Financial Programming

Individual Project

Web Scrapping, Forecasting and Visualization Project

Andres Olivera Caceres

Agenda

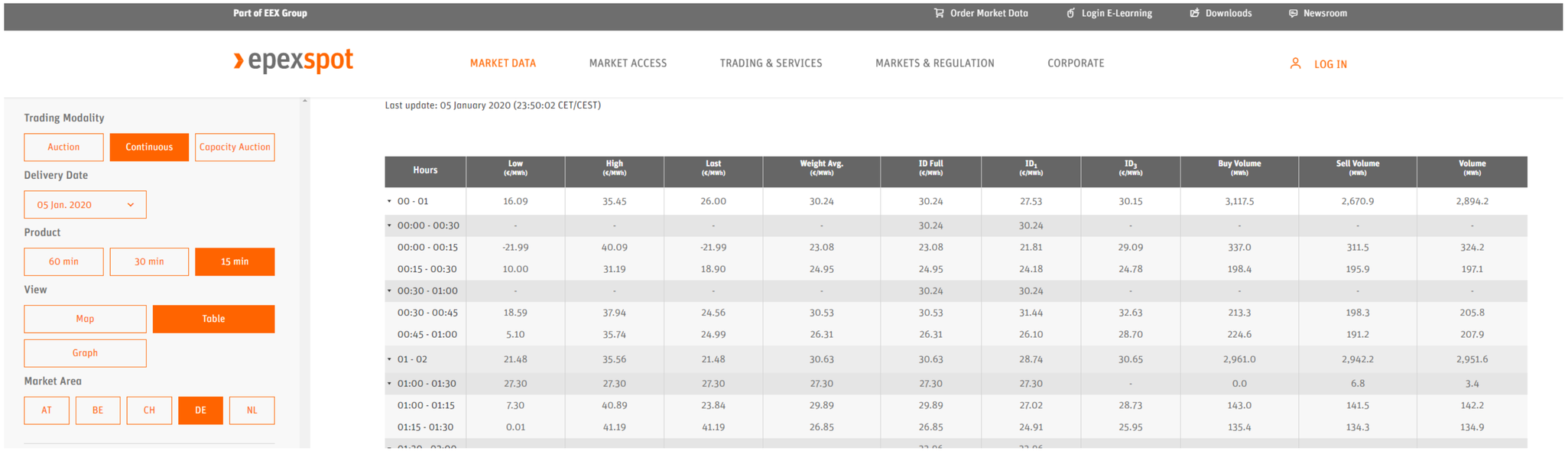
- Web Scrapping
- Data cleansing
- Forecasting
- Dashboard
- Conclusions

Web Scrapping

Web Scrapping

- Data from European Power Exchange (EXPEX SPOT) Germany
- At least 2 weeks

This is how the webpage looks now. After some changes and stopping web scrapping from the webpage



Hours	Low (€/MWh)	High (€/MWh)	Last (€/MWh)	Weight Avg. (€/MWh)	ID Full (€/MWh)	ID_1 (€/MWh)	ID_3 (€/MWh)	Buy Volume (MWh)	Sell Volume (MWh)	Volume (MWh)
00 - 01	16.09	35.45	26.00	30.24	30.24	27.53	30.15	3,117.5	2,670.9	2,894.2
00:00 - 00:30	-	-	-	-	30.24	30.24	-	-	-	-
00:00 - 00:15	-21.99	40.09	-21.99	23.08	23.08	21.81	29.09	337.0	311.5	324.2
00:15 - 00:30	10.00	31.19	18.90	24.95	24.95	24.18	24.78	198.4	195.9	197.1
00:30 - 01:00	-	-	-	-	30.24	30.24	-	-	-	-
00:30 - 00:45	18.59	37.94	24.56	30.53	30.53	31.44	32.63	213.3	198.3	205.8
00:45 - 01:00	5.10	35.74	24.99	26.31	26.31	26.10	28.70	224.6	191.2	207.9
01 - 02	21.48	35.56	21.48	30.63	30.63	28.74	30.65	2,961.0	2,942.2	2,951.6
01:00 - 01:30	27.30	27.30	27.30	27.30	27.30	27.30	-	0.0	6.8	3.4
01:00 - 01:15	7.30	40.89	23.84	29.89	29.89	27.02	28.73	143.0	141.5	142.2
01:15 - 01:30	0.01	41.19	41.19	26.85	26.85	24.91	25.95	135.4	134.3	134.9

