, depreciation and amortization*): Representa o lucro gerado pela empresa, sem incluir investimentos financeiros, empréstimos ou impostos;
- ROI (*Return on investment*): É o retorno sobre o total de investimentos feitos pela empresa, podendo ser entendido como a capacidade dos ativos de uma empresa gerarem lucro ([INFOMONEY, 2020](#)).

2.2 TECNOLOGIAS APLICADAS

Com o crescimento da quantidade de informações geradas diariamente, é necessário a utilização de tecnologias que auxiliem seu tratamento e que possam extrair conhecimento útil. [TAURION \(2013\)](#) define dados como recursos naturais da sociedade da informação, possuindo valor apenas se tratados, analisados e usados para tomada de decisões. A utilização de ferramentas tecnológicas se tornou essencial para a análise de dados, o que possibilitou o desenvolvimento de áreas como mineração e inteligência artificial.

2.2.1 Mineração de dados

O termo mineração de dados é usualmente utilizado para referenciar o processo de descoberta de conhecimento em bases de dados, conhecido como *Knowledge Discovery in Databases* (KDD) ([GOLDSCHMIDT; PASSOS, 2005](#)). Este processo tem como objetivo principal analisar um conjunto de informações a fim de extrair conhecimento e valor para auxiliar tomadas de decisão. A mineração de dados é uma etapa a ser realizada no KDD.

O processo de descoberta de conhecimento em bases de dados é composto por uma estrutura de etapas realizadas, representadas na [Figura 4](#).

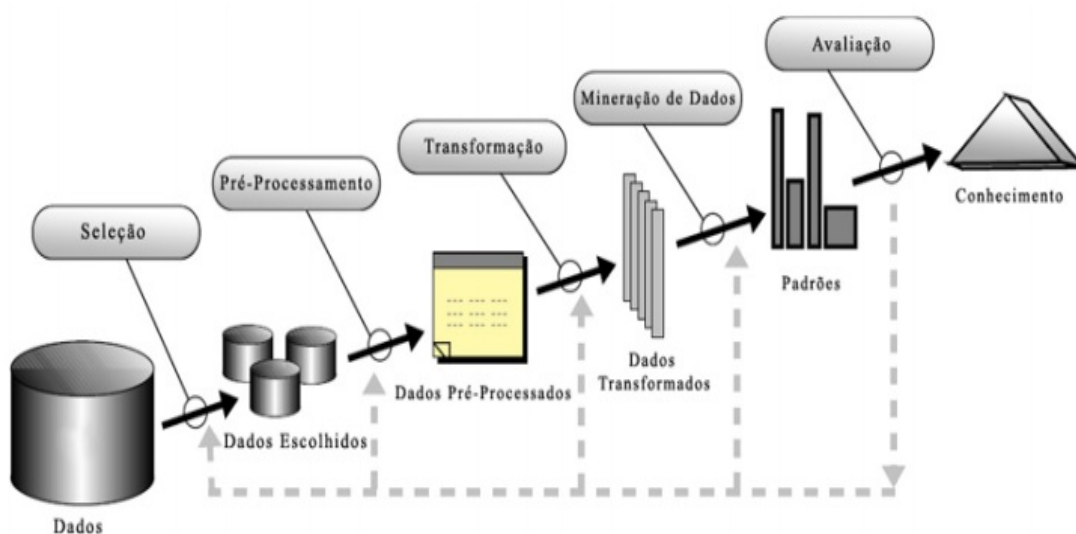


Figura 4 – Etapas do processo de KDD

Fonte: [FAYYAD, PIATETSKY-SHAPIRO e SMYTH \(1996\)](#)

A seleção dos dados é a primeira etapa a ser realizada, que consiste na identificação de quais informações disponíveis serão consideradas no processo de KDD ([GOLDSCHMIDT; PASSOS, 2005](#)).

O pré-processamento realiza, dentre outras funções, a limpeza dos dados. Esta etapa é responsável por identificar possíveis dados irrelevantes, ausentes ou inconsistentes, que podem ser tratados ou eliminados. A transformação, ou codificação, também é uma parte do pré-processamento, onde as informações originais são alteradas para formatos mais apropriados para análise. Podem ser realizados procedimentos como normalização, seleção de atributos, discretização e geração de hierarquia de conceitos ([KUMAR; STEINBACH; TAN, 2009](#)).

Segundo [GOLDSCHMIDT e PASSOS \(2005\)](#), a mineração de dados é a etapa principal do processo de KDD, pois é onde ocorre a busca de conhecimento desejada. Nesta etapa, são aplicados algoritmos sobre as bases de dados previamente tratadas. O processo de mineração pode se dar através de duas abordagens: o aprendizado supervisionado e o não supervisionado.

Ainda de acordo com [GOLDSCHMIDT e PASSOS \(2005\)](#), a abordagem de aprendizado supervisionado considera que deva existir valores de entrada e saída. Ou seja, existe um processo a ser realizado com os dados que geram um resultado esperado. Algoritmos que utilizam esta abordagem, como o *Backpropagation*, necessitam de pelo menos dois conjuntos de dados: o conjunto de treinamento e o conjunto de teste. O modelo de

conhecimento é construído a partir do conjunto de treinamento e avaliado pelo conjunto de teste. Já a abordagem não supervisionada não considera uma informação de saída desejada, e sim um modelo que busca encontrar relacionamentos entre os registros, sendo utilizada em algoritmos como *K-Means* e *Apriori*.

De acordo com o objetivo do processo de KDD, podem ser definidas tarefas de mineração a serem aplicadas. As mais comuns de serem aplicadas, segundo CAMILO e SILVA (2009), são:

- Descrição: tarefa utilizada para descrever padrões e tendências, geralmente fornecendo uma possível interpretação para os resultados obtidos;
- Classificação: realizada para determinar a que grupo previamente classificado um registro apresenta mais semelhanças. Pode ser aplicada por diversos métodos, como árvores de decisão, classificação Bayesiana, baseada em regras, redes neurais, dentre outras. Dependendo do método utilizado, pode ser supervisionada ou não-supervisionada;
- Regressão: também chamado de estimação, é um processo similar à classificação, podendo estimar o valor de uma variável a partir dos demais dados. É utilizada quando o registro possui uma identificação numérica, e não categórica. É aplicada em métodos como a regressão linear e a não-linear;
- Predição: similar à regressão, porém possui como objetivo determinar o possível valor futuro de um atributo. É um conceito que é utilizado em conjunto por métodos que também realizam classificação e regressão;
- Agrupamento: também chamada de *clustering*, realiza a identificação e agrupamento de registros similares entre si. As técnicas que utilizam este conceito são consideradas não-supervisionadas, como métodos de particionamento, hierárquicos, baseados na densidade, em grade ou em modelo;
- Associação: tem como objetivo identificar relações entre atributos, e não entre registros. Utiliza métodos de regras de associação. É um processo não-supervisionado.

2.2.2 Correlação de Pearson

Ao analisar um conjunto de dados, é necessário entender como as informações presentes se relacionam entre si. A correlação entre duas variáveis traz um valor matemático que representa uma comparação entre seus comportamentos.

O coeficiente de correlação de Pearson, ou coeficiente de correlação simples, mede o grau de associação linear entre duas variáveis distintas (FILHO; JÚNIOR, 2009). Esta constante tem valor entre -1 a 1, onde o intermediário, zero, indica ausência de correlação.

Ainda segundo FILHO e JÚNIOR (2009), quanto mais próximo de 1 é o valor, independente do sinal, maior é o grau de dependência estatística linear entre as variáveis analisadas. Se o valor é negativo, a associação é inversa, onde apresentarão comportamentos antagônicos. Ao se obter o valor da correlação de Pearson entre duas variáveis, portanto, é possível entender a força dessa correlação, sendo mais fortes as mais próximas dos extremos 1 e -1, e as mais fracas valores mais próximos à 0. Na Figura 5 é possível identificar uma representação gráfica do coeficiente de correlação simples entre duas variáveis.

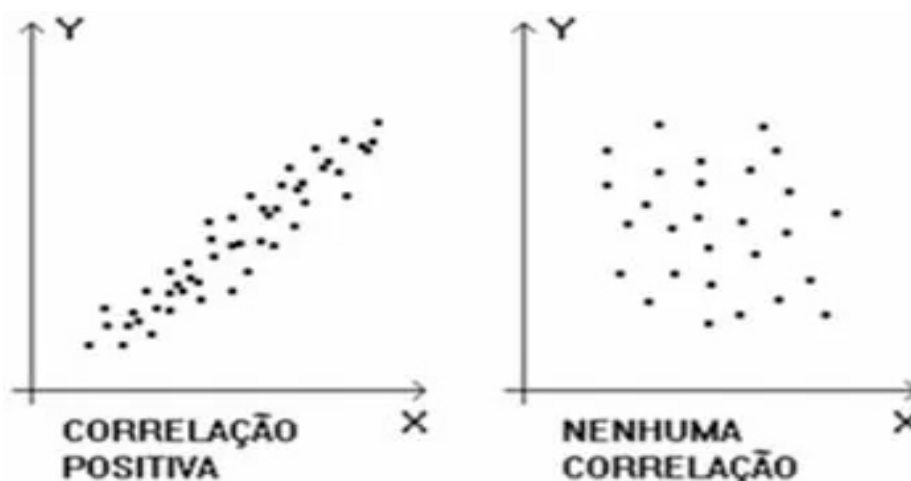


Figura 5 – Exemplo gráfico da correlação de Pearson

Fonte: Adaptado de FIRMINO (2020)

Na representação gráfica mostrada na Figura 5 é possível identificar dois gráficos mostrando a relação entre duas variáveis X e Y. Do lado esquerdo, é visualizado um exemplo onde os valores dos eixos possuem uma correlação de Pearson positiva, onde possuiria valores mais próximos à 1. Já ao lado direito, não existe nenhuma correlação explícita, onde, ao ser calculado o coeficiente em questão, o valor obtido se aproximaria de zero.

2.2.3 Inteligência computacional

Para que um sistema seja considerado "inteligente", TVETER (1998) enumera características fundamentais para essa classificação:

- A capacidade de resolução de novos problemas;

- A velocidade em que o sistema é capaz de resolver estes problemas;
- Experiência do sistema, ou seja, a exposição do mesmo a situações como forma de treinamento;
- A capacidade de aprendizagem através de erros, aprimoramento de parâmetros internos de avaliação e correção.

A inteligência artificial pode ser considerada a aplicação das características definidas por TVETER (1998) a um sistema. A partir disso, problemas propostos a este sistema devem ser resolvidos de forma eficaz.

Existem diversos métodos de aplicação de inteligência artificial, como, por exemplo, algoritmos evolutivos, genéticos e lógica *fuzzy*. Também há a aplicação de técnicas naturais do homem e da natureza para resolver problemas em sistemas tecnológicos. Esse campo de estudo é conhecido como inteligência computacional, e engloba métodos bastante utilizados como as redes neurais artificiais.

2.2.4 Modelos de inteligência computacional

Com a evolução dos estudos sobre inteligência computacional, surgiram diversas técnicas distintas que podem ser utilizadas para resolver problemas propostos. Cada técnica pode ser útil para um determinado tipo de problema, devendo ser escolhida de forma a otimizar os resultados. Alguns modelos conhecidos são os de regressão linear, regressão logística e as árvores de decisão.

2.2.4.1 AdaBoost Classifier

O algoritmo *AdaBoost Classifier*, apresentado pela primeira vez em 1995 e utilizado em diferentes campos, possui diversas vantagens se comparado a outros modelos de inteligência computacional. Segundo FREUND e SCHAPIRE (1999), ele é rápido, simples e fácil de ser programado.

Este modelo é um dos mais populares a utilizar a técnica de *boosting*, onde são combinados diversos modelos mais fracos a fim de criar um modelo de aprendizagem mais forte. Sua principal característica é a definição de pesos para o conjunto de dados onde, inicialmente, todos são definidos igualmente. Ao final de uma rodada, os pesos são ajustados conforme os erros e acertos do modelo. Assim, conforme o processo se repete, é possível minimizar o erro de treinamento e otimizar classificadores mais fracos (FREUND; SCHAPIRE, 1999). Um exemplo do funcionamento deste algoritmo pode ser visto na Figura 6.

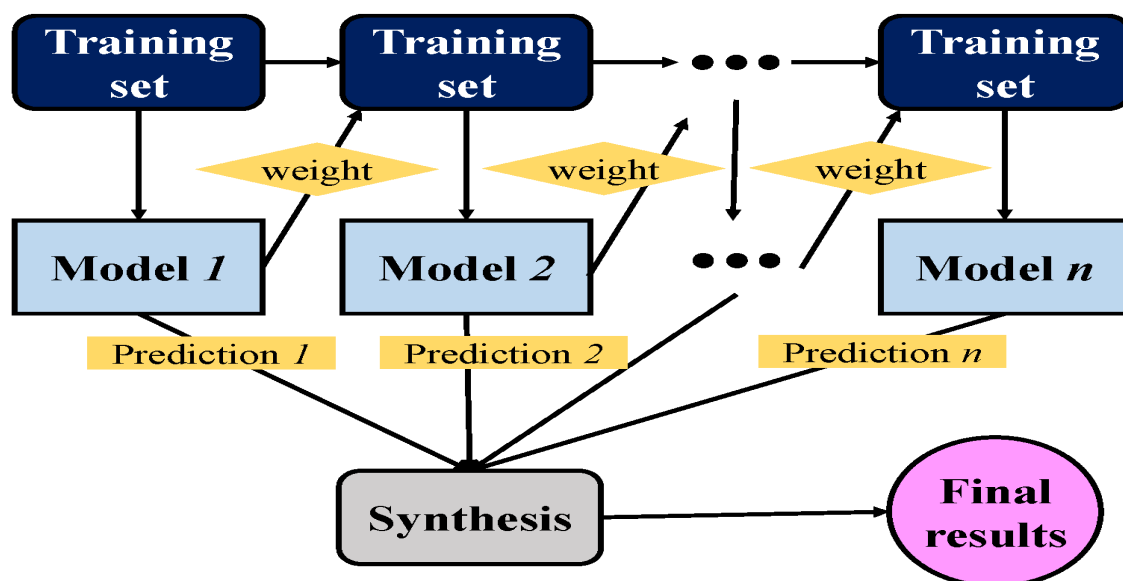


Figura 6 – Funcionamento do algoritmo *AdaBoost Classifier*

Fonte: Wang, Xu e Yang (2021)

Este processo de cálculo realizado pelo *AdaBoost*, representado na Figura 6, consiste em diversos treinamentos da base de dados (*training sets*), onde a cada execução é obtido um novo peso (*weight*), que é adicionado ao treinamento de um novo modelo (*model*), gerando assim as predições (*prediction*). Todos resultados são sintetizados (*synthesis*) a fim de se obter um resultado final (*final results*).

2.2.4.2 Decision Tree Classifier

Também conhecido como árvore de decisão, este modelo se dá por um grafo acíclico capaz de realizar classificações. A cada nó interno do grafo, uma característica é analisada. Caso a condição do nó seja satisfeita, o processo seguirá por um lado da árvore, caso não, seguirá para o lado oposto. Até que seja encontrado um nó folha, onde há a rotulação, o algoritmo se repete (BURKOV, 2019).

Uma árvore de decisão, portanto, trabalha de forma recursiva, dividindo um conjunto de treinamento, até que cada dado deste treinamento pertença a um rótulo pré-definido. Na Figura 7 é possível ver o funcionamento de uma árvore de decisão.

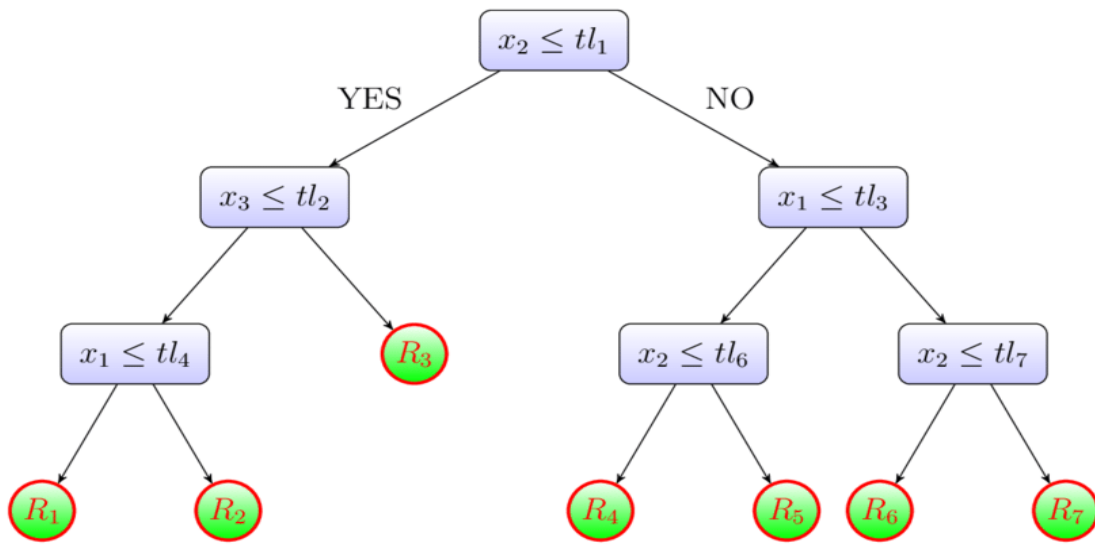


Figura 7 – Funcionamento do algoritmo *Decision Tree Classifier*

Fonte: Akbari, Ng e Solnik (2021)

No processo de execução de um algoritmo de árvore de decisão mostrado na Figura 7, observa-se que o primeiro critério analisado, ao topo, pode proporcionar dois caminhos para o dado. Dependendo se esta entrada satisfaz ou não determinadas condições, ela é levada até uma folha da árvore, ou seja, uma classificação final.

2.2.4.3 Random Forest Classifier

Segundo BREIMAN (2001), o algoritmo de *random forest*, ou floresta aleatória, consta de um conjunto de classificadores de *decision trees*, ou árvores de decisão, onde cada um destes depende dos valores de um vetor aleatório treinado de forma independente, com a mesma distribuição para todas as árvores da floresta.

Uma *random forest* se dá a partir da criação de árvores de decisão, geradas através da quebra da massa de dados e construção de vários subconjuntos. As árvores são construídas pela seleção aleatória de atributos a partir destes subconjuntos gerados. O conjunto destas árvores geradas é o que constitui as florestas aleatórias (LORENZETT; TELOCKEN, 2016). Após a criação das árvores, é feita a classificação de qual possui melhor ganho de conhecimento para resolver o problema proposto.

As *random forests* são consideradas mais poderosas do que uma simples árvore de decisão, sendo menos sensíveis a ruídos (LORENZETT; TELOCKEN, 2016). Na Figura 8 é possível visualizar o funcionamento do algoritmo.

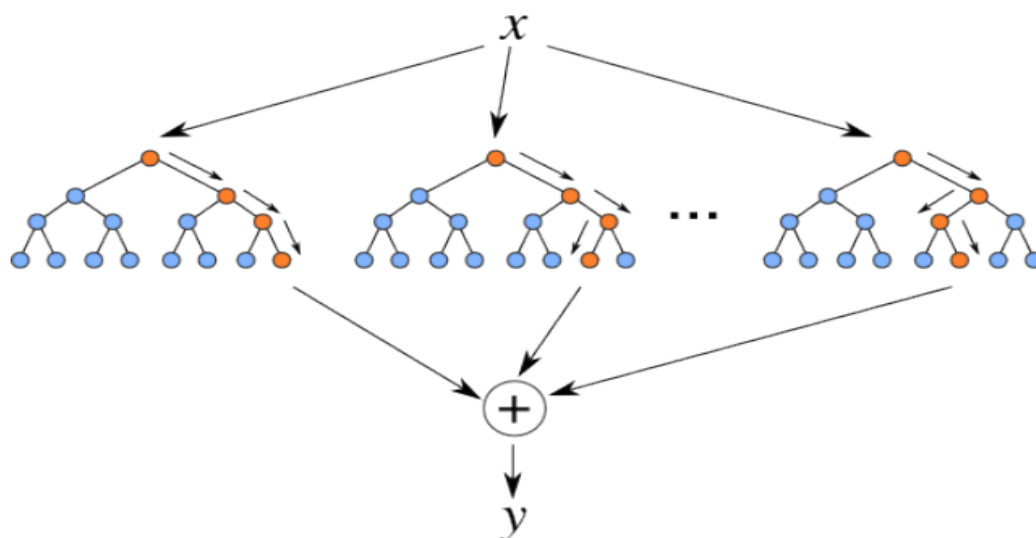


Figura 8 – Funcionamento do algoritmo *Random Forest*

Fonte: LORENZETT e TELOCKEN (2016)

A partir de uma massa de dados, diversas árvores de decisão diferentes são geradas, onde, ao final, são avaliadas de acordo com seu desempenho, como é mostrado na Figura 8.

Segundo BREIMAN (2001), algumas das vantagens deste algoritmo são a acurácia, baixa exigência computacional, ser relativamente robusto a *outliers* e ruídos, fornecer estimativa de importância de uma variável para a classificação e possuir um método eficaz para estimar dados ausentes, garantindo a precisão.

2.2.4.4 Extra Trees Classifier

O algoritmo *Extra Trees*, também é conhecido como *Extremely Randomized Trees*, ou árvores extremamente aleatórias. Este funciona de forma diferente de outros algoritmos baseados em árvores. Segundo ERNST, GEURTS e WEHENKEL (2005), este modelo pode ser definido como um conjunto de árvores de decisão não podadas de acordo com o procedimento tradicional de cima para baixo.

Em seu funcionamento, o *Extra Trees* utiliza pontos de corte totalmente aleatórios e toda a amostra de aprendizado é utilizada para a construção das árvores. Este se assemelha com o algoritmo de *random forest*, porém, devido a aleatoriedade das divisões dos nós, possuem uma menor variabilidade (ERNST; GEURTS; WEHENKEL, 2005).

2.2.4.5 Gradient Boosting Classifier

O *Gradient Boosting Classifier*, assim como o *AdaBoost Classifier*, também é baseado na técnica de *boosting*. Seu funcionamento se dá pela construção de modelos de regressão aditivos, onde há o ajuste sequencial, a cada interação, de uma função parametrizada aos "pseudo"resíduos pelo método dos mínimos quadrados. Estes "pseudo"resíduos são o gradiente da perda funcional minimizada com relação aos valores do modelo em cada ponto de dados de treinamento avaliado na interação (FRIEDMAN, 2002).

Segundo HASTIE, TIBSHIRANI e FRIEDMAN (2008), a partir de diferentes critérios para o cálculo da função de perda, é possível gerar algoritmos específicos. Para a classificação é utilizada uma função de perda chamada *deviance*.

2.2.4.6 Light Gradient Boosting Machine

O modelo *Light Gradient Boosting Machine*, abreviado como *LightGBM*, é baseado no algoritmo de *Gradient Boosting*, utilizando árvores de decisão para realizar classificações. Como um algoritmo que utiliza a técnica de *boosting*, o *LightGBM* coleciona diversos modelos considerados mais simples, ou mais fracos, a fim de criar um modelo mais complexo, ou mais forte (SOUSA, 2020).

Este método introduz o conceito de *Gradient-based One-side Sampling* (GOSS), onde são selecionadas as instâncias de maior gradiente e uma pequena parcela das demais, escolhidas de forma aleatória, para que seja calculado o melhor valor para divisão do nó da árvore. Devido a esta técnica, o *LightGBM* é capaz de lidar bem com grandes conjuntos de dados. Outra característica importante é a expansão das árvores "baseada em folhas", onde a folha que melhor divide o conjunto de dados é expandida (KE et al., 2017).

2.2.4.7 Extreme Gradient Boosting Machine

Mais um derivado do *Gradient Boosting*, o algoritmo de *Extreme Gradient Boosting Machine*, também conhecido como *XGBoost* ou XGB, utiliza também o princípio de árvores de decisão, realizando combinações destas para tomadas de decisão. Este modelo faz uso da formalização do modelo para realizar a regularização e controle do sobreajuste (VERMA; PAL; KUMAR, 2019).

Segundo ROCHA, CARMO e VASCONCELOS (2018), o algoritmo XGB utiliza o gradiente para aumentar a construção das árvores reforçadas, obtendo a importância de cada característica para o modelo em treinamento. Conforme aumenta a frequência de utilização de uma característica específica para tomadas de decisões importantes, maior será a pontuação desta. Este algoritmo também conta com um método de validação cruzada embutido em cada interação.

2.2.4.8 CatBoost Classifier

O algoritmo *CatBoost Classifier*, nome derivado de *categorical boosting*, também é uma adaptação do algoritmo *Gradient Boosting*, tendo suporte a variáveis categóricas (SANTOS, 2020b). Segundo Ghori et al. (2020), este modelo lida com categorias por conta própria e não necessita de grandes conjuntos de dados para um treinamento de forma extensiva, independente da quantidade de parâmetros utilizados.

O *CatBoost Classifier* não requer diversas etapas de pré-processamento para funcionar. Este necessita apenas de índices de recursos categóricos para realizar a transformação destas categorias em dados numéricos (HANCOCK; KHOSHGOFTAAR, 2020).

2.2.4.9 K-Nearest Neighbors Classifier

Também conhecido como *KNN Classifier*, o método *K-Nearest Neighbors Classifier* se baseia na proximidade do elemento com relação aos seus vizinhos, onde é atribuído ao grupo mais comum entre seus k vizinhos mais próximos. As amostras são classificadas de acordo com o grupo a qual pertencem a maioria de seus vizinhos, onde a quantidade destes vizinhos é definida pelo parâmetro k (GUIMARAES, 2019). Este método aprende com o armazenamento dos dados de treinamento baseado em instâncias. Quando uma nova instância surge, ela é classificada de acordo com instâncias similares. A proximidade dos vizinhos é definida de acordo com a distância Euclidiana (LUNARDI; VITERBO; BERNARDINI, 2015).

O KNN tem como vantagens a alta precisão, insensibilidade a *outliers* e não permite suposições sobre os dados. Também funciona com valores numéricos e nominais, porém, tem um alto custo computacional, sendo essa uma desvantagem do modelo (HARRINGTON, 2012). Na Figura 9 é possível identificar uma representação de um novo dado, não classificado, antes e depois de ser submetido ao KNN e ser incluído em um grupo existente.

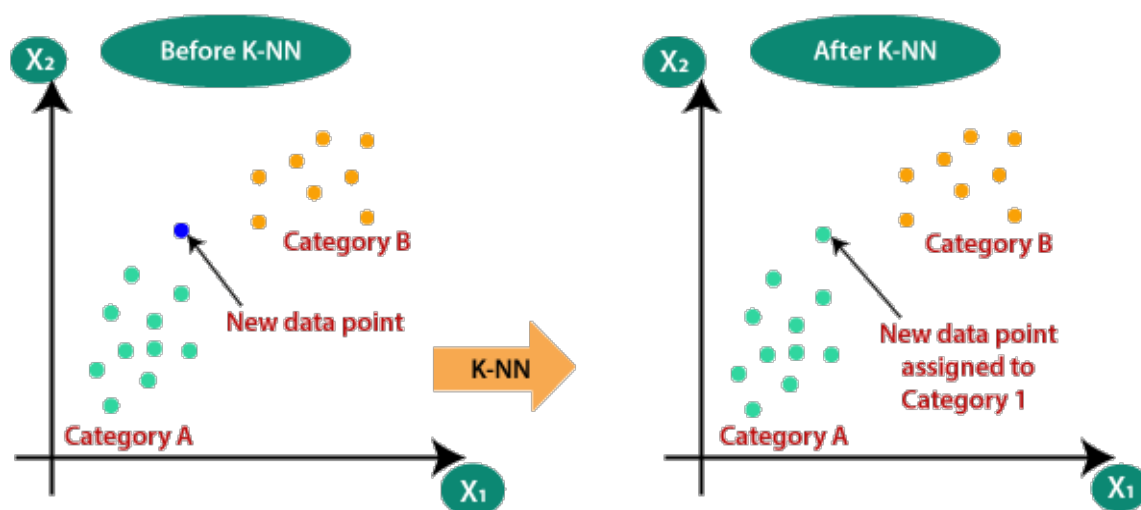


Figura 9 – Funcionamento do algoritmo *KNN Classifier*

Fonte: [JavaTPoint \(2011\)](#)

Neste exemplo mostrado na [Figura 9](#), existem duas categorias, A e B, e um dado novo a ser classificado (*new data point*). Após ser submetido ao KNN, este novo dado foi classificado como categoria A, devido à sua maior proximidade com os outros elementos também deste tipo do que com dados de categoria B.

2.2.4.10 Logistic Regression

O método *Logistic Regression*, ou regressão logística, tem como objetivo prever a probabilidade de cada observação pertencer a uma classe. Essa probabilidade é estimada através da distribuição binomial, onde o parâmetro é a probabilidade de ocorrência de uma classe específica. A probabilidade estimada deve estar limitada ao intervalo $[0,1]$ e apresentar relação direta com os preditores mensurados para cada observação do conjunto de dados ([KUHN; JOHNSON, 2013](#)).

2.2.4.11 Linear Discriminant Analysis

Abreviado como LDA, o modelo de *Linear Discriminant Analysis* é usado para tarefas de classificação. Seu funcionamento se dá ao encontrar uma combinação linear entre as variáveis e separando-as em classes. Este modelo, na prática, tenta separar os dados em diversas classes e desenha uma região entre as mesmas ([PEREIRA, 2020](#)).

2.2.4.12 Quadratic Discriminant Analysis

O QDA (*Quadratic Discriminant Analysis*) é um classificador paramétrico não linear. Neste, não existe uma pressuposição de que os grupos possuem matrizes com covariância igual (AQUINO; FONSECA; OLIVEIRA, 2016).

O modelo classificador QDA tem funcionamento similar ao LDA, onde um item é classificado no grupo onde possui a menor distância quadrada. Segundo AQUINO, FONSECA e OLIVEIRA (2016), porém, ao contrário do LDA, o classificador QDA produz fronteiras de decisão mais flexíveis, tendo mais parâmetros para estimar.

2.2.4.13 Naïve Bayes Classifier

O modelo *Naïve Bayes Classifier*, ou NB, também é utilizado para tarefas de classificação, onde calcula a probabilidade de um item pertencer a uma determinada classe. Este utiliza o teorema de Bayes para o cálculo. O NB possui uma simples implementação e é comumente utilizado para projetos onde é necessário prever em que categoria um item pertence, por exemplo. (MEDHAT; HASSAN; KORASHY, 2014).

2.2.4.14 Ridge Classifier

O modelo classificador *Ridge Classifier* é baseado em regressões lineares, onde utiliza uma forma de regularização através de penalidades (VOVK, 2013). Este modelo converte os dados das classes para um intervalo de -1 a 1 e realiza a classificação utilizando regressão. O valor considerado mais alto na previsão é aceito como uma classe de destino. Para tarefas onde há várias classes envolvidas, os classificadores são treinados utilizando uma abordagem de um contra todos (HASTIE, 2020).

2.2.4.15 Support Vector Machine

O modelo de aprendizagem de máquina *Support Vector Machine* (SVM), ou Máquina de Vetor de Suporte, foi desenvolvida por Vapnik (1998). Este algoritmo é muito utilizado em estudos de reconhecimento de padrões e possui maior generalização.

O modelo SVM consiste em mapear os dados para um espaço de entrada de alta dimensão, onde é construído um hiperplano de separação ideal. O hiperplano ótimo possui máxima margem de separação entre duas classes (SUYKENS; VANDEWALLE, 1999). Na Figura 10 é possível visualizar um hiperplano ótimo e a máxima margem de separação.

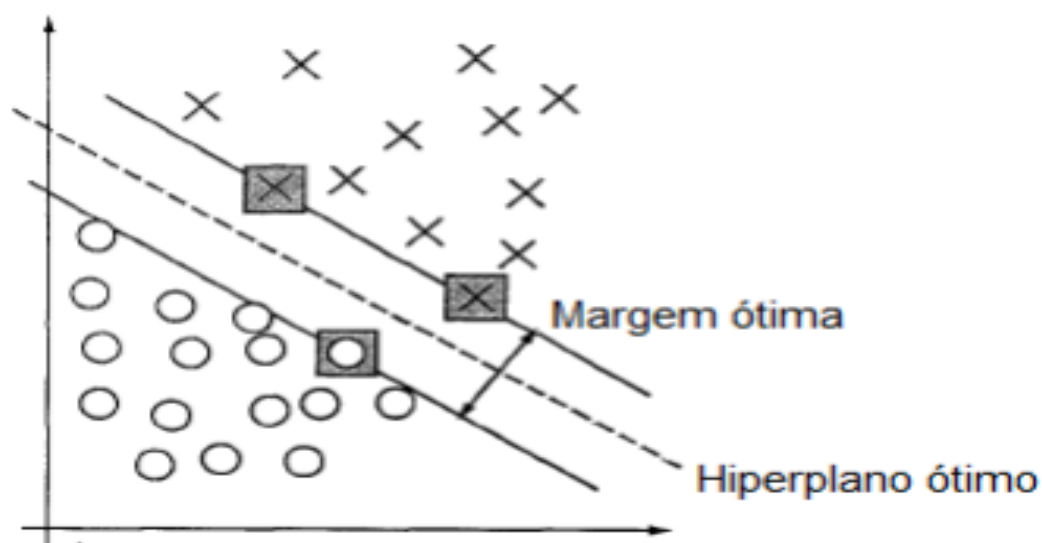


Figura 10 – Hiperplano ótimo e máxima margem de separação

Fonte: [TERAMACHI \(2020\)](#)

Em seu funcionamento, o algoritmo SVM mapeia os dados de entrada e constrói estes hiperplanos em um espaço n -dimensional. A dimensão deste espaço pode gerar um alto custo computacional, sendo necessário a utilização de funções *kernel*, reduzindo-os ([TERAMACHI, 2020](#)).

2.2.4.16 Redes neurais artificiais

As redes neurais artificiais (RNAs) são modelos matemáticos que simulam a forma pelo qual o cérebro humano realiza determinada tarefa ou função ([HAYKIN, 1998](#)), visto que o sistema biológico é altamente complexo, porém possui enorme capacidade de aprendizagem. Estes modelos são formados por neurônios artificiais, que representam unidades de processamento, conectados entre si. Na [Figura 11](#) é mostrado um neurônio artificial genérico.

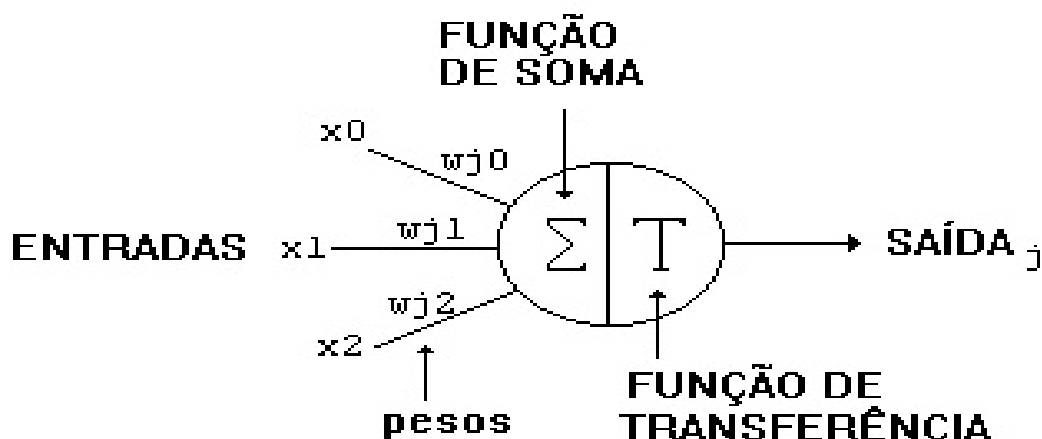


Figura 11 – Exemplo de neurônio artificial

Fonte: TAFNER (1998)

Neste neurônio artificial genérico apresentado na [Figura 11](#) é possível identificar alguns parâmetros necessários para sua existência. Valores de entrada, seus respectivos pesos, sua função de soma, função de transferência e saída são características que devem existir para o funcionamento correto de um neurônio e de uma RNA.

O objetivo de uma rede neural artificial é aprender através de uma entrada de dados e realizar tarefas como classificação e reconhecimento de padrões. Cada neurônio projetado possui uma função de ativação que, a partir de valores de entrada e outros previamente armazenados, gera um valor de saída, que será propagado para os neurônios seguintes ([FREITAS, 2001](#)).

Em cada conexão existente em uma rede neural artificial são atribuídos *pesos sinápticos* como forma de armazenar conhecimento e gerar os resultados esperados. Segundo [HAYKIN \(1998\)](#), é possível, portanto, identificar os elementos básicos de um neurônio artificial como sendo um conjunto de conexões (sinapses) e seus respectivos pesos, um elemento de soma, também chamado de função soma, que realiza o somatório entre os valores de entrada de acordo com os pesos atribuídos, e uma função de ativação, que definirá o valor de saída. Também é possível haver um parâmetro *bias*, a fim de ajustar o valor de saída do neurônio.

2.2.4.16.1 Arquitetura das redes neurais artificiais

Em sua estrutura, uma rede neural apresenta, pelo menos, três camadas de neurônios, sendo uma camada de entrada, uma de saída, e uma intermediária, também conhecida como camada oculta ou escondida, onde ocorre a maior parte do processamento. Na [Figura 12](#) é possível identificar um exemplo de rede neural artificial com as camadas citadas.

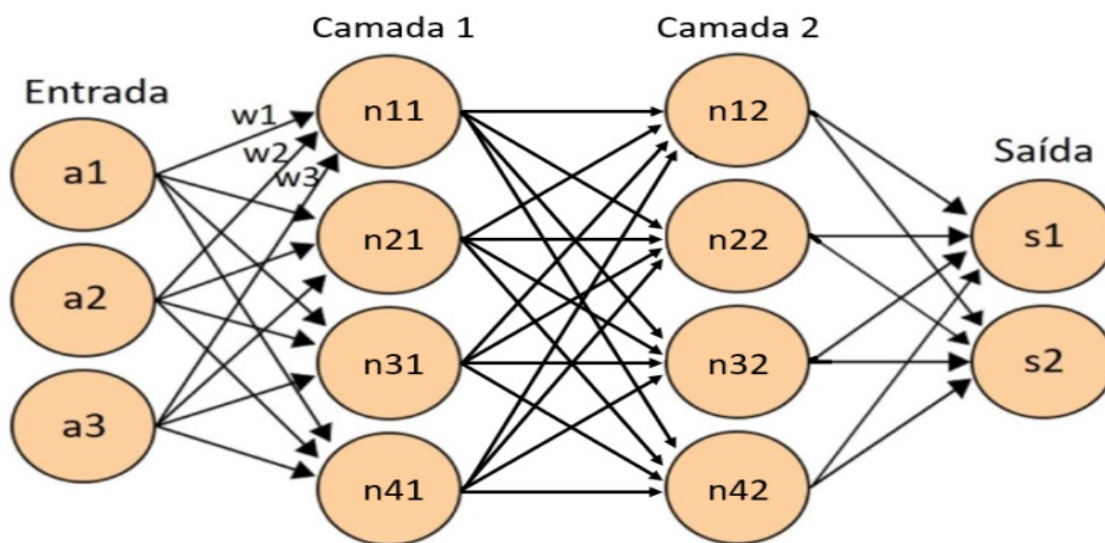


Figura 12 – Exemplo de camadas de uma rede neural

Fonte: [DidaticaTech \(2020\)](#)

De forma geral, na [Figura 12](#), a camada de entrada se comunica com a primeira camada intermediária (camada 1), transmitindo os dados e seus respectivos pesos. Após serem processados por estes neurônios da camada 1, os dados são repassados, junto aos pesos, para a camada 2, também intermediária. Ao fim, estes são transmitidos para a camada de saída, onde, a partir desta, serão devolvidos os resultados finais do processamento da RNA.

