

CENTRO UNIVERSITÁRIO SAGRADO CORAÇÃO

Ana Carolina de Oliveira

**INTELIGÊNCIA ARTIFICIAL APLICADA AO
MERCADO FINANCEIRO PARA TOMADA DE
DECISÃO**

BAURU
2021

Ana Carolina de Oliveira

**INTELIGÊNCIA ARTIFICIAL APLICADA AO
MERCADO FINANCEIRO PARA TOMADA DE
DECISÃO**

Monografia de Iniciação Científica apresentada à Pró-Reitoria de Pesquisa e Pós-Graduação como parte dos pré-requisitos para aprovação do conselho, sob orientação do Prof. Me. Patrick Pedreira Silva.

BAURU
2021

Dados Internacionais de Catalogação na Publicação (CIP) de acordo
com ISBD

O48i	<p>Oliveira, Ana Carolina de</p> <p>Inteligência artificial aplicada ao mercado financeiro para tomada de decisão / Ana Carolina de Oliveira. -- 2021. 35f. : il.</p> <p>Orientador: Prof. M.e Patrick Pedreira Silva</p> <p>Monografia (Iniciação Científica em Ciência da Computação) - Centro Universitário Sagrado Coração - UNISAGRADO - Bauru - SP</p> <p>1. Bolsa de Valores. 2. Dados. 3. Investimento. 4. Inteligência Artificial. I. Silva, Patrick Pedreira. II. Título.</p>
------	---

RESUMO

Aspirando-se que o mercado financeiro e a bolsa de valores estão a cada dia sendo ainda mais percorridos, e, além disso, com o avanço tecnológico exponencial, é possível fazer com que a tecnologia auxilie o investimento neste mercado tão vasto. Diante deste cenário, o objetivo desta pesquisa foi utilizar da Mineração de Dados e da Inteligência Artificial para criar conexões entre os dados da bolsa de valores (B3), sendo assim, possível a produção de informações e resultados mais inteligíveis sobre a conduta das ações de cada empresa, visando constatar o risco de investimento, servindo de ferramenta de apoio à tomada de decisão de investidores financeiros. Com isso, o objetivo foi verificar a previsão de cada ação com relação a fechar em alta ou em baixa do qual o algoritmo se mostrou eficiente para a realização das previsões.

Palavras-chave: Bolsa de Valores, Dados, Investimento, Inteligência Artificial.

SUMÁRIO

<u>1. INTRODUÇÃO</u>	4
<u>2. OBJETIVOS</u>	6
<u>2.1. OBJETIVO GERAL</u>	6
<u>2.2. OBJETIVOS ESPECÍFICOS</u>	6
<u>3. REFERENCIAL TEÓRICO</u>	7
<u>3.1. INTELIGÊNCIA ARTIFICIAL</u>	7
<u>3.2. APRENDIZADO DE MÁQUINA</u>	9
<u>4. MATERIAIS E MÉTODOS</u>	10
<u>4.1. FERRAMENTAS UTILIZADAS – PYTHON E JUPYTER NOTEBOOK</u>	11
<u>4.2. HARDWARE</u>	12
<u>5. RESULTADOS</u>	12
<u>5.1. COLETA E CRIAÇÃO DE UMA BASE DE DADOS</u>	12
<u>5.2. ETAPAS DA MINERAÇÃO</u>	14
<u>5.3. SCRIPTS DESENVOLVIDOS</u>	15
<u>5.4. CONSIDERAÇÕES SOBRE OS RESULTADOS</u>	31
<u>6. CONSIDERAÇÕES FINAIS</u>	32
<u>7. REFERÊNCIAS</u>	34

1. INTRODUÇÃO

O mercado brasileiro de ações tem crescido de uma forma bastante significativa, demonstrando o interesse cada vez maior da população por este tema. No final de 2018, o mercado de ações brasileiro somava 813 mil CPFs inscritos. Já no final de 2019 esse número atingiu cerca de 1,5 milhões de investidores. Com isso, o número de cadastros cresceu em quase 700 mil desde o início do ano – o que equivale a uma entrada média de 70 mil pessoas por mês. O mercado financeiro é dinâmico e, diante de tantas especulações, oscilações e fatores macroeconômicos, investir é algo que demanda uma análise ampla. Tantos fatores envolvidos acaba tornando a análise financeira imprescindível para decidir qual estratégia adotar e o que fazer perante os diversos cenários que estão presentes no dia a dia. Sendo assim é necessário que o indivíduo, seja ele pessoa física ou jurídica, observe com cautela os seus investimentos, atualizando constantemente o seu o portfólio de ativos e aumentando as suas chances de obter o melhor rendimento independente das mudanças de cenário. A observação do mercado financeiro mostra que existem indicativos que podem ser observados para classificar qual seria a melhor decisão e quando ela deve ser tomada. Diante disso, o tempo de resposta necessita acompanhar as mudanças do mercado, prevendo uma janela de investimento mais adequada para aquele determinado caso, aproveitando, assim, o fluxo positivo daquela escolha.

Neste sentido, a tecnologia pode ajudar no processo de avaliação de investimentos financeiros, identificando mudanças e indicando o melhor momento para o investimento. Um exemplo seria a área da Inteligência Artificial (IA) que por meio das suas técnicas, como o aprendizado de máquina, busca entender as diversas variações do mercado financeiro.

Com a invenção da IA, a tecnologia passou a contar com a simulação de processos inteligentes que auxiliam no reconhecimento de padrões, na tomada de decisão ou na execução de tarefas repetitivas. (MEDEIROS, 2018, p 18).

Notou-se que essas técnicas podem ser interessantes para descoberta de padrões através de classificações de dados, que na maioria das vezes seria custoso para um ser humano classificar e, conseqüentemente, perdendo o melhor momento para o investimento.

A inteligência artificial tem se integrado cada vez no cotidiano, auxiliando em tarefas complexas do dia a dia, isso quando não é capaz de executá-las por conta própria.

Esse ramo da computação concentra-se em resolver problemas que estão além da capacidade humana, processando grandes quantidades de dados e obtendo resultados antes inatingíveis. A IA tem um papel importante no processo de análise e descoberta de informações.

Com o descobrimento de informações relevante através da análise dos dados extraídos é possível antecipar fatos, desmistificar conceitos e apontar possíveis melhorias para que os resultado sejam diferentes e possivelmente satisfatórios. Sendo assim, este trabalho tem a finalidade de mostrar como a mineração de dados pode ajudar investidores, iniciantes ou não, a analisar o mercado financeiro, especialmente as ações, e aumentar as suas chances de conquistar os proventos de qualidade.

Para isso a área em destaque para análise de dados são as chamadas técnicas de aprendizado de máquina que capacitam agentes inteligentes a se modificarem ao longo de suas interações apresentando o envolvimento com generalizações a partir de experiências. Diferente dos seres humanos que não precisam de um grande número de exemplos para aprender, para aplicar a aprendizagem de máquina é necessário utilizar um grande número de exemplos para garantir a eficiência diminuindo a probabilidade de erro. (LUGER, 2013).

Conforme Russell Norvig (2004, p. 25) destaca, a IA "[...] vai ainda mais além: ele tenta não apenas compreender, mas também construir entidades inteligentes." Com o descobrimento de informações relevante através da análise dos dados extraídos é possível antecipar fatos, desmistificar conceitos e apontar possíveis melhorias para que os resultado sejam diferentes e possivelmente satisfatórios.

Diante deste cenário, esta pesquisa tem a finalidade de atingir um público que esteja interessado em investir, em ações além de desmitificar o atual

mercado de investimentos, simplificando os principais pontos que são necessários saber antes de investir, para que qualquer indivíduo com formação na área ou não, consiga pelo menos ter uma base para começar os seus investimentos. Para isso considera-se a IA uma grande aliada no auxílio à tomada de decisão, pois por meio de suas técnicas é possível analisar, avaliar e classificar diversos dados de modo eficaz e eficiente. Logo, a IA ajudaria a fundamentar e apoiar decisões que não seriam tomadas sem as evidências destacadas pelos algoritmos.

Considera-se que esse trabalho irá ajudar não só a demonstrar a importância da mineração, mas também como ela pode ajudar a melhorar o cenário e a criar oportunidades que sem ela seriam dificilmente notadas e quando notadas, corre-se a possibilidade de não serem implementadas há tempo.

2. OBJETIVOS

2.1. OBJETIVO GERAL

Utilizar técnicas de Inteligência Artificial para minerar os dados da bolsa de valores, através da linguagem de programação Python, para avaliar investimentos no mercado financeiro.

2.2. OBJETIVOS ESPECÍFICOS

- Pesquisar sobre Inteligência Artificial, Aprendizado de Máquina, Mineração de dados, mercado financeiro e bolsa de valores;
- Pesquisar sobre os algoritmos de aprendizado de máquina mais apropriados para realizar a mineração de dados da forma mais adequada aos objetivos da pesquisa;
- Usar a mineração de dados no conjunto de dados recuperados da bolsa de valores através da linguagem de programação Python;
- Aplicar os dados obtidos para aprendizado de máquina para verificar se as ações irão fechar em alta ou em baixa na bolsa de valores;

- Promover a divulgação da experiência e resultados obtidos em eventos técnicos e científicos, publicações correlatas, e participar do Congresso Anual de Iniciação Científica e Desenvolvimento Tecnológico e Inovação do UNISAGRADO.

3. REFERENCIAL TEÓRICO

A seguir serão tratados alguns aspectos teóricos principais referentes à esta pesquisa.

3.1. INTELIGÊNCIA ARTIFICIAL

A espécie Homo sapiens, ou homem sábio, diz que as capacidades mentais vão além dos outros seres vivos, se possibilitando não só perceber, mas também compreender, prever e manipular o ambiente. O estudo dessas capacidades deu origem a diversas áreas científicas, entre elas a Inteligência Artificial. Pode-se definir a inteligência artificial como o estudo da tecnologia que se localiza entre a ciência e a arte, onde o seu principal alvo é resolver problemas de forma semelhante a um ser inteligente. (RUSSELL; NORVIG, 2004).

Tendo em vista o estudo de como funciona o cérebro humano, percebemos que é possível criar uma espécie de “réplica” da nossa capacidade de raciocinar possibilitando um computador a fazer tomadas de decisões, falar, compreender o comportamento humano, entre outras coisas.

De acordo com o pensamento de Coppin (2013) é possível classificar a inteligência artificial não como uma automação simples, entretanto uma tecnologia capaz de usar os melhores métodos para se resolver um problema, usando comportamentos inteligentes processados a partir de declarações podendo ter algum significado.

Ainda segundo este autor a inteligência artificial se trata do estudo dos sistemas que regem como o observador pode parecer inteligente. Logo existem razões para que a IA não seja encarada como uma simples automação, ela em sua plenitude consegue não apenas agir, mas definir a melhor ação utilizando de vários mecanismos possíveis que podem simular a racionalidade e o

pensamento humano, em busca da melhor opção para solucionar problemas, sejam eles simples ou complexos. A IA não se trata apenas de um conjunto de comportamentos programados, mas de métodos baseados no comportamento inteligente, que são processados a partir de declarações que venham a ter algum significado. (COPPIN, 2013).

Utiliza-se a IA em diversas áreas como por exemplo em um jogo de xadrez, onde se aplica a percepção e o aprendizado para executar tarefas específicas, ou para entregar um diagnóstico de doenças. O interessante na IA é a possibilidade de sistematizar a automatizar tarefas que dependem de conhecimento intelectual humano para serem iniciadas, porém para isso é necessário entender todos os meandros da inteligência, para então replicá-la em uma máquina. O filósofo Aristóteles foi um dos pioneiros no estudo do pensamento e do raciocínio, ele dizia que para estruturar argumentos era preciso utilizar padrões de estrutura linguística e também premissas verdadeiras para embasá-los e assim se deu o início do campo de estudo da lógica. (RUSSELL; NORVIG, 2004).

O modo de agir logicamente, refere-se a um ser racional, ou um agente racional que se baseia em inferência corretas, para então agir racionalmente até a conclusão de qual ação é a melhor a se tomar para alcançar o objetivo pretendido. Mesmo assim, não se pode considerar que sempre haverá apenas uma resposta correta para determinado assunto ou cenário, e sim precisa-se salientar que para cada situação pode ocorrer inúmeras ações diferentes que são corretas para se praticar. (RUSSELL; NORVIG, 2004).

Poder-se-ia certamente definir inteligência pelas propriedades que ela exhibe: uma capacidade de lidar com novas situações; a capacidade de solucionar problemas, de responder a questões, de engendrar planos e assim por diante. (COPPIN, 2013, p. 4).

Segundo o livro “Inteligência Artificial: Uma abordagem de aprendizado de máquina”, a capacidade de aprendizado está diretamente ligada com o comportamento inteligente, pois atividades como memorização, observação e exploração são consideradas parte do aprendizado. A aquisição de conhecimento envolve análise de informações, sejam elas coletadas por meio de entrevistas com especialistas ou bancos de dados, que são utilizadas para

descobrir regras quando um certo evento entra em ação. Com isso a complexidade dos problemas, podem ser tratados não apenas pelo especialista, mas esse pode contar com a ajuda de ferramentas computacionais mais sofisticadas que reduzem a necessidade de tarefas simples, possibilitando o profissional a focar em descobertas mais capciosas, pois é poupado da necessidade de levantamentos básicos. (FACELI; LORENA; GAMA; CARVALHO, 2011).

3.2. APRENDIZADO DE MÁQUINA

A IA pode ser classificada e subdividida em diversos campos, mas o campo em questão que tratar-se-á nesse projeto é o campo da Aprendizagem de Máquina, ou AM, como também é conhecida. Esse é um segmento extremamente importante pois se trata de uma relação e concordância de dados em que um sistema absorve para poder aprender algo que possa resultar em uma ação inteligente, porque se trata de um auto aprimoramento, uma vez que a ação é baseada em uma experiência passada. (COPPIN, 2013).

O campo da AM costuma ser dividido em três casos, são eles:

- Supervisionada
- Não Supervisionada
- Por reforço

A aprendizagem supervisionada é aplicada quando o caso é passível de observação, ou seja, o agente pode observar as suas ações e quais são as consequências delas, assim consegue extrair utilizando os métodos da AM supervisionada e consegue evitar ou até prever acontecimentos. Pode-se utilizar como exemplo um jogo de damas, onde exemplo de jogadas e situações do jogo são passadas para o agente juntamente com os melhores movimentos naquele cenário do jogo. (RUSSELL; NORVIG, 2004).

Em casos se saídas pouco observáveis, utiliza-se a aprendizagem não-supervisionada, onde por exemplo um carro autônomo conseguiria distinguir dias de grande congestionamento e pouco congestionamento de forma gradual sem jamais ter recebido um exemplo de nenhum dos casos. (RUSSELL; NORVIG, 2004).

Resume-se que a AM não-supervisionada não aprende o que fazer, por não ter nenhum exemplo correto ou de estado que seja ideal para a situação. (RUSSELL; NORVIG, 2004).

Já a AM por reforço, como o próprio nome já diz, define-se como a aprendizagem que é conquistada a partir do reforço medindo os resultados, ou o sucesso e o fracasso, e os classificando. Porém para se ter sucesso nesse tipo de aprendizagem é necessário que seja interpretada a situação em questão, como em um jogo, que o agente só saberá se perdeu ou ganhou no final da partida, não existe nenhuma informação pré-moldada que o dirige e o conduz a vitória. (RUSSELL; NORVIG, 2004).

