***Importing modules***

# Import pandas as pd

**from** sklearn.model\_selection **import** train\_test\_split

**from** sklearn.linear\_model **import** LinearRegression

**from** sklearn.metrics **import** confusion\_matrix,accuracy\_score

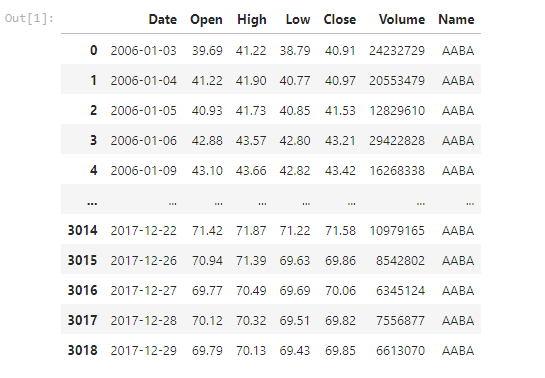
**from** sklearn.ensemble **import** RandomForestRegressor

***Reading the Dataset***

**import** pandas **as** pd

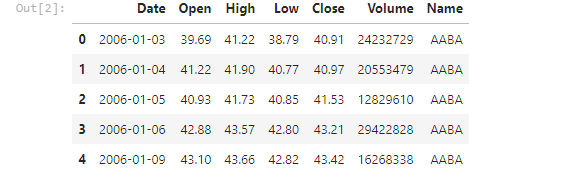
df**=**pd**.**read\_csv("C:\\Users\\GPT BANTWAL\\AI&ML\\AABA\_2006-01-01\_to\_2018-01-01.csv")

df



***check the five columns***

df**.**head()

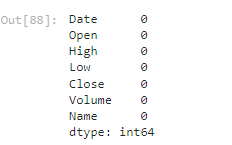


df**.**shape

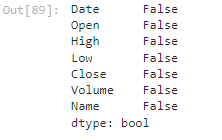


*#Check the missing value*

df**.**isnull()**.**sum()

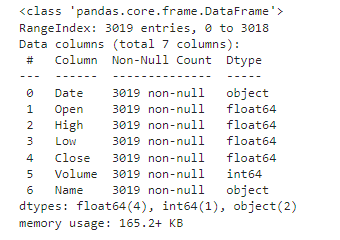


df**.**isnull()**.**any()



*#Checking the information Dataset*

df**.**info()



df**.**describe()



print('lenghth of dataset:',len(df))



*#Linegraph for stock Time Series*

**import** matplotlib.pyplot **as** plt

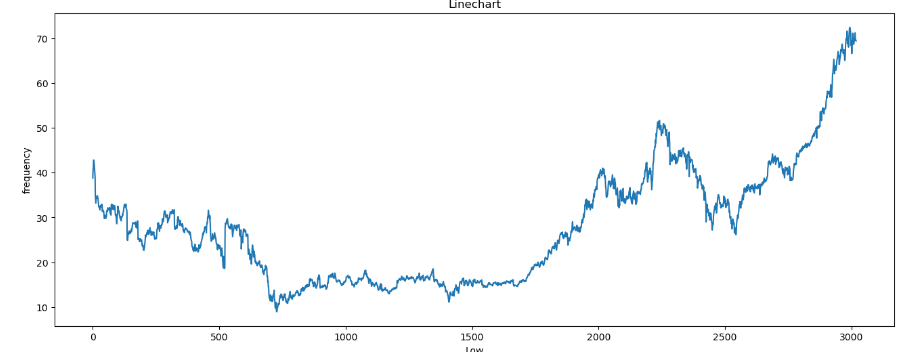
df['Low']**.**plot(figsize**=**(16,6))

plt**.**title('Linechart')

plt**.**xlabel('Low')

