

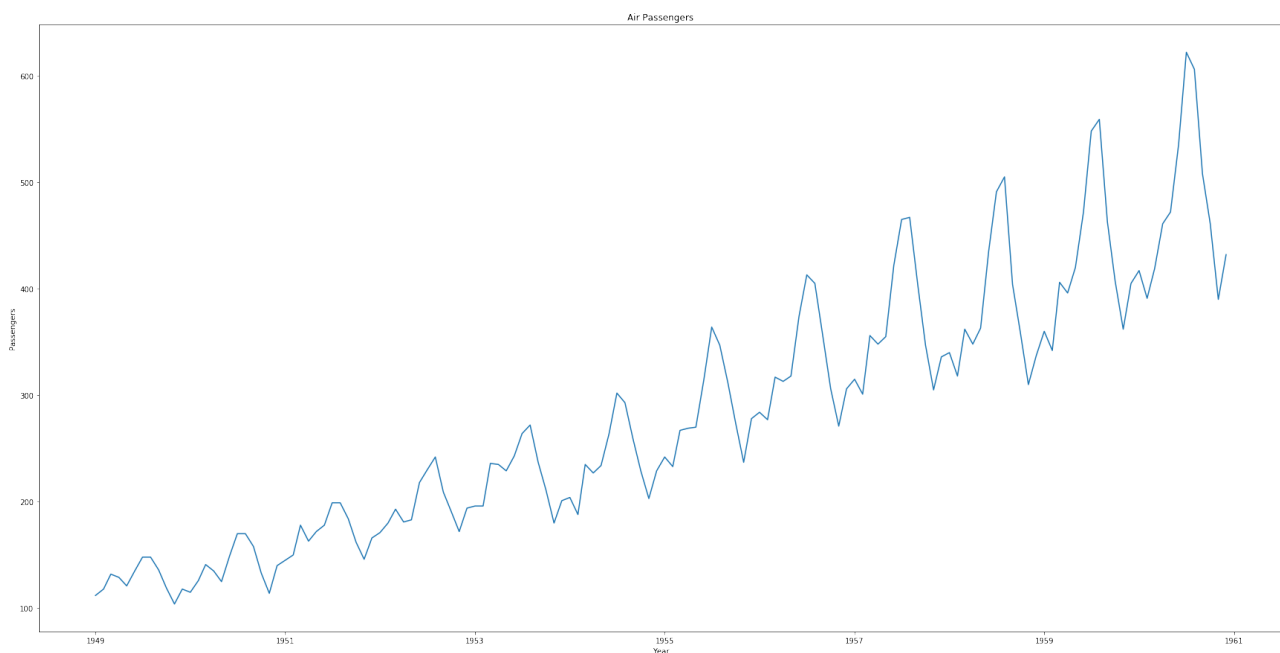
Task : Time Series Forecasting**Data :** Air Passengers Data set and Melbourne Data set**Exercise 1A: Stationary and Arima**

A time series is said to be stationary if doesn't have any seasonality or trend. We can check this in many ways. One of the ways to perform DF test. If we get the p-value less than 0.5 then the series is stationary else it is not.

And calculating the mean after dividing the data set. If there is a big change in the means then it is not stationary else it is stationary.

Basing these two assumptions the task is solved.

