

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
NÚCLEO DE EDUCAÇÃO A DISTÂNCIA
Pós-graduação *Lato Sensu* em Ciência de Dados e Big Data

Alysson Cristiano Estevam de Moura

**MODELOS PREDITIVOS UTILIZANDO SÉRIES TEMPORAIS DA BOLSA DE
VALORES**

Belo Horizonte
2020

Alysson Cristiano Estevam de Moura

**MODELOS PREDITIVOS UTILIZANDO SÉRIES TEMPORAIS DA BOLSA DE
VALORES**

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	5
2. Coleta de Dados	5
3. Processamento/Tratamento de Dados	6
4. Análise e Exploração dos Dados	7
4.1. Análise Exploratória da PETR4	7
4.2. Análise Exploratória da ITUB4	11
4.3. Análise Exploratória da VALE3	15
4.4. Análise das ações em conjunto	18
5. Modelos Preditivos para a PETR4.....	20
5.1. Modelo Preditivo com Facebook Prophet	21
5.2. Modelo Preditivo com AUTO-ARIMA	21
5.2. Modelo Preditivo com RNN (LSTM – <i>Long Short Term Memory</i>).....	26
6. Apresentação dos Resultados	29
6.1. Modelo Preditivo com Facebook Prophet	29
6.2. Modelo Preditivo com AUTO-ARIMA	31
6.2. Modelo Preditivo com RNN (LSTM – <i>Long Short Term Memory</i>).....	33
7. Links	34
REFERÊNCIAS.....	35

1. Introdução

1.1. Contextualização

Os dados de preço de ações da Bolsa de Valores possuem informações muito importantes para os investidores. Porém, estes dados são em sua essência muito dinâmicos, não-lineares, não-paramétricos e caóticos por natureza. Isso significa que os investidores devem lidar com séries temporais não-estacionárias e ruidosas.

Além disso, os movimentos de mercado são afetados por diversos fatores macroeconômicos, tais como eventos políticos, políticas empresariais, condições gerais econômicas, índice de preço de commodities, investimentos institucionais, índices econômicos e até mesmo fatores psicológicos dos investidores. O mercado financeiro é caracterizado por incertezas, das quais são associadas com suas expectativas de curto, médio e longo prazos. A criação de modelos capazes de captura as dinâmicas de mercado, tem como objetivo reduzir essas incertezas e observar as tendências do preço das ações para melhor tomada de decisões dos investidores.

Com um número cada vez maior de profissionais de tecnologia entrando no mercado financeiro, diversas técnicas de Modelagem Estatística e Inteligência Artificial tem sido estudadas para aplicação em séries temporais da Bolsa de Valores.

Diante desse contexto, este trabalho visa utilizar técnicas de Mineração de Dados e Modelos Preditivos em séries temporais da Bolsa de Valores B3. Para isso foi desenvolvido um script em Python utilizando diversas bibliotecas, tais como Yahoo Finance API, Keras, Prophet, ARIMA, além das básicas de uso comum da linguagem.

1.2. O problema proposto

O problema estudado consiste no uso de Análise Exploratória e Modelagem Preditiva para extração de informações importantes das séries temporais do preço de fechamento das ações para auxiliar os investidores a terem *insights* para tomada de decisões e, com isso, obterem maior rentabilidade no Mercado de Ações.

Para isso, são analisadas as séries temporais das ações da Petrobras (PETR4), Vale (VALE3) e Itaú (ITUB4) da Bolsa de Valores B3. Assim, tem-se como objetivos dessa análise:

- Fazer a Análise Descritiva dos dados das ações PETR4, VALE3 e ITUB4;
- Verificar a qualidade dos dados utilizando técnicas estatística;
- Analisar de forma comparativa as três ações para verificar a rentabilidade;
- Verificar se há correlação entre as três ações;
- Criar modelos preditivos para a ação PETR4 utilizando a biblioteca Facebook Prophet, AUTO-ARIMA e RNN baseada na tecnologia LSTM.

O período utilizado na Análise Exploratória e treinamento dos modelos de predição corresponde aos anos de 2014 até 2018. Para os dados de teste dos modelos treinados, foi utilizado o ano de 2019.

2. Coleta de Dados

Para o desenvolvimento deste trabalho foi utilizada a *Yahoo Finance API* para obtenção dos datasets das ações da Petrobras (PETR4), Vale (VALE3) e Itaú (ITUB4) da Bolsa de Valores B3. Para a leitura dos dados foi utilizada a biblioteca *Pandas Datareader* em comunicação com a *Yahoo Finance API*.

Os dados são obtidos na hora da execução do script, bastando apenas passar como parâmetros o nome da ação, data de início e data de fim. Temos por exemplo os dados de treinamento da ação PETR4 no período de 01/01/2014 a 31/12/2018:

```
from pandas_datareader import data
train_dataset_petr4 = data.DataReader('PETR4.SA', 'yahoo', '2014-01-01',
'2018-12-31')
```

O dataset obtido para cada ação tem o formato descrito na tabela abaixo:

Nome da coluna	Descrição	Tipo
Date	Data da negociação	Pandas DatetimeIndex
High	Maior preço da ação no dia	Pandas Series
Low	Menor preço da ação no dia	Pandas Series
Open	Preço de abertura da ação no dia	Pandas Series
Close	Preço de fechamento da ação no dia	Pandas Series
Volume	Volume total negociado no dia	Pandas Series
Adj Close	Valor ajustado da ação no dia. Este valor considera desconto de dividendos, splits, inplits, etc.	Pandas Series

Neste trabalho foi utilizado o preço de fechamento (Close) nos modelos de predição.

3. Processamento/Tratamento de Dados

Após o processo de leitura dos dados, obtém-se os seguintes datasets:

Nome do dataset	Descrição
train_dataset_petr4	Conjunto de dados de treinamento da ação PETR4 no período de 01/01/2014 a 31/12/2018.
train_dataset_itub4	Conjunto de dados de treinamento da ação ITUB4 no período de 01/01/2014 a 31/12/2018.
train_dataset_vale3	Conjunto de dados de treinamento da ação VALE3 no período de 01/01/2014 a 31/12/2018.
test_dataset_petr4	Conjunto de dados de teste a ser usado nos modelos preditivos da ação PETR4 no período de 01/01/2019 a 31/12/2019.

Cada dataset de treinamento possui 1242 registros e o de teste da PETR4 possui 247. Não há registros duplicados, já que cada registro corresponde a uma data diferente. Além disso, os dados correspondem a dias em que a Bolsa de Valores está aberta para negociações, assim esses registros correspondem aos dias úteis do ano. Também foi verificado que não há nenhum valor nulo nos dados.

4. Análise e Exploração dos Dados

4.1. Análise Exploratória da PETR4

Após a coleta dos dados, foi realizada uma descrição estatística do dataset conforme mostra a figura 1.

	High	Low	Open	Close	Volume	Adj Close
count	247.000000	247.000000	247.000000	247.000000	2.470000e+02	247.000000
mean	27.556883	26.938057	27.264292	27.248988	5.283958e+07	26.559664
std	1.658077	1.648094	1.668801	1.655566	2.451955e+07	1.724225
min	24.190001	22.280001	22.549999	23.910000	1.904990e+07	23.286510
25%	26.370001	25.815000	26.155000	26.065000	3.731705e+07	25.312547
50%	27.430000	26.870001	27.150000	27.129999	4.850220e+07	26.424419
75%	28.445000	27.870001	28.205000	28.220000	6.118210e+07	27.388570
max	31.230000	30.500000	30.889999	30.969999	2.273076e+08	30.547239

Figura 1: Descrição estatística da PETR4

Em seguida foi verificado que não há valor nulo no dataset:

```

High      0
Low       0
Open      0
Close     0
Volume    0
Adj Close 0
dtype: int64

```

Plotando-se um histograma do preço de fechamento, obtém-se o resultado apresentado na figura 2.

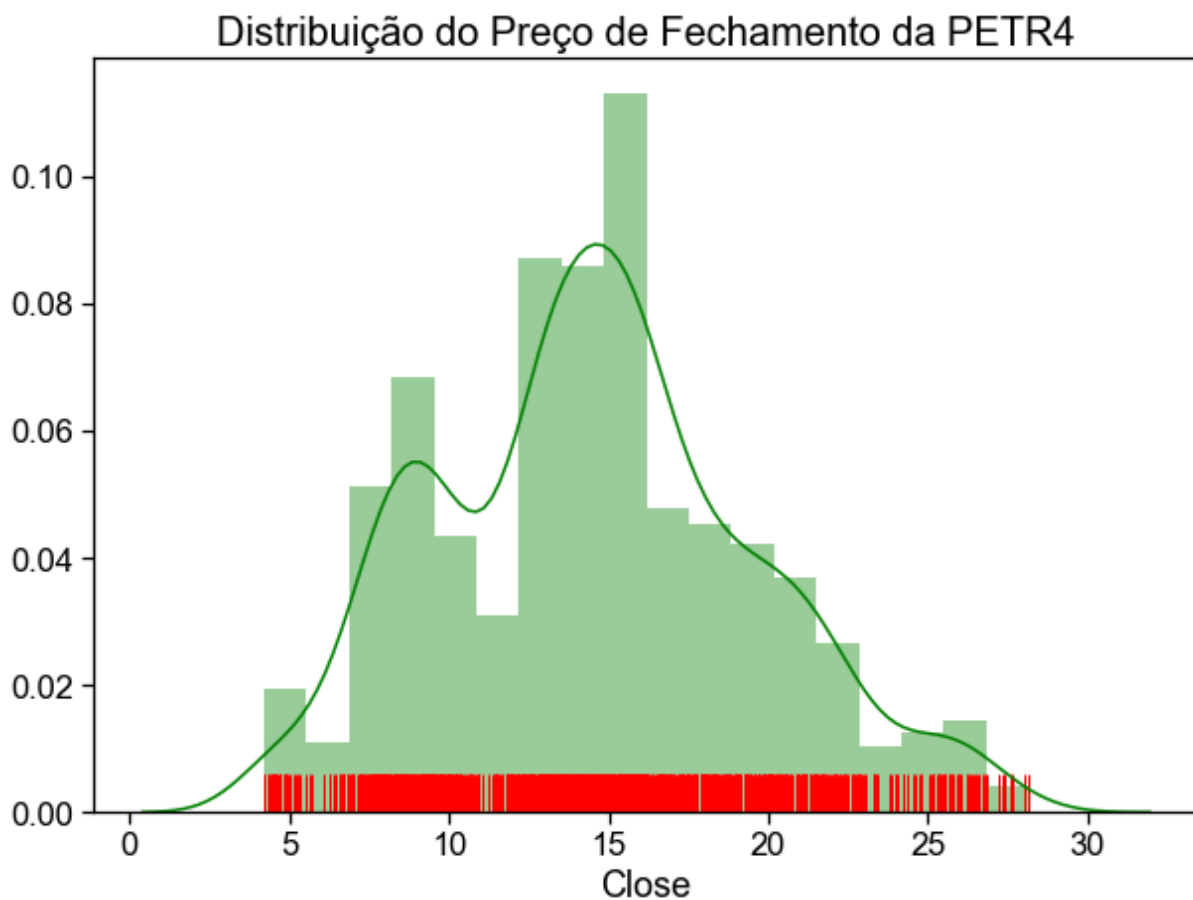


Figura 2: Histograma do preço de fechamento da PETR4

Percebe-se que o histograma apresenta dois picos, representando duas frequências mais elevadas em torno dos preços de fechamento R\$ 8,00 e R\$ 15,00.

