A

REPORT

ON

“ADVANCE ARTIFICIAL INTELLIGENCE: ASSIGNMENT 3”

Submitted To

DR. GAURAV HARIT

IIT JODHPUR

॥ त्वं ज्ञानमयो विज्ञानमयोऽसि ॥

Submitted By

Alok Kumar Vinay Kumar Shukla (M22CS051)
Bikas Dutta(D22CS051)

AAI Assignment-3 Report

Task: To implement time series models.

Objectives: Part 1: ARIMA Model

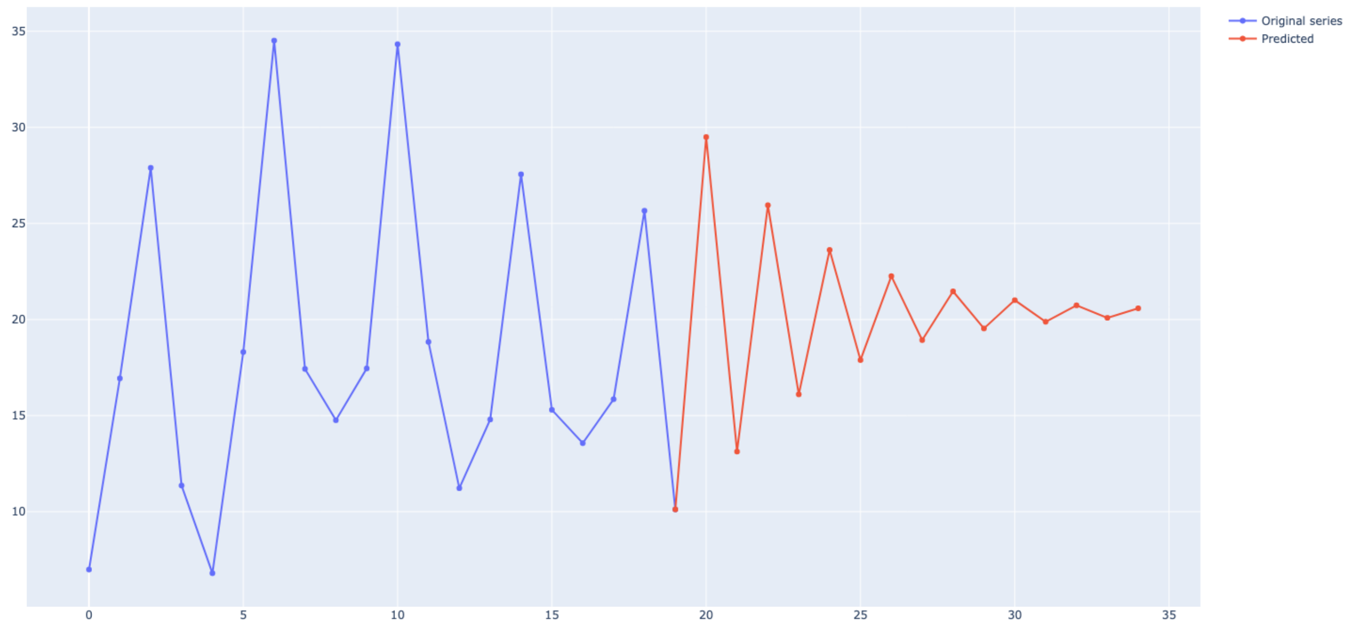
- ARIMA forecast
- Your program should output a list containing only the new terms in the predicted sequence.
- Complete this function from scratch.
- The function to complete is ARIMA_Forecast in the file forecasting.py.
- The primary test function calls the Plot() function.

Procedure:

- Import numpy for array operations.
- Implement the Multiple Linear Regression for fi's thetas and C's for AR and MA models.
- Implement details for regression needs not to discuss.
- The reference used for implementation is cited below at the end.
- Main ARIMA_Forecast implementation starts now.
- Make copies of the original series such that they are not modified non-intently.
- Applying for Shift operation D numbers of times.
- Creating new X inputs for regression.
- Applying Regression and getting parms using P, Q after applying reshaping such that multiplication can happen.
- After getting AR model prams, find MA params.
- Start Prediction once gets all the parameters.
- Predictions are made using previous values and AR and MA's.
- Find epsilons and add to outputs series by adding AR and MA values
- Returning Output Series and plotted using test.py file.
- Result:

$$y'_t = c + \phi_1 y'_{t-1} + \cdots + \phi_p y'_{t-p} + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q} + \varepsilon_t,$$

$$\underset{\text{AR}(p)}{\underset{\uparrow}{(1 - \phi_1 B - \cdots - \phi_p B^p)}} \quad \underset{d \text{ differences}}{\underset{\uparrow}{(1 - B)^d y_t}} \quad = \quad c + \underset{\text{MA}(q)}{\underset{\uparrow}{(1 + \theta_1 B + \cdots + \theta_q B^q)\varepsilon_t}}$$