Sendo assim, considera-se a aprendizagem como um conjunto de percepções que podem ser usadas não somente para efetuar uma ação, mas também para melhorar uma habilidade. Isso só é possível através da observação das interações dos processos com o ambiente e também da comparação dos resultados com o desempenho desejado. A representação das informações coletadas ou aprendidas, tem um papel relevante para a escolha do algoritmo, pois isso ditará como ele irá funcionar e quais são os resultados que ele deve apresentar. Logo, conclui-se que um requisito importante para a AM, é a escolha do algoritmo ideal para que a solução seja encontrada da maneira mais eficaz e eficiente possível, utilizando da capacidade e dos recursos disponíveis no momento. (FACELI; LORENA; GAMA; CARVALHO, 2011).

4. MATERIAIS E MÉTODOS

Pesquisas realizadas com propósitos acadêmicos tendem, no primeiro momento, a assumir o caráter de pesquisa exploratória, pois é pouco provável que o pesquisador tenha uma definição clara do que irá investigar. Segundo Gil (2010), as pesquisas exploratórias têm por finalidade proporcionar maior familiaridade com o problema, com o intuito de torná-lo mais explícito ou a construir hipóteses.

Diante disso, esse projeto é uma pesquisa exploratória, pois o objetivo é a aplicação de algoritmos de mineração de dados a fim de encontrar padrões,

que possam indicar algum tipo de conhecimento implícito sobre uma base de dados, no caso, dados do mercado financeiro.

O projeto foi desenvolvido em duas etapas: na primeira, foi realizadas pesquisas, estudos e desenvolvimento do embasamento teórico e na segunda etapa foram feitos os testes com a linguagem de programação Python com o objetivo de gerar conhecimento sobre os dados estudados.

Para o desenvolvimento do embasamento teórico e a descrição das atividades, haverá a necessidade de realizar pesquisas sobre:

- a) Conceituação de Inteligência Artificial;
- b) Conceitos e aplicações de aprendizagem de Máquina;
- c) Fases do processo de Descoberta de Conhecimento em Base de Dados;
- d) Estudo da linguagem de programação Python;
- e) Coleta e formação de uma base de dados;
- f) Testes com os modelos de treinamento de Regressão Linear e Rede Neural.

4.1. FERRAMENTAS UTILIZADAS – PYTHON E JUPYTER NOTEBOOK

“Python é uma linguagem de programação que permite trabalhar mais rapidamente e integrar seus sistemas de forma mais eficaz” (PYTHON, 2021, tradução nossa). O ambiente de desenvolvimento escolhido foi o Jupyter Notebook. A escolha da linguagem Python deveu-se a:

- a) Ser uma linguagem de fácil aprendizado.
- b) Bibliotecas com recursos que auxiliam no aprendizado de máquina e na mineração de dados, por exemplo, Pandas.

A escolha do ambiente Jupyter Notebook deveu-se a:

- a) Possibilidade de programação em células.
- b) Possibilidade de trabalhar com arquivos com grandes quantidades de informações.
- c) Possuir uma interface gráfica amigável.
- d) Possuir código aberto.

4.2. HARDWARE

O projeto foi desenvolvido utilizando como principal ferramenta um computador pessoal (PC), com o sistema operacional Windows 10 Pro – 64 bits, processador Intel® Core™ i5 CPU 650 @ 3.20GHz com memória RAM de 4 GB. O fundamento da escolha do computador se deve pela razão de pertencer à pesquisadora, além de, em primeiro momento, suprir as necessidades da pesquisa.

5. RESULTADOS

A seguir são descritos os resultados desta pesquisa de acordo com os passos metodológicos aplicados.

5.1. COLETA E CRIAÇÃO DE UMA BASE DE DADOS

As ações correspondem às menores partes do capital social de uma empresa e o mercado de ações ou mercado financeiro, corresponde à compra e venda dessas ações com a finalidade de investimento.

Este mercado abarca milhares de empresas, gerando um grande banco de dados (big data) que contém dados de cada empresa e suas ações. Minerando esses dados é possível abstrair informações com o desígnio de se obter os indicadores econômicos para se lograr o potencial de crescimento do negócio.

A base de dados para a descoberta de conhecimento foi obtida através de Websites, que trazem o histórico e várias outras informações sobre ações das empresas negociadas na Bolsa de Valores do Brasil. Especificamente, os dados foram coletados do banco de dados da B3 (Brasil Bolsa Balcão) através da CEI (Canal Eletrônico do Investidor), da qual foi retirado o histórico de variação das bolsas de valores das empresas na B3. Foram retirados dessa base de dados as ações do período de janeiro de 2000 até dezembro de 2020, como se pode visualizar nas Figuras 1 e 2.

Figura 1 – Repositório de Dados da B3



Fonte: Elaborada pela Autora.

Figura 2 –
Repositório de Dados da B3



Fonte: Elaborada pela Autora.

Depois de realizado o levantamento dos dados foi necessário adaptá-los para um formato que permita a visualização dos dados contidos no arquivo, portanto, houve a necessidade de convertê-los para o formato XLSX. Posteriormente, para a realização da mineração dos dados, é necessário que ocorra a adaptação do formato para CSV. O resultado foi um arquivo CSV de 462MB (Figura 3).

Figura 3 – Tamanho do Arquivo

Tamanho: 462 MB (485.467.046 bytes)

Fonte: Elaborada pela Autora.

O formato CSV é utilizado como padrão para visualização de dados manipulados pelo Jupyter Notebook. Ele corresponde a um arquivo texto contendo um conjunto de registros, procedido por um pequeno cabeçalho e pela separação de cada dado por um ; (Figura 4).

Figura 4 – Arquivo Final

```

1 data_preço;cod_papel;nome_empresa;preço_abertura;preço_max;preço_min;preço_fechamento;total_negociado;volume_negociado
2 04/01/2000;AVIL4FF;ACOS VILL;35.01;35.00;35.00;35.00;1188;7659
3 05/01/2000;AVIL4FF;ACOS VILL;35.01;35.01;35.00;35.00;644;2254
4 03/01/2000;BRDC3FF;BRADESCO;11.50;11.50;11.00;11.30;62115;88687
5 03/01/2000;BRDC4FF;BRADESCO;14.20;15.00;13.37;13.87;1030652;1457942

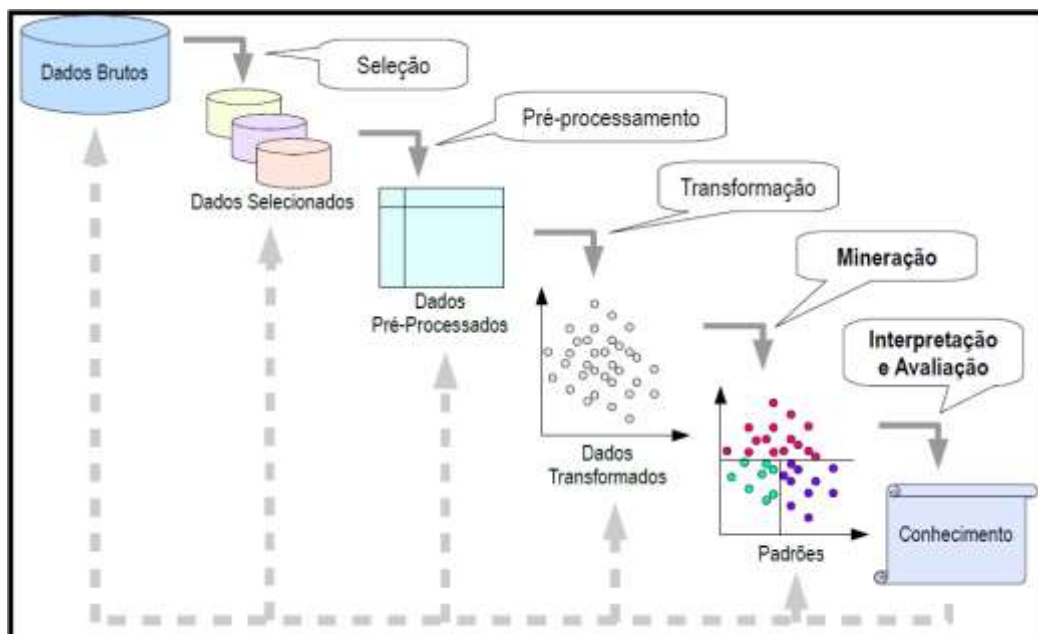
```

Fonte: Elaborada pela Autora.

5.2. ETAPAS DA MINERAÇÃO

Para realizar a Mineração de Dados foi preciso seguir algumas etapas prévias, como se pode observar na Figura 5. As primeiras delas, a seleção e pré-processamento, são descritas a seguir.

Figura 5 - Etapas da Mineração de Dados



Fonte: Elaborada pela Autora.

O primeiro passo para a realização da mineração foi disponibilizar os dados para a seleção e realizar seu pré-processamento, onde o objetivo é melhorar a qualidade, tratando-os para evitar possíveis distorções na extração

do conhecimento. Também foram feitas transformações nessa base, com relação à forma como os dados estão representados a fim de facilitar e retirar possíveis barreiras para a etapa de mineração de dados. Foi selecionada a base de dados criada para este fim com dados coletados da web e importada para o Jupyter Notebook.

O Jupyter Notebook foi utilizado para a criação dos scripts em Python para mineração. A linguagem de programação Python foi utilizada em todas as etapas da mineração, permitindo detectar padrões e criar conexões entre os dados que foram minerados, resultando em respostas e soluções mais precisas sobre o comportamento das ações das empresas selecionadas. Ao final, o objetivo foi verificar a previsão de cada ação com relação a fechar em alta ou em baixa.

5.3. SCRIPTS DESENVOLVIDOS

Primeiramente, importaram-se todas as bibliotecas necessárias (Figura 6). As bibliotecas foram escolhidas com base, principalmente, em suas funções, métodos, modelos de redes neurais e regressão linear. Isso poupou tempo e agilizou o processo de desenvolvimento da aplicação.

Figura 6 – Importação das Bibliotecas

```
#bibliotecas

import numpy as np
import pandas as pd
from sklearn.feature_selection import SelectKBest
from sklearn.model_selection import GridSearchCV
from sklearn.preprocessing import MinMaxScaler
from sklearn import datasets, linear_model
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.neural_network import MLPRegressor
import matplotlib.pyplot as plt
%matplotlib inline
```

Fonte: Elaborada pela Autora.

Realizou-se a importação dos dados em CSV através da função `pd.read_csv`, nativa das bibliotecas Pandas. A função retorna uma estrutura de

dados denominada “data frame”, similar a uma tabela, já com todas as colunas e linhas definidas, como mostram as Figuras 7 e 8.

Figura 7 – Leitura do Arquivo

```
#Lendo arquivos

df = pd.read_csv(r"C:\Users\anacarol\Desktop\Ic\dados.csv", delimiter = ';')
df
```

Fonte: Elaborada pela Autora.

Figura 8 – Leitura do Arquivo

	data_pregao	cod_papel	nome_empresa	preco_abertura	preco_max	preco_min	preco_fechamento	total_negociado	volume_negociado
0	04/01/2000	AVIL4FF	ACOS VILL	35.01	35.01	35.00	35.00	2186.0	7659.0
1	05/01/2000	AVIL4FF	ACOS VILL	35.01	35.01	35.00	35.00	644.0	2254.0
2	03/01/2000	BBDC3FF	BRADESCO	11.50	11.50	11.00	11.30	62110.0	66687.0
3	03/01/2000	BBDC4FF	BRADESCO	14.20	15.00	13.87	13.87	1030952.0	1457942.0
4	04/01/2000	BBDC3	BRADESCO	10.70	10.70	10.11	10.11	69500000.0	71226400.0
...
7680077	22/12/2020	VALED862	VALEE	8.00	8.00	8.00	8.00	3500.0	2800000.0
7680078	29/12/2020	VALED862	VALEE	7.70	7.70	7.70	7.70	6900.0	5313000.0
7680079	30/12/2020	VALEB722	VALEE	15.55	15.55	15.55	15.55	100.0	155500.0
7680080	02/12/2020	VALEB732	VALEE	11.45	11.45	11.45	11.45	5000.0	5725000.0
7680081	03/12/2020	VALEB732	VALEE	9.50	9.50	9.50	9.50	500.0	475000.0

Fonte: Elaborada pela Autora.

Após todos os dados estarem carregados, pede-se ao usuário que escolha qual ação deseja verificar. O programa faz a análise técnica de uma única ação por vez. A escolha da ação é feita pelo usuário através código da ação. No exemplo seguinte, como pode se visualizar nas Figuras 9 e 10, filtra-se todas as ações do Banco Itaú (código “itub4”) desde o ano de 2010 (quando a ação começa a ser comercializada na bolsa) até o ano de 2020.

Figura 9 – Escolha da Ação

```
#inserindo código da ação desejada

codigo_valido = False
while not codigo_valido:
    cod = input("Digite o código da ação desejada: ")
    cod = cod.upper()
    df_cod = df[df['cod_papel'] == cod]
    codigo_valido = bool(df['cod_papel'].str.contains(cod).any())
    if (not codigo_valido):
        print('O código informado não é valido. Digite novamente')
df_cod
```

Digite o código da ação desejada: itub4

Fonte: Elaborada pela Autora.

Figura 10 – Escolha da ação

	data_pregao	cod_papel	nome_empresa	preco_abertura	preco_max	preco_min	preco_fechamento	total_negociado	volume_negociado
296081	04/01/2010	ITUB4	ITAUUNIBANCO	38.89	40.40	38.89	40.12	5441800.0	2.174658e+10
297221	05/01/2010	ITUB4	ITAUUNIBANCO	40.22	40.38	39.90	40.38	4005000.0	1.008022e+10
298453	06/01/2010	ITUB4	ITAUUNIBANCO	40.25	40.49	39.70	40.03	4863900.0	1.943806e+10
299707	07/01/2010	ITUB4	ITAUUNIBANCO	39.75	40.05	39.55	39.62	4655300.0	1.847602e+10
300917	08/01/2010	ITUB4	ITAUUNIBANCO	39.78	39.78	38.92	39.02	4573600.0	1.799907e+10
...
7574354	17/12/2020	ITUB4	ITAUUNIBANCO	31.89	32.15	31.57	31.71	17881500.0	5.882903e+10
7574355	18/12/2020	ITUB4	ITAUUNIBANCO	31.72	32.12	31.51	31.66	31714100.0	1.007450e+11
7574356	21/12/2020	ITUB4	ITAUUNIBANCO	31.05	31.50	30.91	31.10	20014600.0	8.122029e+10
7574357	22/12/2020	ITUB4	ITAUUNIBANCO	31.17	31.65	31.15	31.25	19659000.0	6.169132e+10
7574358	23/12/2020	ITUB4	ITAUUNIBANCO	31.32	32.05	31.27	31.94	19959200.0	6.356557e+10

Fonte: Elaborada pela Autora.

Posteriormente é necessário fazer uma alteração no tipo dos dados, para que o script interprete corretamente as ações subsequentes. A data, por exemplo, vem em formato de texto então para trabalhá-la de forma adequada, como uma data, faz-se a conversão do tipo texto para o tipo datetime64 (Figuras 11 e 12).

Figura 11 – Alteração do Tipo do Dado.

```
#alterando o tipo de variável

df.dtypes
df['data_pregao'] = pd.to_datetime(df['data_pregao'], format = '%d/%m/%y', exact=False)
df.dtypes
```

Fonte: Elaborada pela Autora.

Figura 12 – Alteração do Tipo do Dado

```

data_pregao      datetime64[ns]
cod_papel        object
nome_empresa     object
preco_abertura   float64
preco_max        float64
preco_min        float64
preco_fechamento float64
total_negociado  float64
volume_negociado float64
dtype: object

```

Fonte: Elaborada pela Autora.

Como é necessário criar mais algumas métricas para que a previsão seja eficiente, calcula-se as médias móveis dos valores dos preços de fechamento após 5 e 21 dias, chamadas mm5d e mm21d como demonstrado nas Figuras 13 e 14.

Figura 13 – Criando Médias Móveis

```

#criando médias móveis

df_cod['mm5d'] = df_cod['preco_fechamento'].rolling(5).mean()
df_cod['mm21d'] = df_cod['preco_fechamento'].rolling(21).mean()
df_cod

```

Fonte: Elaborada pela Autora.

Figura 14 – Criando Médias Móveis

	data_pregao	cod_papel	nome_empresa	preco_abertura	preco_max	preco_min	preco_fechamento	total_negociado	volume_negociado	mm5d	mm21d
3081	04/01/2010	ITUB4	ITAUNIBANCO	38.89	40.49	36.89	40.12	5441000.0	2.174058e+10	NaN	NaN
7221	05/01/2010	ITUB4	ITAUNIBANCO	40.22	40.38	39.90	40.38	4005000.0	1.608022e+10	NaN	NaN
8453	06/01/2010	ITUB4	ITAUNIBANCO	40.25	40.49	39.70	40.83	4893000.0	1.943886e+10	NaN	NaN
9707	07/01/2010	ITUB4	ITAUNIBANCO	39.75	40.05	39.55	39.62	4855300.0	1.847002e+10	NaN	NaN
3917	08/01/2010	ITUB4	ITAUNIBANCO	39.78	39.78	38.82	39.82	4573000.0	1.790907e+10	39.834	NaN
...
4354	17/12/2020	ITUB4	ITAUNIBANCO	31.89	32.15	31.57	31.71	17081500.0	5.682963e+10	30.954	30.517619
4355	18/12/2020	ITUB4	ITAUNIBANCO	31.72	32.12	31.51	31.66	31714100.0	1.907450e+11	31.184	30.610476
4356	21/12/2020	ITUB4	ITAUNIBANCO	31.05	31.50	30.91	31.10	26014800.0	8.122829e+10	31.316	30.693333
4357	22/12/2020	ITUB4	ITAUNIBANCO	31.17	31.85	31.15	31.25	19659000.0	6.160132e+10	31.494	30.812857
4358	23/12/2020	ITUB4	ITAUNIBANCO	31.32	32.05	31.27	31.94	19859200.0	6.358557e+10	31.532	30.854762

Fonte: Elaborada pela Autora.

Como o objetivo é a previsão do próximo dia é necessário o uso da função Shift para deslocar o índice do DataFrame um dia para frente (Figuras 15 e 16).

Figura 15 – Empurrando Dados

```
#empurrando os valores das ações

df_cod['preco_fechamento'] = df_cod['preco_fechamento'].shift(-1)
df_cod
```

Fonte: Elaborada pela Autora.

Figura 16 – Empurrando Dados

	data_pregao	cod_papel	nome_empresa	preco_abertura	preco_max	preco_min	preco_fechamento	total_negociado	volume_negociado	mm5d	mm21d
3081	04/01/2010	ITUB4	ITAUNIBANCO	38.89	40.40	38.89	40.38	5441600.0	2.174658e+10	NaN	NaN
7221	05/01/2010	ITUB4	ITAUNIBANCO	40.22	40.38	39.90	40.93	4005000.0	1.608622e+10	NaN	NaN
3453	06/01/2010	ITUB4	ITAUNIBANCO	40.25	40.40	39.70	39.62	4863900.0	1.043886e+10	NaN	NaN
3707	07/01/2010	ITUB4	ITAUNIBANCO	39.75	40.05	39.55	39.02	4655300.0	1.847602e+10	NaN	NaN
3917	08/01/2010	ITUB4	ITAUNIBANCO	39.78	39.78	38.92	38.68	4573600.0	1.790907e+10	39.834	NaN
...
4354	17/12/2020	ITUB4	ITAUNIBANCO	31.89	32.15	31.57	31.86	17861500.0	5.682983e+10	30.904	30.517619
4305	18/12/2020	ITUB4	ITAUNIBANCO	31.72	32.12	31.51	31.10	31714100.0	1.007450e+11	31.184	30.610476
4356	21/12/2020	ITUB4	ITAUNIBANCO	31.05	31.50	30.91	31.25	26014600.0	8.122629e+10	31.316	30.693333
4357	22/12/2020	ITUB4	ITAUNIBANCO	31.17	31.65	31.15	31.94	19659000.0	6.169132e+10	31.494	30.812857
4358	23/12/2020	ITUB4	ITAUNIBANCO	31.32	32.05	31.27	NaN	19959200.0	6.358557e+10	31.532	30.954762

Fonte: Elaborada pela Autora.

Para que a precisão do modelo não seja prejudicada, remove-se os valores nulos do DataFrame, como demonstrado nas Figuras 17 e 18.

Figura 17 – Retirando Dados Nulos

```
#retirando dados nulos

df_cod.dropna(inplace=True)
df_cod
```

Fonte: Elaborada pela Autora.

Figura 18 – Retirando Dados Nulos

	data_pregao	cod_papel	nome_empresa	preco_abertura	preco_max	preco_min	preco_fechamento	total_negociado	volume_negociado	mm5d	mm21d
1536	02/02/2010	ITUB4	ITAUNIBANCO	37.29	37.67	36.82	37.14	4581000.0	1.710181e+10	36.590	37.751428
2800	03/02/2010	ITUB4	ITAUNIBANCO	37.35	37.58	36.71	35.71	3656800.0	1.319585e+10	36.834	37.609524
4114	04/02/2010	ITUB4	ITAUNIBANCO	36.70	36.90	35.29	34.80	5387000.0	1.937636e+10	36.618	37.387143
5455	05/02/2010	ITUB4	ITAUNIBANCO	35.33	36.07	33.81	34.70	9088900.0	3.150582e+10	30.352	37.138995
3774	06/02/2010	ITUB4	ITAUNIBANCO	34.89	35.21	34.42	36.25	5372600.0	1.989791e+10	35.930	36.903810
...
4353	16/12/2020	ITUB4	ITAUNIBANCO	31.24	32.18	31.02	31.71	41207900.0	1.303410e+11	30.554	30.383810
4354	17/12/2020	ITUB4	ITAUNIBANCO	31.89	32.15	31.57	31.66	17861500.0	5.682983e+10	30.904	30.517619
4355	18/12/2020	ITUB4	ITAUNIBANCO	31.72	32.12	31.51	31.10	31714100.0	1.007450e+11	31.184	30.610476
4356	21/12/2020	ITUB4	ITAUNIBANCO	31.05	31.50	30.91	31.25	26014600.0	8.122629e+10	31.316	30.693333
4357	22/12/2020	ITUB4	ITAUNIBANCO	31.17	31.65	31.15	31.94	19659000.0	6.169132e+10	31.494	30.812857

Fonte: Elaborada pela Autora.

Para a realização da previsão propriamente dita, verifica-se a quantidade de linhas do DataFrame e se divide este em três conjuntos: treino, teste e validação (Figura 19).