Também foi plotado o histórico de preços englobando o preço máximo, mínimo e fechamento, conforme mostra a figura 3.

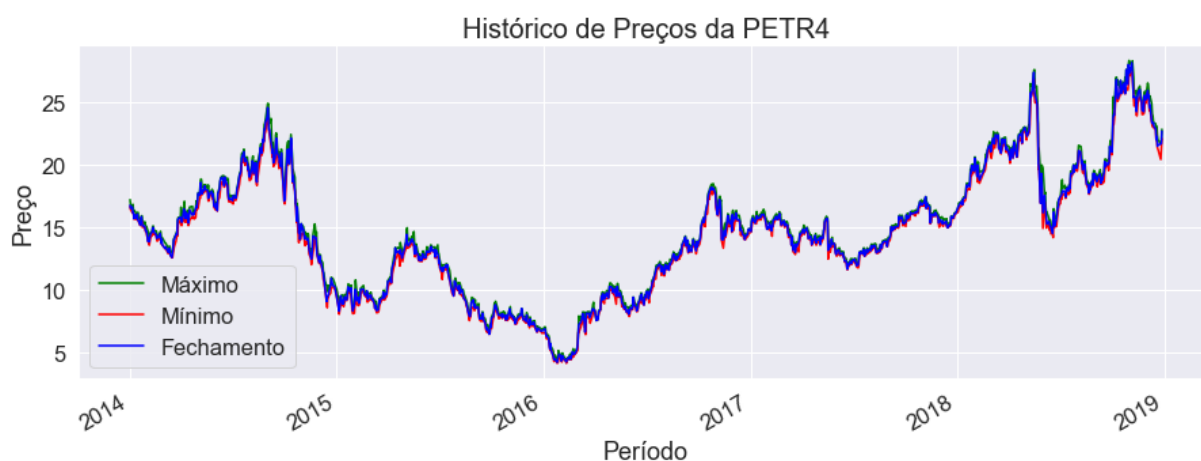


Figura 3: Histórico de Preços da PETR4

Pela figura percebe-se que os três preços estão fortemente correlacionados no longo prazo.

Plotando-se a variação de preço diário (preço de fechamento - preço de abertura), obtém-se o resultado apresentado na figura 4.

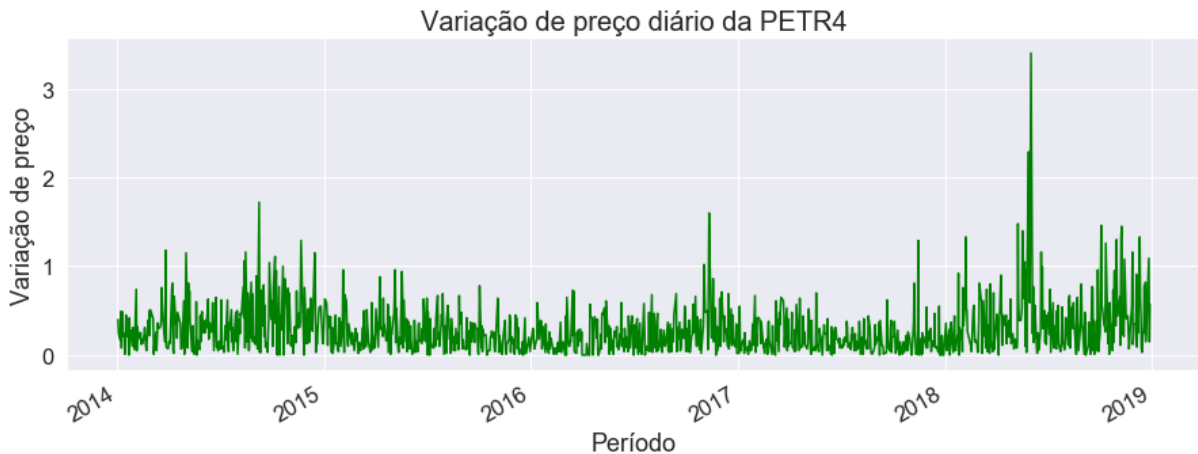


Figura 4: Variação de preço diário (preço de fechamento - preço de abertura) da PETR4

E utilizando a função `argmax()` do Python, verifica-se que a maior variação de preço ocorreu em:

```
Date
2018-06-01    3.4
dtype: float64
```

Verificando no Google Notícias o que aconteceu nesse dia, tem-se as notícias conforme mostra a figura 5.



Figura 5: Notícias da PETR4 encontradas no Google no dia 01/06/2018

Analisando os dados em conjunto com as notícias, percebe-se que o que influenciou essa grande variação foi o anúncio de pedido de demissão do presidente-executivo Pedro Parente.

Plotando-se o gráfico do volume negociado, obtém-se a figura 6.

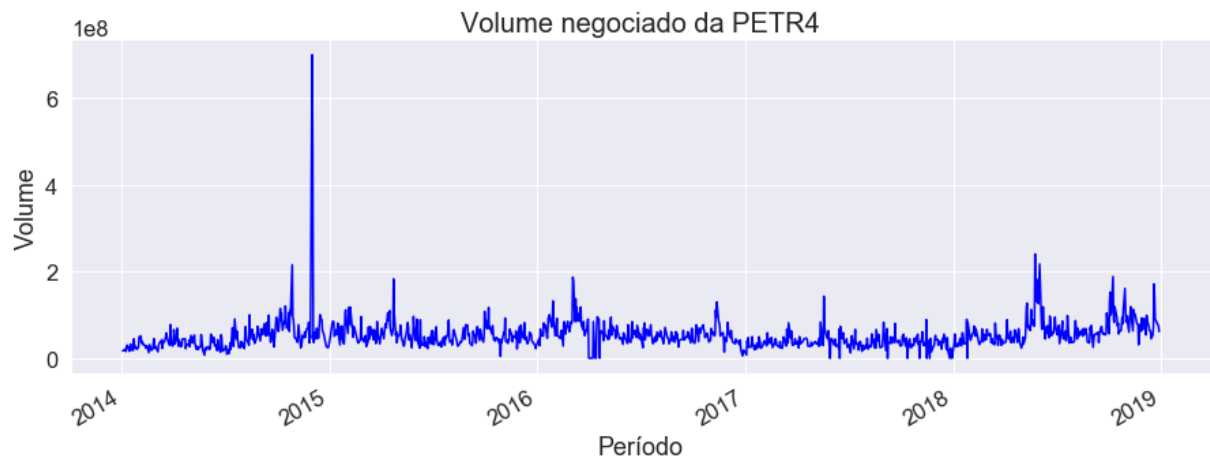


Figura 6: Volume negociado da PETR4

Utilizando-se a função *argmax()* do Python, verifica-se que o dia de maior negociação foi em 01/12/2014:

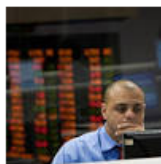
```
High      1.254000e+01
Low       1.221000e+01
Open      1.245000e+01
Close     1.232000e+01
Volume    6.989506e+08
Adj Close 1.143014e+01
Name: 2014-12-01 00:00:00, dtype: float64
```

Verificando no Google Notícias nessa data, obtém-se as notícias mostradas na figura 7.

Rumor de tributação de dividendos e extinção de JCP afunda ...

InfoMoney - 1 de dez. de 2014

Juntamente, ações de peso do índice aceleravam as perdas nesta tarde, como Petrobras (PETR3, R\$ 11,68, -3,87%; PETR4, R\$ 12,45, -2,73%), bancos ...



Analista dá 6 razões pra você zerar sua posição em ações ...

InfoMoney - 1 de dez. de 2014

5º No Brasil ainda temos a Petrobras (PETR3;PETR4) sem balanço do terceiro trimestre de 2014 e está sendo investigada pela SEC (Securities and Exchange ...

Figura 7: Notícias da PETR4 encontradas no Google no dia 01/12/2014

Vê-se que um rumor de tributação de dividendos e extinção de JCP derrubou a Bolsa de Valores no dia 01/12/2014. Além disso, o cenário macroeconômico influenciou muito, pois havia um pessimismo com o noticiário internacional em meio à queda do petróleo e os dados nada positivos vindos da China.

4.2. Análise Exploratória da ITUB4

Após a coleta dos dados, foi realizada uma descrição estatística do dataset conforme mostra a figura 8.

	High	Low	Open	Close	Volume	Adj Close
count	1242.000000	1242.000000	1242.000000	1242.000000	1.242000e+03	1242.000000
mean	23.170940	22.597220	22.886952	22.890062	2.136467e+07	19.835499
std	5.772940	5.683730	5.734574	5.728302	1.086807e+07	6.390696
min	14.072700	13.824200	13.939400	13.981800	0.000000e+00	10.942980
25%	18.715475	18.181801	18.430300	18.422699	1.408901e+07	14.538179
50%	21.477400	20.972449	21.236500	21.220349	1.961033e+07	17.094230
75%	27.713350	27.176675	27.371650	27.386700	2.568126e+07	25.346120
max	36.830002	36.020000	36.799999	36.349998	1.049172e+08	34.282822

Figura 8: Descrição estatística da ITUB4

Em seguida foi verificado que não há valor nulo no dataset:

```

High      0
Low       0
Open      0
Close     0
Volume    0
Adj Close 0
dtype: int64

```

Plotando-se um histograma do preço de fechamento, obtém-se o resultado apresentado na figura 9.