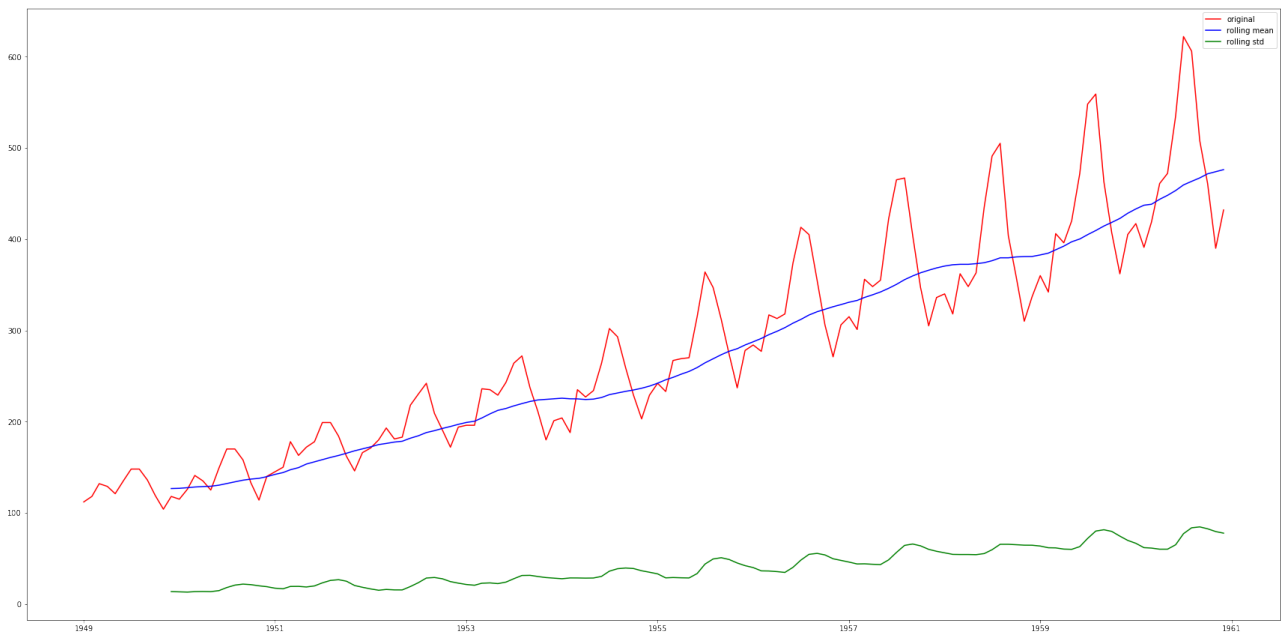
First the Air Passengers Data set is loaded. The time series is plotted. The plot obtained is



The mean is calculated on 2 halves and then the DF test is performed the result is

Mean of first part of data:	182.902777778
Mean of second part of data	377.694444444
Standard deviation of first part of data:	47.371803534
Standard deviation of second part of data	85.8368347002
Test Statistic	0.815369
p-value	0.991880
#Lags Used	13.000000
Number of Observations Used	130.000000
Critical Value (1%)	-3.481682
Critical Value (5%)	-2.884042
Critical Value (10%)	-2.578770

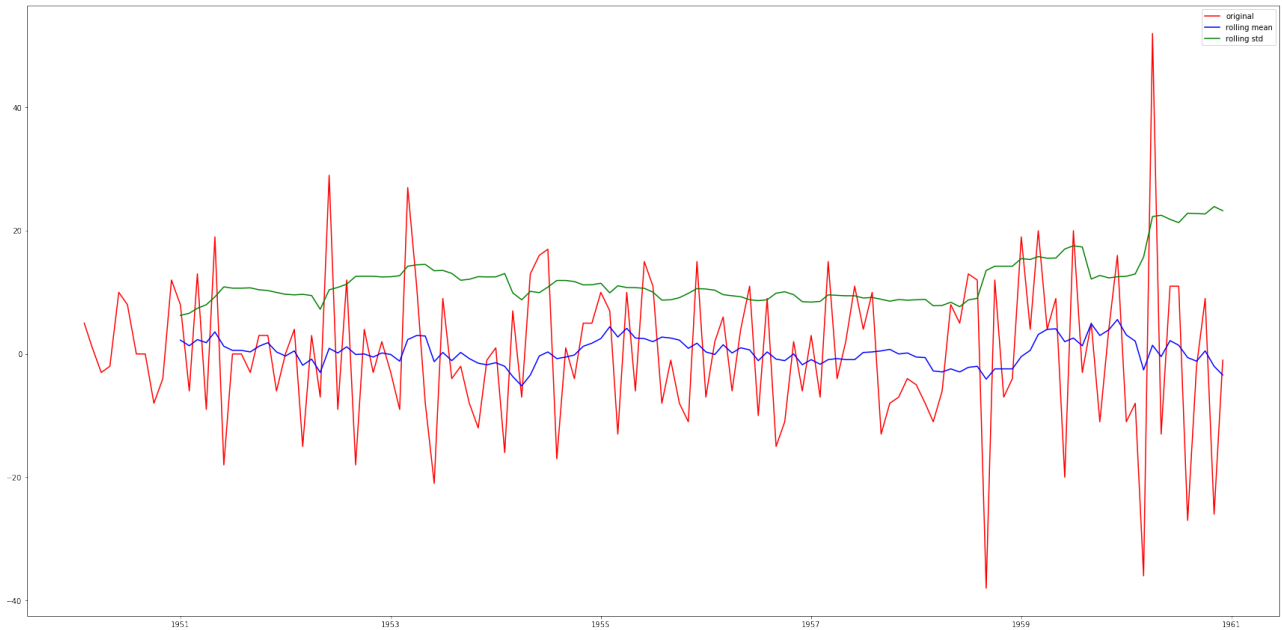
The plot is made with the rolling mean and the standard deviation. The plot is obtained like this