Web Scrapping

```
import requests
import pandas as pd
from requests import get
from requests.exceptions import RequestException
from contextlib import closing
from bs4 import BeautifulSoup
```

```
pip install beautifulsoup4
```

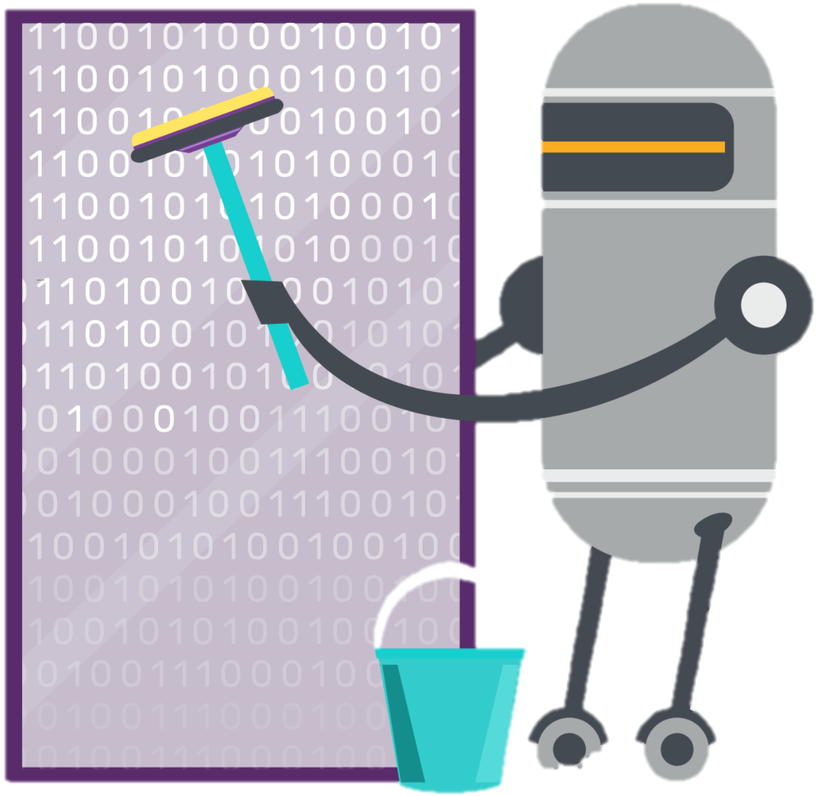
```
url = "https://www.epexspot.com/en/market-data/intradaycontinuous/ir
headers = {'User-Agent': 'Mozilla/5.0 (Windows NT 6.3; Win64; x64) A
r = requests.get(url, headers=headers)
```

```
#Iniyiate BS
soup = BeautifulSoup(r.content, "html.parser")
#Find right table
table = soup.find_all('table')[0]
# get the needed part
rows = table.find_all('tr')
```

```
#Start empty list
row_list = list()
# Iterate into all data and append it into a list
for tr in rows:
    td = tr.find_all('td')
    row = [i.text for i in td]
    row_list.append(row)
```

- Code prepared to scraped the webpage.
- BeutifilSoup package
- Recommended article

Data Cleansing



Data Cleansing

A	B	C	D	E	F	G	H	I	J	K	L	M
	DateTime	Low	High	Last	Weighted_Avg	Idx	ID3	ID1	Buy_Vol	Sell_Vol	Index_Base	Index_Peal
1	01/01/2014 00:00	10	35.15	15.1	20.36	20.36	NA	NA	659	704	11	11.64
2	01/01/2014 01:00	-2	32.96	-2	13.13	13.13	NA	NA	1001	878	11	11.64
3	01/01/2014 02:00	0	23	23	9.81	9.81	NA	NA	1190.1	1061.1	11	11.64
4	01/01/2014 03:00	1	22	22	7.14	7.14	NA	NA	1069	880	11	11.64
5	01/01/2014 04:00	2	18	15	10.25	10.25	NA	NA	932.9	752.9	11	11.64
6	01/01/2014 05:00	1	19	5	9.59	9.59	NA	NA	1078.2	958.2	11	11.64
7	01/01/2014 06:00	2	14.85	5	6.33	6.33	NA	NA	1123.8	883.8	11	11.64
8	01/01/2014 07:00	2	18	8	4.88	4.88	NA	NA	994.8	849.8	11	11.64
9	01/01/2014 08:00	3	19	17	8.7	8.7	NA	NA	906.3	991.3	11	11.64
10	01/01/2014 09:00	1	19	8.5	8.68	8.68	NA	NA	1383	1388	11	11.64
11	01/01/2014 10:00	1	18.5	10	9.28	9.28	NA	NA	2694.3	3348.3	11	11.64
12	01/01/2014 11:00	7.25	24	10.2	10.45	10.45	NA	NA	2417.4	2759.4	11	11.64
13	01/01/2014 12:00	7.1	15	10	10.64	10.64	NA	NA	2071.2	3242.2	11	11.64

- Raw Dataset given

```
df = pd.read_csv('C:/Users/aolivera/OneDrive - IESEG/IESEG/Courses/Python/Individual project/IntradayContinuousEPEXSPOT_DE.csv')
```

```
df.head()
```

	Unnamed: 0	DateTime	Low	High	Last	Weighted_Avg	Idx	ID3	ID1	Buy_Vol	Sell_Vol	Index_Base	Index_Peak
0	1	2014-01-01 00:00:00	10.0	35.15	15.1	20.36	20.36	NaN	NaN	659.0	704.0	11.0	11.64
1	2	2014-01-01 01:00:00	-2.0	32.96	-2.0	13.13	13.13	NaN	NaN	1001.0	878.0	11.0	11.64
2	3	2014-01-01 02:00:00	0.0	23.00	23.0	9.81	9.81	NaN	NaN	1190.1	1061.1	11.0	11.64
3	4	2014-01-01 03:00:00	1.0	22.00	22.0	7.14	7.14	NaN	NaN	1069.0	880.0	11.0	11.64
4	5	2014-01-01 04:00:00	2.0	18.00	15.0	10.25	10.25	NaN	NaN	932.9	752.9	11.0	11.64

- Read in Jupyter with panda

Data Cleansing

Outliers

- No evidence of outliers
- Negative are okay in this case

NA

- Only two rows were deleted

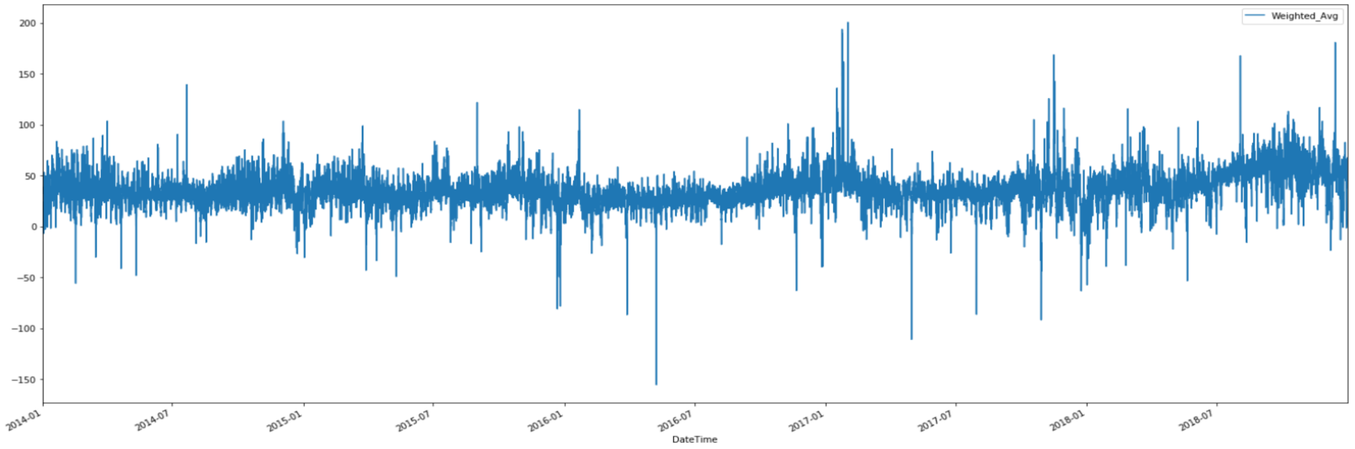
Other

- Final dataset with just DateTime and Wheighted_Avg
- DateTime type changed to date

Forecast

Forecast

Price and Date ploted.