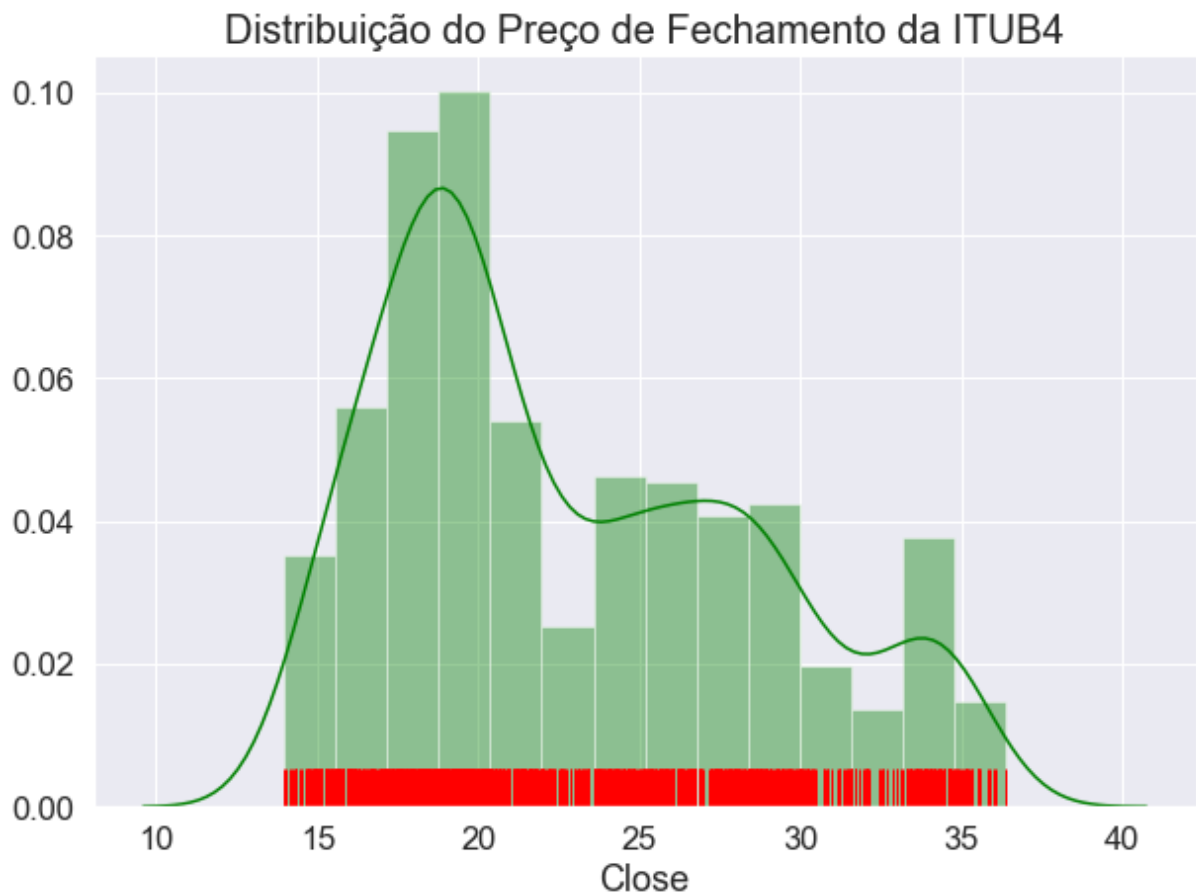


Figura 9: Histograma do preço de fechamento da ITUB4

Percebe-se que o histograma apresenta um pico principal em torno do preço de fechamento R\$ 18,00.

Também foi plotado o histórico de preços englobando o preço máximo, mínimo e fechamento, conforme mostra a figura 10.

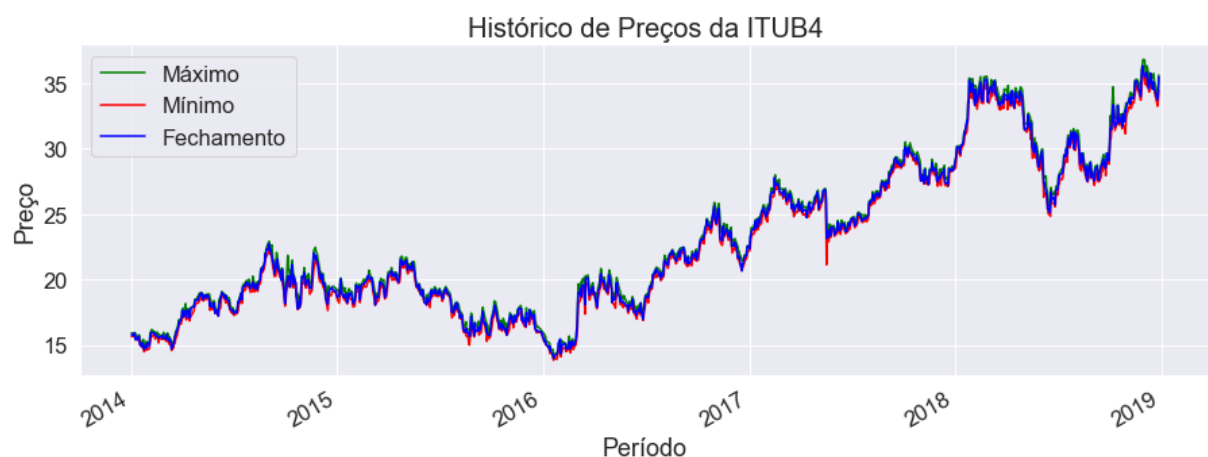


Figura 10: Histórico de Preços da ITUB4

Pela figura percebe-se que os três preços estão fortemente correlacionados no longo prazo.

Plotando-se a variação de preço diário (preço de fechamento – preço de abertura), obtém-se o resultado apresentado na figura 11.

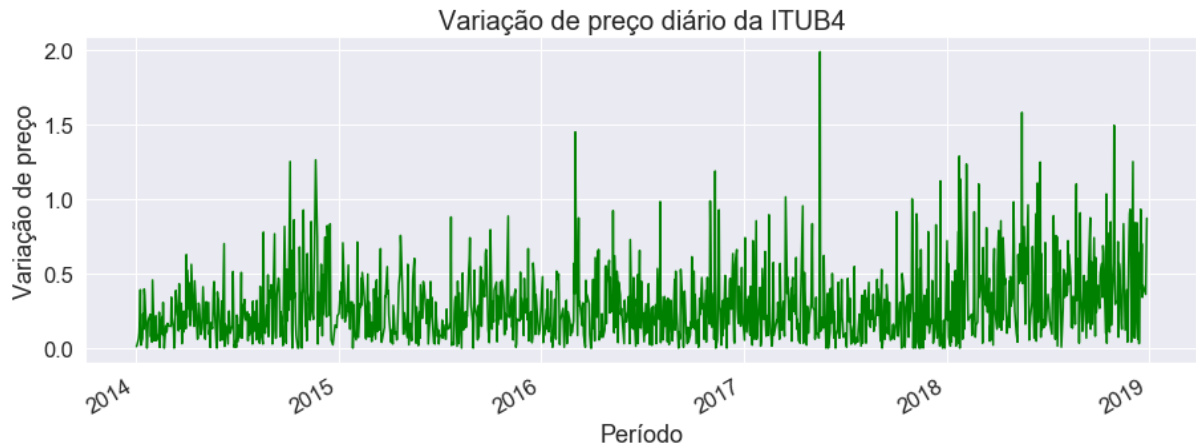


Figura 11: Variação de preço diário (preço de fechamento - preço de abertura) da ITUB4

E utilizando a função *argmax()* do Python, verifica-se que a maior variação de preço ocorreu em:

```
Date
2017-05-18    1.9867
dtype: float64
```

Verificando no Google Notícias o que aconteceu nesse dia, tem-se as notícias conforme mostra a figura 12.



Figura 12: Notícias da ITUB4 encontradas no Google no dia 18/05/2017

Analisando os dados em conjunto com as notícias, verifica-se que o índice Bovespa teve forte queda generalizada devido à notícia que os donos da JBS fizeram uma delação que envolvia o ex-presidente Michel Temer.

Plotando-se o gráfico do volume negociado, obtém-se a figura 13.

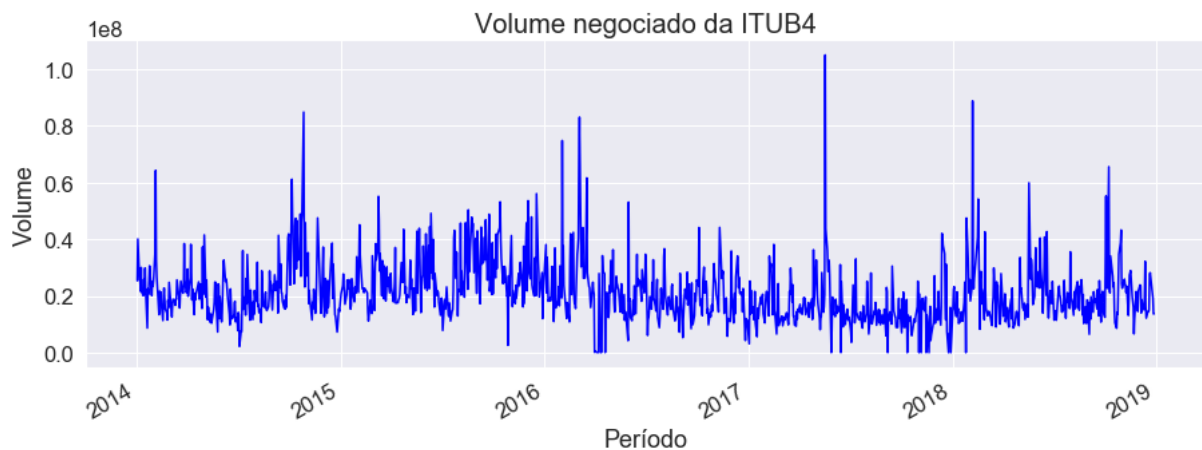


Figura 13: Volume negociado da ITUB4

Utilizando-se a função *argmax()* do Python, verifica-se que o dia de maior negociação foi em 18/05/2017:

```
High      2.396000e+01
Low       2.113330e+01
Open      2.113330e+01
Close     2.312000e+01
Volume    1.049172e+08
Adj Close  2.099524e+01
Name: 2017-05-18 00:00:00, dtype: float64
```

Para essa data, verifica-se pelo Google as notícias apresentadas na figura 12 e o motivo dessa variação conforme explicado anteriormente.

4.3. Análise Exploratória da VALE3

Após a coleta dos dados, foi realizada uma descrição estatística do dataset conforme mostra a figura 14.

	High	Low	Open	Close	Volume	Adj Close
count	1242.000000	1242.000000	1242.000000	1242.000000	1.242000e+03	1242.000000
mean	29.784316	28.847955	29.319589	29.288841	9.732985e+06	26.385362
std	12.432266	12.238706	12.350820	12.349072	9.162519e+06	12.331075
min	9.060000	8.600000	8.750000	8.600000	0.000000e+00	7.664479
25%	19.300001	18.532501	18.910000	18.902500	4.686425e+06	16.531945
50%	28.955000	28.054999	28.490000	28.570001	6.900500e+06	24.171745
75%	34.437499	33.520000	34.000000	33.934999	1.305725e+07	30.516664
max	62.419998	60.480000	62.200001	62.200001	1.744043e+08	60.594360

Figura 14: Descrição estatística da VALE3

Em seguida foi verificado que não há valor nulo no dataset:

```
High      0
Low       0
Open      0
Close     0
Volume    0
Adj Close 0
dtype: int64
```

Plotando-se um histograma do preço de fechamento, obtém-se o resultado apresentado na figura 15.