Os modelos de RNAs podem se diferenciar também de acordo com o tipo de conexão existente entre os neurônios. As redes com conexão *feedforward*, ou acíclicas, se comunicam unidirecionalmente da entrada para a saída. Já as que possuem conexão recorrente (cíclicas), um neurônio pode se comunicar com outro de qualquer camada da rede. A [Figura 12](#) mostra uma rede neural artificial do tipo *feedforward*, enquanto a [Figura 13](#) apresenta uma rede com conexão recorrente.

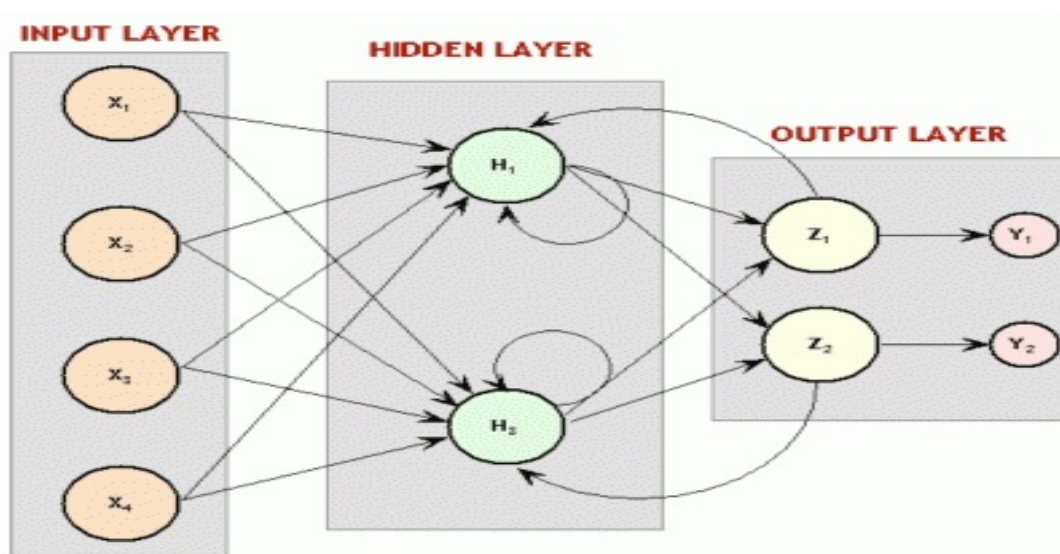


Figura 13 – Exemplo de camadas de uma rede neural

Fonte: [DeepLearningBook](#) (2020)

Como mostrado na [Figura 13](#), uma RNA cíclica possui neurônios que se comunicam com outros de qualquer camada, em qualquer sentido. Por exemplo, é possível identificar os neurônios H_1 e H_2 se comunicando com eles mesmos. Também é mostrado o neurônio Z_1 , da camada de saída, se comunicando com o H_1 , da camada oculta.

2.2.4.16.2 Processo de aprendizagem das redes neurais artificiais

A principal característica desejada em um método de inteligência computacional é a capacidade de aprendizagem do mesmo baseado em dados de entrada. Nas redes neurais artificiais, este processo ocorre internamente com o ajuste dos pesos sinápticos. A definição do conjunto de dados de treinamento é uma parte essencial para o bom funcionamento de uma RNA ([FREITAS, 2001](#)).

Ainda segundo [FREITAS \(2001\)](#), o aprendizado de redes neurais artificiais pode ocorrer de três formas:

- Aprendizagem supervisionada (ou aprendizagem com professor): é indicado a rede, por um agente externo, o resultado final desejado para a entrada em questão. O sistema compara a saída da RNA com o que era esperado para realizar o ajuste dos pesos sinápticos.
- Aprendizagem por reforço: semelhante à supervisionada, porém sem o fornecimento da resposta final desejada, sendo apenas indicado um sinal de reforço para indicar se a resposta gerada pela rede está correta ou não.

- Aprendizagem não-supervisionada (ou aprendizagem sem professor): não existe indicação da resposta desejada.

2.2.4.16.3 Redes neurais artificiais de múltiplas camadas

As redes neurais artificiais de múltiplas camadas, também chamadas de MLP (*Multilayer Perceptron*), permitem a solução de problemas mais complexos, sendo amplamente utilizadas (BICHARA, 2019).

O método de aprendizagem mais utilizado para redes neurais artificiais do tipo MLP é o algoritmo de *backpropagation* (FREITAS, 2001), que segue uma abordagem supervisionada. Este processo é realizado em duas etapas: a primeira, *feedforward*, onde os dados são recebidos na camada de entrada, processados, e encaminhados para a camada de saída; e a segunda, *backpropagation*, onde ocorre o ajuste dos pesos sinápticos de acordo com os resultados obtidos na primeira etapa. Na Figura 14 é representado este aprendizado.

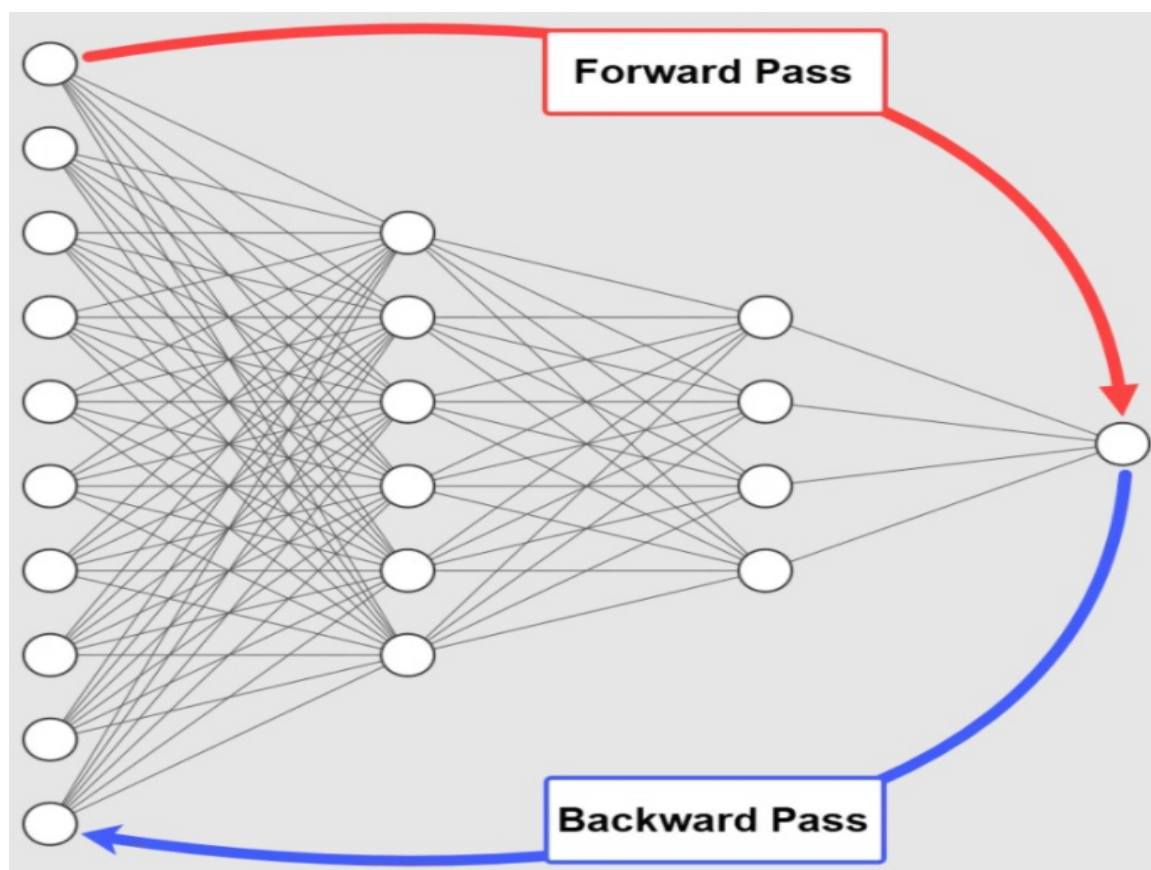


Figura 14 – Representação do aprendizado *backpropagation*

Fonte: GAD (2021)

É possível identificar na representação de um processo de aprendizagem de uma RNA mostrada na [Figura 14](#) tanto a primeira etapa, *feedforward* (*forward pass*), em vermelho, onde os neurônios de entrada seguem até uma saída, quanto a segunda etapa, *backpropagation* (*backward pass*), em azul, partindo da saída em direção às entradas.

Para realizar o treinamento deste modelo de RNA, é necessário a definição de parâmetros relacionados ao momento de parada e frequência de ajuste dos pesos. Segundo [BRAGA, LUDERMIR e CARVALHO \(2007\)](#), existem alguns critérios que são mais utilizados, como limite de ciclos a serem realizados, definição de parada após o erro médio atingir um nível pré-determinado ou após atingir um nível de acerto também pré-determinado.

2.3 FERRAMENTAS DE DESENVOLVIMENTO

Nesta seção serão explicitadas as ferramentas tecnológicas utilizadas para o desenvolvimento do trabalho, desde o ambiente de desenvolvimento e linguagem de programação até bibliotecas e métodos aplicados.

2.3.1 Jupyter Notebook

Ambientes de desenvolvimento com estruturas de *notebooks* são bastante utilizados em projetos de ciência de dados e inteligência computacional. Estes possuem estruturas em células, oferecendo a possibilidade de escrever e executar códigos, visualizar a saída logo abaixo, incluir notas, textos, imagens e observações ([BIEHLER; FLEISCHER, 2021](#)).

Um dos ambientes mais conhecidos com estrutura de *notebooks* é o Jupyter. Acrônimo indireto de três linguagens de programação (Julia, Python e R), é um aplicativo web que facilita diversas etapas necessárias ao se trabalhar com dados, como, por exemplo, a análise exploratória ([ARCANJO, 2022](#)). Na [Figura 15](#) é possível identificar um exemplo de uso deste ambiente de desenvolvimento.

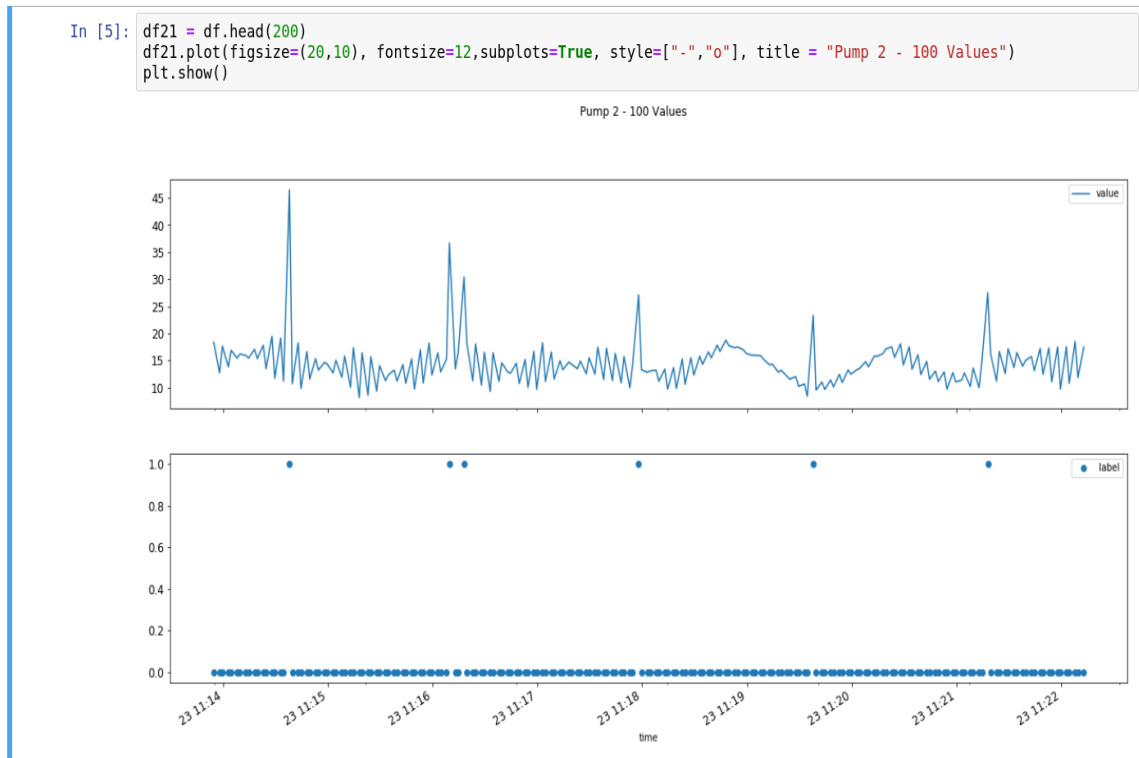


Figura 15 – Exemplo de utilização do Jupyter Notebook

Fonte: VERMA (2021)

É visto na [Figura 15](#) uma utilização do Jupyter Notebook para projetos com dados. Nesta, um grupo de informações agrupadas em um *dataframe* é exibida em dois gráficos distintos logo abaixo da célula em que o código é escrito, característica dos ambientes de desenvolvimento baseados em *notebooks*.

2.3.2 Python

Existem diversas linguagens de programação que podem ser utilizadas para trabalhar com dados e implementar projetos de inteligência computacional. Uma das mais famosas e mais utilizadas é o Python, orientada a objetos e de código aberto ([MONTEIRO, 2022](#)).

Ainda segundo [MONTEIRO \(2022\)](#), o Python tem como algumas vantagens ter o processamento em tempo de execução e ser uma linguagem interativa. Um dos pontos mais significativos é a vasta quantidade de bibliotecas e funcionalidades disponíveis para utilização por parte do programador, permitindo uma vasta gama de possibilidades a serem exploradas.

2.3.3 Bibliotecas utilizadas

Nesta seção serão explicitadas as bibliotecas da linguagem de programação Python utilizadas durante a execução do trabalho. Estas fornecem diversas funcionalidades essenciais a fim de facilitar o desenvolvimento de projetos de ciência de dados e inteligência computacional.

2.3.3.1 Pandas

A biblioteca Pandas é extremamente poderosa para manipulação e análise de dados. De código aberto, permite trabalhar com diversos tipos de informações, como dados tabulares e matrizes, por exemplo (MULINARI, 2022b). Desenvolvida em 2008 por Wes McKinney, tem como vantagens a fácil compreensão de código, utilização de poucos comandos para manipulação e associação com outras bibliotecas do Python como a Numpy e Matplotlib.

Os principais objetos de trabalho do Pandas são as Séries, matrizes unidimensionais com uma sequência de valores, e os DataFrames, estruturas tabulares com rótulos nas linhas e colunas. A partir disto, a biblioteca fornece funcionalidades capazes de trabalhar com estas informações (MULINARI, 2022b). Na Figura 16 é visto um exemplo simples de utilização desta biblioteca.

```
In [36]: import pandas as pd
left = pd.DataFrame({
    'id': [1,2,3,4,5],
    'Name': ['Jack', 'Amy', 'Elias', 'Young', 'Smith'],
    'subject_id': ['sub1', 'sub2', 'sub4', 'sub6', 'sub5']})
right = pd.DataFrame({
    'id': [1,2,3,4,5],
    'Name': ['Billy', 'Brooks', 'Brown', 'Aurier', 'Jose'],
    'subject_id': ['sub2', 'sub4', 'sub3', 'sub6', 'sub5']})
print (pd.merge(left, right, on='id'))
```

| | id | Name_x | subject_id_x | Name_y | subject_id_y |
|---|----|--------|--------------|--------|--------------|
| 0 | 1 | Jack | sub1 | Billy | sub2 |
| 1 | 2 | Amy | sub2 | Brooks | sub4 |
| 2 | 3 | Elias | sub4 | Brown | sub3 |
| 3 | 4 | Young | sub6 | Aurier | sub6 |
| 4 | 5 | Smith | sub5 | Jose | sub5 |

Figura 16 – Exemplo de utilização da biblioteca Pandas

Fonte: S (2022)

É possível visualizar na [Figura 16](#) uma utilização da biblioteca Pandas para manipulação de dados. Nesta, são criados dois *dataframes* que são unidos em um único, através do método de *merge*.

2.3.3.2 NumPy

O pacote Numpy, criado em 2005 por Travis Oliphant, teve como objetivo reunir a comunidade em torno de um único *framework* de processamento de vetores. Esta biblioteca, de código aberto, é destinada a realizar operações matemáticas em *arrays* multidimensionais ([MULINARI, 2022a](#)).

Utilizada majoritariamente em análise de dados, o Numpy oferece funcionalidades para tratamento, limpeza e manipulação. Como vantagens, esta biblioteca ocupa pouca memória, sendo mais veloz e trazendo uma facilidade na execução de cálculos números ([MULINARI, 2022a](#)).

2.3.3.3 Plotly

Funcionando como uma biblioteca que fornece ferramentas de visualização gráfica de dados, o Plotly também é uma biblioteca de código aberta disponível para a linguagem de programação Python. Utilizando uma base de dados, é possível gerar gráficos de barra, área, histogramas, mapas de calor, *scatter plots* e até animações ([PLOTLY, 2022](#)).

A utilização de funções do Plotly permite uma melhor visualização e entendimento do conjunto de dados trabalhado, comum em etapas de um projeto de ciência de dados como a análise exploratória.

2.3.3.4 Matplotlib

O pacote Matplotlib, também disponível para o Python, assim como o Plotly, possui funções gráficas de plotagem de gráficos. Sua utilização é comum para criação de gráficos bidimensionais, podendo ter formato de barra, linha, pizza, histograma, dentre outros ([MATPLOTLIB, 2022](#)). Esta biblioteca é comumente importada para o código junto ao pacote Pyplot, que permite alterar as imagens geradas.

2.3.3.5 Seaborn

O Seaborn é uma biblioteca utilizada principalmente para plotagens estatísticas em Python. Baseada na Matplotlib, fornece maiores poderes de customização dos gráficos, tornando-os mais amigáveis visualmente ([SEABORN, 2022b](#)). Na [Figura 17](#) é possível visualizar uma aplicação desta biblioteca em um conjunto de dados.

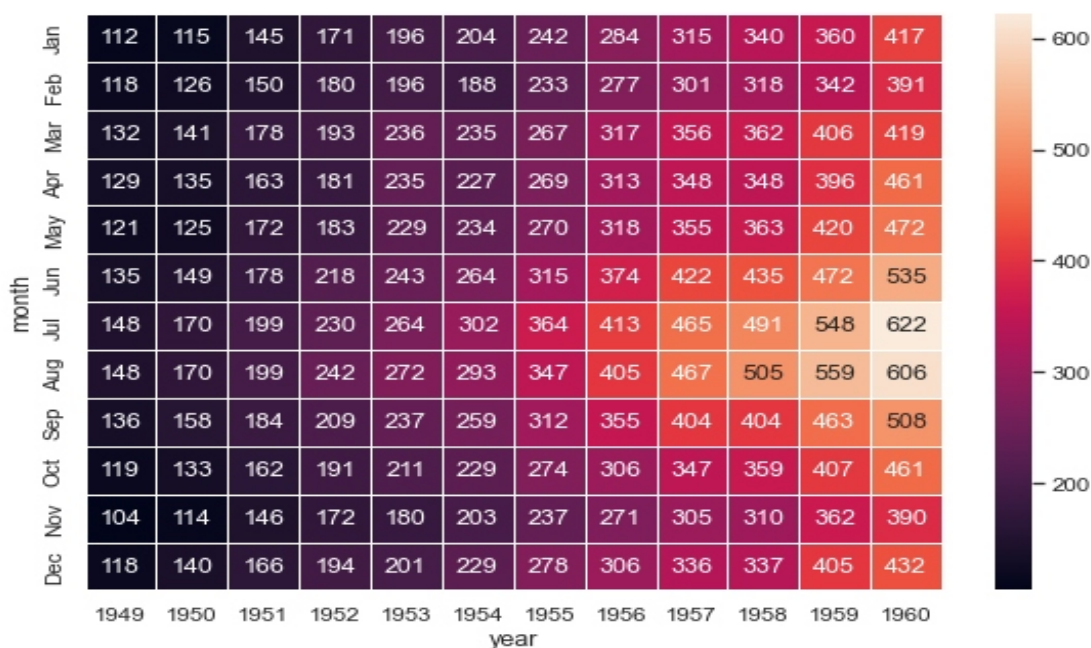


Figura 17 – Exemplo de utilização da biblioteca Seaborn

Fonte: [Seaborn \(2022a\)](#)

Na [Figura 17](#) é visto um exemplo de utilização da biblioteca Seaborn em um conjunto de dados do número de passageiros em vôos americanos, onde as colunas dizem respeito aos anos e as linhas representam os meses onde as viagens ocorreram. Do lado direito, é possível identificar uma guia de identificação de cores, onde, quanto mais próximo do número 600, mais próximo da cor branca o campo dentro do gráfico estará.

