ECE 592 005 – IOT Analytics

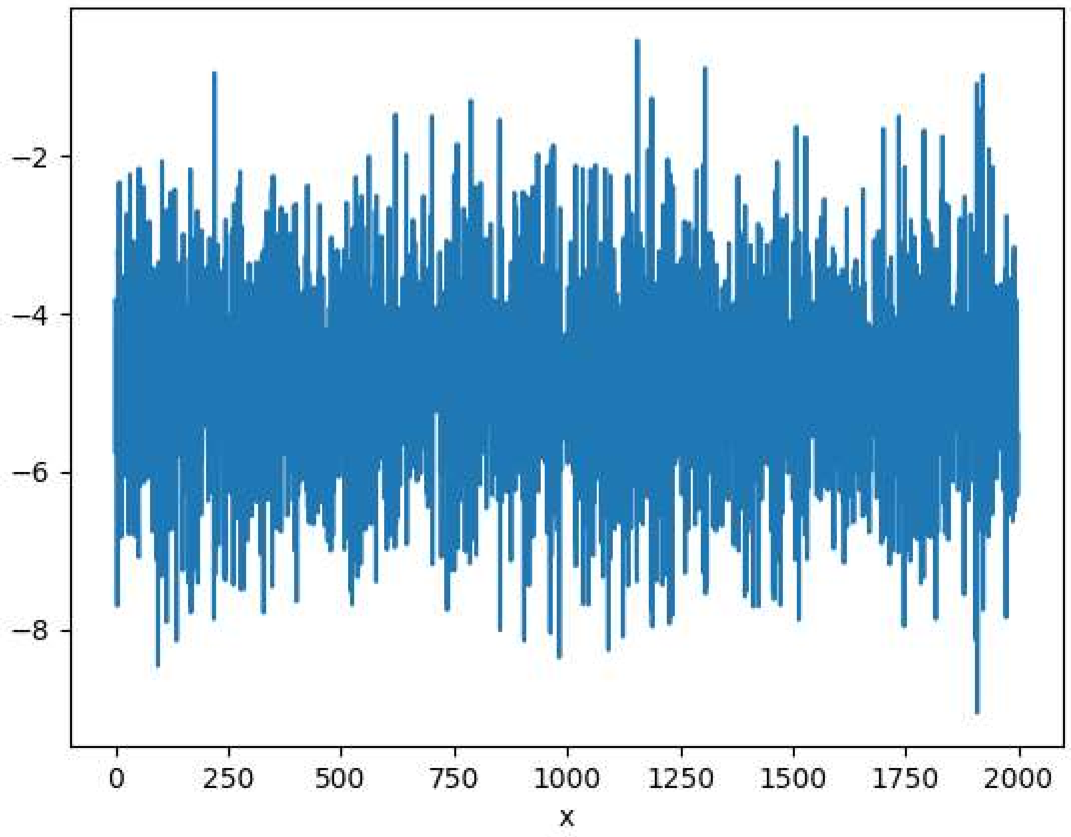
Project 3 – Forecasting

-Ragavi Swarnalatha Raman(rraman2)

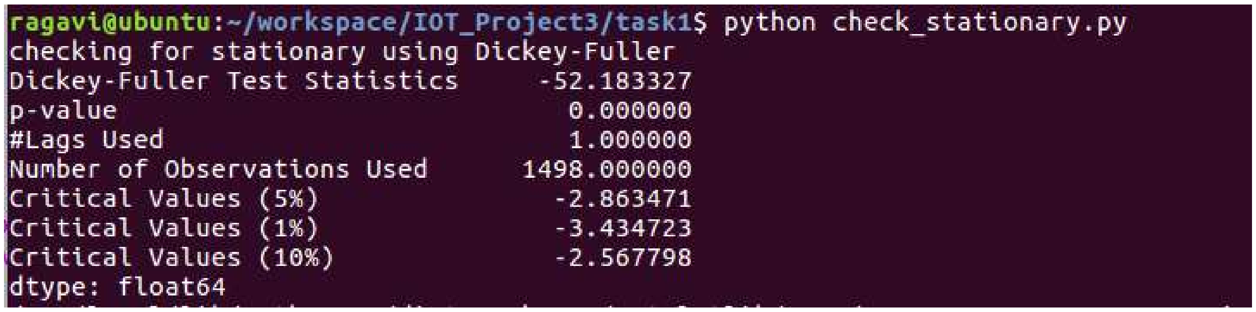
Task-1: Checking for stationarity:

A time series is said to be stationary if it has no trends, variable variance and seasonal patterns and values like mean and variance are consistent for a given time period. We use the entire data set to plot and check for stationarity, and do transformations if necessary.

Dataset Plot:



Visually inspecting the dataset plot, we do not see any trend or seasonal patterns as such. Using this we can predict that the data is stationary. To confirm the same, we so a statistical test. We use the Augmented Dickey-Fuller Test and get the pvalue and critical values.



```
ragavi@ubuntu:~/workspace/IOT_Project3/task1$ python check_stationary.py
checking for stationary using Dickey-Fuller
Dickey-Fuller Test Statistics     -52.183327
p-value                             0.000000
#Lags Used                          1.000000
Number of Observations Used      1498.000000
Critical Values (5%)               -2.863471
Critical Values (1%)               -3.434723
Critical Values (10%)              -2.567798
dtype: float64
```

According to the test,

H0: if the time series can be represented by a unit root,then it is non-stationary.

H1:null hypothesis is rejected. The time series does not have a unit root to represent it, thus it is stationary.

p-value <= 0.05 : Reject the null hypothesis. The time series is stationary.

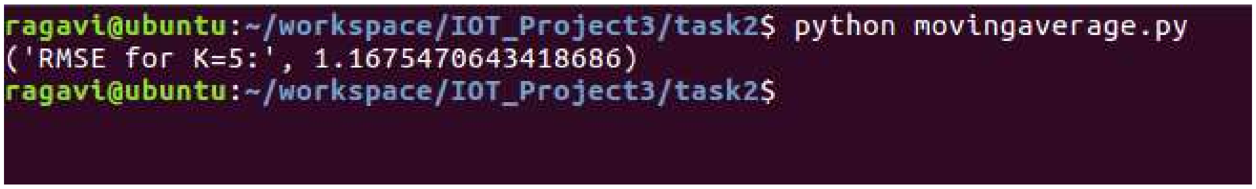
Pvalue > 0.05 : Accept the null hypothesis. The time series is nonstationary.

From the image above we see that the series is stationary as it has a p-value of 0.

Task-2: Simple Moving Average Model:

We use the training set (first 1500 observations) to derive a Simple Moving Average model.

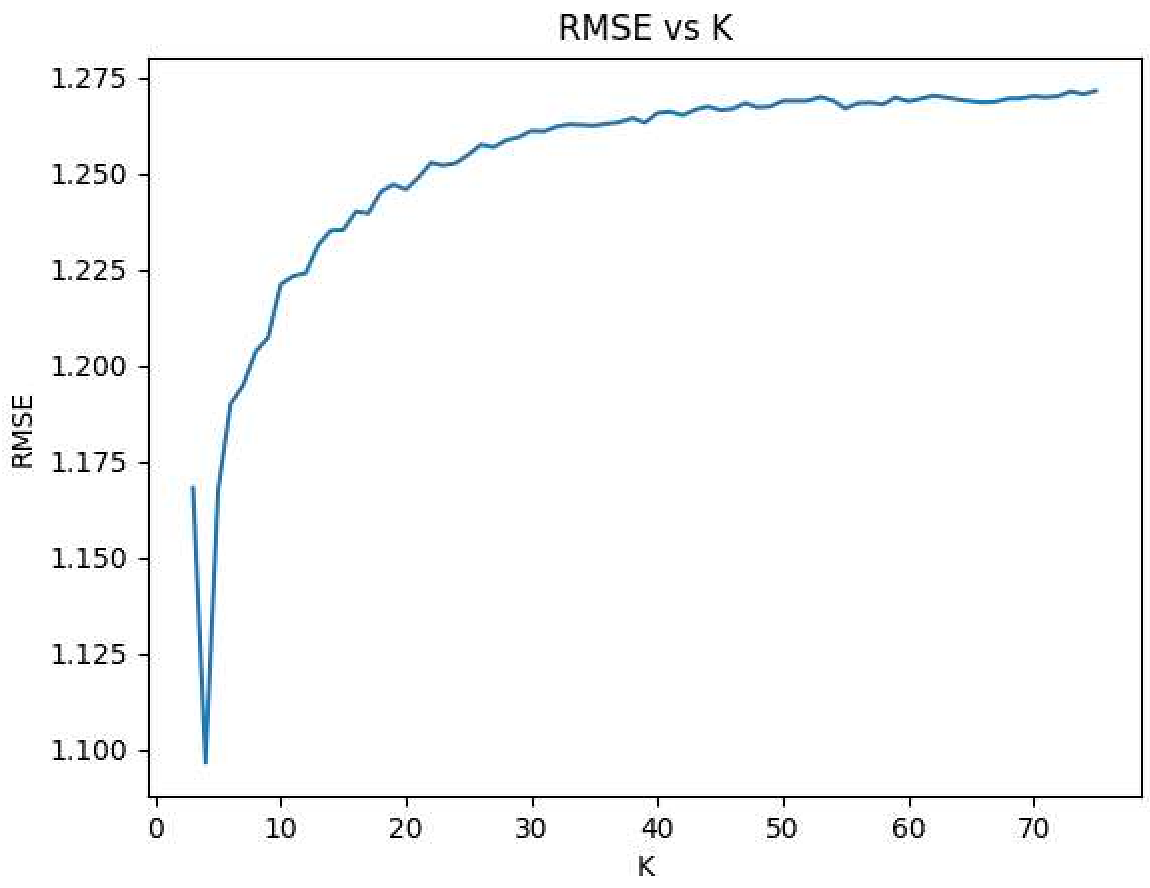
Initially, the K value is fixed to be 5 and the predicted/fitted values are calculated by having a window of 5. The below image shows the RMSE value for a window of K=5.



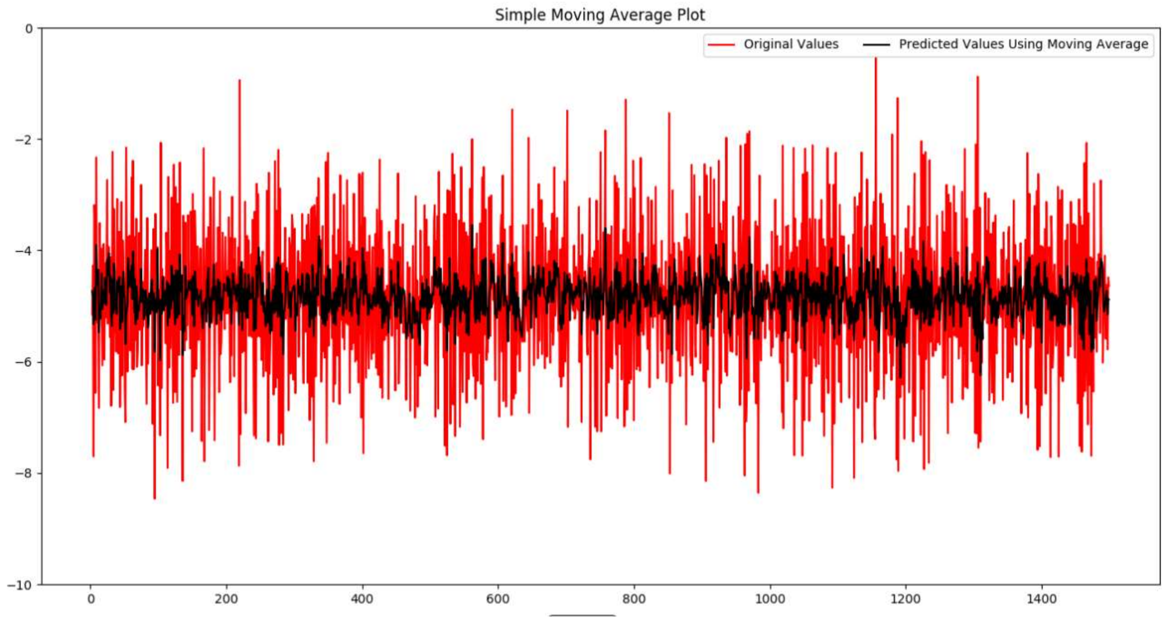
```
ragavi@ubuntu:~/workspace/IOT_Project3/task2$ python movingaverage.py
('RMSE for K=5:', 1.1675470643418686)
ragavi@ubuntu:~/workspace/IOT_Project3/task2$
```

We repeat the same and fit values for the original values for a range of K, we start from K=3 which is the minimum window and proceed up to K=70. We calculate the RMSE value for all K values and calculate the optimal K value using the minimum RMSE.

RMSE vs K plot:



The best value for K using minimum RMSE is 4. For K=4, we plot the original values vs fitted values(predicted).



1.5 The simple moving average model that we use is based on trailing values. The average is calculated based on the previous few window values. For example, a trailing moving average with a window of k=4, We calculate each predicted value as ,

Predictedval(n) = mean(data(n-3), data(n-2), data(n-1), data(n))

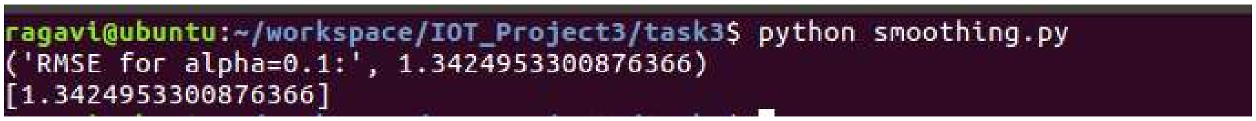
As the next value is dependent on the previous n-1 set of values, and in a walk forward manner we do not get an accurate set of fitted/predicted values always.