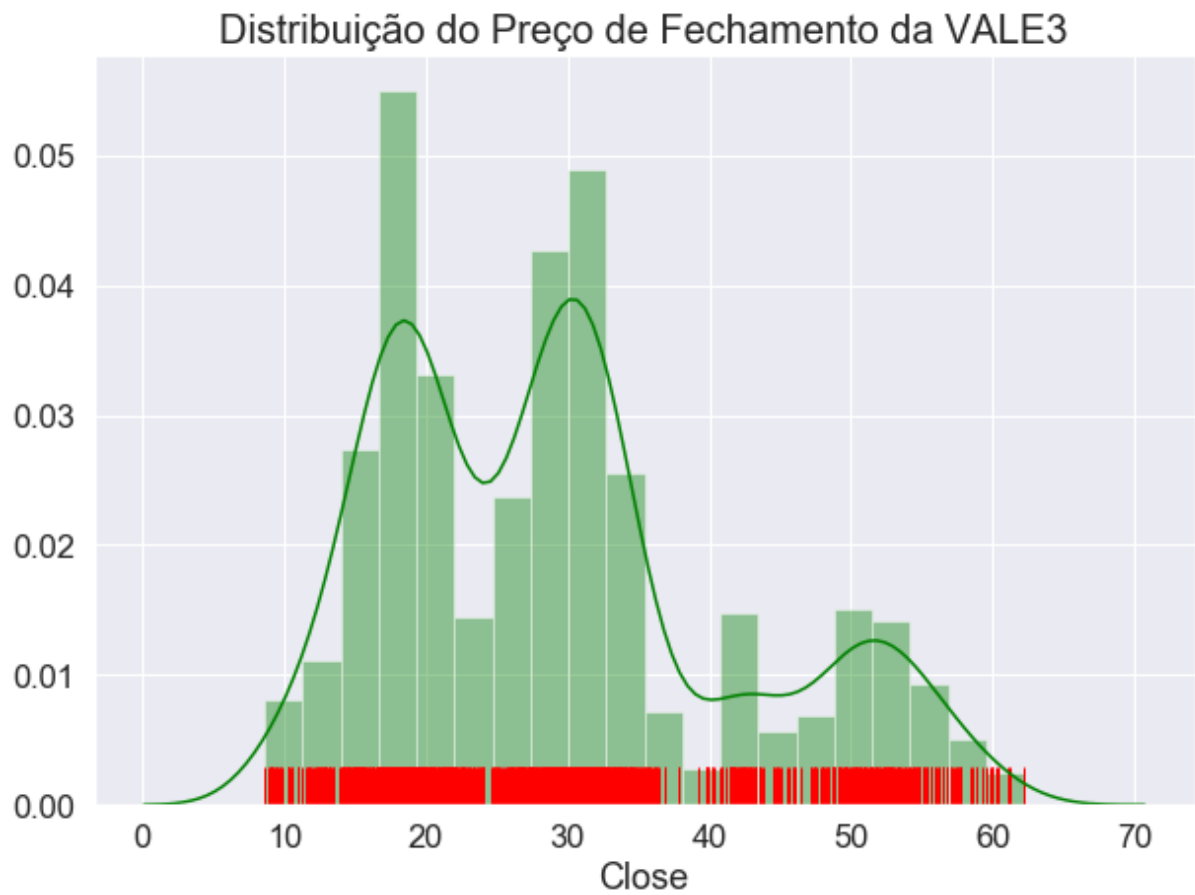


Figura 15: Histograma do preço de fechamento da VALE3

Percebe-se que o histograma apresenta três picos principais em torno dos preços de fechamento R\$ 19,00, R\$ 30,00 e R\$ 52,00.

Também foi plotado o histórico de preços englobando o preço máximo, mínimo e fechamento, conforme mostra a figura 16.



Figura 16: Histórico de Preços da VALE3

Pela figura percebe-se que os três preços estão fortemente correlacionados no longo prazo.

Plotando-se a variação de preço diário (preço de fechamento - preço de abertura), obtém-se o resultado apresentado na figura 17.

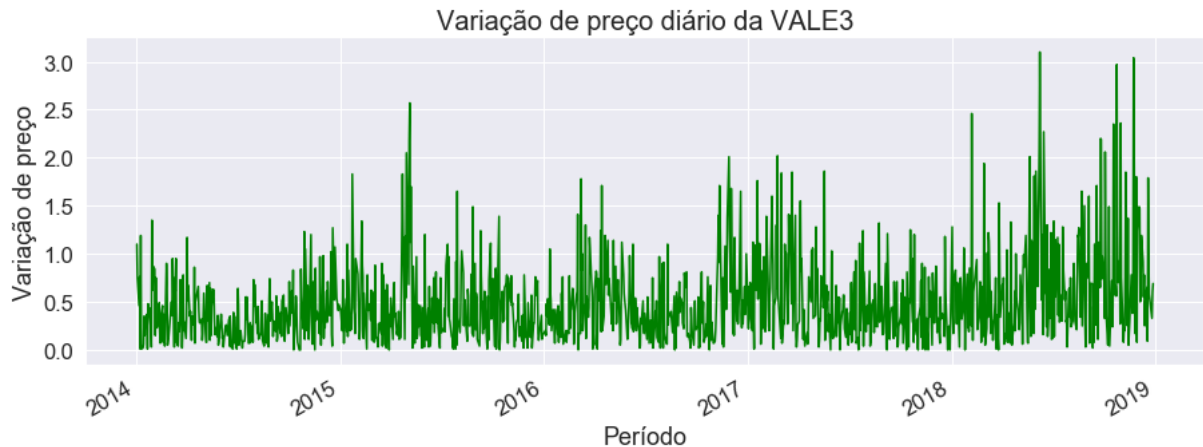


Figura 17: Variação de preço diário (preço de fechamento - preço de abertura) da VALE3

E utilizando a função *argmax()* do Python, verifica-se que a maior variação de preço ocorreu em:

```
Date
2018-06-08    3.100002
dtype: float64
```

Verificando no Google Notícias o que aconteceu nesse dia, tem-se a figura 18.

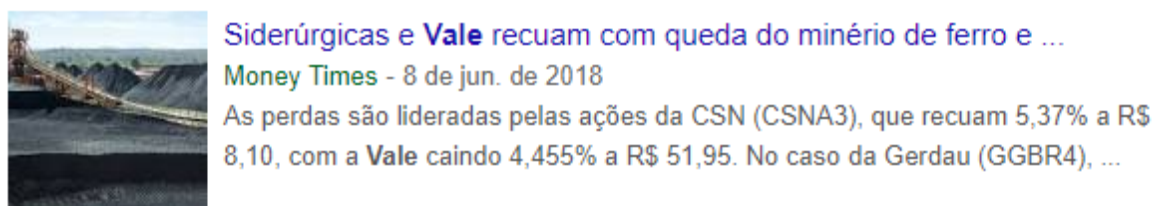


Figura 18: Notícia da VALE3 encontradas no Google no dia 08/06/2018

Pela notícia, verifica-se que a responsável pela variação de preço foi a queda do minério de ferro e a fala do presidente do Banco Central reforçando o aumento de liquidez ao mercado enquanto necessário.

Plotando-se o gráfico do volume negociado, obtém-se a figura 19.

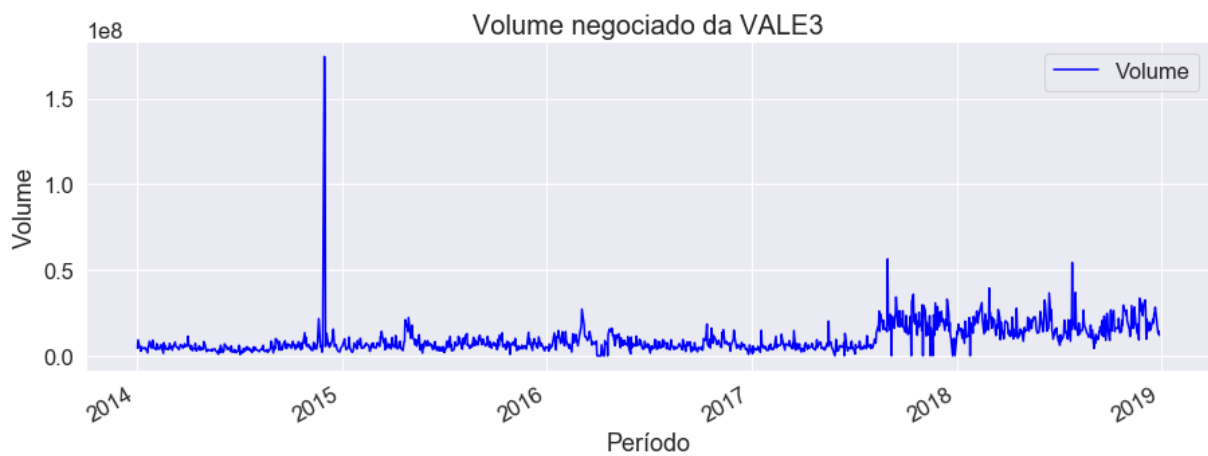


Figura 19: Volume negociado da VALE3

Utilizando-se a função *argmax()* do Python, verifica-se que o dia de maior negociação foi em 01/12/2014:

```
High      2.294000e+01
Low       2.205000e+01
Open      2.281000e+01
Close     2.213000e+01
Volume    1.744043e+08
Adj Close 1.871372e+01
Name: 2014-12-01 00:00:00, dtype: float64
```

Para essa data, verifica-se pelo Google as notícias apresentadas na figura 7 e o motivo dessa variação explicado anteriormente.

4.4. Análise das ações em conjunto

A figura 20 apresenta um gráfico comparativo de desempenho das três ações de forma conjunta. Para isso, foi utilizado o preço ajustado (Adj Close) já que este considera os ajustes de dividendos, splits, inplits, etc. Além disso, o preço ajustado foi normalizado para facilitar a comparação.



Figura 20: Comparativo de retorno das ações PETR4, ITUB4 e VALE3

Claramente pode-se perceber que a ação do Itaú ITUB4 foi a de maior retorno financeiro no período.

A figura 21 apresenta um gráfico comparativo do $\log(\text{retorno diário})$ considerando o valor ajustado das ações.

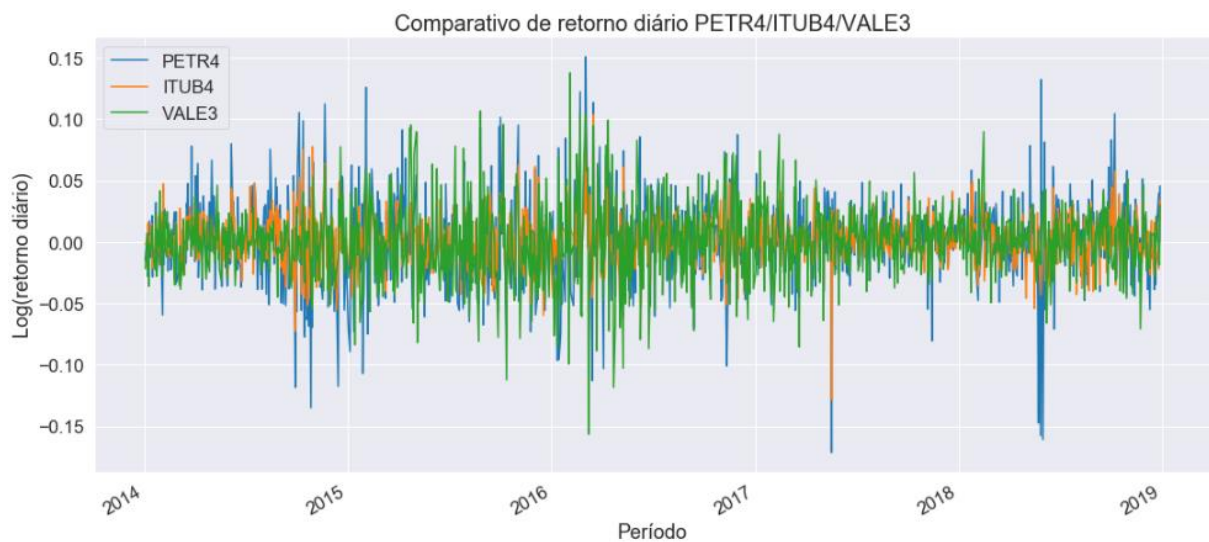


Figura 21: Comparativo de retorno diário da PETR4, ITUB4 e VALE3

Com o gráfico da figura 21, pode-se ter uma ideia dos períodos em que os ativos tiveram maiores variações, como exemplo no início de 2016.

A figura 22 mostra um mapa de correlação, com ele podemos verificar que o preço ajustado das ações está fortemente correlacionado entre si, o que também está de acordo com a figura 20.

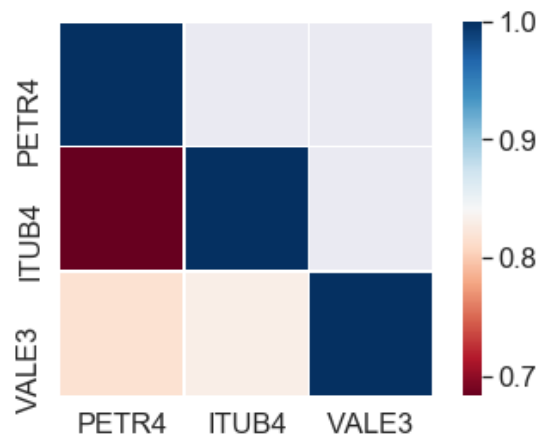


Figura 22: Mapa de correlação da PETR4, ITUB4 e VALE3

Por fim, para cada ativo, foi calculado o retorno e risco conforme mostra a tabela abaixo:

Retorno médio diário		% de retorno anualizado	Risco:		
PETR4	0.000330	PETR4	8.250485	PETR4	0.034134
ITUB4	0.000842	ITUB4	21.049850	ITUB4	0.019859
VALE3	0.000474	VALE3	11.847867	VALE3	0.030058
dtype: float64		dtype: float64		dtype: float64	

5. Modelos Preditivos para a PETR4

Esta seção apresenta os modelos preditivos desenvolvidos em linguagem Python para a ação PETR4 utilizando a biblioteca Facebook Prophet, AUTO-ARIMA e RNN (*Recurrent Neural Network*) utilizando a arquitetura LSTM (*Long Short Term Memory*).

5.1. Modelo Preditivo com Facebook Prophet

Para esse modelo, o dataset de treinamento foi reduzido em apenas duas colunas com o objetivo de atender a especificação da biblioteca. Abaixo o cabeçalho:

	ds	y
0	2014-01-02	16.750000
1	2014-01-03	16.500000
2	2014-01-06	16.620001
3	2014-01-07	16.160000
4	2014-01-08	16.190001

Onde ds representa a data da negociação e y o preço de fechamento do dia.

O código do modelo ficou da seguinte forma:

```
prophet_model = Prophet(changepoint_prior_scale=0.05, interval_width=0.95,
daily_seasonality=False)
```

Onde:

- **Changepoint_prior_scale:**
Parâmetro que modula a flexibilidade da seleção automática de ponto de mudança. Valores grandes permitirão muitos pontos de mudança, valores pequenos permitirão poucos pontos de mudança.
- **Interval_width:**
Largura dos intervalos de incerteza previstos para a previsão.
- **Daily_seasonality:** Ajusta a sazonalidade diária

5.2. Modelo Preditivo com AUTO-ARIMA

Para a utilização da biblioteca AUTO-ARIMA, foi realizado um breve estudo da série temporal, considerando aspectos tais como sazonalidade, estacionaridade e autocorrelação.

Decompondo-se a série temporal, obtém-se as componentes apresentadas na figura 23.

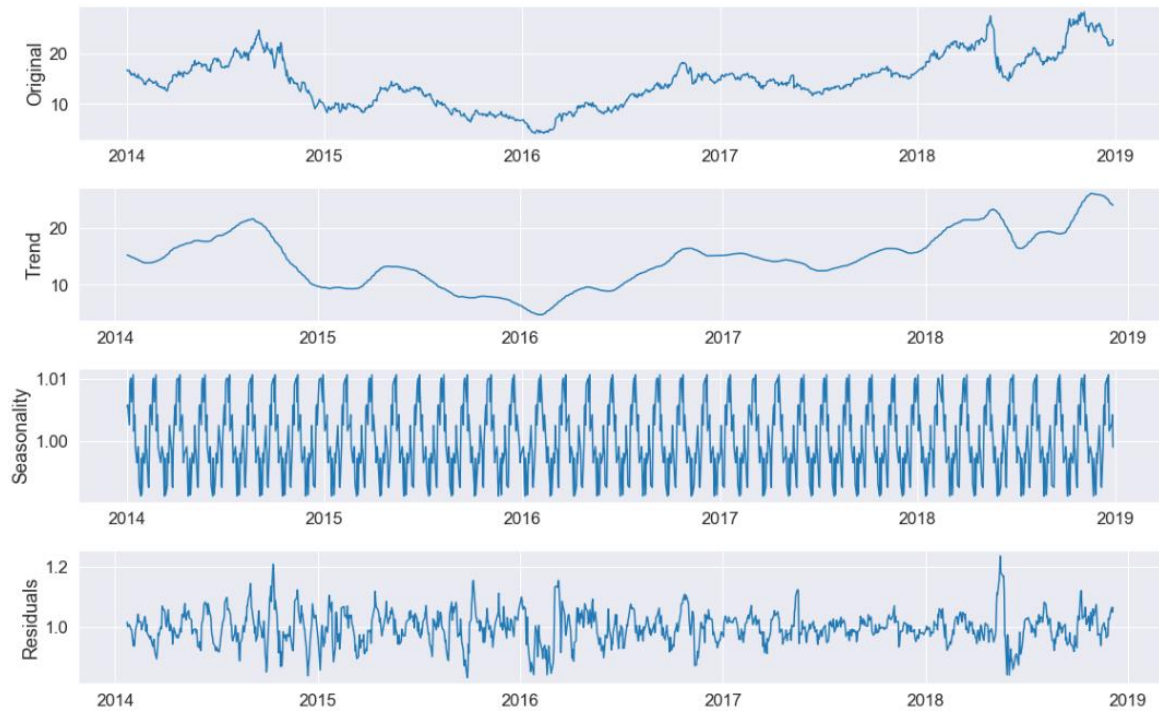


Figura 23: Decomposição da série temporal da PETR4

E plotando-se a Autocorrelação, temos a figura 24.

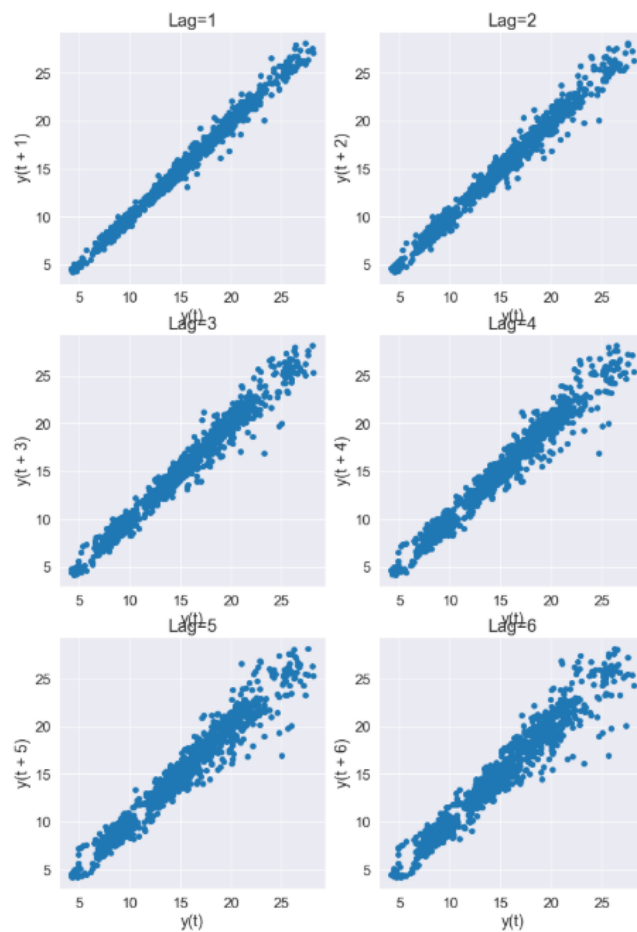


Figura 24: Autocorrelação da série temporal da PETR4

Como a plotagem da autocorrelação parece bem linear, um modelo auto-regressivo poderia ser uma boa escolha. Mas ainda assim foi utilizado o modelo AUTO-ARIMA para identificação do melhor modelo de forma automática.

A estacionalidade foi verificada utilizando o Teste Dickey-Fuller, obtendo-se o seguinte resultado:

```
Resultado do Teste Dickey-Fuller:
Teste                -1.508265
Valor p              0.529419
Nº de lags           10.000000
Nº de observações    1231.000000
Valor Crítico (1%)   -3.435673
Valor Crítico (5%)   -2.863891
Valor Crítico (10%)  -2.568022
dtype: float64
```

Conforme já era esperado pela análise dos gráficos anteriores e como o Valor p é muito maior que 0.05, conclui-se que a série não é estacionária. Para torná-la estacionária, poderia ser tentada a aplicação de uma função logaritma ou a diferenciação da série. Neste caso foi escolhida esta última.

Assim, aplicando-se uma diferenciação de primeira ordem, obtém-se o resultado da figura 25.

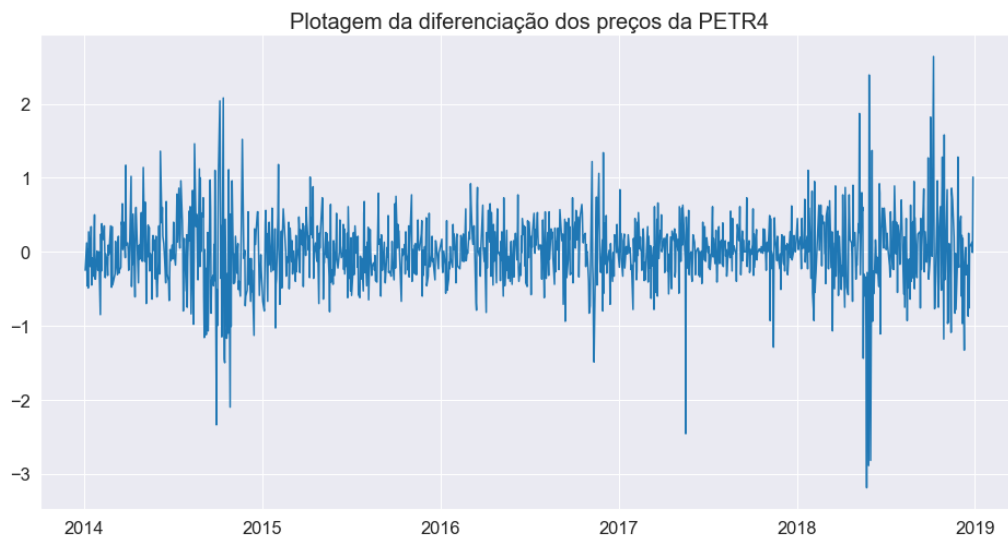


Figura 25: Plotagem da diferenciação de primeira ordem da série temporal

Aparentemente a série é estacionária, para confirmar aplica-se novamente o Teste Dickey-Fuller e obtém-se o seguinte resultado:

```

Resultado do Teste Dickey-Fuller:
Teste                -1.099360e+01
Valor p              6.960303e-20
Nº de lags           9.000000e+00
Nº de observações    1.231000e+03
Valor Crítico (1%)   -3.435673e+00
Valor Crítico (5%)   -2.863891e+00
Valor Crítico (10%)  -2.568022e+00
dtype: float64

```

O teste confirma que a série é estacionária, já que o Valor p é muito menor que 0.05.

Plotando-se os gráficos ACF (*Autocorrelation*) e PACF (*Partial Autocorrelation*), tem-se a figura 26.

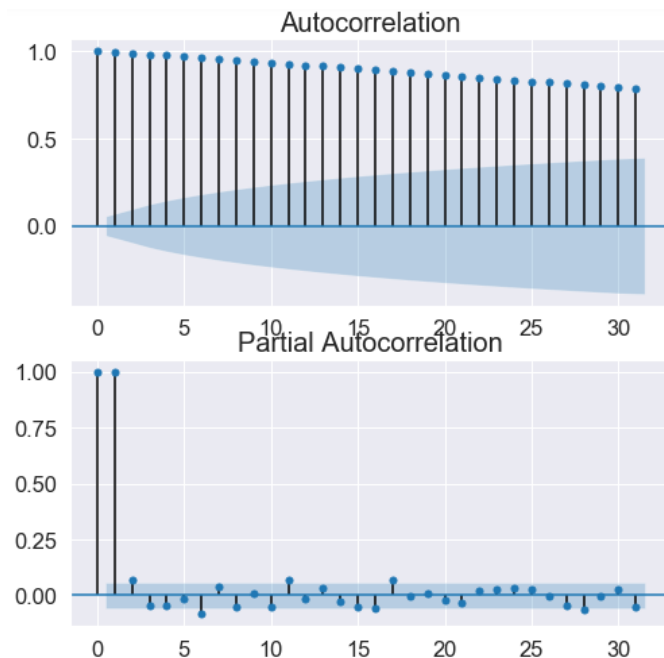


Figura 26: Gráficos ACF e PACF

Com estes gráficos poderiam ser obtidos os componentes p (número de time lags do modelo auto-regressivo) e q (ordem do modelo de média-móvel), caso não fosse utilizado a função AUTO-ARIMA.

Utilizando-se o fato da série ser estacionária para uma diferenciação de primeira ordem ($d=1$), pode-se utilizar a função *auto_arima* para determinar o melhor modelo que se adequa à série temporal estudada.


```

model = pm.auto_arima(train_petr4_arima['Close'], start_p=1, start_q=1,
                      #test='adf', # usa o adftest para encontrar o me-
lhor valor 'd'
                      max_p=3, max_q=3, # Máximo 'p' e 'q'
                      m=12,             # Frequência da série
                      d=1,
                      stationary = False, #Estacionaridade
                      seasonal=True, #Sazonalidade
                      start_P=0,
                      D=1,
                      trace=True, #Se TRUE, a lista de modelos ARIMA
considerados será reportada.
                      error_action='ignore',
                      suppress_warnings=True,
                      stepwise=True) #Se TRUE, fará a seleção gradual
(mais rápido). Caso contrário, ele pesquisará todos os modelos.

```

Os principais parâmetros são:

- p = número de time lags do modelo auto-regressivo (AR)
- q = ordem do modelo de média-móvel (MA)
- d = grau de diferenciação
- P = refere-se ao termo auto-regressivo para a parte sazonal
- Q = refere-se ao termo de diferenciação para a parte sazonal
- D = refere-se ao termo da média-móvel para a parte sazonal

O melhor modelo encontrado foi o SARIMAX:

```

=====
SARIMAX Results
=====
Dep. Variable:          y      No. Observations:      1242
Model:          SARIMAX(2, 1, 1)x(0, 1, 1, 12)      Log Likelihood      -893.491
Date:              Mon, 30 Mar 2020      AIC      1798.983
Time:              17:02:16      BIC      1829.666
Sample:              0      HQIC      1810.527
                    - 1242
Covariance Type:      opg
=====
              coef      std err          z      P>|z|      [0.025      0.975]
-----
intercept      0.0002      0.000      0.910      0.363      -0.000      0.000
ar.L1           0.5880      0.128      4.609      0.000      0.338      0.838
ar.L2           0.0845      0.019      4.506      0.000      0.048      0.121
ma.L1          -0.6329      0.128     -4.958      0.000     -0.883     -0.383
ma.S.L12       -0.9993      0.256     -3.909      0.000     -1.500     -0.498
sigma2          0.2397      0.059      4.042      0.000      0.123      0.356
=====
Ljung-Box (Q):              53.08      Jarque-Bera (JB):              1459.05
Prob(Q):                    0.08      Prob(JB):                    0.00
Heteroskedasticity (H):      1.23      Skew:                        -0.35
Prob(H) (two-sided):         0.03      Kurtosis:                     8.29
=====

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).

```

5.2. Modelo Preditivo com RNN (LSTM – Long Short Term Memory)

Para a criação e treinamento do modelo utilizando Redes Neurais Recorrentes – LSTM, o código foi desenvolvido seguindo as seguintes etapas:

1. Carregamento dos dados
2. Normalização dos dados
3. Definição da janela temporal a ser utilizada (60 timesteps)
4. *Reshaping* do dataset de treinamento
5. Criação do modelo
6. Compilação do modelo
7. Treinamento

Abaixo está o código da criação do modelo (Etapa 5):

```
# Inicialização da RNN
model_rnn = Sequential()

#O parâmetro return_sequences=True indica que a rede terá mais camadas a
frente
#O parâmetro Dropout ajuda no ajuste do Overfitting

# Adiciona a primeira camada LSTM com o Dropout
model_rnn.add(LSTM(units = 128, return_sequences = True, input_shape =
(x_train_lstm.shape[1], 1)))
model_rnn.add(Dropout(0.3))

# Adiciona a segunda camada LSTM com o Dropout
model_rnn.add(LSTM(units = 64))
model_rnn.add(Dropout(0.3))

# Adiciona a camada de saída
#model_rnn.add(Dense(16,init='uniform',activation='relu'))
model_rnn.add(Dense(1,init='uniform',activation='linear'))

# Compila a RNN, neste caso utilizando o otimizador 'Adam'
model_rnn.compile(optimizer = 'adam', loss = 'mean_squared_error', me-
trics=['mean_absolute_error'])
```

O código começa com a inicialização da RNN utilizando a função *Sequential()*. O modelo desenvolvido utiliza duas camadas LSTM com *Dropout* e uma camada densa de saída com ativação linear. O *Dropout* serve para reduzir o overfitting, pois ele “desativa” momentaneamente um percentual de neurônios artificiais, neste caso 30%.

A primeira camada ou camada de entrada, possui 128 neurônios do tipo LSTM, a segunda possui 64 e a última possui apenas uma saída linear. A escolha do número de neurônios foi escolhida de forma empírica e com base na literatura científica.

A compilação utiliza o otimizador *Adam* que apresentou bons resultados. Além disso, foi utilizado como função de perda o erro MSE (*Mean Squared Error*) e métrica o MAE (*Mean Absolute Error*).

A etapa de treinamento utilizou o código

```
model_rnn.fit(x_train_lstm, y_train_lstm, epochs = 100, batch_size = 32)
```

Para o treinamento foram utilizadas 100 épocas, este número também foi escolhido de forma empírica. O batch-size é o número de sequências que são treinadas juntas, ou seja, o número de amostras que serão propagadas através da rede

neural. O valor de 32 foi escolhido com base na literatura que sugere valores como 32, 64, 128, etc.

De um modo geral, o modelo desenvolvido é mostrado na figura 27.

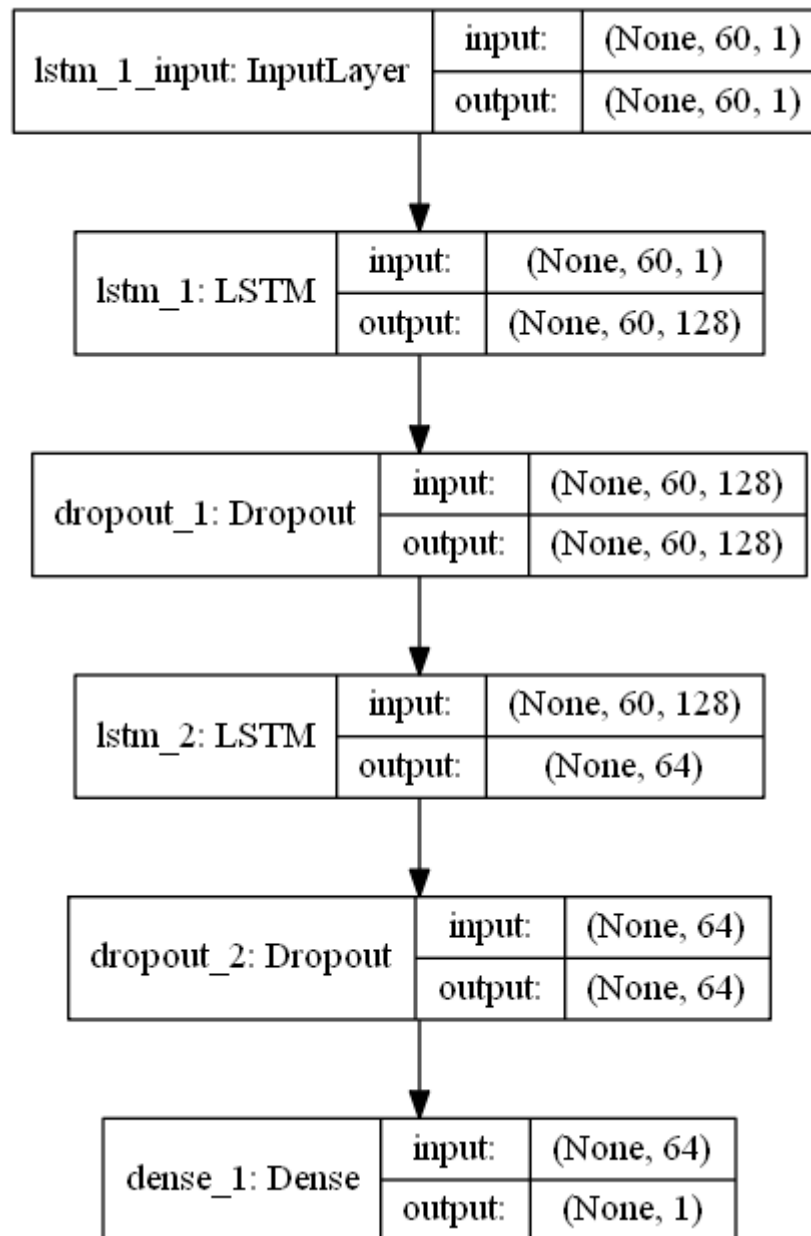


Figura 27: Modelo RNN desenvolvido

6. Apresentação dos Resultados

6.1. Modelo Preditivo com Facebook Prophet

A figura 28 mostra o resultado da previsão para o ano de 2019. Pode-se perceber que para o ano previsto, o preço de fechamento da PETR4 apresenta tendência de subida.

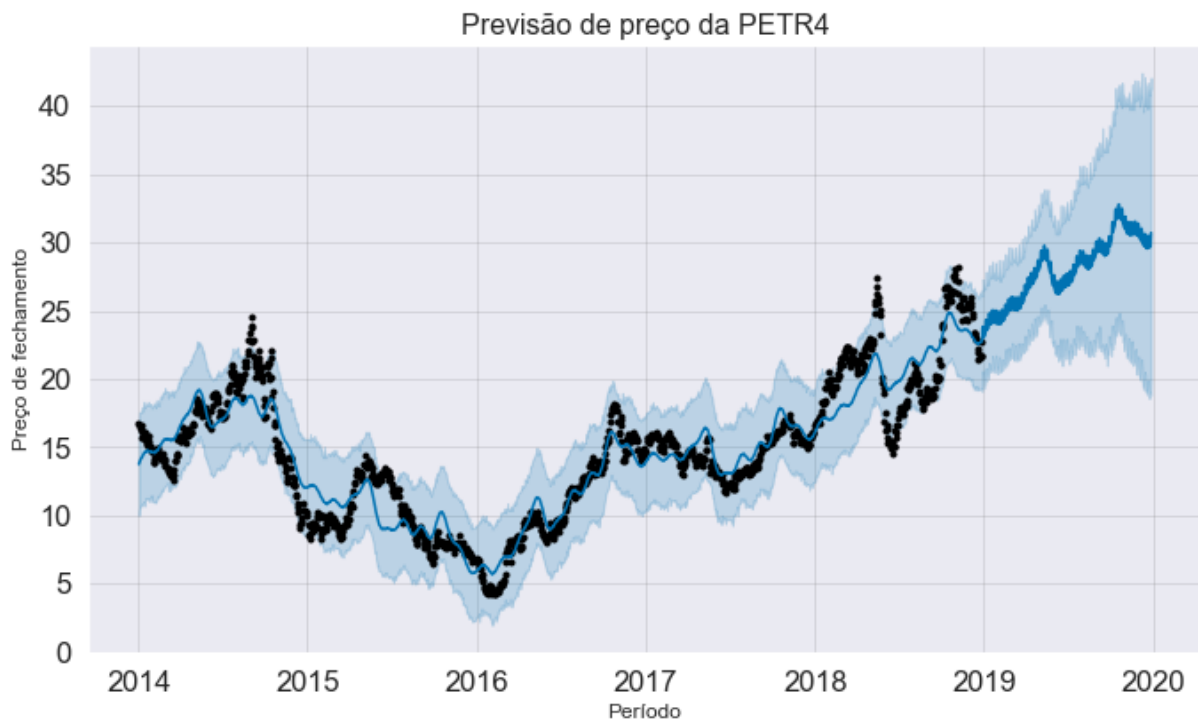


Figura 28: Resultado da previsão de preço da PETR4

A figura 29 apresenta as componentes decompostas da série temporal prevista. Conforme analisado na figura 28, pode-se confirmar que há uma tendência de subida do preço de fechamento.

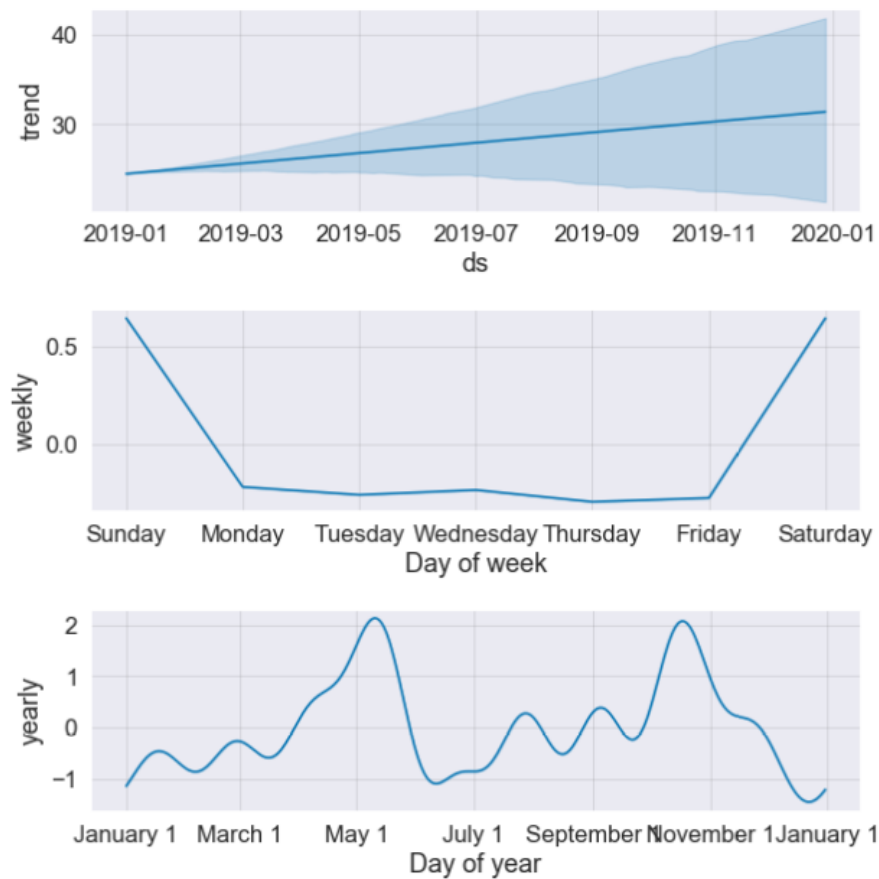


Figura 29: Componentes da previsão de preço da PETR4

A figura 30 mostra os dados de treinamento, teste, previsão e sua banda de variação feita pelo Prophet para a PETR4.



Figura 30: Dados de treinamento, teste e previsão plotados em conjunto

A previsão ficou dentro da banda de variação e acompanha a tendência dos dados de teste. A figura 31 mostra os dados de teste e dados previstos para melhor comparação.

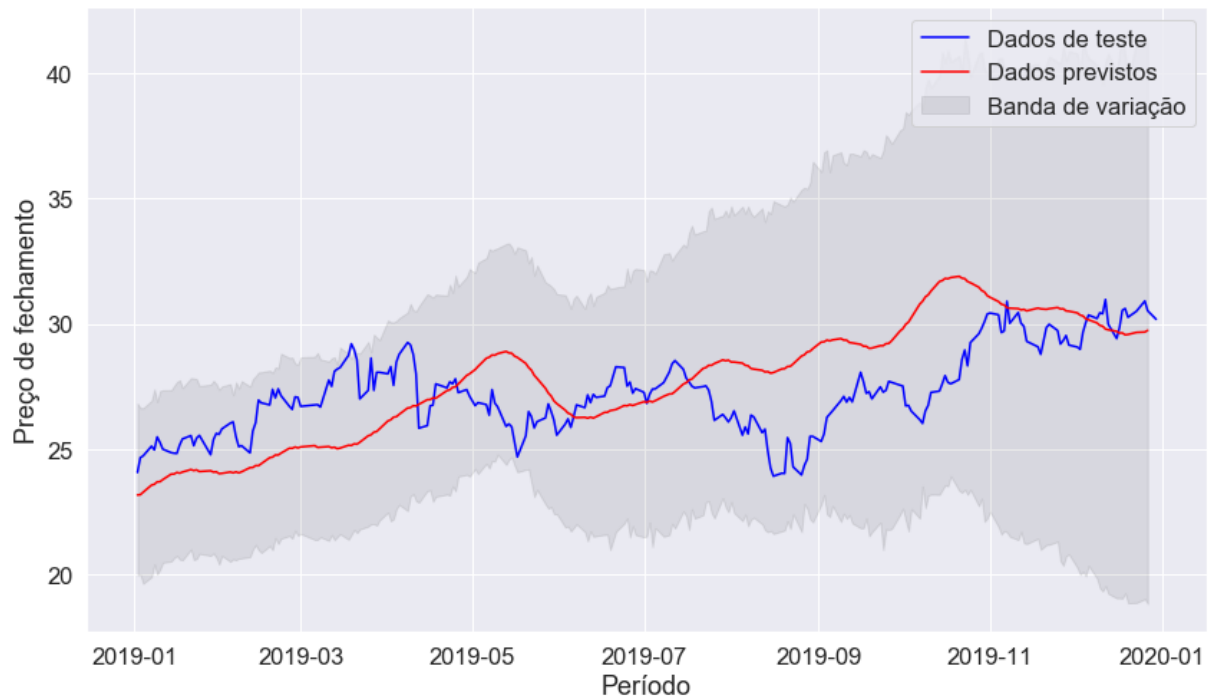


Figura 31: Dados de teste e dados previstos no ano de 2019

Erro obtido:

MAE: 1.7772801725844554
MSE: 4.483243380602945
RMSE: 2.1173670868800585

6.2. Modelo Preditivo com AUTO-ARIMA

A figura 32 mostra o resultado da predição para o ano de 2019. Pode-se perceber que para o ano previsto, o preço de fechamento da PETR4 apresenta tendência de subida.



Figura 32: Dados de treinamento, teste e previsão plotados em conjunto

A previsão ficou dentro da banda de variação e acompanha a tendência dos dados de teste. A figura 33 mostra os dados de teste e dados previstos para melhor comparação.

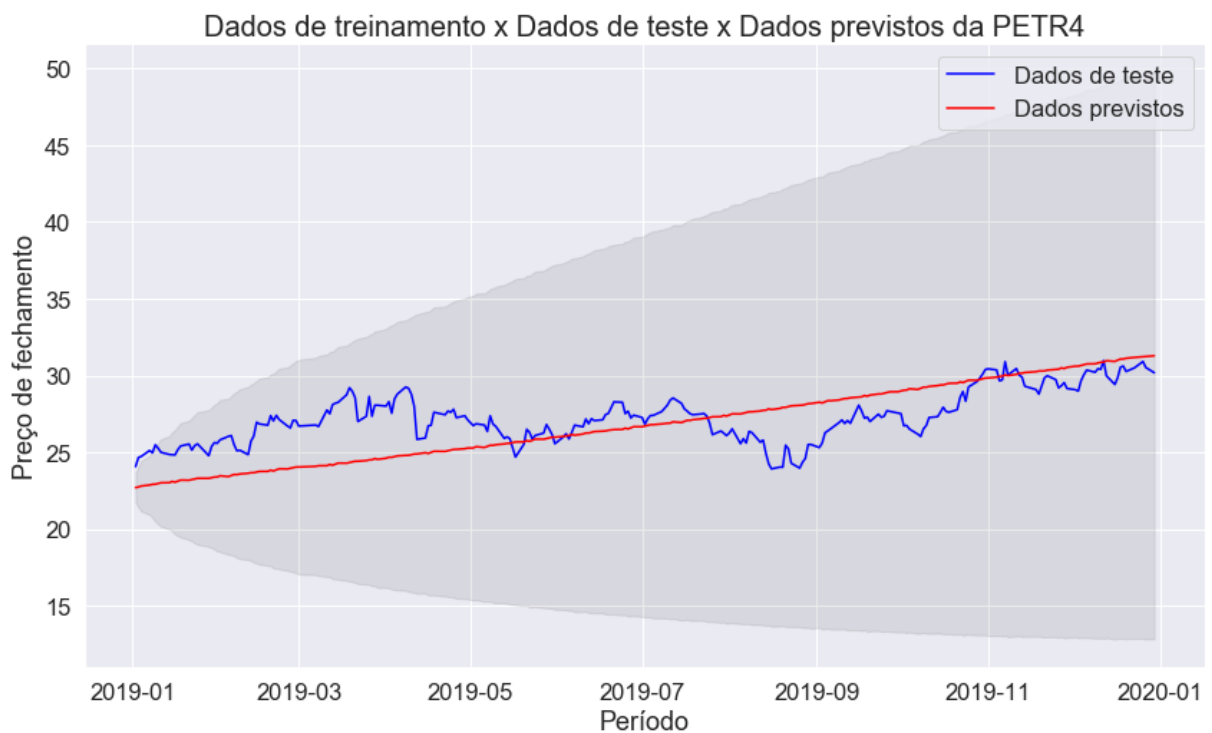


Figura 33: Dados de teste e dados previstos no ano de 2019

Erro obtido:

MSE: 4.227223941642828
 MAE: 1.7053041928041581
 RMSE: 2.05602138647506

Plotagem dos resíduos:

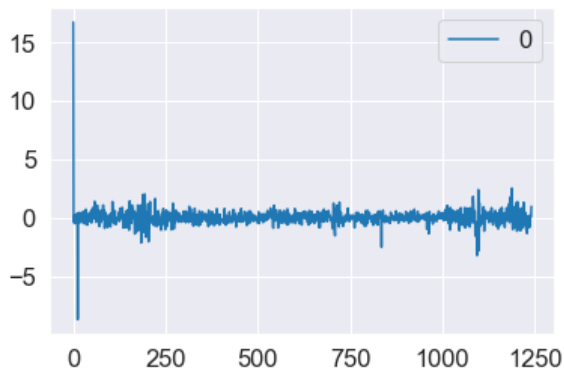


Figura 34: Plotagem dos resíduos

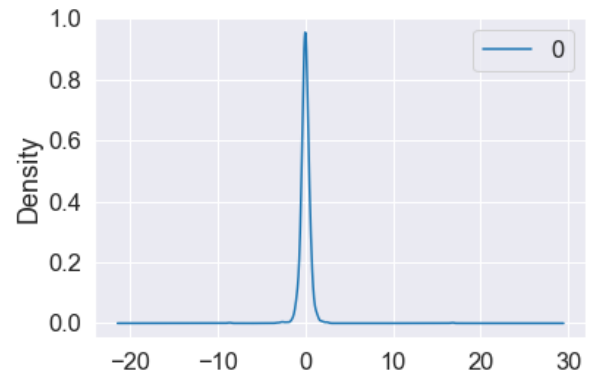


Figura 35: Densidade dos resíduos

6.2. Modelo Preditivo com RNN (LSTM – *Long Short Term Memory*)

A figura 36 mostra o resultado da predição para o ano de 2019. Pode-se perceber que os dados de teste se ajustam de forma muito aproximada com os dados previstos pela RNN.



Figura 36: Dados de treinamento, teste e previsão plotados em conjunto

Os dados previstos se ajustam muito bem nos dados de teste. A figura 37 mostra os dados de teste e dados previstos para melhor comparação.

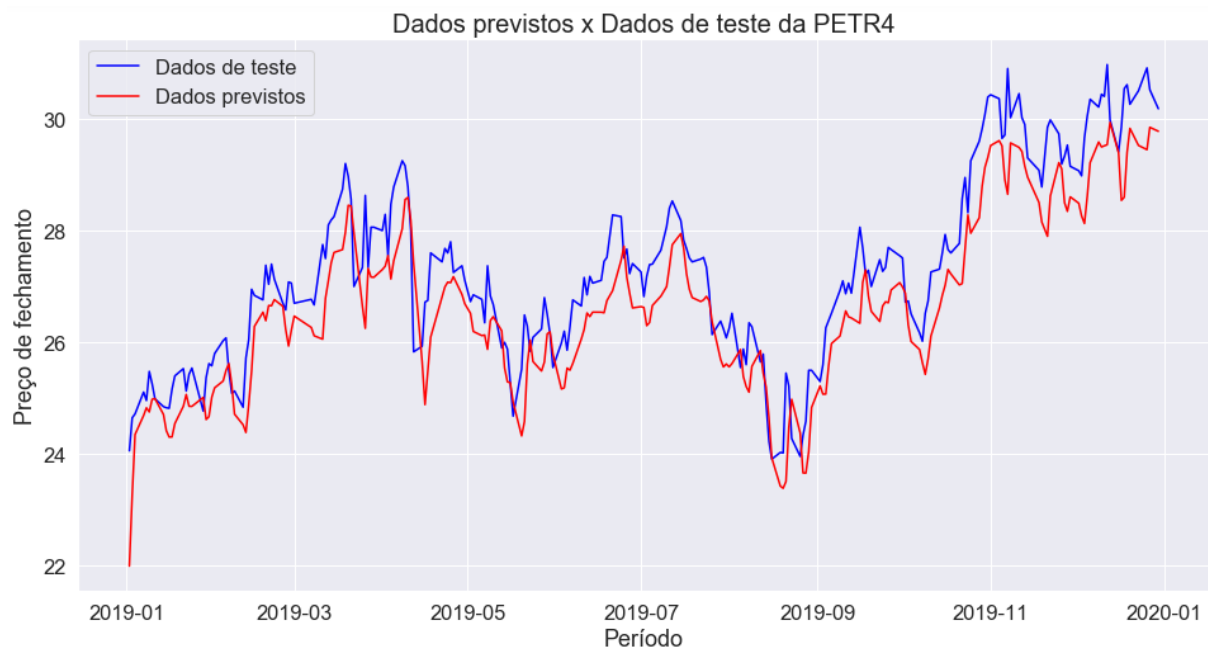


Figura 37: Dados de treinamento, teste e previsão plotados em conjunto

Erro obtido:

MAE: 0.7126203606485838
MSE: 0.7135903958605112
RMSE: 0.8447427986437713

7. Links

- Link para o repositório Github com o conteúdo do trabalho:
<https://github.com/aly2009/ModelosPreditivos-BolsaValores.git>
- Link para o vídeo com a apresentação resumida do trabalho no Youtube:
<https://youtu.be/p-rPxPkRZPg>

REFERÊNCIAS

Ghosh, A., Bose, S., Maji, G., Debnath, N. C., & Sen, S. **Stock Price Prediction Using LSTM on Indian Share Market**. Local: Proceedings of 32nd International Conference on (Vol. 63, pp. 101-110), 2019.

Castelão, R. **Utilização de Redes Neurais para Previsões no Mercado de Ações**. Campinas, SP: Universidade Estadual de Campinas, 2019.

Roondiwala, Murtaza & Patel, Harshal & Varma, Shraddha. **Predicting Stock Prices Using LSTM**. International Journal of Science and Research (IJSR), 2017.

Adebiyi, A. A., Adewumi, A. O., & Ayo, C. K. **Comparison of ARIMA and artificial neural networks models for stock price prediction**. Journal of Applied Mathematics, 2014.

FACEBOOK Prophet. Disponível em: <<https://opensource.facebook.com/>>. Acesso em: 28 jan. 2020.

PORTILLA, Jose Marcial. **Using Python and Auto ARIMA to Forecast Seasonal Time Series**. Disponível em: <<https://medium.com/@josemarcialportilla/using-python-and-auto-arima-to-forecast-seasonal-time-series-90877adff03c>>. Acesso em: 20 fev. 2020.

KERAS. Disponível em: <<https://keras.io/>>. Acesso em: 04 mar. 2020.