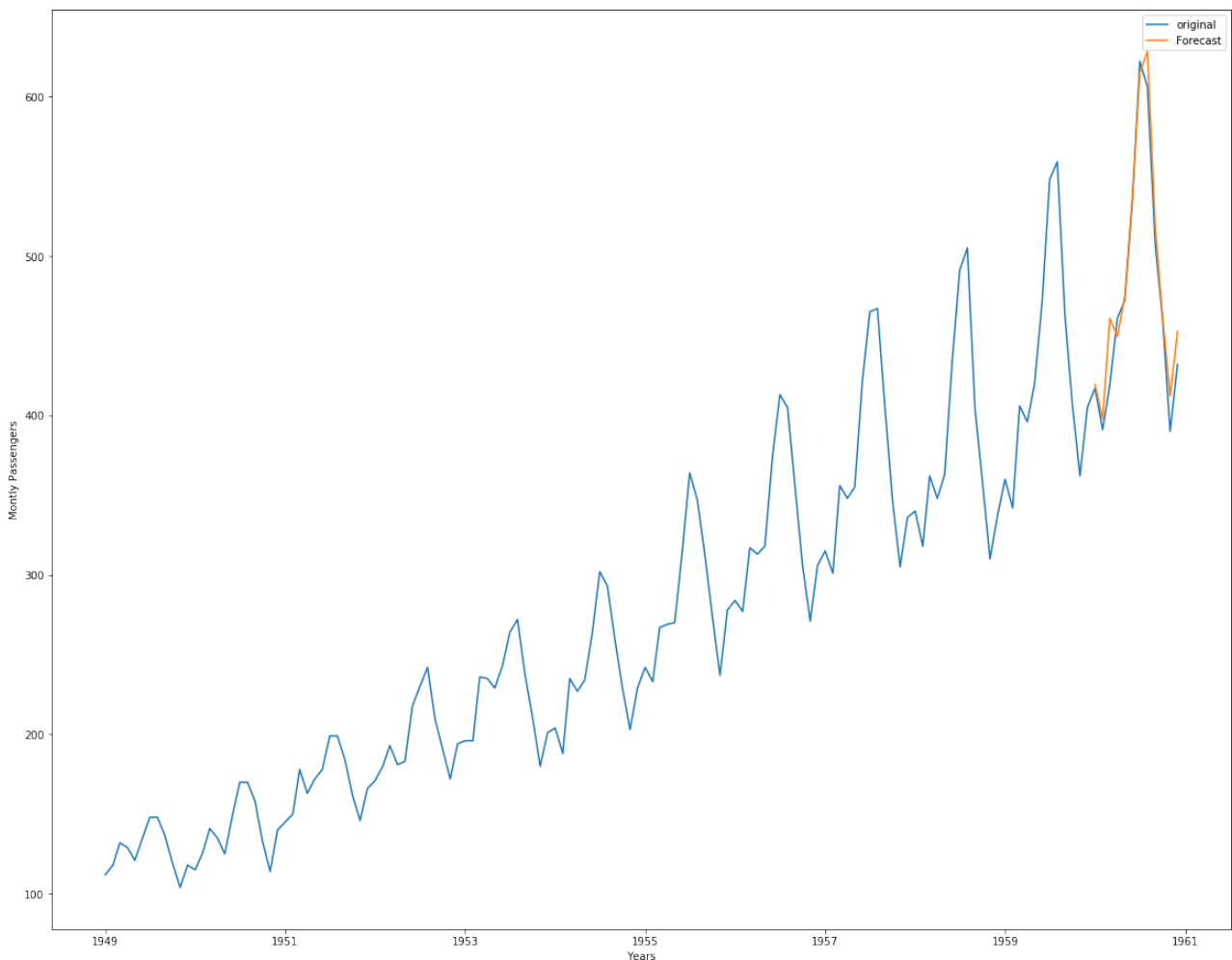
If we observe the plot and the test results we can say that the series is not stationary. To make it stationary we have to remove the trend and seasonality . After making the series stationary the DF test results are

