## IMPORTING LIBARIRES

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

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

from sklearn.linear\_model import LinearRegression

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

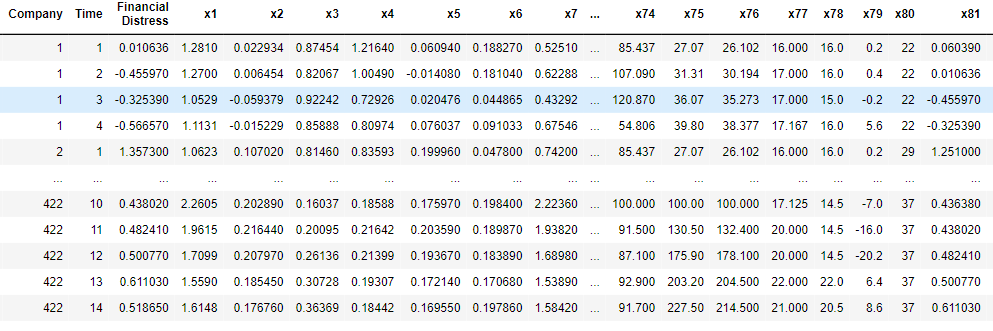
from sklearn.ensemble import RandomForestRegressor

## REDAING DATASET

df=pd.read\_csv('Financial Distress.csv.zip')

df

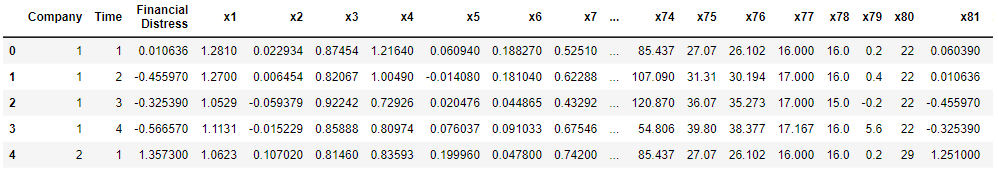
## OUTPUT



## TO CHECK FIRST FIVE COLUMNS

df.head()

## output



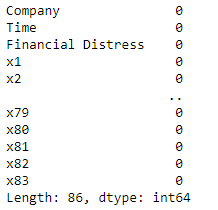
df.shape

## output

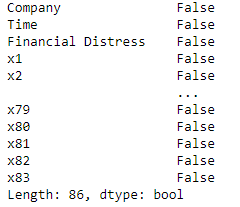


## CHECKING MISSING VALUE

df.isnull().sum()



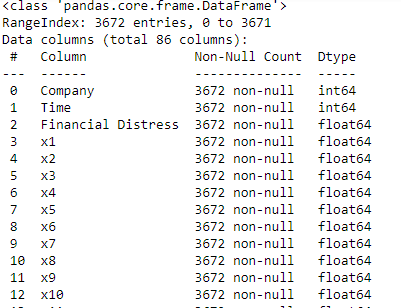
df.isnull().any()



## INFORMATION ABOUT DATASET

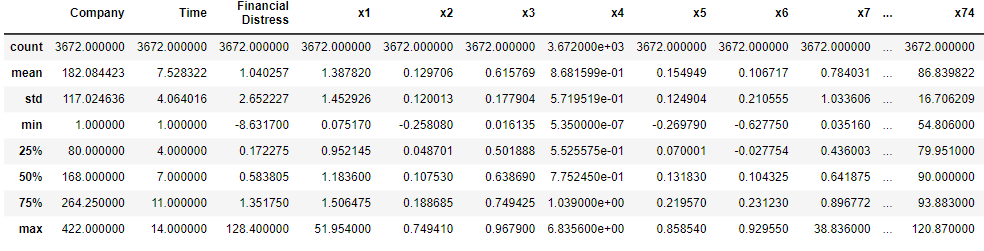
df.info()

## output:



df.describe()

