Time Series Analysis

Project done by **Maryna Borovyk** and **Illia Pastushok**

Dataset

We used dataset from Kaggle that contains Russians losses of equipment in period of Russian-Ukrainian war. This dataset contains columns with incremental data about amount of destroyed equipment (tanks, drones, APC (Armored personnel carriers)

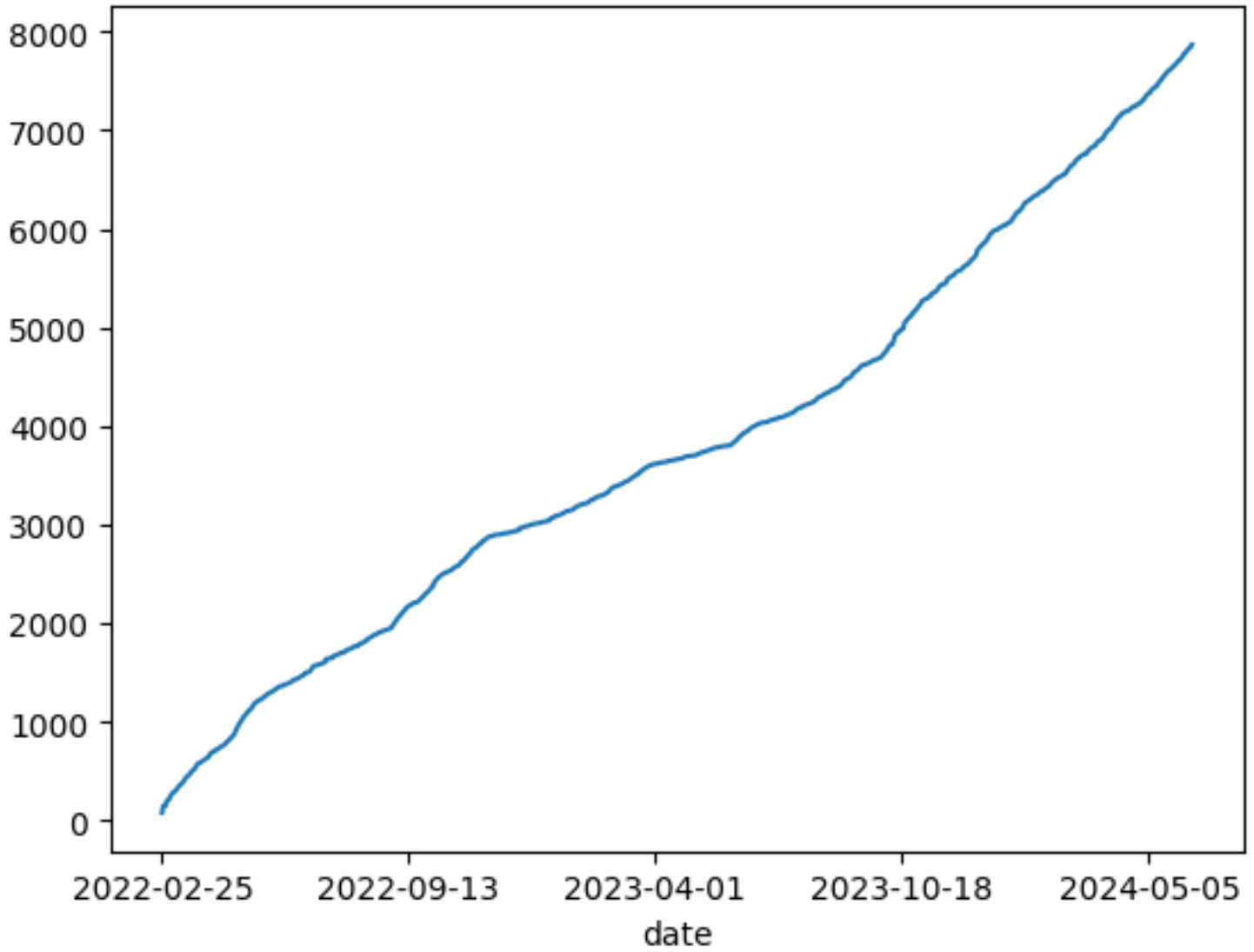
```
import pandas as pd
df = pd.read_csv("russia_losses_equipment.csv", index_col="date")
df = df.iloc[::-1]
df
```