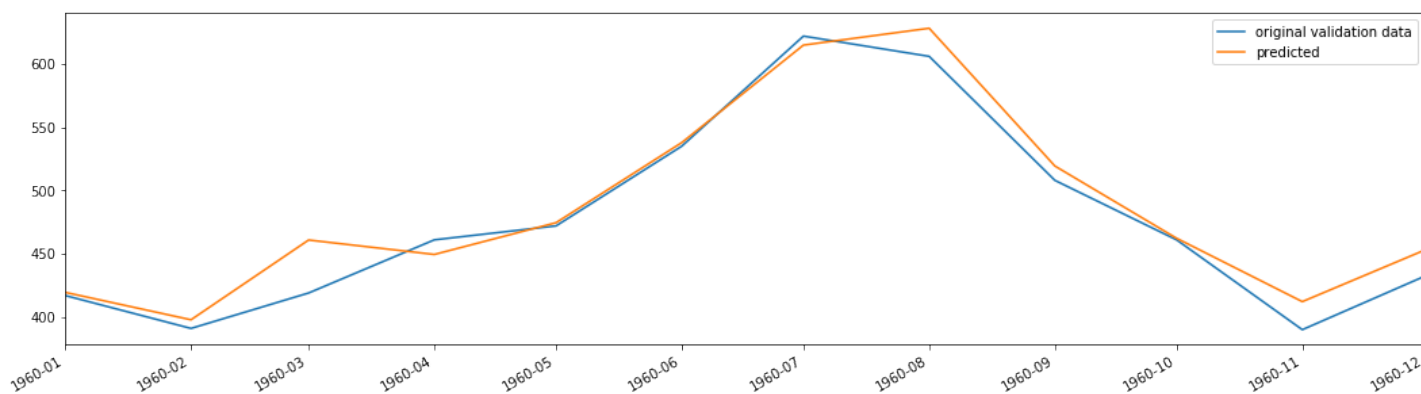
Mean of first part of data:	0.738461538462
Mean of second part of data	-0.363636363636
Standard deviation of first part of data:	10.4267067197
Standard deviation of second part of data	13.895318989
Test Statistic	-1.559562e+01
p-value	1.856512e-28
#Lags Used	0.000000e+00
Number of Observations Used	1.300000e+02
Critical Value (1%)	-3.481682e+00
Critical Value (5%)	-2.884042e+00
Critical Value (10%)	-2.578770e+00

If we observe the P-value it is below 0.5. So we can say that the series is stationary now. The plot obtained after making the series stationary is



Since the Air Passenger has seasonality **SARIMAX** is used. For the parameters a grid search is made and the best parameters are used for training the model and then the forecast is made and the rmse is calculated. A graph is plotted for the forecasted data and the original data.

