

Pós Tech - Data Analysis
Turma 7DTAT

Tech Challenge #2 - Machine Learning and Time Series

Daniela Meneghello - RM: 356004

Maurício José de Lima - RM: 358817

Pedro Vitor da Silva Pinto - RM: 358876

1. Descrição do Problema

- Imagine que você foi escalado para um time de investimentos e precisará realizar um modelo preditivo com dados da IBOVESPA (Bolsa de valores) para criar uma série temporal e prever diariamente o fechamento da base. Para isso, utilize a base de dados contida no site da Investing (<https://br.investing.com/indices/bovespa-historical-data>) e selecione o período “diário”, com o intervalo de tempo que achar adequado.
- Você precisará demonstrar para o time de investimentos:
 1. O modelo com o storytelling, desde a captura do dado até a entrega do modelo;
 2. Justificar a técnica utilizada;
 3. Atingir uma acuracidade adequada (acima de 70%).

2. Análise Exploratória dos Dados

- Realizamos o download do arquivo .csv do site Investing, conforme informado, contendo os dados de fechamento da Ibovespa ente 2005 e 2024.

```
# Leitura do arquivo ** mostrar de onde veio o dado - 2005 até final de 2024  
  
df = pd.read_csv('https://raw.githubusercontent.com/jdlmauricio/techalleg_fase_2/refs/heads/main/Dados%20Hist%C3%B3ricos%20-%20Ibovespa.csv')
```

	Data	Último	Abertura	Máxima	Mínima	Vol.	Var%
0	30.12.2024	120.283	120.267	121.050	120.158	8,90M	0,01%
1	27.12.2024	120.269	121.078	121.609	120.252	8,94M	-0,67%
2	26.12.2024	121.078	120.767	121.612	120.428	8,34M	0,26%
3	23.12.2024	120.767	122.105	122.105	120.617	9,95M	-1,09%
4	20.12.2024	122.102	121.183	122.209	120.700	18,13M	0,75%

2. Análise Exploratória dos Dados

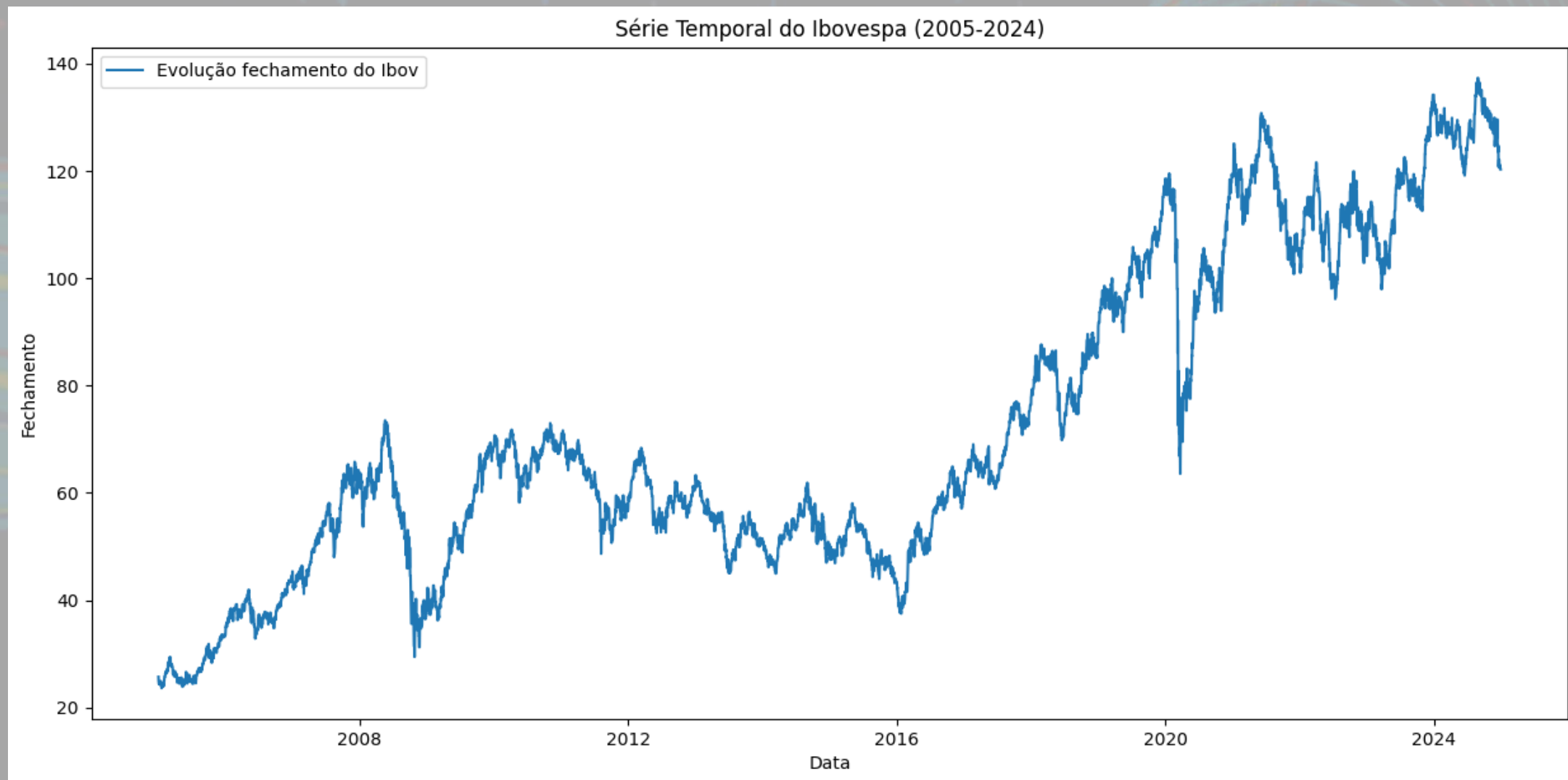
- Realizamos então a transformação dos dados, como exclusão de colunas desnecessárias, renomeação de colunas, e conversões de tipos de dados, para os padrões que utilizaremos em nossas análises:
- Criamos a série completa de dados e preenchemos valores ausentes com o valor do dia anterior:

```
# Remove colunas desnecessárias ***
df = df.drop(columns=['Abertura', 'Máxima', 'Mínima', 'Vol.', 'Var%'])
# Renomeia colunas
df = df.rename(columns={'Data': 'ds', 'Último': 'y'})
# Converte coluna de data
df['ds'] = pd.to_datetime(df['ds'], format='%d.%m.%Y')
# Colocando a data como index
df = df.set_index('ds')
```

```
# Criando uma série completa com finais de semana e feriados ***
datas_completas = pd.date_range(start=df.index.min(), end=df.index.max(), freq='D')
df = df.reindex(datas_completas)
```

