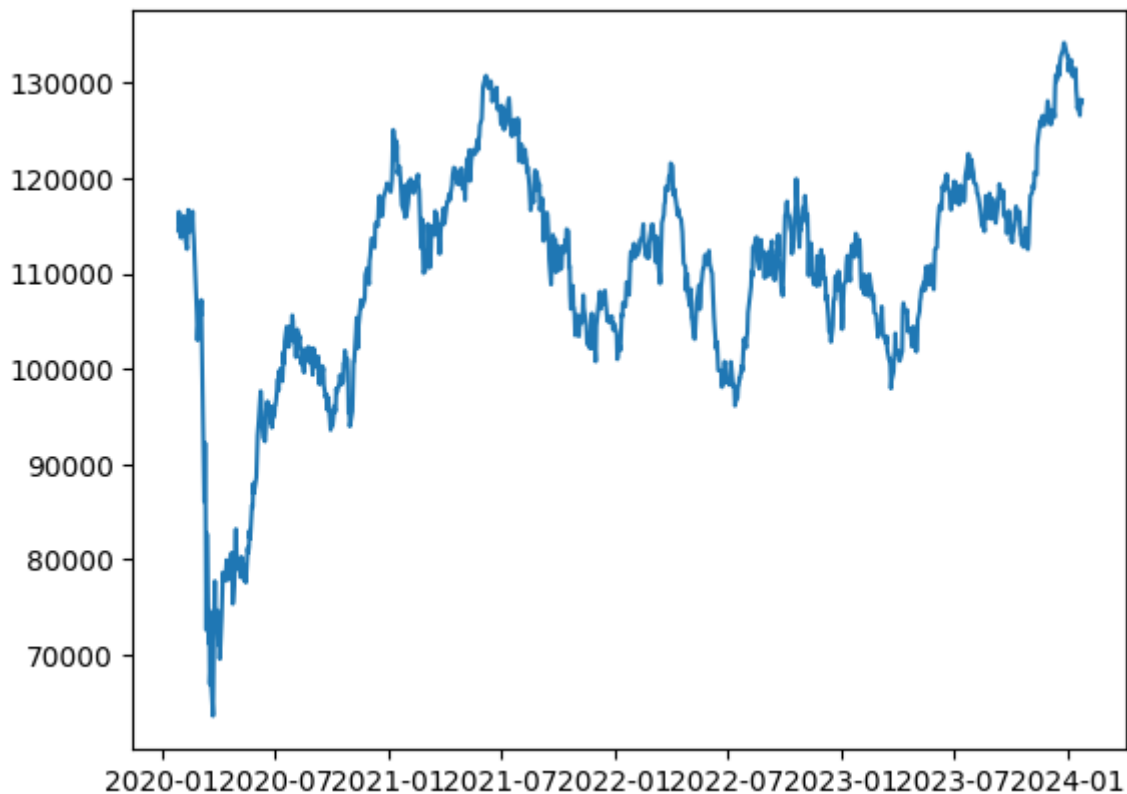


Relatório de série temporal do índice IBOVESPA

	Open	High	Low	Close	Adj Close	Volume
Date						
2020-01-27	118347.0	118347.0	114376.0	114482.0	114482.0	6097800
2020-01-28	114482.0	116797.0	114475.0	116479.0	116479.0	5229400
2020-01-29	116494.0	117171.0	115164.0	115385.0	115385.0	5097800
2020-01-30	115375.0	115528.0	112825.0	115528.0	115528.0	6315600
2020-01-31	115518.0	115518.0	113148.0	113761.0	113761.0	6219900
...
2024-01-19	127319.0	127820.0	126533.0	127636.0	127636.0	11956900
2024-01-22	127636.0	127843.0	125876.0	126602.0	126602.0	9509100
2024-01-23	126612.0	128331.0	126612.0	128263.0	128263.0	9366100
2024-01-24	128275.0	129446.0	127680.0	127816.0	127816.0	8823800
2024-01-25	127815.0	128697.0	127803.0	128169.0	128169.0	8757600

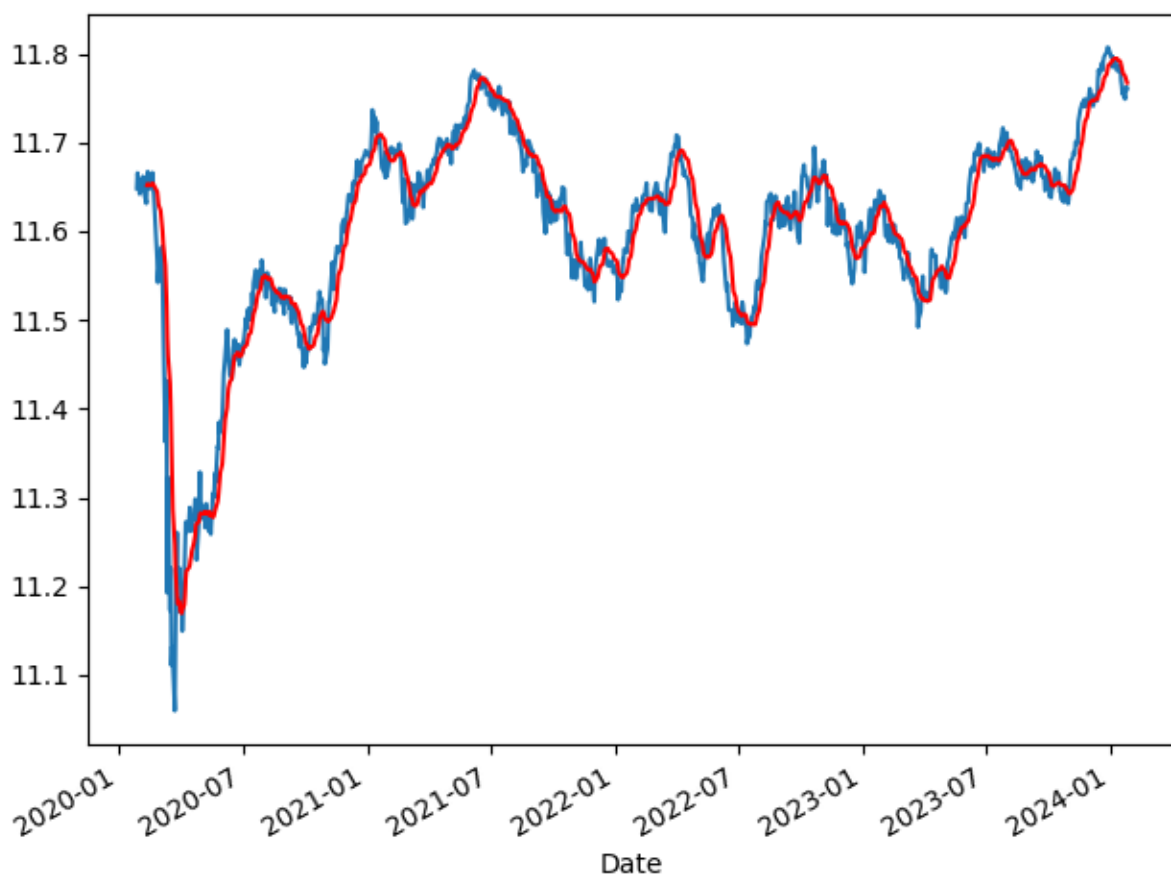
Para a coleta de dados, utilizamos a biblioteca "yfinance" e coletamos dados entre os dias 26 de janeiro de 2020 e 26 de janeiro de 2024 obtendo no total 499 dias válidos. Após a coleta de dados, produzimos uma EDA simples e como variável de análise utilizamos os valores de fechamento do índice!

Plotando um gráfico simples obtemos esse resultado.

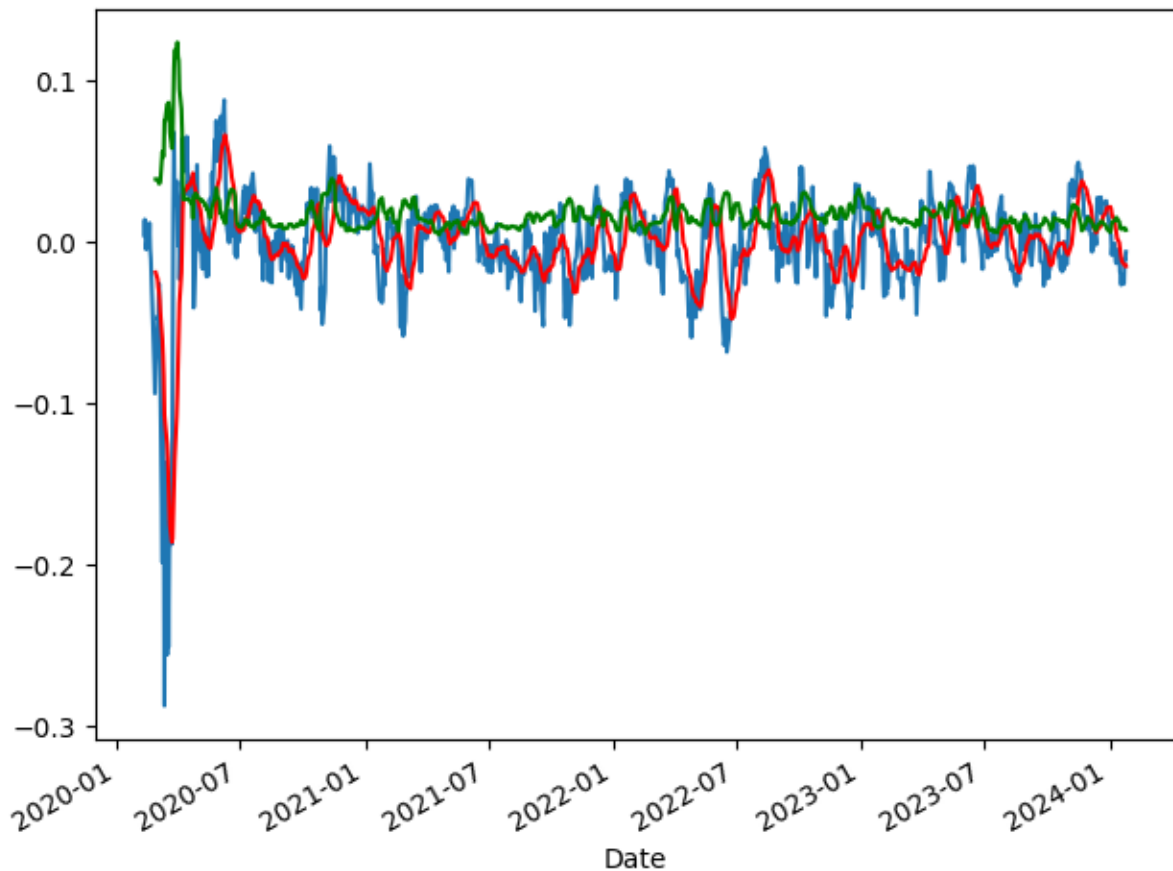


Decompondo a série do período podemos obter a tendência crescente, a sazonalidade (não significativa) e o resíduo demonstra que há muita oscilação, imputando assim uma série não estacionária. Considerando o p-value em 0,05 ou 5% teríamos um nível de confiança em 0,95 ou 95%, entretanto o p-value permaneceu em 0,50 e o teste estatístico maior que os valores críticos provando não ser uma série estacionária!

A série sendo não estacionária, aplicamos o log uma vez e temos o resultado esperado.



Evoluindo na análise, subtraímos a média móvel!

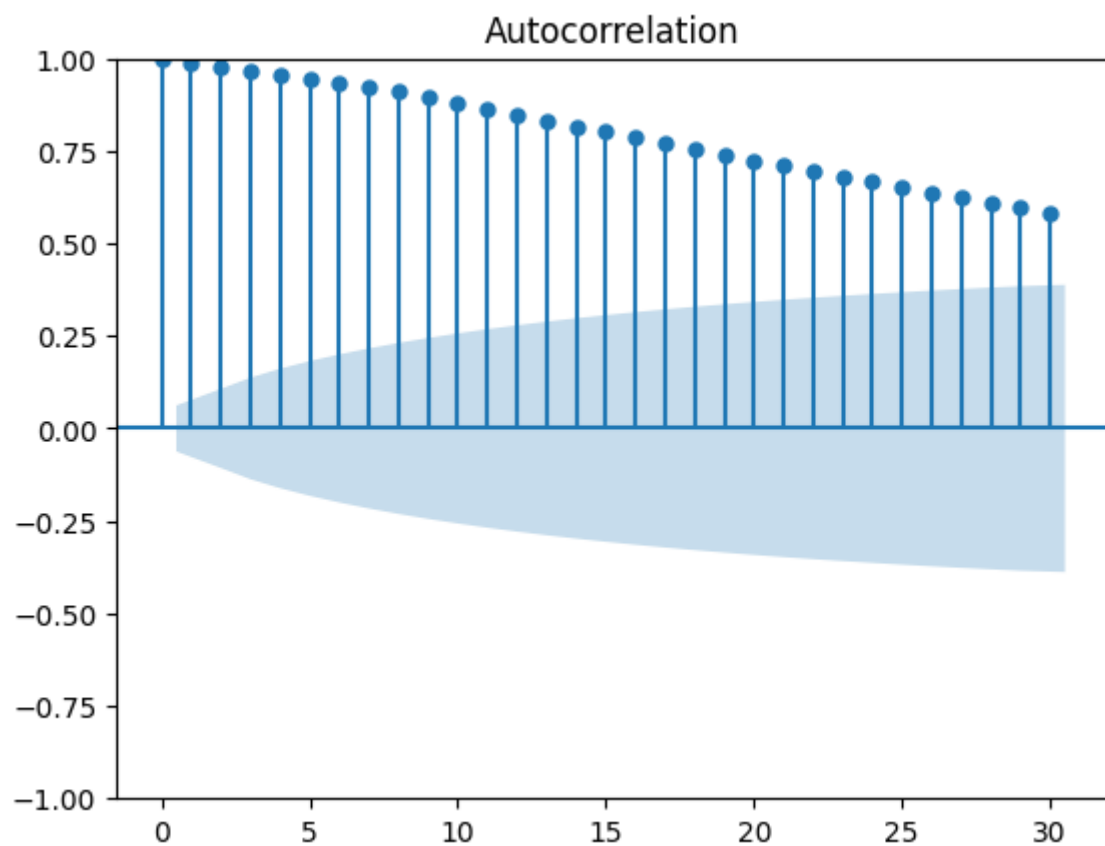


Após a aplicação desses dois métodos, nota-se que o p-value fica muito abaixo de 5% e o Teste estatístico agora se encontra com valores inferiores aos valores críticos, tornando nossa série estacionária. Aplicamos a diferenciação e modificamos a média móvel para o mais constante possível.

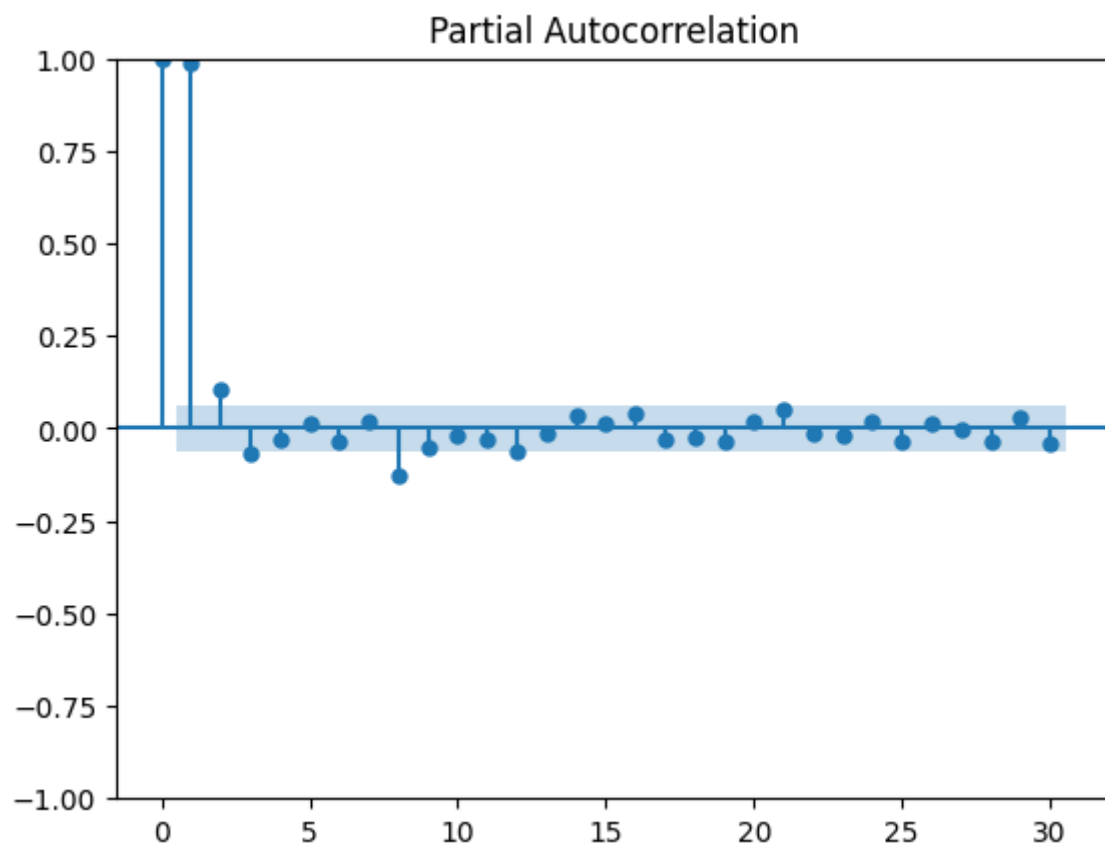
```
Teste ADF
Teste Estatístico: -7.669979455932923
P-Value: 1.6032298445159105e-11
Valores críticos:
1%: -3.43707471791689
5%: -2.8645089073679597
10%: -2.5683508010519396
```

Diferenciação feita, podemos observar um p-value tendendo a 0.0 e um nível de confiança em quase 1.00 ou 100%.

Na correlação de dados foram definidos duas variáveis onde aplicaríamos as funções ACF e PACF, além de considerarmos a autocorrelação. O que fica muito claro, é que a autocorrelação nos dados de ACF, tem grande grau de impacto nos dados, como podemos ver por grande parte dos pontos fora do cone azul do plot de autocorrelação.



Já na autocorrelação parcial, grande parte dos dados antecessores estão no nível de confiança e não representam impactos da correlação parcial.



Criamos um modelo usando a metodologia ARIMA. Foram verificados todos os pontos nulos, além de realizado o tratamento dos mesmos. Para realizar a validação do modelo, utilizamos a métrica MAPE, onde foram feitas as previsões dos dados. Embora obtivemos um resultado de MAPE de 1158%, desconsideração a utilização da metodologia para o modelo.

```
predictions = resultado_AR.fittedvalues

predictions.index = df_diff.index

predicted_values = df_ajustado_log['Close'].iloc[0] + np.cumsum(predictions)

mape = mean_absolute_error(df_diff['Close'], predicted_values) * 100

print(f"MAPE: {mape:.2f}%")

MAPE: 1148.65%
```

Com o resultado tão alto do MAPE indica que as previsões do ARIMA não foram eficientes.

Coletamos os dados novamente com a mesma biblioteca para utilizá-los no prophet, após a coleta de dados é necessário renomear as colunas para que as bibliotecas e métodos funcionem sem erros.

```
#Coletando os dados novamente

symbol = '^BVSP'
start_date = '2020-01-26'
end_date = '2024-01-26'

df = yf.download(symbol, start=start_date, end=end_date)
df = df.reset_index('Date')
df['Date'] = pd.to_datetime(df['Date']) #realizando a conversão da data para formato datetime
df.drop(columns=['Open', 'High', 'Low', 'Volume', 'Adj Close'], inplace=True)
# Renomeando as colunas para 'ds' e 'y'
df = df.rename(columns={'Date': 'ds', 'Close': 'y'})
#df[['ds','y']] = df[['Date','Close']]
df.head()
```