Figura 19 – Verificando Quantidade de Linhas

```
#verificando quantidade de Linhas

qtd_linhas = len(df_cod)
qtd_linhas_treino = round((qtd_linhas * 0.75) - 0.5)
qtd_linhas_teste = round((qtd_linhas * 0.99) - 0.5)

info = {
    f" linhas treino = 0 : {qtd_linhas_treino}"
    f" linhas teste = {qtd_linhas_treino} : {qtd_linhas_teste}"
    f" linhas validação = {qtd_linhas_teste} : {qtd_linhas}"
}

info

{' linhas treino = 0 : 2173 linhas teste = 2173 : 2869 linhas validação = 2869 : 2898'}
```

Fonte: Elaborada pela Autora.

Posteriormente à separação das linhas refaz-se os índices do DataFrame para que a Inteligência Artificial trabalhe nas linhas corretas (Figuras 20 e 21).

Figura 20 – Reindexando DataFrame

```
#reindexando dataframe

df_cod = df_cod.reset_index(drop = True)
df_cod
```

Fonte: Elaborada pela Autora.

Figura 21 – Reindexando DataFrame

	data_pregao	cod_papel	nome_empresa	preco_abertura	preco_max	preco_min	preco_fechamento	total_negociado	volume_negociado	mm5d	mes21
0	02/02/2010	ITUB4	ITAUNIBANCO	37.29	37.57	36.82	37.14	4581000.0	1.710181e+10	36.590	37.75142
1	03/02/2010	ITUB4	ITAUNIBANCO	37.35	37.58	36.71	35.71	3558800.0	1.319585e+10	36.854	37.60952
2	04/02/2010	ITUB4	ITAUNIBANCO	36.70	36.90	35.29	34.80	5397000.0	1.937930e+10	36.618	37.38714
3	05/02/2010	ITUB4	ITAUNIBANCO	35.33	36.07	33.81	34.70	9086900.0	3.150582e+10	36.352	37.13809
4	06/02/2010	ITUB4	ITAUNIBANCO	34.89	35.21	34.42	36.25	5372600.0	1.880701e+10	35.930	36.99381
...
2893	16/12/2020	ITUB4	ITAUNIBANCO	31.24	32.18	31.02	31.71	41297900.0	1.303410e+11	30.554	30.38381
2894	17/12/2020	ITUB4	ITAUNIBANCO	31.89	32.15	31.57	31.86	17981500.0	5.682983e+10	30.924	30.51781
2895	18/12/2020	ITUB4	ITAUNIBANCO	31.72	32.12	31.51	31.10	31714100.0	1.007450e+11	31.184	30.81047
2896	21/12/2020	ITUB4	ITAUNIBANCO	31.05	31.50	30.91	31.25	26014600.0	8.122029e+10	31.316	30.69333
2897	22/12/2020	ITUB4	ITAUNIBANCO	31.17	31.85	31.15	31.94	19658000.0	6.180132e+10	31.484	30.81285

Fonte: Elaborada pela Autora.

Separa-se então as Features das Labels. Features são as entradas utilizadas para realizar a previsão. As Labels são os resultados desejados. A separação é feita removendo da lista de Features as informações que não serão utilizadas através da função Drop. Posteriormente são adicionadas as informações necessárias na lista de Labels (Figura 22).

Figura 22 – Separando Features e Labels

```
#separando features e labels
features = df_cod.drop(['cod_papel', 'nome_empresa', 'data_pregao', 'preco_fechamento'], 1)
labels = df_cod['preco_fechamento']
```

Fonte: Elaborada pela Autora.

Com as Features separadas é necessário verificar quais são as melhores para serem usadas. A seleção é feita através da função SelectKBest informando com scores quais são as melhores Features a serem usadas (Figuras 23 e 24). Features classificadas como ruins são removidas (Figura 25).

Figura 23 – Escolhendo as Melhores Features

```
#Escolhendo as melhores features com Kbest

features_list = ('preco_abertura', 'qtd_neg_efetuados', 'volume_titulos_neg', 'mm5d', 'mm21d')

k_best_features = SelectKBest(k = 'all')
k_best_features.fit_transform(features, labels)
k_best_features_scores = k_best_features.scores_
raw_pairs = zip(features_list[1:], k_best_features_scores)
ordered_pairs = list(reversed(sorted(raw_pairs, key = lambda x: x[1])))

k_best_features_final = dict(ordered_pairs[:15])
k_best_features = k_best_features_final.keys()
print ('')
print ('Melhores features:')
print (k_best_features_final)
```

Fonte: Elaborada pela Autora.

Figura 24 – Escolhendo as Melhores Features

```
Melhores features:
{'volume_titulos_neg': 91.15712357240304, 'mm5d': 87.53600128851702, 'qtd_neg_efetuados': 71.5168433644155, 'mm21d': 2.56003004
53829397}
```

Fonte: Elaborada pela Autora.

Figura 25 – Separando as Features Escolhidas

```
#separando as features escolhidas

features = df_cod.drop(['cod_papel', 'nome_empresa', 'data_pregao', 'preco_fechamento', 'preco_abertura', 'mm21d'], 1)
```

Fonte: Elaborada pela Autora.

Como existem valores diferentes é necessário normalizar a entrada destes valores e gerar um novo padrão a fim de evitar que um valor maior seja considerado melhor somente por ser maior, com isso se utiliza a função MinMaxScaler para obter dados que podem ser melhores analisados (Figuras 26 e 27).

Figura 26 – Normalizando Dados e Gerando Novo Padrão

```
#normalizando os dados de entrada (features)

#gerando novo padrão
scaler = MinMaxScaler().fit(features)
features_scale = scaler.transform(features)

print ('Features: ', features_scale.shape)
print (features_scale)
```

Fonte: Elaborada pela Autora.

Figura 27 – Normalizando Dados e Gerando Novo Padrão

```
Features: (2898, 5)
[[0.5081607  0.5157927  0.0379043  0.04220574 0.48082156]
 [0.50533584 0.5124195  0.02758956 0.03046365 0.48937411]
 [0.48399247 0.46887458 0.04612225 0.04904946 0.48172865]
 ...
 [0.33396108 0.35295921 0.31116232 0.29365319 0.30568874]
 [0.31450094 0.33455995 0.25376253 0.23497621 0.30996501]
 [0.31920904 0.34191966 0.18975515 0.17625033 0.3157315 ]]
```

Fonte: Elaborada pela Autora.

Em seguida os dados são separados para que seja realizada a previsão (Figura 28).

Figura 28 – Separando Dados

```
#separa os dados de treino, teste e validação

X_train = features_scale[:qtd_linhas_treino]
X_test = features_scale[qtd_linhas_treino:qtd_linhas_teste]

y_train = labels[:qtd_linhas_treino]
y_test = labels[qtd_linhas_treino:qtd_linhas_teste]

print( len(X_train), len(y_train))

print (len(X_test), len(y_test))

2173 2173
696 696
```

Fonte: Elaborada pela Autora.

Para realizar a previsão foram selecionados três métodos de Aprendizado de Máquina: Regressão Linear, Rede Neural e Rede Neural com HyperParameters. A seleção foi feita com base na análise do Coeficiente de Determinação de cada método.

A Regressão Linear gera uma equação que encontra uma linha que representa melhor as variáveis de entrada com as variáveis de saída (Figura 29).

Rede Neural são um conjunto de funções matemáticas que simulam o funcionamento do cérebro humano (Figura 30).

A Rede Neural com HyperParameters funciona como a Rede Neural, porém já tem os parâmetros pré-definidos (Figura 31).

Figura 29 – Regressão Linear

```
#treinamento usando regressão linear

lr = linear_model.LinearRegression()
lr.fit(X_train, y_train)
pred = lr.predict(X_test)
cd = r2_score(y_test, pred)

f'Coeficiência de determinação:{cd * 100:.2f}'

'Coeficiência de determinação:96.53'
```

Fonte: Elaborada pela Autora.

Figura 30 – Rede Neural

```
#rede neural

rn = MLPRegressor(max_iter=2000)

rn.fit(X_train, y_train)
pred = rn.predict(X_test)

cd = rn.score(X_test, y_test)

f'Coeficiente de determinação:{cd * 100:.2f}'

'Coeficiente de determinação:96.18'
```

Fonte: Elaborada pela Autora.

Figura 31 – Rede Neural com HyperParameters

```

#rede neural com ajuste hyper parameters

rn = MLPRegressor()

parameter_space = {
    'hidden_layer_sizes': [(i,) for i in list(range(1,21))],
    'activation': ['tanh', 'relu'],
    'solver': ['sgd', 'adam', 'lbfgs'],
    'alpha': [0.0001, 0.05],
    'learning_rate': ['constant', 'adaptive'],
}

search = GridSearchCV(rn, parameter_space, n_jobs=-1, cv=5)

search.fit(X_train,y_train)
clf = search.best_estimator_
pred = search.predict(X_test)

cd = search.score(X_test, y_test)

f'Coeficiente de determinação:{cd * 100:.2f}'

'Coeficiente de determinação:96.46'

```

Fonte: Elaborada pela Autora.

Sendo o melhor Coeficiente de Determinação o da Regressão Linear, realiza-se então a previsão dos valores dos Preços de Fechamento (Figuras 32 33).

Figura 32 – Executando a Previsão

```
#executando a previsão

previsao = features_scale[qtd_linhas_teste:qtd_linhas]

data_pregao_full = df_cod['data_pregao']
data_pregao = data_pregao_full[qtd_linhas_teste:qtd_linhas]

res_full = df_cod['preco_fechamento']
res = res_full[qtd_linhas_teste:qtd_linhas]

ras_full = df_cod['preco_abertura']
ras = ras_full[qtd_linhas_teste:qtd_linhas]

pred = lr.predict(previsao)

df = pd.DataFrame({'data_pregao':data_pregao, 'abertura':ras,'real':res, 'previsao':pred})
df['real'] = df['real'].shift(+1)

df.set_index('data_pregao', inplace=True)

print(df)
```

Fonte: Elaborada pela Autora.

Figura 33 – Executando a Previsão

	abertura	real	previsao
data_pregao			
17/11/2020	29.37	NaN	29.361510
18/11/2020	29.43	28.77	29.259662
19/11/2020	28.90	28.81	29.125725
20/11/2020	28.72	28.55	28.756340
23/11/2020	28.85	28.90	29.047426
24/11/2020	29.10	29.71	29.769927
25/11/2020	29.62	29.36	29.474134
26/11/2020	29.29	28.74	29.076576
27/11/2020	29.05	28.96	29.462512
30/11/2020	28.97	28.56	29.013080
28/12/2020	32.22	32.17	32.267297
29/12/2020	32.27	32.10	32.205142
30/12/2020	32.20	31.63	32.038531
09/12/2020	30.39	30.54	30.565373
10/12/2020	30.76	31.55	31.320679
11/12/2020	31.17	31.30	31.195008
14/12/2020	31.47	30.88	31.314798
15/12/2020	30.97	31.16	31.207600
01/12/2020	29.05	29.75	29.846634
02/12/2020	29.92	29.98	29.920468
03/12/2020	30.14	29.96	30.487466
04/12/2020	30.22	30.26	30.111804
07/12/2020	30.25	30.44	30.637693
08/12/2020	30.44	30.36	30.744636
16/12/2020	31.24	31.75	31.695323
17/12/2020	31.89	31.71	31.905514
18/12/2020	31.72	31.66	31.865777
21/12/2020	31.05	31.10	31.253069
22/12/2020	31.17	31.25	31.436750

Fonte: Elaborada pela Autora.

Com a previsão de valores é possível verificar se no dia a ação fechou em um valor mais baixo ou mais alto que o valor de abertura do dia. Em caso de alta, resulta-se "verde", já em caso de baixa resulta-se em "vermelho".

Compara-se então o valor do Preço de Abertura com o valor real do Preço de Fechamento (comp_real), posteriormente o valor do Preço de Abertura com o valor do Preço de Fechamento que foi previsto (comp_previsao), posteriormente é feita a comparação entre “comp_real” e “comp_previsao” para verificar o acerto (comp). Conforme Figura 34.

Figura 24 – Verificando Valores em Relação ao Preço de Abertura

```
#verificando se o candle do dia é verde ou vermelho

#valor real
df['comp_real'] = np.where(df['real'] < df['abertura'],'vermelho','verde')

#valor previsto
df['comp_previsao'] = np.where(df['abertura'] < df['previsao'],'verde','vermelho')

#comparação
df['comp'] = np.where(df['comp_previsao'] == df['comp_real'],'igual','diferente')
```

Fonte: Elaborada pela Autora.

Também é possível fazer análise se a ação teve um dia de alta ou de baixa na bolsa, sendo primeiramente entre o Preço de Fechamento real (verif_real), posteriormente entre o Preço de Fechamento previsto (verif_previsao) e por fim a comparação entre os dois para verificação dos acertos (verif_comp). Conforme Figura 35.

Figura 35 – Verificando Dia de Alta ou Baixa

```
#verificando se o dia foi de alta ou baixa na bolsa

#valor real
novas_linhas = []
linha_anterior = df['real'][0]
for linha in df['real']:
    if(linha_anterior != linha):
        if(linha_anterior < linha):
            novas_linhas.append('alto')
        else:
            novas_linhas.append('baixo')
    linha_anterior = linha
df['verif_real'] = novas_linhas

#valor previsto
novas_linhas = []
linha_anterior = df['previsao'][0]
for linha in df['previsao']:
    if(linha_anterior < linha):
        novas_linhas.append('alto')
    else:
        novas_linhas.append('baixo')
    linha_anterior = linha
df['verif_previsao'] = novas_linhas

#comparação
df['verif_comp'] = np.where(df['verif_real'] == df['verif_previsao'],'igual','diferente')
```

Fonte: Elaborada pela Autora.

É calculado então o Desvio Padrão para verificar o índice de erro da previsão (Figura 36).

Figura 36 – Desvio Padrão

```
#calculando desvio padrão
df['diferenca'] = 0.0
total_linhas = range(0,len(df['real']))
for i in total_linhas:
    df['diferenca'][i] = abs(df['real'][i] - df['previsao'][i])
print(df['diferenca'].std())
0.15683596989270487
```

Fonte: Elaborada pela Autora.

Nas Figuras 37, 38 e 39 é possível verificar todas as análises no DataFrame final.

Figura 37 – DataFrame Final

```
#dataframe final
print(df)
```

Fonte: Elaborada pela Autora.

Figura 38 – DataFrame Final

	abertura	real	previsao	comp_real	comp_previsao	comp
data_pregao						
17/11/2020	29.37	NaN	29.361510	verde	vermelho	diferente
18/11/2020	29.43	28.77	29.259662	vermelho	vermelho	igual
19/11/2020	28.90	28.81	29.125725	vermelho	verde	diferente
20/11/2020	28.72	28.55	28.756340	vermelho	verde	diferente
23/11/2020	28.85	28.90	29.047426	verde	verde	igual
24/11/2020	29.10	29.71	29.769927	verde	verde	igual
25/11/2020	29.62	29.36	29.474134	vermelho	vermelho	igual
26/11/2020	29.29	28.74	29.076576	vermelho	vermelho	igual
27/11/2020	29.05	28.96	29.462512	vermelho	verde	diferente
30/11/2020	28.97	28.56	29.013080	vermelho	verde	diferente
28/12/2020	32.22	32.17	32.267297	vermelho	verde	diferente
29/12/2020	32.27	32.10	32.205142	vermelho	vermelho	igual
30/12/2020	32.20	31.63	32.038531	vermelho	vermelho	igual
09/12/2020	30.39	30.54	30.565373	verde	verde	igual
10/12/2020	30.76	31.55	31.320679	verde	verde	igual
11/12/2020	31.17	31.30	31.195008	verde	verde	igual
14/12/2020	31.47	30.88	31.314798	vermelho	vermelho	igual
15/12/2020	30.97	31.16	31.207600	verde	verde	igual
01/12/2020	29.05	29.75	29.846634	verde	verde	igual
02/12/2020	29.92	29.98	29.920468	verde	verde	igual
03/12/2020	30.14	29.96	30.487466	vermelho	verde	diferente
04/12/2020	30.22	30.26	30.111804	verde	vermelho	diferente
07/12/2020	30.25	30.44	30.637693	verde	verde	igual
08/12/2020	30.44	30.36	30.744636	vermelho	verde	diferente
16/12/2020	31.24	31.75	31.695323	verde	verde	igual
17/12/2020	31.89	31.71	31.905514	vermelho	verde	diferente
18/12/2020	31.72	31.66	31.865777	vermelho	verde	diferente
21/12/2020	31.05	31.10	31.253069	verde	verde	igual
22/12/2020	31.17	31.25	31.436750	verde	verde	igual

