**ACCIDENT FORECASTING**

**USING ARIMA MODEL**

**(PYTHON-PANDAS PACKAGE)**

**SOURCE CODE:**

from pandas import read\_csv

from pandas import datetime

from matplotlib import pyplot

def parser(x):

return datetime.strptime('201'+x, '%Y' )

series = read\_csv('statte-dataa.csv', header=0, parse\_dates=[0], index\_col=0, squeeze=True, date\_parser=parser)

print(series.head())

series.plot()

pyplot.show()

from pandas import read\_csv

from pandas import datetime

from matplotlib import pyplot

from pandas.plotting import autocorrelation\_plot

def parser(x):

return datetime.strptime('201'+x, '%Y')

series = read\_csv('statte-dataa.csv', header=0, parse\_dates=[0], index\_col=0, squeeze=True, date\_parser=parser)

autocorrelation\_plot(series)

pyplot.show()

from pandas import read\_csv

from pandas import datetime

from pandas import DataFrame

from statsmodels.tsa.arima\_model import ARIMA

from matplotlib import pyplot

def parser(x):

return datetime.strptime('201'+x, '%Y')

series = read\_csv('statte-dataa.csv', header=0, parse\_dates=[0], index\_col=0, squeeze=True, date\_parser=parser)

# fit model

model = ARIMA(series, order=(5,1,0))

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

print(model\_fit.summary())

# plot residual errors

residuals = DataFrame(model\_fit.resid)

residuals.plot()

pyplot.show()

residuals.plot(kind='kde')

pyplot.show()

print(residuals.describe())

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('201'+x, '%Y')

series = read\_csv('statte-dataa.csv', header=0, parse\_dates=[0], index\_col=0, squeeze=True, date\_parser=parser)

X = series.values

size = int(len(X) \* 0.66)

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)

# plot

pyplot.plot(test)

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

pyplot.show()

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('201'+x, '%Y')

series = read\_csv('statte-dataa.csv', header=0, parse\_dates=[0], index\_col=0, squeeze=True, date\_parser=parser)

X = series.values

size = int(len(X) \* 0.66)

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)

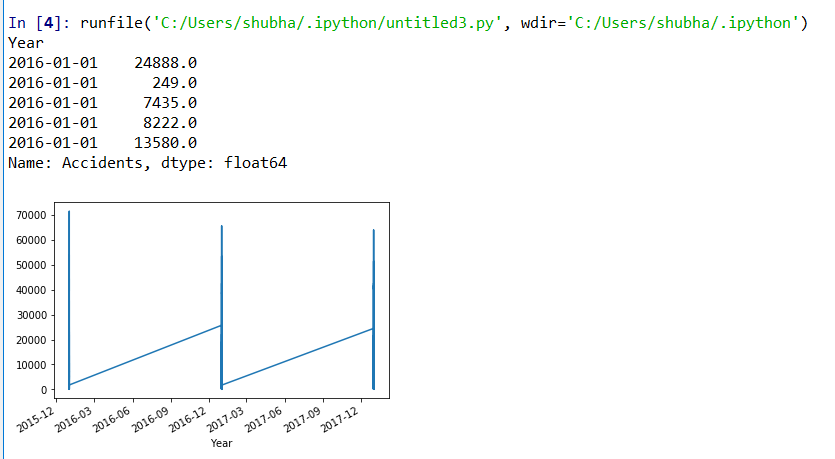
# plot

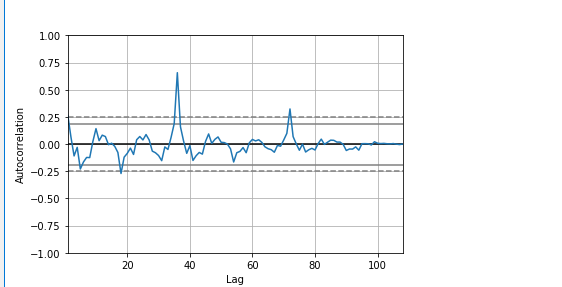
pyplot.plot(test)

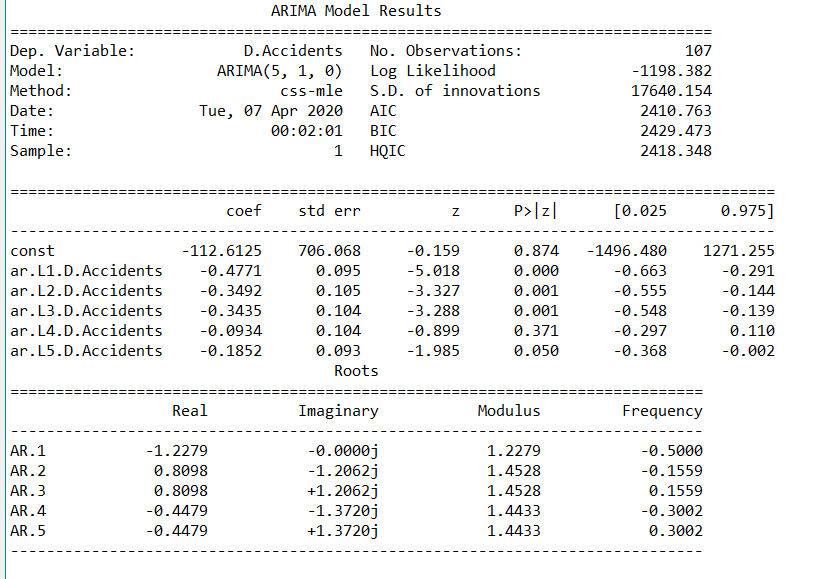
pyplot.plot(predictions, color='red')