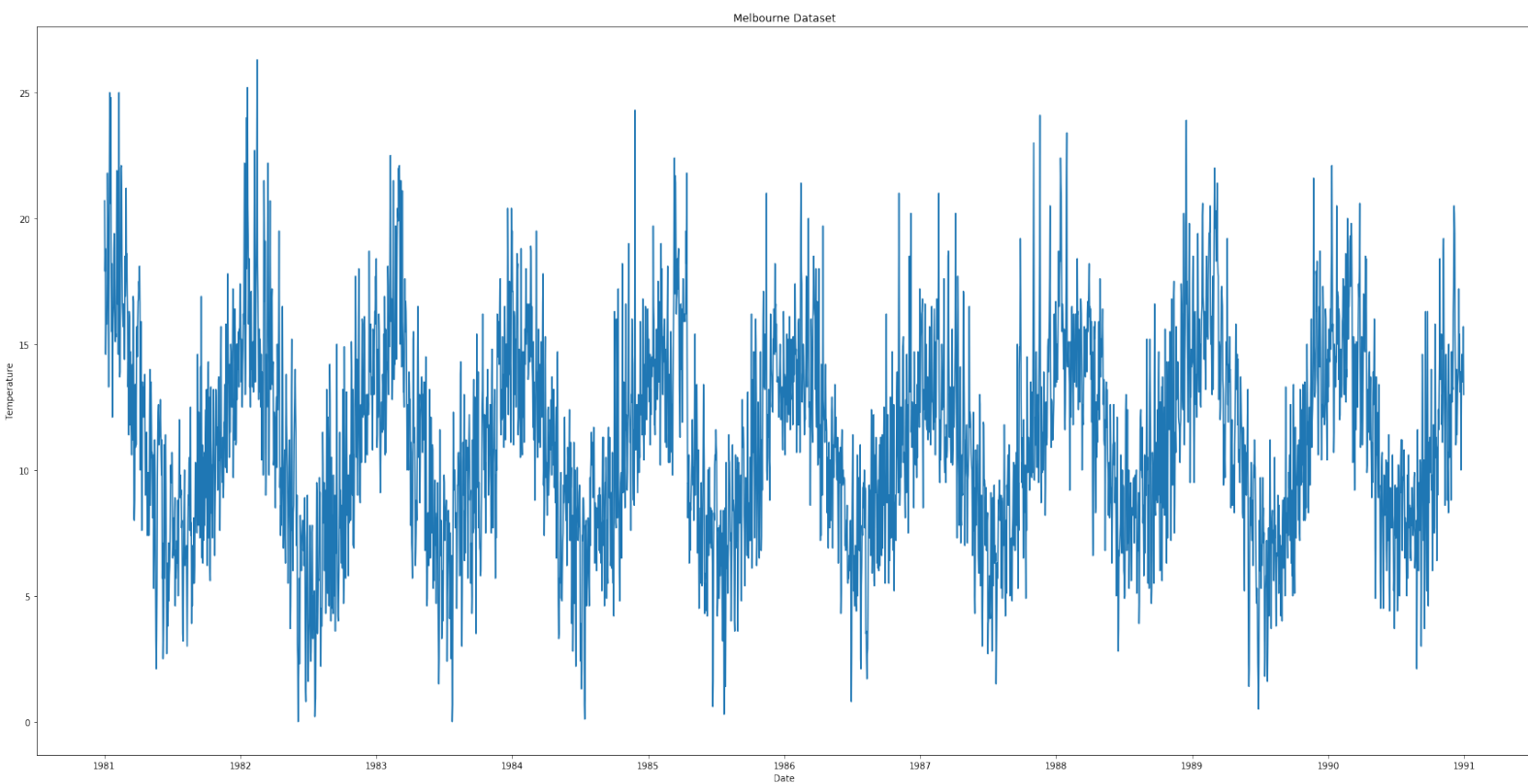




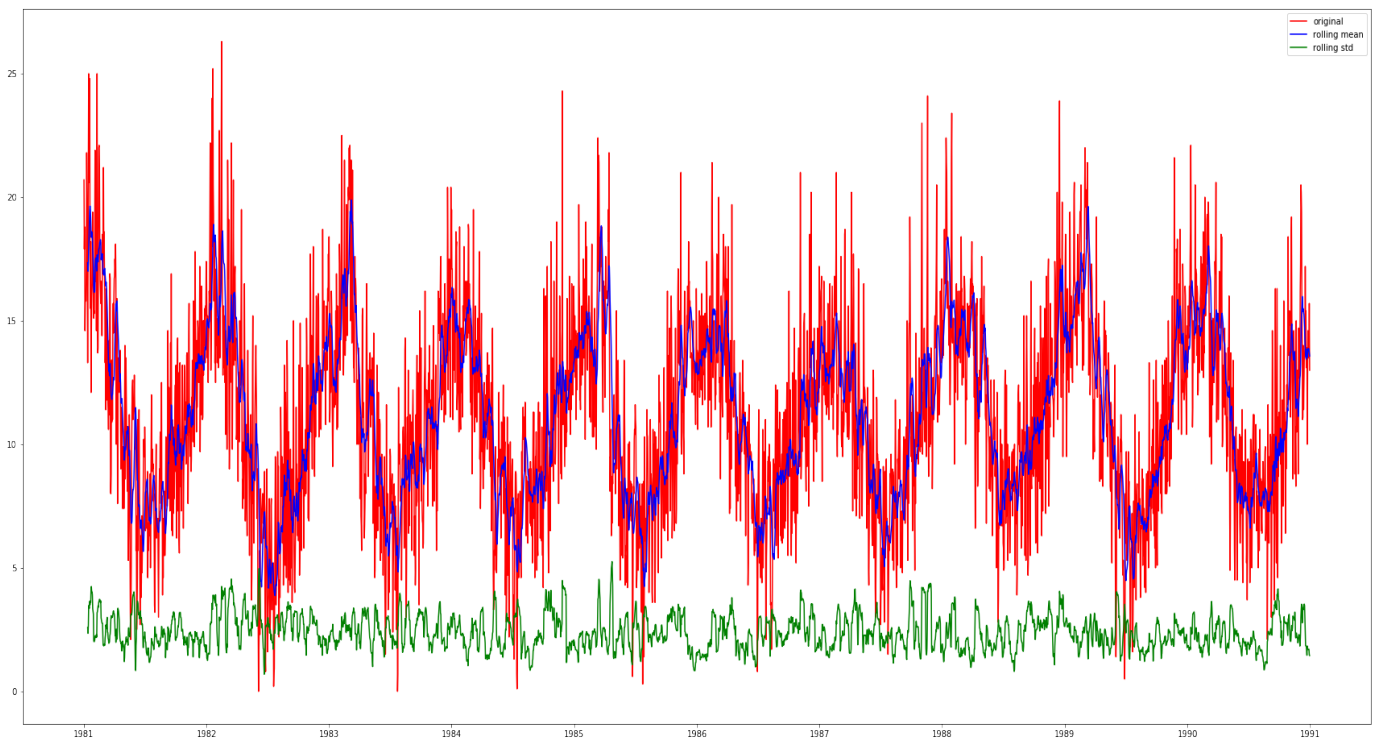
The RMSE is calculated for forecasts and it is 17.187245320475341.

Melbourne Data set

The other data set which is given for the task is melbourne data set. First the data set is plotted.