- No sign of statinality at first sight

Dicky Fuller Test (First two numbers)

`(-16.440010691384987, 2.432844e-29 …)`

- It passed the test. First number is negative and second < 0.05

Forecast

```
# Get best model using auto_arima
best_model = auto_arima( timeseries1,              # data
                         d=0,              # non-seasonal difference order
                         start_p=1,        # initial guess for p
                         start_q=1,        # initial guess for q
                         max_p=3,          # max value of p to test
                         max_q=3,          # max value of q to test
                         information_criterion='aic',
                         trace=True,
                         error_action='ignore'
                       )
```

- Auto_arima. Simplifies the process

```
order_aic_bic =[]
# Loop over AR order
for p in range(3):
    # Loop over MA order
    for q in range(3):
        try:
            # Fit model
            model = SARIMAX(timeseries1, order=(p,0,q))
            results = model.fit()

            # Add order and scores to list
            order_aic_bic.append((p, q, results.aic, results.bic))
        except:
            # Print AIC and BIC as None when fails
            order_aic_bic.append((p, q, None, None))

# Make DataFrame of model order and AIC/BIC scores
order_df = pd.DataFrame(order_aic_bic, columns=['p','q', 'aic', 'bic'])


print(order_df.sort_values('aic'))
```

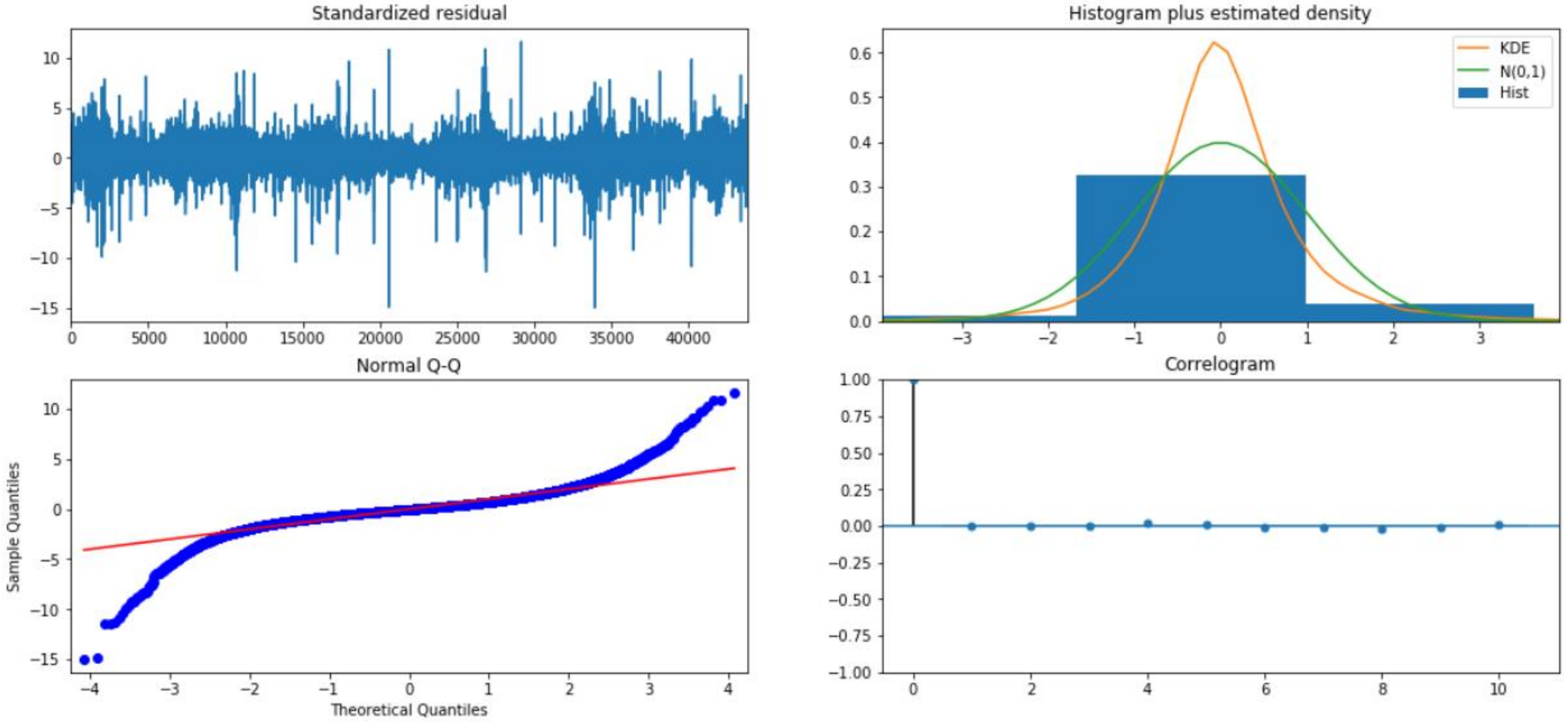
- Iterations with all posible orders. Fit and then get the best AIC

(3, 0, 2)

Forecast

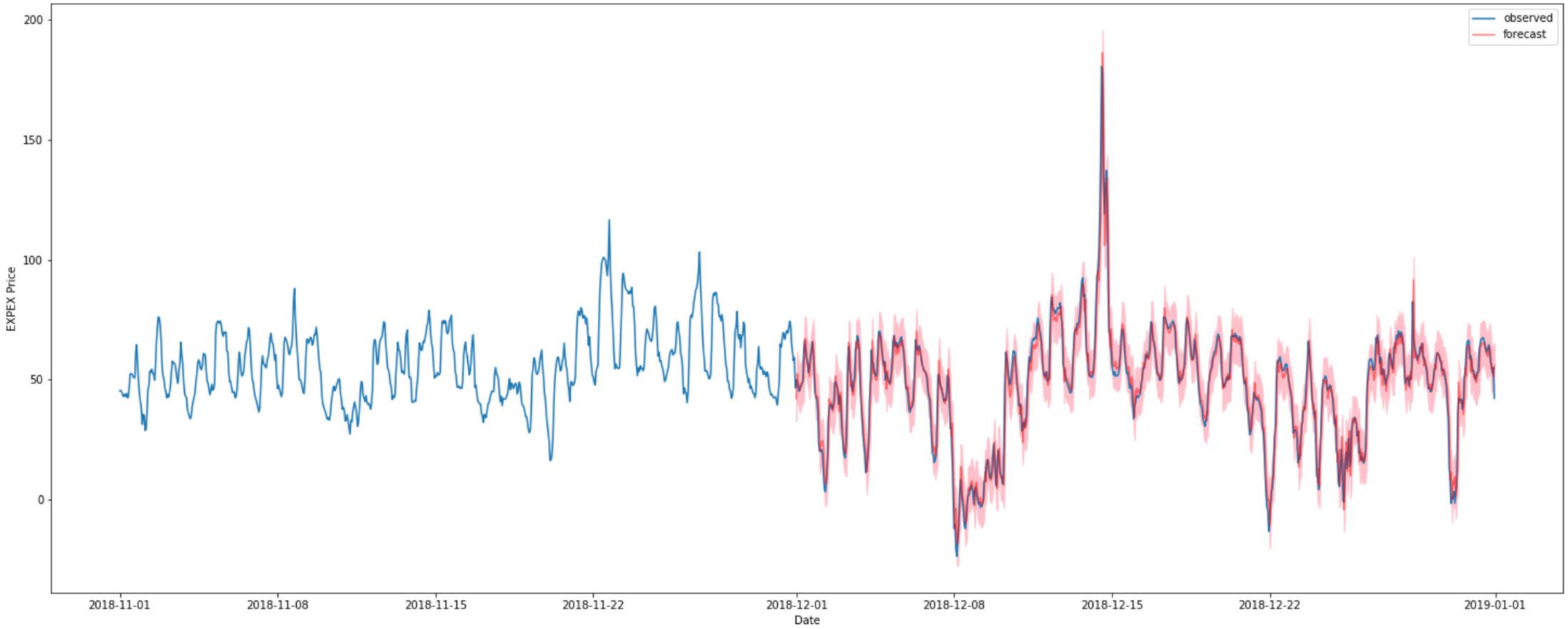
- Both way gave the same diagnostic and summary result.