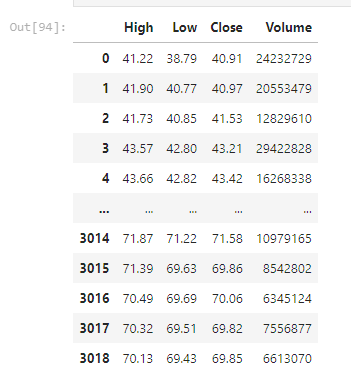
plt**.**ylabel('frequency')

plt**.**show()



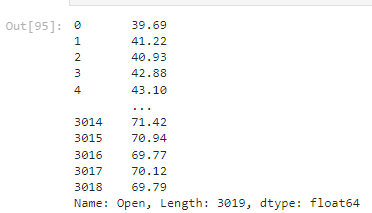
x**=**df**.**drop(['Open','Date','Name'],axis**=**1)

x



y**=**df['Open']

y



**from** sklearn.model\_selection **import** train\_test\_split

x\_train,x\_test,y\_train,y\_test**=**train\_test\_split(x,y,random\_state**=**0)

print(x\_train**.**shape)

print(x\_test**.**shape)

print(y\_train**.**shape)

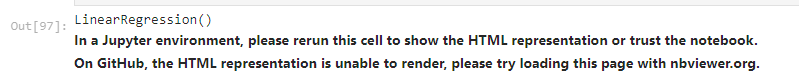
print(y\_test**.**shape)



**from** sklearn.linear\_model **import** LinearRegression

regressor**=**LinearRegression()

regressor**.**fit(x\_train,y\_train)



print(regressor**.**coef\_)



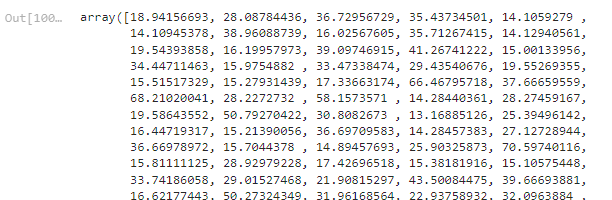
print(regressor**.**intercept\_)



*#Prediction*

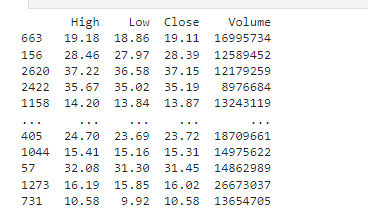
predicted**=**regressor**.**predict(x\_test)

predicted



print(x\_test)

print(x\_test)

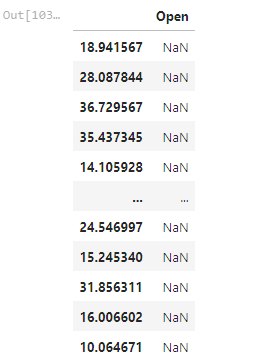


predicted**.**shape



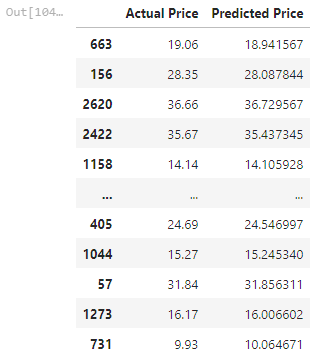
dframe**=**pd**.**DataFrame(y\_test,predicted)

dframe

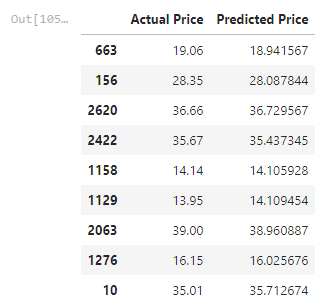


dfr**=**pd**.**DataFrame({'Actual Price':y\_test,'Predicted Price':predicted})

dfr



dfr**.**head(60)



**from** sklearn.metrics **import** confusion\_matrix,accuracy\_score

train\_accuracy**=**regressor**.**score(x\_train,y\_train)

print('train\_accuracy:',train\_accuracy)

test\_accuracy**=**regressor**.**score(x\_test,y\_test)

print('test\_accuracy:',test\_accuracy)



