



# ANÁLISE DE SINAIS E PADRÕES DE MERCADO EM CRIPTOMOEDAS UTILIZANDO TÉCNICAS DE MACHINE LEARNING PARA OTIMIZAR ESTRATÉGIA DE TRADING

Felipe Segundo Marçal  
Instituto Federal do Espírito Santo. E-mail: (felipe-marcal@hotmail.com).

Dr. Pedro Matos da Silva  
Instituto Federal do Espírito Santo. E-mail: (pedroms@ifes.com.br).

**Resumo:** O presente estudo tem como objetivo principal investigar e aplicar técnicas de machine learning para otimizar estratégias de trading no mercado de criptomoedas, com foco específico no Bitcoin e Ethereum. A pesquisa envolve a coleta e análise de dados da corretora Binance utilizando a linguagem de programação Python e bibliotecas como Pandas, NumPy e Scikit-Learn. Os dados coletados abrangem um período de 2018 a 2024, incluindo variáveis como volume de negociação, número de trades e índice de força relativa (RSI). A metodologia adotada inclui a análise de regressão linear múltipla e o algoritmo k-Nearest Neighbors (kNN) para prever preços de fechamento. Os resultados indicam que ambas as técnicas são eficazes, com o kNN apresentando maior precisão preditiva, com um coeficiente de determinação ( $R^2$ ) de 0,95, MSE de 94.174,61 e MAE de 175,84. As variáveis "eth\_volume", "eth\_num\_trades", "RSI25", "btc\_volume", "btc\_num\_trades" e 'sp500\_Close' mostraram alta correlação com o preço de fechamento do Ethereum. A pesquisa conclui que a análise detalhada de dados e a aplicação de machine learning podem fornecer insights valiosos e melhorar a precisão das previsões de preços, beneficiando investidores em suas decisões de trading.

**Palavras-chave:** Criptomoedas, Machine Learning, Regressão Linear, k-Nearest Neighbors, Trading, Visualização de Dados, Previsão de Preços

**Abstract:** This study primarily aims to investigate and apply machine learning techniques to optimize trading strategies in the cryptocurrency market, focusing specifically on Bitcoin and Ethereum. The research involves the collection and analysis of data from the Binance platform using Python programming and libraries such as Pandas, NumPy, and Scikit-Learn. Data collected spans from 2018 to 2024, including variables such as trading volume, number of trades, and the relative strength index (RSI). The methodology includes multiple linear regression analysis and the k-Nearest Neighbors (kNN) algorithm to predict closing prices. Results indicate that both techniques are effective, with kNN demonstrating higher predictive accuracy, with a coefficient of determination ( $R^2$ ) of 0.95, MSE of 94,174.61 and MAE of 175.84. Variables such as "eth\_volume", "eth\_num\_trades", "RSI25", "btc\_volume", "btc\_num\_trades" and 'sp500\_Close'



showed high correlation with the closing price of Ethereum. The study concludes that detailed data analysis and the application of machine learning can provide valuable insights and improve the accuracy of price predictions, benefiting investors in their trading decisions.

**Keywords:** Cryptocurrencies, Machine Learning, Linear Regression, k-Nearest Neighbors, Trading, Data Visualization, Price Prediction

## 1 INTRODUÇÃO

Desde os primórdios das civilizações, sistemas de trocas confiáveis têm sido a base das relações humanas (Nascimento et al., 2022). Nos pequenos grupos humanos, o comércio era sustentado pelo bem comum e pela confiança quase familiar entre seus membros. À medida que as sociedades se expandiram globalmente e os grupos humanos cresceram exponencialmente, tornou-se necessário que o comércio, como um sistema confiável, se apoiasse em bases materiais mais sólidas, surgindo então o dinheiro.

Com o tempo, as trocas baseadas em materiais tornaram-se mais complexas e dispendiosas, levando ao surgimento dos primeiros bancos e sistemas de saque e depósito baseados em autenticação (Nascimento et al., 2022). Esses sistemas, que emitiam cartas assinadas, permitiam que o dinheiro fosse retirado apenas pelo seu portador, constituindo os primeiros passos em direção aos sistemas financeiros modernos. Muito se fala sobre Bitcoin, mas é importante destacar que ele é apenas uma das muitas criptomoedas existentes (Moraes et al., 2021). Na realidade, existe um vasto universo de criptomoedas, com mais de 5.392 diferentes até dezembro de 2020, avaliadas em mais de 200 bilhões de dólares.

As principais criptomoedas: o Bitcoin, ETH (Ethereum), XRP (Ripple), BCH (Bitcoin cash), LTC (Litecoin), todas essas apresentam características distintas das moedas tradicionais, com facilidade de uso, anonimato, transparência, baixos custos de transação e irreversibilidade o diferenciam no mercado financeiro (Moraes et al., 2021).

O mercado de criptomoedas é diversificado e dinâmico, com mais de 1.600 criptomoedas disponíveis (Danial, 2022), algumas com potencial para substituir o dinheiro tradicional. Com isso, Moraes (2021) destaca desafios significativos no mercado de criptomoedas, como volatilidade, risco de roubo digital, falta de regulação e dependência tecnológica. Esses aspectos influenciam a confiabilidade e a aceitação dessas moedas, além de levantar questões sobre sua natureza como investimento ou moeda.

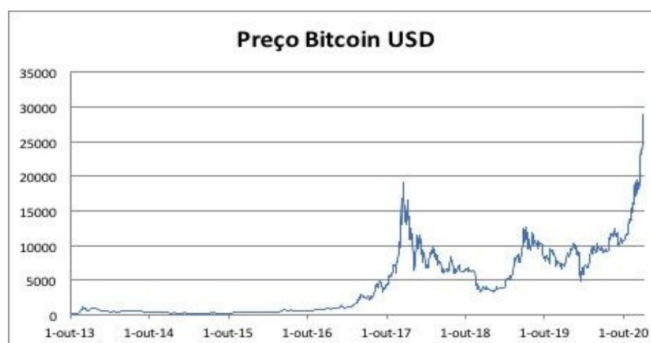


Figura 1 – Variação da criptomoeda Bitcoin (2013- 2020)

Este estudo emprega uma metodologia de análise de dados quantitativos com o uso de técnicas de machine learning para identificar perfis e momentos adequados para investimento (compra e venda) de criptomoedas. O processo metodológico é dividido em quatro etapas principais: seleção da corretora, coleta de dados por meio da API, análise e visualização dos dados, aplicação de técnicas de machine (Regressão Linear ou k-Nearest Neighbors (kNN)) para previsão e identificação de oportunidades de investimento.

