LAB ASSIGNMENT 4

19103112-Shubhjot Singh

Question :

We will use a simple linear regression model to describe the relationship between the Columns X and Y. A residual is the difference between what is plotted in your scatter plot at a specific point, and what the regression equation predicts "should be plotted" at this specific point. If the scatter plot and the regression equation "agree" on a y-value (no difference), the residual will be zero. Plot a residual graph for the given data using linear regression.

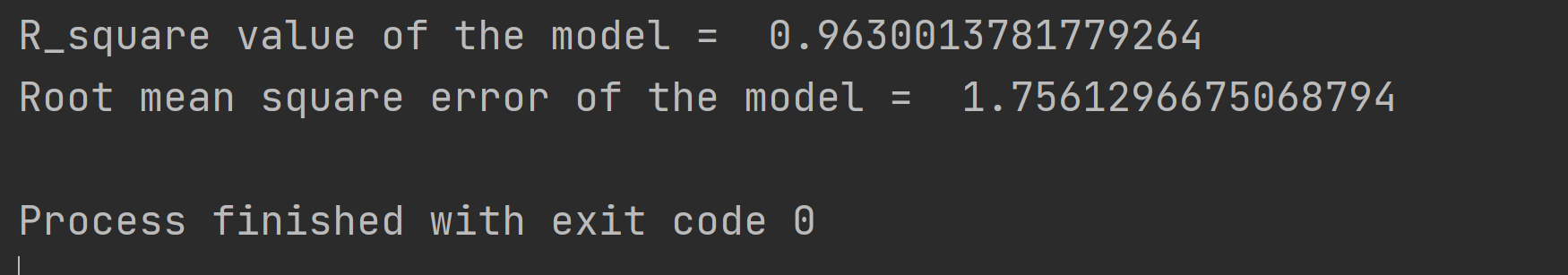
Code :

import pandas as pd  
from matplotlib import pyplot as plt  
import math  
import numpy as np  
  
train\_set = pd.read\_csv('train (1).csv')  
m=0  
c=0  
#y=mx+c  
#m=sum((x-mean(x))\*(y-mean(y)))/sum(x-mean(x)))  
#c=mean(y)-m\*(mean(x))  
s\_x = 0  
s\_y = 0  
  
def average(train\_set):  
 s\_x=0  
 s\_y=0  
 num=len(train\_set)  
 for i in range(num):  
 s\_x = s\_x+train\_set['x'][i]  
 s\_y = s\_y + train\_set['y'][i]  
 return s\_x/num,s\_y/num  
  
  
def slope\_constant\_calculate(s\_x,s\_y,train\_set):  
 temp1=0  
 temp2=0  
 n=len(train\_set)  
 for i in range(n):  
 temp1+=(train\_set['x'][i]-s\_x)\*(train\_set['y'][i]-s\_y)  
 temp2+=(train\_set['x'][i]-s\_x)\*(train\_set['x'][i]-s\_x)  
 m = temp1/temp2  
 c = s\_y-m\*s\_x  
 return m,c  
  
  
def load\_data(train\_set):  
 n=len(train\_set)  
 list\_x=[]  
 for i in range(n):  
 list\_x.append(train\_set['x'][i])  
 return list\_x  
  
  
  
#Performance calculator  
  
def performance(c,m,test\_set):  
 n=len(test\_set)  
 x=0  
 y=0  
 predicted\_list=[]  
 predicted=0  
 temp1=0  
 temp2=0  
 residual\_list=[]  
 y\_mean=0  
 y\_root=0  
 acc=0  
  
 for i in range(n):  
 x=test\_set['x'][i]  
 y=test\_set['y'][i]  
 predicted=(m\*x+c)  
 predicted\_list.append(m\*x+c)  
 y\_mean=y\_mean+y  
 residual\_list.append(y-predicted)  
 y\_root+=(y-predicted)  
  
 y\_mean=y\_mean/n  
 for i in range(n):  
 y=test\_set['y'][i]  
 temp1+=((predicted\_list[i]-y\_mean)\*(predicted\_list[i]-y\_mean))  
 temp2+=((y-y\_mean)\*(y-y\_mean))  
 acc = acc + (y - y\_root) \* (y - y\_root)  
  
 R\_square=temp1/temp2  
 return R\_square,residual\_list,math.sqrt(acc)  
  
  
  
  
  
  
  
  
  
  
list\_x=load\_data(train\_set)  
s\_x,s\_y=average(train\_set)  
m,c = slope\_constant\_calculate(s\_x,s\_y,train\_set)  
  
  
  
#predictions  
  
test\_set=pd.read\_csv('test (1).csv')  
n=len(test\_set)  
x=[]  
y=[]  
list\_x=np.array(list\_x)  
  
for i in range(n):  
 x.append(test\_set['x'][i])  
 y.append(test\_set['y'][i])  
  
plt.scatter(x,y,color='blue')  
plt.plot(list\_x,m\*list\_x+c,color='red')  
plt.xlabel("Independent variable")  
plt.ylabel("Dependent variable")  
plt.show()  
  