## output:



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

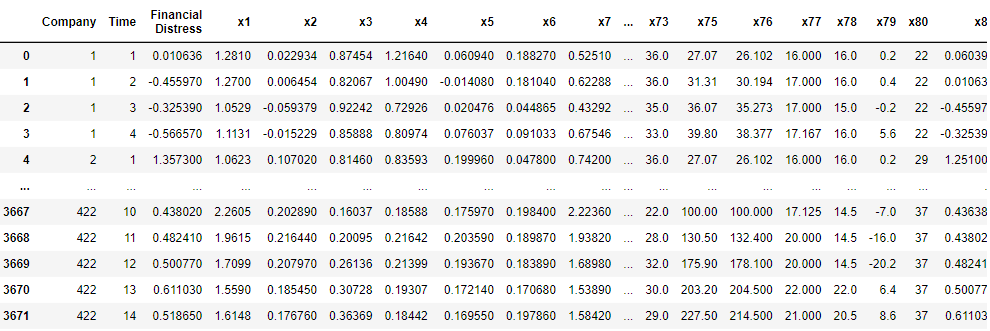


## SPLITTING THEDATASET

x=df.drop('x74',axis=1)

x

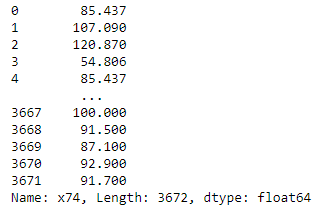
## output:



y=df['x74']

y

## output:



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

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

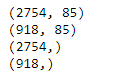
print(x\_train.shape)

print(x\_test.shape)

print(y\_train.shape)

print(y\_test.shape)

## output:

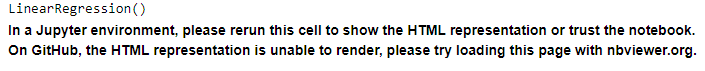


from sklearn.linear\_model import LinearRegression

regressor=LinearRegression()

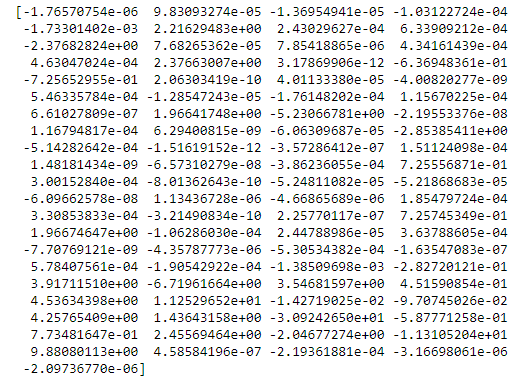
regressor.fit(x\_train,y\_train)

## output:



print(regressor.coef\_)

## output:



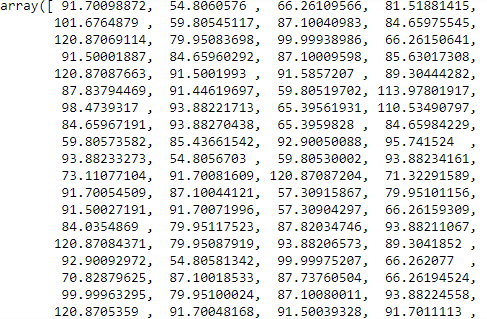
print(regressor.intercept\_)

## output:

-46.268254777815955

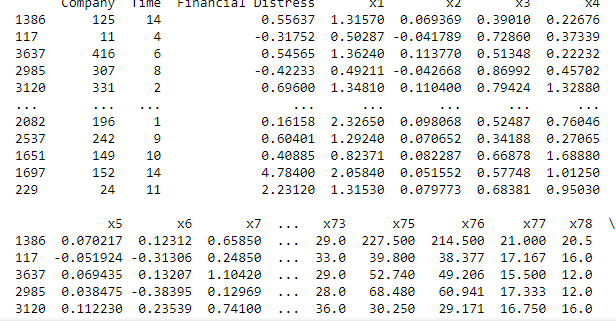
predicted=regressor.predict(x\_test)

predicted



print(x\_test)