The plot of the forecasted values show that they lie mostly around the average of the total data set observation range as it is based on the average of a window of values considered. Hence, this model is always less accurate.

Task3: Exponential Smoothing Model:

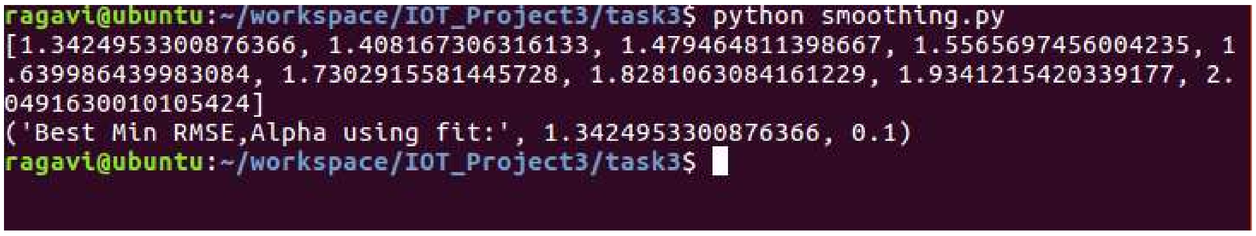
We use the training set (first 1500 observations) to derive a Simple Exponential Smoothing Model. Ie) the next value is based on the just one previous observation.

Initially, the alpha value is fixed to be 0.1 and the predicted/fitted values are calculated The below image shows the RMSE value for a window of alpha=0.1



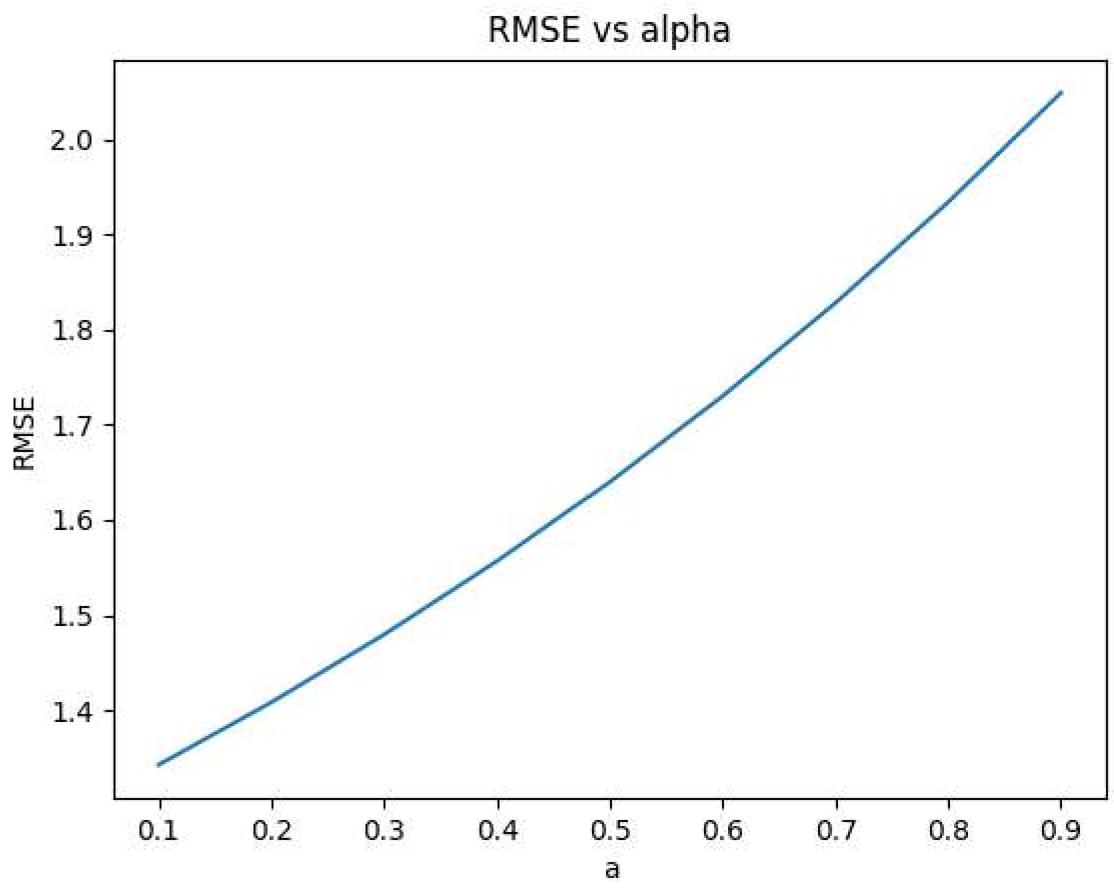
```
ragavi@ubuntu:~/workspace/IOT_Project3/task3$ python smoothing.py
('RMSE for alpha=0.1:', 1.3424953300876366)
[1.3424953300876366]
```

Now,we increase alpha by 0.1 each time and proceed until alpha=0.9. From the predicted values, we calculate the RMSE vales for all alpha values and get the best alpha value based on the lowest RMSE value.



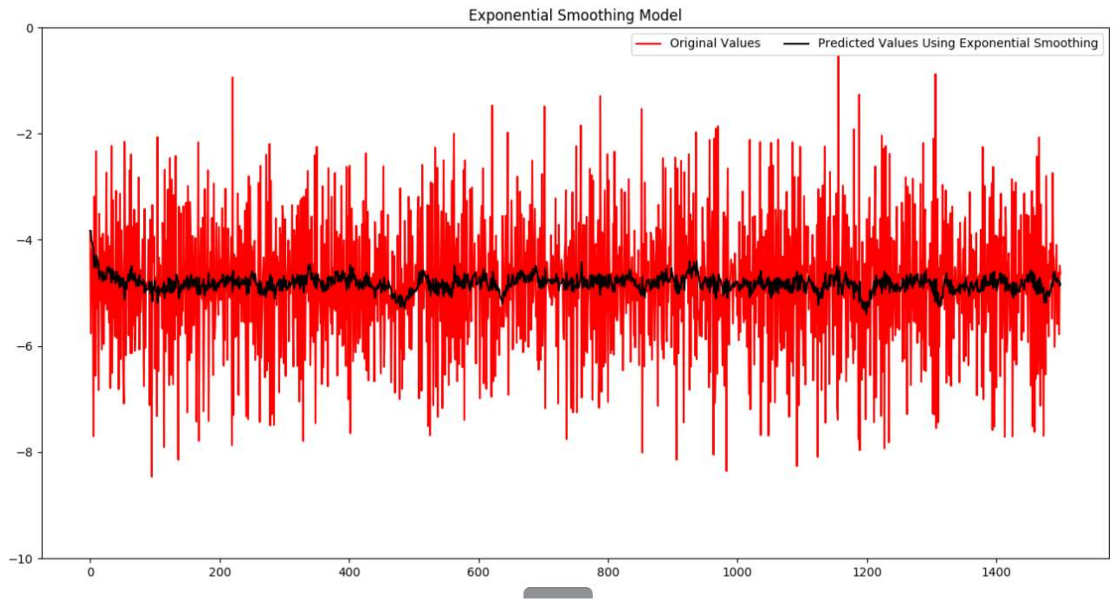
```
ragavi@ubuntu:~/workspace/IOT_Project3/task3$ python smoothing.py
[1.3424953300876366, 1.408167306316133, 1.479464811398667, 1.5565697456004235, 1
.639986439983084, 1.7302915581445728, 1.8281063084161229, 1.9341215420339177, 2.
0491630010105424]
('Best Min RMSE,Alpha using fit:', 1.3424953300876366, 0.1)
ragavi@ubuntu:~/workspace/IOT_Project3/task3$
```