Original series
Predicted
35
30
25
20
15
10
0
5
10
15
20
25
30
35

Objectives: Part 2: Holt-Winter's Forecast

- **Holt-Winter's** forecast
- Your program should output a list containing only the new terms in the predicted sequence.
- Complete this function from scratch.
- The function to complete is HoltWinter_Forecast in the file forecasting.py.
- The primary test function calls the Plot() function.

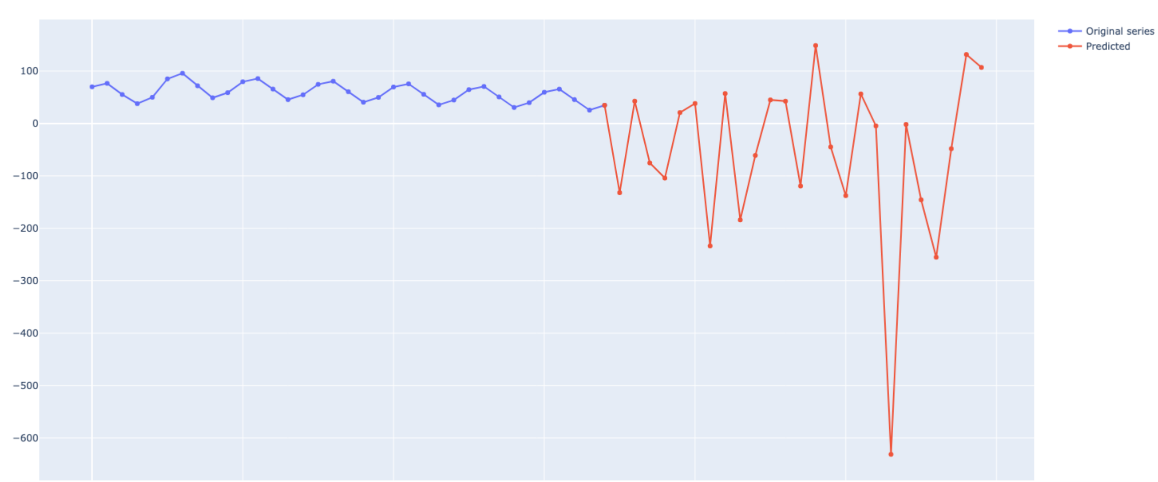
Procedure:

- Import numpy for array operations.
- Initializing level,trend,seasonal appropriately.
- Make copies of the original series such that they are not modified non-intently.
- Making Predictions using the following equations.

$$
\begin{aligned}
\hat{y}_{t+h|t} &= \ell_t + hb_t + s_{t+h-m(k+1)} \\
\ell_t &= \alpha(y_t - s_{t-m}) + (1-\alpha)(\ell_{t-1} + b_{t-1}) \\
b_t &= \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1} \\
s_t &= \gamma(y_t - \ell_{t-1} - b_{t-1}) + (1-\gamma)s_{t-m},
\end{aligned}
$$

Note: Holt-Winters' additive method is used.

- Results:



Objectives: Part 2: Forecast the series

- There are 5 initialised series, namely S1, S2, S3, S4, S5
- Your program should output a list containing only the new terms in the predicted sequence.
- Forecast the next 20 values for each of these series.
- The functions to complete are ARIMA_Paramters which returns the tuple of, and HoltWinter_Parameters, which returns a tuple of in this specific order.
- You may use any 3rd party libraries for this task.
- Edit the primary function in the file tests.py and use the predefined Plot() function to visualize your results on each series in the same format.

Procedure:

FOR ARIMA PARMS

- Installing statsmodels if not installed already.
- Importing required libs.
- Defining the parameters for combinations.

```
# install state model package
## Required libs ##
os.system("pip3 -q install statsmodels")
os.system("pip3 -q install scikit-learn")
import statsmodels.api as sm
from itertools import product
import pandas as pd
from sklearn.metrics import mean_squared_error
## Required libs ##

## parms inti ##
p = [2]
d = [0,1]
q = [0,1]

train_split = 0.8
best_order = (0,0,0)
Min_MSE = np.inf
## prams inti ##
```

- Creating different parameters for combinations using itertools product function.
- It does Product([1,2], [3,4]) returns [(1,3),(1,4),(2,3),(2,4)]
- Creating Training and Testing data frames using pandas.
- Initializing the ARIMA model from the stats model.
- Finding MSE error using sklearn.metrics.

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

MSE = mean squared error

n = number of data points

Y_i = observed values

$\hat{Y}_i$ = predicted values

- Setting best orders based on minimum error.
- Return the order and print the order.

FOR HoltWinter PARMS

- Same as above.
- Installing statsmodels if not installed already.
- Importing required libs.
- Defining the parameters for combinations.

```python
# install state model package
## Required libs ##
os.system("pip3 -q install statsmodels")
os.system("pip3 -q install scikit-learn")
from statsmodels.tsa.holtwinters import ExponentialSmoothing
from itertools import product
import pandas as pd
from sklearn.metrics import mean_squared_error
## Required libs ##

alpha = [0.3, 0.7, 0.5]
beta = [0.2, 0.6, 0.8]
gamma = [0.25, 0.35, 0.65]
seasonality = [2, 3, 5]

best_order = (0,0,0,0)
Min_MSE = np.inf
```

- Creating different parameters for combinations using itertools product function.
- It does Product([1,2], [3,4]) returns [(1,3),(1,4),(2,3),(2,4)]
- Creating Training and Testing data frames using pandas.
- Initializing the ARIMA model from the stats model.
- Finding MSE error using sklearn.metrics.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

MSE = mean squared error

n = number of data points

Y_i = observed values

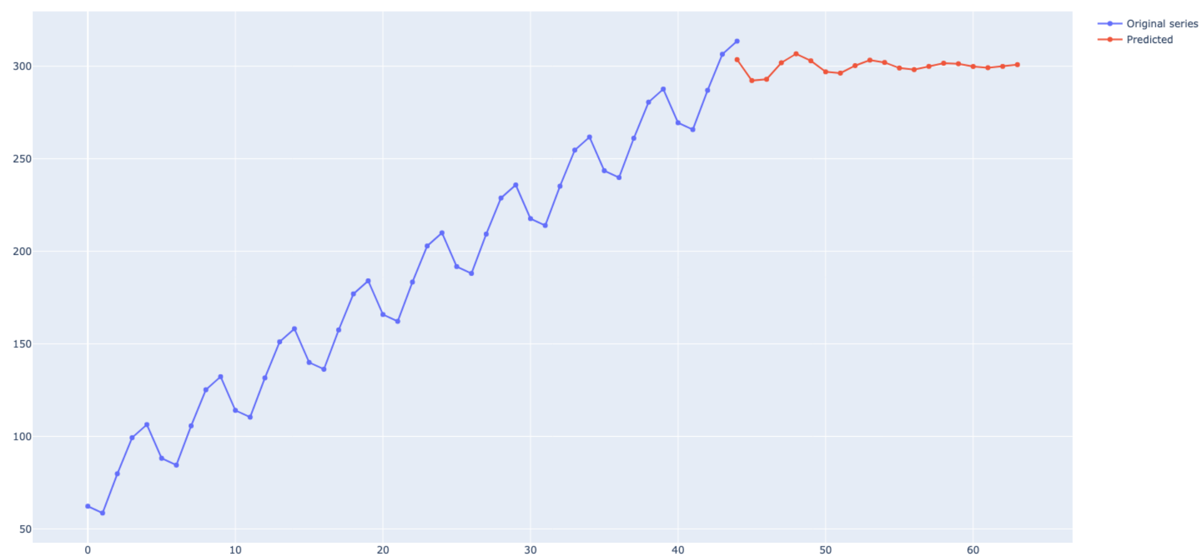
$\hat{Y}_i$ = predicted values

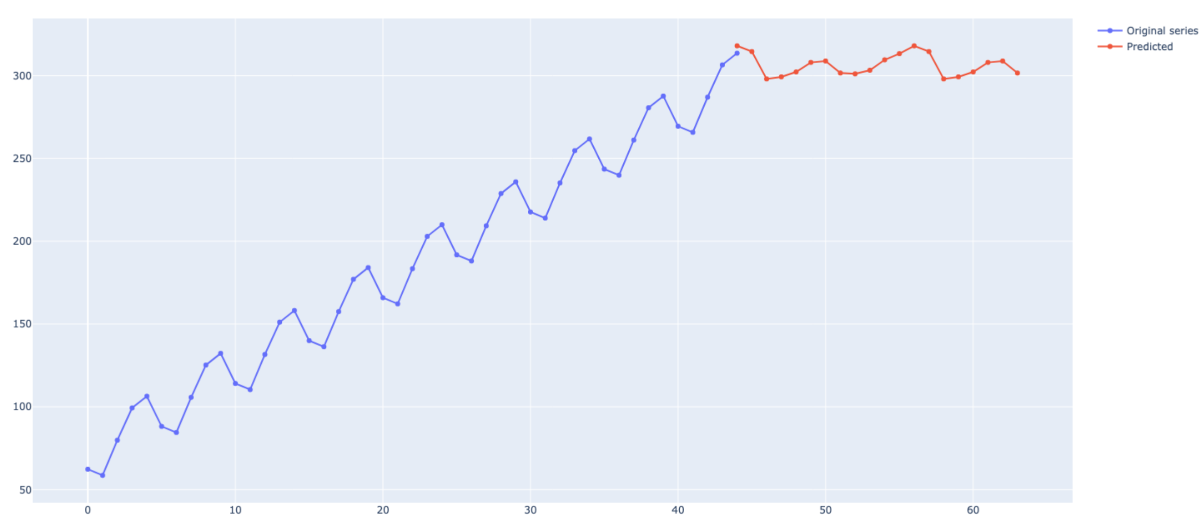
- Setting best orders based on minimum error.
- Return the order and print the order.