Sarimax(3,0,1) was chosen



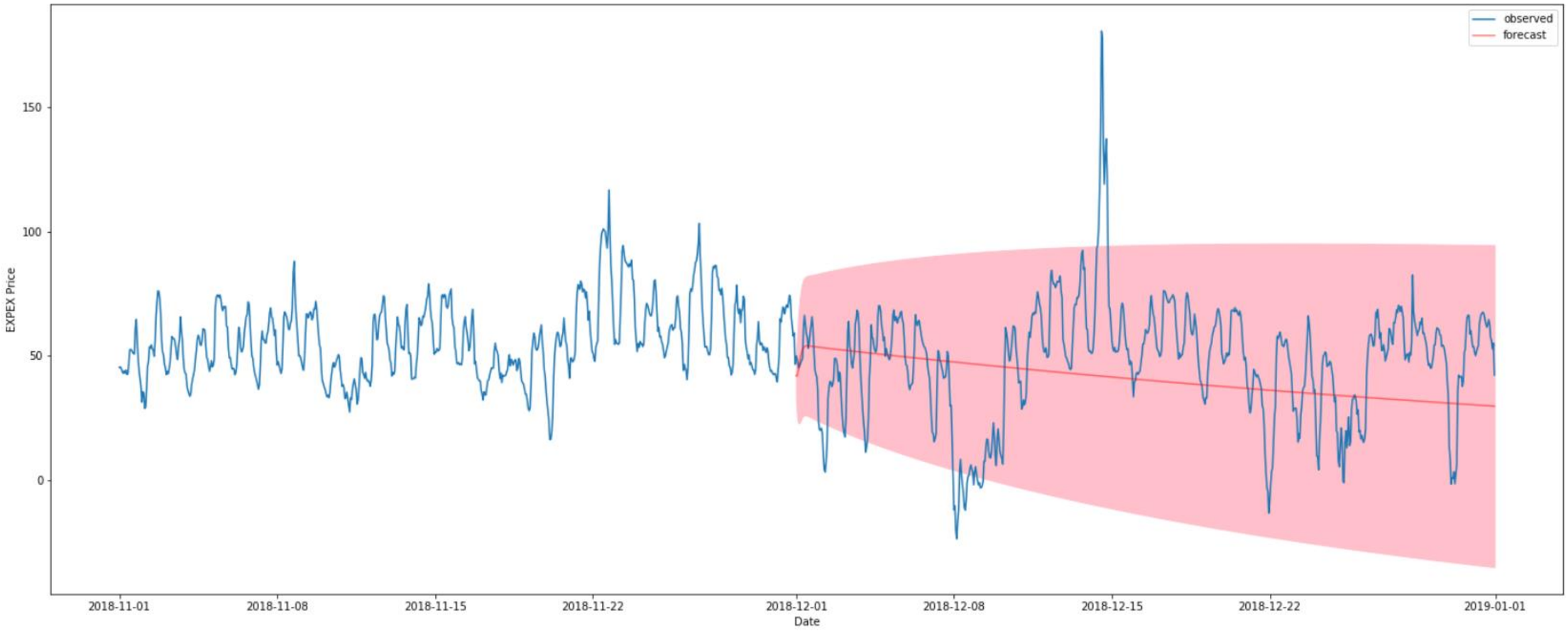
Forecast

- In sample prediction of the last month (2018-12-11 to 2019-01-01)