Plot of RMSE vs a:



For alpha=0.1, we get the lowest RMSE value.

For alpha=0.1, we plot the original values vs fitted values(predicted).



2.6 The simple exponential smoothing is a naïve model that gives importance only to the most recent observation, with respect to the formula, $s_t = ax_{t-1} + (1 - a)s_{t-1}$

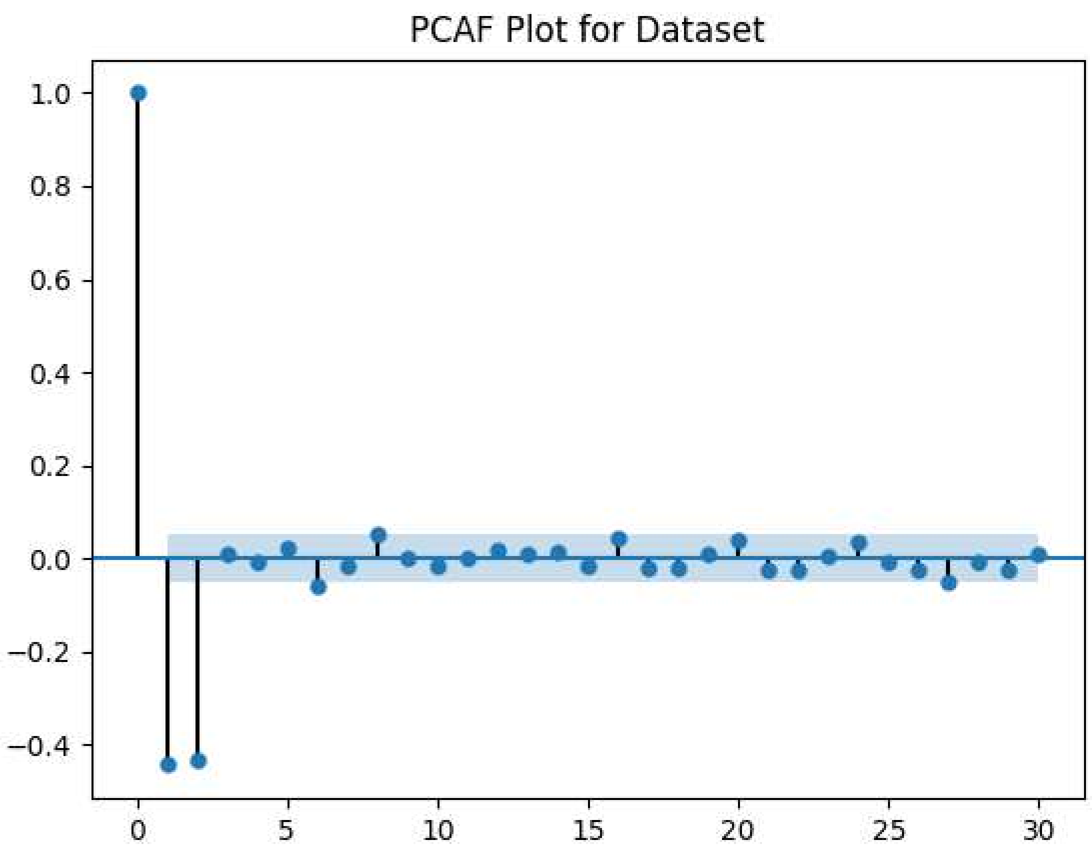
This again is not quite an accurate model nor an optimized one as observed in the image above that plots the original vs predicted values. The predicted values form a very thin streak over the original values. Other higher order exponential smoothing techniques like Double or Triple Exponential Smoothing might consider greater parameters to predict values and hence might produce better predicted results.

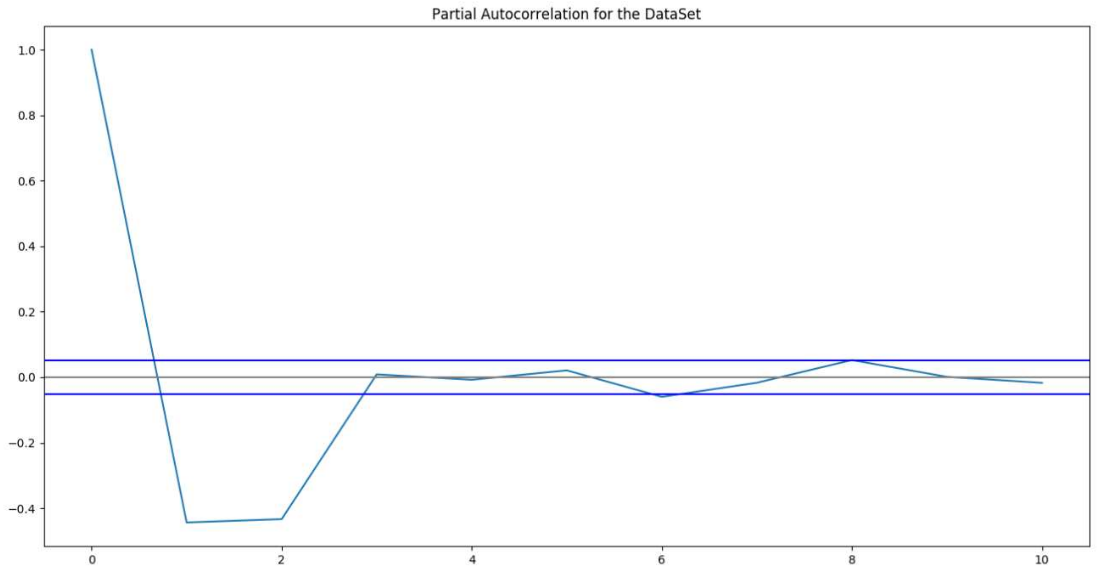
Task4:AR Model:

The AR model is solely based on the auto-regressive terms. These auto-regressive terms are lags of the dependent variable of the dataset and they will be used as predictors. These auto-regressive terms are also called p value and the model is given AR(p).