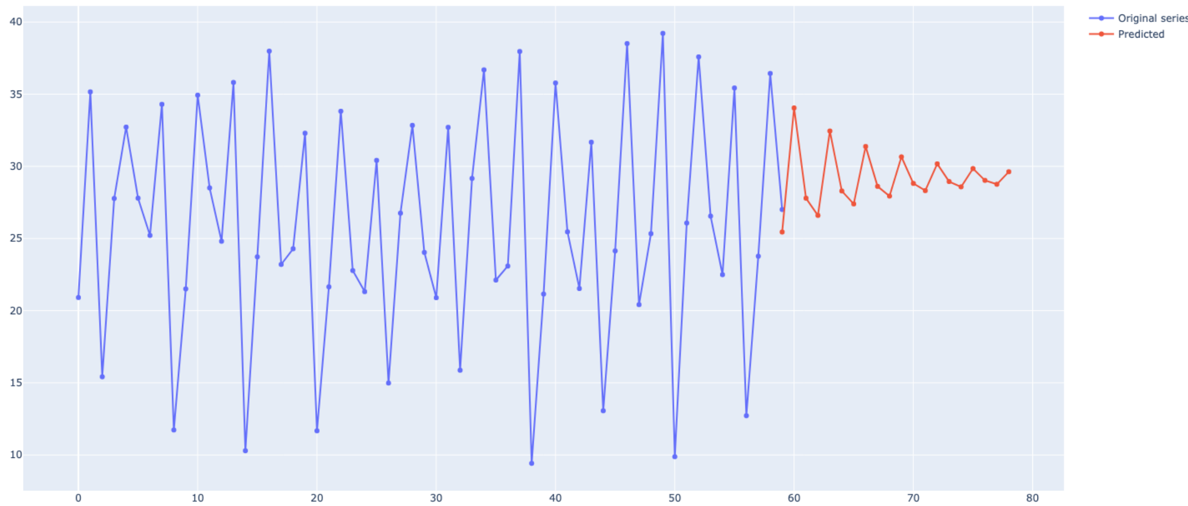
Results for the above-computed orders

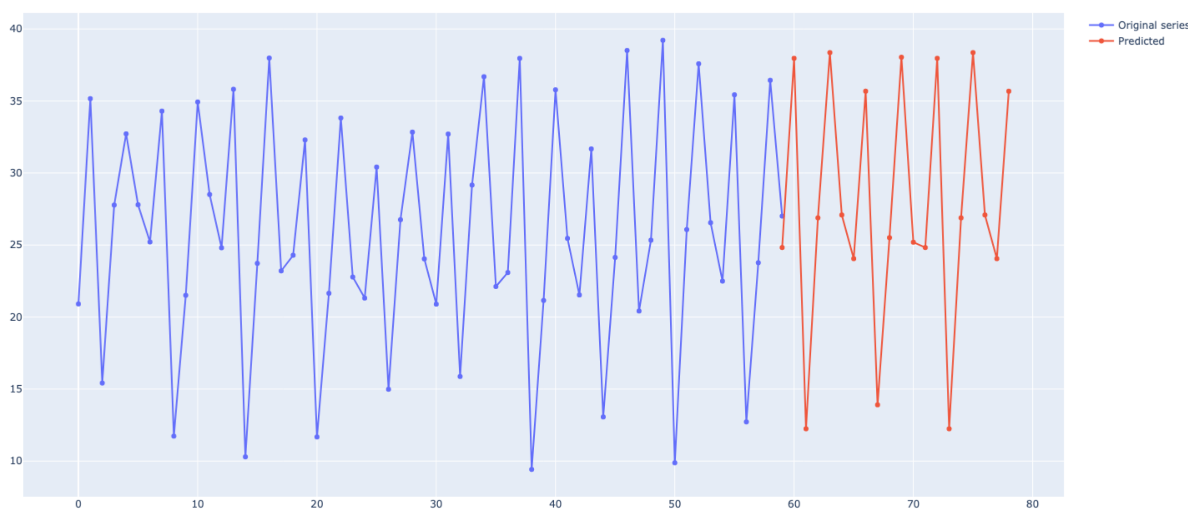
```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL                                                     zsh

~/IIT-J/AAI/Assignment3 ❯ python3 tests.py                                                04:43:46 AM
/Users/tron/opt/anaconda3/envs/iitj/lib/python3.10/site-packages/statsmodels/tsa/statespace/sarimax.py:966: UserWarning: Non-stationary starting autore
gressive parameters found. Using zeros as starting parameters.
  warn('Non-stationary starting autoregressive parameters'
/Users/tron/opt/anaconda3/envs/iitj/lib/python3.10/site-packages/statsmodels/tsa/statespace/sarimax.py:978: UserWarning: Non-invertible starting MA par
ameters found. Using zeros as starting parameters.
  warn('Non-invertible starting MA parameters found.'
ARIMA parms: (2, 1, 1) MSE: 1838.898
HOLT_WINTER parms: (0.7, 0.2, 0.25, 2) MSE: 2345.210
ARIMA parms: (2, 1, 0) MSE: 39.109
HOLT_WINTER parms: (0.3, 0.2, 0.25, 2) MSE: 94.194
ARIMA parms: (2, 0, 0) MSE: 125.353
HOLT_WINTER parms: (0.3, 0.2, 0.25, 2) MSE: 109.095
ARIMA parms: (2, 1, 0) MSE: 1821.061
HOLT_WINTER parms: (0.7, 0.2, 0.25, 2) MSE: 2521.950
ARIMA parms: (2, 0, 1) MSE: 36111.838
HOLT_WINTER parms: (0.3, 0.2, 0.25, 2) MSE: 80485.623

~/IIT-J/AAI/Assignment3 30s ❯                                                             04:44:46 AM
```