**import** math

**from** sklearn **import** metrics

print('Mean Absolute Error:',metrics**.**mean\_absolute\_error(y\_test,predicted))

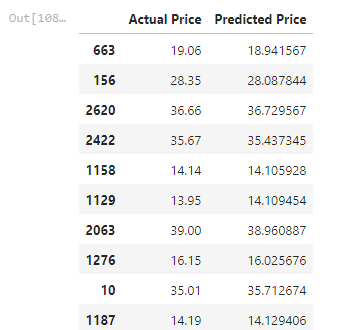
print('Mean Squared Error:',metrics**.**mean\_squared\_error(y\_test,predicted))

print('Root Mean Squared Error:',math**.**sqrt(metrics**.**mean\_squared\_error(y\_test,predicted)))



graph**=**dfr**.**head(20)

graph



*#plotting the bargraph to check difference between actualprice and predicted price*

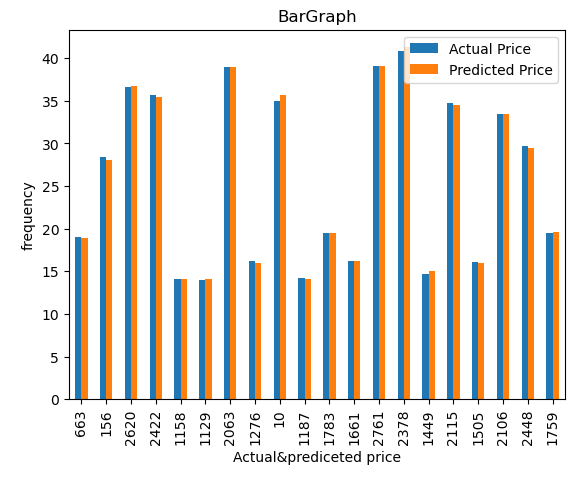
graph**.**plot(kind**=**'bar')

plt**.**title('BarGraph')

plt**.**xlabel('Actual&prediceted price')

plt**.**ylabel('frequency')

plt**.**show()



**from** sklearn.model\_selection **import** train\_test\_split

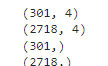
x\_train,x\_test,y\_train,y\_test**=**train\_test\_split(x,y,test\_size**=**0.90,random\_state**=**100)

print(x\_train**.**shape)

print(x\_test**.**shape)

print(y\_train**.**shape)

print(y\_test**.**shape)

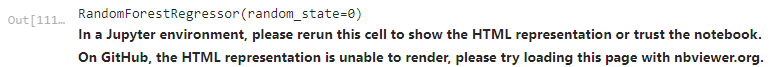


*#Model trainning*

**from** sklearn.ensemble **import** RandomForestRegressor

regressor **=** RandomForestRegressor(n\_estimators**=**100,random\_state**=**0)

regressor**.**fit(x\_train,y\_train)



predicted**=**regressor**.**predict(x\_test)

predicted



*#Evaluating the model*

**from** sklearn.metrics **import** confusion\_matrix,accuracy\_score

train\_accuracy**=**regressor**.**score(x\_train,y\_train)

print('train\_accuracy:',train\_accuracy)

R\_test\_accuracy**=**regressor**.**score(x\_test,y\_test)

print('test\_accuracy:',R\_test\_accuracy)



*#Linear and Randomforestregression using bar plot*

**import** matplotlib.pyplot **as** plt

linear\_regression\_accuracy **=** 0.999700039585113

random\_forest\_accuracy **=** 0.998957807027415

accuracy\_scores **=** [linear\_regression\_accuracy, random\_forest\_accuracy]

model\_names **=** ['Linear Regression', 'Random Forest Regression']

plt**.**bar(model\_names, accuracy\_scores)

plt**.**xlabel('Regression Models')

plt**.**ylabel('Test Accuracy')

plt**.**title('Comparison of Test Accuracy: Linear Regression vs Random Forest Regression')

plt**.**show()