2.3.3.6 Scikit-Learn

O pacote Scikit-Learn é utilizado para tarefas de aprendizado de máquina. Gratuita e de código aberto, esta biblioteca é essencial em projetos de modelagem estatística, análise e mineração de dados, sendo extremamente versátil e possuindo interação com os demais pacotes do Python, como Matplotlib, Numpy e Pandas ([AWARI, 2022](#)).

Esta biblioteca fornece uma ampla variedade de algoritmos de *machine learning* integrados que podem ser aplicados em poucas linhas de código. Modelos de classificação, regressão, agrupamento, pré-processamento, dentre outros, são englobados pelo Scikit-Learn. Podem ser implementados de maneira rápida e prática métodos *ensemble*, como *AdaBoost Classifier*, *Gradient Boosting Machine* e *Random Forest Classifier*, além de modelos como *Dummy Classifier*, árvores de decisão, *KNN Classifier*, regressão logística, *Naive Bayes*, *Support Vector Machine*, redes neurais artificiais, dentre outros ([SCIKIT-LEARN, 2022d](#)).

É possível utilizar os modelos de *machine learning* implementados pelo Scikit-Learn para tarefas como cálculo de *feature importances*. Modelos como *Extra Trees Classifier*, árvores de decisão e regressão logística possuem métodos que permitem encontrar o quão uma determinada variável influencia em predições realizadas (BROWNLEE, 2022).

A biblioteca Scikit-Learn também fornece diversas ferramentas para auxílio de pré-processamento de dados, uma das etapas mais importantes e determinante no resultado final. Através do pacote *preprocessing*, é possível utilizar, por exemplo, o método de *Standard Scaler*, que realiza a normalização de um conjunto de informações (SCIKIT-LEARN, 2022c).

O pacote *model selection* traz funcionalidades como a *train test split*, onde, a partir de um conjunto de dados, separa de forma aleatória em grupos de treino e teste, a fim de serem utilizados em modelos de inteligência computacional (SCIKIT-LEARN, 2022b).

Além desta, o *model selection* também possibilita realizar otimização de hiperparâmetros de modelos. Cada algoritmo de *machine learning* é construído através de hiperparâmetros, e esta biblioteca é capaz de, utilizando uma variedade de opções para cada um, testar as possibilidades de combinações a fim de encontrar aquela que fornece os melhores desempenhos.

Este processo de otimização pode ser realizado através dos métodos de *Grid Search* e *Random Search*. O primeiro, implementado pela função *GridSearchCV*, realiza uma busca profunda, testando todas as combinações possíveis. Dependendo do número de possibilidades, pode se tornar um processo demorado e com alto custo computacional. Já o segundo, implementado pela função *RandomSearchCV*, possui um número definido de execuções, onde são testadas combinações aleatórias e, ao fim, a melhor delas é escolhida. Ambos realizam processo de validação cruzada (SCIKIT-LEARN, 2022e).

Por fim, existe também o pacote *metrics*, que traz funcionalidades relacionadas à métricas de avaliação, como o *classification report*, que mostra de forma resumida os resultados e métricas de avaliação de uma execução de um modelo de *machine learning* e a matriz de confusão, que também auxilia nesta visualização de desempenho, além de ser responsável pelos cálculos destas métricas, como a acurácia, *recall* e *f1-score*, dentre outras (SCIKIT-LEARN, 2022a).

2.3.3.7 PyCaret

A biblioteca PyCaret, gratuita e de código aberto, funciona como uma forma de automatizar processos de *machine learning*, funcionando como uma alternativa *low-code*, ou seja, pouco código é utilizado para gerar resultados. Esta permite comparar diversos modelos de inteligência computacional de forma rápida e eficiente (PYCARET, 2022).

3 METODOLOGIA

Neste capítulo é apresentada a metodologia adotada neste trabalho, a fim de atingir os objetivos previamente estabelecidos. Serão descritas as etapas executadas a fim de realizar uma avaliação de modelos de aprendizado de máquina que sejam capazes de pré-selecionar empresas do mercado financeiro brasileiro. Os resultados foram avaliados visando uma posterior utilização em aplicações voltadas ao investidor iniciante.

Este trabalho foi baseado em dados históricos de cotações, balanços patrimoniais trimestrais e demonstrativos de resultados, todos coletados de maneira manual em plataformas gratuitas e armazenados localmente. Cada arquivo obtido possuía formato de tabela, com suas devidas informações organizadas em linhas e colunas que, posteriormente, foram analisadas e tratadas.

Todas as etapas do trabalho foram realizadas para servir como um guia para posterior utilização em aplicações reais. Com isso, foi definido uma estrutura de pré-processamento a ser seguida, a fim de tratar os dados e tornar possível a entrada destes nos modelos de *machine learning*. Em sequência, foram realizados processos de análise exploratória e seleção de dados, essenciais para uma melhor execução das etapas seguintes.

Com os dados devidamente estruturados e tratados, foi possível realizar a aplicação de diversos modelos de inteligência computacional e, assim, avaliá-los conforme critérios definidos na [seção 4.5](#). Cada modelo foi treinado para que pudesse classificar um determinado trimestre de um ativo como *buy*, *hold* ou *sell*. Ao final desta etapa, foram selecionados os cinco modelos mais bem avaliados para que pudessem ser otimizados e, posteriormente, pudessem ser submetidos à simulações com dados reais.

Nesta etapa de simulações, cada modelo recebeu como entrada os valores já relativizados, conforme o pré-processamento estabelecido, do balanço patrimonial e demonstrativo de resultados de um trimestre específico, devendo classificá-lo como compra, venda ou *hold*, visando o trimestre seguinte. Todas as empresas especificadas no trabalho foram avaliadas e as classificadas como *buy* foram adicionadas a uma carteira de investimentos fictícia. Foi avaliado, então, o tamanho e o rendimento desta carteira no período de um ano, a fim de definir qual modelo obteve o melhor desempenho, cumprindo assim os objetivos estabelecidos pelo trabalho.

Tendo em mãos o desempenho dos diferentes modelos de inteligência computacional será possível, em trabalhos futuros, a automatização do processo de coleta e pré-processamento dos dados e a integração do melhor modelo encontrado numa aplicação

disponível aos investidores, com foco em iniciantes.

Neste capítulo serão descritas as etapas realizadas na execução deste projeto. De forma geral, pode-se visualizar o fluxograma de etapas de desenvolvimento na [Figura 18](#).

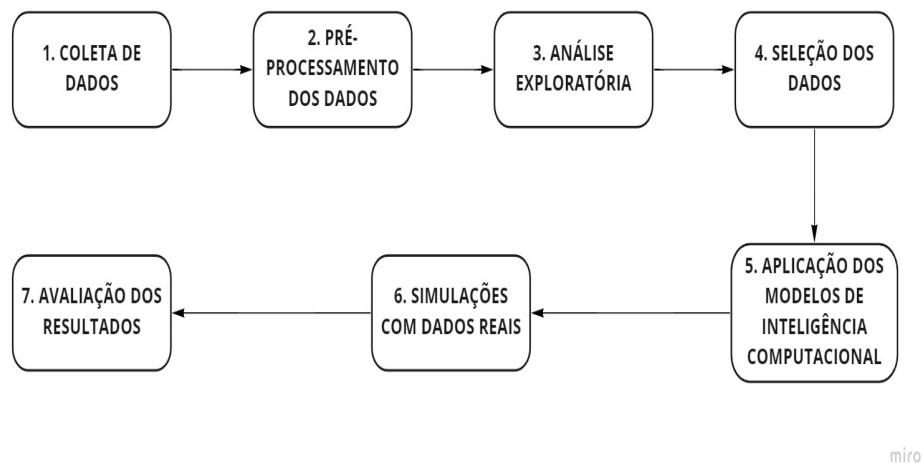


Figura 18 – Etapas de desenvolvimento do projeto

Fonte: Elaborado pelo autor.

Este fluxograma apresentado na [Figura 18](#) mostra, de forma mais ampla, cada passo metodológico executado, descritos de forma mais detalhada ao longo deste capítulo.

3.1 COLETA DE DADOS

Para a realização do trabalho foi necessário realizar a coleta dos dados, onde foram selecionados os portais Fundamentus e Yahoo! Finanças como base de informações para o projeto. O primeiro fornece o histórico de balanços patrimoniais trimestrais e de demonstrativos de resultados de todas as empresas disponíveis na B3, enquanto, com o segundo, foi possível obter o histórico de cotações destas empresas, com informações como preço de abertura, fechamento e volume de ações listadas.

3.2 PRÉ-PROCESSAMENTO DOS DADOS

Com os dados devidamente em mãos, o passo seguinte foi o de pré-processamento. Foram feitos processos de limpeza e transformação, como definição de uma janela de tempo a ser analisada, inserção de alguns indicadores fundamentalistas junto aos dados dos balanços patrimoniais trimestrais e demonstrativos de resultados, adição dos setores

de cada empresa, além da criação de rótulos classificatórios como *buy*, *hold* e *sell*. Na Figura 19, é possível identificar uma representação da sequência de passos realizadas nesta etapa.

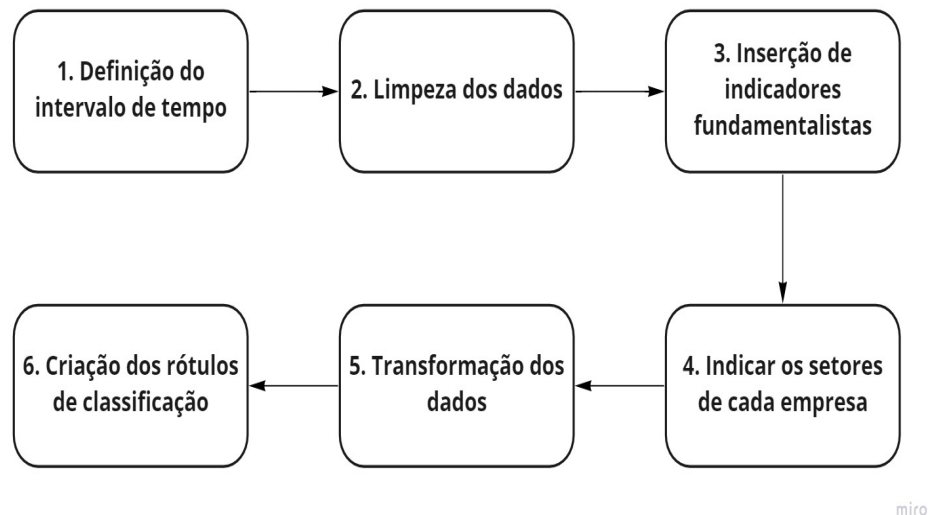


Figura 19 – Etapas de pré-processamento dos dados

Fonte: Elaborado pelo autor.

O processo de pré-processamento se deu através destas seis etapas vistas na Figura 19. Cada uma destas foi executada em cima dos dados obtidos no processo de coleta, a fim de torná-los utilizáveis posteriormente nos modelos de inteligência computacional.

3.3 ANÁLISE EXPLORATÓRIA

Em sequência, para um melhor entendimento das informações após todo o pré-processamento realizado, também foi feita uma análise exploratória, identificando questões como correlação de Pearson entre os dados. Esta etapa permite uma melhor perspectiva sobre os dados e o que esperar em resultados futuros.

3.4 SELEÇÃO DE DADOS

Na seleção dos dados, foi feito um processo de *feature selection* a fim de otimizar o projeto futuramente. Estes dados também foram normalizados e separados em treino e teste. Esta etapa é de extrema importância para a qualidade dos resultados, visto que impacta diretamente no desempenho dos modelos e no tempo e carga de processamento de cada um.

3.5 APLICAÇÃO DE MODELOS DE INTELIGÊNCIA COMPUTACIONAL

Em uma quinta etapa, foram aplicados diversos modelos de inteligência computacional a fim de entender o funcionamento dos mesmos no contexto do trabalho. Esta etapa foi dividida em três passos: o primeiro, testando modelos em seu formato mais genérico; o segundo, utilizando uma biblioteca de AutoML, ou *machine learning* automatizado, para complementar o primeiro passo; por fim, uma otimização de hiperparâmetros dos cinco modelos com melhor performance nos dois primeiros passos. Na [Figura 20](#) é possível identificar os passos executados nesta etapa da metodologia.



miro

Figura 20 – Etapas de aplicação dos modelos

Fonte: Elaborado pelo autor.

Cada etapa explicitada na [Figura 20](#) ocorreu de maneira sequencial, sendo essencial para o desenvolvimento do trabalho. Visto a pouca quantidade de trabalhos relacionados a este tema, os passos um e dois foram essenciais para um maior entendimento do funcionamento dos modelos no contexto do trabalho. O terceiro passo se dá através da análise dos desempenhos de cada modelo, onde apenas os cinco melhores de acordo com critérios definidos posteriormente na [seção 4.5](#). Já o quarto e último passo desta seção se dá pela otimização dos hiperparâmetros dos modelos selecionados, visando melhorar seus resultados.

3.6 SIMULAÇÕES COM DADOS REAIS

Com os cinco melhores modelos otimizados, foi realizada uma simulação com dados reais para verificar o verdadeiro desempenho dos mesmos. Nesta etapa, cada modelo fez uma pré-seleção de empresas no primeiro trimestre de março de 2021, gerando possíveis opções de investimento visando o trimestre seguinte, ou seja, junho de 2021.

Tendo carteiras fictícias formadas pelas empresas pré-selecionadas para o trimestre de junho de 2021, foi analisado o desempenho destas no intervalo de 1 ano, ou seja, até o mês de junho de 2022. Com isso, é possível verificar se os modelos são válidos na avaliação de empresas visando o longo prazo.

3.7 AVALIAÇÃO DOS RESULTADOS

Na última etapa da metodologia, é realizada a análise dos resultados obtidos nas simulações. O desempenho de cada modelo nas simulações propostas é comparado a fim de entender os resultados obtidos e, posteriormente, verificar se os objetivos previamente definidos foram atingidos.

4 DESENVOLVIMENTO

Neste capítulo serão detalhadas todas as etapas realizadas no trabalho, seguindo a metodologia explicitada no [Capítulo 3](#). Serão explicitadas desde a coleta dos dados e pré-processamento até as simulações finais realizadas.

Para o desenvolvimento do trabalho, foi escolhida a linguagem de programação Python, característica de projetos de ciência de dados, visto que possui diversas bibliotecas e tecnologias, como Pandas e NumPy, por exemplo, que enriqueceram e facilitaram o trabalho. Também foi levada em consideração a experiência do autor com a linguagem para esta decisão.

Como ambiente de desenvolvimento, foi utilizado o Jupyter Notebook. Ambientes com este formato, dividido por células, facilitam a visualização de informações e o próprio desenvolvimento em trabalhos deste tipo. Além de rápido e de melhorar a disposição de informações e organização de código, a possibilidade de compilação por partes foi essencial para o andamento do trabalho. Outra vez, a experiência do autor com o ambiente também foi levada em consideração nesta decisão.

4.1 COLETA DE DADOS

A etapa inicial do trabalho foi a de coleta dos dados. Para isso, foram selecionados dois portais como base para fornecê-los: o Fundamentus, onde foram obtidos os balanços patrimoniais trimestrais e demonstrativos de resultados de todas as empresas listadas na B3, e o Yahoo! Finanças, que possibilitou a coleta do histórico de cotações, bem como volume de ações negociadas por ativo, dentre outras informações.

Ambos os portais foram escolhidos devido às suas facilidades de aquisição das informações e de estarem disponíveis de forma gratuita. Além disso, são bem conhecidos e recomendados por entusiastas do mercado financeiro, fornecendo dados precisos e seguros.

Os dados foram coletados e armazenados localmente em agosto de 2021. Na [Figura 21](#) é possível visualizar a estrutura do arquivo de balanços patrimoniais trimestrais e demonstrativos de resultados disponibilizados no portal Fundamentus, onde é mostrada uma parte destas informações do ativo YDUQ3, representando à empresa educacional YDUQS.

| | 30/06/2021 | 31/03/2021 | 31/12/2020 | 30/09/2020 | 30/06/2020 | 31/03/2020 | 31/12/2019 | 30/09/2019 | 30/06/2019 |
|--|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| YDUQ3 | | | | | | | | | |
| Ativo Total | 9641204.736 | 9642667.008 | 9265265.664 | 9411001.344 | 9294304.256 | 7635795.968 | 5512492.032 | 5740264.96 | 5564189.184 |
| Ativo Circulante | 3075841.024 | 3067963.904 | 2736397.056 | 2994115.072 | 3047205.12 | 3332005.12 | 1475683.968 | 1664539.008 | 1618936.96 |
| Caixa e Equivalentes de Caixa | 1212306.944 | 1278475.008 | 28407 | 25176 | 20415 | 10392 | 12251 | 10071 | 19436 |
| Aplicações Financeiras | 755078.016 | 771812.992 | 1604868.992 | 1895389.056 | 1886955.008 | 2535168 | 596860.992 | 855699.008 | 698840 |
| Contas a Receber | 964782.976 | 863148.992 | 890150.976 | 873300.992 | 955827.968 | 641708.992 | 759622.016 | 714601.024 | 813102.976 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| IR Diferido | -1472 | 15187 | 37923 | 17439 | 49837 | 27825 | -5763 | 13165 | -7007 |
| Participações/Contribuições Estatutárias | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN |
| Reversão dos Juros sobre Capital Próprio | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN |
| Part. de Acionistas Não Controladores | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Lucro/Prejuízo do Período | 116465 | 43225 | -102641 | 112476 | -79542 | 167888 | 58059.04 | 152511.008 | 194772.992 |

Figura 21 – Estrutura do arquivo de balanços patrimoniais trimestrais e demonstrativos de resultados do ativo YDUQ3

Fonte: [Fundamentus...](#) (2021)

Pode-se observar a organização do arquivo coletado, onde os dados recentes se encontram mais à esquerda. Outro ponto importante é a presença de campos não acessíveis ou vazios, do tipo "NaN" (*not a number*), significando que não é um número, que foram posteriormente tratados.

Na [Figura 22](#) é mostrado um exemplo da estrutura dos dados de cotação coletados no Yahoo! Finanças, sendo possível visualizar a estrutura de um arquivo coletado do ativo ABEV3, referente à fabricante de bebidas AMBEV.

| | Date | Open | High | Low | Close | Adj Close | Volume |
|-----|------------|-----------|-----------|-----------|-----------|-----------|------------|
| 0 | 2016-01-04 | 17.730000 | 17.730000 | 17.209999 | 17.209999 | 14.775684 | 13206900.0 |
| 1 | 2016-01-05 | 17.250000 | 17.520000 | 17.110001 | 17.480000 | 15.007492 | 10774200.0 |
| 2 | 2016-01-06 | 17.360001 | 17.480000 | 17.200001 | 17.309999 | 14.861543 | 7739100.0 |
| 3 | 2016-01-07 | 17.170000 | 17.320000 | 16.850000 | 16.850000 | 14.466606 | 15316400.0 |
| 4 | 2016-01-08 | 16.930000 | 17.200001 | 16.930000 | 17.070000 | 14.655489 | 10684000.0 |
| ... | ... | ... | ... | ... | ... | ... | ... |

Figura 22 – Estrutura de cotações históricas do ativo ABEV3

Fonte: [Yahoo!...](#) (2021)

Ao interpretar a estrutura dos dados mostrada na [Figura 22](#), é possível identificar que os dados nas primeiras linhas são os mais antigos, datando de 2016, portanto, estão demonstrados em ordem cronológica. Cada linha diz respeito a uma data, enquanto as demais colunas dizem respeito às cotações e informações como preço de abertura, fechamento e volume de ações negociadas no dia.

É importante destacar a quantidade de ativos disponibilizados por estes portais. Como ambos armazenam o histórico completo, muitos ativos que já não estão mais disponíveis na B3 estão presentes nos dados, sendo necessário removê-los na etapa de pré-processamento. Informações de um total de 898 papéis foram coletados nesta primeira etapa.

4.2 PRÉ-PROCESSAMENTO DOS DADOS

A etapa de pré-processamento dos dados é essencial para projetos de ciência de dados e inteligência computacional. Nesta, foi possível repaginar os dados a fim de otimizar os resultados finais posteriormente.

4.2.1 Definição do intervalo de tempo

Inicialmente, foi definido o intervalo de tempo a ser tratado no trabalho. A fim de descartar dados muito antigos que não representam o cenário atual do mercado financeiro brasileiro e poderiam impactar nos resultados, dados anteriores ao ano de 2016 foram descartados. Como os dados foram coletados em agosto de 2021, tanto os balanços patrimoniais trimestrais e demonstrativos de resultados como as cotações históricas se concentraram num intervalo de cinco anos para o desenvolvimento do projeto.

Com a definição do intervalo de tempo em que o trabalho está focado, muitas empresas que já não se encontram disponíveis na B3 foram descartadas, visto que não possuíam balanços, DREs ou cotações neste intervalo de 5 anos. Após este primeiro descarte, manteve-se um total de 427 ativos disponíveis para análise. Ou seja, mais de 50% dos papéis coletados na primeira etapa do trabalho foram descartados somente com esta definição, o que corrobora para a decisão tomada, visto que empresas que já não existem ou fecharam seu capital seriam maioria dentre os dados coletados.

4.2.2 Limpeza dos dados

Nesta seção do pré-processamento, foram analisados mais a fundo os dados restantes para que pudessem ser tratados. Inicialmente, no arquivo de cotações, foi identificado que haviam muitas datas com valor vazio e, após verificações, entendeu-se que eram feriados e finais de semana, onde não há abertura do mercado financeiro. A fim de evitar

complicações durante o trabalho, estas datas especiais foram preenchidas com dados de cotação do dia anterior.

Também foram tratadas questões do tipo dos dados disponibilizados. Nos arquivos de cotação, dados de datas estavam no formato "*object*", enquanto nos dados dos balanços patrimoniais trimestrais e DREs, estavam no formato "*datetime64*". Para não haver conflito, todas as datas foram transformadas para o formato "*datetime64*".

Após esta alteração, pode-se verificar que, para cada empresa existiam duas tabelas, uma com dados dos balanços patrimoniais e DREs, e outra com as variáveis de cotações históricas. Foi realizada uma junção destes dados em uma tabela única para cada ativo, possuindo todas as informações necessárias.

Ainda na limpeza dos dados, foram identificadas algumas empresas que não seguiam o padrão dos arquivos de balanços patrimoniais trimestrais e DREs, o que prejudicaria a aplicação destas nos modelos de inteligência computacional em etapas futuras. Como o número de empresas com estruturas distintas era relativamente pequeno com relação às demais, cerca de 9% do total, optou-se por removê-las da análise a fim de facilitar o processo de desenvolvimento. Ao final desta etapa, é possível constatar que existem 387 ativos de empresas disponíveis para a sequência do trabalho.

Outro ponto analisado foram as colunas dos dados coletados. Ao analisar os balanços patrimoniais e os DREs de forma conjunta, existem colunas comuns aos dois. Portanto, foi necessário tratar esta condição a fim de evitar conflitos. Na [Figura 23](#) é possível este processo e seu resultado.

```
In [33]: for empresa in dicionario_final:
          dicionario_final[empresa].columns = colunas

          display(dicionario_final["ABEV3"])
```

| | Ativo Total | Ativo Circulante | Caixa e Equivalentes de Caixa | Aplicações Financeiras | Contas a Receber_1 | Estoques_1 |
|-------------------|---------------|------------------|-------------------------------|------------------------|--------------------|-------------|
| ABEV3 | | | | | | |
| 2021-06-30 | 124440133.632 | 32705665.024 | 13269346.304 | 1245607.04 | 3702152.96 | 9583373.312 |
| 2021-03-31 | 133417828.352 | 37059858.432 | 17286068.224 | 2049628.032 | 3357889.024 | 9698229.248 |
| 2020-12-31 | 125196574.72 | 35342614.528 | 17090335.744 | 1700028.032 | 4303137.792 | 7605904.896 |
| 2020-09-30 | 127056781.312 | 39098793.984 | 21660450.816 | 1442923.008 | 4156922.88 | 7341836.8 |

Figura 23 – Tratamento de colunas nos dados de balanços e DREs

Fonte: Elaborado pelo autor.

Na parte superior da [Figura 23](#) é visto o código desenvolvido para este processo de renomeação de colunas. Abaixo, o resultado da execução deste código, podendo ser visualizadas algumas colunas que foram alteradas como "Estoques_1" e "Contas a receber_1".

Ao tratar as colunas dos dados, também foi identificado uma grande quantidade de informações vazias dentro dos arquivos. Colunas como "Receita Bruta de Vendas e/ou Serviços", "Deduções da Receita Bruta" e "Reversão dos Juros sobre Capital Próprio" apresentaram grande quantidade de campos sem informação nenhuma. De um total de 7411 campos, estas apresentaram mais de 7400 vazios, sendo, portanto, descartadas do processo.

4.2.3 Inserção dos indicadores fundamentalistas

A fim de aumentar a robustez das informações e visando resultados futuros, foram escolhidos alguns indicadores fundamentalistas para serem adicionados aos dados dos balanços patrimoniais e DREs. Os indicadores escolhidos foram os seguintes:

- P/L (Preço / Lucro);
- P/VPA (Preço / Valor patrimonial por ação);
- DY (Dividend Yield);
- Payout;
- LPA (Lucro por ação).

Cada indicador foi adicionado de maneira matemática, ou seja, via cálculo com as informações disponibilizadas nos balanços patrimoniais e DREs, além das cotações históricas.

4.2.4 Indicação dos setores de cada empresa

Ao realizar a coleta dos balanços patrimoniais e DREs, não é obtido de forma explícita os setores de atuação de cada empresa. Para isso, foi extraído, também do portal Fundamentus, os setores a que cada uma pertence e adicionados aos dados. Cada ativo foi classificado como um dos seguintes setores:

- Bens industriais;
- Comunicações;

- Consumo cíclico;
- Consumo não-cíclico;
- Financeiro;
- Materiais básicos;
- Outros;
- Petróleo, gás e biocombustíveis;
- Saúde;
- Tecnologia da informação;
- Utilidade pública.

Ao classificar cada empresa em seu determinado setor, foi adicionado também, para cada trimestre de uma determinada empresa, a média das cotações de todas as empresas atuantes daquele setor. Ou seja, na linha de dados do primeiro trimestre de 2021 da Petrobras, foi adicionada a média de cotações de todas as empresas que atuam no setor "Petróleo, gás e biocombustíveis", por exemplo.

4.2.5 Transformação dos dados

Para que os dados pudessem ser trabalhados pelos modelos de inteligência computacional posteriormente, foi necessário realizar algumas alterações nos mesmos. Informações absolutas precisavam ser modificadas para demonstrar variações, ou seja, precisavam ser relativizadas.

Tendo em vista o contexto do trabalho, que propõe-se a realizar uma seleção de empresas visando o futuro, foi necessário encontrar uma estratégia de relativização dos dados a fim de atender este objetivo. Os dados dos balanços patrimoniais trimestrais e DREs foram alterados, então, utilizando informações do passado, ou seja, ao invés de representarem a informação absoluta de um trimestre, estará representando a variação desta informação do trimestre anterior para o atual. Por exemplo, o "Ativo total" do segundo trimestre de 2020 será alterado para a variação deste campo entre o primeiro e o segundo trimestre de 2020.

Com relação às cotações históricas, esta análise foi feita objetivando o futuro, ou seja, o trimestre seguinte. Cada dado de cotação foi alterado para a variação entre o trimestre atual e o seguinte, a fim de identificar a variação futura. Por exemplo, a cotação do segundo trimestre de 2020 será alterada para a variação deste campo entre o segundo e

o terceiro trimestre de 2020. Uma representação desta relativização dos dados pode ser vista na [Figura 24](#).

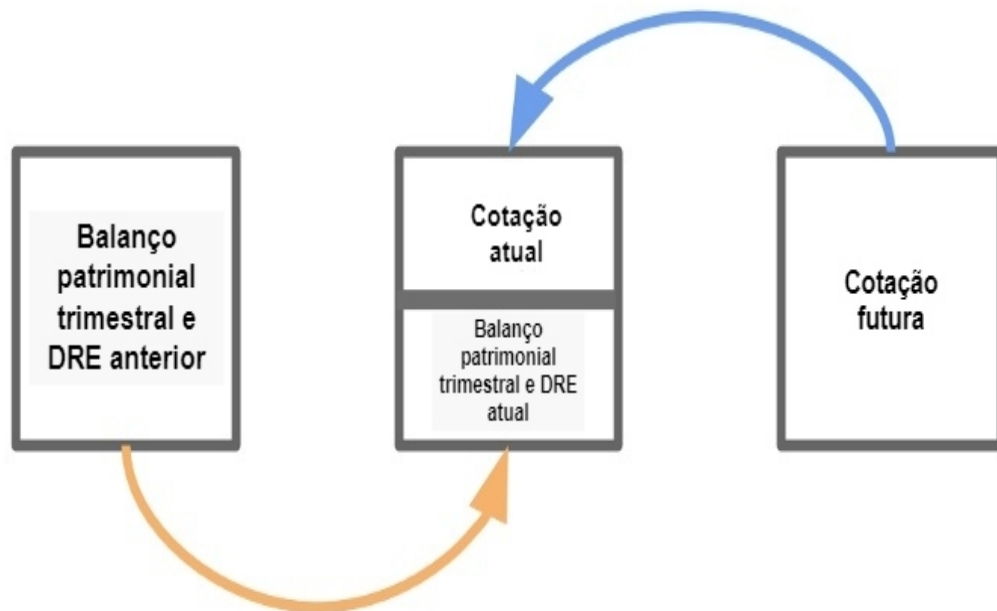


Figura 24 – Ilustração da relativização dos dados existentes

Fonte: Adaptado de [SANTOS \(2020a\)](#)

O trimestre atual, representado no centro da [Figura 25](#), terá dados relativizados tanto com o passado quanto com o futuro. Os dados fundamentalistas, balanços patrimoniais e DREs, serão transformados na variação do trimestre anterior para o atual, como é mostrado pela linha laranja, na parte inferior da imagem. Já os dados de cotações (preços) serão transformados na variação do trimestre atual para o seguinte, como é mostrado pela linha azul, na parte superior da figura.

4.2.6 Criação dos rótulos de classificação

Após a relativização dos dados realizada, foi possível criar os rótulos de classificação necessários. Para cada trimestre de cada empresa, foi criada uma coluna chamada "Decisão", abrigando o valor em que aquele trimestre foi classificado. Esta decisão foi tomada de acordo com a comparação entre a variação da cotação da empresa em comparação à média de variação das cotações de todas as empresas do mesmo setor de atuação desta. Para isto, foi definido um valor mínimo de 3%, para servir de referência nesta etapa.

Com isso, foram analisados todos os trimestres de todas as empresas disponíveis. Se a variação de cotação da empresa for superior a 3% em comparação às empresas de mesmo setor, então ela será classificada como compra, ou *buy*, significando que esta teve uma variação de cotação maior do que as empresas que atuam no mesmo setor e, portanto, deveria ter sido comprada anteriormente. Já se a variação de cotação da empresa for 3% menor, ou abaixo deste valor, em comparação às empresas de mesmo setor, então ela será classificada como venda, ou *sell*, significando que esta teve uma variação de cotação abaixo das demais de mesmo campo de atuação, portanto, deveria ter sido vendida anteriormente.

Em resumo, os rótulos de classificação dados a cada trimestre foram os seguintes:

- *Buy* (Valor 2): se a cotação foi igual ou superou 3% com relação às demais empresas de mesmo setor;
- *Sell* (Valor 0): se a cotação foi menor em 3% ou menos com relação às demais empresas de mesmo setor;
- *Hold* (Valor 1): se a cotação, com relação às demais empresas de mesmo setor, ficar entre -3% e 3%.

É importante destacar que o valor de 3% utilizado não é otimizado, podendo ser alterado futuramente em outras aplicações. Este valor foi utilizado por SANTOS (2020a) ao trabalhar com ações do mercado financeiro americano. Na seção 6.1 é citada esta possibilidade para trabalhos futuros, onde poderia ser encontrado, por meio de testes, o valor ideal para a bolsa de valores brasileira.

Após todo o processo de transformação, os dados estão devidamente preparados para melhores análises. Na Figura 25 é mostrada uma parte dos dados após este processo, exemplificando com o ativo ABEV3, da empresa AMBEV.

| | Ativo Total | Ativo Circulante | Caixa e Equivalentes de Caixa | Aplicações Financeiras | Decisao |
|----------------|----------------|---------------------|-------------------------------------|---------------------------|---------|
| 2016- 09-30 | 0.032006 | 0.077012 | 0.271493 | 0.040486 | 0 |
| 2016- 12-31 | 0.059805 | 0.194882 | 0.081210 | 0.029842 | 2 |
| 2017- 03-31 | -0.037064 | -0.108406 | -0.082087 | -0.964979 | 1 |
| 2017- 06-30 | 0.041991 | 0.081957 | 0.211207 | -0.114410 | 2 |

Figura 25 – Exemplo de parte dos dados da empresa AMBEV

Fonte: Elaborado pelo autor.

Na [Figura 25](#) é visto o resultado final do processo de pré-processamento dos dados, exemplificado com uma parte da tabela final da empresa AMBEV. Nesta, é possível identificar as linhas em ordem cronológica, representando os trimestres nos quais os balanços patrimoniais e DREs foram divulgados. As colunas possuem seus dados relativizados conforme explicitado na [subseção 4.2.5](#). Já a coluna mais a direita representa a decisão, ou seja, o rótulo de classificação para aquele trimestre, definida de acordo com a [subseção 4.2.6](#).

Por fim, tendo todas as empresas com dados limpos e transformados, foi necessário juntar os dados em apenas uma tabela, a fim de desassociar as empresas e trabalhar apenas com os dados e, assim, poder trabalhá-los nos modelos de inteligência computacional. Ao realizar este processo, foi obtida uma tabela com todas as informações dos balanços patrimoniais e DREs já relativizados de todas as empresas disponíveis com suas determinadas classificações, indicadores e cotações. Ao final, obteve-se uma tabela com 78 colunas e 6143 linhas.

4.3 ANÁLISE EXPLORATÓRIA

Nesta seção, é apresentada uma rápida análise exploratória dos dados após todos os processos de pré-processamento. Esta etapa permite um melhor entendimento do estado atual das informações e em como podemos trabalhá-las e otimizá-las.

Utilizando bibliotecas do Python como a Matplotlib, Plotly e Seaborn, foi possível

ter uma melhor visualização dos dados tratados. Primeiramente, foi verificado a quantidade de cada rótulo de classificação realizado no processo de pré-processamento. Na [Figura 26](#), é mostrada uma representação da quantidade de cada rótulo adotado no pré-processamento.

Análise exploratória sobre as classificações realizadas

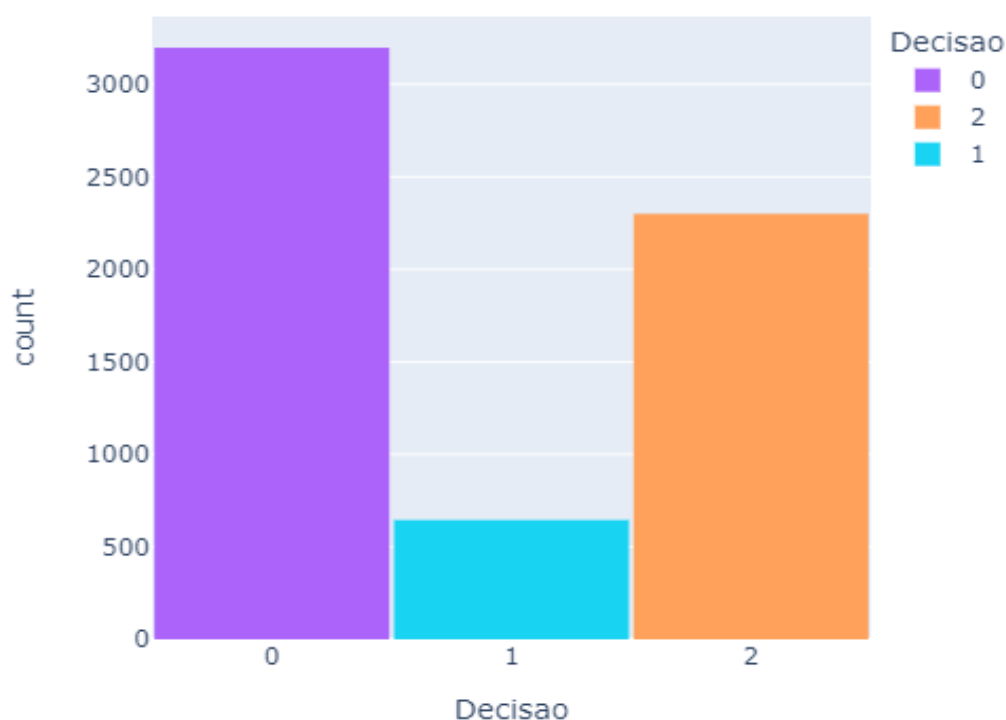


Figura 26 – Quantidade de cada rótulo realizada na classificação

Fonte: Elaborado pelo autor.

Pode-se observar na [Figura 26](#) que a classificação de venda, ou "*sell*", com valor zero, foi a que ocorreu em maior quantidade, em cerca de 52% dos dados. Já o rótulo "*hold*", com valor um, apareceu somente em cerca de 10% dos resultados.

Também foi verificado a correlação de Pearson entre as colunas existentes. Dados que possuem uma alta correlação entre si podem prejudicar a análise, visto que representam praticamente as mesmas informações. Para visualizar que haviam altas correlações entre os dados, foi analisado um mapa de calor, representado na [Figura 27](#).



Figura 27 – Análise de correlação de Pearson entre as colunas

Fonte: Elaborado pelo autor.

É possível observar na [Figura 27](#) ao guiar-se pela barra lateral que, quanto mais forte a cor amarela, maior a correlação entre as colunas, e quanto mais próximo ao branco,

mais negativo é o valor. Tanto valores muito altos como muito baixos não são interessantes de se manter para análises futuras.

Após esta análise, optou-se por descartar uma entre um par de colunas com correlação de Pearson muito alta ou muito baixa entre si. Como os valores máximo e mínimo são 1 e -1 respectivamente, foram escolhidos pares com uma correlação acima de 0.8 e abaixo de -0.8, onde uma de cada par foi descartada. Ao final deste processo, por exemplo, as colunas "Ativo Total" e "Ativo Circulante" possuíam correlação de Pearson de 0.9999998435868556 entre si, onde optou-se por eliminar a segunda. Ao final desta etapa, verificou-se que 27 colunas foram descartadas, restando um total de 51. Como critério de exclusão, foi definido que a coluna que possuísse um número maior de outras colunas similares permanecesse, em detrimento das demais. Em caso de empate, permaneceu a primeira coluna encontrada, mais à esquerda na tabela de dados.

