```
1 import pandas as pd
2 import numpy as np
3 import matplotlib.pyplot as plt
4
5 from statsmodels.tsa.holtwinters import ExponentialSmoothing
6 from sklearn.metrics import mean_squared_error
```

```
1 df_bax_m = pd.read_csv(r'/content/drive/MyDrive/PRN23039142546/df_bax_cleaned_to_view_outliers.csv', index_col=0,parse_dates=True)
2 df_bax_m.head()
```

Date	Price	Open	High	Low	Vol.	Change %
2010-05-24	1482.42	1491.98	1491.98	1482.42	926980.0	-0.64
2010-05-25	1454.85	1482.42	1482.42	1454.85	1660000.0	-1.86
2010-05-26	1472.29	1456.50	1472.29	1454.85	1500000.0	1.20
2010-05-27	1453.82	1472.29	1478.07	1453.82	2480000.0	-1.25
2010-05-30	1455.16	1453.82	1462.04	1453.72	5910000.0	0.09

Next steps: Generate code with df_bax_m | View recommended plots | New interactive sheet

```
1 df = df_bax_m.copy()
```

```
1 series = df['Price']
```

```
1 # train-test split
2 train_size = int(len(series)*0.8)
3 train, test = series[:train_size],series[train_size:]
```

```
1 model = ExponentialSmoothing(train, trend='add', seasonal='add', seasonal_periods=730,damped_trend=True).fit()
2
```

```
/usr/local/lib/python3.11/dist-packages/statsmodels/tsa/base/tsa_model.py:473: ValueWarning: A date index has been provided, but it
  self._init_dates(dates, freq)
/usr/local/lib/python3.11/dist-packages/statsmodels/tsa/holtwinters/model.py:903: ConvergenceWarning: Optimization failed to converg
  warnings.warn(
```

```
1 preds = model.forecast(len(test))
```

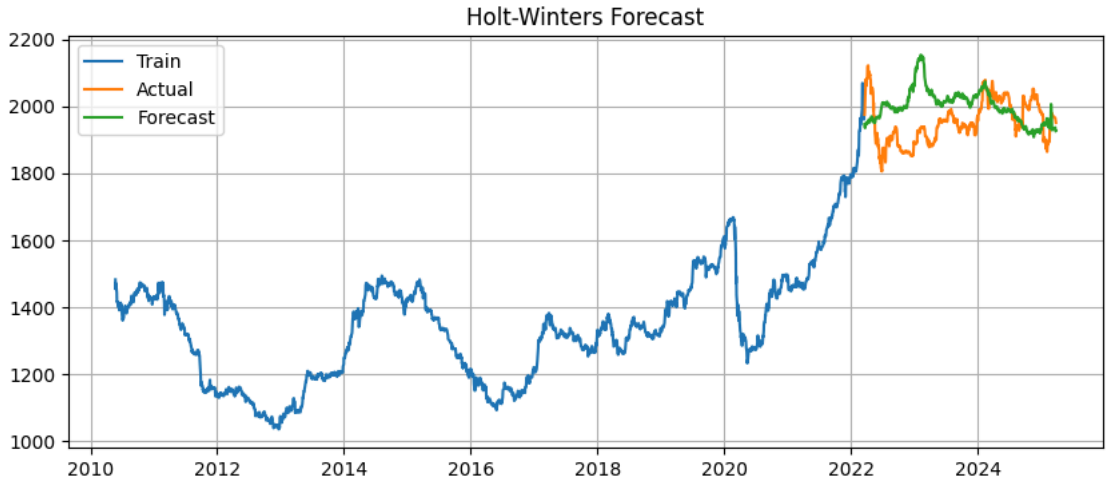
```
/usr/local/lib/python3.11/dist-packages/statsmodels/tsa/base/tsa_model.py:837: ValueWarning: No supported index is available. Predic
  return get_prediction_index(
/usr/local/lib/python3.11/dist-packages/statsmodels/tsa/base/tsa_model.py:837: FutureWarning: No supported index is available. In th
  return get_prediction_index(
```

```
1 # EValuate
2 rmse = np.sqrt(mean_squared_error(test, preds))
3 print(f"Holt-Winters Exponential Smoothing - RMSE: {rmse:.2f}")
```

```
Holt-Winters Exponential Smoothing - RMSE: 102.50
```

```
1 # Plot
2 plt.figure(figsize=(10, 4))
3 plt.plot(train.index, train, label='Train')
4 plt.plot(test.index, test, label='Actual')
5 plt.plot(test.index, preds, label='Forecast')
6 plt.title('Holt-Winters Forecast')
7 plt.legend()
8 plt.grid(True)
9 plt.show()
```

What can I help you build?


Holt-Winters Forecast
Train
Actual
Forecast
2200
2000
1800
1600
1400
1200
1000
2010
2012
2014
2016
2018
2020
2022
2024