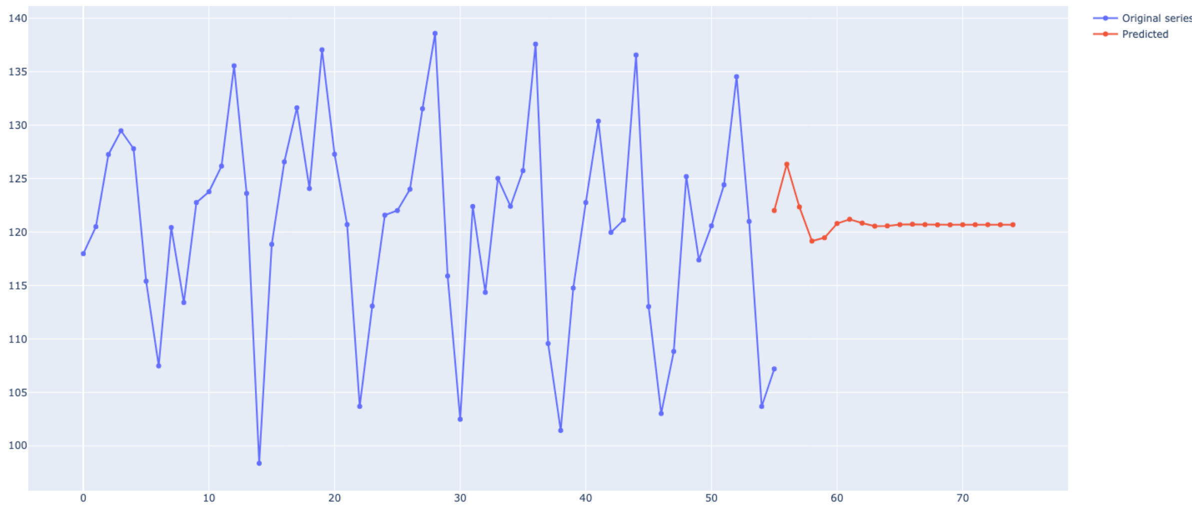
NOTE:- Ignore User Warning

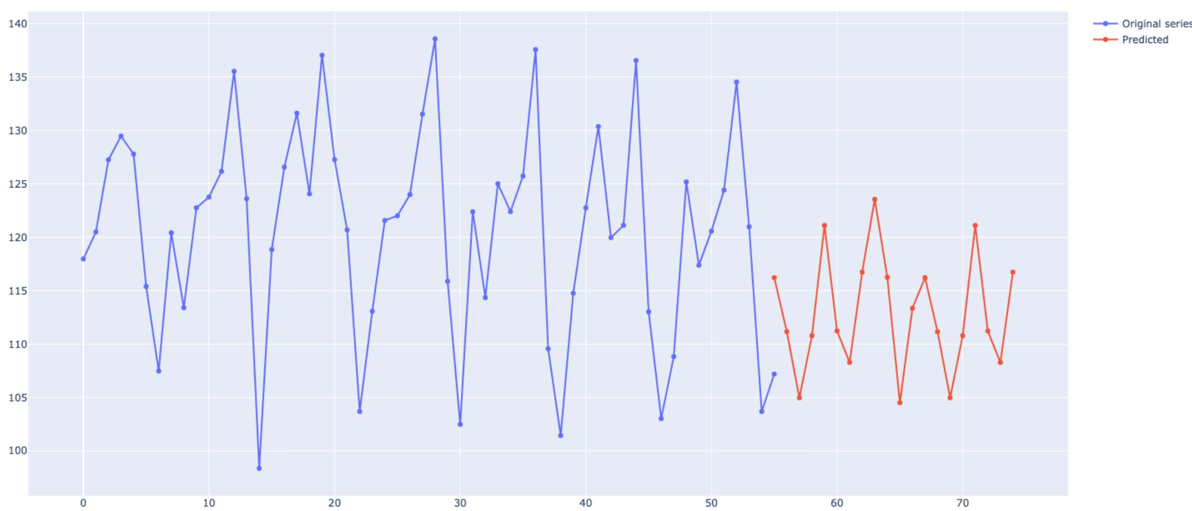

Original series
Predicted
300
250
200
150
100
50
0
10
20
30
40
50
60

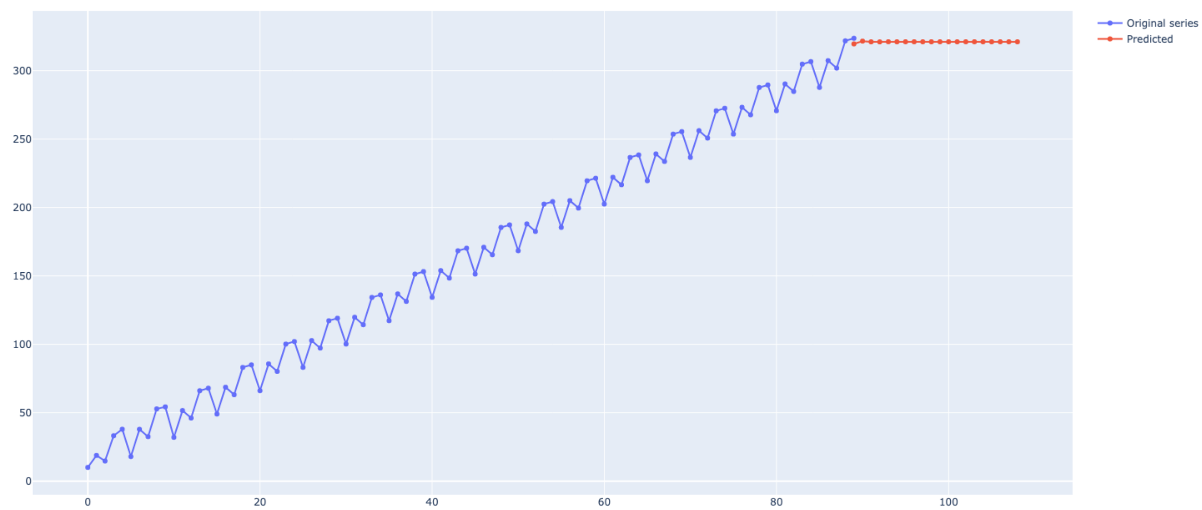

Original series
Predicted
300
250
200
150
100
50
0
10
20
30
40
50
60

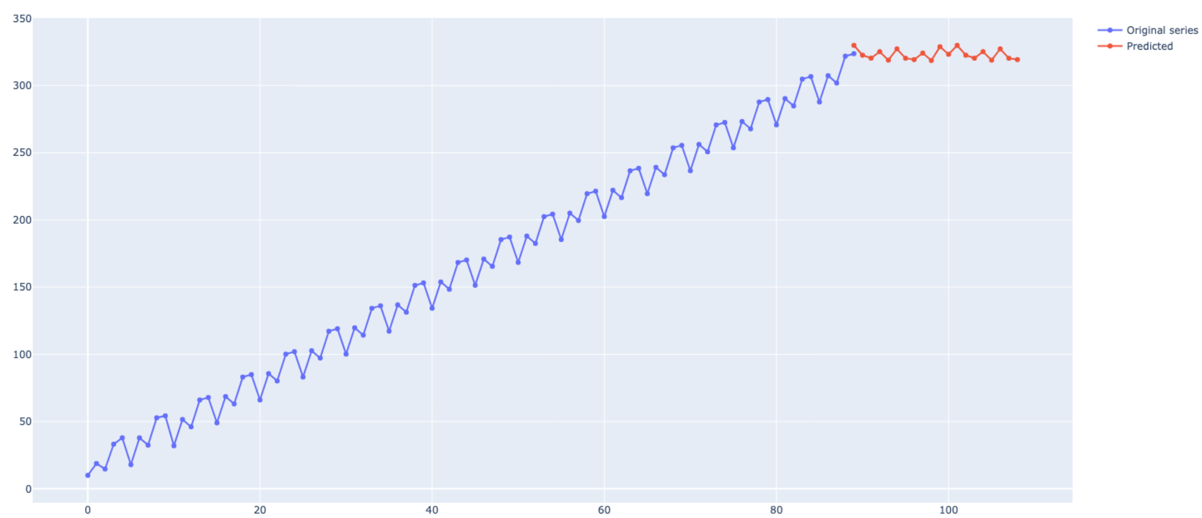

Original series
Predicted
40
35
30
25
20
15
10
0
10
20
30
40
50
60
70
80

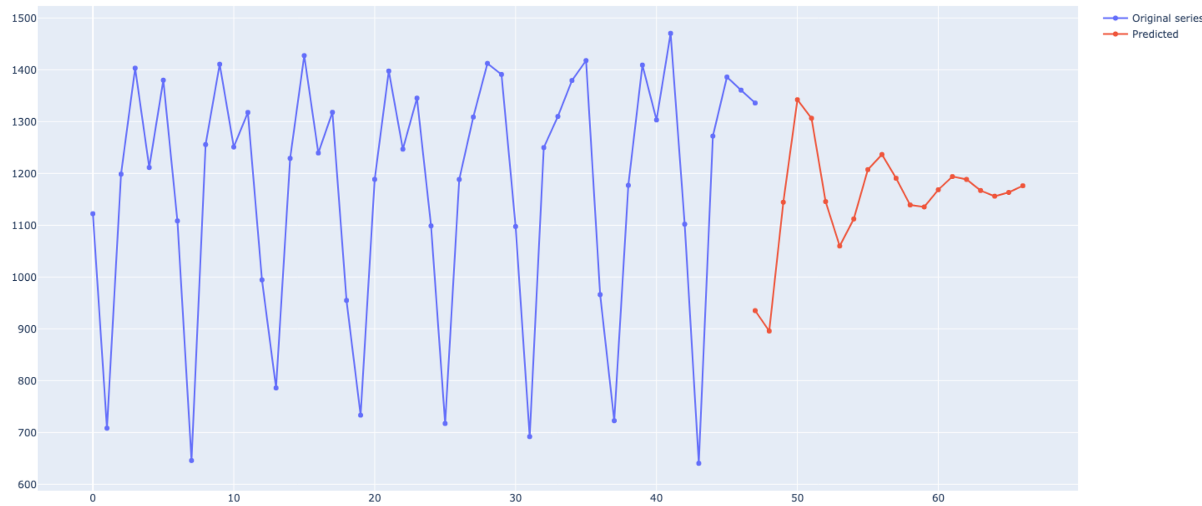

Original series
Predicted
40
35
30
25
20
15
10
0
10
20
30
40
50
60
70
80

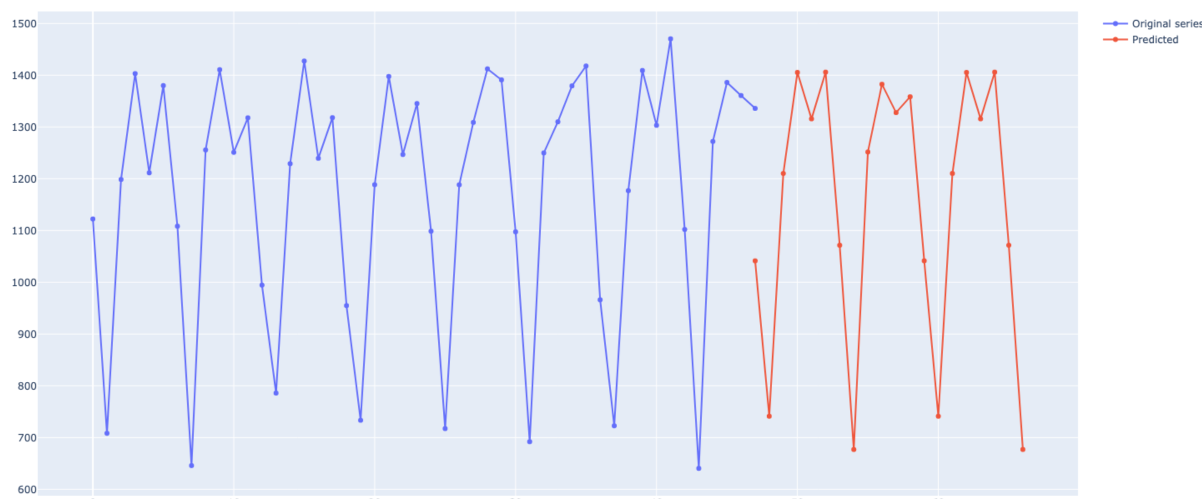

Original series
Predicted
140
135
130
125
120
115
110
105
100
0
10
20
30
40
50
60
70


Original series
Predicted
140
135
130
125
120
115
110
105
100
0
10
20
30
40
50
60
70


Original series
Predicted
0
50
100
150
200
250
300
0
20
40
60
80
100


Original series
Predicted
0
50
100
150
200
250
300
350
0
20
40
60
80
100





Notes:-

- The Graphs are shown as this because of scale variations.
- The Liner Regression Implementation is not optimal; the results can show impact.
- We are not finding good initial values for hyperparameter tuning using PAC and APAC curves; hence the results suffer while selecting the orders.

Addition in test.py files:-

Python

```python
    series = [S1,S2,S3,S4,S5]
    for series in series:
        # for arima model
        (P, D, Q), ARIMA = forecasting.ARIMA_Paramters(series) #
getting best parms and preds
        Plot(series, ARIMA)

        # for holt winters
        (Alpha, Beta, Gamma, Seasonality),HoltWinters =
forecasting.HoltWinter_Parameters(series) # getting best parms
and preds
        Plot(series, HoltWinters)
```

```python

    HoltWinters_Sample = [ # A series to test the Holt-Winters forecasting
        70, 76.60000000000001, 55.32200000000001, 37.885740000000006, 49.9868458,
        85.125464686, 95.88600759961999, 71.91134568594542, 49.053790529697416, 58.84815205311365,
        79.66504402709089, 85.68802902863068, 65.71096388270314, 45.61979921835485, 54.835723887304894,
        74.64950881982993, 80.67249382136974, 60.69542867544219, 40.60426401109391, 49.82018868004395,
        69.633973612569, 75.6569586141088, 55.67989346818124, 35.58872880383297, 44.80465347278301,
        64.61843840530805, 70.64142340684786, 50.6643582609203, 30.573193596572033, 39.78911826552206,
        59.60290319804711, 65.62588819958692, 45.64882305365936, 25.557658389311086, 34.773583058261124
    ]
    HoltWinters_Forecast = forecasting.HoltWinter_Forecast(HoltWinters_Sample, 0.7, 0.9, 0.4, 5, 25)
    Plot(HoltWinters_Sample, HoltWinters_Forecast)
    # View the plot for the forecast

    series = [S1,S2,S3,S4,S5]
    for series in series:
        # for arima model
        (P, D, Q), ARIMA = forecasting.ARIMA_Paramters(series) # getting best parms and preds
        Plot(series, ARIMA)

        # for holt winters
        (Alpha, Beta, Gamma, Seasonality),HoltWinters = forecasting.HoltWinter_Parameters(series) # getting best parms and pr
        Plot(series, HoltWinters)

```


References:-

- https://otexts.com/fpp2/arima.html
- https://otexts.com/fpp2/AR.html
- https://otexts.com/fpp2/MA.html
- https://otexts.com/fpp2/non-seasonal-arima.html
- https://otexts.com/fpp2/expsmooth.html
- https://otexts.com/fpp2/holt-winters.html
- https://www.youtube.com/watch?v=ltXSoduiVwY

- https://stackoverflow.com/questions/50785479/holt-winters-time-series-forecasting-with-statsmodels
- https://www.statsmodels.org/dev/generated/statsmodels.tsa.arima.model.ARIMA.html
- https://www.statsmodels.org/dev/generated/statsmodels.tsa.holtwinters.ExponentialSmoothing.html