## output:



predicted.shape

## output:

(918,)

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

dframe

## output:

| **x74** |
| --- |
| **91.700989** | NaN |
| **54.806058** | NaN |
| **66.261096** | NaN |
| **81.518814** | NaN |
| **101.676488** | NaN |
| **...** | ... |
| **78.786461** | NaN |
| **93.882140** | NaN |
| **100.000214** | NaN |
| **91.700595** | NaN |
| **91.500303** | NaN |
|  |  |
|  |  |

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

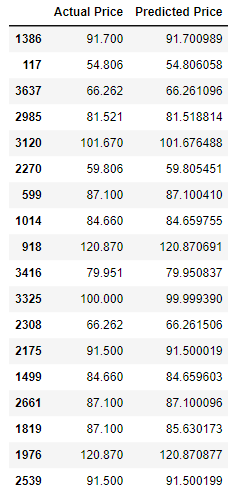
dfr

## output:

|  | **Actual Price** | **Predicted Price** |
| --- | --- | --- |
| **1386** | 91.700 | 91.700989 |
| **117** | 54.806 | 54.806058 |
| **3637** | 66.262 | 66.261096 |
| **2985** | 81.521 | 81.518814 |
| **3120** | 101.670 | 101.676488 |
| **...** | ... | ... |
| **2082** | 78.786 | 78.786461 |
| **2537** | 93.883 | 93.882140 |
| **1651** | 100.000 | 100.000214 |
| **1697** | 91.700 | 91.700595 |
| **229** | 91.500 | 91.500303 |

dfr.head(60)

## output:



from sklearn.metrics import confusion\_matrix,accuracy\_scorein\_accuracy

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)

## output:



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)))

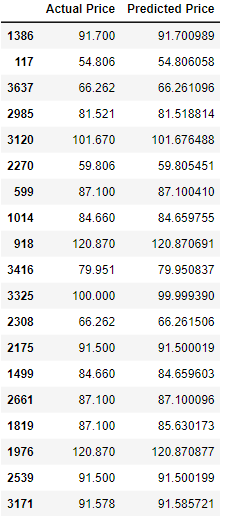
## output:



graph=dfr.head(20)

graph

## output:



graph.plot(kind='bar')

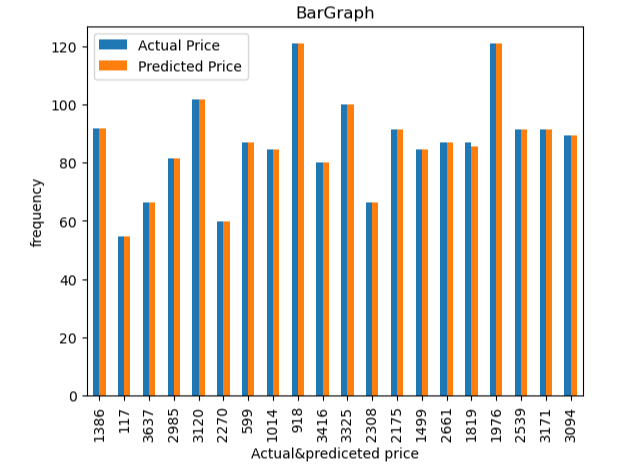
plt.title('BarGraph')

plt.xlabel('Actual&prediceted price')

plt.ylabel('frequency')

plt.show()

## output:



from sklearn.metrics import confusion\_matrix,accuracy\_scorein\_accuracy

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)

## output:

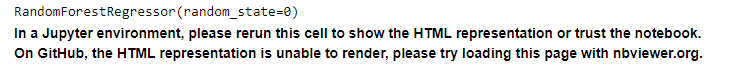


from sklearn.ensemble import RandomForestRegressor

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

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

## output:



predicted=regressor.predict(x\_test)

predicted

## output:



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)

## output:



import matplotlib.pyplot as plt

linear\_regression\_accuracy =0.9999999825628604

random\_forest\_accuracy =0.992675587067571

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()

## output:

