

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

Jonathan Alves de Lima

**Um Estudo Sobre o Índice Ibovespa e Suas Movimentações em Tempos de
Coronavírus**

Belo Horizonte
2020

Jonathan Alves de Lima

**Um Estudo Sobre o Índice Ibovespa e Suas Movimentações em Tempos de
Coronavírus**

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	7
3. Processamento/Tratamento de Dados	13
4. Análise e Exploração dos Dados	16
5. Criação de Modelos de Machine Learning de Regressão.....	29
5.1. Simple Linear Regression (Regressão Linear Simples)	30
5.2. Decision Tree Regression (Árvore de Regressão)	31
5.3. Support Vector Regression (Máquina/Regressão de Vetor Suporte)	32
6. Apresentação dos Resultados	33
6.1. Resultados da Regressão Linear Simples	33
6.2. Resultados da Árvore de Regressão	35
6.3. Resultados da Regressão de Vetor Suporte (SVR)	37
6.4. Comparação dos Modelos.....	40
6.5. Um desafio para os Modelos Lineares	41
6.6. Conclusões	48
7. Links.....	51
Referências	52

1. Introdução

1.1. Contextualização

O Índice IBOVESPA é o principal indicador de desempenho das ações negociadas na B3, a bolsa de valores brasileira. Foi construído em 1986 e funciona com uma carteira teórica de ações. Atualmente conta com cerca de 500 papéis distribuídos em diferentes percentuais.

Os dados e movimentações da bolsa são de suma importância para os investidores, e aqui no Brasil acaba servindo como parâmetro para avaliar se o mercado fechou em alta ou em baixa. Essas informações, no geral, ajudam na tomada de decisão por parte dos investidores e empresas especializadas no assunto.

Por se tratar de empresas de capital aberto e estar exposta ao mundo, os preços das ações acabam sendo afetados por diversos fatores externos, como Política, relações comerciais e internacionais, preço de moeda, forças da natureza e diversos possíveis eventos que venham ocorrer de forma micro ou macro no ambiente que estão inseridas. Como exemplo, nesse ano de 2020 foi decretado estado de Pandemia por conta do vírus Covid-19, conhecido como Coronavírus, que afetou fortemente as negociações em bolsa ao redor do mundo.

Com isso fica difícil prever como as ações vão oscilar no decorrer do tempo, mas ainda assim podemos tentar utilizar as tecnologias que se adaptam e evoluem com grande velocidade ao decorrer do tempo para tentar prever e estudar comportamento da bolsa de valores para a tomada de decisões.

O Objetivo desse estudo é observar como diferentes modelos de machine learning, de regressão se comportam na previsão de do índice IBOVESPA.

Para esse estudo foram utilizadas técnicas de processamento de dados e os modelos de Machine Learning: Simple Linear Regression, Support Vector Regression e Decision Tree Regression

1.2. O problema proposto

O estudo visa analisar e identificar os principais comportamentos do índice principal do Brasil, o IBOVESPA.

Para isso, o estudo propõe uma análise exploratória, levantar insights, entender oscilações, correlações, e utilizando modelos de previsão para comparar como eles se comportam com o índice. Além disso também contextualizar com os efeitos da pandemia, focada exclusivamente no Brasil, verificando se o aumento do número de casos no país afetou realmente o índice. Para facilitar o entendimento do problema foi a técnica do 5W, que consiste em responder as seguintes perguntas:

- Why? Por que esse problema é importante?
 - Esse problema é importante pois a bolsa de valores é extremamente relevante para a economia do país, e entender suas movimentações e oscilações é de suma importância, ainda mais se podemos aplicar isso com tecnologia e comparar com acontecimentos recentes, nesse caso, além de analisar os dados do índice, as comparações com outros papéis são feitas para verificar se eles realmente acompanham o índice ou não.

Atualmente, empresas e profissionais da área financeira já utilizam diversos sistemas e modelos de Machine Learning (com treinos diários) para tentar prever e negociar em curto prazo, conhecido como Day-Trade. Além de ter outras empresas que já utilizam robôs de investimentos, tudo isso utilizando tecnologias de machine learning.

- (Who?) De quem são os dados analisados?
 - Os dados analisados são do Yahoo Finance para análise do IBOV (índice Ibovespa) e alguns outros papéis para comparação Itaú Unibanco e B3 (ITUB4 e B3SA3 respectivamente), além de dados Kaggle para análise dos dados de Covid-19.

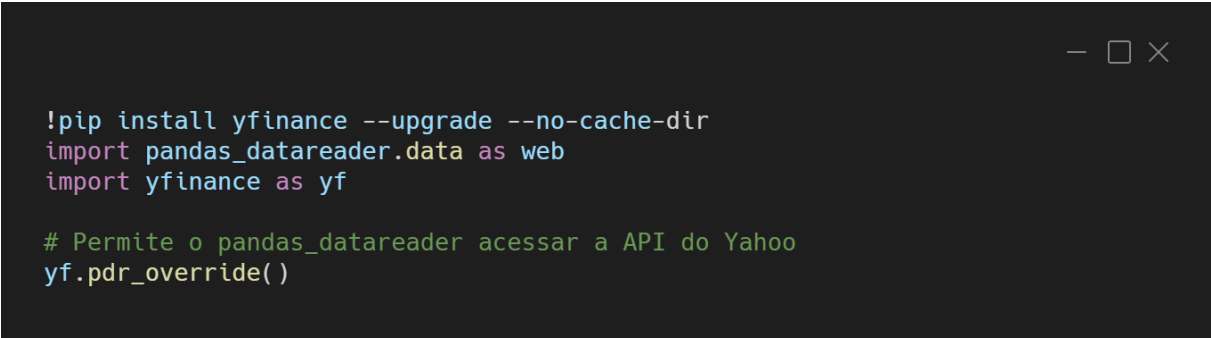
- (What?): Quais os objetivos com essa análise?
 - O objetivo dessa análise é entender como que o índice tem se comportado ao longo dos anos, se outros papeis tem relação com ele e entender se realmente os casos de Covid-19 no Brasil realmente afetaram o índice ou não. Além desses objetivos, esse estudo visa entender como diferentes modelos de regressão se comportam na predição do índice Ibovespa.
- (Where?): De onde são os dados?
 - Todos os dados analisados são nacionais. Os dados do Yahoo Finance são de uma API do Yahoo, que não é nacional, mas os dados analisados são do índice e de papéis pertencentes ao Brasil.
- (When?): Qual o período está sendo analisado?
 - Para os dados relacionados ao IBOV e ações no geral, foi analisado um período grande, de jan/2016 a jun/2020. Para os dados de Covid-19 foram selecionados dados de jan/2020 ao maio/2020.

2. Coleta de Dados

O Trabalho foi desenvolvido no Google Colab, onde a coleta de dados foi separada em três partes: a primeira e a segunda com dados do Yahoo Finance para dados da bolsa de valores e a terceira com dados do Kaggle, referente ao número de casos de corona vírus no Brasil.

O primeiro e o segundo dataset, foram extraídos da API do Yahoo Finance. Ela foi utilizada para download dos dados relacionados ao estudo do Índice e dos demais papeis (ITUB4 e B3SA3). Para utilizar a biblioteca do Yahoo Finance é necessário instalar ela. Como estava utilizando o Google Colab fiz a instalação diretamente no arquivo.

Após instalar a biblioteca foi necessário importar ela, o Pandas Data reader e utilizar um comando para permitir que a leitura fosse feita pela Pandas Data Reader.



```
!pip install yfinance --upgrade --no-cache-dir
import pandas_datareader.data as web
import yfinance as yf

# Permite o pandas_datareader acessar a API do Yahoo
yf.pdr_override()
```

E, por último, foi utilizado datareader para trazer os dados. Esses dois datasets foram separados em duas partes: Uma para estudo e análise diante do cenário atual, de pandemia, ou seja, verificar se há correlação com o Corona Vírus e a outra parte foi para a aplicação de diferentes modelos de Regressão, de Machine Learning, nesse caso fazendo um estudo apenas com o Índice IBOVESPA.

Para a primeira extração (versus Corona Vírus) foi utilizado um dataset com três papéis diferentes, com os campos o fechamento (close) e a data (date), onde foram comparados o aumento do número de casos da doença versus o fechamento diário desses papéis. No caso os papéis são: O Índice IBOVESPA, ITUB4 (Itaú) e B3SA3 (Brasil Bolsa Balcão - B3). Já para a segunda extração, onde foram aplicados os modelos de Machine Learning, foi utilizado apenas o índice IBOVESPA,

dessa vez, em sua completude, ou seja, todos os dados: abertura, alta, baixa, fechamento etc.

Abaixo a extração dos dados para o caso 1 (versus o Corona Vírus) e em seguida como foi feita a extração para o caso 2, para a aplicação do modelo de Machine Learning no Índice IBOVESPA.

```
start_date = "2015-01-01"
end_date = "2020-06-01"

df_acoes = pd.DataFrame()
tickers = ['ITUB4.SA', 'B3SA3.SA', '^BVSP']
for ticker in tickers:
    df_acoes[ticker] = web.get_data_yahoo(ticker, start = start_date, end =
end_date)['Close']
```

Para esse caso, foi feito um laço de repetição para trazer os dados de fechamento dos três papéis diferentes (ITUB4, B3SA3 e IBOV) e armazenar no dataframe.

A extração apenas do índice IBOVESPA é executada conforme o script abaixo.

```
df_ibov = web.get_data_yahoo('^BVSP', start=start_date, end=end_date)
```

Os dados são descritos conforme abaixo:

Nome da coluna/campo	Descrição	Tipo
Date	Data da Negociação (Pregão)	Date
High	Preço mais alto da ação no dia	Float
Low	Preço mais baixo da ação no dia	Float
Open	Preço de abertura da ação no dia	Float
Close	Preço de fechamento da ação no dia	Float
Volume	Quantidade de ações negociadas no dia	Int
Adj. Close	Preço de fechamento da ação no dia	Float

Quando falamos de ações, os principais dados relacionados as negociação, além dos indicadores específicos das empresas, são os dados das movimentações relacionadas a esses papéis, nesse caso é o objeto desse estudo.

- Date (Data) é o dia da negociação. Aqui no Brasil o pregão funciona somente em dias úteis, das 10:00 as 18:00. Então são esses os dias que são abordados e estão disponíveis no data-set.
- Open (Abertura) é o preço que o papel iniciou suas negociações no dia em questão. O preço de abertura também é o preço de fechamento do dia anterior.
- High (Máxima) é o preço máximo que um papel chegou em determinado dia, não necessariamente vai ser o preço de fechamento. Exemplo com uma ação fictícia: Imaginemos que o papel PUCM3 abriu o dia custando R\$ 50,00 e em determinado as negociações chegaram ao preço máximo de R\$ 55,00 e não subiu mais que isso. Esse foi o preço máximo do papel no dia.
- Low (Mínima): é o preço mínimo que um papel chegou em determinado dia, não necessariamente vai ser o preço de fechamento. Continuando com o exemplo de PUCM3: em determinado as negociações chegaram ao preço de R\$ 49,00, devido a alguma oscilação, e não caiu mais que isso. Esse foi o preço mínimo do papel no dia.
- Close (Fechamento) é de fato o preço do papel no final do dia. Continuando com o exemplo de PUCM3, no final do dia o papel fechou a R\$ 52,50. É nesse momento que temos o percentual final da ação no dia, nesse caso o papel subiu 5%. Este dado será o principal foco desse estudo.
- Volume é quantidade de papeis negociados no dia.
- Adj. Close (Fechamento ajustado) é o preço de fechamento após os ajustes para todos os desdobramentos e distribuições de dividendos aplicáveis. Nesse estudo esse dado não será analisado, o estudo será com foco no fechamento do dia.

Geralmente os esses dados, quando analisados em conjunto, são apresentados em formato de candlestick, o único que não fica nesse formato é o Adj. Close (fechamento ajustado).

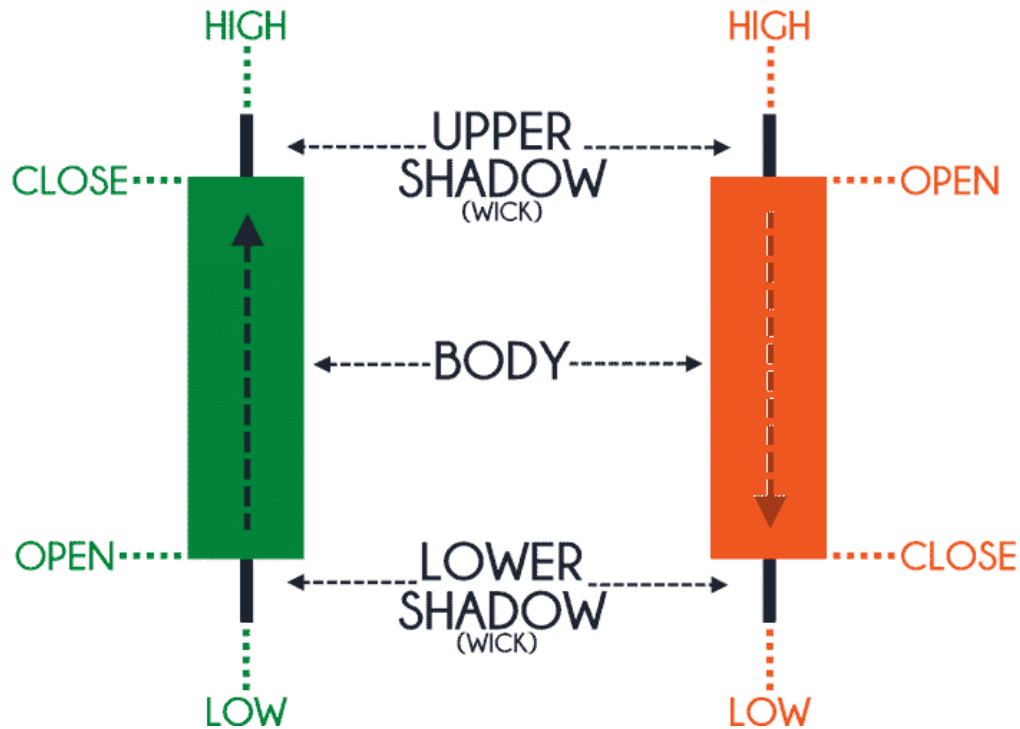


Figura 1: Estrutura do Candlestick (Fonte: analyzingalpha.com)

Quando analisamos um item apenas, utilizamos o gráfico de linha, no caso desse estudo o foco será no fechamento (Close) para a aplicação dos modelos de machine Learning e também para verificação diante do contexto atual da pandemia do Corona Vírus.



Para esse estudo será utilizado principalmente esse tipo de gráfico, pois o foco do mesmo será o fechamento.

A segunda etapa na extração dos dados, foi utilizar dados do Kaggle sobre Covid-19. Esses dados serão utilizados para verificar se há alguma correlação entre o aumento do número de casos da epidemia com os números apresentados na bolsa de valores.

Esse dataset contém apenas casos confirmados da doença pelo governo brasileiro e pelo ministério da saúde, o dataset em questão foi hospedado no Kaggle, mas a fonte dele é o próprio ministério da saúde.

A base apresenta dados diários agrupados pelo país como um todo e por cada uma das cidades. Para esse estudo os dados utilizados foram apenas os dados diários agrupados do Brasil (que é a soma de todas as cidades, já disponível dessa forma no dataset), nesse caso, olhando o crescente número de casos confirmados.

O dataset mencionado contém os campos abaixo:

Nome da coluna/campo	Descrição	Tipo
regiao	Nome da região	String
estado	UF do Estado	String
municipio	Nome do município	Float
coduf	Código do UF	Int
codmun	Código do município	Int
codRegiaoSaude	Código da região saúde	Int
nomeRegiaoSaude	Nome da região saúde	String
data	Data de referência	Date
semanaEpi	Semana epidemiológica	Int
populacaoTCU2019	População total da região	Int
casosAcumulado	Total de casos reportados da doença	Int
casosNovos	Total de novos casos da doença	Int
obitosAcumulado	Total de óbitos causados pela doença	Int
obitosNovos	Total de novos óbitos pela doença	Int
Recuperadosnovos	Total de recuperados da doença	Int
emAcompanhamentoNovos	Total de casos da doença que estão em acompanhamento	Int

Para o estudo foram utilizados os seguintes dados:

- data (26/02/2020) até (01/06/2020)
- região (filtrado por Brasil)

- casosAcumulados

No google colab, onde o projeto foi executado, a importação foi feita conforme abaixo:

Importar o dataset do Kaggle, o qual foi feito download. Para isso foi necessário utilizar uma importação do google, selecionar o arquivo (fazendo upload para o Google Drive) e preparando ele para o Pandas fazer a leitura.

```
from google.colab import files
import io
covid_file = files.upload()
covid_file = io.BytesIO(covid_file['covid_brasil.xlsx'])
df_covid_file = pd.read_excel(covid_file)
```

Dessa forma os dados estão preparados para serem trabalhados e posteriormente analisados e treinados com os modelos de Machine Learning.

3. Processamento/Tratamento de Dados

O primeiro e o segundo dataset, sobre a bolsa de valores, mesmo separado em duas partes, a extração dos dados, tratamento e exploração, foram praticamente iguais, justamente por se tratar de dados semelhantes e da mesma fonte. Nessa sessão o dataset relacionado ao estudo da correlação com a Pandemia será chamado de primeira base de dados e o dataset para a aplicação dos modelos de machine learning será chamado de segunda base de dados.