## 2 PROCESSOS METODOLÓGICOS/MATERIAIS E MÉTODOS

### 2.1 Seleção da Corretora

Maciel (2019), a história das criptomoedas teve início em 2009, quando Satoshi Nakamoto introduziu o Bitcoin, a primeira moeda digital descentralizada. Embora não tenha criado a blockchain (tecnologia de registro distribuído que permite a criação de um banco de dados seguro e transparente), Nakamoto associou uma moeda digital a essa tecnologia, inaugurando um sistema de segurança criptografado. Desde então, outras criptomoedas como Ethereum, Litecoin e Cardano têm surgido, desafiando o dinheiro tradicional e redefinindo as formas de transação. Segundo Leite (2022), a corretora Binance é a preferida entre os investidores de criptomoedas, registrando o maior volume mundial de negociações diárias desde 2017. Em contrapartida, as corretoras FXT e CoinNext são menos utilizadas, representando apenas 5% de participação cada.

Lisboa et al. (2023) destacam que a escolha de uma corretora de criptomoeda muitas vezes está relacionada à reputação da corretora no mercado, à segurança oferecida para as transações e à variedade de criptomoedas disponíveis para negociação."

Para viabilizar este estudo, a seleção da corretora de testes é fundamental. Assim, buscamos dados reais de transações na principal corretora de negociação de

criptomoedas em termos de volume diário, o [www.binance.com](http://www.binance.com). Para acessar esses dados, procedemos com o cadastro e login na corretora e, em seguida, utilizamos sua API para baixar todas as transações realizadas dentro de um período específico.

## 2.2 Ferramenta de programação com linguagem Python

De acordo com NETO e MACIEL (2024), Python é extensivamente utilizado em análise de dados e machine learning devido à sua sintaxe amigável e tipagem robusta. Essa linguagem, que é dinamicamente tipada e multiparadigma, simplifica o desenvolvimento de programas eficientes e compactos, sendo interpretada e executada através de interpretadores específicos para diferentes sistemas operacionais. Lambert (2022), um programa Python é composto por um ou mais módulos, sendo cada módulo um arquivo de código que pode conter instruções, definições de funções e classes. Tanto pequenos programas, chamados de scripts, quanto programas mais extensos e complexos, contam com um módulo principal e outros de suporte.

Utilizando-o como premissa e ferramenta de programação, o mesmo deverá ser a ponte através de um script que capturará do ecossistema da Binance os dados necessários para a coleta, análise, visualização e aplicação de técnicas de Machine Learning.

## 2.3 Coleta de dados por meio da API

A API, segundo Ferreira (2018), é um conjunto de normas e ferramentas que facilitam a comunicação entre diferentes aplicações, permitindo essa interação de forma transparente para o usuário final. Ela oferece acesso a diversas funcionalidades de um sistema, possibilitando sua utilização em outras aplicações de forma eficiente. Algumas características essenciais incluem sua disponibilidade pública, restrita ou privada, sua base na arquitetura cliente-servidor e o uso do protocolo HTTP, promovendo a interoperabilidade entre sistemas.

Tal conexão se faz através do utilizador ao ecossistema Binance conforme figura 2.

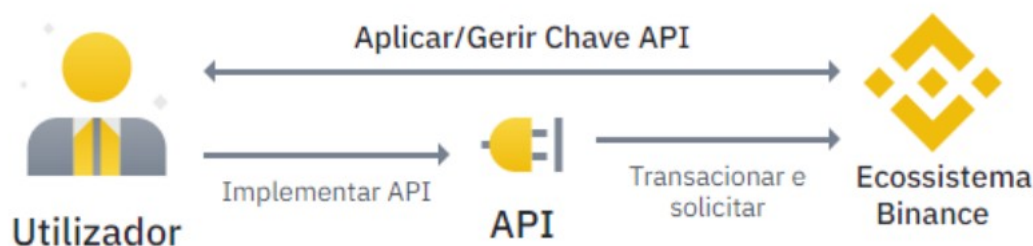


Figura 2 – figura ilustrativa de conexão ao ecossistema da Corretora Binance



## 2.4 Análise de dados e visualização dos dados

Milani (2020) destaca que a apresentação eficaz dos dados é fundamental para extrair insights valiosos, evidenciando a importância da visualização na ciência de dados.

Neto e Maciel (2024) ressaltam a importância de uma abordagem variada na etapa de Preparação dos Dados, garantindo que a amostra coletada represente fielmente o mundo real. Além disso, enfatizam a necessidade de conhecimento em bancos de dados relacionais, pois os dados para problemas de aprendizado de máquina geralmente estão disponíveis digitalmente, seja em bancos de dados ou em arquivos de texto na web. Knafllic (2017), cita que a habilidade de contar histórias por meio de dados é crucial em um mundo cada vez mais orientado por informações. A prática da visualização de dados remonta ao desenvolvimento de mapas e diagramas no século XVII, assim como à criação do gráfico em formato de pizza no século XIX.

A visualização de dados, também chamada de data visualization, consiste na representação gráfica de informações ou dados por meio de elementos visuais como tabelas, gráficos e mapas. Essa técnica facilita a compreensão dos dados pelo público, uma vez que utiliza cores, padrões e outros recursos visuais para tornar as informações mais acessíveis e atrativas (MILANI, 2020).

Nesta fase da pesquisa, através de uma análise quantitativa, os dados da corretora Binance serão analisados e visualizados para identificar insights e informações relevantes, visando embasar a aplicação de técnicas de Machine Learning no futuro, identificando tendências, padrões e correlações entre as diferentes variáveis.

## 2.5 Aplicação de Técnicas de Machine Learning e identificação de oportunidades

Nelli (2015) define Machine Learning como a disciplina que utiliza uma variedade de técnicas e algoritmos para descobrir padrões e informações relevantes de forma automatizada, através do treinamento de algoritmos por métodos matemáticos. Em Neto e Maciel (2024), as bibliotecas Pandas, NumPy e Scikit-Learn são fundamentais para Data Science e Machine Learning. Pandas simplifica a manipulação de dados, enquanto NumPy facilita cálculos científicos e Scikit-Learn é o padrão para algoritmos de Machine Learning, economizando tempo e minimizando erros para cientistas de dados.

É fundamental ressaltar que, ao avaliar e comparar diferentes modelos, é necessário ir além das métricas de desempenho. Conforme destacado por Sicsú et al. (2023), essa análise deve considerar o método como um todo, selecionando aquele mais adequado para a solução do problema em questão. Isso envolve

justificar o modelo dentro do contexto do problema, sua simplicidade e a facilidade de interpretar e explicar os resultados aos usuários.