The plot after plotting the mean and the standard deviation is



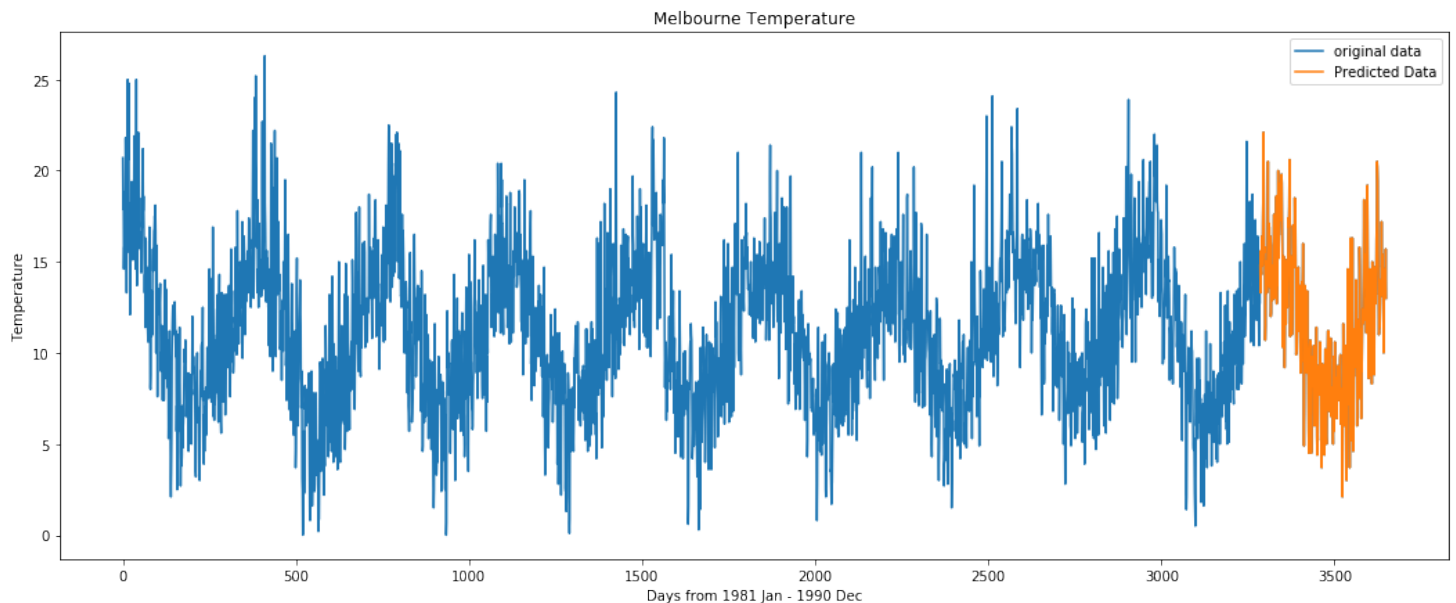
The results for the DF test are given below

```
Mean of first part of data: 11.0435068493
Mean of second part of data 11.312
Standard deviation of first part of data: 4.26155194293
Standard deviation of second part of data 3.86700884016
Test Statistic          -4.444805
p-value                0.000247
#Lags Used              20.000000
Number of Observations Used 3629.000000
Critical Value (1%)      -3.432153
Critical Value (5%)      -2.862337
Critical Value (10%)     -2.567194
```

If we observe the p-value it is less than 0.5 and the difference in the means are less when compared. So the given time series is stationary and we can apply ARIMA for forecasting.

The parameters for the ARIMA are found by grid search and then the parameters are used for training the model and then the model is used to make the forecasts and then the RMSE is calculated and the plots are made for the forecasted data.