Dividimos os dados em treinamento (70%), validação (15%) e teste (15%), ficando 349 linhas para treinamento e 76 para teste.

```
#train_data = df.sample(frac=0.8, random_state=0)
#test_data = df.drop(train_data.index)

# Dividindo os dados em treinamento, validação e teste
train_size = int(len(df) * 0.7)
val_size = int(len(df) * 0.15)
test_size = len(df) - train_size - val_size

train_data, test_and_val_data = np.split(df, [train_size])
val_data, test_data = np.split(test_and_val_data, [val_size])

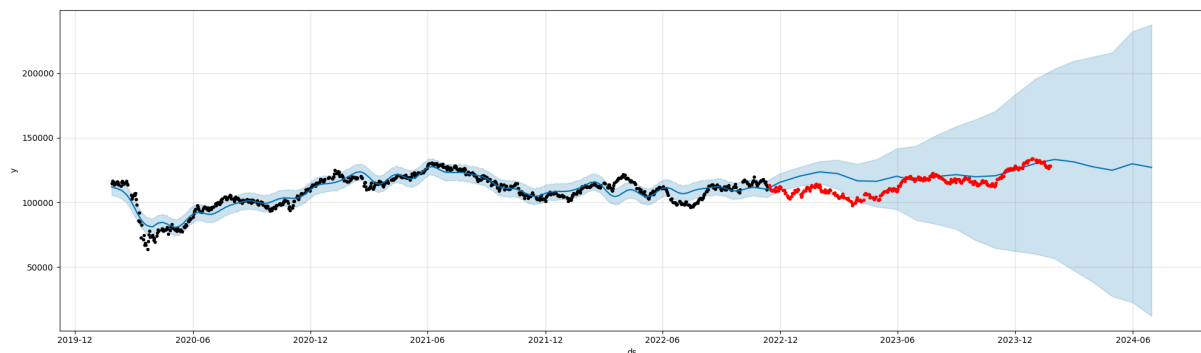
print(f'training data size : {train_data.shape}')
print(f'testing data size : {test_data.shape}')

training data size : (695, 2)
testing data size : (150, 2)
```

Após dividirmos e validarmos a divisão pode ser aplicado o prophet na série temporal, resultando o y chapel (previsão do modelo) y chapel maior e menor (margem de erro do modelo).

```
modelo = Prophet(daily_seasonality=True)
modelo.fit(train_data)
dataFramefuture = modelo.make_future_dataframe(periods=20, freq='M')
previsao = modelo.predict(dataFramefuture)
previsao.head()
```

Para medirmos a acurácia (nível de assertividade) do modelos foi utilizado o indice MAPE que retorna o erro percentual absoluto médio, ou seja, a chance que nosso modelo tem de errar!



O MAPE de 7,01% implica um percentual mínimo de erro, traduzindo um nível altíssimo de acurácia.


```

previsao_cols = ['ds', 'yhat']
valores_reais_cols = ['ds', 'y']

previsao = previsao[previsao_cols]
valores_reais = test_and_val_data[valores_reais_cols]

# Mesclar os DataFrames nas colunas 'ds' para comparar previsões e valores reais
resultados = pd.merge(previsao, valores_reais, on='ds', how='inner')

# Calcular o erro percentual absoluto para cada ponto de dados
resultados['erro_percentual_absoluto'] = np.abs((resultados['y'] - resultados['yhat']) / resultados['y']) * 100

# Calcular o MAPE
mape = np.mean(resultados['erro_percentual_absoluto'])

print(f"MAPE: {mape:.2f}%")

```

MAPE: 7.01%

Conclui-se que para esse projeto o melhor modelo de análise preditiva e as melhores informações devem vir do prophet!

	ds	yhat	y	erro_percentual_absoluto
0	2022-11-30	115539.085834	112486.0	2.714192
1	2023-01-31	123668.259230	113532.0	8.928108
2	2023-02-28	122341.848358	104932.0	16.591553
3	2023-03-31	116701.909841	101882.0	14.546151
4	2023-05-31	120266.942142	108335.0	11.013931
5	2023-06-30	116555.938433	118087.0	1.296554
6	2023-07-31	120050.229017	121943.0	1.552177
7	2023-08-31	121493.970216	115742.0	4.969648
8	2023-10-31	120633.048893	113144.0	6.619042
9	2023-11-30	124953.561364	127331.0	1.867133