Ao avaliar a performance de um modelo de previsão ou classificação, é crucial evitar o uso da mesma amostra para treinamento e teste, a fim de evitar resultados otimistas e detectar possíveis problemas de overfitting. Segundo os princípios descritos por Murphy (2023), recomenda-se dividir aleatoriamente a amostra em conjuntos de treinamento e teste, usualmente com proporções entre 50% e 70% para treinamento e o restante para teste. Essa abordagem permite comparar métricas obtidas em ambos os conjuntos, identificando discrepâncias que sugerem overfitting.

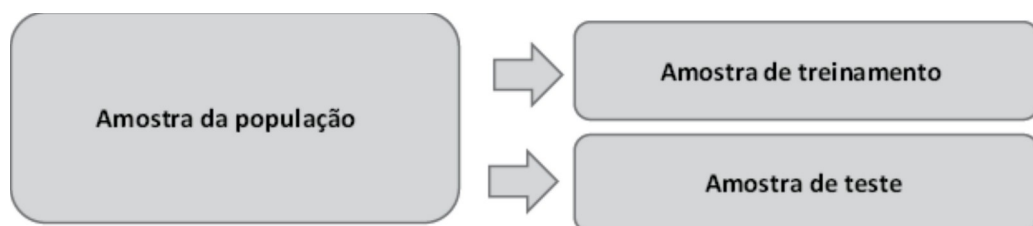


Figura 3 –Partição da amostra para treinamento e teste.

## 2.6 Regressão Linear

Frequentemente, o profissional de análise de dados enfrenta a necessidade de estimar valores de dados não observados. Nesse contexto, a regressão linear desempenha um papel fundamental ao fornecer suporte para essa tarefa.

A finalidade da regressão linear é estabelecer uma associação linear, sempre que viável, entre dois conjuntos de variáveis, um independente e outro dependente (FARIA; OLIVEIRA; PINTO; SZWARCFITER, 2021).

Dentro das técnicas mais empregadas que envolvem transformações lineares, destaca-se a Regressão Múltipla. Segundo Hair (1998), seu propósito é utilizar as variáveis independentes conhecidas para prever os valores da variável dependente. No entanto, ao realizar a regressão múltipla, é fundamental avaliar a multicolinearidade, que se manifesta quando há correlação entre três ou mais variáveis independentes (Hair, 1998)

O desenvolvimento e uso desta ferramenta sugere como conclusão, a previsibilidade de ocorrência de alguns cenários, sendo estes orientados por meio da precisão e eficácia através do coeficiente de determinação ( $R^2$ ), o Erro quadrático médio (MSE) e o Erro Absoluto médio (MAE)

**$R^2$ :** Avalia a qualidade do ajuste do modelo, ou seja, quanto da variabilidade dos dados o modelo consegue explicar.

**MSE:** Avalia a magnitude dos erros ao quadrado, sendo útil para entender a variabilidade dos erros do modelo.



**MAE:** Avalia a magnitude dos erros de forma linear, oferecendo uma interpretação direta das diferenças médias entre valores reais e previstos.

A análise de regressão linear, frequentemente aplicada em contextos financeiros, visa estimar o valor futuro de uma variável dependente com base em uma ou mais variáveis independentes. Essa técnica permite prever, por exemplo, a valorização de uma criptomoeda com base em fatores históricos. A relação entre a variável dependente Y e a variável independente X é descrita por:

$$Y = \beta_0 + \beta_1 X + \epsilon \quad (1.0)$$

onde  $\beta_0$  representa o intercepto,  $\beta_1$  é o coeficiente angular que indica a variação esperada em Y para cada unidade de variação em X, e  $\epsilon$  é o termo de erro. O objetivo é ajustar os parâmetros  $\beta_0$  e  $\beta_1$  de forma a minimizar a soma dos quadrados dos resíduos  $(Y_i - \hat{Y}_i)^2$

### **K-Nearest Neighbor (kNN)**

Os algoritmos de aprendizado de máquina, como o KNN, têm a capacidade de identificar padrões em grandes conjuntos de dados, sintetizando conhecimento humano e gerando modelos úteis para a resolução de problemas futuros (WITTEN, 2011). Esses algoritmos buscam extrair insights significativos a partir das informações disponíveis, contribuindo para a criação de teorias ou modelos preditivos que agregam valor nas tomadas de decisão (WITTEN, 2011).

No método dos k-vizinhos mais próximos (KNN), a classificação de uma observação é realizada com base nos K vizinhos mais próximos, onde K é um hiperparâmetro definido (SICSÚ et al., 2024). A observação é classificada no grupo predominante de seus K vizinhos mais próximos. Para evitar empates, geralmente é utilizado um valor ímpar para K em classificações de dois grupos distintos. Além disso, o método KNN pode ser aplicado para classificação em três ou mais grupos, aumentando a possibilidade de empates. O hiperparâmetro K é geralmente determinado empiricamente, otimizando medidas de desempenho, como a Área Sob a Curva ROC (AUROC), através de validação cruzada.

## **3 RESULTADO**

Esse estudo tem como premissa extrair e verificar as diversas correlações e influências do Preço do Bitcoin, índice S&P500 (índice de mercado ponderado pelo valor de mercado que mede o desempenho das 500 maiores empresas dos Estados Unidos), seu volume, número de negociações, somado a estes, o RSI (indicador de força de mercado sobre o preço de fechamento das criptomoedas. Esta análise tem como referência o período de 2018 a 2024, considerando suas variações semana a semana.

Inicialmente, em posse das principais ferramentas descritas acima, com o auxílio do Jupyter Notebook (software de programação) inicia-se a extração e coleta dos dados deste estudo. Via API o mesmo foi extraído da corretora Binance e também do Yahoo Finance, conforme apresentado na figura 4 e 5.

```
In [5]: api_key = [REDACTED]
        api_secret = [REDACTED]

        client = Client(api_key=api_key, api_secret=api_secret)
        precos = client.get_all_tickers()

In [6]: def get_binance_data(api_key, pair, countdown='open', interval='1W', start='1 Mar 2018', end='1 Abr 2024'):
        client = Client(api_key=api_key)
        intervals = {
            '15m': Client.KLINE_INTERVAL_15MINUTE,
            '30m': Client.KLINE_INTERVAL_30MINUTE,
            '1h': Client.KLINE_INTERVAL_1HOUR,
            '4h': Client.KLINE_INTERVAL_4HOUR,
            '1d': Client.KLINE_INTERVAL_1DAY,
            '1w': Client.KLINE_INTERVAL_1WEEK
        }
        interval = intervals.get(interval, '1W')
        print(f'Historical interval {interval}')
        klines = client.get_historical_klines(symbol=pair, interval=interval, start_str=start, end_str=end)
        data = pd.DataFrame(klines)
        data.columns = ['open_time', 'open', 'high', 'low', 'close', 'volume', 'close_time', 'qav', 'num_trades', 'taker_base_v']
        data.index = [pd.to_datetime(x, unit='ms').strftime('%Y-%m-%d %H:%M:%S') for x in data.open_time]
        usecols=['open', 'high', 'low', 'close', 'volume', 'num_trades']
        data = data[usecols]
        data = data.astype('float')
        return data

In [7]: api_key = [REDACTED]
        symbol = 'ETHBTC'
        eth_df = get_binance_data(api_key, symbol)
        eth_df = eth_df.rename(columns=lambda x: 'eth_' + x)
        eth_df
```

Figura 4 – Conectando API da Binance ao SPARK.

```
import yfinance as yf

# Defina o símbolo do S&P 500
symbol = '^GSPC' # Símbolo do S&P 500 no Yahoo Finance

# Obtenha os dados do S&P 500
sp500_data = yf.download(symbol, start='2018-03-05', end='2024-04-05', interval='1wk')

sp500_data = sp500_data.rename(columns=lambda x: 'sp500_' + x)
sp500_data
```

figura 5 – Realizando o Download do Yahoo Finance.

Para este caso, tendo como base a análise semanal, foram inicialmente extraídas 03 tabelas por este método, conteúdo nas mesmas, as informações e valores de mercado (preço abertura, fechamento, alto, baixo número de negociações e volume de compra das moedas, Ethereum figura 6 , Bitcoin figura 7 e o índice SP&500 figura 8. Posterior remoção de valores nulos ocasionados pelo índice de força relativa, concatena-se as tabelas em uma única tabela com 294 linhas e 19 colunas (figura 9).





### Historical interval 1w

	<b>eth_open</b>	<b>eth_high</b>	<b>eth_low</b>	<b>eth_close</b>	<b>eth_volume</b>	<b>eth_num_trades</b>
<b>2018-03-05 00:00:00</b>	866.61	871.75	640.00	720.85	5.180773e+05	814448.0
<b>2018-03-12 00:00:00</b>	721.00	740.50	450.00	537.00	6.893589e+05	940610.0
<b>2018-03-19 00:00:00</b>	537.39	590.52	506.71	523.74	8.143811e+05	912398.0
<b>2018-03-26 00:00:00</b>	523.95	527.67	356.90	378.58	1.215093e+06	1036957.0
<b>2018-04-02 00:00:00</b>	378.58	419.08	362.54	400.40	7.153698e+05	646777.0
...	...	...	...	...	...	...
<b>2024-03-04 00:00:00</b>	3487.80	3993.75	3200.00	3878.47	4.876444e+06	13083480.0
<b>2024-03-11 00:00:00</b>	3878.47	4093.92	3412.00	3644.71	4.498502e+06	13237137.0
<b>2024-03-18 00:00:00</b>	3644.70	3645.02	3056.56	3454.98	4.573958e+06	13302135.0
<b>2024-03-25 00:00:00</b>	3454.99	3678.86	3420.12	3645.29	2.678017e+06	8003687.0

Figura 6 – Dados da Ethereum.

### Historical interval 1w

	<b>btc_open</b>	<b>btc_high</b>	<b>btc_low</b>	<b>btc_close</b>	<b>btc_volume</b>	<b>btc_num_trades</b>
<b>2018-03-05 00:00:00</b>	11515.00	11710.00	8329.00	9533.57	282035.225382	2232004.0
<b>2018-03-12 00:00:00</b>	9533.57	9888.88	7322.00	8189.99	315835.151606	2450216.0
<b>2018-03-19 00:00:00</b>	8189.00	9177.01	8088.40	8470.15	285211.539511	1993351.0
<b>2018-03-26 00:00:00</b>	8470.14	8514.89	6430.00	6813.01	308729.492031	2021451.0
<b>2018-04-02 00:00:00</b>	6813.01	7520.00	6500.00	7018.00	232320.070499	1464142.0
...	...	...	...	...	...	...
<b>2024-03-04 00:00:00</b>	63113.97	69990.00	59005.00	68955.88	481870.181782	21358553.0
<b>2024-03-11 00:00:00</b>	68955.88	73777.00	64533.00	68393.48	477496.917520	20669990.0
<b>2024-03-18 00:00:00</b>	68393.47	68956.00	60775.00	67209.99	409762.744140	18123751.0
<b>2024-03-25 00:00:00</b>	67210.00	71769.54	66385.06	71280.01	235409.957550	12877311.0

Figura 7 – Dados do Bitcoin.



Date	sp500_Open	sp500_High	sp500_Low	sp500_Close	sp500_Adj Close	sp500_Volume
2018-03-05 00:00:00	2681.060059	2786.570068	2675.750000	2786.570068	2786.570068	17112270000
2018-03-12 00:00:00	2790.540039	2801.899902	2741.469971	2752.010010	2752.010010	18908730000
2018-03-19 00:00:00	2741.379883	2741.379883	2585.889893	2588.260010	2588.260010	17629670000
2018-03-26 00:00:00	2619.350098	2674.780029	2593.060059	2640.870117	2640.870117	14747480000
2018-04-02 00:00:00	2633.449951	2672.080078	2553.800049	2604.469971	2604.469971	16943670000
...	...	...	...	...	...	...
2024-03-04 00:00:00	5130.990234	5189.259766	5056.819824	5123.689941	5123.689941	22082750000
2024-03-11 00:00:00	5111.959961	5179.870117	5091.140137	5117.089844	5117.089844	24701470000
2024-03-18 00:00:00	5154.770020	5261.100098	5131.589844	5234.180176	5234.180176	19715260000
2024-03-25 00:00:00	5219.520020	5264.850098	5203.419922	5254.350098	5254.350098	15051920000
2024-04-01 00:00:00	5257.970215	5263.950195	5146.060059	5204.339844	5204.339844	18378230000

Figura 8 – Dados do Índice S&P 500.