4.4 SELEÇÃO DE DADOS

Como visto na seção anterior, nesta etapa existiam 51 colunas de dados de todas as empresas, um número considerado alto. Para que o processo pudesse ser otimizado, optou-se por reduzir esta quantidade de colunas. Levou-se em conta, também, a qualidade dos resultados que pode ser prejudicada ao manter todas as colunas existentes.

Para isto, foi aplicado um processo de *feature selection*, onde foram selecionadas, matematicamente, as dez colunas que mais interferiam na coluna de decisão, ou seja, no resultado da classificação. Foi utilizado um modelo simples de *Extra Trees Classifier*, o qual possui um parâmetro de *feature importances*, ou importância de cada coluna.

Após a aplicação deste modelo, foram identificadas as colunas mais importantes para a análise de classificação, que podem ser vistas na [Figura 28](#). Pode-se observar que as quatro melhores classificadas são indicadores fundamentalistas calculados no pré-processamento, o que corrobora para a importância de considerá-los e vai de encontro à visão fundamentalista de análise de mercado.

| | |
|----------------------------------|-----------------|
| P/VPA | 0.048986 |
| P/L | 0.041931 |
| LPA | 0.033950 |
| DY | 0.029078 |
| Obrigações Fiscais | 0.026876 |
| Financeiras | 0.026175 |
| Estoques_1 | 0.025328 |
| Tributos a Recuperar | 0.024958 |
| Outros Ativos Circulantes | 0.024792 |
| Despesas Com Vendas | 0.024453 |

Figura 28 – Colunas mais importantes na classificação

Fonte: Elaborado pelo autor.

Com a realização deste processo de *feature selection* e a identificação das colunas mais importantes, foram descartadas dos dados as demais que não estavam entre as mostradas na Figura 28. Em uma última etapa de preparação, os dados foram normalizados utilizando o *Standard Scaler*, método disponível na biblioteca sklearn do Python e, posteriormente, separados em grupos de treino e teste para serem submetidos aos modelos de inteligência computacional. Após a finalização destes procedimentos, pode-se verificar o estado dos dados na Figura 29.

| | Estoques_1 | Tributos a Recuperar | Outros Ativos Circulantes | Obrigações Fiscais | Despesas Com Vendas | Financeiras | P/L | P/VPA | DY | LPA | Decisao |
|-----|------------|----------------------|---------------------------|--------------------|---------------------|-------------|----------|-----------|-----------|-----------|---------|
| 0 | -0.011016 | -0.033460 | -0.037767 | -0.038725 | 0.080915 | 0.025797 | 0.015865 | -0.021737 | -0.039389 | -0.000196 | 0 |
| 1 | -0.025936 | -0.042931 | -0.042693 | -0.041154 | 0.080915 | 0.027443 | 0.013109 | -0.036060 | -0.039389 | 7.527497 | 2 |
| 2 | -0.026638 | -0.032441 | -0.040249 | -0.035474 | 0.080915 | 0.031635 | 0.015340 | -0.028014 | -0.039389 | -0.019925 | 2 |
| 3 | -0.028158 | -0.035007 | -0.041271 | -0.036348 | 0.080915 | 0.009097 | 0.017182 | -0.007593 | -0.039389 | -0.024033 | 0 |
| 4 | -0.024555 | -0.039936 | -0.042392 | -0.044370 | 0.080915 | 0.027734 | 0.013487 | -0.026166 | -0.039389 | 0.022231 | 0 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |

Figura 29 – Dados após o processo de *feature selection*

Fonte: Elaborado pelo autor.

4.5 APLICAÇÃO DOS MODELOS DE INTELIGÊNCIA COMPUTACIONAL

Nesta etapa do trabalho foram testados os modelos de inteligência computacional com os dados tratados anteriormente. Primeiramente, foram testados diversos modelos genéricos, a fim de dar um melhor entendimento sobre o comportamento deles no cenário do projeto. Em sequência, foi utilizada uma biblioteca de *auto machine learning*, onde pode-se analisar o desempenho de outros modelos, também de forma genérica. Por fim, os que melhor performarem nestas duas primeiras etapas serão otimizados para uma melhor análise.

O desempenho poderá ser compreendido conforme métricas de precisão, revocação e média F1, ou *precision*, *recall* e *f1-score*, além da acurácia. A acurácia por si só não pode ser o único parâmetro analisado, visto que não é o melhor avaliador em um cenário onde a quantidade de dados é desbalanceada.

A precisão, por definição, trabalha com falsos positivos, sendo que quanto maior o seu valor, maior o número de amostras classificadas positivamente de forma correta. Em outras palavras, este parâmetro indica a proporção de identificações positivas que estavam corretas. Já a revocação, ou *recall*, leva em consideração falsos negativos. Neste, é analisada a proporção de positivos verdadeiros que foram analisados corretamente.

A fim de considerar ambos os parâmetros em nossa análise, um terceiro foi utilizado no trabalho. O *f1-score* une a precisão e o *recall* em um único valor, sendo uma média harmônica. Para que atinja um valor alto, é necessário que ambos parâmetros também sejam altos.

Também é importante destacar que as métricas foram consideradas sobre a rotulação de compra, ou *buy*, que possuíam rótulo de valor 2. Isso se dá devido ao contexto do trabalho, onde o objetivo definido diz respeito à seleção de empresas visando uma possível compra. Então, desta forma, foi analisado o desempenho dos modelos ao classificar empresas como compráveis.

A aplicação dos modelos, portanto, foi realizada e, para que pudessem ser avaliadas de forma a evitar qualquer tipo de viés ou tendência, cada um foi executado 25 vezes onde, ao final, foi tirado a média e desvio padrão dos parâmetros de avaliação.

4.5.1 Testes com versões genéricas de diversos modelos

Primeiramente, vários modelos genéricos foram submetidos aos dados de treino e teste anteriormente separados e o desempenho destes foi testado. Os modelos escolhidos para esta primeira etapa foram:

- *AdaBoost Classifier*;
- *Decision Tree Classifier*;
- *Dummy Classifier*;
- *Extra Trees Classifier*;
- *Gradient Boost Classifier*;
- *KNN Classifier*;
- *Logistic Regression*;
- *Naive Bayes*;
- *Random Forest Classifier*;
- Redes Neurais Artificiais Multi-camadas (MLP);
- *Support Vector Machine*.

Cada modelo, ao ser executado, gerou uma matriz de confusão e seus próprios valores de parâmetros precisão, *recall* e *f1-score*. Na [Figura 30](#) é possível verificar os resultados de uma execução do modelo *AdaBoost Classifier* genérico. Na imagem, pode-se observar na parte superior os valores dos parâmetros para esta execução, onde o valor de *f1-score* para o rótulo de compra foi de 0.33. Já na parte inferior, pode-se observar uma matriz de confusão, onde observa-se que, de um total de 1536 amostras, foram classificadas corretamente como compra 151.

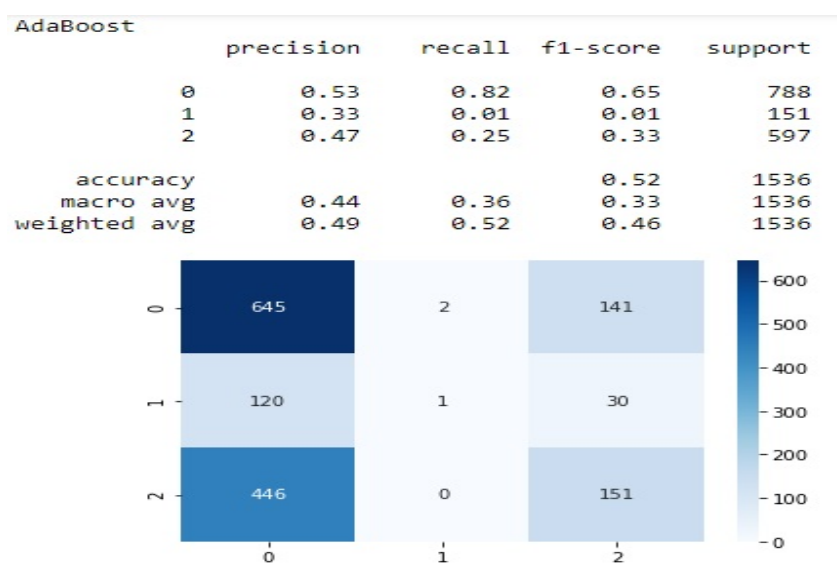


Figura 30 – Resultados de uma execução do modelo *AdaBoost Classifier*

Fonte: Elaborado pelo autor.

Após cada modelo ser executado 25 vezes, foi avaliada a média e desvio padrão dos valores de *f1-score* obtidos. Os resultados destas execuções podem ser verificados na [Tabela 1](#).

| Nome do modelo | Média de <i>f1-score</i> |
|---------------------------------|--------------------------|
| <i>AdaBoost Classifier</i> | 0.33 (\pm 0.0) |
| <i>Decision Tree Classifier</i> | 0.44 (\pm 0.01) |
| <i>Dummy Classifier</i> | 0.38 (\pm 0.02) |
| <i>Extra Trees Classifier</i> | 0.48 (\pm 0.01) |
| <i>Gradient Classifier</i> | 0.34 (\pm 0.0) |
| KNN | 0.40 (\pm 0.0) |
| <i>Logistic Regression</i> | 0.01 (\pm 0.0) |
| <i>Naive Bayes</i> | 0.07 (\pm 0.0) |
| <i>Random Forest Classifier</i> | 0.46 (\pm 0.01) |
| Rede neural MLP | 0.02 (\pm 0.01) |
| SVM | 0.02 (\pm 0.0) |

Tabela 1 – Resultados da primeira etapa de aplicação de modelos

Com estes resultados apresentados na [Tabela 1](#), é possível identificar modelos que obtiveram performance melhor do que outros. O *Random Forest*, por exemplo, foi o que apresentou uma maior média de *f1-score* com o menor desvio padrão. Também é importante observar que seis modelos obtiveram uma média menor do que o algoritmo de *Dummy Classifier*, que apenas realiza classificações randômicas, sem nenhum parâmetro. Portanto, estes podem ser considerados com desempenho pior do que uma escolha aleatória.

4.5.2 Testes com *Machine Learning* Automatizado

Em uma segunda etapa de aplicação de modelos, a fim de ampliar o espectro de informações sobre o desempenho dos mesmos no contexto do trabalho, foi utilizada uma biblioteca de *machine learning* automatizado, a PyCaret. Nesta, foi possível realizar testes em outros modelos disponíveis, além de também testar algumas opções previamente testadas.

Assim como na aplicação anterior, cada modelo foi executado 25 vezes para que pudesse ser avaliada a média e desvio padrão dos valores de *f1-score*. Foram avaliados os

seguintes:

- *AdaBoost Classifier*
- *CatBoost Classifier*
- *Decision Tree Classifier*
- *Dummy Classifier*
- *Extra Trees Classifier*
- *Extreme Gradient Boosting Machine*
- *Light Gradient Boosting Machine*
- *Linear Discriminant Analysis*
- *Logistic Regression*
- *Gradient Boosting Classifier*
- *KNN Classifier*
- *Naive Bayes*
- *Quadratic Discriminant Analysis*
- *Random Forest Classifier*
- *Ridge Classifier*
- *Support Vector Machine - Linear Kernel*

A biblioteca PyCaret fornece resultados de uma forma visualmente simples e de fácil entendimento, fornecendo diversos parâmetros para avaliação. Um exemplo de uma execução desta é visto na [Figura 31](#). Os resultados finais desta etapa podem ser visualizados na [Tabela 2](#).

| | Model | Accuracy | AUC | Recall | Prec. | F1 | Kappa | MCC | TT (Sec) |
|-----------------|---------------------------------|----------|--------|--------|--------|--------|---------|---------|----------|
| rf | Random Forest Classifier | 0.5679 | 0.6485 | 0.4271 | 0.5472 | 0.5401 | 0.1899 | 0.1973 | 0.5590 |
| et | Extra Trees Classifier | 0.5651 | 0.6567 | 0.4286 | 0.5421 | 0.5371 | 0.1861 | 0.1949 | 0.5480 |
| catboost | CatBoost Classifier | 0.5614 | 0.6324 | 0.4150 | 0.5319 | 0.5341 | 0.1820 | 0.1871 | 7.9990 |
| lightgbm | Light Gradient Boosting Machine | 0.5593 | 0.6311 | 0.4182 | 0.5358 | 0.5311 | 0.1737 | 0.1800 | 0.6920 |
| xgboost | Extreme Gradient Boosting | 0.5579 | 0.6370 | 0.4209 | 0.5368 | 0.5341 | 0.1793 | 0.1837 | 1.8720 |
| gbc | Gradient Boosting Classifier | 0.5407 | 0.5984 | 0.3759 | 0.5075 | 0.4858 | 0.1050 | 0.1200 | 1.7670 |
| dummy | Dummy Classifier | 0.5198 | 0.5000 | 0.3333 | 0.2702 | 0.3555 | 0.0000 | 0.0000 | 0.0160 |
| ridge | Ridge Classifier | 0.5195 | 0.0000 | 0.3339 | 0.4344 | 0.3604 | 0.0017 | 0.0054 | 0.0100 |
| lda | Linear Discriminant Analysis | 0.5195 | 0.5072 | 0.3339 | 0.4344 | 0.3604 | 0.0017 | 0.0054 | 0.0180 |
| lr | Logistic Regression | 0.5193 | 0.5102 | 0.3338 | 0.4368 | 0.3611 | 0.0017 | 0.0048 | 0.0630 |
| ada | Ada Boost Classifier | 0.5174 | 0.5494 | 0.3558 | 0.4544 | 0.4597 | 0.0578 | 0.0670 | 0.1820 |
| svm | SVM - Linear Kernel | 0.5128 | 0.0000 | 0.3337 | 0.3961 | 0.3693 | -0.0015 | -0.0028 | 0.0290 |
| dt | Decision Tree Classifier | 0.5053 | 0.5768 | 0.4246 | 0.5104 | 0.5074 | 0.1494 | 0.1496 | 0.0300 |
| knn | K Neighbors Classifier | 0.5000 | 0.5596 | 0.3732 | 0.4686 | 0.4752 | 0.0769 | 0.0793 | 0.0680 |
| qda | Quadratic Discriminant Analysis | 0.1272 | 0.5129 | 0.3348 | 0.4329 | 0.0781 | 0.0004 | 0.0005 | 0.0120 |
| nb | Naive Bayes | 0.1253 | 0.5130 | 0.3351 | 0.4329 | 0.0735 | -0.0002 | -0.0014 | 0.0190 |

Figura 31 – Resultados da biblioteca PyCaret

Fonte: Elaborado pelo autor.

| Nome do modelo | Média de $f1$ -score |
|--|----------------------|
| <i>AdaBoost Classifier</i> | 0.46 (\pm 0.01) |
| <i>CatBoost Classifier</i> | 0.52 (\pm 0.01) |
| <i>Decision Tree Classifier</i> | 0.49 (\pm 0.01) |
| <i>Dummy Classifier</i> | 0.35 (\pm 0.01) |
| <i>Extra Trees Classifier</i> | 0.54 (\pm 0.01) |
| <i>Extreme Gradient Boosting Machine</i> | 0.53 (\pm 0.01) |
| <i>Light Gradient Boosting Machine</i> | 0.52 (\pm 0.01) |
| <i>Linear Discriminant Analysis</i> | 0.36 (\pm 0.01) |
| <i>Logistic Regression</i> | 0.36 (\pm 0.01) |
| <i>Gradient Boosting Classifier</i> | 0.48 (\pm 0.01) |
| <i>KNN Classifier</i> | 0.47 (\pm 0.02) |
| <i>Naive Bayes</i> | 0.07 (\pm 0.0) |
| <i>Quadratic Discriminant Analysis</i> | 0.07 (\pm 0.0) |
| <i>Random Forest Classifier</i> | 0.53 (\pm 0.01) |
| <i>Ridge Classifier</i> | 0.36 (\pm 0.01) |
| <i>SVM - Linear Kernel</i> | 0.37 (\pm 0.01) |

Tabela 2 – Resultados da segunda etapa de aplicação de modelos

Assim como na primeira etapa realizada, é possível identificar alguns modelos com desempenho inferior ao *Dummy Classifier*, que podem ser descartados de imediato. Nesta segunda avaliação, pode-se observar que o melhor resultado de média de $f1$ -score foi obtido pelo modelo *Extra Trees Classifier*, cujo desvio padrão médio também é considerado relativamente baixo e semelhante aos demais.

4.5.3 Seleção dos melhores modelos

Após a realização das duas primeiras etapas de aplicação de modelos de inteligência computacional, foram selecionados os cinco melhores modelos para que pudessem ser

submetidos a um processo de otimização dos hiperparâmetros. Ao analisar os resultados das médias de *f1-score* e avaliar a média dos desvios padrão obtidos, foram selecionados cinco modelos que obtiveram melhor desempenho.

Os algoritmos escolhidos para a sequência do trabalho foram o *CatBoost Classifier*, *Extra Trees Classifier*, *Extreme Gradient Boosting Machine* (XGBoost), *Random Forest Classifier* e *Light Gradient Boosting Machine* (LightGBM). Cada um destes foi submetido, posteriormente, a um processo de otimização e execução de simulações com dados reais do mercado financeiro brasileiro.

4.5.4 Otimização de hiperparâmetros dos melhores modelos

Com os modelos selecionados, foi realizado um processo de otimização de hiperparâmetros numa tentativa de tornar o desempenho destes ainda melhor. Para tal, foi necessário utilizar técnicas de testes de uma série de valores para cada parâmetro de cada modelo. Dentre as opções existentes, há a busca profunda (*Grid Search*) e a busca randômica (*Random Search*).

O *Grid Search* é uma técnica que, tendo uma série de parâmetros, testa todas as possibilidades existentes a fim de encontrar a que tem o melhor desempenho. Dependendo da quantidade de cenários existentes, este processo exige um grande poder de processamento e pode demorar uma grande quantidade de horas para ser finalizado.

Já o *Random Search* pode ser considerado uma alternativa viável quando há um poder de processamento baixo disponível. É um processo mais rápido de ser executado, visto que não testa todas as combinações de parâmetros possíveis. Neste, é selecionada uma quantidade pré-definida de cenários que serão testados de forma aleatória, retornando um resultado não tão preciso quanto uma busca profunda, mas com desempenho próximo ao que seria obtido.

Neste trabalho, visto que a disponibilidade de processamento era limitada, foi decidido pela utilização da busca randômica no processo de otimização. Na [seção 6.1](#) é sugerida uma aplicação deste trabalho utilizando a busca profunda em um ambiente computacionalmente mais robusto. Também é importante ressaltar a utilização de validação cruzada nesta etapa. A busca se deu através da biblioteca *sklearn* do Python, utilizando o método *RandomizedSearchCV* e ao longo das otimizações, em cada modelo foram realizadas cerca de 10.000 iterações, utilizando como parâmetro de pontuação o *f1-weighted*. A escolha dos valores foi dada conforme a documentação de cada um dos modelos, selecionando um espectro de valores válidos

4.5.4.1 CatBoost Classifier

A otimização do modelo de *CatBoost* se deu através dos parâmetros *depth*, *iterations*, *learning_rate*, *l2_leaf_reg*, *border_count* e *thread_count*. Pode-se observar o código realizado na busca randômica na Figura 32. Já na Tabela 3 vê-se os resultados encontrados por essa busca randômica.

```
from sklearn.model_selection import GridSearchCV

param_grid_catboost = dict(depth=[3,1,2,6,4,5,7,8,9,10],
                             iterations=[250,100,500,1000],
                             learning_rate=[0.03,0.001,0.01,0.1,0.2,0.3],
                             l2_leaf_reg=[3,1,5,10,100],
                             border_count=[32,5,10,20,50,100,200],
                             thread_count=[1,2,3,4,5],
                             verbose=[False]
                             )

catboost_classifier = CatBoostClassifier()
grid_catboost = RandomizedSearchCV(catboost_classifier, param_grid_catboost, cv=5, n_iter=10000, scoring='f1_weighted')
grid_catboost_result = grid_catboost.fit(x_treino, y_treino)
```

Figura 32 – Aplicação da busca randômica para o algoritmo *CatBoost*

Fonte: Elaborado pelo autor.

| Parâmetro | Resultado encontrado pela busca randômica |
|----------------------|---|
| <i>depth</i> | 10 |
| <i>iterations</i> | 500 |
| <i>learning_rate</i> | 0.3 |
| <i>l2_leaf_reg</i> | 1 |
| <i>border_count</i> | 32 |
| <i>thread_count</i> | 1 |

Tabela 3 – Resultados da busca randômica do modelo *CatBoost*.

4.5.4.2 Extra Trees Classifier

O modelo de *Extra Trees* teve os seguintes parâmetros otimizados pela busca randômica: *criterion*, *max_depth*, *min_samples_split*, *min_samples_leaf*, *max_features* e *n_estimators*. A aplicação da busca é observada na Figura 33 e os resultados da mesma na Tabela 4.

```
#RANDOMIZED SEARCH
modelo_extra_trees = ExtraTreesClassifier()

param_grid_extra_trees = dict(criterion=['gini', 'entropy'], max_depth=range(1,50,1),
                              min_samples_split=range(2,50,2), min_samples_leaf=range(1,50,1),
                              max_features=['auto', 'sqrt', 'log2'], n_estimators=range(1,50,1))

grid_extra_trees = RandomizedSearchCV(modelo_extra_trees, param_grid_extra_trees, cv=5, n_iter=10000,
                                      scoring='f1_weighted')
```

Figura 33 – Aplicação da busca randômica para o algoritmo *Extra Trees*

Fonte: Elaborado pelo autor.

| Parâmetro | Resultado encontrado pela busca randômica |
|--------------------------|---|
| <i>criterion</i> | <i>gini</i> |
| <i>max_depth</i> | 48 |
| <i>min_samples_split</i> | 14 |
| <i>min_samples_leaf</i> | 1 |
| <i>max_features</i> | <i>auto</i> |
| <i>n_estimators</i> | 4 |

Tabela 4 – Resultados da busca randômica do modelo *Extra Trees*.

4.5.4.3 *Extreme Gradient Boosting Machine* (XGBoost)

Para que fosse realizada a busca randômica para o modelo XGBoost, foram selecionados os parâmetros *max_depth*, *min_child_weight*, *subsample*, *learning_rate*, *n_estimators* e *colsample_bytree*. Na [Figura 34](#) é vista a busca realizada e na [Tabela 5](#) pode-se visualizar os resultados encontrados.

```
#RANDOMIZED SEARCH
param_grid_xgboost = dict(max_depth=[3,4,5,6,7,8,9,10,11,12,15,20,30,40,50,70,90,120],
                           min_child_weight=[1, 2, 4, 8, 16, 32, 64],
                           subsample=[0,0.25,0.5,0.75,1],
                           learning_rate=[0.01,0.1,0.2,0.3,0.5,1],
                           n_estimators=[1, 2, 4, 8, 16, 32, 64, 128],
                           colsample_bytree=[0.5,0.7,1],
                           eval_metric=['mlogloss'])

xgboost = xgb.XGBClassifier()
grid_xgboost = RandomizedSearchCV(xgboost, param_grid_xgboost, cv=5, n_iter=10000, scoring='f1_weighted')
```

Figura 34 – Aplicação da busca randômica para o algoritmo XGBoost

Fonte: Elaborado pelo autor.

| Parâmetro | Resultado encontrado pela busca randômica |
|-------------------------|---|
| <i>max_depth</i> | 20 |
| <i>min_child_weight</i> | 4 |
| <i>subsample</i> | 0.75 |
| <i>learning_rate</i> | 0.2 |
| <i>n_estimators</i> | 128 |
| <i>colsample_bytree</i> | 0.7 |

Tabela 5 – Resultados da busca randômica do modelo XGBoost.

4.5.4.4 *Random Forest Classifier*

A otimização do modelo de *Random Forest* se deu pelos parâmetros *bootstrap*, *max_depth*, *max_features*, *min_samples_leaf*, *min_samples_split* e *criterion*. A [Figura 35](#) e a [Tabela 6](#) mostram o código de busca realizada e os resultados obtidos.

```
#RANDOMIZED SEARCH
param_grid_random_forest = dict(bootstrap=[True, False],
max_depth=[4,5,6,7,8,9,10,11,12,15,20,30,40,50,70,90,120,150],
max_features=['auto', 'sqrt'],
min_samples_leaf=[1, 2, 4, 8, 16, 32, 64, 128, 256],
min_samples_split=[2, 4, 8, 16, 32, 64, 128, 256],
criterion=['gini', 'entropy'])

random_forest_classifier = RandomForestClassifier()

grid_random_forest_classifier = RandomizedSearchCV(random_forest_classifier, param_grid_random_forest, cv=5, n_iter=10000,
scoring='f1_weighted')

grid_random_forest_classifier_result = grid_random_forest_classifier.fit(x_treino, y_treino)
```

Figura 35 – Aplicação da busca randômica para o algoritmo *Random Forest*

Fonte: Elaborado pelo autor.

| Parâmetro | Resultado encontrado pela busca randômica |
|--------------------------|---|
| <i>bootstrap</i> | <i>True</i> |
| <i>criterion</i> | <i>gini</i> |
| <i>max_depth</i> | 70 |
| <i>max_features</i> | <i>auto</i> |
| <i>min_samples_leaf</i> | 1 |
| <i>min_samples_split</i> | 2 |

Tabela 6 – Resultados da busca randômica do modelo *Random Forest*.

4.5.4.5 *Light Gradient Boosting Machine* (LightGBM)

O modelo LightGBM foi otimizado utilizando a busca randômica dos seguintes parâmetros: *learning_rate*, *num_leaves*, *subsample*, *colsample_bytree*, *max_depth*, *min_child_samples* e *bagging_fraction*. Na [Figura 36](#) é verificada a busca realizada e, na [Tabela 7](#), os resultados deste processo.

```

param_grid_lightgbm = {'learning_rate': [0.03,0.001,0.01,0.1,0.2,0.3],
                        'objective':['multiclass'],
                        'num_leaves': range(2,3000,20),
                        'subsample': [0,0.25,0.5,0.75,1],
                        'colsample_bytree': [0.5,0.7,1],
                        'max_depth': [3,4,5,6,7,8,9,10,11,12,15,20,30,40,50,70,90,120],
                        'min_child_samples': [1, 2, 4, 8, 16, 32, 64],
                        'bagging_fraction': [0.1,0.3,0.5,0.7],
                        'early_stopping_rounds': [None],
                        'verbose': [-1]
                        }

lightgbm = lgb.LGBMClassifier()

grid_lightgbm = RandomizedSearchCV(lightgbm, param_grid_lightgbm, cv=5, n_iter=10000, scoring='f1_weighted')
grid_lightgbm_result = grid_lightgbm.fit(x_treino, y_treino)

```

Figura 36 – Aplicação da busca randômica para o algoritmo LightGBM

Fonte: Elaborado pelo autor.

| Parâmetro | Resultado encontrado pela busca randômica |
|--------------------------|---|
| <i>learning_rate</i> | 0.1 |
| <i>num_leaves</i> | 222 |
| <i>subsample</i> | 1 |
| <i>colsample_bytree</i> | 0.3 |
| <i>max_depth</i> | 90 |
| <i>min_child_samples</i> | 1 |
| <i>bagging_fraction</i> | 0.3 |

Tabela 7 – Resultados da busca randômica do modelo LightGBM.

4.6 SIMULAÇÕES COM DADOS REAIS

Após o processo de otimização de hiperparâmetros realizado, cada modelo foi submetido a uma simulação para verificar seu real desempenho com dados reais, a fim de entender se a utilização dos mesmos é válida ou não. Cada algoritmo realizou, estando com dados do primeiro trimestre de março de 2021, previsões para o trimestre seguinte, de junho de 2021, onde selecionou empresas possivelmente lucrativas para serem compradas.

Ao realizar esta seleção, foi considerada uma compra, de forma igualitária, de todas as empresas selecionadas anteriormente para junho de 2021, a fim de obter uma carteira fictícia com todas as empresas escolhidas, todas com a mesma quantidade de ações. Com isso, foi analisada a valorização desta carteira um ano depois, no cenário do mês de junho

de 2022. Foi possível, portanto, verificar se esta carteira teria ou não potencial lucrativo no longo prazo e se os algoritmos foram válidos ao selecionar boas empresas.

Assim como nas etapas de testes anteriores, cada modelo realizou 25 simulações, onde, ao final, foi retirada a média e desvio padrão da valorização obtida em cada execução, além da quantidade de ações selecionadas.

5 RESULTADOS E DISCUSSÕES

Neste capítulo serão apresentados os resultados das simulações realizadas pelos modelos otimizados no capítulo anterior. A análise destes resultados se dará através da média e desvio padrão das 25 execuções de cada modelo. Serão consideradas a valorização da carteira no intervalo de 1 ano, a quantidade de ações selecionadas e a variação do tamanho desta carteira fictícia gerada.

A fim de comparação, o total de empresas analisadas no trabalho, após todos os processos de pré-processamento foi de 370. O rendimento de uma carteira igualitária com ativos de todas estas empresas, no intervalo adotado entre junho de 2021 e junho de 2022, foi de -18,59%, tendo, portanto, uma desvalorização. Estes valores foram utilizados como base de comparação para os resultados obtidos nas simulações.

5.1 CATBOOST CLASSIFIER

Os resultados da simulação do modelo otimizado *CatBoost Classifier* podem ser vistos na [Tabela 8](#).

| Variável | Média |
|---------------------------------|-------------------------|
| Quantidade de ações na carteira | 129 (± 24) |
| Variação no tamanho da carteira | -65,06% ($\pm 6,5\%$) |
| Rendimento da carteira | -16,65% ($\pm 1\%$) |

Tabela 8 – Resultados da simulação do modelo *CatBoost Classifier*.

Pode-se observar que este algoritmo teve uma média de valorização de rendimento de -16,65%, quase 2% maior do que a média do mercado, com uma carteira de tamanho médio de 129 ações, uma redução de cerca de 65% com relação ao total de 370. Ou seja, obteve um valorização maior com um número reduzido de ações na carteira, tendo, portanto, pré-selecionado empresas lucrativas corretamente.

5.2 EXTRA TREES CLASSIFIER

Após a realização das simulações, pode-se observar os resultados obtidos pelo modelo *Extra Trees Classifier* na [Tabela 9](#).

| Variável | Média |
|---------------------------------|--------------------------|
| Quantidade de ações na carteira | 152 (± 54) |
| Variação no tamanho da carteira | -58,83% ($\pm 14,7\%$) |
| Rendimento da carteira | -17,64% ($\pm 2,5\%$) |

Tabela 9 – Resultados da simulação do modelo *Extra Trees Classifier*.

Com estes resultados, é possível verificar o desempenho deste algoritmo. Com uma quantidade de ações reduzida em aproximadamente 59%, foi possível obter um rendimento pouco acima do constatado na carteira com todas as empresas. Ou seja, foi capaz de, na média das simulações, gerar carteiras com 152 ações e ter quase o mesmo resultado de uma carteira com 370, conseguindo ainda uma média pouco acima.

5.3 *EXTREME GRADIENT BOOSTING MACHINE (XGBOOST)*

Na [Tabela 10](#) são mostrados os resultados das execuções das simulações para o modelo XGBoost.

| Variável | Média |
|---------------------------------|-------------------------|
| Quantidade de ações na carteira | 93 (± 31) |
| Variação no tamanho da carteira | -74,86% ($\pm 8,3\%$) |
| Rendimento da carteira | -15,75% ($\pm 1,5\%$) |

Tabela 10 – Resultados da simulação do modelo XGBoost.

Este algoritmo otimizado foi capaz de gerar carteiras, em média, aproximadamente 75% menores e quase 3% mais rentáveis, em comparação com uma carteira fictícia com todas as empresas disponíveis.

5.4 *RANDOM FOREST CLASSIFIER*

Os resultados das execuções de simulação do modelo *Random Forest Classifier* são mostrados na [Tabela 11](#).

| Variável | Média |
|---------------------------------|-------------------------|
| Quantidade de ações na carteira | 181 (± 31) |
| Variação no tamanho da carteira | -51% ($\pm 8,3\%$) |
| Rendimento da carteira | -18,24% ($\pm 1,4\%$) |

Tabela 11 – Resultados da simulação do modelo *Random Forest*.

É possível observar que este algoritmo apresentou uma redução do número de ações de cerca de 51% com relação ao total de empresas disponíveis. Já com relação ao rendimento, o resultado foi praticamente o mesmo.

5.5 *LIGHT GRADIENT BOOSTING MACHINE* (LIGHTGBM)

Na [Tabela 12](#) é possível verificar os resultados das simulações do modelo otimizado de LightGBM.

| Variável | Média |
|---------------------------------|------------------------|
| Quantidade de ações na carteira | 129 (± 60) |
| Variação no tamanho da carteira | -65% ($\pm 16,2\%$) |
| Rendimento da carteira | -19,5% ($\pm 2,7\%$) |

Tabela 12 – Resultados da simulação do modelo LightGBM.

É possível constatar que este foi o único modelo a apresentar um rendimento médio menor do que a carteira fictícia com todas as empresas disponibilizadas, apesar de apresentar uma redução no tamanho de cerca de 65%.

5.6 DISCUSSÕES SOBRE OS RESULTADOS

Após a execução de todas as simulações e análise dos resultados, foi possível constatar que nenhum dos modelos apresentou um rendimento muito superior à média do mercado brasileiro, onde o que mais se destacou foi o XGBoost, tendo uma média cerca de 3% maior. Porém, o ponto mais importante a ser verificado foi a redução do tamanho da carteira. Todos obtiveram reduções superiores à 50%, chegando a índices superiores à 74%.

O modelo que apresentou os melhores desempenhos foi o XGBoost, como constatado anteriormente. Este obteve a maior média de valorização, além de apresentar a maior redução do tamanho da carteira, sempre com valores de desvio padrão baixos. O segundo

melhor foi o *CatBoost Classifier*, com uma redução média de 65% e uma valorização quase 2% maior do que a média do mercado.

Por outro lado, o algoritmo LightGBM foi o único que apresentou uma média de rendimentos inferior à carteira fictícia com todas as empresas disponíveis. Apesar de ser uma variação de aproximadamente 1% a menos, pode-se considerar que este não realizou a seleção de ativos de uma maneira tão qualificada quanto os demais, visto que todos os outros quatro obtiveram médias superiores.

6 CONCLUSÃO

Este trabalho teve como objetivo propor uma metodologia capaz de pré-selecionar empresas do mercado financeiro brasileiro, visando o investidor iniciante. Para que este pudesse ser atingido, foram avaliados os desempenhos de diversos modelos de inteligência computacional ao selecionar ativos com base em seus balanços patrimoniais trimestrais, demonstrativos de resultados de exercício e indicadores fundamentalistas. Foram aplicadas todas as etapas de entendimento do problema, coleta e pré-processamento dos dados, análise exploratória, *feature selection* e execução dos modelos.

A coleta dos dados foi realizada através dos portais "Fundamentus", onde foram obtidos o histórico de balanços patrimoniais trimestrais e DREs de todas as empresas listadas na B3, e "Yahoo! Finanças", de onde foram extraídos o histórico de cotações desses ativos. A partir desta etapa, foi realizado o pré-processamento, onde foi possível tratar as informações para que pudessem ser melhor utilizadas no processo e otimizar os resultados. Em sequência, o processo de análise exploratória possibilitou um melhor entendimento destas informações trabalhadas no projeto. Também foi realizado o processo de seleção de dados, ou *feature selection*, a fim de otimizá-los e facilitar a posterior aplicação dos mesmos nos modelos propostos.

Após o pré-processamento e com os dados prontos para utilização, foi realizada uma série de testes com diversos modelos de inteligência computacional a fim de entender o comportamento dos mesmos no cenário do trabalho. Foram testadas versões simplificadas de modelos como árvores de decisão, redes neurais artificiais multi-camadas e KNN. Ao final, seguindo os critérios de precisão, *recall* e *f1-score*, foram selecionados os cinco modelos com melhor performance para uma posterior otimização dos mesmos.

Os modelos de inteligência computacional selecionados para um processo de otimização foram o *Extra Trees*, *Random Forest*, *CatBoost*, *XGBoost* e *LightGBM*. Cada um destes foi submetido a uma busca randômica a fim de melhorar seus hiperparâmetros. Ao final, estes cinco modelos otimizados foram submetidos a simulações com dados reais para verificação de seus desempenhos, onde cada modelo selecionou empresas que julgou possivelmente lucrativas para o trimestre seguinte.

A simulação foi realizada com dados de março de 2021, a fim de selecionar empresas possivelmente lucrativas visando o trimestre seguinte, ou seja, junho de 2021. Foi analisado o desempenho de cada carteira fictícia gerada no intervalo de um ano. Em resumo, cada modelo escolheu ações visando o mês de junho de 2021 e foi verificado o desempenho dessas ações escolhidas no intervalo de um ano, em junho de 2022.

Após a realização das simulações, é visto que o modelo com melhor performance foi o XGBoost, onde apresentou a maior redução do tamanho da carteira, descartando mais de 74% dos ativos listados, além de um rendimento de quase 3% maior com relação à média de todas as empresas listadas.

Desta forma, pode ser constatado que o objetivo principal de pré-seleção de empresas possivelmente lucrativas foi atingido, visto que os cinco modelos apresentaram rendimentos semelhantes à média do mercado, porém com um número significativamente menor de ativos em uma carteira, tendo uma redução sempre acima de 50%. Ao aplicar esta metodologia proposta durante o trabalho, um investidor iniciante poderia reduzir ao menos pela metade seu escopo de empresas a analisar, a fim de tomar melhores decisões de investimento.

6.1 TRABALHOS FUTUROS

Durante o desenvolvimento deste trabalho, foram observados alguns pontos que podem ser melhorados e implementados. Um destes diz respeito à fase de pré-processamento, mais especificamente onde são criados as classificações de *buy*, *hold* e *sell*. Nesta etapa, é utilizado um percentual de 3% para definir a rotulação de cada campo. Podem ser feitas simulações e testes a fim de definir o valor ideal para o mercado nacional e o contexto do trabalho. Outro ponto de atenção seria a adição de mais dados sobre as empresas e de outros indicadores fundamentalistas, como o índice de Graham, por exemplo.

Durante o processo de otimização dos modelos de inteligência computacional, pode ser feito uma maior otimização dos hiperparâmetros, utilizando busca profunda e um espectro maior de valores possíveis, a fim de encontrar os parâmetros ótimos e ideais para cada um.

Outro ponto a ser trabalhado é a integração deste projeto a uma aplicação real. Pode-se automatizar o processo de pré-processamento, a fim de facilitar a entrada dos dados por parte do usuário, além de utilizar o melhor modelo encontrado para realizar as classificações e, assim, realizar a filtragem de ativos para que o investidor possa avaliá-los individualmente.

Por fim, pode-se aplicar, junto a esta metodologia proposta, o uso de métodos de auxílio multicritério à decisão, como o *Anarchy Hierarchy Process* (AHP). Ao utilizá-lo, será possível fornecer um poder maior ao investidor iniciante que utilizaria a metodologia, onde poderia customizar os pesos de determinadas características a fim de ter um resultado mais personalizado, em comum acordo com suas preferências de investimento.

REFERÊNCIAS

- AKBARI, A.; NG, L.; SOLNIK, B. Drivers of economic and financial integration: A machine learning approach. *Journal of Empirical Finance*, v. 61, 01 2021. Citado na página 16.
- ALMONACID, G. A.; SANTOVITO, R. F. Aplicabilidade da teoria de markowitz para investimentos em ativos do real estate: Estudo de caso de uma carteira mista. In: *10ª Conferência Internacional da LARES*. São Paulo: [s.n.], 2010. (LARES). Citado na página 6.
- AQUINO, C. P.; FONSECA, A. U. d.; OLIVEIRA, L. L. G. d. Avaliação de classificadores no diagnóstico de câncer de mama. *XV Congresso Brasileiro de Informática em Saúde*, p. 915–927, 2016. Citado na página 21.
- ARCANJO, J. *Jupyter Notebook- A melhor maneira de criar uma história com dados*. 2022. Acessado em 29/11/2022". Disponível em: <<https://medium.com/data-hackers/jupyter-notebook-a-melhor-maneira-de-criar-uma-história-com-dados-dbc2e8e3dd9a>>. Citado na página 27.
- AWARI. *Entenda o que é Scikit Learn e aprenda como usar essa biblioteca*. 2022. Acessado em 30/11/2022". Disponível em: <https://awari.com.br/scikit-learn/?utm_source=blog>. Citado na página 31.
- BICHARA, G. L. G. Redes neurais profundas para auxílio à tomada de decisão no mercado de ações. 2019. Citado 2 vezes nas páginas 5 e 26.
- BIEHLER, R.; FLEISCHER, Y. Introducing students to machine learning with decision trees using codap and jupyter notebooks. *Teaching Statistics*, v. 43, n. S1, p. S133–S142, 2021. Disponível em: <<https://onlinelibrary.wiley.com/doi/abs/10.1111/test.12279>>. Citado na página 27.
- BLOGRICO. *"Análise Técnica de Ações - Tudo Sobre Gráficos e Tendências"*. 2019. Acessado em 05/05/2021". Disponível em: <<https://blog.rico.com.vc/analise-tecnica-o-que-e>>. Citado 2 vezes nas páginas 8 e 9.
- BRAGA, A.; LUDERMIR, T.; CARVALHO, A. *Redes Neurais Artificiais - Teoria e Aplicações*. 2ª. ed. [S.l.]: LTC, 2007. Citado na página 27.
- BREIMAN, L. Random forests. *Machine Learning*, v. 45, p. 5–32, 2001. Citado 2 vezes nas páginas 16 e 17.
- BROWNLEE, J. *How to Calculate Feature Importance With Python*. 2022. Acessado em 30/11/2022". Disponível em: <<https://machinelearningmastery.com/calculate-feature-importance-with-python/>>. Citado na página 32.
- BURKOV, A. *The Hundred-Page Machine Learning Book*. [S.l.: s.n.], 2019. 27 p. Citado na página 15.

CAMILO, C. O.; SILVA, J. C. Mineração de dados: Conceitos, tarefas, métodos e ferramentas. *Instituto de Informática da Universidade Federal de Goiás*, 2009. Citado na página 12.

DEEPLEARNINGBOOK. *Capítulo 10 – As 10 Principais Arquiteturas de Redes Neurais*. 2020. Acessado em 06/05/2021. Disponível em: <<https://www.deeplearningbook.com.br/as-10-principais-arquiteturas-de-redes-neurais/>>. Citado na página 25.

DIDATICATECH. *Introdução a Redes Neurais e Deep Learning*. 2020. Acessado em 06/05/2021. Disponível em: <<https://didatica.tech/introducao-a-redes-neurais-e-deep-learning/>>. Citado na página 24.

D'ÁVILA, M. Z. *Bolsa conquista 1,5 milhão de novos investidores em 2020, um aumento de 92% no ano*. 2021. Acessado em 03/05/2021. Disponível em: <<https://www.infomoney.com.br/onde-investir/bolsa-conquista-15-milhao-de-novos-investidores-em-2020-um-aumento-de-92-no-ano/>>. Citado na página 1.

ELDER, A. *Como se Transformar em um Operador e Investidor de Sucesso*. 15^a. ed. [S.l.]: Elsevier Editora Ltda, 2004. Citado na página 5.

ELIAS, J. *Com 'boom' de IPOs, bolsa volta a ter mais de 400 empresas –em 1990, eram 615*. 2021. Acessado em 21/06/2021. Disponível em: <<https://www.cnnbrasil.com.br/business/com-boom-de-ipos-bolsa-volta-a-ter-mais-de-400-empresas-em-1990-eram-615/>>. Citado na página 2.

ERNST, D.; GEURTS, P.; WEHENKEL, L. Tree-based batch mode reinforcement learning. *Journal of Machine Learning Research*, v. 6, p. 503–556, 2005. Citado na página 17.

EXAME. *Produtos de inteligência artificial registram aumento de patentes*. 2019. Acessado em 03/05/2021. Disponível em: <<https://exame.com/ciencia/produtos-de-inteligencia-artificial-registram-aumento-de-patentes/>>. Citado na página 2.

FAYYAD, U.; PIATETSKY-SHAPIRO, G.; SMYTH, P. Knowledge discovery and data mining: Towards a unifying framework. *Association for the Advancement of Artificial Intelligence*, 1996. Citado na página 11.

FILHO, D. B. F.; JÚNIOR, J. A. d. S. Desvendando os mistérios do coeficiente de correlação de pearson (r). *Revista Política Hoje*, v. 18, n. 1, p. 115–146, 2009. Citado na página 13.

FIRMINO, D. A. *"Análise de correlação: Passo a passo no Excel e aplicações"*. 2020. Acessado em 29/11/2022. Disponível em: <<https://www.opuspesquisa.com/blog/tecnicas/analise-de-correlacao/>>. Citado na página 13.

FREITAS, S. O. D. Utilização de um modelo baseado em redes neurais para a precificação de opções. 2001. Citado 3 vezes nas páginas 23, 25 e 26.

FREUND, Y.; SCHAPIRE, R. E. A short introduction to boosting. *Journal of Japanese Society for Artificial Intelligence*, 1999. Citado na página 14.

FRIEDMAN, J. H. Stochastic gradient boosting. *Computational Statistics Data Analysis*, v. 38, p. 367–378, 02 2002. Citado na página 18.

- FUNDAMENTUS - Investimento consciente. 2021. Acessado em 10/08/2021. Disponível em: <<https://www.fundamentus.com.br/consciente.php>>. Citado na página 39.
- GAD, A. *A Comprehensive Guide to the Backpropagation Algorithm in Neural Networks*. 2021. Acessado em 06/05/2021. Disponível em: <<https://neptune.ai/blog/backpropagation-algorithm-in-neural-networks-guide>>. Citado na página 26.
- GHORI, K. M. et al. Performance analysis of different types of machine learning classifiers for non-technical loss detection. *IEEE Access*, v. 8, p. 16033–16048, 2020. Citado na página 19.
- GOLDSCHMIDT, R.; PASSOS, E. *Data Mining: Um Guia Prático*. [S.l.]: Elsevier Editora Ltda, 2005. Citado 3 vezes nas páginas 2, 10 e 11.
- GUIMARAES, T. J. R. Identificação de doenças cardíacas a partir de eletrocardiogramas utilizando machine learning. 2019. Citado na página 19.
- HANCOCK, J. T.; KHOSHGOFTAAR, T. M. Catboost for big data: an interdisciplinary review. *Journal of big data*, 2020. Citado na página 19.
- HARRINGTON, P. *Machine Learning in Action*. [S.l.]: Manning, 2012. Citado na página 19.
- HASTIE, T. Ridge regularization: An essential concept in data science. *Technometrics*, Taylor Francis, v. 62, n. 4, p. 426–433, 2020. Disponível em: <<https://doi.org/10.1080/00401706.2020.1791959>>. Citado na página 21.
- HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2ª. ed. [S.l.]: Springer, 2008. Citado na página 18.
- HAYKIN, S. O. *Neural Networks: A Comprehensive Foundation: United States Edition*. 2ª. ed. [S.l.]: Pearson, 1998. Citado 2 vezes nas páginas 22 e 23.
- INFOMONEY. *Análise Fundamentalista de Ações: como identificar empresas sólidas e rentáveis a longo prazo*. 2020. Acessado em 05/05/2021. Disponível em: <<https://www.infomoney.com.br/guias/analise-fundamentalista/>>. Citado na página 10.
- INVESTIDOREMVALOR. "A Teoria Moderna do Portfólio: Em que ela pode ajudar na hora de investir". 2020. Acessado em 03/05/2021". Disponível em: <<https://investidoremvalor.com/teoria-moderna-do-portfolio/>>. Citado na página 7.
- JAVATPOINT. *K-Nearest Neighbor(KNN) Algorithm for Machine Learning*. 2011. Acessado em 30/08/2022. Disponível em: <<https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning>>. Citado na página 20.
- KE, G. et al. Lightgbm: A highly efficient gradient boosting decision tree. In: GUYON, I. et al. (Ed.). *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2017. v. 30. Disponível em: <<https://proceedings.neurips.cc/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf>>. Citado na página 18.
- KUHN, M.; JOHNSON, K. *Applied Predictive Modeling*. [S.l.]: Springer New York, 2013. v. 1. Citado na página 20.

KUMAR, V.; STEINBACH, M.; TAN, P.-N. *Introdução ao Data Mining. Mineração de Dados*. [S.l.]: Ciência Moderna, 2009. Citado na página 11.

LARGHI, N. *Maioria dos novos investidores da bolsa tem foco no longo prazo e aprende com influenciadores*. 2020. Acessado em 03/05/2021. Disponível em: <<https://valorinveste.globo.com/objetivo/hora-de-investir/noticia/2020/12/14/maioria-dos-novos-investidores-da-bolsa-tem-foco-no-longo-prazo-e-aprende-com-influenciadores.ghml>>. Citado na página 1.

LEITE, D. *Teoria Moderna do Portfólio*. 2020. Acessado em 03/05/2021. Disponível em: <<https://www.ligafeausp.com/single-post/2020/05/21/teoria-moderna-do-portfólio>>. Citado 2 vezes nas páginas 6 e 7.

LORENZETT, C. D. C.; TELOCKEN, A. V. Estudo comparativo entre os algoritmos de mineração de dados random forest e j48 na tomada de decisão. 2016. Citado 2 vezes nas páginas 16 e 17.

LUNARDI, A. d. C.; VITERBO, J.; BERNARDINI, F. C. Um levantamento do uso de algoritmos de aprendizado supervisionado em mineração de opiniões. *Proceedings of XII Encontro Nacional de Inteligência Artificial e Computacional-ENIAC*, p. 262–269, 2015. Citado na página 19.

MARKOWITZ, H. Portfolio selection*. *The Journal of Finance*, v. 7, n. 1, 1952. Citado na página 6.

MARKOWITZ, H. *Portfolio Selection: Efficient Diversification of Investments*. [S.l.]: John Wiley & Sons, New-York, 1959. Citado na página 6.

MATPLOTLIB. *Matplotlib: Visualization with Python*. 2022. Acessado em 30/11/2022". Disponível em: <<https://matplotlib.org>>. Citado na página 30.

MEDHAT, W.; HASSAN, A.; KORASHY, H. Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*, v. 5, p. 1093–1113, 2014. Citado na página 21.

MIRANDA, A. P. et al. Sistema de análise de ativos através de redes neurais de múltiplas camadas. *Revista de Administração da UFSM*, v. 5, n. 1, p. 145–162, jun. 2012. ISSN 1983-4659, 1983-4659. Disponível em: <<https://periodicos.ufsm.br/reaufsm/article/view/3993>>. Citado 2 vezes nas páginas 8 e 9.

MONTEIRO, F. F. E. C. Desenvolvimento de um modelo em python usando machine learning para previsão e controle da diluição mineral associada com desmonte em minas a céu aberto. 2022. Citado na página 28.

MOORE, M. *Ações: Quais Comprar e Quando Comprar - Aprenda a Investir Utilizando Análise Fundamentalista com Análise Gráfica*. 1ª. ed. [S.l.]: Elsevier, 2012. ISBN 9788535250749. Citado na página 10.

MULINARI, B. *Numpy Python: O que é, vantagens e tutorial inicial*. 2022. Acessado em 29/11/2022". Disponível em: <<https://harve.com.br/blog/programacao-python-blog/numpy-python-o-que-e-vantagens-e-tutorial-inicial/>>. Citado na página 30.

- MULINARI, B. *Pandas Python: vantagens e como começar*. 2022. Acessado em 29/11/2022". Disponível em: <<https://harve.com.br/blog/programacao-python-blog/pandas-python-vantagens-e-como-comecar/>>. Citado na página 29.
- MUPRHY, J. J. *Technical Analysis of the Financial Markets: A Comprehensive Guide to Trading Methods and Applications*. [S.l.]: Prentice Hall Press, 1999. Citado na página 8.
- PEREIRA, P. M. P. Análise de risco de crédito usando algoritmos de machine learning. 2020. Citado na página 20.
- PINHEIRO, L. "Suporte e resistência: entenda definitivamente estes conceitos de Análise Técnica". 2019. Acessado em 19/10/2022". Disponível em: <<https://blog.neologica.com.br/suporte-e-resistencia-entenda-definitivamente-esses-conceitos-de-analise-tecnica/>>. Citado na página 8.
- PLOTLY. *Plotly Open Source Graphing Library for Python*. 2022. Acessado em 30/11/2022". Disponível em: <<https://plotly.com/python/>>. Citado na página 30.
- PYCARET. *Welcome to PyCaret: An open-source, low-code machine learning library in Python*. 2022. Acessado em 30/11/2022". Disponível em: <<https://pycaret.gitbook.io/docs/>>. Citado na página 32.
- REIS, T. *Os indicadores mais importantes em uma análise*. 2017. Disponível em: <<https://www.suno.com.br/artigos/os-indicadores-mais-importantes-em-uma-analise/>>. Citado na página 1.
- ROCHA, D. S. C.; CARMO, A. C. d.; VASCONCELOS, J. A. d. Máquina de aprendizagem aplicada ao reconhecimento automático de falhas em motores elétricos. *Computer on the Beach*, p. 482–491, 2018. Citado na página 18.
- S, R. A. *The Best Python Pandas Tutorial*. 2022. Acessado em 30/11/2022". Disponível em: <<https://www.simplilearn.com/tutorials/python-tutorial/python-pandas>>. Citado na página 29.
- SANTOS, M. *I Built a Machine Learning Model to Trade Stocks Like Warren Buffett (Part 1)*. 2020. Acessado em 06/05/2021. Disponível em: <<https://medium.com/swlh/teaching-a-machine-to-trade-stocks-like-warren-buffett-part-i-445849b208c6>>. Citado 2 vezes nas páginas 44 e 45.
- SANTOS, V. B. Um ensemble baseado em Árvores de decisão para prever a ocorrência de aglomerados de Ônibus. 2020. Citado na página 19.
- SCIKIT-LEARN. *Metrics and scoring: quantifying the quality of predictions*. 2022. Acessado em 30/11/2022". Disponível em: <https://scikit-learn.org/stable/modules/model_evaluation.html>. Citado na página 32.
- SCIKIT-LEARN. *Model selection and evaluation*. 2022. Acessado em 30/11/2022". Disponível em: <https://scikit-learn.org/stable/model_selection.html#model-selection>. Citado na página 32.
- SCIKIT-LEARN. *Preprocessing data*. 2022. Acessado em 30/11/2022". Disponível em: <<https://scikit-learn.org/stable/modules/preprocessing.html#preprocessing>>. Citado na página 32.

- SCIKIT-LEARN. *Supervised learning*. 2022. Acessado em 30/11/2022". Disponível em: <https://scikit-learn.org/stable/supervised_learning.html#supervised-learning>. Citado na página 31.
- SCIKIT-LEARN. *Tuning the hyper-parameters of an estimator*. 2022. Acessado em 30/11/2022". Disponível em: <https://scikit-learn.org/stable/modules/grid_search.html>. Citado na página 32.
- SEABORN. *Annotated heatmaps*. 2022. Acessado em 30/11/2022". Disponível em: <https://seaborn.pydata.org/examples/spreadsheet_heatmap.html>. Citado na página 31.
- SEABORN. *Seaborn: statistical data visualization*. 2022. Acessado em 30/11/2022". Disponível em: <<https://seaborn.pydata.org>>. Citado na página 30.
- SILVEIRA, D. *Desemprego diante da pandemia bate recorde no Brasil em setembro, aponta IBGE*. 2020. Acessado em 03/05/2021. Disponível em: <<https://g1.globo.com/economia/noticia/2020/10/23/no-de-desempregados-diante-da-pandemia-aumentou-em-34-milhoes-em-cinco-meses-aponta-ibge.ghtml>>. Citado na página 1.
- SOUSA, D. A. d. *Descoberta de exploits usando dados da rede social twitter*. 2020. Citado na página 18.
- SUYKENS, J.; VANDEWALLE, J. Least squares support vector machine classifiers. *Neural Processing Letters*, v. 9, p. 293–300, 06 1999. Citado na página 21.
- SWENSON, D. F. *Pioneering Portfolio Management*. [S.l.]: Free Press, 2009. Citado na página 6.
- TAFNER, M. A. Redes neurais artificiais: Aprendizado e plasticidade. *Revista Cérebro Mente*, 1998. Acessado em 06/05/2021. Disponível em: <<https://cerebromente.org.br/n05/tecnologia/rna.htm>>. Citado na página 23.
- TAURION, C. *BIG DATA*. 1^a. ed. [S.l.]: BRASPORT, 2013. Citado na página 10.
- TERAMACHI, A. G. *Aprendizado de máquina para classificação da desocupação de leitos pós cirúrgicos: Um estudo sobre pacientes com cardiopatias congênitas*. 2020. Citado na página 22.
- TVETER, D. *The Pattern Recognition Basis of Artificial Intelligence*. 1^a. ed. [S.l.]: Wiley-IEEE Computer Society Pr, 1998. Citado 2 vezes nas páginas 13 e 14.
- VAPNIK, V. N. *Statistical Learning Theory*. [S.l.]: Wiley-Interscience, 1998. Citado na página 21.
- VERMA, A. K.; PAL, S.; KUMAR, S. Comparison of skin disease prediction by feature selection using ensemble data mining techniques. *Informatics in Medicine Unlocked*, v. 16, p. 100202, 2019. ISSN 2352-9148. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S2352914819300838>>. Citado na página 18.

VERMA, I. *Introduction to machine learning with Jupyter notebooks*. 2021. Acessado em 29/11/2022". Disponível em: <<https://developers.redhat.com/articles/2021/05/21/introduction-machine-learning-jupyter-notebooks#>>. Citado na página 28.

VOGLINO, E. *O que é Volatilidade do Mercado Financeiro*. 2020. Disponível em: <<https://comoinvestir.thecap.com.br/o-que-e-volatilidade-de-mercado/>>. Citado na página 5.

VOVK, V. Kernel ridge regression. In: _____. *Empirical Inference: Festschrift in Honor of Vladimir N. Vapnik*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. p. 105–116. ISBN 978-3-642-41136-6. Disponível em: <https://doi.org/10.1007/978-3-642-41136-6_11>. Citado na página 21.

WANG, C.; XU, S.; YANG, J. Adaboost algorithm in artificial intelligence for optimizing the iri prediction accuracy of asphalt concrete pavement. *Sensors*, v. 21, n. 17, 2021. ISSN 1424-8220. Disponível em: <<https://www.mdpi.com/1424-8220/21/17/5682>>. Citado na página 15.

XPINVESTIMENTOS. *O que são ações?* 2021. Acessado em 03/05/2021. Disponível em: <<https://www.xpi.com.br/investimentos/acoes/o-que-sao-acoes/?gclid=Cj0KCQiAx9mABhD0ARIsAEfpavQXqpFDQoat9U25phWqImwUocYyCasYvApRKldjV0fpMCPsuL4wcB>>. Citado na página 5.

YAHOO! Finanças. 2021. Acessado em 25/08/2021". Disponível em: <<https://br.financas.yahoo.com>>. Citado na página 39.