Executed at 2024.06.12 13:57:51 in 4s 228ms

1-10 | 836 rows × 18 columns pd.DataFrame | CSV

date	day	aircraft	helicopter	tank	APC	field artillery	MRL	military auto	fuel tank	drone	naval ship	anti-airc
2022-02-25	2	10	7	80	516	49	4	100.0	60.0	0	2	
2022-02-26	3	27	26	146	706	49	4	130.0	60.0	2	2	
2022-02-27	4	27	26	150	706	50	4	130.0	60.0	2	2	
2022-02-28	5	29	29	150	816	74	21	291.0	60.0	3	2	
2022-03-01	6	29	29	198	846	77	24	305.0	60.0	3	2	
2022-03-02	7	30	31	211	862	85	40	355.0	60.0	3	2	
2022-03-03	8	30	31	217	900	90	42	374.0	60.0	3	2	
2022-03-04	9	33	37	251	939	105	50	404.0	60.0	3	2	
2022-03-05	10	39	40	269	945	105	50	409.0	60.0	3	2	

Imported from *https://www.kaggle.com/datasets/piterfm/2022-ukraine-russian-war*



Plotted df for tanks

Which analysis models do we use?

The analysis of the dataset was performed using 3 different models:

- Prophet
- SARIMAX
- Holt’s Winter

Every method was used to predict number of destroyed tanks for 280 last values used as test data.

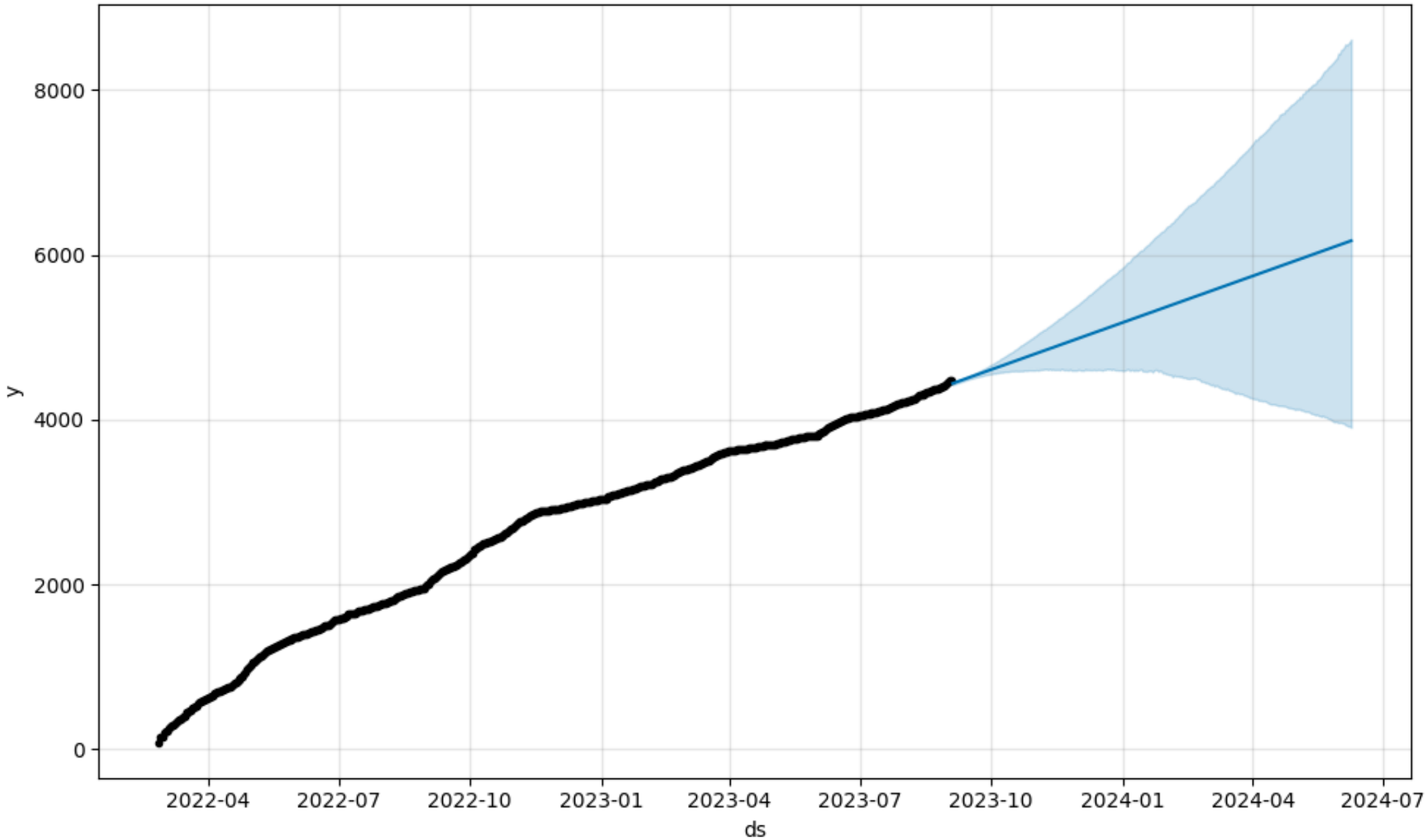
```python
from prophet import Prophet
import matplotlib.pyplot as plt
from sklearn.metrics import mean_absolute_percentage_error
# Prepare the data
train = df[['tank']].iloc[:-280].reset_index()
train.columns = ['ds', 'y']
test = df[['tank']].iloc[-280:].reset_index()
test.columns = ['ds', 'y']

# Initialize the Model
model=Prophet()

# Fit the Model
model.fit(train)

# Predict
forecast = model.predict(test)

# Plot the forecast
model.plot(forecast)
plt.show()

print(mean_absolute_percentage_error(test['y'],forecast['yhat'][-len(test):])*100)
```

Executed at 2024.06.13 07:54:16 in 387ms

Prophet

Prophet is a forecasting model maintained by Meta.
At its core, the Prophet procedure is an additive regression model with four main components:

- A piecewise linear or logistic growth curve trend. Prophet automatically detects changes in trends by selecting changepoints from the data.
- A yearly seasonal component modeled using Fourier series.
- A weekly seasonal component using dummy variables.
- A user-provided list of important holidays.


8000
6000
4000
2000
0
y
2022-04
2022-07
2022-10
2023-01
2023-04
2023-07
2023-10
2024-01
2024-04
2024-07
ds

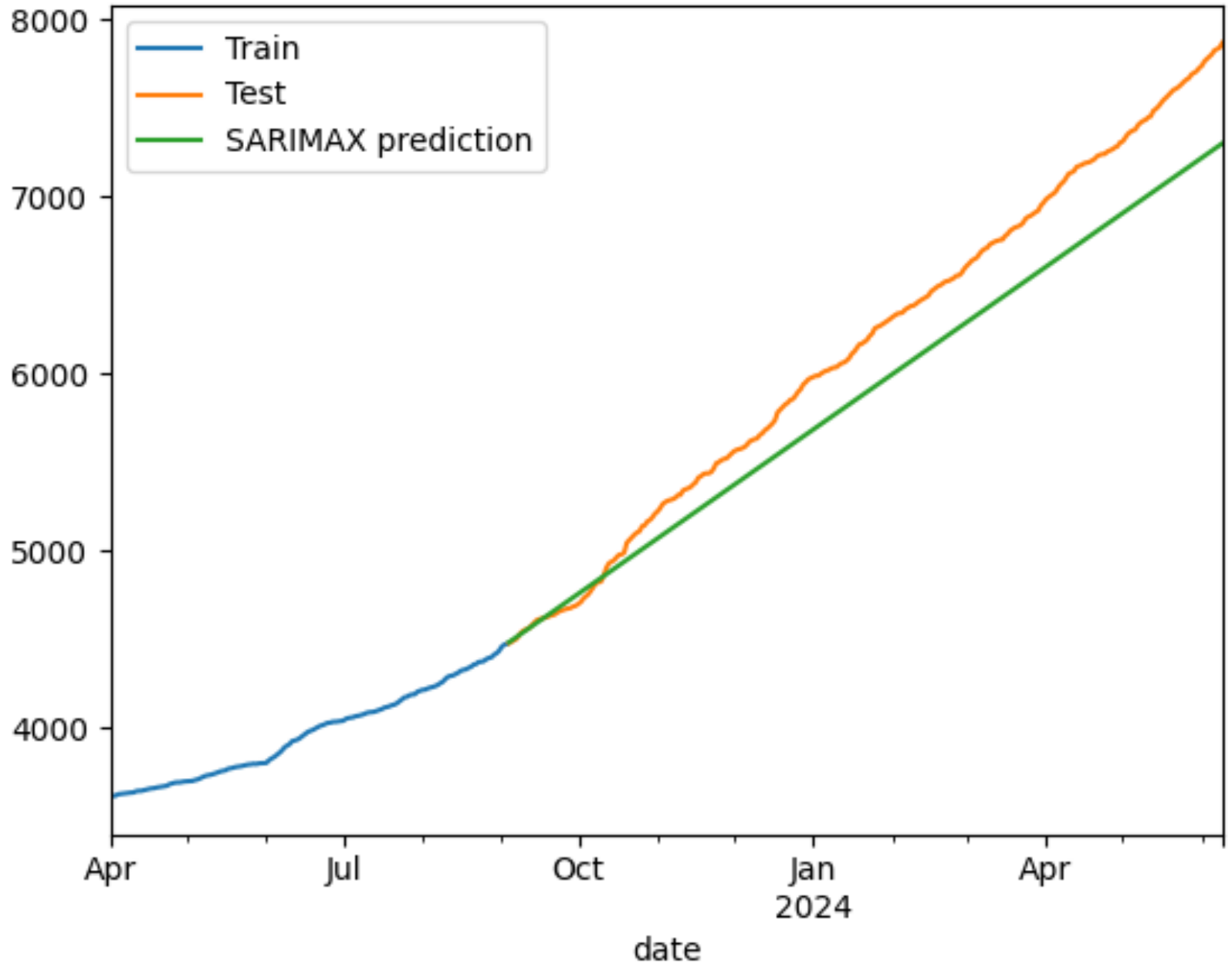
SARIMAX

The Seasonal Autoregressive Integrated Moving Average with Exogenous Regressors (SARIMAX) model is a powerful time series forecasting technique that extends the traditional ARIMA model to account for seasonality and external factors.

```python
from statsmodels.tsa.statespace.sarimax import SARIMAX
from pmdarima import auto_arima
auto_arima(df['tank'],seasonal=False,trace=True).summary()

train=df.iloc[:-280]
test=df.iloc[-280:]
train.index = pd.to_datetime(train.index)
test.index = pd.to_datetime(test.index)
modelSARIMAX=SARIMAX(train['tank'],order=(0,2,1), freq='D').fit()

start=len(train)
end=start+len(test)-1
predictions=modelSARIMAX.predict(start=start,end=end, dynamic=False).rename('SARIMAX for tanks')
train['tank'].iloc[400:].plot(legend=True,label='Train')
test['tank'].plot(legend=True, label='Test')
predictions.plot(legend=True, label='SARIMAX prediction')

print(mean_absolute_percentage_error(test['tank'],predictions)*100)
```

Executed at 2024.06.13 07:54:17 in 1s 224ms

```
Best model:  ARIMA(0,2,1)(0,0,0)[0]
```