The RMSE obtained is 3.69205616048



Task 2: Tensor Flow

Exercise : Logistic Regression on the Olivetti faces data set

1. The **olivetti faces** data set is a data set that contains a set of face images taken between April 1992 and April 1994 at AT&T Laboratories Cambridge.
2. The data set has ten different images of each of 40 distinct subjects. The image is quantized to 256 grey levels and stored as unsigned 8 bit integers, the loader will convert these to floating point values on the interval [0,1].
3. The target has 0-39 indicating identity of person.
4. The data set consists of 400 rows with 4096 features.
5. The 3D data set consists of 400*64*64 is converted into 400*4096 and the targets are encoded by using Label binarizer which is provided by sklearn package.
6. The data is split into train and test with 90 and 10 percentage.
7. The train data is used for training the model.
8. By using the tensor flow place holders the weights and the biases are defined . Then cross entropy is used as the cost function for the model and the accuracy is also defined for the model. All these are done by the tensor flow variables.
9. The tensor board variables is defined with the tensor.summary.scalar. And the graphs as well.
10. When the session is initiated and till it finishes the following process occurs the training data is divided into batches of size 100 and fed to the tensorflow model and the logistic regression is implemented on the batch.
11. The cost is calculated for each epoch and the accuracy for the training set is calculated at the end. The cost is plotted and at the end the test accuracy is also calculated.
12. Once the program is run the tensor board is initiated and the graphs are observed.

```
Activities Terminal MI 21:09
nazeer@sunny: ~/bigdata/Lab/Lab05

File Edit View Search Terminal Help
nazeer@sunny:~/bigdata/Lab/Lab05$ python ex2.py
Epoch: 0 cost = 3.67762843768
Epoch: 1 cost = 3.65088486671
Epoch: 2 cost = 3.62807623545
Epoch: 3 cost = 3.60578632355
Epoch: 4 cost = 3.5836815834
Epoch: 5 cost = 3.56172434489
Epoch: 6 cost = 3.53990896543
Epoch: 7 cost = 3.51823369662
Epoch: 8 cost = 3.49669909477
Epoch: 9 cost = 3.47530285517
Epoch: 10 cost = 3.45404402415
Epoch: 11 cost = 3.43292268117
Epoch: 12 cost = 3.41193731626
Epoch: 13 cost = 3.39108705521
Epoch: 14 cost = 3.37037158012
Epoch: 15 cost = 3.34978938103
Epoch: 16 cost = 3.32934006055
Epoch: 17 cost = 3.30902345975
Epoch: 18 cost = 3.28883798917
Epoch: 19 cost = 3.26878269513
Epoch: 20 cost = 3.24885781606
Epoch: 21 cost = 3.22906208038
Epoch: 22 cost = 3.20939501127
Epoch: 23 cost = 3.18985525767
Epoch: 24 cost = 3.17044313749
Epoch: 25 cost = 3.15115698179
Epoch: 26 cost = 3.13199718793
Epoch: 27 cost = 3.11296192805
Epoch: 28 cost = 3.09405167898
Epoch: 29 cost = 3.07526540756
Epoch: 30 cost = 3.05660200119
Epoch: 31 cost = 3.03806177775
Epoch: 32 cost = 3.01964306831
Epoch: 33 cost = 3.00134627024
Epoch: 34 cost = 2.98317058881
Epoch: 35 cost = 2.96511419614
Epoch: 36 cost = 2.94717852275
```

Learning Rate=0.01
Training Epochs=200
Batch size=100

The training accuracy obtained is 96.9 %
The test accuracy obtained is 89.9 %

```
Activities Terminal MI 21:09
nazeer@sunny: ~/bigdata/Lab/Lab05

File Edit View Search Terminal Help
Epoch: 168 cost = 1.4165131251
Epoch: 169 cost = 1.40965958436
Epoch: 170 cost = 1.40285523733
Epoch: 171 cost = 1.39609996478
Epoch: 172 cost = 1.38939317067
Epoch: 173 cost = 1.38273449739
Epoch: 174 cost = 1.37612362703
Epoch: 175 cost = 1.36956008275
Epoch: 176 cost = 1.3630434672
Epoch: 177 cost = 1.35657350222
Epoch: 178 cost = 1.35014947255
Epoch: 179 cost = 1.34377149741
Epoch: 180 cost = 1.33743882179
Epoch: 181 cost = 1.33115108808
Epoch: 182 cost = 1.32490809759
Epoch: 183 cost = 1.31870941321
Epoch: 184 cost = 1.31255447865
Epoch: 185 cost = 1.30644317468
Epoch: 186 cost = 1.30037498474
Epoch: 187 cost = 1.29434971015
Epoch: 188 cost = 1.28836671511
Epoch: 189 cost = 1.28242592017
Epoch: 190 cost = 1.27652680874
Epoch: 191 cost = 1.2706690232
Epoch: 192 cost = 1.26485224565
Epoch: 193 cost = 1.25907615821
Epoch: 194 cost = 1.25334060192
Epoch: 195 cost = 1.24764486154
Epoch: 196 cost = 1.24198885759
Epoch: 197 cost = 1.23637211323
Epoch: 198 cost = 1.23079439004
Epoch: 199 cost = 1.2252553304
Training accuracy 96.9444453716
Test accuracy 89.999976158
Run the command line:
--> tensorboard --logdir= /tmp/tensorflow_logs/example1.2/
Then open http://sunny:6006 into your web browser
nazeer@sunny:~/bigdata/Lab/Lab05$
```

After deploying the tensor board and deploying it we can see it by
 Run the command line:
 --> tensorboard --logdir= /tmp/tensorflow_logs/example1.4/
 Then open <http://sunny:6006> into your web browser