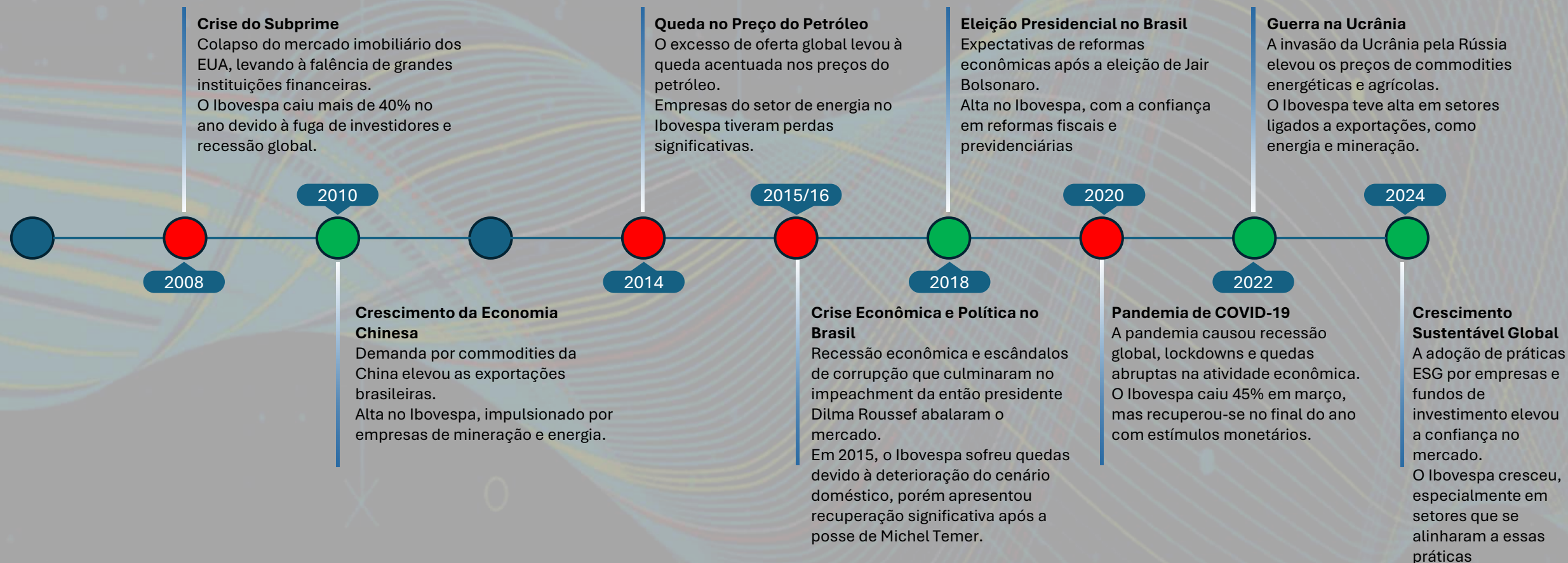
```
## Preenchendo valores ausentes com o último valor conhecido (forward-fill) ***
df['y'] = df['y'].fillna(method='ffill')
```

2. Análise Exploratória dos Dados



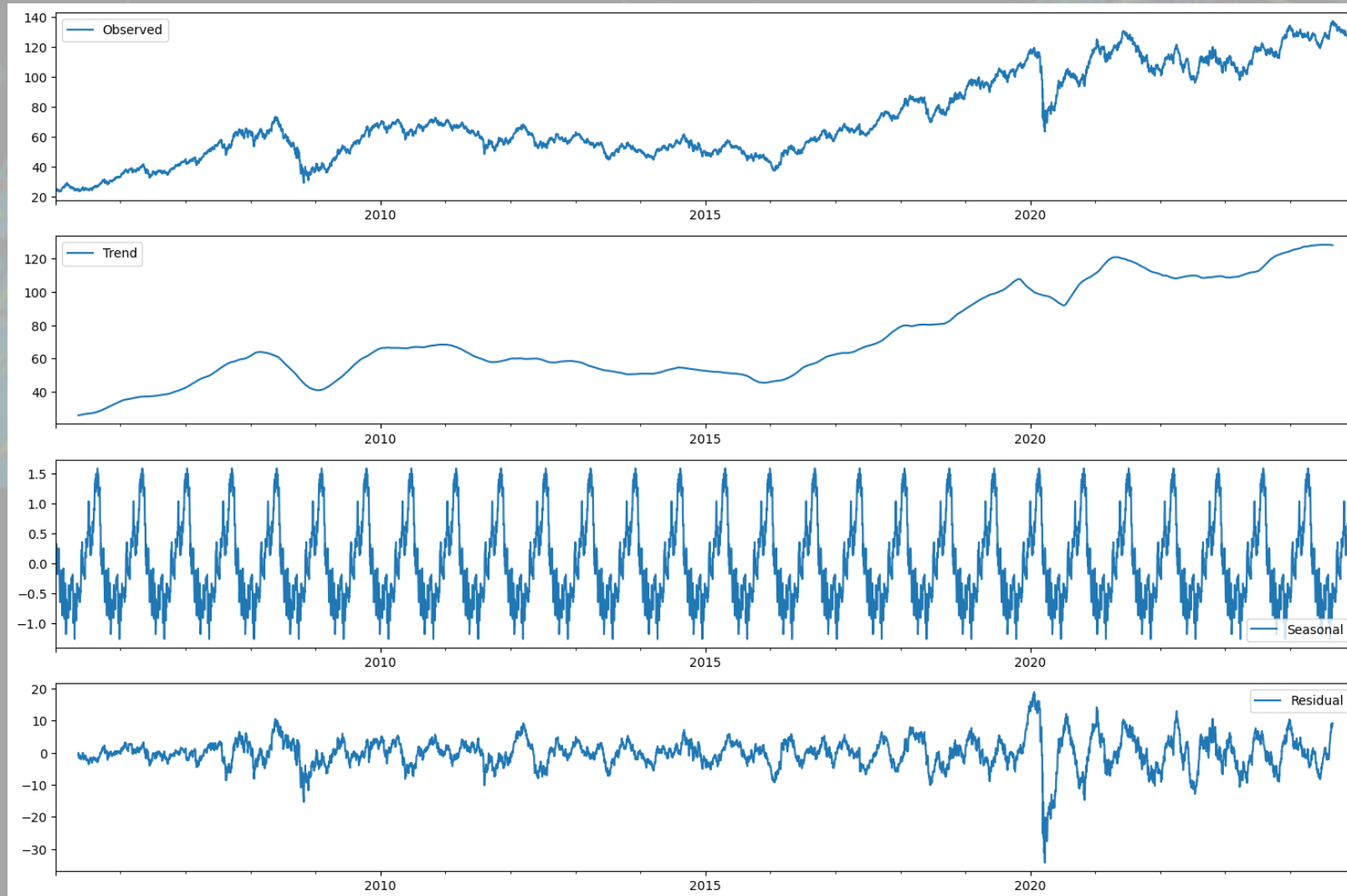
2. Análise Exploratória dos Dados

Linha temporal com fatos relevantes



3. Decomposição da Série Temporal

Decomposição aditiva



4. Modelos

- Inicialmente, realizamos a separação da base de treino e validação;
- Definimos para 15 dias o período de previsão.

```
# Definição de período de Treino e validação ***  
  
treino = df.loc[(df['ds'] >= '2021-01-01') & (df['ds'] < '2024-01-01')]  
|  
valid = df.loc[(df['ds'] >= '2024-01-01') & (df['ds'] < '2024-01-16')]  
  
h = valid['ds'].nunique() # Quantidade de dias a serem previstos
```

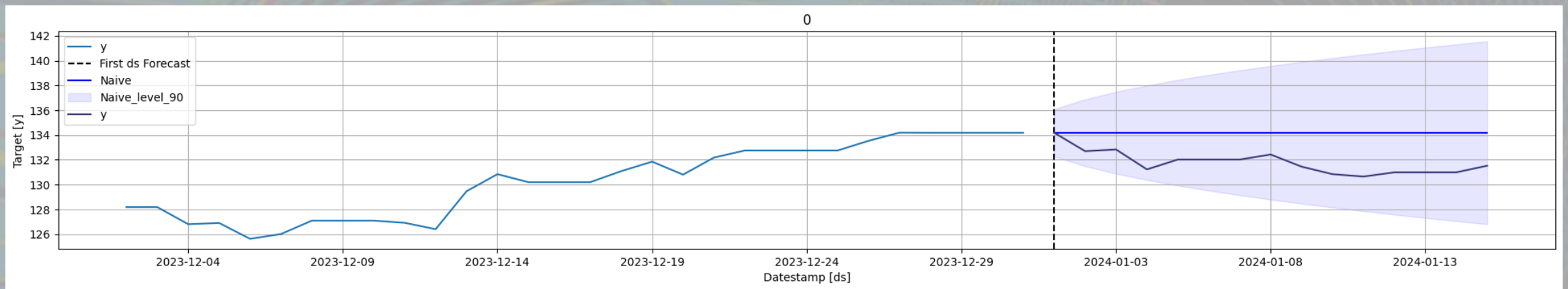
h

15

4. Modelos

Naive

- Realizamos a previsão dos 15 primeiros dias de 2024, utilizando o período de treino para o modelo:



```
MAE Naive Baseline: 2.394331217447918
rmse Naive Baseline: 2.567096227697468
MAPE Naive Baseline: 0.018217591255696986
WMAPE Naive Baseline: 1.82%
MSE Naive Baseline: 6.589983042258569
```

MAE (Mean Absolute Error): Fornece o erro médio absoluto.

RMSE (Root Mean Squared Error): Destaca grandes erros; útil para entender discrepâncias maiores.

MAPE (Mean Absolute Percentage Error): Erro relativo em porcentagem, ajuda na comparação de escalas diferentes.

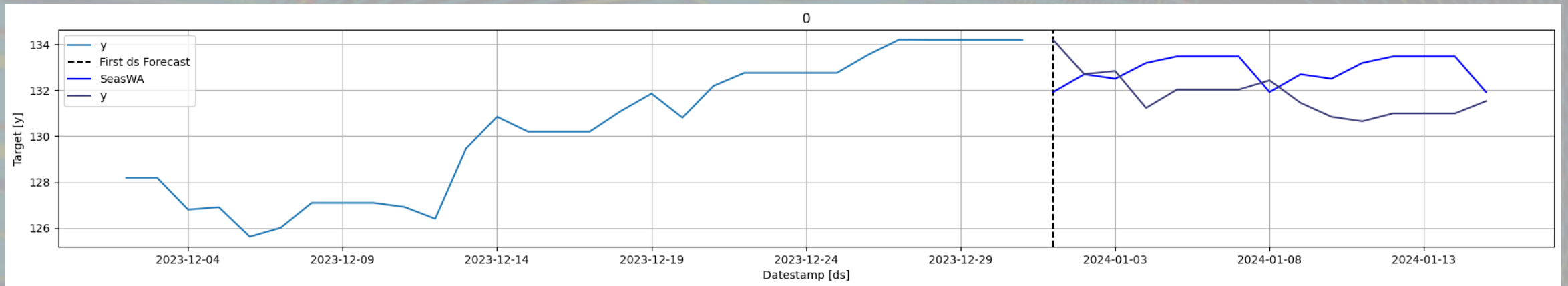
WMAPE (Weighted Mean Absolute Percentage Error): Corrige distorções do MAPE para séries heterogêneas.

MSE (Mean Squared Error): Ideal para análises que priorizam grandes desvios, mas pode ser difícil de interpretar devido à unidade ao quadrado.

4. Modelos

Seasonal Window Average

- Realizamos a previsão dos 15 primeiros dias de 2024, utilizando o período de treino para o modelo:



```
MAE swa : 1.5125644287109385
rmse swa : 1.7312619708939774
MAPE swa : 0.011496800549552691
WMAPE swa : 1.15%
MSE swa : 2.997268011863699
```

MAE (Mean Absolute Error): Indica o erro absoluto médio do modelo sazonal.

RMSE (Root Mean Squared Error): Destaca grandes erros entre as previsões e os valores reais.

MAPE (Mean Absolute Percentage Error): Mede a precisão relativa, sendo sensível a valores pequenos.

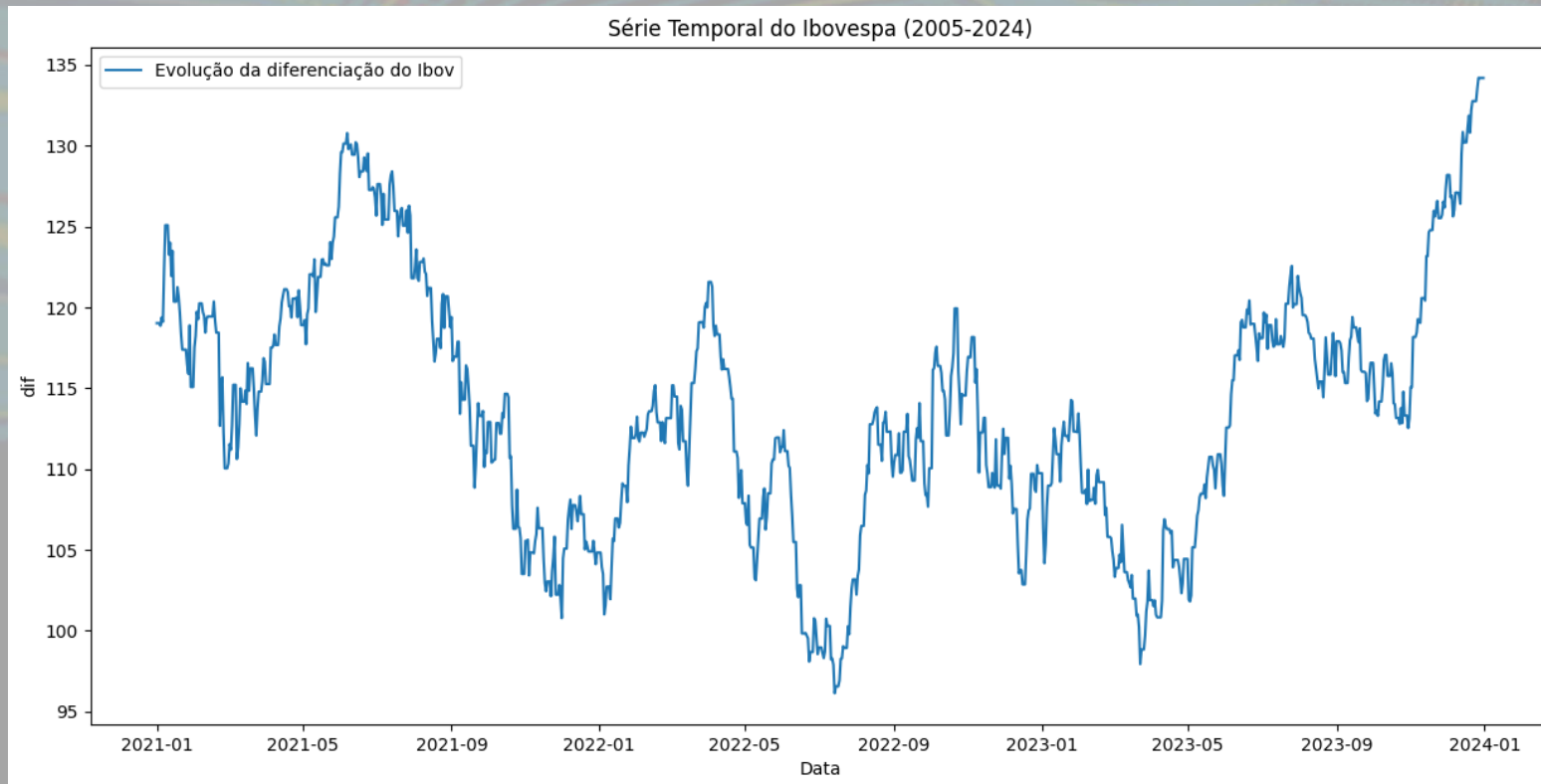
WMAPE (Weighted Mean Absolute Percentage Error): Avalia o erro proporcional com peso, corrigindo distorções do MAPE.

MSE (Mean Squared Error): Útil para analisar a variabilidade dos erros, mas é sensível a outliers.

4. Modelos

Sarima

- Para aplicação do modelo Sarima, realizamos inicialmente a aplicação da derivação na série temporal e o teste de Dickey – Fuller Aumentado (ADF)

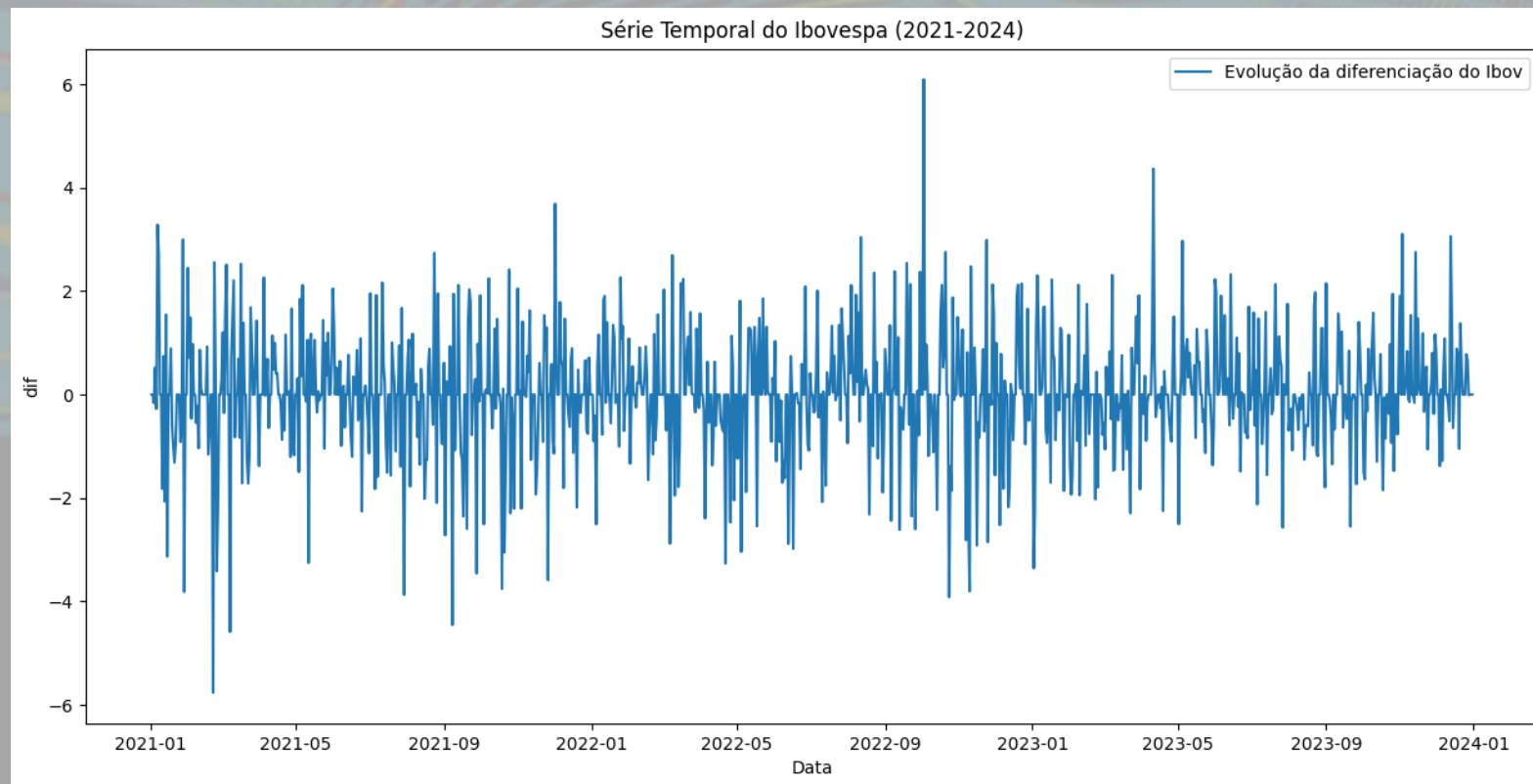


```
Estatística ADF: -1.8202703522041512
Valor-p: 3.704454724514131e-01
Valor Crítico 1%: -3.436341508283391
Valor Crítico 5%: -2.864185524365606
Valor Crítico 10%: -2.5681785627437677
A série não é estacionária.
```


4. Modelos

Sarima

- Aplicamos o método da Diferenciação de Primeira Ordem e novamente o teste ADF obtendo o resultado de que a série é estacionária:



```
Estatística ADF: -34.056002586118545
Valor-p: 0.0000000000000000e+00
Valor Crítico 1%: -3.4363470029475525
Valor Crítico 5%: -2.864187948086107
Valor Crítico 10%: -2.568179853605536
A série é estacionária.
```

4. Modelos

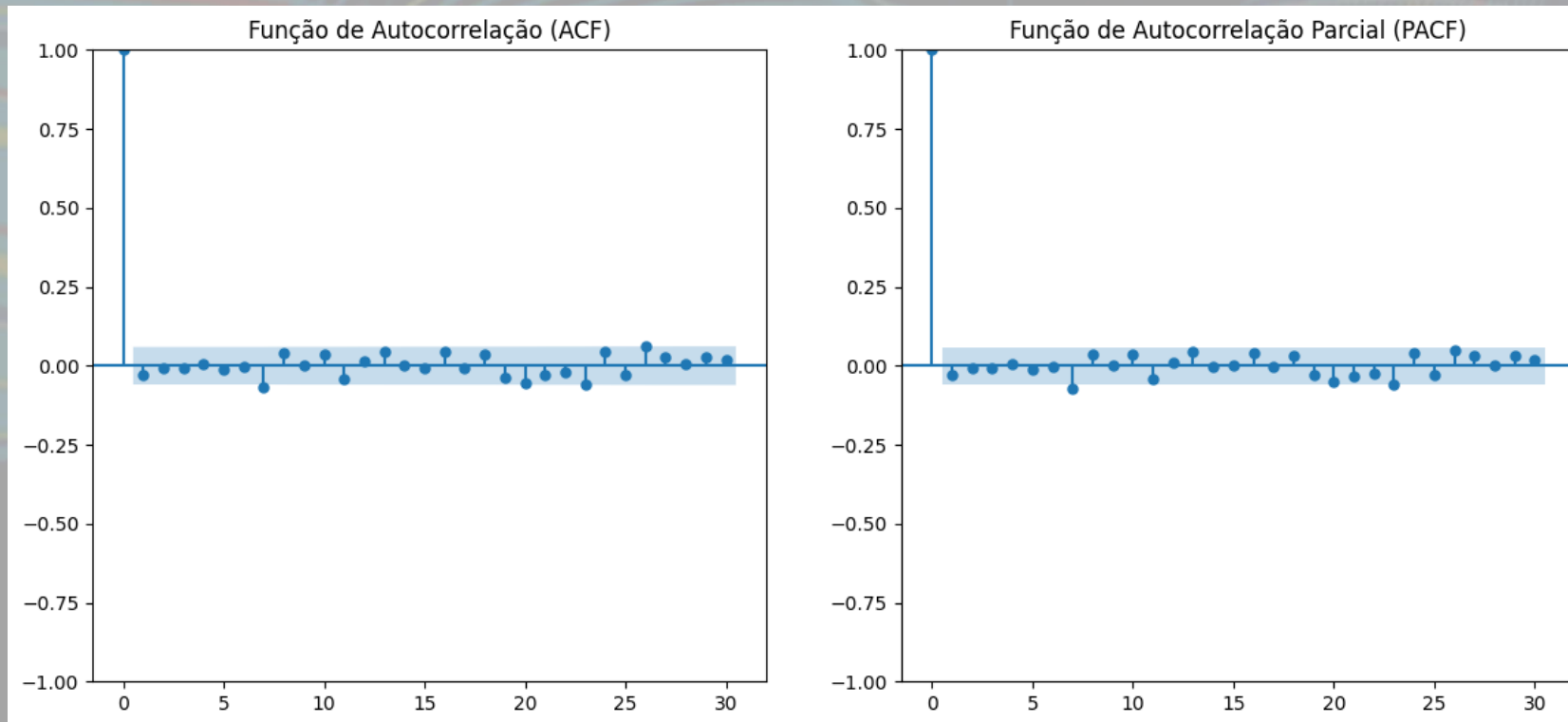
Sarima

- Realizamos a identificação dos Parâmetros para o Modelo – ACF e PACF
- ACF (Autocorrelation Function): Mede a correlação entre a série e suas defasagens, considerando efeitos diretos e indiretos. Ajuda a identificar o parâmetro MA (q) em modelos SARIMA.
- PACF (Partial Autocorrelation Function): Mede a correlação direta entre a série e uma defasagem específica, eliminando influências intermediárias. Ajuda a definir o parâmetro AR (p) no modelo.

4. Modelos

Sarima

- ACF e PACF



```
p = 1 # componente AR ***  
d = 1 # Componente I  
q = 1 # Componente MA
```


Sarima

- Para a aplicação do modelo Sarima, assumimos uma sazonalidade de 21 dias:

```

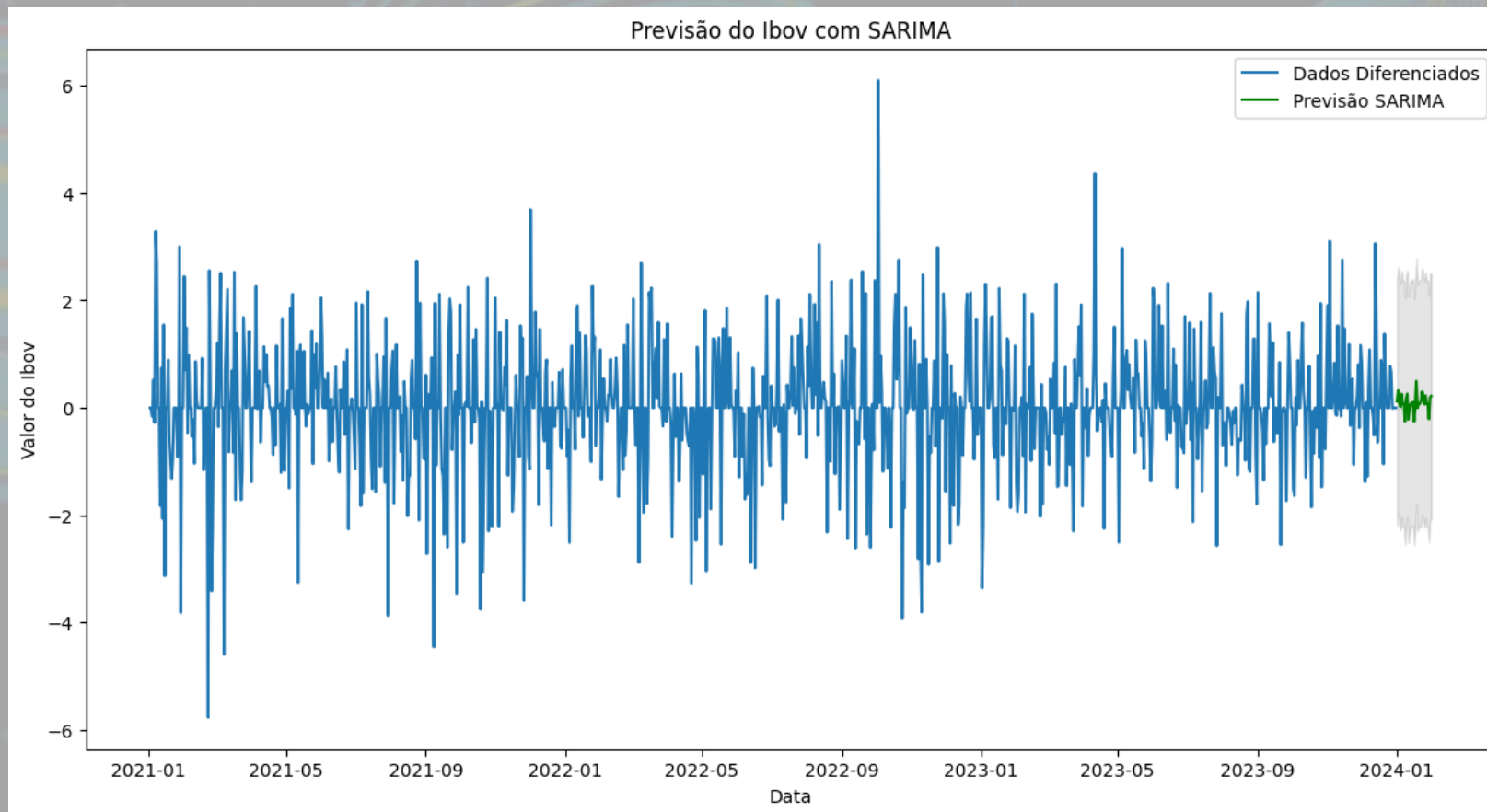
=====
SARIMAX Results
=====
Dep. Variable:          dif      No. Observations:          1094
Model:                  SARIMAX(1, 1, 1)x(1, 1, 1, 21)      Log Likelihood          -1725.821
Date:                  Sun, 19 Jan 2025      AIC          3461.643
Time:                  15:13:52      BIC          3486.529
Sample:                01-02-2021      HQIC          3471.069
                    - 12-31-2023
Covariance Type:                opg
=====
              coef      std err          z      P>|z|      [0.025      0.975]
-----
ar.L1          -0.0330        0.032      -1.041      0.298      -0.095        0.029
ma.L1          -0.9999        0.288      -3.474      0.001      -1.564      -0.436
ar.S.L21       -0.0355        0.025      -1.438      0.150      -0.084        0.013
ma.S.L21       -0.9999        6.675      -0.150      0.881     -14.082     12.082
sigma2          1.3386        8.939        0.150      0.881     -16.181     18.859
=====
Ljung-Box (L1) (Q):                0.00      Jarque-Bera (JB):                223.50
Prob(Q):                0.96      Prob(JB):                0.00
Heteroskedasticity (H):                0.63      Skew:                -0.13
Prob(H) (two-sided):                0.00      Kurtosis:                5.22
=====

```

4. Modelos

Sarima

- Realizamos a previsão dos 15 primeiros dias de 2024 :

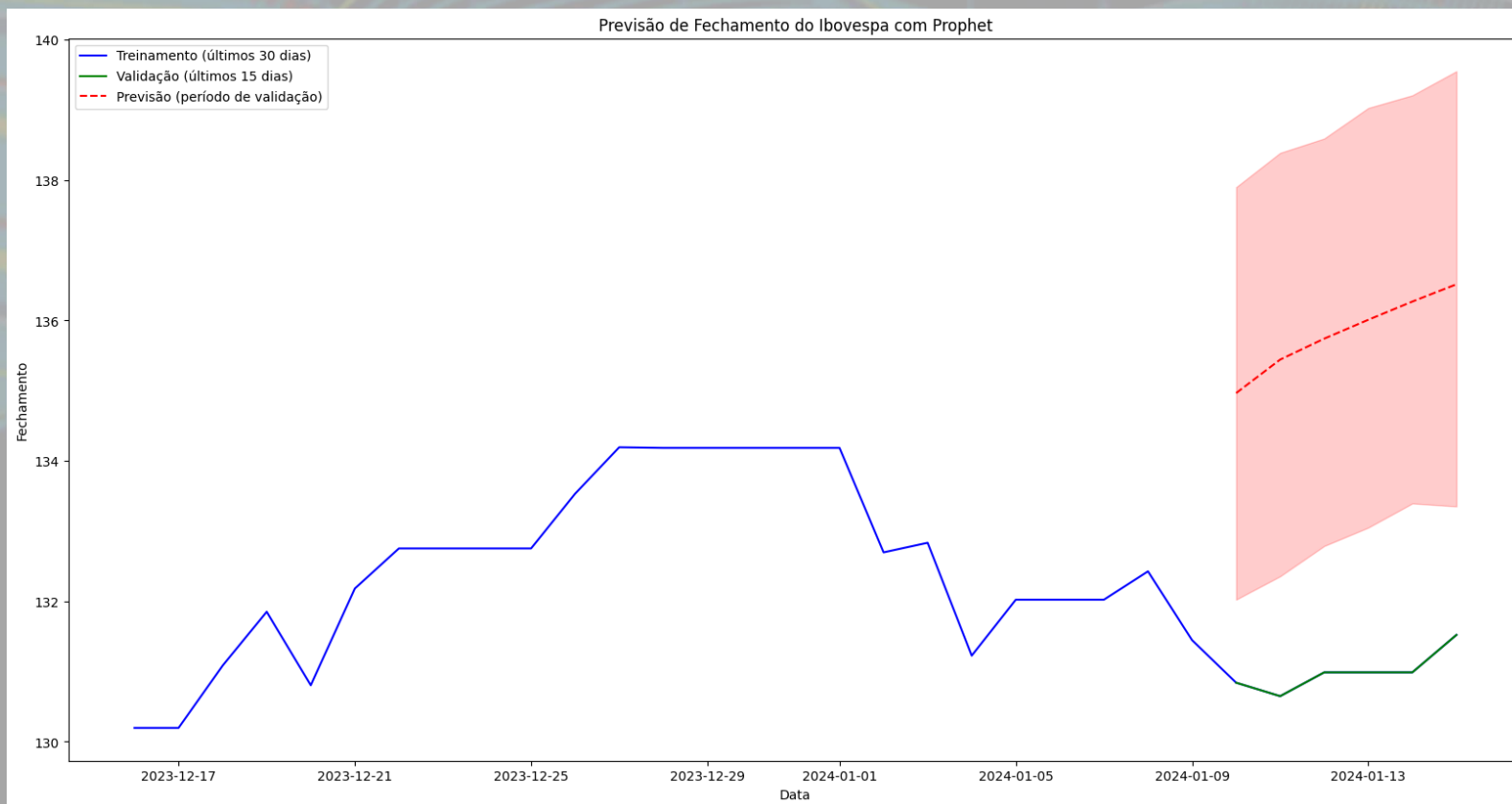


MAE sarima : 131.7123315795186
rmse sarima : 131.71545126367053
MAPE sarima : 0.9994072711825925
MSE sarima : 17348.96010159237

4. Modelos

Prophet

- Realizamos a previsão dos 15 primeiros dias de 2024, utilizando o período de treino entre 01 e 15 de janeiro de 2024 para o modelo:



MAE: 4.828219014817894
RMSE: 4.8415400163985
MAPE: 0.04%
WMAPE: 0.03685780602297448
MSE: 23.440509730387983

8. Conclusão

- Durante a execução do Tech Challenge, realizamos treinos e testes com os modelos Naive, Seasonal Window Average, Sarima e Prophet;
- Realizando uma comparação entre os modelos, observamos os seguintes resultados:

Métrica	Naive	SWA	SARIMA	Prophet
MAE	2.39	1.51	131.71	4.82
RMSE	2.57	1.73	131.71	4.84
MAPE	0.02	0.01	1.00	0.000400
WMAPE	1.82%	1.15%	-	0.037%
MSE	6.59	2,99	17.35	23.44

8. Conclusão

- O modelo Seasonal Window Average foi o que se destacou entre todas as opções avaliadas, demonstrando o melhor desempenho com base nas principais métricas. Ele apresentou o menor erro absoluto médio (MAE: 1.51), o menor erro quadrático médio (RMSE: 1.73) e um erro percentual médio (MAPE) de aproximadamente 1.15%. Esses resultados indicam que suas previsões são mais precisas e também menos variáveis. Além disso, o erro médio quadrático (MSE: 2.99) foi o mais baixo, confirmando sua consistência.
- Em comparação, o Modelo Sarima teve erros excessivamente altos, o que compromete sua confiabilidade, enquanto o Prophet, embora com um MAPE baixo, apresentou erros absolutos maiores, sugerindo menor precisão geral, porém poderia ser uma escolha secundária.
- Com base nessa análise, o Modelo **Seasonal Window Average** demonstrou ser a melhor escolha para o tipo de previsão realizada.