A primeira base de dados, não possui dados duplicados e contém um total de 1.334 registros, com dados de 01/01/2015 até 01/06/2020, esse longo período é para verificar uma evolução e movimentação ao longo do tempo. O dataset da primeira base de dados, o índice IBOVESPA foi normalizado com a mesma escala dos demais papeis, pois ele é apresentado em X mil pontos, então ele foi dividido por 1000, ficando na mesma escala dos papéis, que são tratados a partir de unidade, crescendo para dezena e/ou centena.

As colunas foram renomeadas para melhor identificação ao longo do estudo no Google Colab.

```
df_acoes.rename(columns = {'ITUB4.SA': 'ITUB4', 'B3SA3.SA': 'B3SA3',  
                           '^BVSP': 'IBOV'}, inplace=True)
```

Também foi feita a padronização do index após isso:

```
df_acoes.reset_index(inplace = True)
```

Foram identificados 8 registros em branco, no dataset que contém mais de uma ação, ou seja, o fechamento do IBOVESPA, ITUB4 e B3. Para isso foi feita a remoção dos dias correspondentes:

```
print(df_acoes.isna().sum())  
df_acoes.dropna(inplace=True)
```

Para a segunda base de dados os tratamentos foram bem semelhantes, mudando apenas que não houve necessidade de renomear os campos, pois já vieram com os nomes necessários para o estudo (Open, Close etc.)

Após isso, temos o segundo dataset, sobre a pandemia propriamente dita, esse dataset contém 142.800 registros e um total de 16 campos, essa grande quantidade é pela segmentação do por região, estado e município. Para esse estudo em específico, o dado que interessa é o da Região, pois existe um dado já agrupado pelo país inteiro, nesse caso a região é igual a “Brasil”. Além desses campos, também precisamos da Data e do total de casos acumulados.

Após fazer a leitura com o Pandas foi feita a seleção dos campos mencionados: data, região e casos acumulados e adicionado a um dataset:

```
df_covid = df_covid_file.  
    filter(['data', 'regiao', 'casosAcumulado'], axis=1)
```

Por fim, foi feito o filtro da Região, selecionando apenas Brasil e a remoção dessa coluna, deixando apenas a data e total de casos do Brasil.

```
df_covid = df_covid.loc[df_covid['regiao'] == 'Brasil']  
df_covid = df_covid.drop('regiao', 1)
```

Também foi feita a padronização do index e dado novos nomes para as colunas:

```
df_covid.reset_index(inplace=True)  
df_covid.rename(columns = {'data': 'Date', 'casosAcumulado': 'Confirmed  
Cases'}, inplace=True)
```

Por fim, para juntar os dois datasets (primeira base de dados, selecionada para estudar junto aos dados da pandemia). Para isso foi feito um merge,

semelhante ao *left-join*, em SQL, utilizando a data como chave. Criando assim uma única fonte de dados para analisar.

```
df_merge = df_acoes.merge(df_covid, on='Date', how='left')
```

Por conta desse merge, os dados que não têm uma data correspondente acabaram por ficarem como NaN no campo Confirmed Cases (ex. 01/01/2020, data onde não havia casos do Corona Vírus no Brasil). Para isso, foi utilizado o a função `fillna()` para preencher com zeros esses casos.

```
df_merge['Confirmed Cases'] = df_merge['Confirmed Cases'].fillna(0)
print(df_merge)
```

Gerando assim um dataset como esse abaixo, com a Data, o fechamento de ITUB4, B3, Índice IBOVESPA e Total de Casos confirmado do Corona Vírus.

	Date	ITUB4	B3SA3	IBOV	Confirmed Cases
0	2015-01-02	18.639099	9.510000	48.512	0.0
1	2015-01-05	18.732800	9.250000	47.517	0.0
2	2015-01-06	19.035801	9.340000	48.001	0.0
3	2015-01-07	19.724501	9.710000	49.463	0.0
4	2015-01-08	20.033100	9.580000	49.943	0.0
...
1329	2020-05-25	23.959999	46.500000	85.663	374898.0
1330	2020-05-26	23.010000	46.500000	85.469	391222.0
1331	2020-05-27	23.730000	46.590000	87.946	411821.0
1332	2020-05-28	23.309999	45.950001	86.949	438238.0
1333	2020-05-29	23.040001	45.549999	87.403	465166.0

[1334 rows x 5 columns]

Dessa forma, agora os dados estão preparados para uma exploração, e nesse caso, verificar se houve alguma correlação entre o aumento de casos de Corona Vírus no Brasil com os números da Bolsa de Valores.

4. Análise e Exploração dos Dados

A exploração dos dados foi feita, novamente em duas partes, sendo que a primeira foi a exploração dos dados extraídos da API Yahoo Finance e a segunda com dados extraídos do Kaggle.

Como os dados sobre a bolsa de valores foram extraídos em dois diferentes datasets `df_ibov` que terá somente dados do índice IBOVESPA (Abertura, fechamento, alta etc.) e o `data_acoes` que terá o fechamento do índice IBOVESPA, Itaú e B3.

Para iniciar, primeiro foi feita a descrição estatística deles:

Primeiro o data frame com mais de uma ação, `df_acoes`:

```
df_acoes.describe()
```

	ITUB4	B3SA3	IBOV
count	1334.000000	1334.000000	1334.000000
mean	26.374373	23.811064	72.401909
std	6.868770	11.300389	20.450424
min	13.981800	8.970000	37.497000
25%	20.039975	16.042500	53.779750
50%	26.016700	21.090000	70.792500
75%	33.450001	30.437500	86.087000
max	39.689999	53.130001	119.528000

E em segundo dataframe, com o Índice IBOVESPA o `df_ibov`.

```
df_ibov.describe()
```

	Open	High	Low	Close	Adj Close	Volume
count	1334.000000	1334.000000	1334.000000	1334.000000	1334.000000	1.334000e+03
mean	72370.781109	73093.95952	71666.671664	72401.908546	72401.908546	4.213815e+06
std	20446.408706	20565.48466	20307.051005	20450.423745	20450.423745	2.057388e+06
min	37501.000000	38031.000000	37046.000000	37497.000000	37497.000000	0.000000e+00
25%	53764.500000	54236.000000	53309.750000	53779.750000	53779.750000	3.129075e+06
50%	70699.500000	71480.000000	69576.000000	70792.500000	70792.500000	3.751400e+06
75%	86066.500000	87250.250000	85459.250000	86087.000000	86087.000000	4.654675e+06
max	119528.000000	119593.000000	118108.000000	119528.000000	119528.000000	1.675150e+07

Nessa descrição podemos ver que a mínima e máxima modificaram bastante, principalmente o índice IBOVESPA, o que já indica uma grande movimentação e crescimento da economia privada.

Após isso, foi verificado se havia algum dado nulo nos datasets. O `df_ibov`, com dados somente do IBOVESPA não tinha nenhum dado nulo, porém conforme dito na sessão anterior, para o dataset de ações, foram encontrados 8 valores nulos, os quais foram verificados conforme abaixo:

```
print(df_acoes.isna().sum())
```

Date	0
ITUB4	0
B3SA3	0
IBOV	8
dtype:	int64

```
print(df_acoes[df_acoes.isna().any(axis=1)])
```

	Date	ITUB4	B3SA3	IBOV
607	2017-06-15	24.273300	18.780001	NaN
667	2017-09-07	28.059999	22.760000	NaN
692	2017-10-12	30.000000	24.040001	NaN
707	2017-11-02	27.553301	23.879999	NaN
716	2017-11-15	27.280001	22.330000	NaN
719	2017-11-20	28.353300	23.680000	NaN
744	2017-12-25	28.353300	22.290001	NaN
766	2018-01-25	33.580002	25.870001	NaN

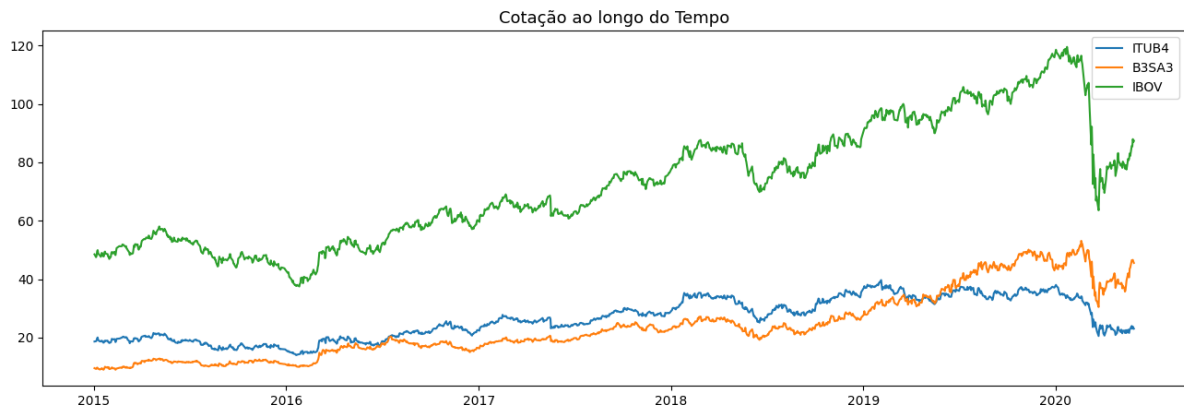
Analisando esses dados, foi visto que por alguma razão os dados são referentes a datas consideradas como feriado aqui no Brasil, como não houveram movimentações nesses dias, essas linhas foram removidas.

A próxima etapa para analisar e deixar de forma mais visual os dados tratados, foi utilizada a biblioteca Matplotlib para plotar graficos de linhas referente aos datasets. Nesse caso, o dado de interesse aqui é o fechamento, dessa forma esse foi o dado plotado.

```
plt.figure(figsize=(10, 4))
for ticker in tickers:
    plt.plot(df_acoes['Date'], df_acoes[ticker])

plt.legend(tickers, loc='upper left')
plt.grid()
plt.title('Cotação ao longo do Tempo', fontsize = 13)
plt.show()
```

Gerando o gráfico abaixo:

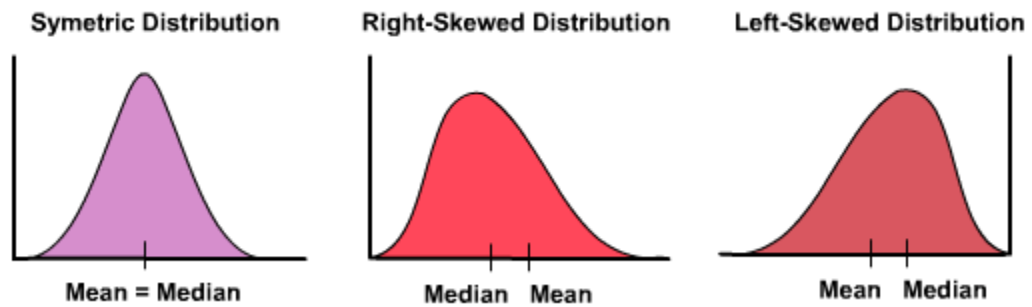


Observando o gráfico podemos perceber uma crescente movimentação do índice e dos demais papéis ao longo do tempo. Mas olhando para 2020 percebemos essa grande queda, que está ligada ao Corona Vírus. Mas isso levou ao questionamento: Qual a correlação dessa queda com os casos que temos no Brasil? Na sessão dos resultados veremos exatamente se realmente houve impacto no bolsa de valores, dado o cenário de aumento de número de casos de Corona Vírus no Brasil.

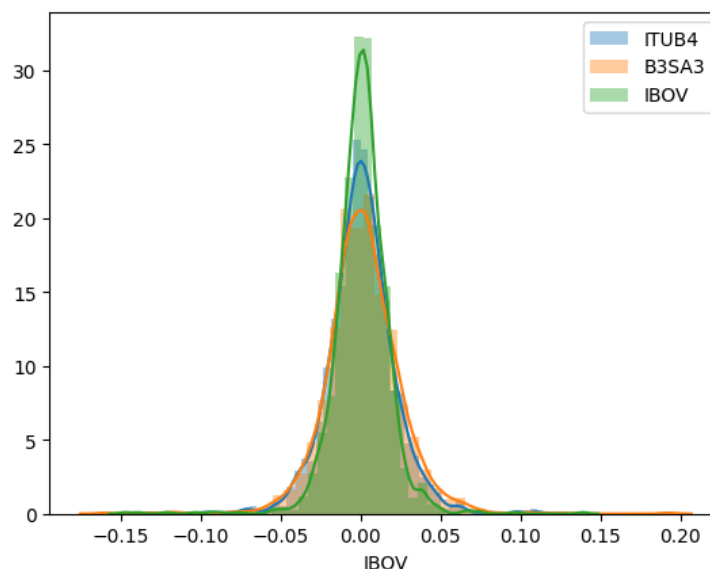
Antes de aprofundar a análise na comparação com o dataset do Coronavírus, foi verificado a distribuição do dataset, se ele é normalizado ou não. Para isso foi utilizada a distribuição Gaussiana.

A distribuição normal conhecida também como distribuição gaussiana é uma das mais importante distribuições contínuas. Sua importância se deve a vários fatores, entre eles podemos citar o teorema central do limite, o qual é um resultado fundamental em aplicações práticas e teóricas, pois ele garante que mesmo que os dados não sejam distribuídos segundo uma normal a média dos dados converge para uma distribuição normal conforme o número de dados aumenta.

As suas variáveis podem estar com 3 tipos de formato, como mostra a figura abaixo. temos três gráficos, onde o primeiro da esquerda representa a distribuição normal e já os demais, são distribuições assimétricas, podendo ser assimétrica positiva ou negativa.

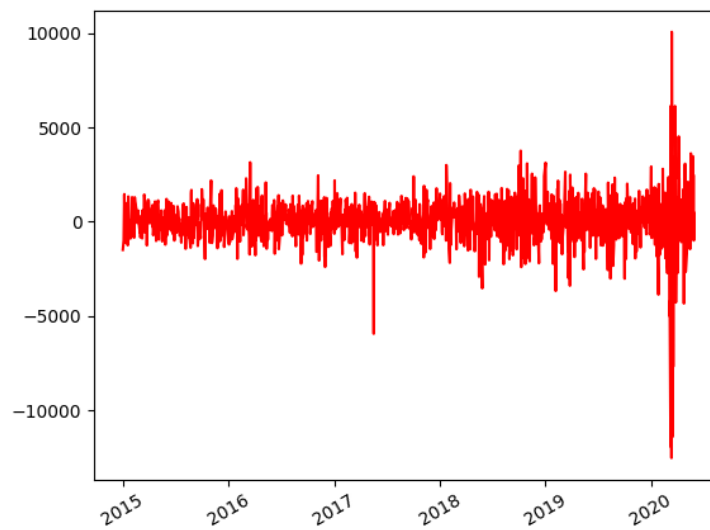


Geralmente quando há assimetria, é interessante tratar esses dados, podendo aplicar Raiz quadrada ou cúbica, log, e muitas outras técnicas. Para o caso do nosso estudo, podemos ver que os dados da bolsa de valores são normalmente distribuídos conforme o gráfico gerado abaixo:

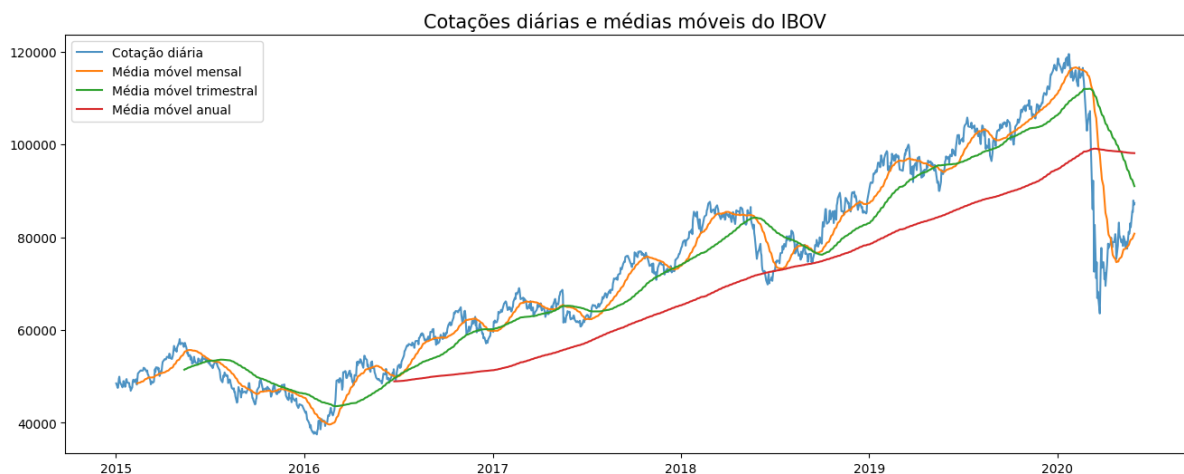


Para entender a movimentação do Índice IBOVESPA foi plotado gráfico com a variação (gráfico do tipo *mdates* no *matplotlib*) e em seguida um gráfico de linha com o valor do fechamento diário versus sua média dos últimos 30 dias, 90 dias e 365 dias.

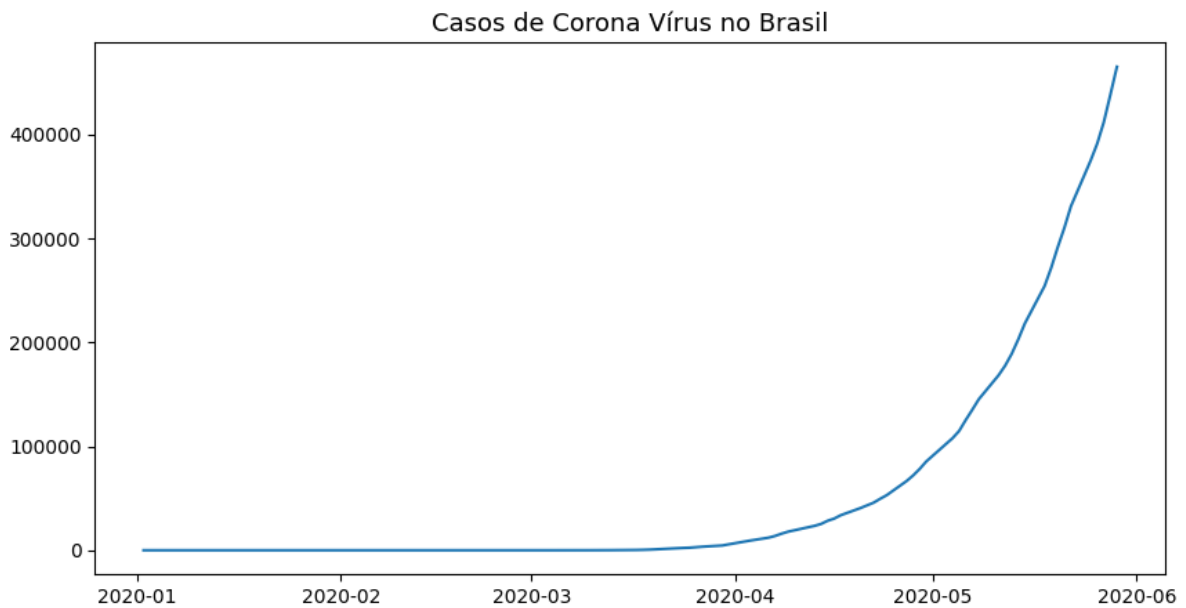
Para o gráfico abaixo, podemos perceber que a variação ocorre bem simétrica também, mas sem grandes oscilações ao longo do tempo, entretanto, quando olhamos apenas 2020 vemos essa movimentação muito acima da curva padrão que foi observada no ao longo do período analisado.



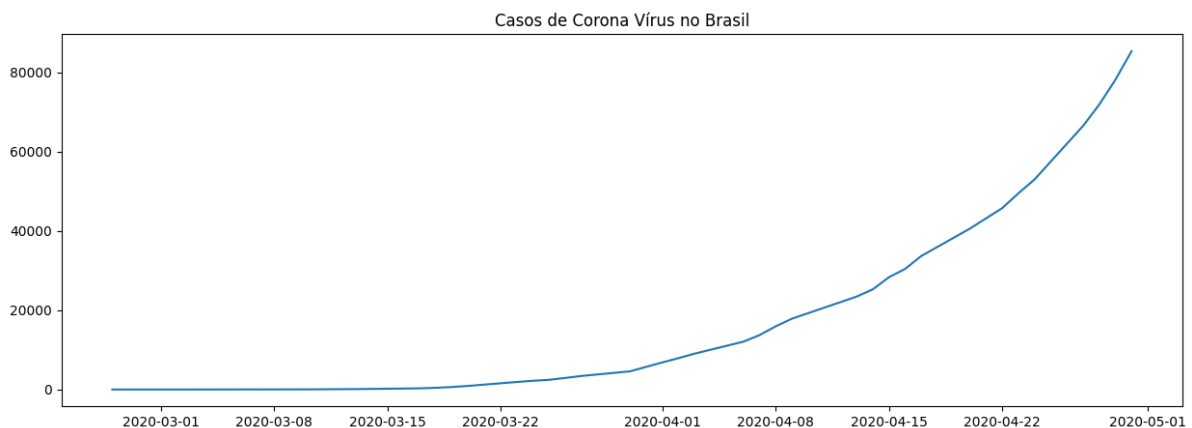
Quando olhamos apenas os 30 últimos dias podemos ver que segue bastante a tendência apresentada, oscilando pouco. Já quando olhamos 90 dias e 365, o equivalente a um trimestre e um ano, respectivamente, que são justamente os períodos em que as empresas divulgam os resultados, podemos ver uma oscilação maior, com grandes altas ou grandes quedas, mas no geral tende a subir. Ainda observando a janela de 30 dias, podemos ver que o período dado pela Pandemia, no início de 2020, foi onde houve uma grande queda.



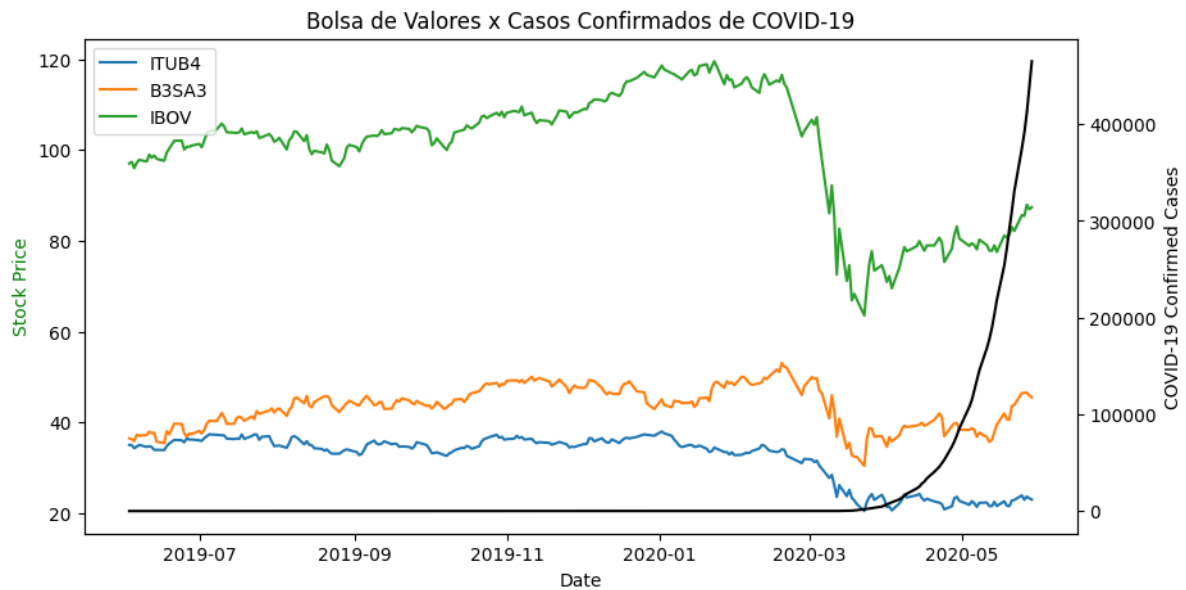
Com essa hipótese levantada, foi o momento que entrou a análise do dataset sobre o Corona Vírus. Então, foi plotado o gráfico do número de casos no Brasil, partindo do dataset em que foi feito o merge desses dados.



Com esse gráfico somado as informações iniciais sobre o tema, podemos ver que os casos no Brasil começaram no final de fevereiro, precisamente no dia 26. A curva começou a subir de fato no final de março, abaixo um gráfico mais aproximado do período inicial dos casos ocorrendo no Brasil, de 26/02/2020 a 30/04/2020 onde podemos ver com mais clareza o crescimento dos números, saindo de 1 caso em janeiro e passando de 80 mil casos no final de Abril.

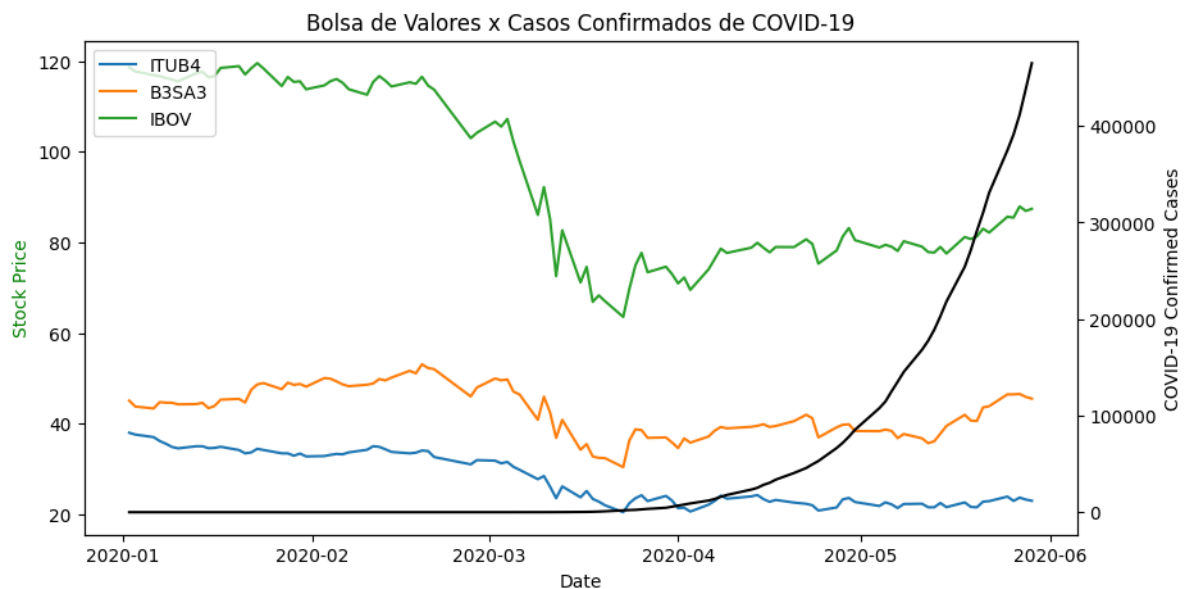


Seguindo a análise, os gráficos foram plotados juntos, para verificar, inicialmente um plot com um horizonte de tempo maior, onde podemos ver exatamente onde o IBOVESPA e demais papeis começaram a apresentar uma forte queda.



Essa análise levantou um *insight*, indicando que há uma grande chance do aumento do número de casos confirmados da doença no Brasil não ter afetado a bolsa de valores, onde verificaremos logo em seguida.

Em seguida, foi feita a aproximação da visualização do mesmo gráfico e verificar que a curva de subida da doença começa a realmente crescer em abril, onde o IBOVESPA já começava a indicar uma possível recuperação.

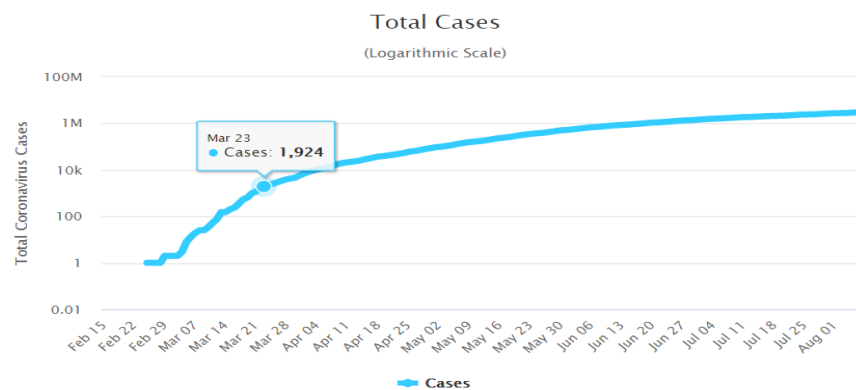


Após essa verificação, uma outra análise foi feita com esses dados de 2020 no dataset em questão, verificando a mínima e a máxima do Índice IBOVESPA e do dataset sobre o Corona Vírus.

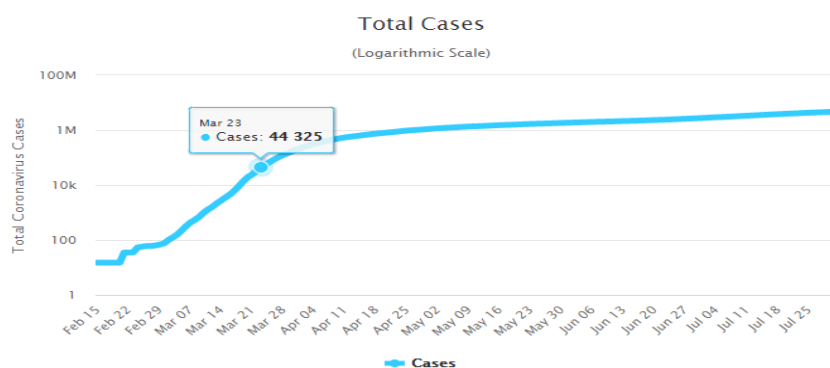
Máxima no período		Mínima no período	
IBOV	119528.0	IBOV	63570.0
Confirmed Cases	465166.0	Confirmed Cases	0.0
dtype: float64		dtype: float64	
Datas de referência		Datas de referência	
IBOV	2020-01-23	IBOV	2020-03-23
Confirmed Cases	2020-05-29	Confirmed Cases	2020-01-02
dtype: datetime64[ns]		dtype: datetime64[ns]	

Aqui podemos ver que o índice IBOVESPA saiu de sua máxima em 2020, em 23 de janeiro, com pouco mais de 119 mil pontos e atingiu sua mínima 63 mil pontos, em 23 de março, data em que o total de casos confirmados de Corona Vírus no Brasil estava num total de 1.924, um valor bem baixo se comparado, no mesmo dia, com os Estados Unidos com 44.325 casos e China com 81.171 casos (segundo o site Worldometers). O que pode indicar que o medo da doença e do desconhecido pode ter causado efeito manada e/ou de pessimismo na bolsa de valores, o que é bem comum quando eventos globais ou eventos locais com grandes proporções ocorrem. Abaixo o gráfico do estados unidos e em seguida o da china, precisamente na data indicada:

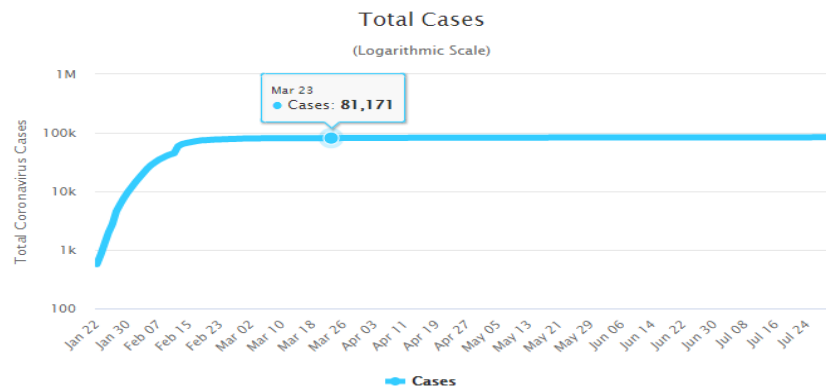
Brasil com 1.924 casos em 23 de março de 2020.



Estados unidos com 44.325 casos em 23 de março de 2020.



China com 81.171 casos em 23 de março de 2020.



O total de casos confirmados no Brasil só chegou no mesmo patamar que a China por volta do dia 30 de abril, e nessa data o IBOVESPA já estava na casa dos 80 mil pontos, ou seja, já havia subido bastante, próximo de 25%. Enquanto isso, o total de casos também estava subindo, indicando que realmente o total de casos no Brasil não foram afetados tão fortemente pelos números. Para ter certeza, verificamos através da Correlação de Pearson.

A Correlação de Pearson, também chamada de “coeficiente de correlação produto-momento” ou simplesmente de “ ρ de Pearson” mede o grau da correlação entre duas variáveis. Este coeficiente, normalmente representado por ρ assume apenas valores entre -1 e 1.

Este coeficiente, normalmente representado pela letra “r” assume apenas valores entre -1 e 1.

- $r = 1$ Significa uma correlação perfeita positiva entre as duas variáveis. Isto é, na medida que uma cresce, a outra também cresce.
- $r = -1$ Significa uma correlação negativa perfeita entre as duas variáveis. Isto é, se uma aumenta, a outra sempre diminui.
- $r = 0$ Significa que as duas variáveis não dependem linearmente uma da outra. No entanto, pode existir uma outra dependência que seja “não linear”. Assim, o resultado $r=0$ deve ser investigado por outros meios.

Outra forma de interpretar a correlação é com

- 0.9 a 1 positivo ou negativo indica uma correlação muito forte.
- 0.7 a 0.9 positivo ou negativo indica uma correlação forte.
- 0.5 a 0.7 positivo ou negativo indica uma correlação moderada.
- 0.3 a 0.5 positivo ou negativo indica uma correlação fraca.

- 0 a 0.3 positivo ou negativo indica uma correlação desprezível.

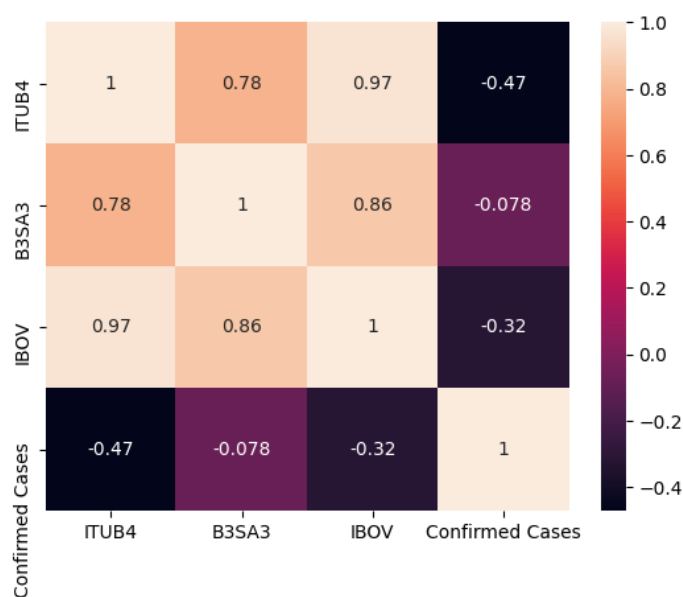
Seguindo assim, foi aplicado o método `corr()` do pandas, para verificar a correlação entre as variáveis dentro do dataset e também plotado um gráfico, mapa de calor para ficar visualmente melhor. Abaixo o script utilizado para verificar essa correlação.

```
df_merge_2020.corr()

#script para mapa de calor, utilizando o seaborn
import seaborn as sns
sns.heatmap(df_merge_2020.drop('Date', 1).corr(), annot = True)
plt.show()
```

Após a execução do script obtemos os seguintes resultados, sendo o segundo mais visual:

	ITUB4	B3SA3	IBOV	Confirmed Cases
ITUB4	1.000000	0.781913	0.968441	-0.470914
B3SA3	0.781913	1.000000	0.864907	-0.078020
IBOV	0.968441	0.864907	1.000000	-0.321265
Confirmed Cases	-0.470914	-0.078020	-0.321265	1.000000

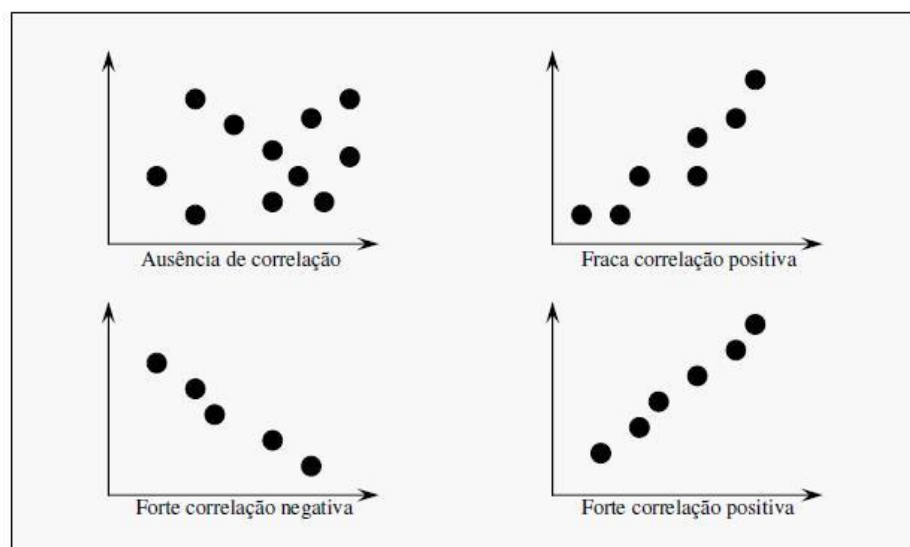


Como podemos observar, a correlação entre o total de casos do Corona Vírus (Confirmed Cases) versus Índice IBOVESPA é de -0,32, ou seja, uma correlação

negativa fraca, Correlação fraca, isto é, na medida que os casos foram aumentando no Brasil, o efeito no índice IBOVESPA foi bem fraco, gerando baixa oscilação, porém mesmo oscilando o índice continuou a crescer. Da mesma forma que se comparado com outros papéis, a correlação continua negativa. Sendo -0,47, fraca, quando comparada com Itaú e -0,078, desprezível, quando comparada com B3.

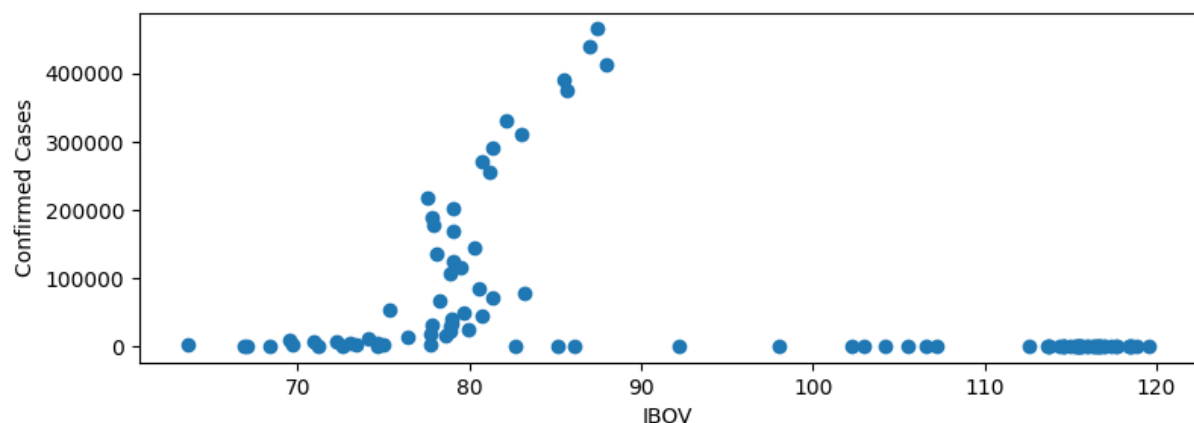
Apenas para efeito de comparação, os papéis são bem correlacionados entre si, sendo que o índice IBOVESPA tem uma correlação muito forte, 0,97, com Itaú e uma correlação forte, 0,86, com B3.

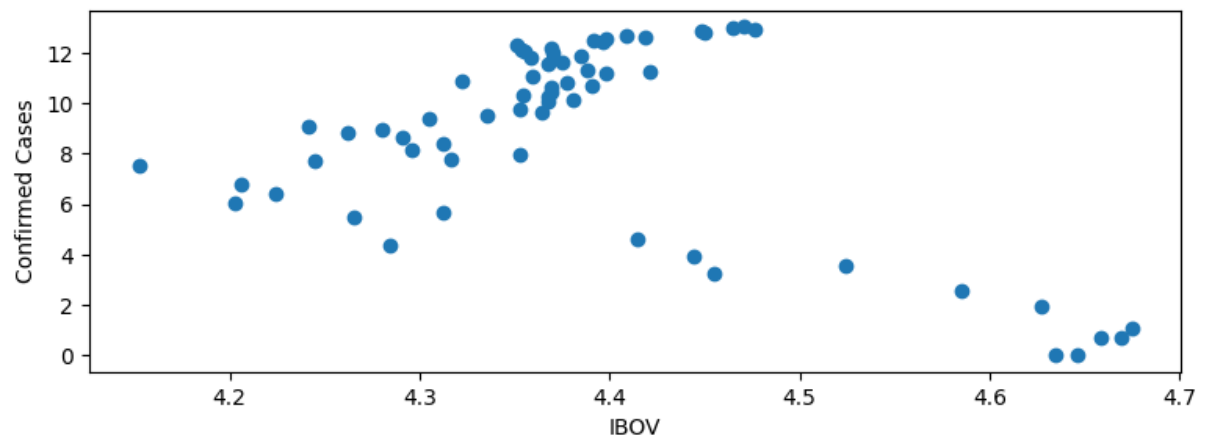
Para finalizar, foi feito gerado o diagrama de dispersão para deixar visualmente melhor a distribuição das variáveis.



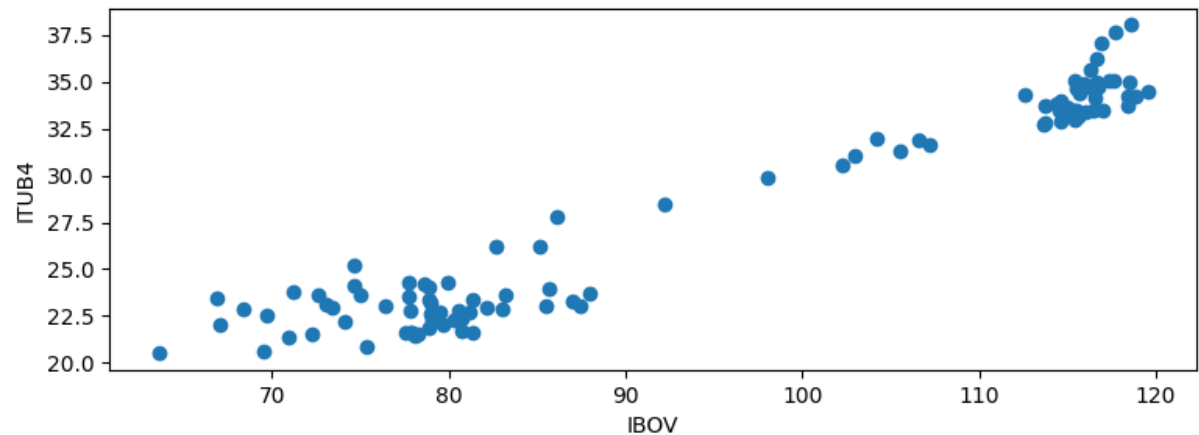
Diagramas de Dispersão.

O diagrama de dispersão de Casos de Corona Vírus em relação ao índice IBOVESPA fica completamente disperso, mostrando que realmente não correlação. Abaixo na escala normal deles e em seguida na escala logarítmica.





Novamente, para efeitos de comparação, o gráfico de dispersão entre Itaú e Índice IBOVESPA mostra uma forte correlação positiva:



Essa análise nos leva a conclusão de que mesmo no cenário de pandemia os casos da doença em território nacional tiveram pouco impacto nos números da Bolsa de Valores.

Podemos sim afirmar que o surto da doença, olhando de forma global e principalmente os países mais influentes como China e Estados Unidos, afetaram muito os números da Bolsa de valores, principalmente quando foi decretada a pandemia, em 11 de março, o que levou a uma grande onda de medo ao redor do mundo. Nesse momento o vírus estava em seus estágios iniciais no Brasil, mas olhando de forma global, a doença espalhava-se com grande velocidade.

Na mesma data em que foi decretado o estado de Pandemia pela OMS as bolsas de forma geral despencaram, sendo que no Brasil houve grandes ondas de desespero, gerando a queda que observamos nos gráficos de fechamento do índice. Nessa mesma semana vimos algumas notícias que comprovavam esses fatos.

Economia

Bolsa despenca 7,64% após OMS decretar pandemia de coronavírus

Dólar subiu e voltou a fechar acima de R\$ 4,70

Economia

Índice Bovespa despenca 14,78% em sessão com quase 3 circuit breakers

Bolsa fechou com pior desempenho desde 1998



Publicado em 12/03/2020 - 19:15 Por Paula Arend Laier - Repórter da Reuters - São Paulo

Circuit breaker é um mecanismo da bolsa de valores que foi projetado para proteger os investidores quando há muitas vendas, ocasionando quedas bruscas nos preços dos ativos negociados. O mecanismo atua por meio de critérios pré-estabelecidos, que determinam a paralisação do pregão por um tempo determinado quando a queda do dia atinge certo patamar percentual. Esses eventos são bem raros de acontecer, mas quando ocorrem é justamente marcado por alguma influência de acontecimentos no mundo/território nacional.

Segundo o site da folha, nos anos 2000, o primeiro circuit break que houve no Brasil foi na crise financeira de 2008, onde esse mecanismo foi acionado mais de 5 vezes dentro de 25 dias (entre o final de setembro e outubro). Após isso, apenas em 2017, onde foi acionado apenas uma vez, no dia conhecido como Joesley Day, acontecimento nacional. Seguindo a crise de coronavírus, onde após a OMS decretar a pandemia em março, o mecanismo foi acionado 6 vezes dentro de 10 dias.

Se fossemos olhar para o gráfico da Bolsa de Valores, em algum momento mais para o futuro, tenderíamos a acreditar que o efeito maior da queda relacionada a 2020 é fruto do vírus no Brasil, mas podemos concluir que o cenário mundial, nos estágios iniciais, teve um impacto muito maior nos números da bolsa de valores, do que os casos no âmbito nacional. Dessa forma, após essa análise podemos aproveitar as grandes oscilações no índice IBOVESPA e aplicar modelos de machine learning para verificar como esses se comportam diante desses números apresentados.

5. Criação de Modelos de Machine Learning de Regressão

Feita a análise exploratória e os tratamentos necessários com os dados, foram aplicados modelos de regressão sobre os dados do IBOVESPA para prever o fechamento baseado no valor de abertura. Veremos agora como os diferentes modelos foram aplicados.

Em todos os modelos utilizado, a biblioteca "*Train Test Split*" foi utilizada. Esse procedimento é usado para estimar o desempenho dos algoritmos de aprendizado de máquina quando eles são usados para fazer previsões de dados com dados que nunca foram usados para treinar o modelo em questão.

É um procedimento rápido e fácil de executar, cujos resultados permitem comparar o desempenho dos algoritmos de aprendizado de máquina. Embora seja simples de usar e interpretar, há momentos em que o procedimento não deve ser usado, como quando temos um pequeno conjunto de dados e situações em que é necessária uma configuração adicional. O procedimento é apropriado quando você tem um conjunto de dados muito grande ou que seja um modelo caro para treinar. Foi utilizada a biblioteca de aprendizado de máquina do scikit-learn para executar o procedimento Train Test Split.

E de forma resumida os modelos de regressão seguem a seguinte lógica: estimar o valor de algo baseado em uma série de outros dados históricos, os modelos seguirão conforme abaixo:

Os Modelos de regressão nos permitem estudar as relações entre duas variáveis numéricas contínuas e prever baseado nesses dados.

Esses modelos serão avaliados com o coeficiente de determinação, conhecido como R^2 . Ele diz o quanto o modelo explica seus resultados. O cálculo dele, envolve três medidas: Soma Total dos Quadrados (STQ), Soma dos Quadrados dos Resíduos (SQU) e Soma dos Quadrados de Regressão (SQR). O valor de R^2 é a divisão da variação explicada pela variação total dos dados (SQR dividido pelo SQT). O Resultado sempre é um valor entre 0 e 1. Quanto mais próximo de 1, melhor o resultado.

Abaixo veremos como cada um dos modelos foi aplicado em Regressão Linear Simples, Árvore de Decisão e Support Vector Regression.

5.1. Simple Linear Regression (Regressão Linear Simples)

Um modelo de regressão linear é uma equação matemática que fornece uma relação linear, ou seja, de linha reta entre duas variáveis, comumente chamada de x e y. o modelo de regressão linear simples é uma equação matemática que inclui somente duas variáveis e apresenta uma relação em linha reta entre elas

Nesse estudo teremos uma variável de entrada (x) também chamada de variável preditor, explicativa, independente. No caso desse estudo, será a Abertura (Open). E teremos uma variável de saída (y) também chamada de variável dependente resposta, resultado ou dependente. Esse é o vamos tentar descobrir/prever com nossos modelos. No caso o fechamento (Close)

Para a regressão linear simples foi feita a importação das bibliotecas para a aplicação do modelo, aplicada a divisão da base em treino e teste, aplicado o treino e por fim feita a predição.

```
# Importação das bibliotecas
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split

# Definição das Variaveis X e Y
X = df_ibov[['Open']]
y = df_ibov[['Close']]

# Divisão do dataset em Treino e Teste
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
                                                    random_state = 1)

# Criação e Treinamento do Modelo
regressor = LinearRegression()
regressor.fit(X_train, y_train)

# Predição do valor de X
y_pred = regressor.predict(X_test)
```

5.2. Decision Tree Regression (Árvore de Regressão)

Outro modelo que foi aplicado foi a Árvore de Regressão, modelo da árvore de decisão, para variáveis contínuas. Uma árvore de regressão é idêntica a uma árvore de decisão porque também é formada por um conjunto de nós de decisão e perguntas, mas o resultado, em vez de uma categoria, é um escalar (número que pertence a uma escala).

Seguindo a mesma lógica, foram feitas as importações, definição das variáveis necessárias para o modelo, divisão em treino e teste e por fim treino/predição baseado nisso.

```
# Importação das bibliotecas
from sklearn.tree import DecisionTreeRegressor
from sklearn.model_selection import train_test_split

# Definição das Variáveis X e Y
X = df_ibov[['Open']]
y = df_ibov[['Close']]

# Divisão do dataset em Treino e Teste
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
                                                    random_state = 1)

# Criação e Treinamento do Modelo
regressor = DecisionTreeRegressor(random_state = 0)
regressor.fit(X_train, y_train)

# Predição do valor de X
y_pred = regressor.predict(X_test)
```

Porém utilizando o Regressor dessa forma, gerou um pequeno problema: o modelo sofreu overfitting, então foi necessário ajustar as variáveis. A definição do modelo ficou da seguinte forma:

```
regressor = DecisionTreeRegressor(max_depth=5, max_leaf_nodes=20,
                                   random_state = 0)
regressor.fit(X_train, y_train)
```

A profundidade da árvore foi limitada a 5 níveis e máximo de nós folha foi de 20, permitindo assim um modelo que não seja super ajustado.

5.3. Support Vector Regression (Máquina/Regressão de Vetor Suporte)

O último modelo que foi utilizado nesse estudo foi o Support Vector Regression. O algoritmo Support Vector Regrssions (SVR) é um tipo de Support Vector Machine (SVM). É um método de aprendizagem de máquina que tenta tomar dados de entrada e classificá-los em uma entre duas categorias, no caso desse estudo vamos utilizar a versão para dados contínuos, ou seja, para a aplicar a regressão. Para que SVR seja eficaz, primeiramente é necessário utilizar um conjunto de dados de entrada e de saída de treinamento para construir o modelo de máquina de vetores de suporte que pode ser utilizado para classificação de novos dados. Foram utilizados dois métodos de cálculo: Linear e RBF (Radial Basis Function) que é utilizado para dados não lineares (para aplicar ele em dados lineares é preciso fazer um escala padrão com os dados, standard scale).

Por fim, nesse modelo, a lógica foi um pouco diferente, de forma resumida: foram feitas as importações, definição das variáveis necessárias para o modelo, divisão em treino e teste e por fim treino/predição nos dois modelos, sendo Linear e RBF.

```
# Importação das bibliotecas
from sklearn.svm import SVR
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler

# Definição das Variaveis X e Y
X = df_ibov[['Open']]
y = df_ibov[['Close']]

# Divisão do dataset em Treino e Teste
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state = 1)

# Criação e Treinamento do Modelo
regressor_rbf = SVR(kernel = 'rbf')
regressor_rbf.fit(X_train, y_train)

regressor_linear = SVR(kernel = 'linear')
regressor_linear.fit(X_train, y_train)

# Predição do valor de X
y_pred_rbf = regressor_rbf.predict(X_test)
y_pred_linear = regressor_linear.predict(X_test)
```


6. Apresentação dos Resultados

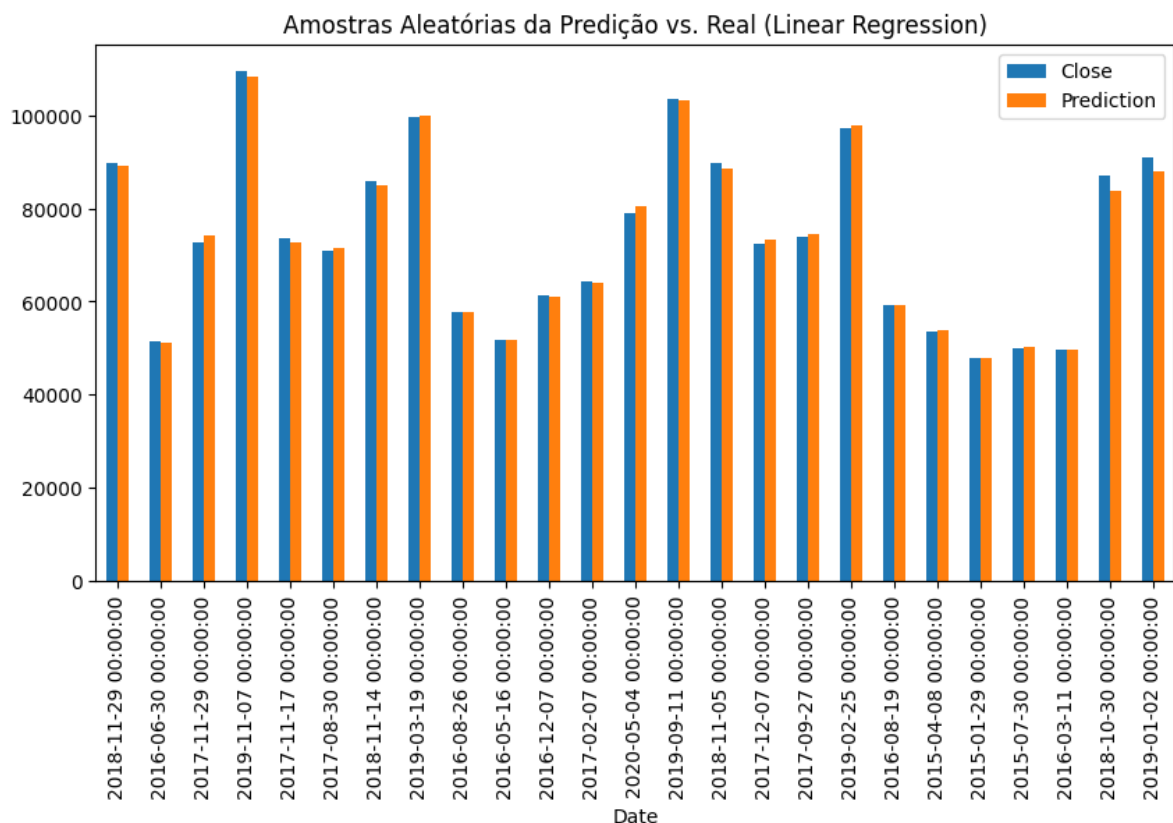
Nessa seção veremos os resultados obtidos pelos modelos de Machine Learning treinados com Abertura para prever o Fechamento. Veremos também a comparação entre eles. Também, no final, os dois modelos lineares (Regressão Linear e Support Vector Regression com o Kernel Linear) também serão comparados e verificados em um outro tipo de predição.

6.1. Resultados da Regressão Linear Simples

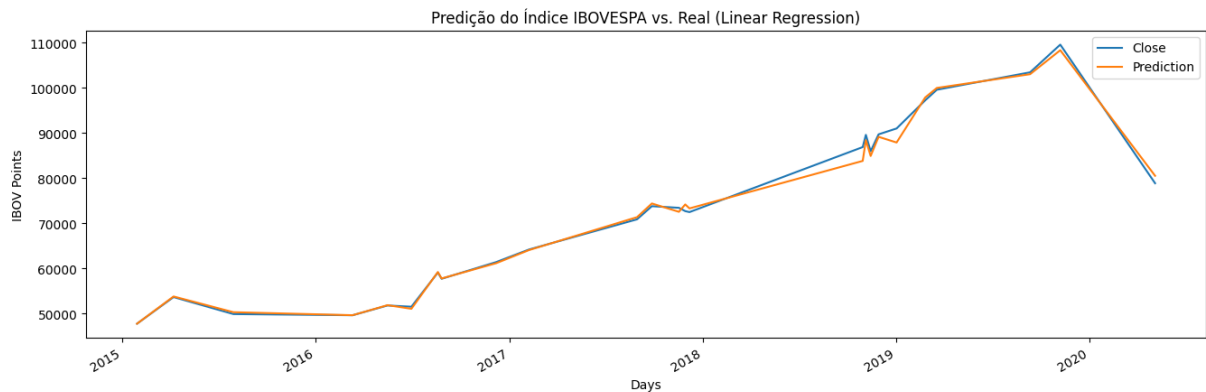
O Primeiro modelo treinado foi o de Regressão Linear Simples. Analisando os resultados, podemos perceber que o modelo se comportou muito bem, apresentando resultados consistentes e bem próximos do que realmente ocorreu, quando comparado com o dataset original.

O gráfico abaixo é representado com alguns selecionados de forma aleatória, com esses dados podemos ver que o modelo atingiu um valor bem próximo do real, na maioria dos casos dessa amostra.

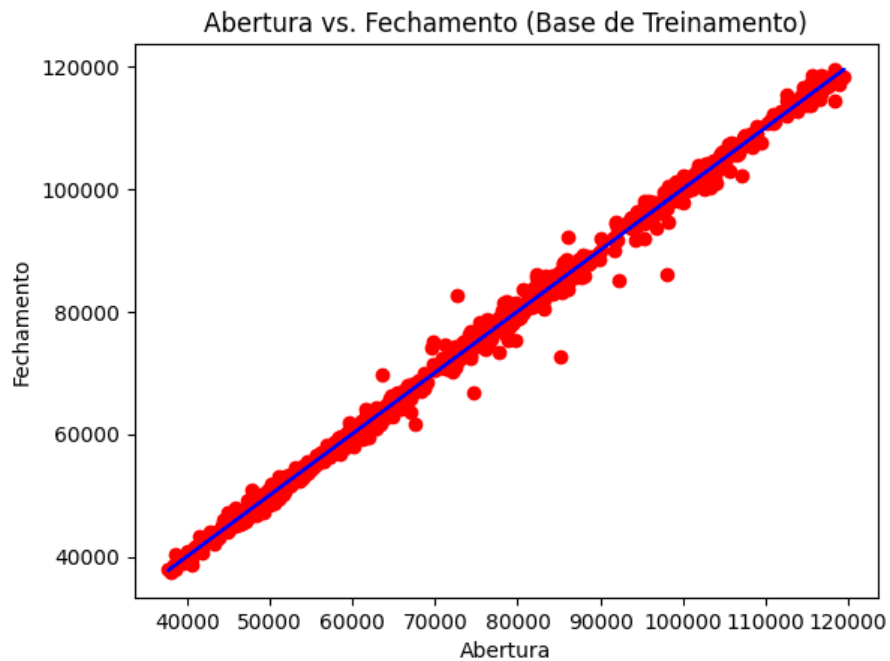
espalhados de forma aleatório que nas predições feitas pelo modelo, ele se aproximou bastante na maioria.



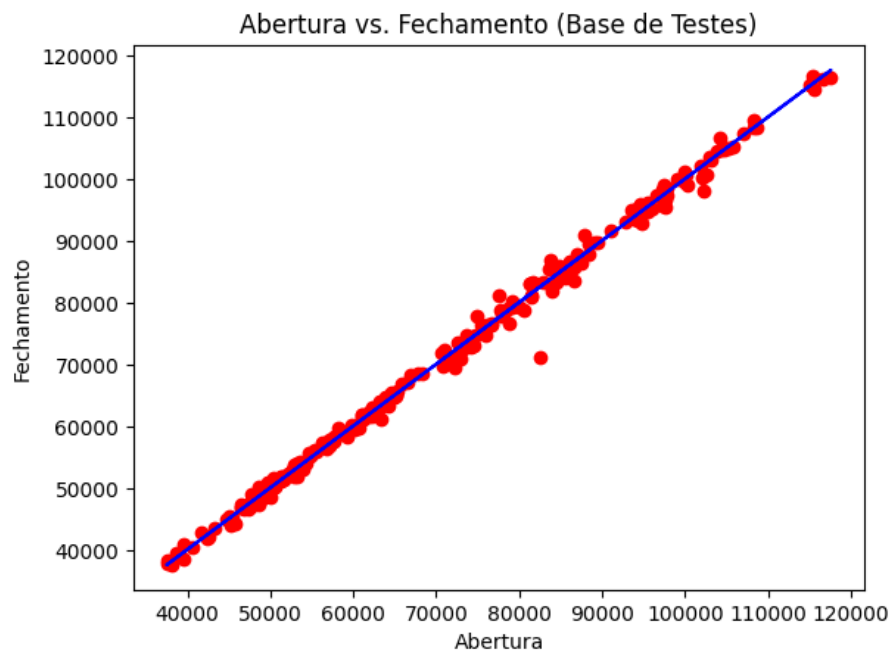
Com o gráfico de linha, ao longo do tempo podemos ver com mais precisão como o modelo se comportou.



Com o gráfico de dispersão, podemos comprovar que ele foi treinado de uma forma que passando o valor da abertura, ele tende a acertar a grande maioria com uma grande acurácia, onde somente em alguns poucos casos ele não consegue prever. O gráfico abaixo é relacionado a base de treinamento.



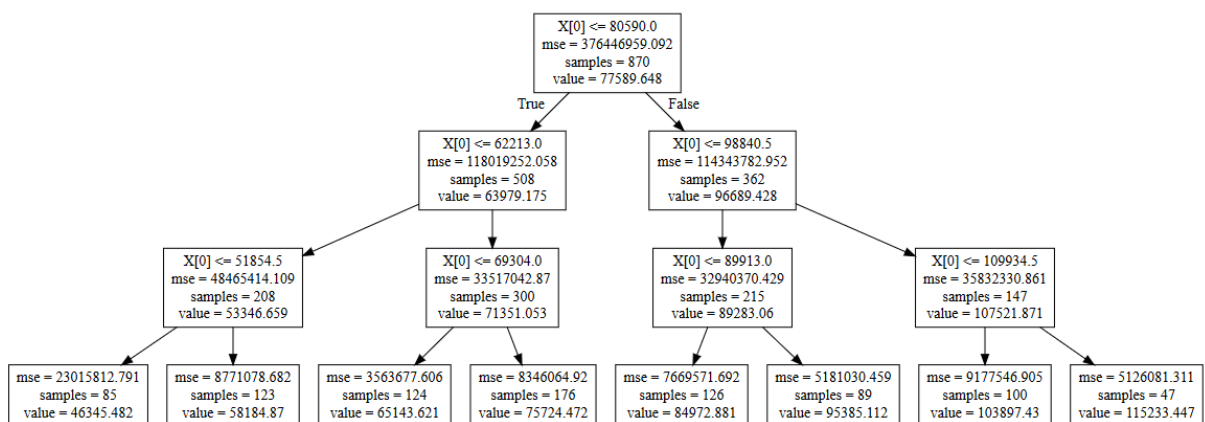
Quando aplicamos utilizamos a base de testes, podemos perceber que o comportamento se assemelha bastante. Abaixo podemos ver esse gráfico.



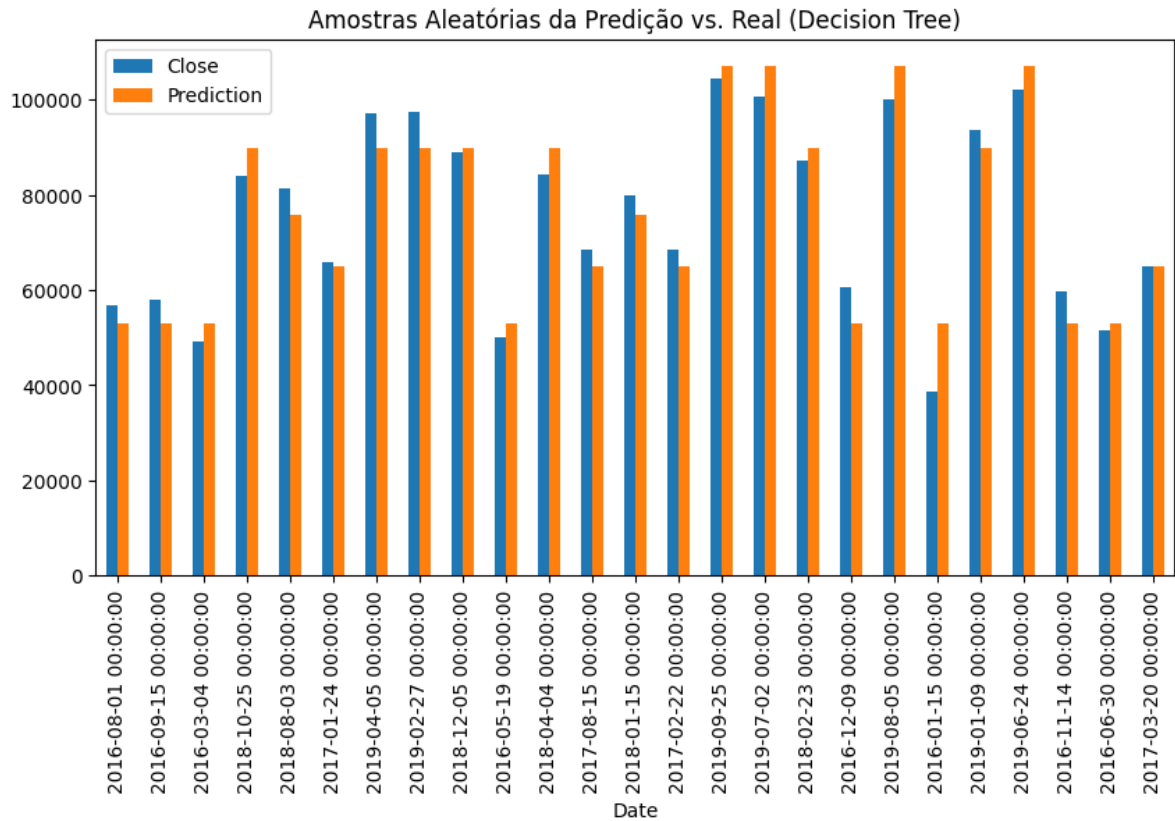
Por fim, utilizando o Coeficiente de Determinação, chegamos em um ótimo resultado para essa predição feita pelo modelo: **0.9963**.

6.2. Resultados da Árvore de Regressão

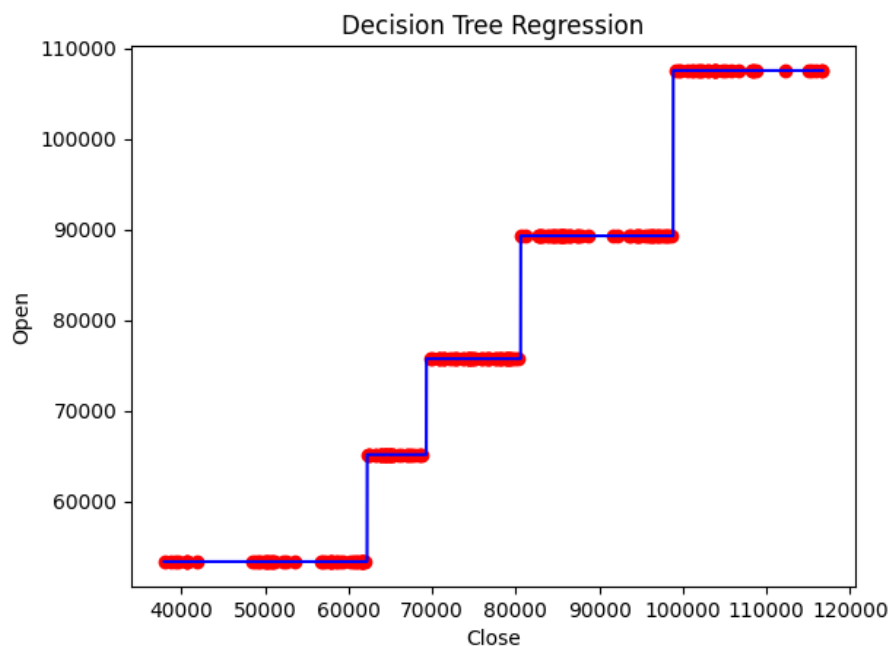
O segundo modelo analisado foi a Árvore de Regressão, esse modelo também se comportou bem, porém teve um resultado menor que o a Regressão Linear Simples. Conforme podemos o modelo foi ajustado para gerar poucas Nós Folha e Poucos níveis de profundidade, isso foi necessário pois sem esse ajuste o modelo sofreu *Overfitting*, ou seja, ele se super ajustou ao modelo. Por isso foi necessário limitar, e árvore gerada foi esse abaixo, conforme podemos visualizar:



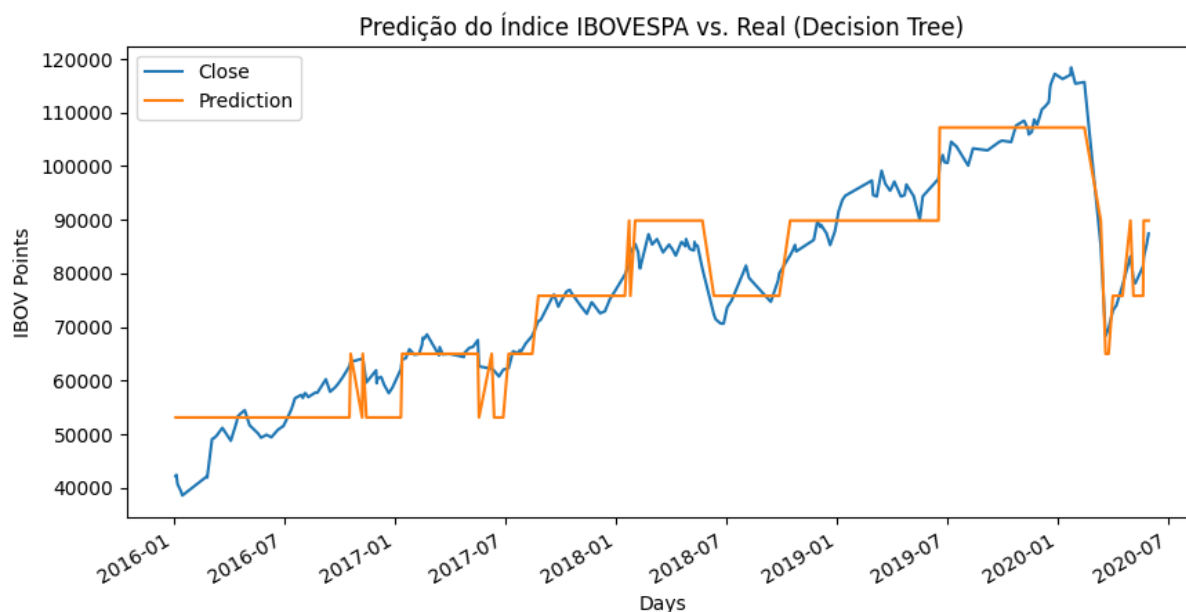
Em seguida, novamente, foram selecionados alguns dados da predição, de forma aleatória, e com esses dados podemos ver que o modelo atingiu também um valor bem próximo do real, quando utilizado a predição com base na Abertura.



Após isso, podemos analisar a dispersão do modelo, que segue exatamente da árvore de decisão que temos. Onde os espaços maiores foram os que menos tivemos assertividade.



Como última análise da árvore de regressão, também foi gerado o gráfico do que foi predito vs. o que realmente ocorreu, nesse caso podemos ver que em alguns momentos o modelo se comportou bem e em outros ele passou um tanto longe do que realmente ocorreu.

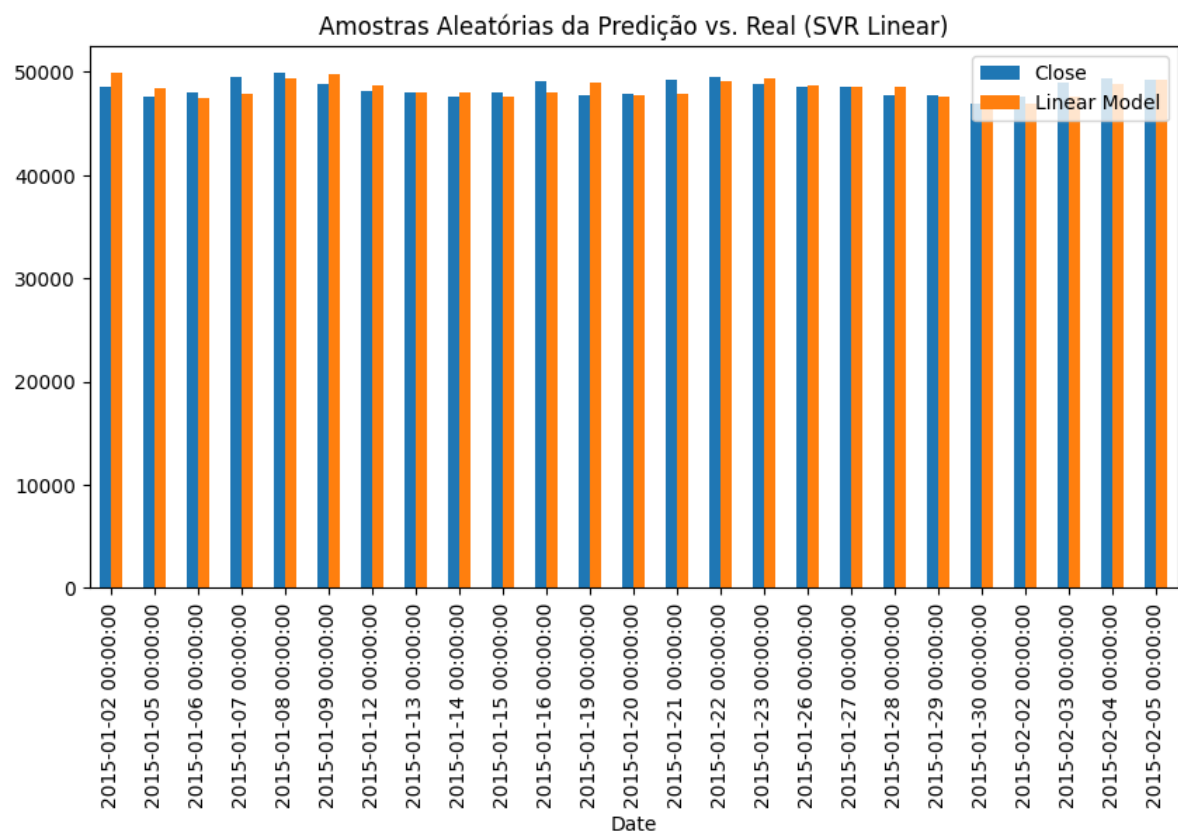
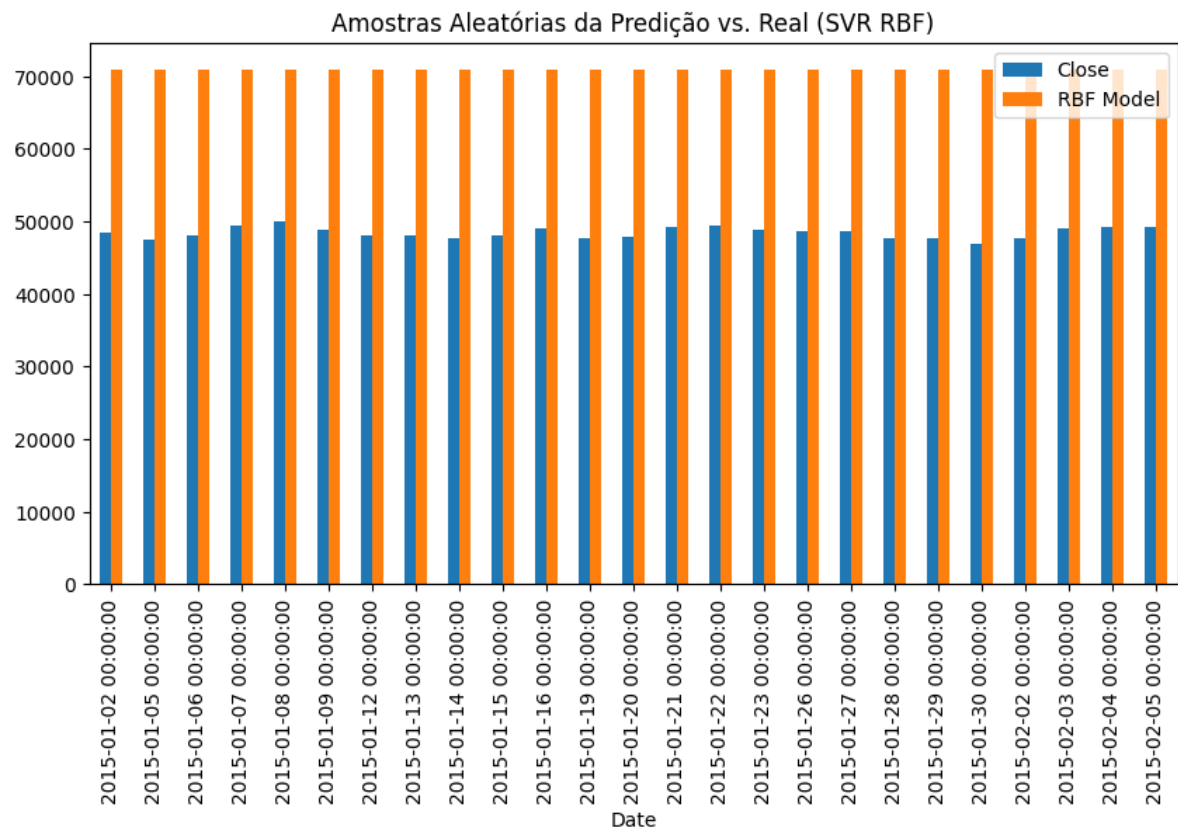


Por fim, utilizando o Coeficiente de Determinação, chegamos em bom resultado, sendo que para essa predição feita pelo modelo o R^2 foi de **0.9246**.

6.3. Resultados da Regressão de Vetor Suporte (SVR)

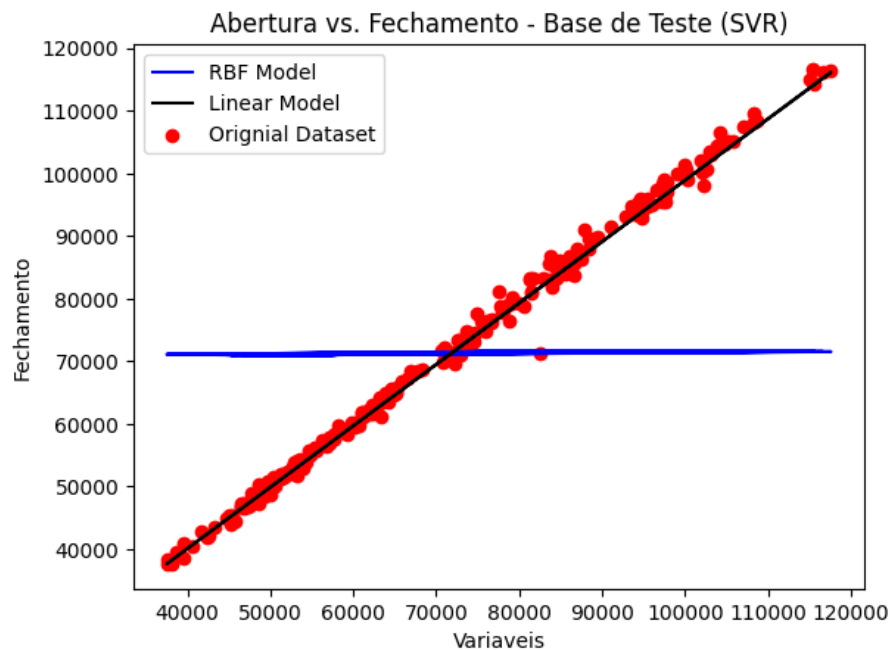
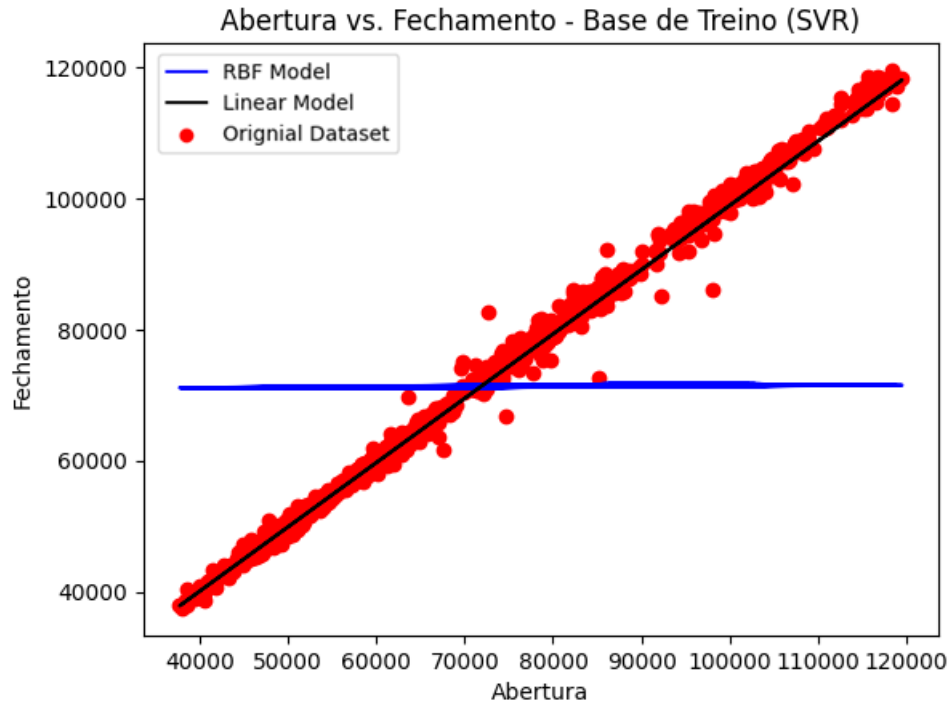
O último modelo que foi treinado foi o Support Vector Regression, utilizando dois Kernels, um para dados lineares (kernel = linear) e outro para dados não lineares (kernel = rbf). Justamente para verificar o comportamento desses diferentes modelos foi feita essa experimentação. E o resultado, como esperado, foi que o RBF não conseguiu ajustar com esses dados, tendo um péssimo desempenho, enquanto o linear obteve um excelente desempenho. Abaixo podemos comparar esses dados gerados por eles e os modelos em si.

Primeiro, após treinar o modelo, foram selecionados dados de forma aleatória para verificar como eles estavam predizendo os dados e esses dados foram comparados com o real, assim como nos demais modelos.

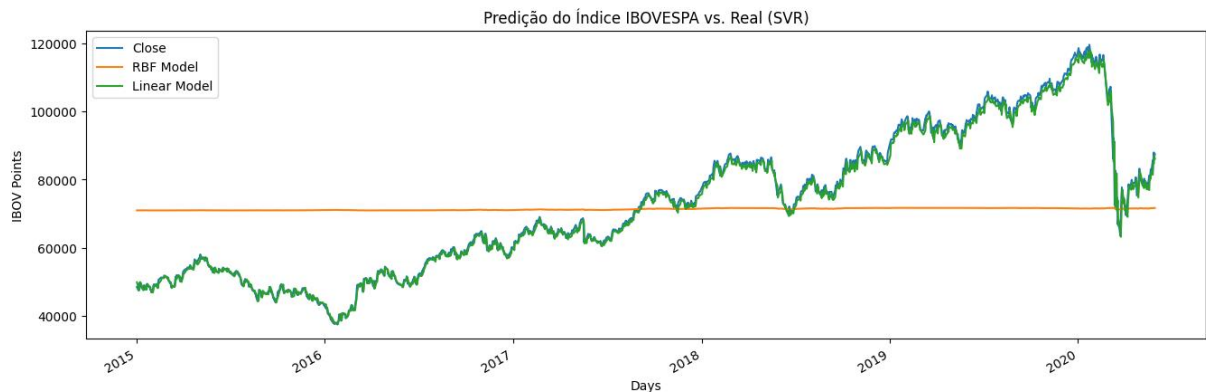


Podemos ver claramente que os do modelo Linear se comportaram muito melhor que o RBF.

Em seguida, para uma segunda análise, foi gerado o gráfico de dispersão para comparar os modelos. Como podemos perceber, o RBF não conseguiu prever os dados enquanto o Linear atingiu um ótimo resultado.



Como última análise do modelo SVR, foi gerado o gráfico de linha sobre o que foi predito vs. o que realmente ocorreu, nesse caso podemos exatamente o mesmo comportamento do gráfico de dispersão: O Modelo Linear conseguiu ficar extremamente próximo do real e o RBF não conseguiu.

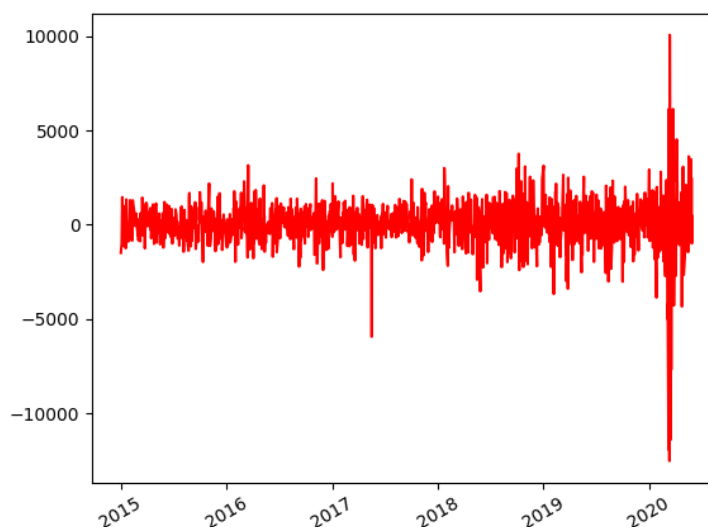


Por fim, utilizando o Coeficiente de Determinação, chegamos no seguinte resultado: O modelo linear obteve o R^2 de **0.9949** e o RBF obteve o R^2 de **0.0224**.

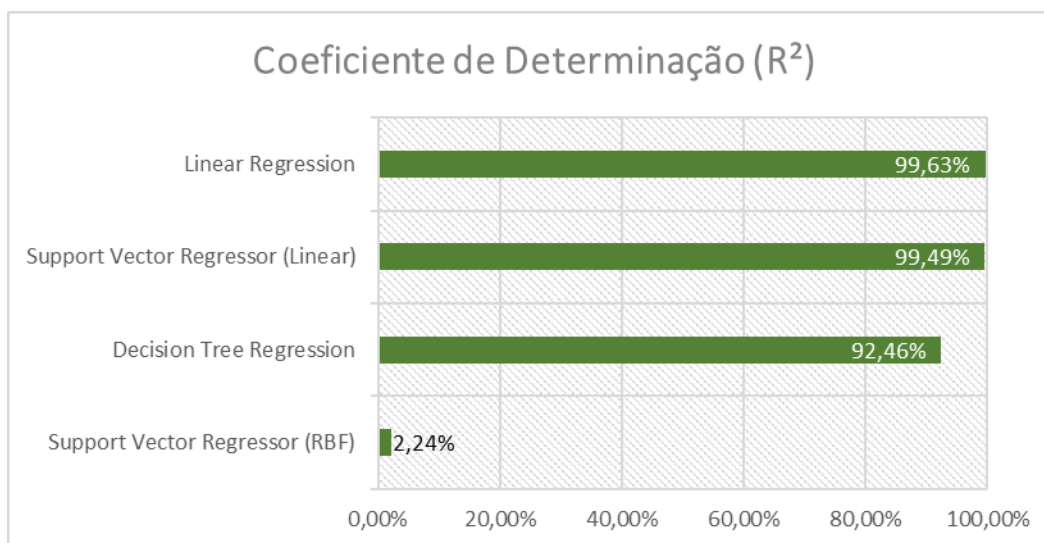
Para o modelo SVR ser treinado com melhores resultados, o ideal seria ter dados não lineares ou aplicar **Feature Scalling**, ou seja, colocar os dados em uma escala padrão (de -1 a 1, por exemplo) e treinar o modelo com esses dados. Nesse caso o SVR pode apresentar resultados excelentes.

6.4. Comparação dos Modelos

Com esse estudo podemos perceber que os modelos lineares se comportam muito bem quando temos uma variável que ajuda a prever a outra, no caso desse estudo foi a Abertura. Ela guia o modelo para uma possível conclusão do dia, baseado no que o modelo aprendeu no treinamento. No estudo podemos perceber que grandes variações (como grandes quedas e grandes altas) esses modelos possuem bastante dificuldade em prever, pois no geral, a variação é baixa, como pudemos ver nesse estudo, e quando fatos externos afetam a bolsa o modelo não consegue se ajustar, justamente por esperar uma previsibilidade maior.



Dos modelos estudados aqui, os que apresentaram melhores resultados foram os Modelos Lineares, com um Coeficiente de Determinação excelente e bem semelhantes.



Para finalizar o estudo, um outro treinamento foi feito com os modelos campeões: Como eles se comportariam na predição dos grandes dias de queda no ano de 2020, se ele tivesse apenas o fechamento para se guiar?

6.5. Um desafio para os Modelos Lineares

Um novo treino foi feito com os modelos lineares, no caso, os que tiveram melhores comportamentos, sendo eles: Regressão Linear e Support Vector Regression.

Como eles tiveram um comportamento extremamente bom com o dataset com uma grande período (de 2015 a 2020), um novo desafio foi proposto para verificar o comportamento: Um dataset menor, com dados de 2019 para frente e a variável preditora seria com base apenas no fechamento do dia (Close) com o objetivo de prever os dados do final de janeiro até o começo de junho (período inicial da pandemia e onde mais houveram quedas na bolsa de valores), ou seja, prever 90 dias. Como a Bolsa de Valores funciona somente em dias úteis esses 90 dias deram exatamente do final de janeiro até o começo de junho (último registro do dataset em 01/06/2020).

6.5.1 Preparação dos Modelos

Para esse estudo a mesma lógica para consumir e tratar os dados foi utilizada, a única diferença foi que nesse treino não foi utilizada a variável de abertura como variável independente (x), e sim o próprio fechamento, essa definição foi feita da seguinte forma:

Foi definida uma variável com a quantidade de dias que seriam previstos (90 dias), com base nisso, foi feita uma cópia do fechamento em uma coluna chamada Prediction. Essa coluna tem todos os dados que têm no fechamento, exceto os último 90 dias.

```
# Quantos dias queremos prever para a predição
forecast_in_days = 90;

#criar uma nova coluna (alvo/variavel dependente Y)
df['Prediction'] = df[['Close']].shift(-forecast_in_days)
```

Após isso definimos o x e y para treinar o modelo. O x recebe os dados de fechamento do IBOVESPA originais, sem os últimos 90 dias e o y, a mesma lógica, ele vai receber a coluna Prediction (ela tem exatamente os dados do fechamento, porém sem os últimos 90 dias), os dados foram separados em treino e teste para aplicar os treinos.

```
# criar a variavel dependente (X)
X = np.array(df.drop(['Prediction'], axis=1))

# remover os ultimos 'forecast_in_days' do array
X = X[:-forecast_in_days]

# criar a variavel dependente (Y)
y = np.array(df['Prediction'])

# remover os ultimos 'forecast_in_days' do array
y = y[:-forecast_in_days]

# treino do modelo
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=1)
```

Com essa separação, foi possível treinar os modelos.

```
# Linear Regression
lr = LinearRegression()
lr.fit(X_train, y_train)

# Support Vector Regressor
svr_linear = SVR(kernel='linear')
svr_linear.fit(X_train, y_train)
```

Agora que os modelos já foram treinados, podemos tentar prever os 90 dias que não foram treinados, para isso foi criado um array com esse período e o fechamento correspondente a ele. Por fim, feita uma predição com esses dados, conforme abaixo.

```
# Criação da variável com apenas os últimos 90 dias
x_forecast = np.array(df.drop(['Prediction'], 1))[-forecast_in_days: ]

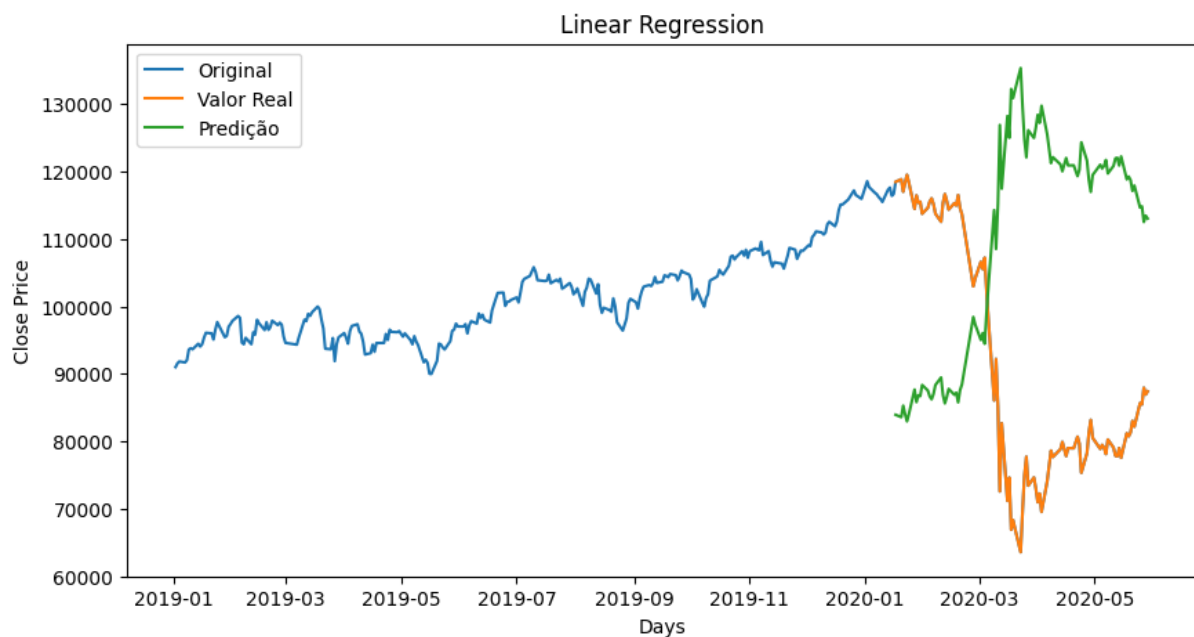
# Linear Regression
lr_pred = lr.predict(x_forecast)

# Support Vector Regressor
svr_linear_pred = svr_linear.predict(x_forecast)
```

Com esses dados podemos verificar os resultados. O primeiro resultado a ser analisado é o da Regressão Linear.

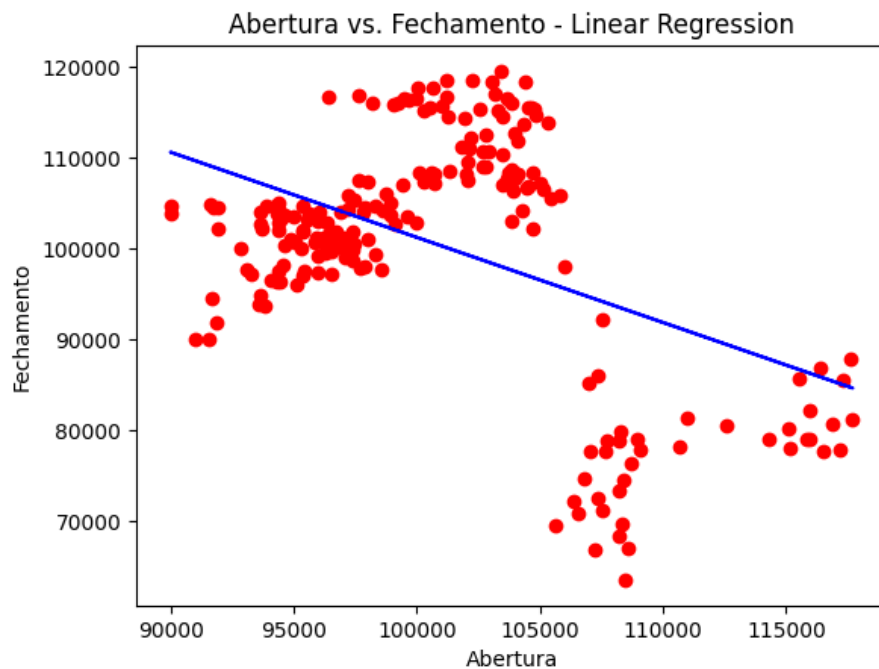
6.5.2 Regressão Linear Simples

Para a Regressão Linear, nesse cenário, como podemos ver abaixo o resultado foi bem diferente do esperado, seguindo a tendência do modelo, ele previu um cenário otimista e com base nos dados de 2019 o modelo sugeriu um crescimento, sem contar com os efeitos da pandemia.



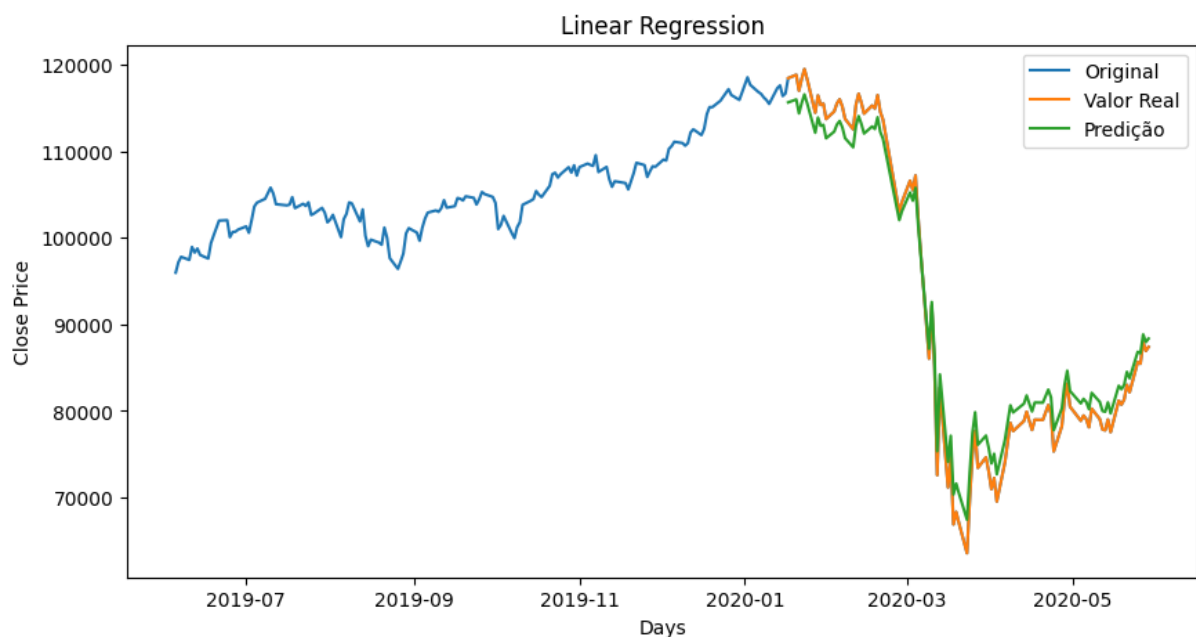
Com o gráfico de dispersão, podemos perceber exatamente isso, uma correlação negativa forte, onde os valores caíram fortemente no cenário real e o modelo

previu um grande crescimento, acertando algumas poucas previsões nas intersecções.

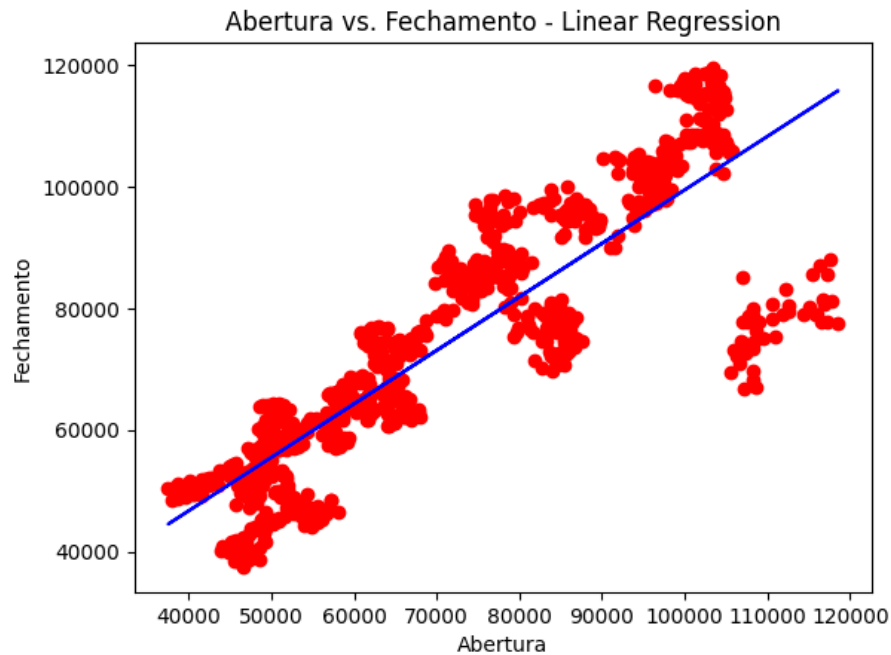


Com isso, ao calcular o coeficiente de Determinação, chegamos no valor de **R²: 0.3764**, ou seja, consideravelmente baixo, não sendo um modelo muito confiável nesse cenário.

Quando utilizamos mais dados para treinar o modelo, novamente de janeiro de 2015 até junho de 2020, porém com a mesma lógica (prever com base apenas no fechamento), o modelo melhora bastante, porém não é tão assertivo quanto quando ele é treinado com a variável de Abertura, conforme podemos ver:



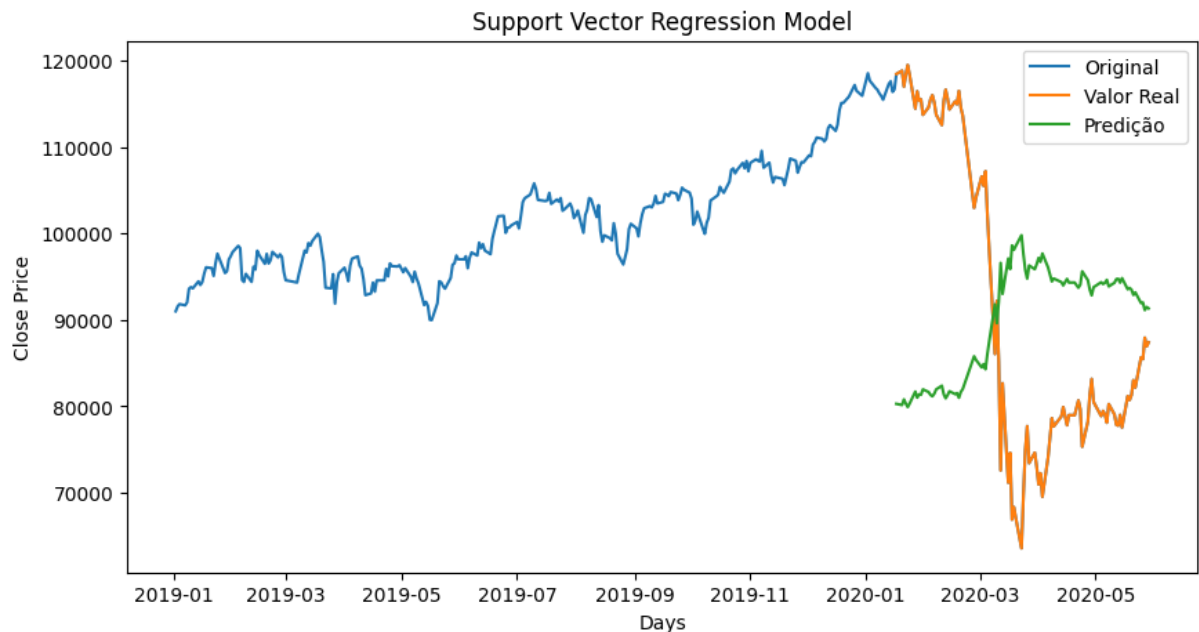
O modelo acerta muito mais e se comporta bem melhor, porém não tanto como no modelo com a Abertura como variável independente e preditora. O gráfico de dispersão mostra a melhora do comportamento, trazendo uma correlação positiva média, onde ele tende a seguir a mesma direção, porém com uma assertividade não tão alta.



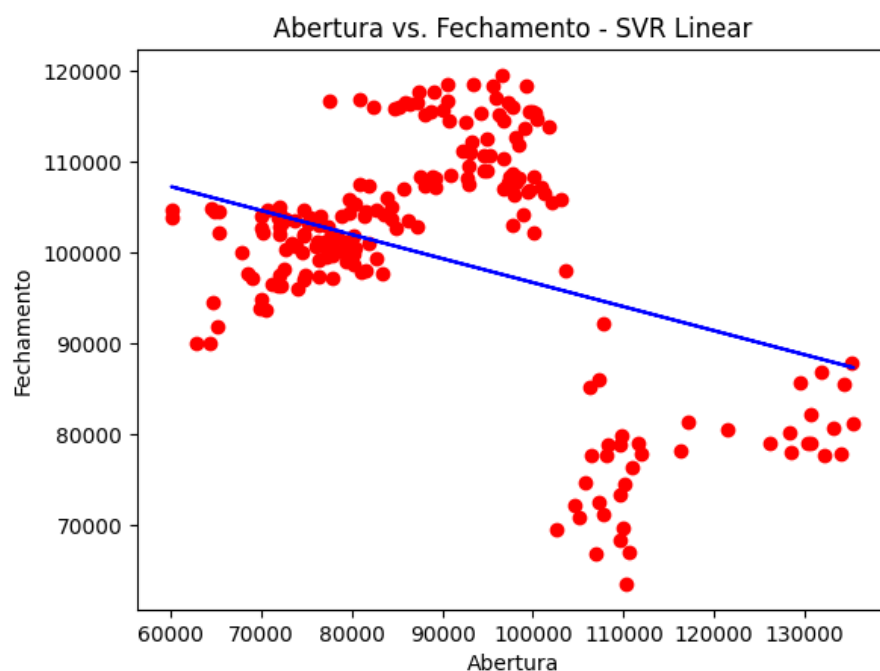
Com isso, ao calcular o coeficiente de Determinação, chegamos no valor de **R²: 0.7248**, O modelo melhorou muito comparado com a tentativa com um dataset menor, mas ainda assim ele se comportou muito melhor com uma variável preditora diferente.

6.5.3 Regressão Linear de Vetor Suporte (Linear SVR)

Para a Regressão Linear de Vetor Suporte, o comportamento seguiu bastante os resultados da Regressão Linear Simples, seguindo a tendência do modelo, ele previu um cenário bem diferente, sem grandes quedas e sem grandes altas.



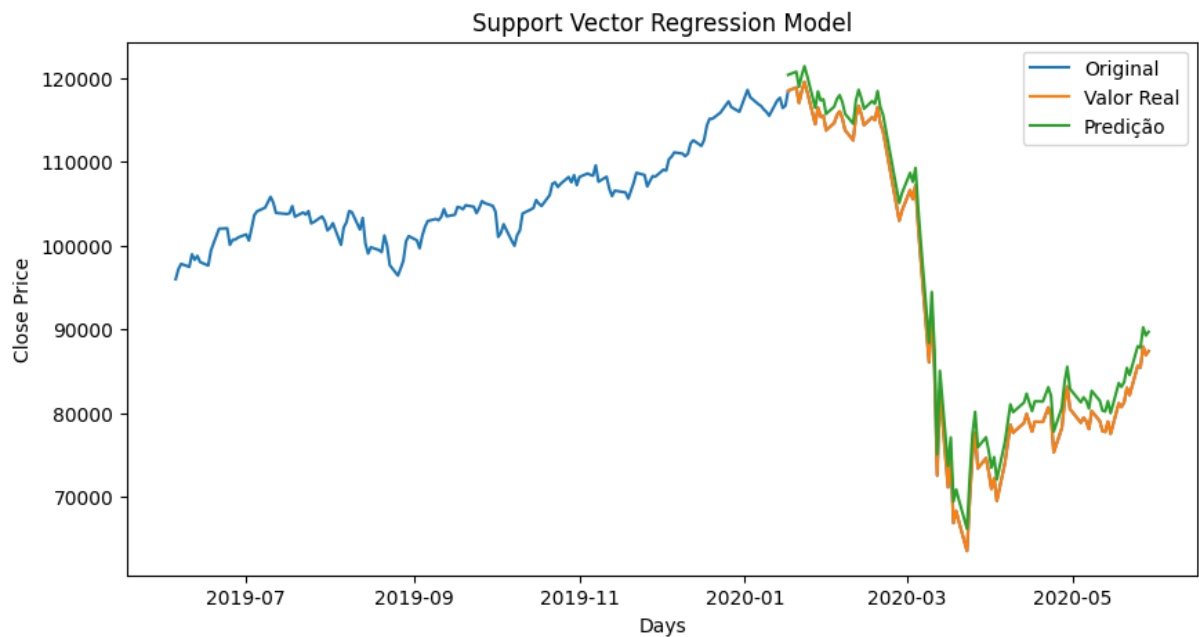
Com o gráfico de dispersão, também podemos perceber isso, uma correlação negativa forte, onde os valores caíram fortemente no cenário real e o modelo previu cenário com movimentações moderadas, e assim como a Regressão linear, acertou algumas poucas previsões nas intersecções.



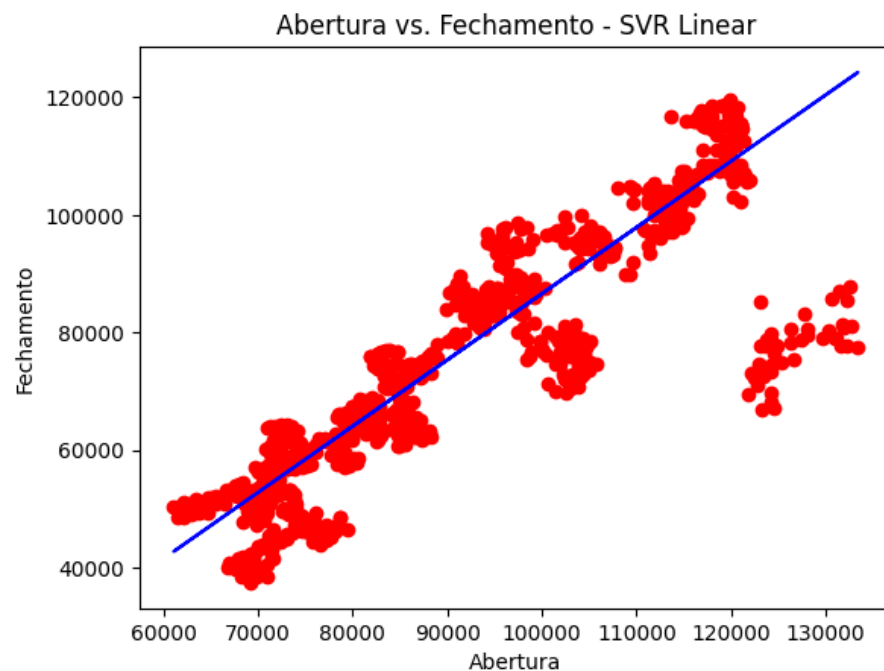
Calculando o coeficiente de Determinação do modelo, chegamos no valor de **R²: 0.3318**, ou seja, menor que o da Regressão Linear, valor também considerado baixo, não sendo um modelo muito confiável nesse cenário.

Justamente por esse motivo, o segundo teste também foi aplicado: mais dados para treinar o modelo (janeiro de 2015 até junho de 2020), porém com a mesma lógica (prever com base apenas no fechamento).

Assim como na Regressão Linear o modelo melhora bastante, porém não é tão assertivo quanto quando ele é treinado com a variável de Abertura.



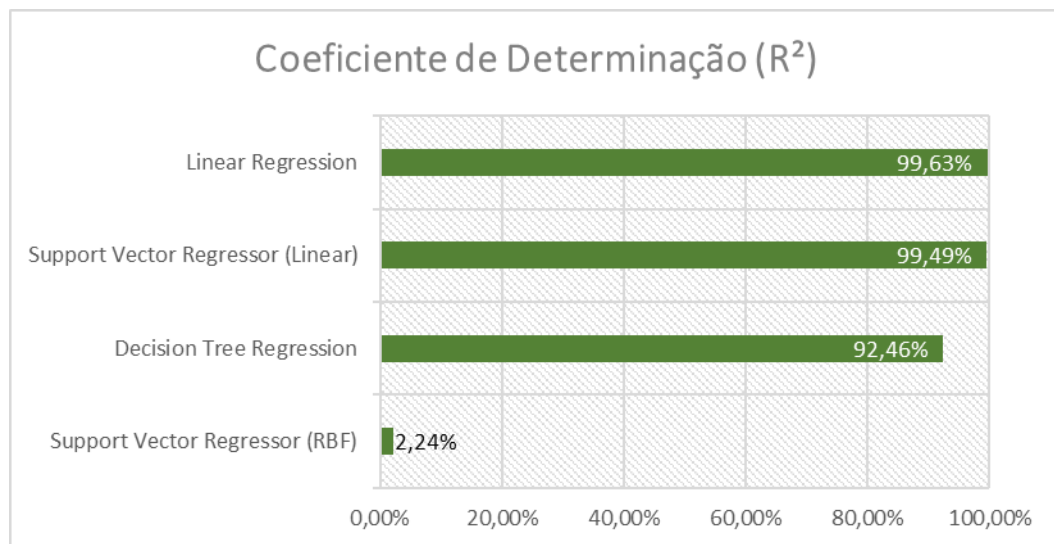
Podemos perceber que o modelo agora apresenta uma correlação positiva.



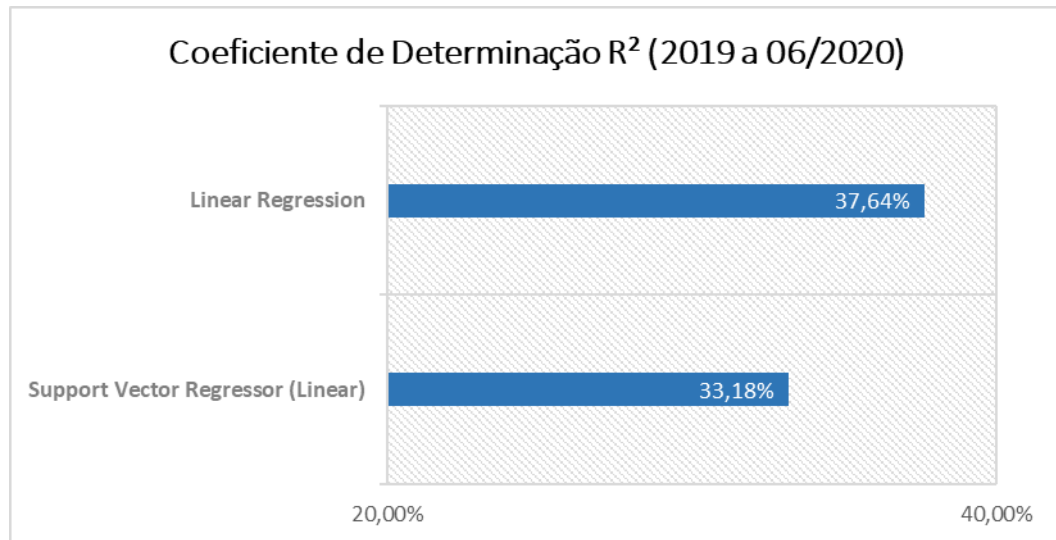
Com isso, ao calcular o coeficiente de Determinação, chegamos no valor de **R²: 0.6861**, O modelo melhorou muito comparado com a tentativa com um dataset menor, mas ainda assim ele se comportou muito melhor com uma variável preditora diferente

6.6. Conclusões

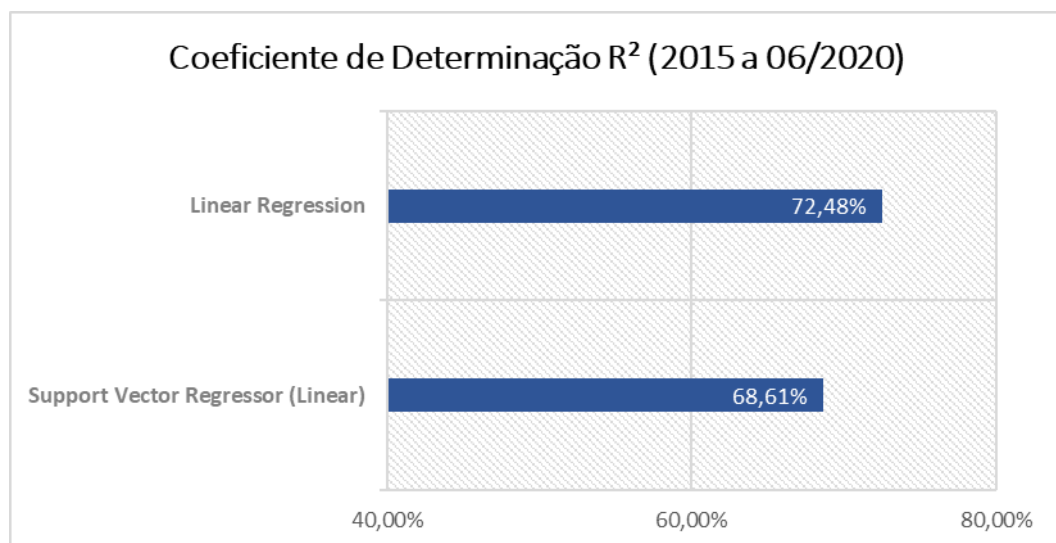
Com um dataset maior e com a variável de abertura podemos ver que todos eles se comportaram muito bem, justamente por ter um norte maior para onde ir. Por isso tivemos esse resultado abaixo. O único que não conseguiu criar a classificação foi o SVR RBF justamente por ele esperar dados não lineares ou a aplicação de uma escala (*Feature Scalling*).



Já quando os modelos lineares são colocados em cenário “mais escuro”, ou seja, com um norte não tão definido, ele apresenta comportamentos bem diferentes, com performance em baixa. Nesse caso o cenário foi: temos poucos dados e apenas uma variável para análise: Abaixo podemos ver o coeficiente de determinação deles, onde a Regressão Linear Simples teve melhores resultados, porém ainda assim bem baixos.



Já quando o cenário tem uma quantidade maior de dados no dataset (de 2015 até 2020), mas ainda assim com apenas uma variável para treino, podemos perceber, conforme abaixo, que o coeficiente de determinação deles tem uma grande melhoria, mas não atinge o mesmo patamar de quando havia uma outra variável para o treino. Novamente a Regressão Linear Simples teve melhores resultados.



Com esse estudo podemos concluir que a melhor forma de utilizar esses modelos é com datasets com grandes volumes e de preferência com uma variável preditora/independente para guiar os modelos ao longo das predições.

Por fim, abaixo podemos ver de forma resumida toda a estrutura do projeto.

ÍNDICE IBOVESPA E BOLSA DE VALORES EM
MEIO A PANDEMIA

RESUMO DO PROJETO

Coleta dos Dados



DATASETS

API Yahoo Finance:

Ações: Índice Ibovespa, Itaú e B3
Período: Jan/2015 a Jun/2020

Kaggle

Dataset COVID-19
Período: Fev/2020 a Jun/2020



Tratamento dos Dados

Alguns tratamentos foram necessários para deixar a base de dados preparada.

- Colunas Renomeadas
- Remoção de dados em Branco
- Dados Filtrados
- Merge dos datasets



Análise e Exploração dos Dados

Diversas análises foram aplicadas para entender os cenários, fazer comparações e levantar diversos insights e considerações para avaliar no processo.



Treinamento dos Modelos de Machine Learning

Os seguintes modelos foram treinados e testados em diversos cenários com dados do índice Ibovespa:

- Linear Regression
- Decision Tree Regression
- Support Vector Regression
 - Kernel RBF e Linear



Conclusões

Com todas as análises feitas e modelos treinados, foi possível entender bem os resultados gerados por todos os modelos, comparar o desempenho deles e ajustar os treinamentos. Além disso também foi adicionado um novo cenário para verificar o comportamento deles, gerando as conclusões necessários para o projeto e qual foi o melhor modelo.

JONATHAN ALVES DE LIMA

7. Links

Repositório Github e Vídeo Resumo:

<https://github.com/jlimadev/pucm-datascience-bigdata-tcc>

<https://www.youtube.com/watch?v=sZcM5psCSo8&feature=youtu.be>

Referências

GARETH, JAMES. **An Introduction to Statistical Learning**. New York: Springer, 2013.

SKLEARN. Disponível em: <<https://scikit-learn.org/stable/index.html>>

Correlação Direto Ao Ponto. Disponível em:
<<https://operdata.com.br/blog/coeficientes-de-correlacao/>>

Gráficos Do Coronavírus China E Estados Unidos. Disponível em:
<<https://www.worldometers.info/coronavirus/>>

Kaggle Covid-19 - Dataset Brasil. Disponível em:
<<https://www.kaggle.com/cprete/covid19-open-datasets-for-brazil>>