**Assignment**

In this assignment students have to make ARIMA model over shampoo

sales data and check the MSE between predicted and actual value.

Student can download data in .csv format from the following link:

<https://raw.githubusercontent.com/blue-yonder/pydse/master/pydse/data/sales-of-shampoo-over-a-three-ye.csv>

Hint:

Following is the command import

packages and data from pandas import

read\_csv from pandas import datetime

from matplotlib import pyplot

from statsmodels.tsa.arima\_model

import ARIMA from sklearn.metrics

import mean\_squared\_error

def parser(x):

return datetime.strptime('190'+x, '%Y-%m')

series = read\_csv('<https://raw.githubusercontent.com/blue-yonder/pydse/master/pydse/data/sales-of-shampoo-over-a-three-ye.csv>', header=0, parse\_dates=[0],

index\_col=0, squeeze=True, date\_parser=parser)

**Task:**Deploy this assignment in any cloud platform.(Try to look for free cloud platform)

**Assignment:** Submit assignment’s deployable link only.

**import pandas as pd**

**from pandas import datetime**

**from matplotlib import pyplot**

**from statsmodels.tsa.arima\_model import ARIMA**

**from sklearn.metrics import mean\_squared\_error**

**def parser(x):**

**return datetime.strptime('190'+x, '%Y-%m')**

**series = pd.read\_csv('sales-of-shampoo-over-a-three-ye.csv', header=0, parse\_dates=True, index\_col='Month')**

**series.head()**

**series.plot()**

**pyplot.show()**

**X = series.values**

**X**

**size = int(len(X) \* 0.60)**

**print(len(X))**

**print(size)**

**train, test = X[0:size], X[size:len(X)]**

**history = [x for x in train]**

**predictions = list()**

**for t in range(len(test)):**

**model = ARIMA(history, order=(5,1,0))**

**model\_fit = model.fit(disp=0)**

**output = model\_fit.forecast()**

**yhat = output[0]**

**predictions.append(yhat)**

**obs = test[t]**

**history.append(obs)**

**print('predicted=%f, expected=%f' % (yhat, obs))**

**error = mean\_squared\_error(test, predictions)**

**print('Test MSE: %.3f' % error)**

**pyplot.plot(test)**

**pyplot.plot(predictions, color='red')**

**pyplot.show()**