	eth_open	eth_high	eth_low	eth_close	eth_volume	eth_num_trades	RSI25	btc_open	btc_high	btc_low	btc_close	btc_volume	btc_nur
2018-08-20 00:00:00	299.30	304.04	259.50	274.29	1.882848e+06	887972.0	31.752017	6477.53	6882.54	6220.00	6700.00	286906.539311	1:
2018-08-27 00:00:00	274.17	302.00	271.59	295.41	2.011520e+06	931502.0	33.462654	6700.00	7345.45	6646.50	7302.01	294674.913005	1:
2018-09-03 00:00:00	295.51	296.62	185.02	195.81	2.833082e+06	1122752.0	29.794117	7302.00	7410.00	6111.00	6250.81	306231.623424	1:
2018-09-10 00:00:00	196.13	227.42	167.32	220.17	3.839318e+06	1189784.0	31.701724	6252.26	6584.99	6169.68	6505.00	227428.182854	1:
2018-09-17 00:00:00	220.24	255.13	192.40	244.50	3.979104e+06	1164773.0	33.579364	6500.08	6839.03	6123.00	6708.00	244256.490121	1:
...	...	...	...	...	...	...	...	...	...	...	...	...	...
2024-03-04 00:00:00	3487.80	3993.75	3200.00	3878.47	4.876444e+06	13083480.0	75.934365	63113.97	69990.00	59005.00	68955.88	481870.181782	21:
2024-03-11 00:00:00	3878.47	4093.92	3412.00	3644.71	4.498502e+06	13237137.0	70.852680	68955.88	73777.00	64533.00	68393.48	477496.917520	20:
2024-03-18 00:00:00	3644.70	3645.02	3056.56	3454.98	4.573958e+06	13302135.0	67.058501	68393.47	68956.00	60775.00	67209.99	409762.744140	18:
2024-03-25 00:00:00	3454.99	3678.86	3420.12	3645.29	2.678017e+06	8003687.0	68.803982	67210.00	71769.54	66385.06	71280.01	235409.957550	12:
2024-04-01 00:00:00	3645.29	3645.95	3202.79	3454.20	2.537270e+06	6928534.0	65.191010	71280.00	71288.23	64493.07	69360.39	274227.114880	13:

294 rows x 19 columns

Figura 9 – Tabela geral concatenada às moedas Eth, Bitcoin e índices S&P500.

Posterior a concatenação das tabelas, normaliza-se para evitar erros ou divergência na interpretação seguinte por causa da amplitude destes valores.

Seguindo mais adiante, compreende-se que as informações estão normalizadas e aptas a serem verificadas as suas correlações e influência das várias variáveis coletadas tanto da corretora da Binance quanto do Yahoo Finance.

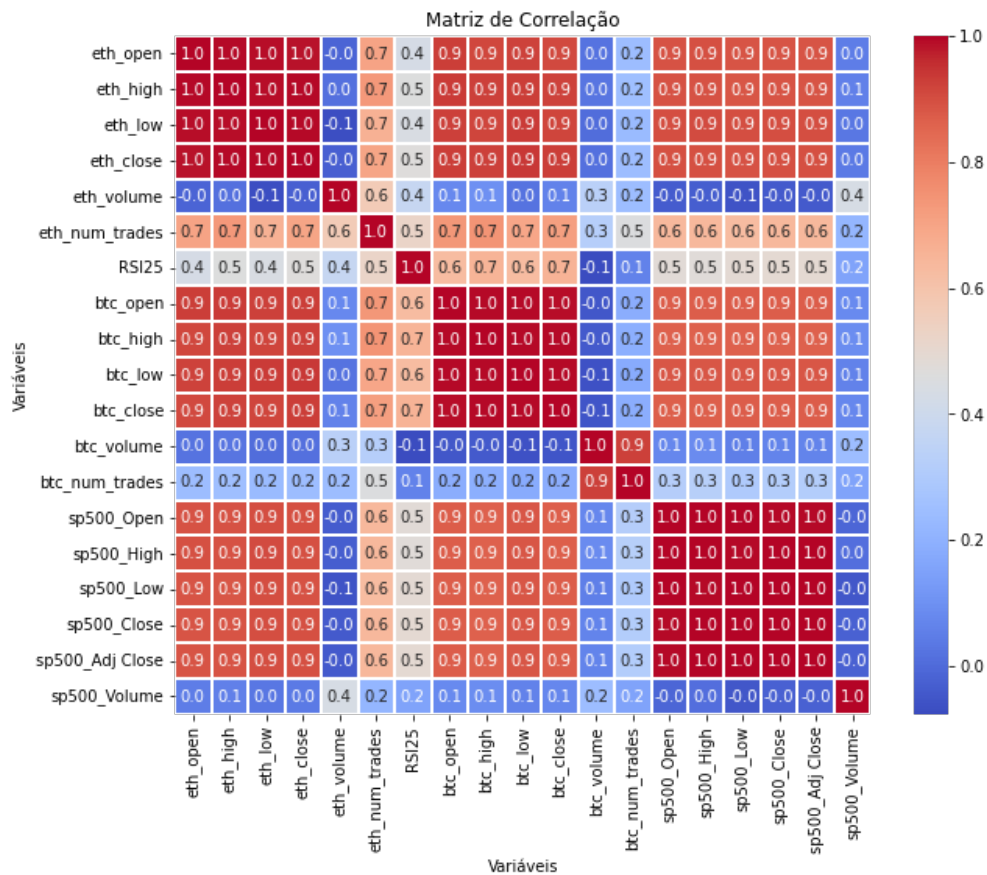


Figura 10 – Matriz de Correlação com todas as variáveis e dados coletados (normalizado).

Neste momento optei por desconsiderar os preços de abertura, preço mais baixo e alto, sendo a principal variável de análise o preço de fechamento. Sendo assim, a fim garantir variáveis que fornecerão uma análise coerente ao resultado esperado, meio a matriz de correção optei por utilizar para a técnica de machine learning, regressão linear múltipla e KNN, as seguintes variáveis:

Para X, "eth\_volume","eth\_num\_trades","RSI25","btc\_volume","btc\_num\_trades","sp500\_Close"; (2.0)

Para Y, "eth\_close" (3.0)

Como a própria literatura sugere, inicia-se o fatiamento do conjunto de dados da amostra. Para este caso, utiliza-se a biblioteca Sklearn com definições iniciais de 20% destinado a treino e 80% para teste para posterior treinamento e previsão por meio da regressão linear múltipla que deverá sugerir valores para teste e sua previsão. Por fim a execução do programa nos permite avaliar os resultados das métricas de avaliação de uma regressão linear, obtidos por meio das funções do sklearn.metrics, que avaliam a precisão e a eficácia do modelo de regressão linear. Neste caso, obtemos o coeficiente de determinação ( $R^2$ ), Erro quadrático médio (MSE) e o erro absoluto médio (MAE), respectivamente 0,87, 231.981,04 e 334,00.

Quando comparamos ambos valores reais e valores da previsão, temos a seguinte relação apresentada na figura 11.

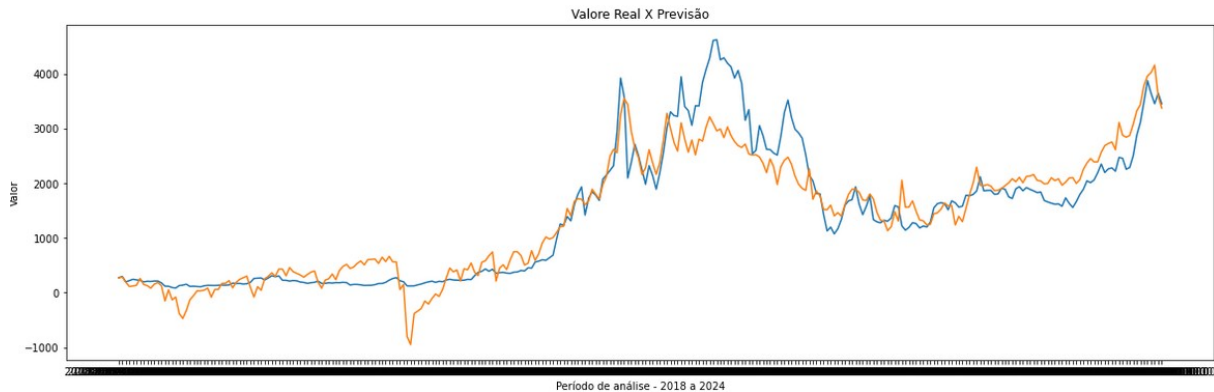


Figura 11 – Relação entre os valores reais (preço de fechamento) da Ethereum Vs. valores preditivos por regressão linear.

A técnica KNN, permitiu alcançar os seguintes resultados, coeficiente de determinação ( $R^2$ ), Erro quadrático médio (MSE) e o erro absoluto médio (MAE), respectivamente 0,95, 94.174,61 e 175,84. Quando comparamos ambos valores reais e valores da previsão, temos a seguinte relação apresentada na figura 12.

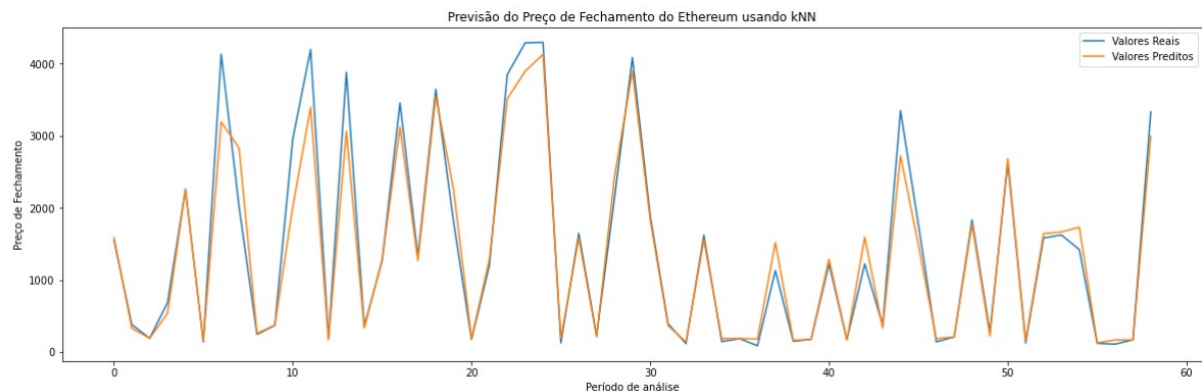


Figura 12 – Relação entre os valores reais (preço de fechamento) da Ethereum Vs. valores preditivos por KNN.

## 4 CONCLUSÃO

Este estudo apresenta uma análise detalhada de sinais e padrões de mercado em criptomoedas, utilizando técnicas de machine learning para otimizar estratégias de trading. A metodologia adotada inclui a coleta de dados via API da Binance, análise e visualização de dados, e aplicação de técnicas de machine learning, como Regressão Linear Múltipla e k-Nearest Neighbors (kNN), resultando em insights valiosos para o mercado de criptomoedas.



A análise evidenciou uma alta correlação entre variáveis-chave, como o volume de negociação, o número de trades e o índice de força relativa (RSI) com o preço de fechamento do Ethereum. Essa correlação sugere pontos de compra e venda aos investidores. O coeficiente de determinação ( $R^2$ ) de 0,87 na Regressão Linear indica que o modelo preditivo explica significativamente a variabilidade dos dados. Além disso, os valores de MSE (231.981,04) e MAE (334,00) reforçam a precisão do modelo, proporcionando uma base sólida para estratégias de trading mais informadas e eficazes.

A técnica kNN apresentou resultados ainda mais robustos, com um coeficiente de determinação ( $R^2$ ) de 0,95, MSE de 94.174,61 e MAE de 175,84. Estes resultados indicam uma precisão ainda maior na previsão dos preços de fechamento do Ethereum, evidenciando a eficácia desta abordagem para identificar tendências e oportunidades de investimento.

Para os investidores, os dados destacam a importância de monitorar variáveis específicas, como volume de negociação e número de trades, bem como indicadores técnicos como o RSI. A precisão dos modelos de machine learning utilizados sugere que essas variáveis possuem uma influência substancial nas previsões de preço, oferecendo um valor significativo para estratégias de trading mais assertivas.

Em conclusão, este estudo sublinha a relevância de uma análise de dados abrangente e a aplicação de técnicas avançadas de machine learning para identificar e prever tendências no mercado de criptomoedas. A identificação de variáveis com alta correlação e a eficácia dos modelos preditivos utilizados demonstram que é possível obter previsões precisas, oferecendo um valor significativo para investidores que buscam aprimorar suas estratégias de trading e tomar decisões mais informadas e eficazes no dinâmico mercado de criptomoedas.

A comparação entre os resultados das técnicas de Regressão Linear Múltipla e kNN mostra que ambas são eficazes, mas o kNN se destaca pela maior precisão preditiva, como evidenciado pelas métricas de desempenho.