  
R\_square\_measure,residual,accuracy=performance(c,m,test\_set)  
plt.scatter(x,residual,color='green')  
plt.xlabel("Independent Variable")  
plt.ylabel("residuals ( y\_predicted-y\_dataset )")  
plt.title("Residual Graph")  
plt.show()  
  
print("R\_square value of the model = ",R\_square\_measure)  
print("Root mean square error of the model = ",accuracy/(2\*n))

OUTPUT :

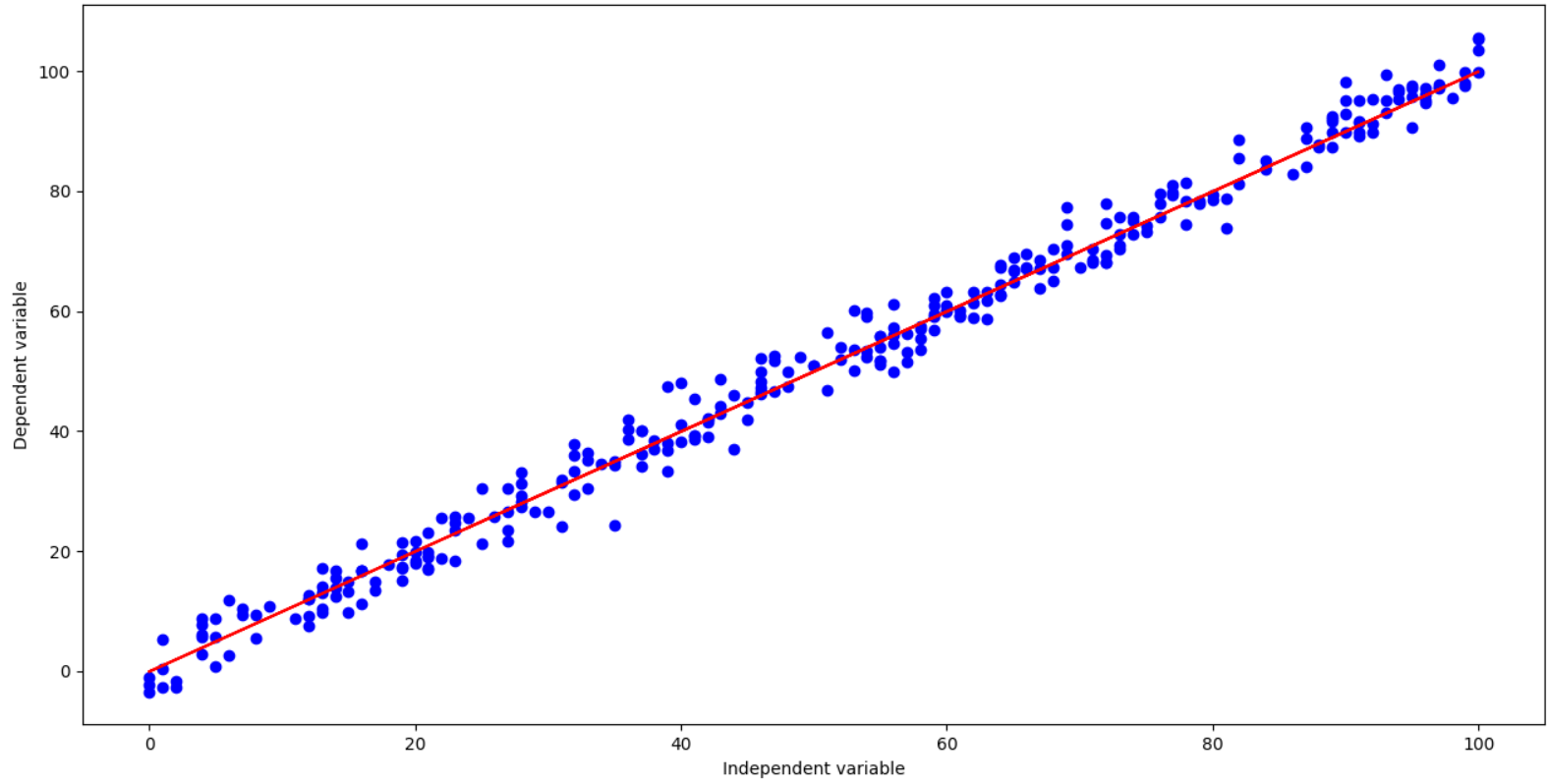
Root mean square error and R Square value

Used to determine the accuracy of the model



Graphical Outputs:

1. Actual value and predicted regression graph



2.

Residual Graph:

