

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
NÚCLEO DE EDUCAÇÃO A DISTÂNCIA**

Pós-graduação *Lato Sensu* em Ciência de Dados e Big Data

Dimas Tadeu Parreiras

**TRADING INTELIGENTE –
UTILIZANDO ANÁLISE DE SÉRIES TEMPORAIS PARA
PREVER TENDÊNCIAS FUTURAS EM ATIVOS LISTADOS NA B3**

Belo Horizonte

2020

Dimas Tadeu Parreiras

**TRADING INTELIGENTE –
UTILIZANDO ANÁLISE DE SÉRIES TEMPORAIS PARA
PREVER TENDÊNCIAS FUTURAS EM ATIVOS LISTADOS NA B3**

Trabalho de Conclusão de Curso apresentado
ao Curso de Especialização em Ciência de
Dados e Big Data como requisito parcial à
obtenção do título de especialista.

Belo Horizonte

2020

SUMÁRIO

1. Introdução.....	4
1.1. Contextualização	4
1.2. O problema proposto	6
2. Coleta de Dados.....	7
3. Processamento/Tratamento de Dados.....	8
4. Análise e Exploração dos Dados	9
5. Criação de Modelos de Machine Learning	16
5.1. Dividindo o DataFrame	17
5.2. Implementação do AutoArima.....	18
5.3. Treino e Teste do Modelo	21
6. Apresentação dos Resultados	22
7. Links.....	24
REFERÊNCIAS.....	25

1. Introdução

1.1. Contextualização

1.1.1. Investimento em Mercado de Capitais

O investimento em ações sempre foi um grande desafio para os investidores. A oportunidade de grandes ganhos que essa modalidade oferece vem acompanhada de enormes riscos. E embora o comportamento desse mercado seja aparentemente aleatório, diversas técnicas foram desenvolvidas e continuam sendo aperfeiçoadas por investidores e operadores profissionais para tentar prever os próximos movimentos que ocorrerão e, assim, melhorar seus retornos. Existem dois tipos principais de análise que são usadas há muito tempo para prever a tendência do mercado, e que podem ser utilizados separadamente ou em conjunto, dependendo da estratégia escolhida: a análise fundamentalista e a análise técnica.

No primeiro tipo, o valor das ações de uma companhia é determinado por fatores econômicos e pelo desempenho financeiro da própria companhia. Eventuais flutuações nos preços são corrigidas pelo mercado nos próximos períodos uma vez que a análise fundamentalista afirma que o preço de uma ação é sempre justo. Esse tipo de análise é amplamente utilizado por investidores que seguem a estratégia “*Buy and Hold*”, em que as ações são adquiridas visando o longo-prazo, ou seja, uma mesma ação pode ser mantida por anos ou décadas.

Já a análise técnica, por sua vez, considera que toda a informação utilizável já está contida no valor das ações e, portanto, os únicos dados usados na análise do mercado podem ser os valores anteriores deste próprio mercado. Ela também afirma que os movimentos do mercado que ocorrem no passado tendem a se repetir no futuro, o que reforça a necessidade de observar estes valores passados quando se quer prever os futuros. Esse tipo de análise, por sua vez, destaca-se entre os investidores que seguem a metodologia do *Day Trade* e *Swing Trade*, voltadas para investimentos a curto e médio prazo.

Essa última modalidade de negociação em bolsa, o *Swing Trade*, é uma das principais estratégias de investimento utilizadas para negociações no mercado de

capitais. Ela é caracterizada como uma forma de negociação de ativos que oferece menos riscos e maior possibilidade de ganhos aos investidores voltados para aplicações de curto e médio prazo. Geralmente, o tempo que o *Swing Trader*, nome dado a este tipo investidor, mantém os papéis em sua posse varia de três dias a alguns meses. Porém, a estratégia adotada depende do momento de mercado e da posição desejada.

1.1.2. Séries Temporais

As séries temporais são classificadas como uma coleção de observações realizadas ao longo do tempo sobre uma determinada variável, ordenadas e registradas a períodos regulares. Neste tipo de conjunto de dados, as observações vizinhas são dependentes entre si e, tal dependência, é importante objeto de análise e modelagem. Tais dados podem ser encontrados nos mais diversos campos de conhecimento como Economia (taxa mensal de desemprego, produção industrial, valores mensais do IPC-A), Medicina (eletrocardiograma, eletroencefalograma), Epidemiologia (número diário de casos de Covid-19), Meteorologia (temperatura diária, precipitação pluviométrica, velocidade do vento), entre vários outros.

A análise desse tipo de dados é uma importante ferramenta estatística para entender o mercado e possibilitar a formulação de planos de ação e estratégias. Ela permite avaliar o comportamento de uma variável ao longo do tempo e identificar padrões não aleatórios, períodos de crescimento/decrescimento, sazonalidade e, até mesmo, previsão de observações futuras.

O conceito básico que norteia estas análises está relacionado a dependência entre os valores que compõem a série. Parte-se da premissa de que há um sistema causal mais ou menos constante, relacionado com o tempo, que exerceu influência sobre os dados no passado e pode continuar a fazê-lo no futuro. Este sistema causal costuma atuar criando padrões não aleatórios que podem ser detectados mediante algum processo estatístico.

A possibilidade de prever valores e comportamentos futuros tem se tornado de fundamental importância em processos de análise e tomada de decisões nos mais diversos segmentos. Essa característica preditiva que a análise de séries temporais

oferece é um problema que tem recebido especial atenção nos últimos anos e continua sendo um dos maiores desafios para a estatística e a computação.

Uma das áreas em que a análise preditiva se mostra uma ferramenta útil é a Econômica, na qual dados como os de negociações de ativos na B3 (Brasil Bolsa Balcão) podem ser expressos em séries temporais e ter seus comportamentos e tendências analisados. Esses resultados são interessantes para os investidores que realizam operações de curto e médio prazo na bolsa, pois junto aos demais dados que possuam da ação podem melhorar e facilitar a tomada de decisões quanto a compra e venda de ativos.

1.2. O problema proposto

Os investidores que optam por operar na modalidade *Swing Trade* utilizam a análise técnica com o objetivo de encontrar tendências de mercado e os melhores momentos de negociação para obter melhores lucros. E é neste ponto que a análise preditiva de séries temporais pode ser empregada como uma ferramenta a mais no processo de tomada de decisão.

Como forma de auxiliar aos investidores focados nesse tipo de negociação, este trabalho propõe um projeto de análise de dados cujo objetivo é prever tendências futuras de crescimento e decréscimo em cotações de ativos listados na B3. Dessa maneira, os resultados obtidos poderão ser utilizados como uma ferramenta de auxílio e facilitação ao *Swing Trader* para determinar os melhores momentos para compra e venda de ativos.

Para tal, serão utilizados os dados históricos de negociações de ativos listados na B3, coletados do Yahoo Finance. O projeto desenvolvido permitirá ao usuário a escolha do ativo objeto de análise. Porém para apresentação do projeto, serão utilizados os dados de negociação do ativo PETR4 no período de 02 de janeiro de 2001 a 15 de abril de 2020. Esse ativo corresponde as ações preferencias da empresa Petrobras (Petróleo Brasileiro S.A.).

2. Coleta de Dados

A coleta dos dados é feita por meio da biblioteca Pandas DataReader, disponível para linguagem Python. Conforme definição da própria documentação dessa biblioteca, ela consiste em uma ferramenta de acesso remoto a dados atualizados e que é capaz de extrair dados de diversas fontes da Internet e gerar um DataFrame em Pandas como resultado. Para esse projeto, os dados são coletados do site Yahoo Finance, que fornece dados sobre cotações históricas de ações listadas em diversas bolsas ao redor do mundo e é atualizado diariamente.

Na figura 1, logo abaixo, está expressa a implementação necessária para realizar tal importação dos dados.

```
import pandas_datareader.data as wb
ds_acao = pandas.DataFrame()
ds_acao = wb.DataReader(name='PETR4.SA', data_source='yahoo', start='2001-1-1')
```

Figura 1 - Código usado para Coleta dos Dados

Nas linhas de código acima, primeiramente é inicializado um Pandas DataFrame denominado *ds_acao* e que irá guardar os dados coletados. Em seguida, a função DataReader é executada para os seguintes parâmetros:

- *Name = PETR4.SA*: corresponde ao código de identificação do ativo listado na bolsa.
- *Data_source = 'yahoo'*: informa à função qual interface da internet ela deve buscar para fazer a coleta dos dados
- *Start = '2001-1-1'*: data do início do período pelo qual a função deve buscar os dados, sendo que o período corresponderá da data informada nesse campo até a data de execução da função.

O resultado dessa execução gera como retorno um DataFrame com os dados de cada dia de negociação do ativo na bolsa dentro do período indicado. Dentre os dados coletados estão as informações de Data de Negociação, Valores de Abertura, Fechamento, Máxima, Mínima, Fechamento Ajustado do Ativo e Volume de Negociações. Na tabela 1, abaixo, estão listados os campos e sua descrições:

Coluna	Descrição	Tipo
Date	Data de realização do pregão na B3	DateTime
High	Preço mais alto que o ativo foi negociado	Float64
Low	Preço mais baixo que o ativo foi negociado	Float64
Open	Preço do primeiro negócio realizado no pregão	Float64
Close	Preço do último negócio realizado no pregão	Float64
Volume	Quantidade de títulos negociados no pregão	Float64
Adj Close	Preço de fechamento após os ajustes para todos os desdobramentos e distribuições de dividendos aplicáveis	Float64

Tabela 1 - Campos do DataFrame gerado na Coleta de Dados

3. Processamento/Tratamento de Dados

O DataFrame resultante da coleta de dados anteriormente expressa é composto por 7 (sete) colunas e 4.732 (quatro mil, setecentos e trinta e duas) linhas. A descrição dos dados que o compõe pode ser vista na figura 2.

	High	Low	Open	Close	Volume	Adj Close
count	4732.000000	4732.000000	4732.000000	4732.000000	4.732000e+03	4732.000000
mean	18.961988	18.370575	18.679532	18.646553	4.450201e+07	15.665034
std	9.580042	9.269550	9.437227	9.397593	7.224981e+07	8.215299
min	4.270000	4.120000	4.200000	4.200000	0.000000e+00	3.016177
25%	10.720000	10.362500	10.562500	10.570900	2.029478e+07	8.256634
50%	18.605000	18.035001	18.350000	18.299999	3.036665e+07	15.664264
75%	25.412500	24.573750	24.900000	24.978750	4.647470e+07	21.072121
max	67.500000	51.950001	67.500000	52.509998	1.336049e+09	42.206223

Figura 2 - Resultado da execução do método Describe sobre os dados coletados

Ressalta se que, como explicado anteriormente, o ativo PETR4 foi utilizado apenas para exemplificação do projeto neste documento. Porém, no projeto original o usuário é o responsável pela escolha de qual será o ativo analisado. Essa escolha é realizada antes da execução da coleta de dados, quando é solicitado ao usuário que informe o código da ação para a qual deseja que a análise seja feita.

Após a coleta, os dados passam por algumas verificações com o objetivo de evitar que ocorram dados faltantes ou duplicados que possam comprometer a análise. Essas verificações são feitas através de dois métodos da biblioteca Pandas, o *dropna* e *drop_duplicates*. Entretanto, para o exemplo utilizado nenhuma dessas duas inconsistências foi identificada e o DataFrame não sofreu nenhuma alteração.

```
ds_acao.dropna(inplace=True)
ds_acao.drop_duplicates(inplace = True)
```

Figura 3 - Execução dos métodos *dropna* e *drop_duplicates*

Verificadas as inconsistências, os dados da coluna Date, que também comporta-se como índice do DataFrame, são convertidos para o formato DateTime. Essa conversão visa transformar os dados para um formato de data propriamente dito e, assim, permitir a correta interpretação e manipulação dessas informações. Seguido a conversão, os dados são ordenados para garantir que sejam listados de forma crescente cronologicamente.

```
ds_acao.index = pd.to_datetime(ds_acao.index)
ds_acao.sort_index
```

Figura 4 - Execução da conversão de dados e ordenação

Outros processamentos dos dados desse DataFrame não foram entendidos como necessários a execução do projeto, por isso não foram realizados.

4. Análise e Exploração dos Dados

Concluída a coleta e o processamento dos dados, algumas propriedades foram analisadas para melhor conhecimento da base de dados trabalhada. Primeiramente, foi feita a plotagem gráfica dos dados da coluna Close. Essa coluna armazena os valores de fechamento das negociações no período e é objeto de análise preditiva deste trabalho. Na figura 5 é mostrado o resultado dessa plotagem, na qual é perceptível ao longo do tempo o padrão oscilatório característico do mercado de renda variável apresentado.