## 5 APÊNDICE

```
import json  
import numpy as np  
pip install python-binance  
import matplotlib.pyplot as plt  
import pandas as pd  
import time
```



```
from binance.client import Client
from IPython.display import display

api_key = 'XXXXXXXXXX'
api_secret = 'XXXXXXXXXX'
client = Client(api_key=api_key, api_secret=api_secret)
precos = client.get_all_tickers()

In [ ]: def get_binance_data(api_key, pair, countdown='open', interval='1W',
client = Client(api_key=api_key)
intervals = {
'15m': Client.KLINE_INTERVAL_15MINUTE,
'30m': Client.KLINE_INTERVAL_30MINUTE,
'1h': Client.KLINE_INTERVAL_1HOUR,
'4h': Client.KLINE_INTERVAL_4HOUR,
'1d': Client.KLINE_INTERVAL_1DAY,
'1W': Client.KLINE_INTERVAL_1WEEK
}
interval = intervals.get(interval,'1W')
print(f'Historical interval {interval}')
klines = client.get_historical_klines(symbol=pair, interval=interval)
data = pd.DataFrame(klines)
data.columns = ['open_time','open', 'high', 'low', 'close', 'volume'
data.index = [pd.to_datetime(x, unit='ms').strftime('%Y-%m-%d %H:%M:%S')
usecols=['open', 'high', 'low', 'close', 'volume','num_trades']
data = data[usecols]
data = data.astype('float')
return data
api_key = 'XXXXXXX'
symbol = 'ETHUSDT'
eth_df = get_binance_data(api_key, symbol)
eth_df = eth_df.rename(columns=lambda x: 'eth_' + x)
eth_df
```



```
import pandas as pd
import ta
import yfinance as yf
# Calcular o RSI com uma janela de 14 períodos
eth_df['RSI25'] = ta.momentum.RSIIndicator(close=eth_df['eth_close'])
pip install ta
eth_df.describe()

""" BITCOIN E SUA INFLUENCIA """
api_key = 'XXXXXXXXXX'
symbol = 'BTCUSDT'
btc = get_binance_data(api_key, symbol)
def get_binance_data(api_key, pair, countdown='open', interval='1W',
client = Client(api_key=api_key)
intervals = {
'15m': Client.KLINE_INTERVAL_15MINUTE,
'30m': Client.KLINE_INTERVAL_30MINUTE,
'1h': Client.KLINE_INTERVAL_1HOUR,
'4h': Client.KLINE_INTERVAL_4HOUR,
'1d': Client.KLINE_INTERVAL_1DAY,
'1W': Client.KLINE_INTERVAL_1WEEK
}
interval = intervals.get(interval,'1W')
print(f'Historical interval {interval}')
klines = client.get_historical_klines(symbol=pair, interval=interval)
data = pd.DataFrame(klines)
data.columns = ['open_time','open', 'high', 'low', 'close', 'volume'
data.index = [pd.to_datetime(x, unit='ms').strftime('%Y-%m-%d %H:%M:%S')
usecols=['open', 'high', 'low', 'close', 'volume','num_trades']
data = data[usecols]
data = data.astype('float')
```



return data

```
btc_df = btc.rename(columns=lambda x: 'btc_' + x)
```

```
btc_df
```

```
In [ ]: import yfinance as yf
```

```
# Defina o símbolo do S&P 500
```

```
symbol = '^GSPC' # Símbolo do S&P 500 no Yahoo Finance
```

```
# Obtenha os dados do S&P 500
```

```
sp500_data = yf.download(symbol, start='2018-03-05', end='2024-04-05')
```

```
sp500_data = sp500_data.rename(columns=lambda x: 'sp500_' + x)
```

```
sp500_data
```

```
# Formatar o índice adicionando "00:00:00"
```

```
sp500_data.index = pd.to_datetime(sp500_data.index).strftime("%Y-%m-%d 00:00:00")
```

```
sp500_data
```

```
"""UNINDO BANCO DA DADOS"""
```

```
# Concatenando os DataFrames ao longo das colunas (axis=1)
```

```
resultado_concat = pd.concat([eth_df, btc_df], axis=1)
```

```
resultado_concat
```

```
"""UNINDO BANCO DA DADOS"""
```

```
# Concatenando os DataFrames ao longo das colunas (axis=1)
```

```
resultado_concat = pd.concat([resultado_concat, sp500_data], axis=1)
```

```
resultado_concat
```

```
resultado_concat.describe().round(2)
```

```
# Contar valores NaN em cada coluna
```

```
contagem_nan_por_coluna = resultado_concat.isna().sum()
```

```
print(contagem_nan_por_coluna)
```

```
# Remover linhas que contenham valores NaN em qualquer coluna
```

```
df_sem_nan = resultado_concat.dropna()
```

```
# Contar valores NaN em cada coluna
```

```
contagem_nan_por_coluna = df_sem_nan.isna().sum()
```

```
df_sem_nan
```

```
from sklearn.preprocessing import StandardScaler
```





```
# Inicializando o StandardScaler
scaler = StandardScaler()

# Ajustando o scaler aos dados e transformando os dados
normalized_data = scaler.fit_transform(df_sem_nan)

# Convertendo os dados normalizados de volta para um DataFrame
normalized_df = pd.DataFrame(normalized_data, columns=resultado_concat)
print("Dados Normalizados:")
print(normalized_df)

# análise de correlação
correlation = df_sem_nan.corr()
display(correlation)
df_sem_nan.describe()

# Criar uma nova figura com tamanho personalizado
plt.figure(figsize=(20, 6)) # Largura x Altura em polegadas

# Plotar a coluna 'sp500_Close'
plt.plot(df_sem_nan['sp500_Close'])

# Adicionar rótulos e título
plt.xlabel('Período de análise')
plt.ylabel('Valor')
plt.title('S&P500')

# Exibir o gráfico
plt.show()

# Criar uma nova figura com tamanho personalizado
plt.figure(figsize=(20, 6)) # Largura x Altura em polegadas

# Plotar a coluna 'sp500_Close'
plt.plot(df_sem_nan['eth_volume'])

# Adicionar rótulos e título
plt.xlabel('Período de análise')
plt.ylabel('Valor')
plt.title('eth_volume')

# Exibir o gráfico
```



```
plt.show()

# Criar uma nova figura com tamanho personalizado
plt.figure(figsize=(20, 6)) # Largura x Altura em polegadas
# Plotar a coluna 'sp500_Close'
plt.plot(df_sem_nan['eth_num_trades'])
# Adicionar rótulos e título
plt.xlabel('Período de análise')
plt.ylabel('Valor')
plt.title('eth_num_trades')
# Exibir o gráfico
plt.show()

# Criar uma nova figura com tamanho personalizado
plt.figure(figsize=(20, 6)) # Largura x Altura em polegadas
# Plotar a coluna 'sp500_Close'
plt.plot(df_sem_nan['RSI25'])
# Adicionar rótulos e título
plt.xlabel('Período de análise')
plt.ylabel('Valor')
plt.title('RSI - Índice de Força relativa 25 períodos')
# Exibir o gráfico
plt.show()

# Criar uma nova figura com tamanho personalizado
plt.figure(figsize=(20, 6)) # Largura x Altura em polegadas
# Plotar a coluna 'sp500_Close'
plt.plot(df_sem_nan['btc_volume'])
# Adicionar rótulos e título
plt.xlabel('Período de análise')
plt.ylabel('Valor')
plt.title('Btc - Volume')
# Exibir o gráfico
plt.show()
```



```
# Criar uma nova figura com tamanho personalizado
plt.figure(figsize=(20, 6)) # Largura x Altura em polegadas
# Plotar a coluna 'sp500_Close'
plt.plot(df_sem_nan['btc_num_trades'])
# Adicionar rótulos e título
plt.xlabel('Período de análise')
plt.ylabel('Valor')
plt.title('Btc Número de negociações')
# Exibir o gráfico
plt.show()
import seaborn as sn
# Aumentar o tamanho do gráfico
plt.figure(figsize=(10, 8))
# Criar o heatmap
plot = sn.heatmap(correlation, annot=True, fmt=".1f", linewidths=.1,
# Adicionar rótulos aos eixos
plt.xlabel("Variáveis")
plt.ylabel("Variáveis")
# Adicionar título
plt.title("Matriz de Correlação")
# Exibir o gráfico
plt.show()
df_sem_nan[['eth_volume', "eth_num_trades", "RSI25", "btc_volume", "btc_num
Y = df_sem_nan["eth_close"]
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2
X_train
X_test
y_train
y_test
from sklearn.linear_model import LinearRegression
```



```
# TREINANDO O MODELO
modelo = LinearRegression().fit(X_train, y_train)
display(modelo)
y_pred = modelo.predict(X_test)
y_pred
X_test
from sklearn.metrics import r2_score, mean_squared_error, mean_absolute_err
print('R²:', r2_score(y_test, y_pred))
print('Erro quadrático médio:', mean_squared_error(y_test, y_pred))
print('Erro absoluto médio:', mean_absolute_error(y_test, y_pred))
y_pred = modelo.predict(X)
print(y_pred)
print(Y)
a = modelo.coef_
b = modelo.intercept_
print(f'A equação que representa este modelo é: \nY = +{b:.3} +{a[0]:
X_test
# Criar uma nova figura com tamanho personalizado
plt.figure(figsize=(20, 6)) # Largura x Altura em polegadas
# Plotar valores reais
plt.plot(Y)
# Plotar valores de previsão
plt.plot(y_pred)
# Adicionar rótulos e título
plt.xlabel('Período de análise - 2018 a 2024')
plt.ylabel('Valor')
plt.title('Valore Real X Previsão')
# Exibir o gráfico
plt.show()
In [:]: # Normalizar os dados
scaler = StandardScaler()
```



```
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
# Definir o modelo kNN
knn = KNeighborsRegressor(n_neighbors=5) # Você pode ajustar o valor de 'n'
# Treinar o modelo
knn.fit(X_train_scaled, y_train)
# Fazer previsões no conjunto de teste
y_pred = knn.predict(X_test_scaled)
# Avaliar o modelo
mse = mean_squared_error(y_test, y_pred)
mae = mean_absolute_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
print(f'MSE: {mse}')
print(f'MAE: {mae}')
print(f'R²: {r2}')
# Plotar valores reais vs. preditos
plt.figure(figsize=(20, 6))
plt.plot(y_test.values, label='Valores Reais')
plt.plot(y_pred, label='Valores Preditos')
plt.xlabel('Período de análise')
plt.ylabel('Preço de Fechamento')
plt.title('Previsão do Preço de Fechamento do Ethereum usando kNN')
plt.legend()
plt.show()
```

## 6 REFERÊNCIAS

MORAES, Alexandre Fernandes de. **Bitcoin e blockchain: a revolução das moedas digitais**. São Paulo: Saraiva, 2021. E-book. ISBN 9786558110293. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9786558110293/>. Acesso em: 15 abr. 2024.



DANIAL, Kiana. **Investindo em criptomoedas Para Leigos**. Editora Alta Books, 2022. E-book. ISBN 9786555205237. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9786555205237/>. Acesso em: 15 abr. 2024.

MACIEL, F. A. **Introdução as criptomoedas: uma análise de possíveis impactos na economia, investimentos e contabilidade**. 2018. Monografia (Graduação em Ciências Contábeis) - Universidade de Caxias do Sul, Caxias do Sul, RS, 2018.

LISBOA, B. M. et al. **Curso de Educação financeira para investidores iniciantes Módulo VII: Criptomoedas**. Revista Científica Doctum Multidisciplinar, v. 3, n. 10, 27 nov. 2023.

BERNARDO, M. Â. V. DA F. F. **Comparação do mercado de criptomoedas com os mercados de ativos tradicionais**. Disponível em: <https://dspace.uevora.pt/rdpc/handle/10174/32711>. Acesso em: 17 abr. 2024.

NETTO, Amilcar; MACIEL, Francisco. **Python para Data Science e Machine Learning Descomplicado**. Editora Alta Books, 2021. E-book. ISBN 9786555203172. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9786555203172/>. Acesso em: 17 abr. 2024.

LAMBERT, Kenneth A. **Fundamentos de Python: estruturas de dados**. Cengage Learning Brasil, 2022. E-book. ISBN 9786555584288. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9786555584288/>. Acesso em: 17 abr. 2024.

SICSÚ, Abraham L.; SAMARTINI, André; BARTH, Nelson L. **Técnicas de machine learning**. Editora Blucher, 2023. E-book. ISBN 9786555063974. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9786555063974/>. Acesso em: 21 abr. 2024.

FARIA, J.C. **Regressão Linear Simples e Múltipla**. Disponível em: <https://lec.pro.br/download/faria/seminarios/rl.pdf>. Acesso em: 25 abr. 2024.

HAIR Jr., J.F.; ANDERSON, R.E.; TATHAM, R.L.; BLACK, W.C. **Multivariate data analysis**. 5. ed. Upper Saddle River: Prentice Hall, 1998. 730 p.

WITTEN, I. H.; EIBE, F.; HALL, M. A. **Data mining: practical machine learning tools and techniques**. 3. ed. Elsevier, 2011. 665 p.

NASCIMENTO, Leonardo Brendo Gomes et al. **Criptomoedas e blockchain**. Porto Alegre: SAGAH, 2022. ISBN 9786556900094.

MILANI, R. **Transformar Exercícios em Cenários para Investigação: uma Possibilidade de Inserção na Educação Matemática Crítica**. Perspectivas da



Educação Matemática. DOI: <https://doi.org/10.46312/pem.v13i31.9863>. Acesso em: 15 abr. 2024.

CASTRO FERREIRA, Diogo. **As criptomoedas e os desafios regulatórios e fiscais**. 2018

LEITE, R. S. M. M., dos Santos Filho, T. A., Pereira Junior, W. M., & da Silva, S. F. (2023). **Predição da cotação Real/Bitcoin usando a rede neural Long Short Term Memory**. *Brazilian Journal of Development*, 9(6), 21160–21177.

KNAFLIC, Cole Nussbaumer. **Storytelling com dados: um guia sobre visualização de dados para profissionais de negócios**. Rio de Janeiro: Alta Books, 2018.

NELLI, Fabio. **Python data analytics: Data analysis and science using PANDAs, Matplotlib and the Python Programming Language**. Apress, 2015.

FARIA, L., Oliveira, F. S., Pinto, P. E. D. & Szwarcfiter, J. L. (2021) **Ciência de dados: Algoritmos e aplicações**. Rio de Janeiro: IMPA.

MURPHY, Kevin P. **Machine Learning: A Probabilistic Perspective**. Cambridge, MA: MIT Press, 2012.