```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

from statsmodels.tsa.holtwinters import ExponentialSmoothing
from sklearn.metrics import mean_squared_error
```

```
df_bax_m = pd.read_csv(r'/content/drive/MyDrive/PRN23039142546/Holt-Winters_data.csv', index_col=0,parse_dates=True)
df_bax_m.head()
```

Date	Price	Open	High	Low	Vol.	Change %
2022-03-28	2075.40	2088.99	2088.99	2061.64	1650000.0	-0.24
2022-03-29	2078.53	2075.40	2078.53	2068.92	3700000.0	0.15
2022-03-30	2061.94	2091.40	2091.40	2061.94	4930000.0	-0.80
2022-03-31	2073.54	2061.76	2073.57	2055.86	3760000.0	0.56
2022-04-03	2084.28	2073.54	2084.28	2066.57	4450000.0	0.52

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

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

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

```
model = ExponentialSmoothing(train, trend='add', seasonal='mul', seasonal_periods=272,damped_trend=True).fit()

```

```
/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(
```

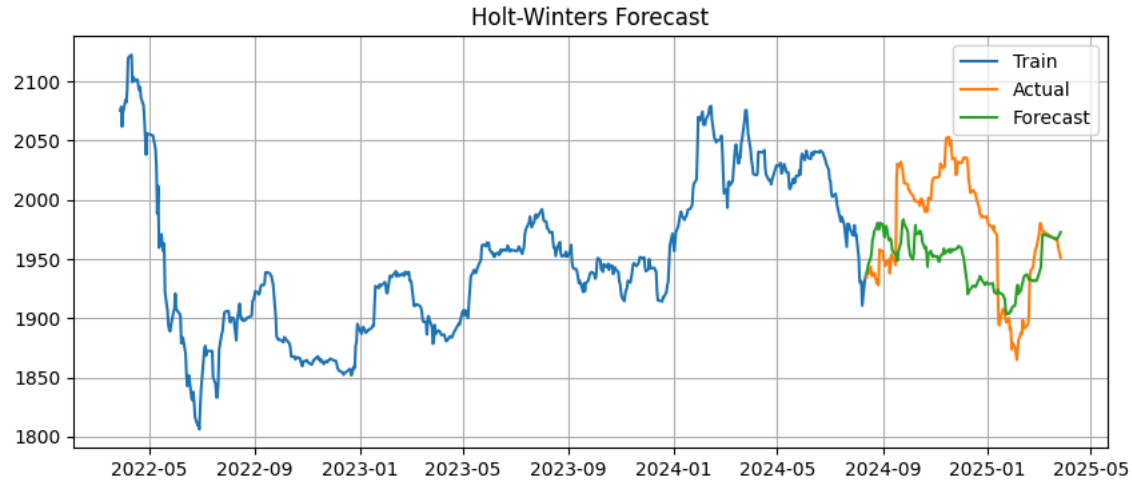
```
preds = model.forecast(len(test))
```

```
/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(
```

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

```
Holt-Winters Exponential Smoothing - RMSE: 51.41
```

```
# Plot
plt.figure(figsize=(10, 4))
plt.plot(train.index, train, label='Train')
plt.plot(test.index, test, label='Actual')
plt.plot(test.index, preds, label='Forecast')
plt.title('Holt-Winters Forecast')
plt.legend()
plt.grid(True)
plt.show()
```




```
1 Start coding or generate with AI.
```