PETR4 | Valor de Fechamento ao Longo do Tempo

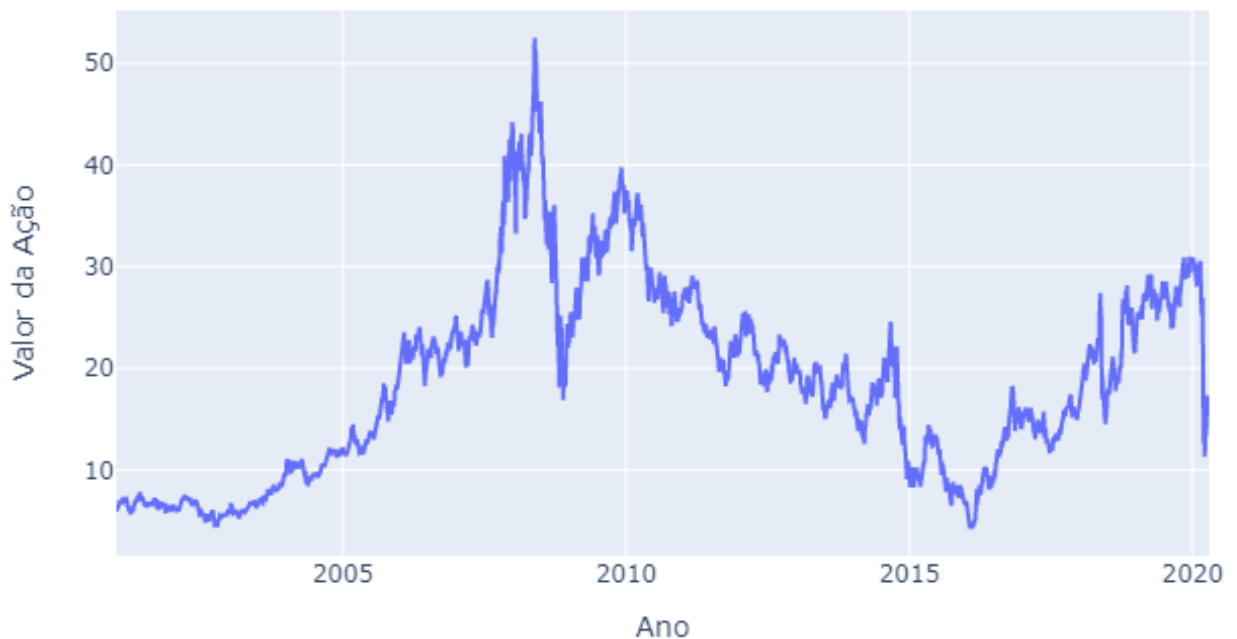


Figura 5 - Gráfico de Valores de Fechamento ao Longo do Tempo

Em seguida é feita a plotagem dos valores de negociação nos últimos 60 (sessenta) dias de operação na bolsa de valores. Entretanto, dessa vez os dados são plotados em um gráfico Candlestick, amplamente utilizado no mercado de ações e na análise técnica. Nele cada candle é formado por quatro pontos que se referem aos preços de abertura, mínima, máxima e fechamento de cada dia de negociação na bolsa. Na figura 6, o resultado dessa plotagem pode ser visto.

A análise desse gráfico permite a identificação de um padrão conhecido na análise técnica como Nuvem Negra. Esse termo é utilizado por analistas e investidores para denotar pontos que podem representar uma possível reversão de um padrão de tendência. No caso da Nuvem Negra, pode sinalizar uma quebra da tendência de alta e sugerir uma queda nos preços no curto-médio prazo. Tal informação pode ser identificada no gráfico entre os dias 19/02/2020 e 20/02/2020, conforme mostrado na figura 7.

PETR4 - Valores de Negociação dos Últimos 60 dias de Operação na B3



Figura 6 - Candlestick - Valores de Negociação dos Últimos 60 Dias de Operação

PETR4 - Valores de Negociação dos Últimos 60 dias de Operação na B3

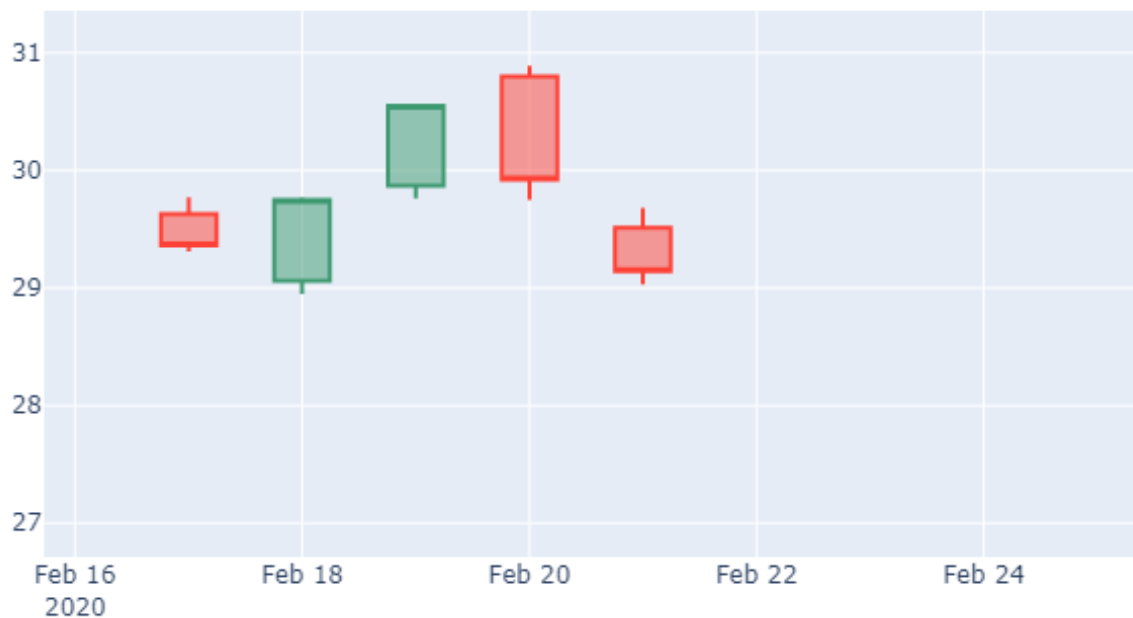


Figura 7 - Padrão de Nuvem Negra Identificado na Análise Gráfica

O coeficiente de correlação entre o valor de fechamento e os outros valores de negociação também foi analisado. Para o cálculo foi utilizado o método de Pearson,

que mede o grau de correlação entre duas variáveis quantitativas em um índice adimensional com valores situados entre -1 e 1. Nesse índice, os valores indicam:

- 1: correlação forte entre as variáveis e de sentido positivo, ou seja, ambas aumentam seu valor simultaneamente;
- -1: correlação forte entre as variáveis, mas em sentido negativo, ou seja, quando o valor de uma variável aumenta o da outra diminui;
- 0: significa que as variáveis não dependem linearmente uma da outra.

Na tabela 2, é exibido o resultado da execução do cálculo do coeficiente de correlação de Pearson:

	High	Low	Open	Close	Volume	Adj Close
High	1.000000	0.990701	0.995294	0.990921	-0.029233	0.972833
Low	0.990701	1.000000	0.995026	0.999465	-0.048138	0.981614
Open	0.995294	0.995026	1.000000	0.994550	-0.036995	0.976592
Close	0.990921	0.999465	0.994550	1.000000	-0.045322	0.981884
Volume	-0.029233	-0.048138	-0.036995	-0.045322	1.000000	-0.046718
Adj Close	0.972833	0.981614	0.976592	0.981884	-0.046718	1.000000

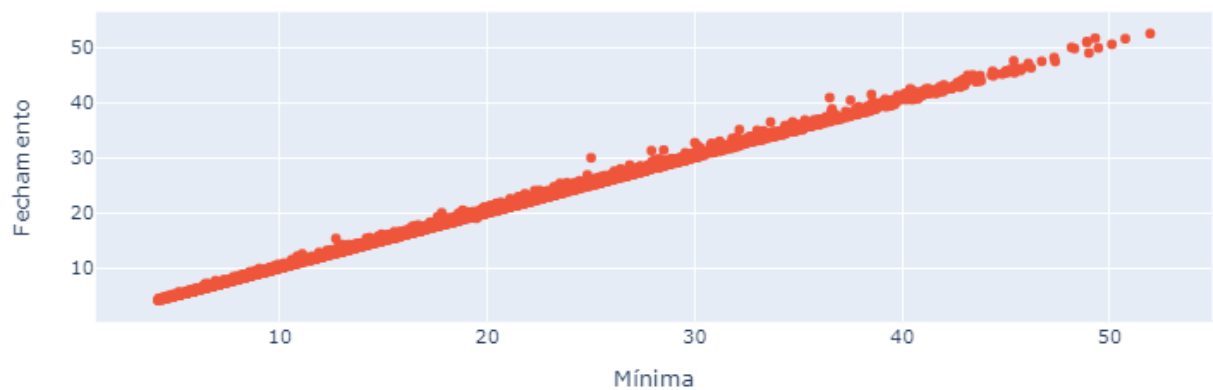
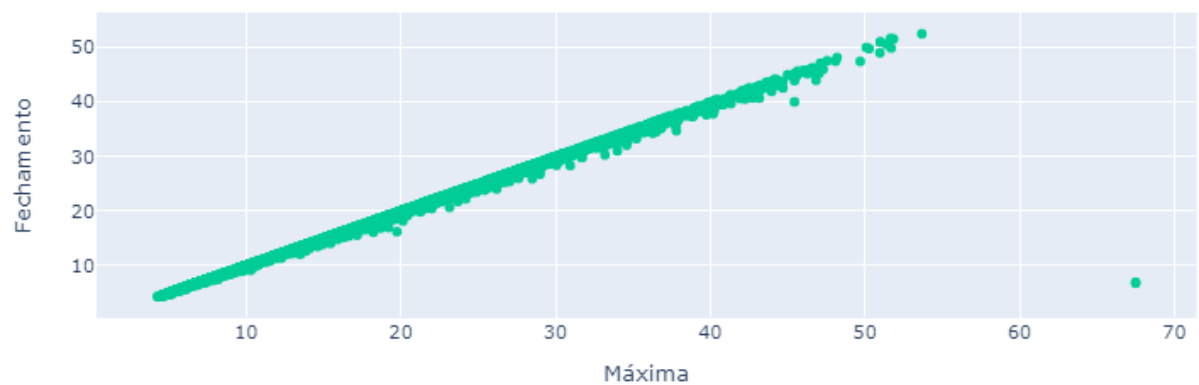
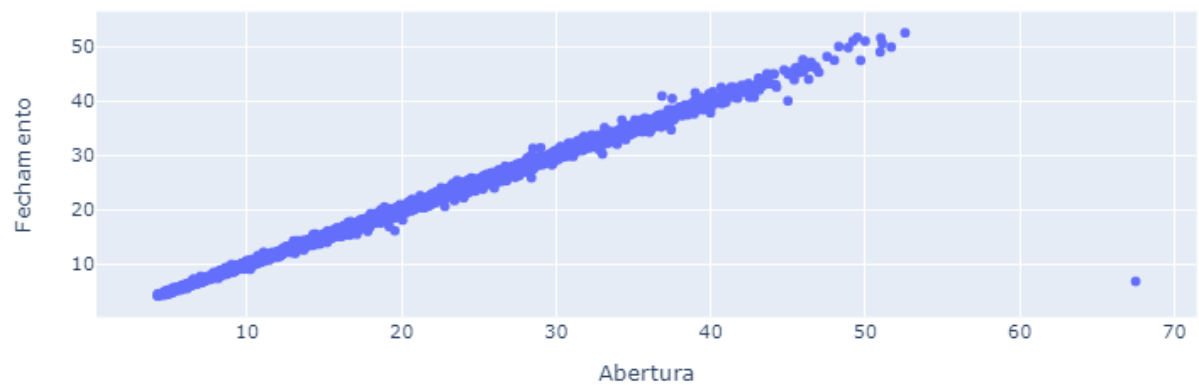
Tabela 2 - Coeficiente de Correlação de Pearson

Ao analisar os resultados para a variável Close é perceptível que há uma forte correlação entre ela e os valores de abertura, mínima e máxima, todos na faixa de 0,99. Com relação ao valor de fechamento ajustado também há uma forte correlação, porém ligeiramente menor que a com os outros valores, na faixa de 0,98. Já em relação ao volume, a correlação entre as variáveis é fraca e com comportamento inverso, bem próxima ao 0.

Para melhorar a visualização dos resultados obtidos no cálculo, foi feita também a plotagem do gráfico de dispersão entre o valor de fechamento e os demais valores que compõem o DataFrame. Essa plotagem é exibida na figura 8 e é possível perceber que nos gráficos referentes a correlação entre o valor de fechamento e os valores de abertura, mínima e máxima há um crescimento linear quase perfeito. No gráfico do Valor de Fechamento x Valor de Fechamento Ajustado o crescimento linear também é facilmente percebido, porém não é tanto quanto nos outros. Já no

gráfico do Valor de Fechamento x Volume não é possível estimar nenhum crescimento linear e percebe-se que os maiores volumes de negociação desse ativo historicamente ocorreram quando o valor da ação estava próximo ao seu valor médio.

PETR4 | Correlação entre Valores



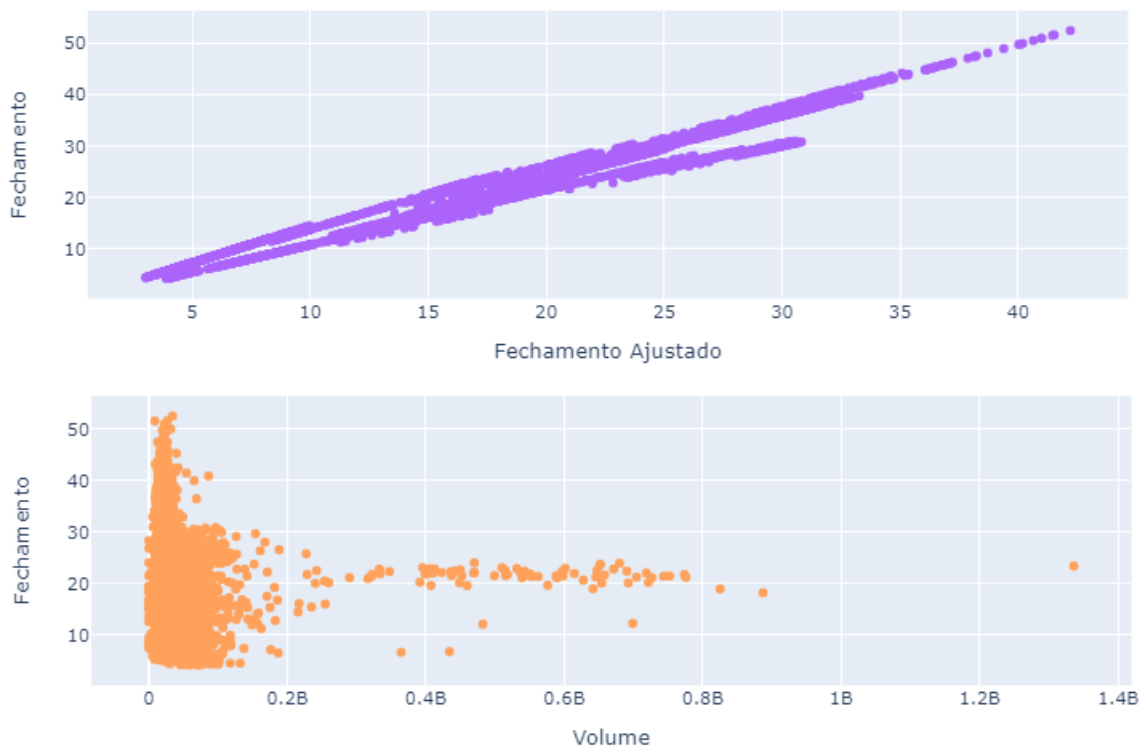


Figura 8 - Gráficos de Dispersão da Correlação Entre o Valor de Fechamento e Demais Valores

Outra característica analisada foi o retorno e o risco que os dados históricos desse ativo sugerem. Para o cálculo do retorno optou-se por utilizar o log retorno, que consiste em uma forma de normalizar a distribuição dos retornos e cuja fórmula pode ser vista na figura 9. Além dessa normalização, esse tipo de cálculo também dá um peso igual aos retornos positivos e negativos. No retorno percentual, mais comumente utilizado, o máximo que um retorno negativo conseguiria atingir é 100%, enquanto um retorno positivo tem ganho máximo infinito. Essa característica do retorno percentual provoca um desequilíbrio entre os pesos dados aos retornos positivos e negativos, o que não ocorre no log retorno.

$$\text{Log Retorno} = \text{Ln} \left(\frac{\text{Preço}_{\text{tempo final}}}{\text{Preço}_{\text{tempo inicial}}} \right)$$

Figura 9 - Fórmula do Log Retorno

O resultado da aplicação desse método sobre a coluna Close do DataFrame foi plotado no gráfico exibido na figura 10 e demonstra que essa ação possui o perfil

oscilatório característico do mercado acionário, entretanto não se nota oscilações extremamente altas dos retornos do ativo. Em geral, o retorno desse ativo variou abaixo dos 10% (dez por cento) para mais ou para menos.

PETR4 | Retornos Diários

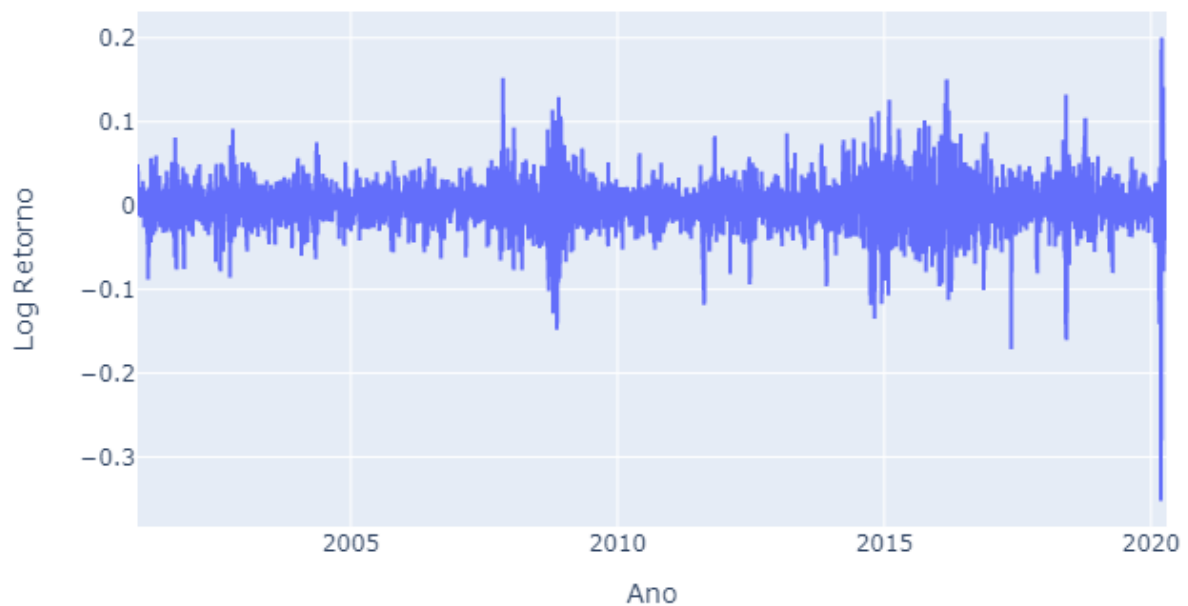


Figura 10 - Gráfico do Retorno Diário do Ativo

Baseado no resultado do log retorno diário, foram calculados também o retorno médio diário, o retorno médio anual e desvio padrão (risco). Para o retorno anual, considerou-se 250 (duzentos e cinquenta) dias de negociação na bolsa de valores. Os resultados obtidos para os cálculos destas medidas foram:

- Retorno Médio Diário: 0,002%
- Retorno Médio Anual: 5,2%
- Desvio Padrão (Risco): 5,2%

Por fim, foi testada a estacionariedade da série temporal formada pelos valores da coluna Close por meio do teste estatístico de Dickey-Fuller. O resultado desse teste indica que a série é não-estacionária pois apresenta um valor P de 17% e o valor crítico de 5% é menor que o valor do teste estatístico calculado. Porém, pela análise gráfica percebe-se que a série apresenta um comportamento não estacionário homogêneo, ou seja, não possui um comportamento explosivo e seus valores

flutuam ao redor de um nível por um certo tempo e ao mudar de nível esse comportamento se repete. Na figura 11 é exibido o resultado para esse teste.

PETR4 | Retornos Diários



```

Resultado do Teste de Dickey-Fuller:
Teste Estatístico      -2.291926
p-value                0.174585
#Lags                  30.000000
Número de Observações  4694.000000
Valores Críticos(1%)  -3.431744
Valores Críticos(5%)  -2.862156
Valores Críticos(10%) -2.567098
dtype: float64

```

Figura 11 - Teste de Estacionariedade

5. Criação de Modelos de Machine Learning

Nesse ponto do trabalho, o modelo preditivo é então implementado e o objetivo deste projeto é de fato abordado. Para essa implementação optou-se por utilizar o método estatístico ARIMA (AutoRegressive Integrated Moving Average), que utiliza dados passados para prever o futuro por meio de dois principais recursos: a autocorrelação e médias móveis. A execução desse modelo requer três termos como parâmetros e que são denotados por:

- p : ordem do modelo autorregressivo, ou seja, o número de observações passadas;
- d : grau de diferenciação;
- q : ordem do modelo de média móvel, ou seja, o tamanho da função “janela” nos dados de suas séries temporais.

A definição de uma boa configuração desses termos para a série temporal em que o método é aplicado está estreitamente atrelada a obtenção de bons resultados. E dada a característica deste projeto de fornecer uma ferramenta generalista em que o usuário tem a liberdade de definir qual ativo e período será analisado, o DataFrame gerado na fase de coleta será diferente para cada configuração. Consequentemente, a configuração dos termos que resultariam em resultados melhores também poderá variar para série temporal gerada.

Para tratar esse problema, viu-se necessário a implementação de um método ARI-MA em que a melhor configuração dos termos p , d e q pudesse ser identificada automaticamente. A forma escolhida para realizar essa implementação foi através da utilização da biblioteca *pmdarima*, disponível para a linguagem Python. Ela é, conforme sua documentação, caracterizada como uma biblioteca estatística projetada para trabalhar com séries temporais e criar uma interface familiar para aplicação de funções estatísticas.

5.1. Dividindo o DataFrame

O DataFrame coletado na fase inicial foi dividido em três para possibilitar o treino, teste e a validação do modelo. Essa divisão foi feita por datas, sendo:

- Intervalo de Treino: 01/01/2001 a 01/01/2018
- Intervalo de Teste: 02/01/2018 a 01/01/2020
- Intervalo de Validação: 02/01/2020 a 15/04/2020

Os intervalos podem ser vistos também na figura 12, que demonstra graficamente os intervalos definidos.

PETR4 | Bases de Treino, Teste e Validação



Figura 12 - Intervalos de Treino, Teste e Validação

5.2. Implementação do AutoArima

A implementação do método AutoArima foi dividida em 3 funções para possibilitar o melhor reaproveitamento de código no desenvolvimento do projeto.

5.2.1. Função *def_auto*

A primeira função implementada é responsável por resolver o problema da definição dos termos p , d e q . Ela recebe como parâmetro uma série, denominada na função por `ts_treino`, com os valores que compõem o intervalo de treino da série temporal analisada e para qual os termos estão sendo buscados. Ao final da execução, ela retorna ao seu ponto de chamada o conteúdo da variável `auto` com o modelo obtido para execução do ARIMA.

Essa busca pela melhor configuração é feita através da função *auto_arima* da biblioteca *pmdarima*. Para a execução, foram utilizados os seguintes parâmetros:

- `y`: série com os valores do intervalo de análise (`ts_treino`);
- `seasonal`: se a série temporal tem comportamento sazonal, o que não ocorre na série analisada;
- `stepwise`: indica que o algoritmo stepwise descrito em Hyndman e Khandakar (2008) deve ser usado para identificar os parâmetros ideais do modelo;

- `suppress_warnings`: durante a execução do modelo estatístico, diversas mensagens de erro podem ser exibidas. Para reprimir essa exibição, esse parâmetro recebeu o valor `True`;
- `error_action`: indica que o método deve ignorar quando ocorrerem problemas de estacionariedade durante a aplicação;
- `max_order`: verifica se a soma de `p` e `q` é maior ou igual a `max_order`, o que indica que modelo não será adequado a esses parâmetros e passará para a próxima combinação. O padrão do método é 5, mas quando é atribuído o valor `None` significa que não há restrições.
- `trace`: indica que informações de depuração devem ser impressas durante a execução.

```
def def_auto(ts_treino):
    auto = pm.auto_arima(ts_treino, seasonal=False, stepwise=True,
                        suppress_warnings=True, error_action="ignore",
                        max_order=None, trace=True)
    return auto
```

Figura 13 - Função `def_auto`

5.2.2. Função *prever_um_passo*

A segunda função implementada tem como finalidade prever o próximo valor futuro a cada passo do modelo. Essa função recebe como parâmetro o modelo ARIMA definido na execução da função anterior e o utilizada para realizar a previsão do próximo valor. A previsão é feito usando-se o método *predict* e o valor previsto na execução da função é retornado a chamada inicial.

```
def prever_um_passo(modelo):
    prev = modelo.predict(n_periods=1)
    return prev.tolist()[0]
```

Figura 14 - Função `prever_um_passo`

5.2.3. Função *aplicar_arima*

A última função implementada é a responsável pelo controle do fluxo de execução do modelo ARIMA. Essa função recebe como parâmetro duas séries, denominadas por *ts_treino* e *ts_teste*, com os valores que compõem o intervalo de treino e de teste da série temporal analisada. Como resultado, a função retorna o intervalo de previsões a que o modelo chegou.

O fluxo de execução que essa função estabelece consiste em, inicialmente, realizar uma chamada à função *def_auto* para definição do modelo ARIMA e que cujo retorno é salvo na variável *modelo*. Os valores da ordem definida e armazenadas em *modelo* são impressos na tela.

Uma lista para armazenar os valores das previsões é inicializada com o valor vazio. Na sequência, uma estrutura de repetição é criada para percorrer toda a série *ts_teste*. A cada execução do laço de repetição, a função *prever_um_passo* é chamada para prever qual o próximo valor da série. O retorno dela é armazenado na variável *prev* e cujo valor é adicionado à lista *previsoes*. Finalizando o ciclo de execução, o modelo ARIMA é atualizado para possibilitar a próxima previsão. E esse processo se repete enquanto houverem itens a serem percorridos em *ts_teste*.

Ao final da repetição, os valores previstos pelo modelo são avaliados quanto ao seu erro. Para isso, são utilizadas duas medidas de precisão, o SMAPE (Erro Percentual Absoluto Médio Simétrico) e o MSE (Erro Quadrático Médio). Por fim, a lista com os valores previstos é retornada à chamada principal.

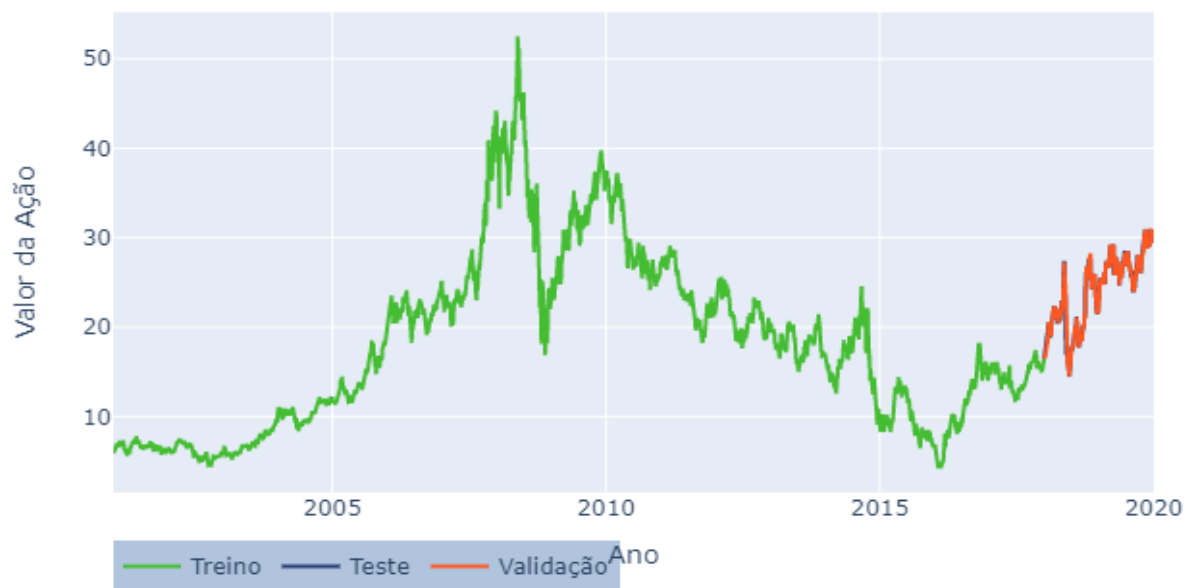
```
def aplicar_arima(ts_treino, ts_teste):  
    modelo = def_auto(ts_treino)  
    print('Termos p, d, q:', modelo.order)  
    previsoes = []  
    for nova_obs in ts_teste:  
        prev = prever_um_passo(modelo)  
        previsoes.append(prev)  
        modelo.update(nova_obs)  
    print(f"Mean squared error: {mean_squared_error(ts_teste, previsoes)}")  
    print(f"SMAPE: {smape(ts_teste, previsoes)}")  
    return previsoes
```

Figura 15 - Função aplicar_arima

5.3. Treino e Teste do Modelo

Implementado o modelo, o mesmo foi testado com base nos intervalos de treino e teste definidos para o projeto. A execução do modelo para esses valores retornou o resultado apresentado nas figuras 16 e 17, que representam graficamente os valores previstos em comparação aos valores reais.

PETR4 | Bases de Treino, Teste e Validação



PETR4 | Bases de Treino, Teste e Validação

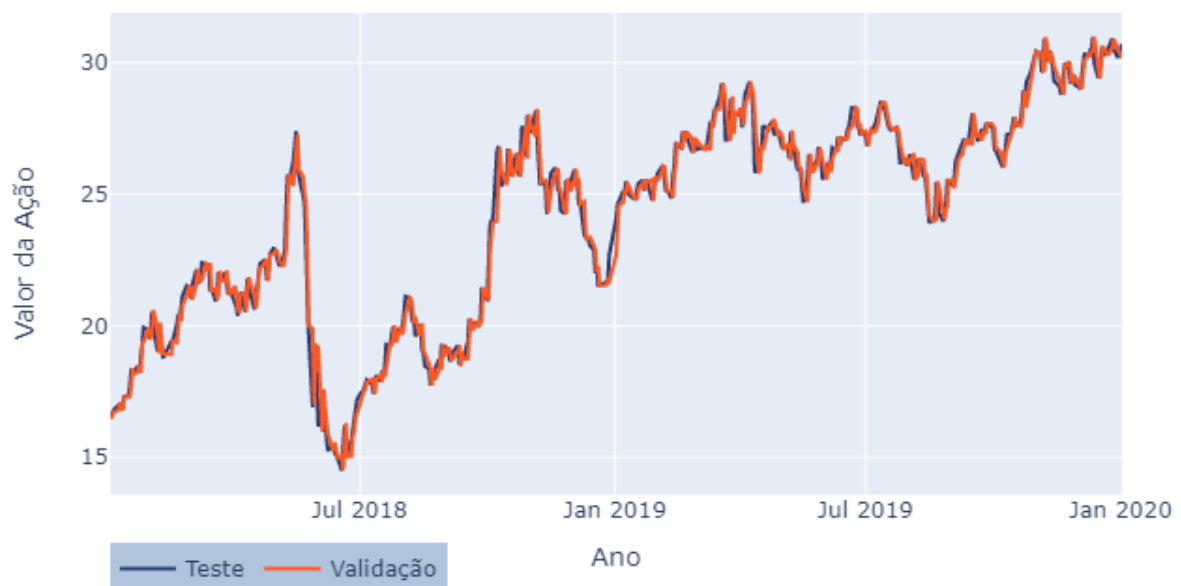


Figura 16 - Resultado do Treino e Teste do Modelo

Como pode ser visto pelo gráfico, o modelo testado apresentou um bom resultado quanto as previsões. Há erros quanto aos valores futuros preditos, porém o modelo acerta com grande eficácia as tendências de valorização e queda que ocorrem durante o período testado.

6. Apresentação dos Resultados

Definido o modelo estatístico e testado, foi realizada a previsão dos valores de fechamento do período entre 02/01/2020 e 15/04/2020. Embora esse período escolhido não seja propriamente um período futuro em relação a data de desenvolvimento deste projeto, ele foi escolhido para que haja uma melhor validação dos resultados obtidos. Ao ser aplicado o modelo estatístico sobre esse período, obteve-se o resultado demonstrado a seguir.

Resultado das medidas de precisão de erro:

- MSE: 1.7417
- SMAPE: 4.1577

PETR4 | Bases de Treino, Teste e Validação



Figura 17 - Resultado da Validação do Modelo

PETR4 | Bases de Treino, Teste e Validação

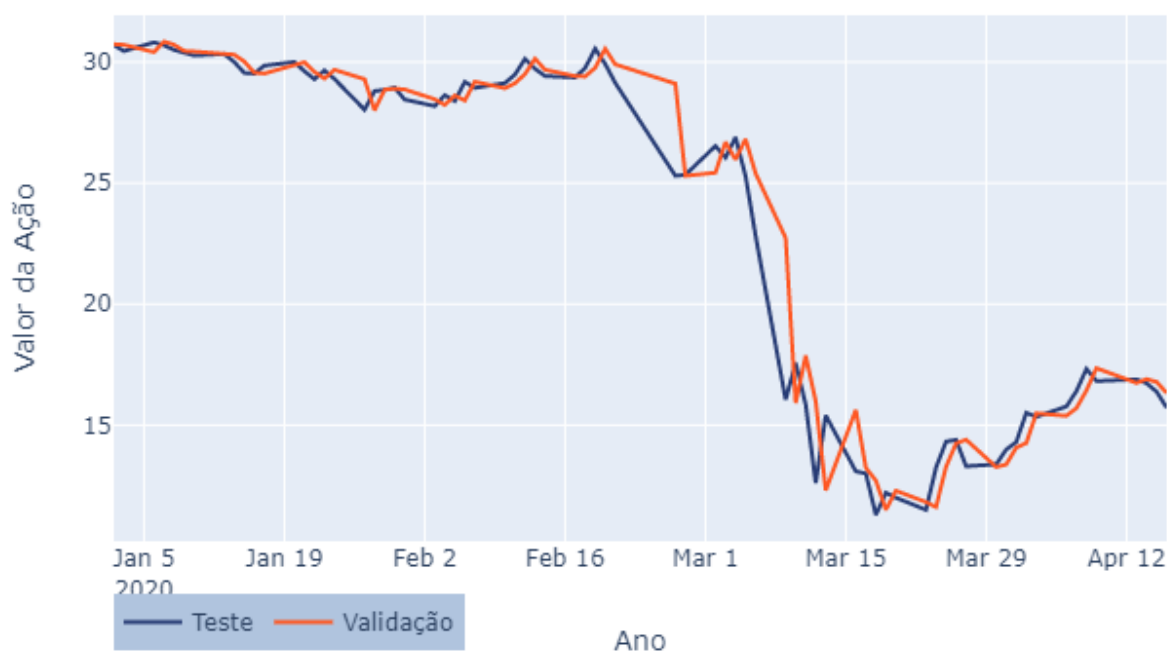


Figura 18 - Detalhamento do Resultado Obtido

O resultado obtido durante a validação do modelo demonstrou um resultado menos preciso em comparação com o resultado do teste. Porém deve-se considerar o fato de que no período escolhido para validação do modelo a economia global sofreu uma de suas mais bruscas reversões. E mesmo assim, é possível perceber que o modelo consegue prever tendências de valorização e desvalorização no período muito próximas as que ocorreram, apenas com pouco atraso em alguns casos.

Sendo assim, o modelo mostra-se como uma boa alternativa para auxiliar investidores de médio prazo. É perceptível que o modelo demonstra sua capacidade de prever tendências na bolsa de valores e que aliado com as outras ferramentas de análise já comumente utilizada, é capaz de melhorar e facilitar a definição dos melhores momentos para movimentações no mercado financeiro e potencializar os ganhos.

“As quatro palavras mais perigosas em investimento são: ‘Dessa vez é diferente’”

Sir John Templeton

7. Links

Repositório do GitHub:

https://github.com/dimasparreiras/Valores_Futuros_B3_ARIMA

Vídeo de Apresentação:

https://drive.google.com/drive/folders/1A_xchDS1ybg91j6VKoaUaxZp3nlo4k-a?usp=sharing

REFERÊNCIAS

SANTANA, Rodrigo. **Como Utilizar Machine Learning para Prever Preços de Ações da Bolsa de Valores**: o estudo de caso completo. 2017. Disponível em: <https://minerandodados.com.br/prevendo-precos-de-acoes-da-bolsa-de-valores-com-machine-learning/>. Acesso em: 03 fev. 2020.

GONÇALVES, Leonardo. **CARACTERÍSTICAS DAS SÉRIES TEMPORAIS**. 2019. Disponível em: <https://operdata.com.br/blog/caracteristicas-das-series-temporais/>. Acesso em: 05 abr. 2020.

IPPOLITO, Pier Paolo. **Stock Market Analysis Using ARIMA**. 2019. Disponível em: <https://towardsdatascience.com/stock-market-analysis-using-arima-8731ded2447a>. Acesso em: 12 fev. 2020.

VASCONCELLOS, Paulo. **Dicas para criar um modelo de previsão de séries temporais**. 2018. Disponível em: <https://medium.com/techbloghotmart/dicas-para-criar-um-modelo-de-previs%C3%A3o-de-s%C3%A9ries-temporais-d4bb2e32e148>. Acesso em: 03 fev. 2020.

DVORAK, Tomas. **Why isn't out-of-time validation more ubiquitous?** 2019. Disponível em: <https://towardsdatascience.com/why-isnt-out-of-time-validation-more-ubiquitous-7397098c4ab6>. Acesso em: 02 mar. 2020.

MORATO, Lucas. **Introduction to Stock Analysis in Python**. 2020. Disponível em: <https://medium.com/datadriveninvestor/introduction-to-stock-analysis-in-python-574246e689e3>. Acesso em: 20 fev. 2020.

SMITH, Taylor G. **Stock market prediction**. Disponível em: <https://alkaline-ml.com/pmdarima/usecases/stocks.html>. Acesso em: 02 mar. 2020.