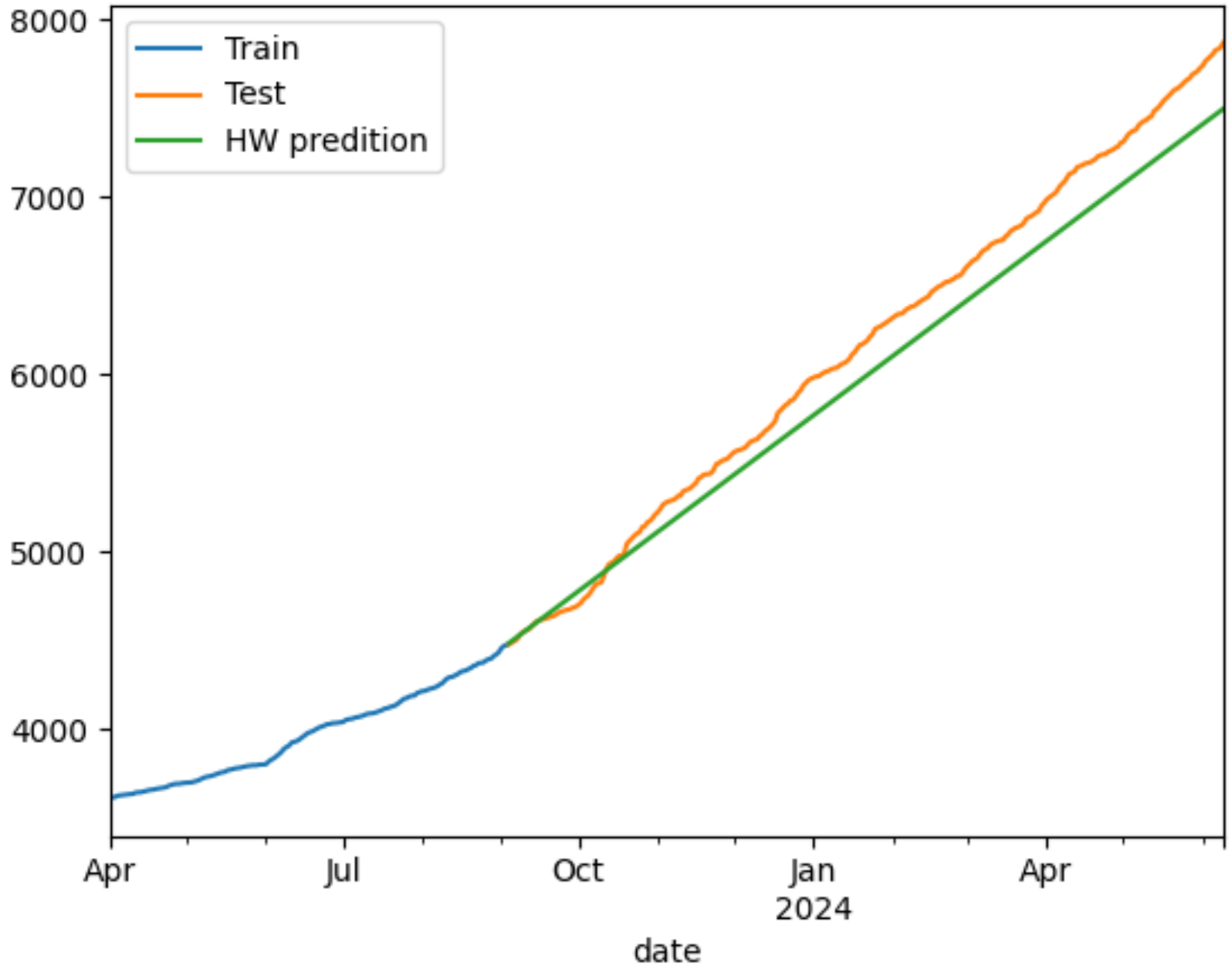

Train
Test
SARIMAX prediction
8000
7000
6000
5000
4000
Apr
Jul
Oct
Jan
2024
Apr
date

```python
from statsmodels.tsa.holtwinters import ExponentialSmoothing
train=df.iloc[:-280]
test=df.iloc[-280:]
train.index = pd.to_datetime(train.index)
test.index = pd.to_datetime(test.index)
fitHoltWinter=ExponentialSmoothing(train['tank'],trend='add').fit()

fcastHoltrWinter=fitHoltWinter.forecast(len(test)).rename('HW Predict')

train['tank'].iloc[400:].plot(legend=True,label='Train')
test['tank'].plot(legend=True,label='Test')
fcastHoltrWinter.plot(legend=True,label='HW predition')

print(mean_absolute_percentage_error(test['tank'],fcastHoltrWinter)*100)
```

Executed at 2024.06.13 07:54:17 in 320ms

Holt Winters

The Holt-Winters algorithm is a time-series forecasting method that uses exponential smoothing to make predictions based on past observations.


8000
7000
6000
5000
4000
Train
Test
HW predition
Apr
Jul
Oct
Jan
2024
Apr
date

Mean Absolute Percentage Errors

- For **Prophet**: 12.937646108752975
- For **SARIMAX**: 4.061681357223066
- For **Holt's Winter**: 2.642802854920465 6

Conclusion: As result of MAPE analysis we can conclude that Holt's Winter model from Meta fitted the best with prediction of our data

Thank you for attention!