Fonte: Elaborada pela Autora.

Figura 39 – DataFrame Final

	verif_real	verif_previsao	verif_comp
data_pregao			
17/11/2020	baixo	baixo	igual
18/11/2020	baixo	baixo	igual
19/11/2020	alto	baixo	diferente
20/11/2020	baixo	baixo	igual
23/11/2020	alto	alto	igual
24/11/2020	alto	alto	igual
25/11/2020	baixo	baixo	igual
26/11/2020	baixo	baixo	igual
27/11/2020	alto	alto	igual
30/11/2020	baixo	baixo	igual
28/12/2020	alto	alto	igual
29/12/2020	baixo	baixo	igual
30/12/2020	baixo	baixo	igual
09/12/2020	baixo	baixo	igual
10/12/2020	alto	alto	igual
11/12/2020	baixo	baixo	igual
14/12/2020	baixo	alto	diferente
15/12/2020	alto	baixo	diferente
01/12/2020	baixo	baixo	igual
02/12/2020	alto	alto	igual
03/12/2020	baixo	alto	diferente
04/12/2020	alto	baixo	diferente
07/12/2020	alto	alto	igual
08/12/2020	baixo	alto	diferente
16/12/2020	alto	alto	igual
17/12/2020	baixo	alto	diferente
18/12/2020	baixo	baixo	igual
21/12/2020	baixo	baixo	igual
22/12/2020	alto	alto	igual

Fonte: Elaborada pela Autora.

5.4. CONSIDERAÇÕES SOBRE OS RESULTADOS

O intuito do projeto era definir se o algoritmo seria capaz de prever as valorizações ou desvalorizações das ações de uma determinada entidade no mercado financeiro. O algoritmo apresentou uma medida de dispersão de aproximadamente 0.156 centavos.

O mercado financeiro é extremamente volátil e complexo. Desastres naturais, ações e falas de políticos e empresários e entre outros, são capazes de influenciar os mercados financeiros pelo mundo todo e, graças a tecnologia atual, quase que instantaneamente. Levando isso em conta, a taxa de acerto

sobre as valorizações e desvalorizações apresenta razoável acurácia e pode sim ajudar positivamente pessoas e/ou entidades a tomarem melhores decisões sobre seus investimentos.

6. CONSIDERAÇÕES FINAIS

O algoritmo desenvolvido, mostra-se com uma eficiência satisfatória na previsão de valorizações ou desvalorizações de ações no mercado financeiro. Acredita-se que há potencial para auxiliar nas tomadas de decisão referentes a investimentos para todos os públicos, ou seja, tanto para investidores independentes quanto para grandes entidades e corporações, já que seu uso é simples e não exige grandes requisitos de hardware, rede, etc. Por ter sido desenvolvido com a linguagem Python, ainda há uma grande portabilidade da aplicação, isto é, vários dispositivos e sistemas operacionais podem executar o algoritmo sem grandes dificuldades.

Para trabalhos futuros que envolvam a previsão dos valores das ações, sugere-se o uso de artifícios mais complexos e elaborados além do histórico das ações. Será necessário levar em consideração uma série de aspectos não financeiros e imprevisíveis como política, situação social, aprovação - ou desaprovação - do público consumidor etc. O mercado financeiro é extremamente volátil e complexo, porém é plenamente possível, com o estudo e aplicação de técnicas corretas, desenvolver algoritmos que conseguem prever com eficiência satisfatória os estados futuros que ações irão assumir.

A execução desse projeto desenvolveu o conhecimento em Python. Esta linguagem demonstra-se cada vez mais presente em todos os ramos da computação, indo desde aplicações utilitárias cotidianas até processamento de imagens e inteligência artificial. Além da linguagem Python propriamente dita, também houve proveito no aprendizado da ferramenta Jupyter Notebook que além ser de uma ótima ferramenta para aprendizado em Python, ainda auxilia na organização do código, visualização de resultados e compartilhamento.

Também foram desenvolvidos vários conceitos das áreas de Machine Learning e Big Data, ambas de suma importância para a computação atual. A

manipulação e o processamento de grandes quantidades de dados são habilidades necessárias e cada vez mais importantes para a geração de informação útil para resolução de problemas.

Conclui-se que o projeto foi de grande proveito, gerando resultados que podem auxiliar a tomada de decisão ao investir no mercado financeiro e também contribuiu para o desenvolvimento acadêmico, profissional e intelectual da autora, desenvolvendo diversas novas habilidades e conhecimentos.

7. REFERÊNCIAS

ABOUT. Python, 2021. Disponível em: < <https://www.python.org/about/>>. Acesso em: 15 de fev. de 2021.

COMO FUNCIONA O MERCADO DE AÇÕES? ENTENDA!. André Bona, 2019. Disponível em: <<https://andrebona.com.br/como-funciona-o-mercado-de-acoes-entenda/>>. Acesso em: 26 de fev. de 2020.

FACELI, K.; LORENA, A. C.; GAMA, J.; CARVALHO, A. C. P. L. F. de. Inteligência Artificial: Uma Abordagem de Aprendizado de Máquina. Rio de Janeiro. LTC, 2011.

GAIDARGI, J. Mineração de Dados: o que é, para que serve e como fazer?. Infonova, 2019. Disponível em: <<https://www.infonova.com.br/negocios/mineracao-de-dados-para-que-serve-como-fazer/>>. Acesso em: 26 de fev. de 2020.

GIL. A. C.; Como Elaborar Projetos de Pesquisa. 5.ed. São Paulo: Atlas, 2010.

LAROSE, D. T.; Discovering Knowledge in Data, An Introduction to Data. [S. I.]: John Wiley & Sons, 2005.

LUGER, G. Inteligência Artificial. 6 Ed. São Paulo. Pearson Education do Brasil, 2013.

MEDEIROS, L. F. de. Inteligência artificial aplicada: uma abordagem introdutória. Curitiba. InterSaberes, 2018.

MENDES, R. D. Inteligencia Artificial: Sistemas Especialistas no Gerenciamento da Informação. Scielo, 1997. Disponível em: <http://www.scielo.br/scielo.php?pid=S0100-19651997000100006&script=sci_arttext>. Acesso em: 14 de fev. de 2020.

O QUE É MACHINE LEARNING OU APRENDIZAGEM DE MÁQUINA?. Sap, 2017. Disponível em: <<https://news.sap.com/brazil/2017/10/o-que-e-machine-learning-ou-aprendizagem-de-maquina/>>. Acesso em: 14 de fev. de 2020.

RUSSEL, S.; NORVIG, P. Inteligência Artificial: tradução da segunda edição. 2 Ed. Rio de Janeiro. Elsevier, 2004.