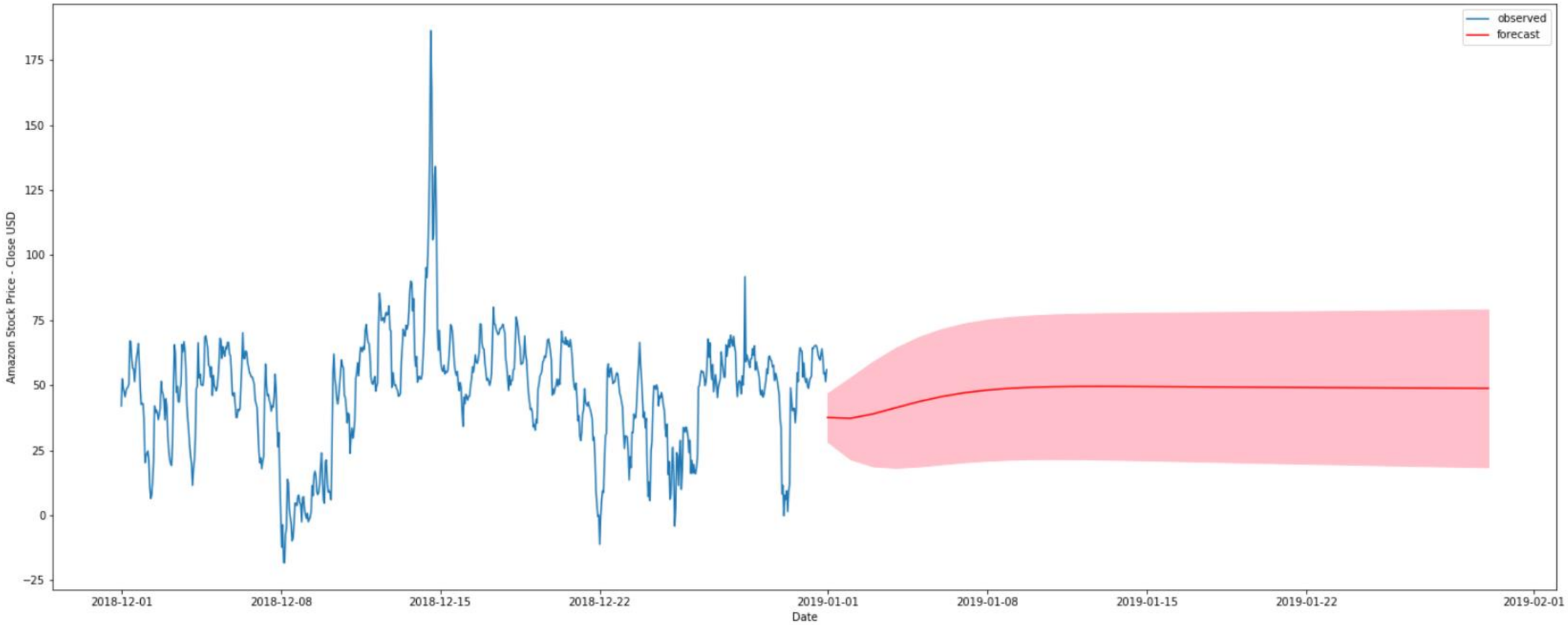
Forecast

- In sample prediction but Dynamic of the last month (2018-12-11 to 2019-01-01)