Partial Auto-Correlation plot (PACF) is used to determine the lag K at which PACF cuts off

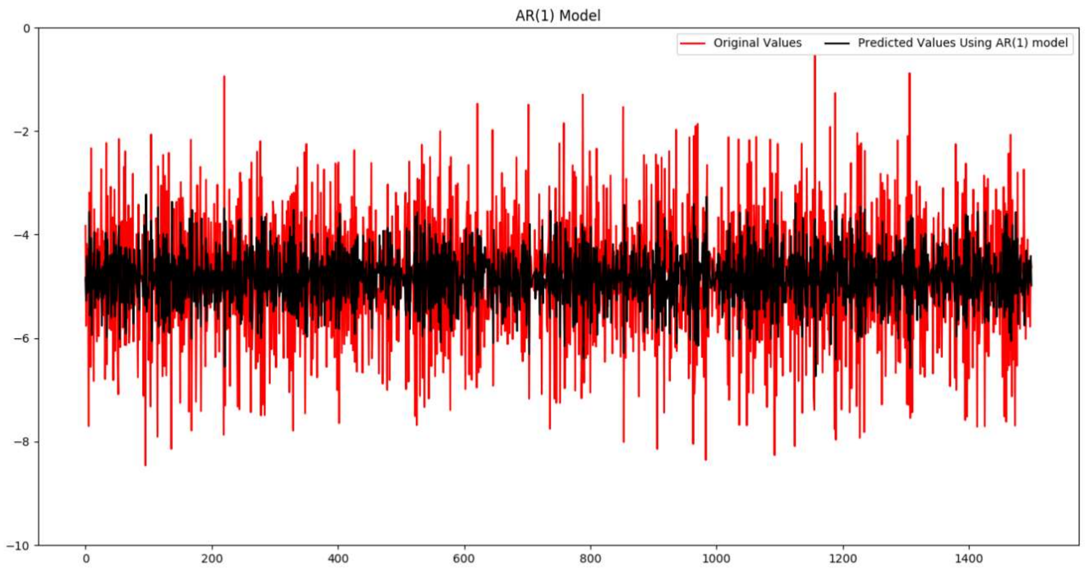
Ie) The point where the plot crosses the upper confidence interval for the first time. On our dataset, we see that cut off point to be 1.

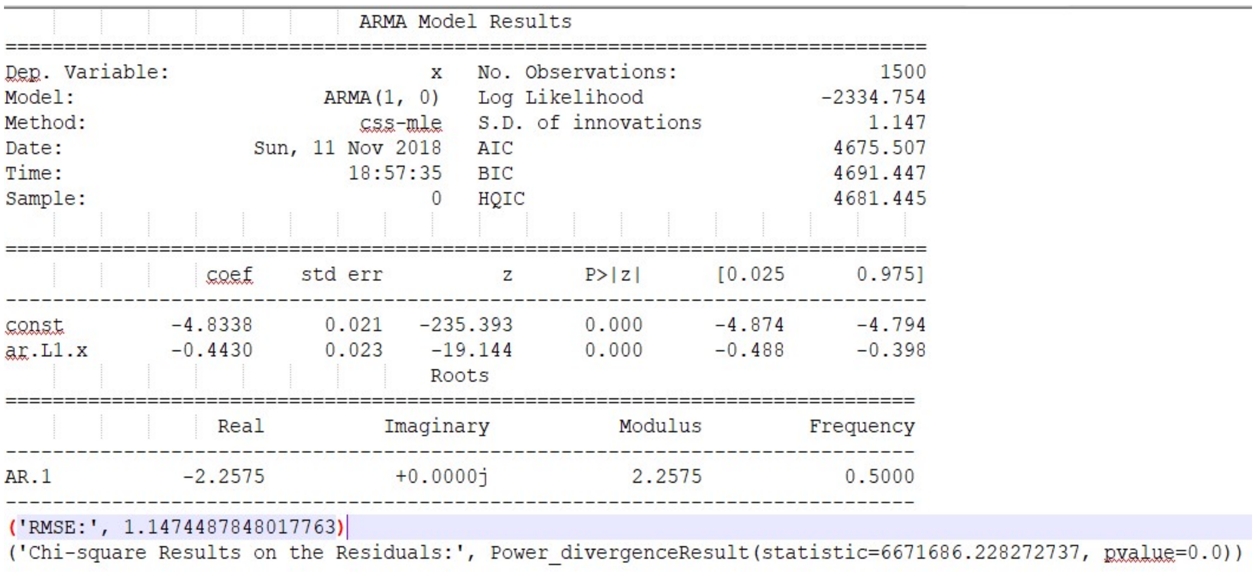

PCAF Plot for Dataset
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
0
5
10
15
20
25
30


Partial Autocorrelation for the DataSet
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
0
2
4
6
8
10

Hence, the AR(1) model is chosen. We use the ARIMA module with p=1,d =0 and q=0.

A plot on the original training set vs the predicted values is given below,





```
                              ARMA Model Results
==============================================================================
Dep. Variable:                      x   No. Observations:                 1500
Model:                     ARMA(1, 0)   Log Likelihood               -2334.754
Method:                       css-mle   S.D. of innovations              1.147
Date:                Sun, 11 Nov 2018   AIC                           4675.507
Time:                        18:57:35   BIC                           4691.447
Sample:                             0   HQIC                          4681.445

==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const         -4.8338      0.021   -235.393      0.000      -4.874      -4.794
ar.L1.x       -0.4430      0.023    -19.144      0.000      -0.488      -0.398
                                    Roots
=============================================================================
                 Real          Imaginary           Modulus         Frequency
-----------------------------------------------------------------------------
AR.1          -2.2575           +0.0000j            2.2575            0.5000
-----------------------------------------------------------------------------
('RMSE:', 1.1474487848017763)
('Chi-square Results on the Residuals:', Power_divergenceResult(statistic=6671686.228272737, pvalue=0.0))
```

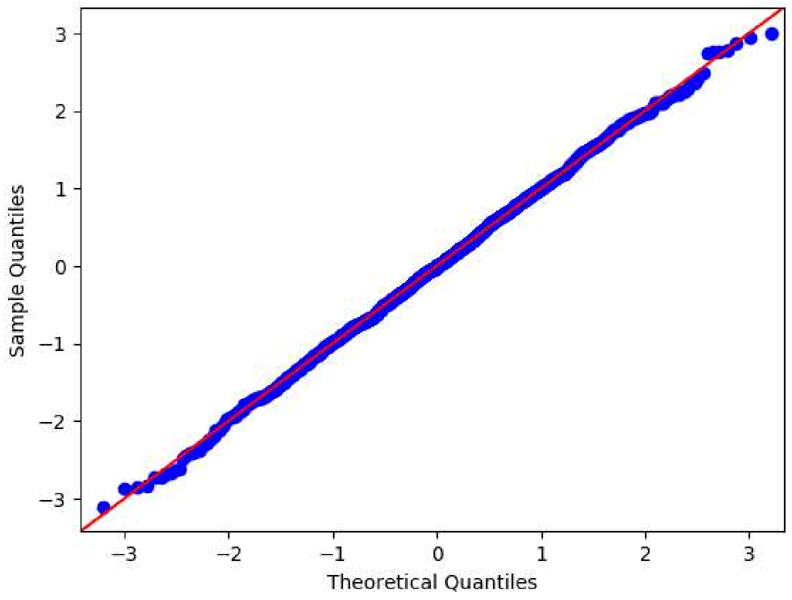
RMSE value for the AR(1) model : 1.1474487848017763

The above image also gives a summary of the ARMA(1,0) model.

Residual Analysis:

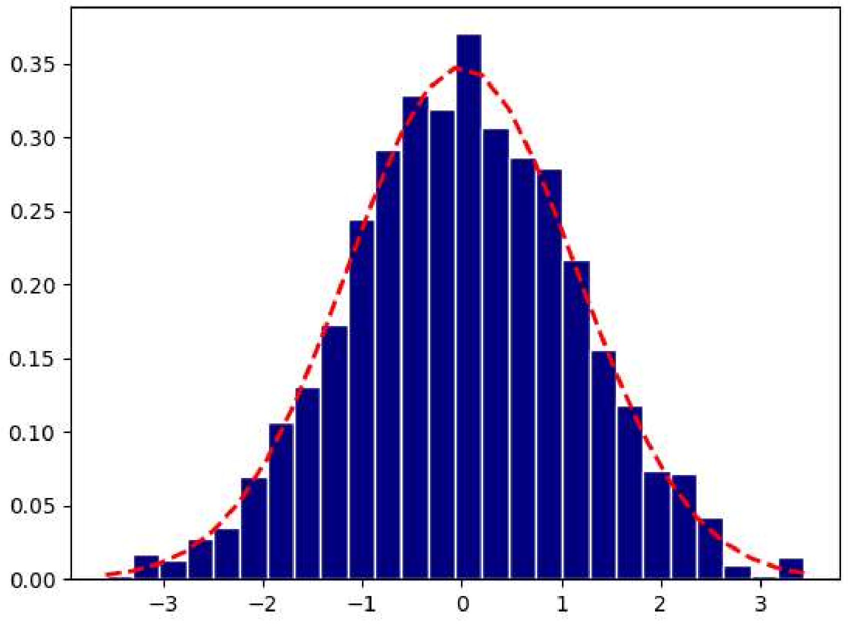
From the predicted model, we calculate the list of residuals and do several tests to verify the validity of the model.

Q-Q Plot:



The Q-Q plot directly lies over the line and hence can be seen as a good fit and follows a distribution.

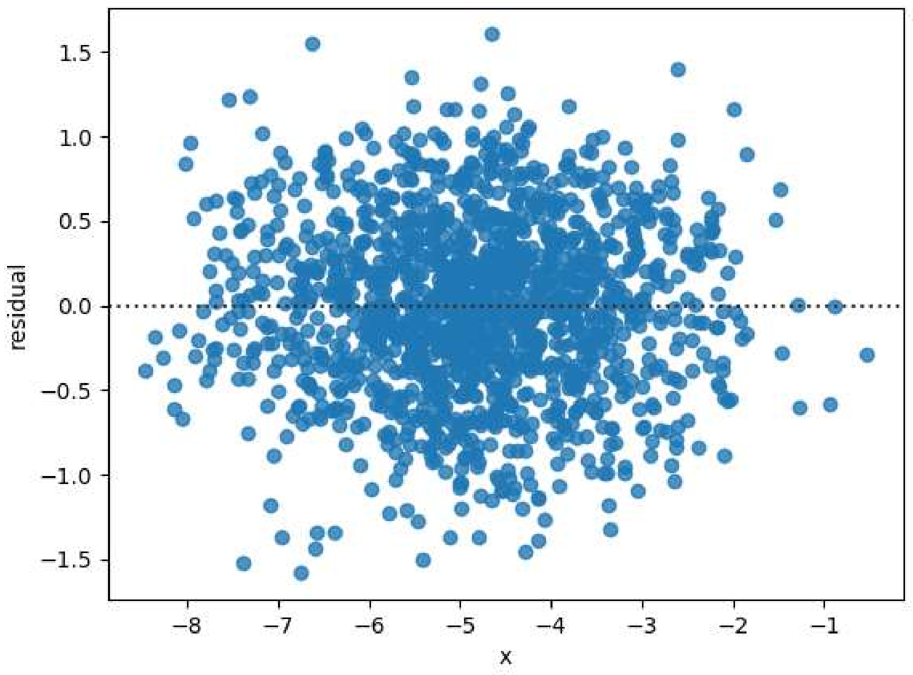
Histogram:



Chi-Square Result: Power_divergenceResult(statistic=6671686.228272737, pvalue=0.0))

The histogram and the chi-square results clearly shows that the residuals are normally distributed .

Scatter Plot:



The scatter plot has no trends which is an important criterion for time series residuals.

3.4 The AR(p) model tends to capture serial correlation present in the time series and try to capture them while predicting values. On using the AR model with a p value 1, we observe a better plot on the original vs fitted values of the dataset. The model can hence be said to be better accurate than the simple moving average or exponential smoothing techniques used earlier. Further, on getting the residuals out of the fitted model and we see that the residuals follow a normal distribution, give a good fit on the q-q plot and show no trends on a scatter plot.