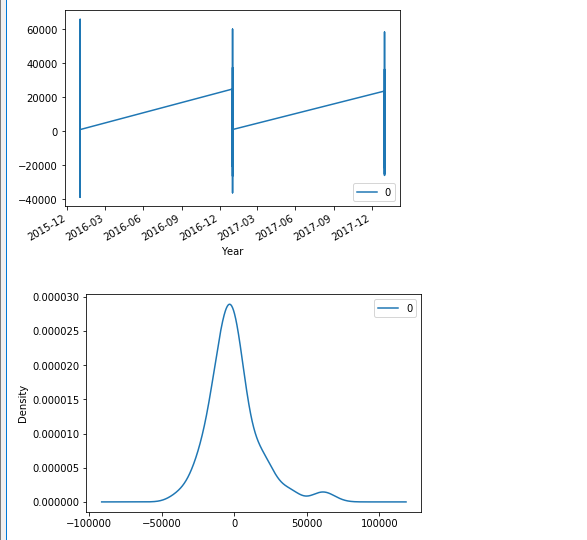
pyplot.show()

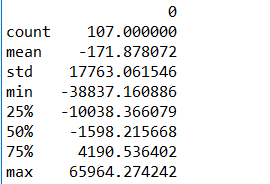
**OUTPUT:**

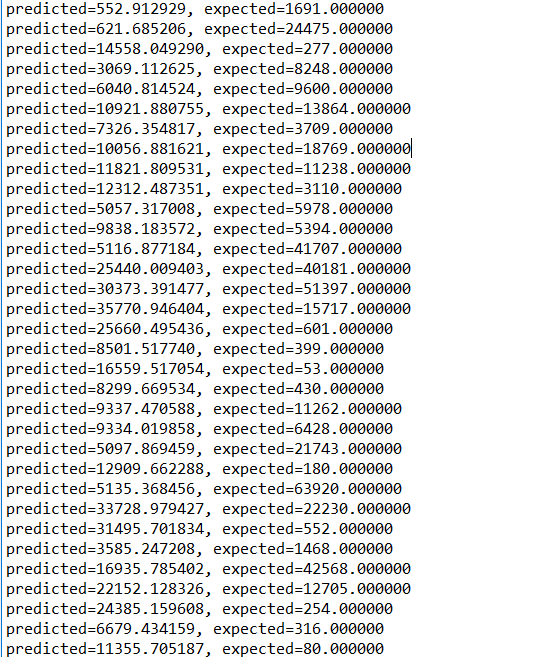
****

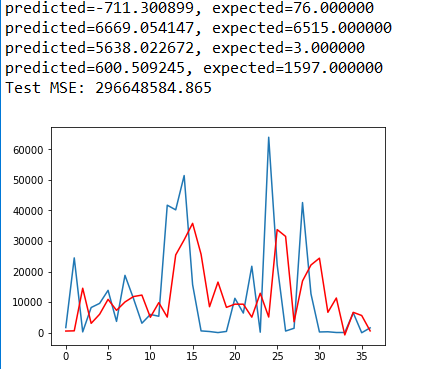
****

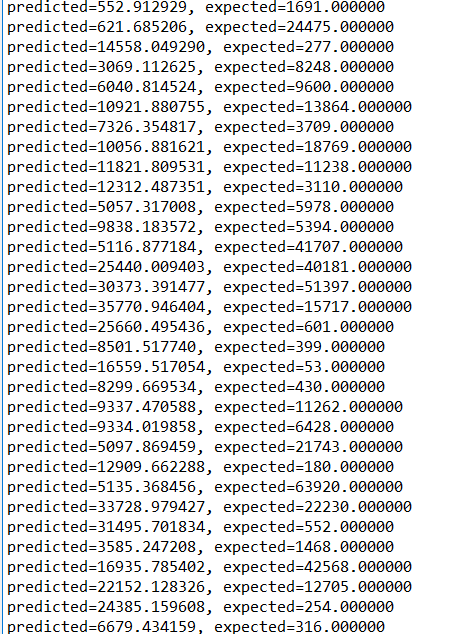
****

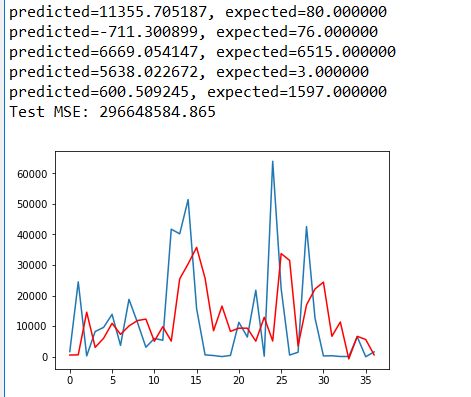
****

****

****

****

****

****