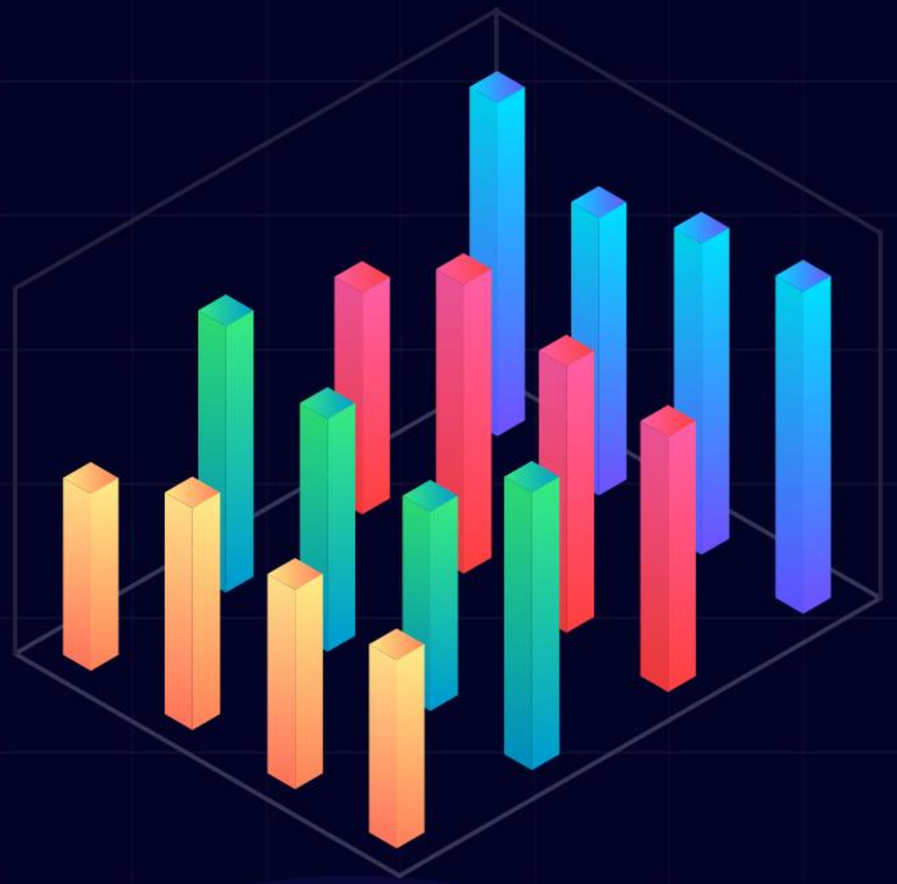


Forecast

- Forecast of next month (2019-01-01 to 2019-01-30)



DATA VISUALIZATION



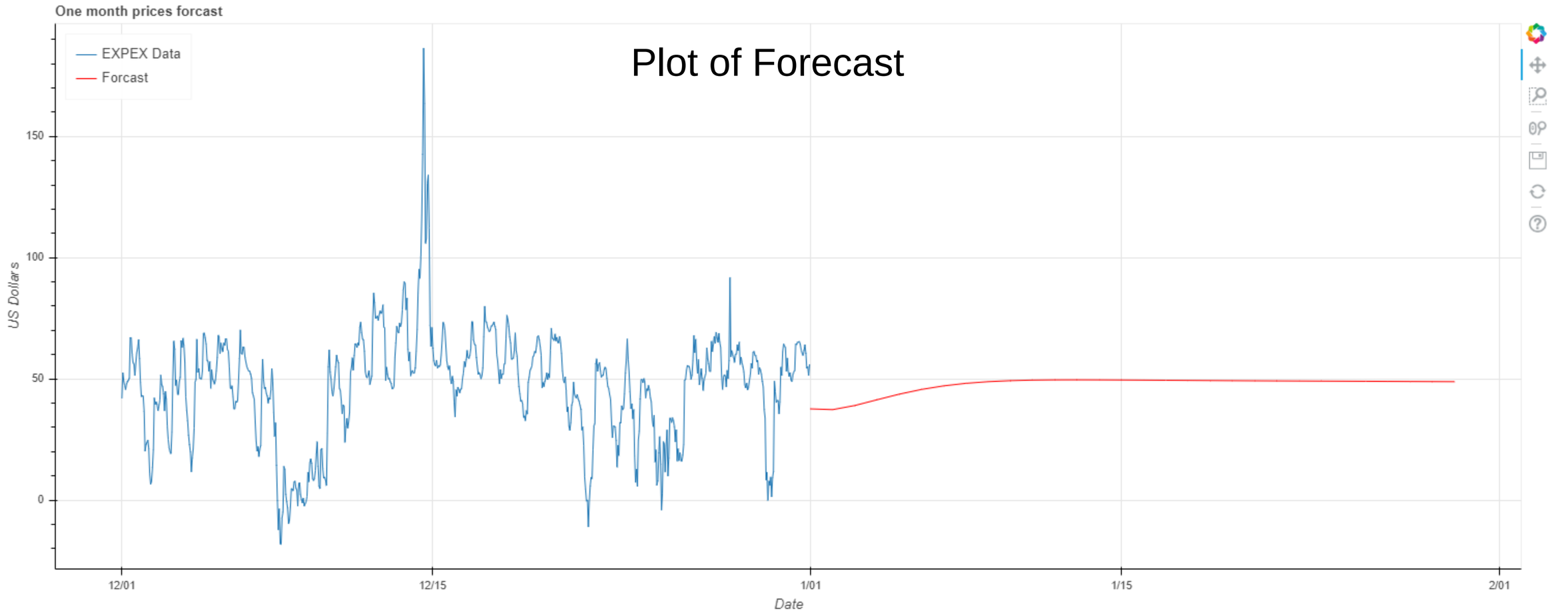
Dashboard

- Using Bokeh Serve an app with two apps was created

2 tabs

Dashboard

Origianl Data | Forecast



Conclusions

- The model can improve. The AIC is can be lower
- Leave more time to the visualization. Not as fast as thought
- I love the project. I learned so much. All applications new to me
- Most difficult part was the forecast. A lot of research to deem weather one was on the right track or not