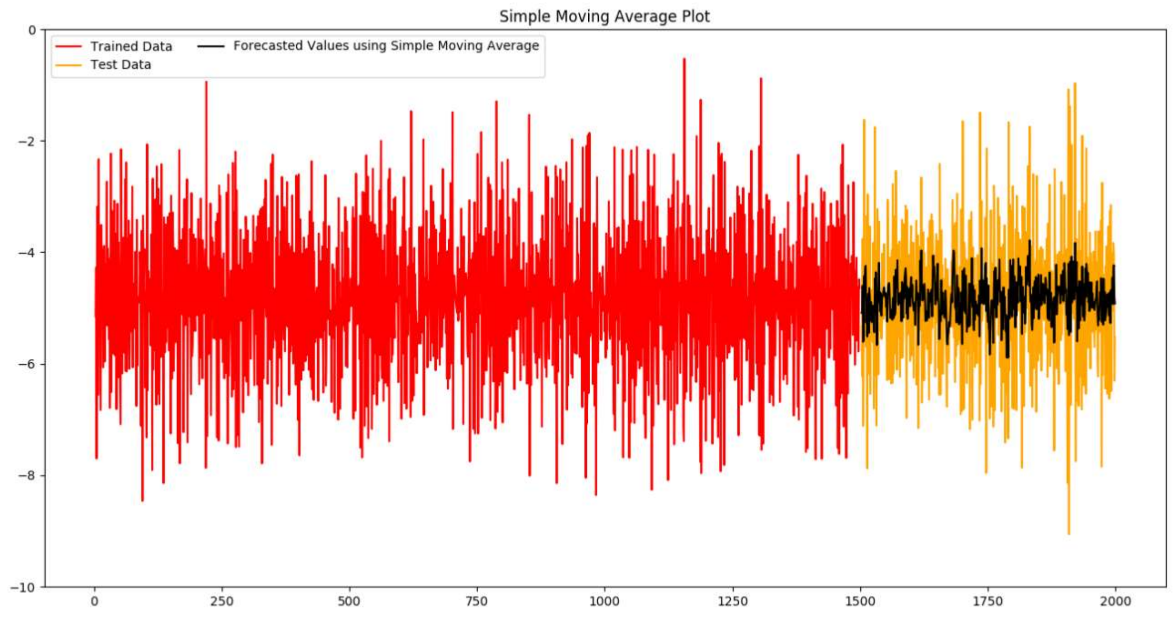
The AIC value from the model summary gives a notion of how well the model fits the data. We do not get a lower AIC score. Hence even though AR(1) fitted value plot is slightly better than the other two, we cannot come to conclusion that this model is necessarily good for the given dataset.

Task5: Comparison of all models:

We had previously separated the entire data set into a 1500 value set known as training set and a 500 value set known as test data. The above models were all performed on the training data.

We basically train the model on the first 1500 values, find fitted values and then forecast into the future by making the model forecast on the train data. We take the RMSE values here based on the difference between the original and the forecasted values and decide on the best model.

Simple Moving Average Plot: (K=4)

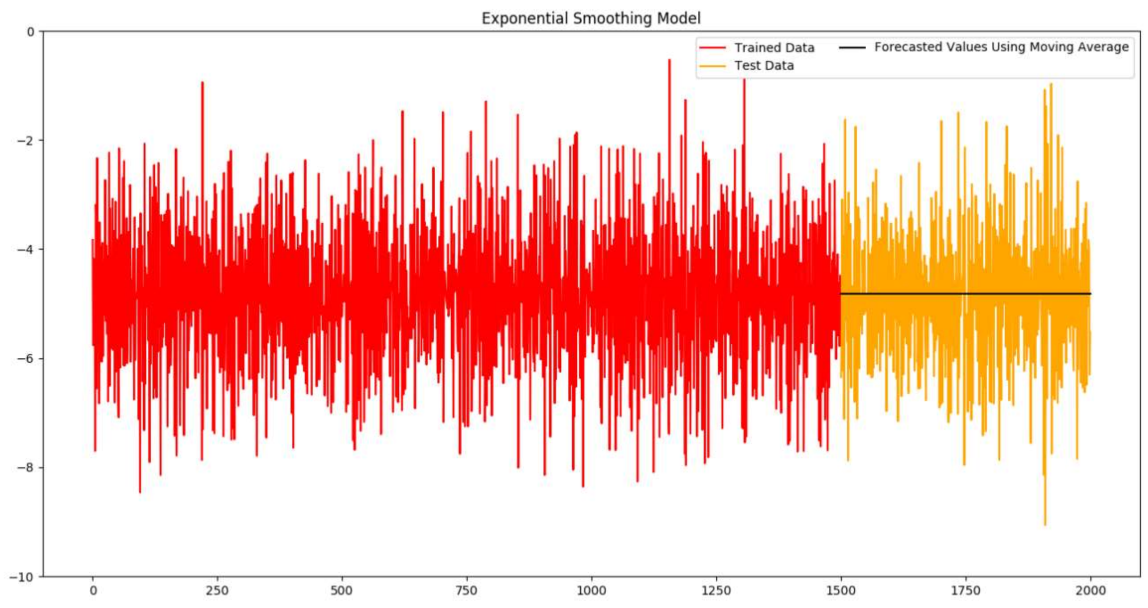


The plot of the forecasted values show that they lie mostly around the average of the total data set observation range as it is based on the average of a window of values considered. We see a good forecasting on the test data, but not very accurate.

('RMSE for Moving Average Model Forecasted Values:', 1.0645150415940963)

('Best Min RMSE,K:', 1.0965410683105459, 4)

Simple Exponential Smoothing:

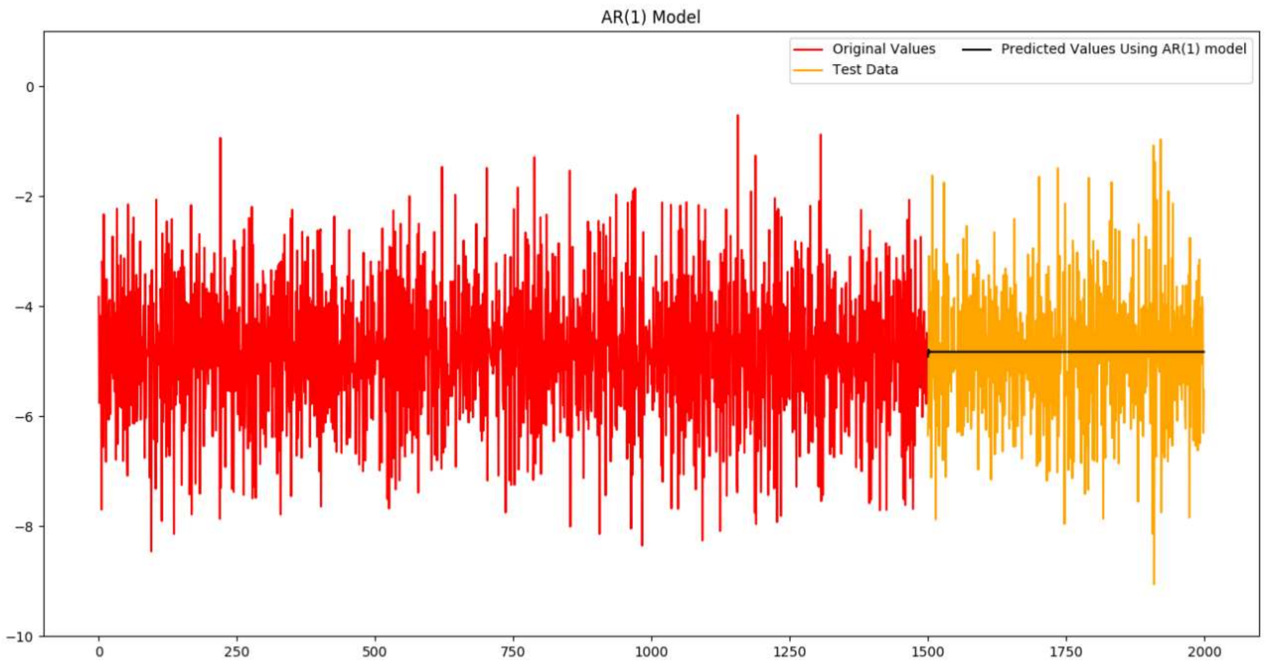


Since every future observation is majorly based on the most recent observation, we get a plot that is almost a straight line ie) the values that is forecasts are almost the same.

```
('Best Min RMSE,Alpha using fit:', 1.3424953300876366, 0.1)
```

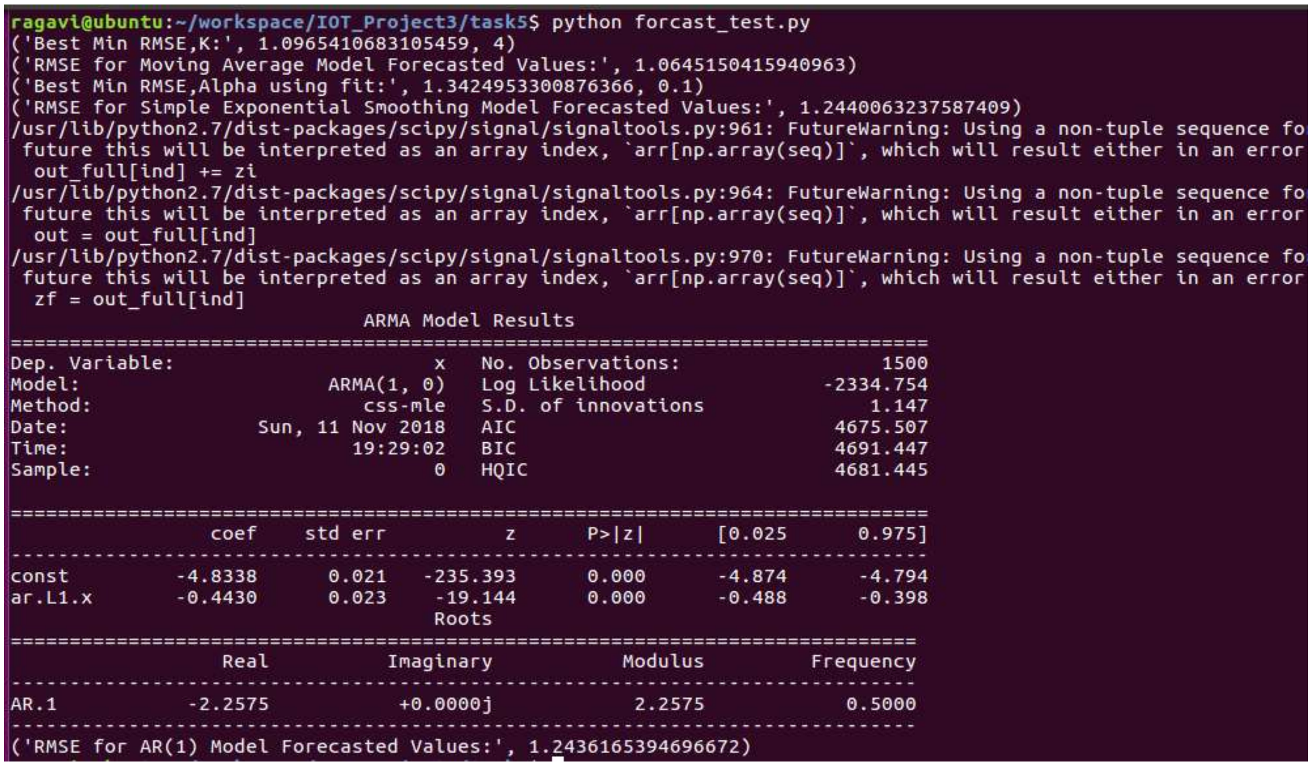
```
('RMSE for Simple Exponential Smoothing Model Forecasted Values:', 1.2440063237587409)
```

AR(1) Model:



The AR(1) model is based on a predictor lag of 1. Even though the fitted values on the training data gave a better plot which seemed a slightly better fit than the other two models, forecasting using the testing data gives a plot very similar to the exponential smoothing model. This might because of the lag considered to be just 1 and not considering any other parameters while doing a model forecasting. Other models like ARIMA (combined) might forecast better values in this case.

A summary of the forecasting done using all the models. Based on the minimum RMSE value alone, we find that the moving average model itself is the best for the given data set.



```
ragavi@ubuntu:~/workspace/IOT_Project3/task5$ python forcast_test.py
('Best Min RMSE,K:', 1.0965410683105459, 4)
('RMSE for Moving Average Model Forecasted Values:', 1.0645150415940963)
('Best Min RMSE,Alpha using fit:', 1.3424953300876366, 0.1)
('RMSE for Simple Exponential Smoothing Model Forecasted Values:', 1.2440063237587409)
/usr/lib/python2.7/dist-packages/scipy/signal/signaltools.py:961: FutureWarning: Using a non-tuple sequence fo
 future this will be interpreted as an array index, `arr[np.array(seq)]`, which will result either in an error
  out_full[ind] += zi
/usr/lib/python2.7/dist-packages/scipy/signal/signaltools.py:964: FutureWarning: Using a non-tuple sequence fo
 future this will be interpreted as an array index, `arr[np.array(seq)]`, which will result either in an error
  out = out_full[ind]
/usr/lib/python2.7/dist-packages/scipy/signal/signaltools.py:970: FutureWarning: Using a non-tuple sequence fo
 future this will be interpreted as an array index, `arr[np.array(seq)]`, which will result either in an error
  zf = out_full[ind]
                              ARMA Model Results
==============================================================================
Dep. Variable:                      x   No. Observations:                 1500
Model:                     ARMA(1, 0)   Log Likelihood               -2334.754
Method:                       css-mle   S.D. of innovations              1.147
Date:                Sun, 11 Nov 2018   AIC                           4675.507
Time:                        19:29:02   BIC                           4691.447
Sample:                             0   HQIC                          4681.445

==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const         -4.8338      0.021   -235.393      0.000      -4.874      -4.794
ar.L1.x       -0.4430      0.023    -19.144      0.000      -0.488      -0.398
                                    Roots
=============================================================================
                  Real          Imaginary           Modulus         Frequency
-----------------------------------------------------------------------------
AR.1           -2.2575           +0.0000j            2.2575            0.5000
-----------------------------------------------------------------------------
('RMSE for AR(1) Model Forecasted Values:', 1.2436165394696672)
```