Q-Build a simple linear regression model by performing EDA and do necessary transformations and select the best model using R or Python.

1) Delivery time -> Predict delivery time using sorting time

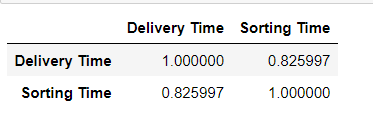
Python Code-

import pandas as pd

data=pd.read\_csv("C:\\Users\\advay\\Downloads\\Assignment 4\\delivery\_time.csv")

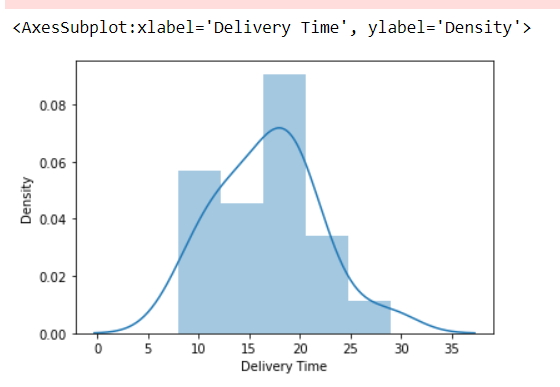
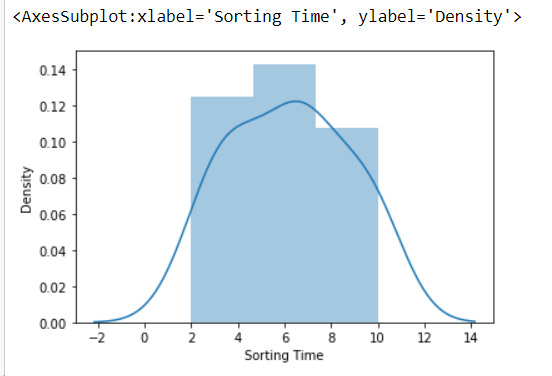
data.info()

data.corr() -----{To check Correlation between the columns}



import seaborn as sns

sns.distplot(data['Delivery Time']) sns.distplot(data['Sorting Time'])

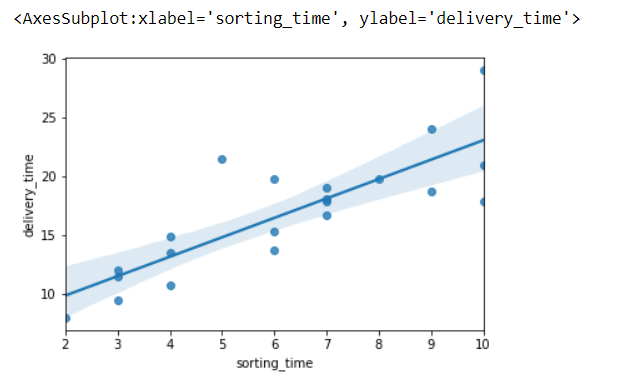
Distribution Plots of the Columns

data.rename(columns={'Delivery Time':'delivery\_time','Sorting Time':'sorting\_time'},inplace=True)

import statsmodels.formula.api as smf

model=smf.ols("delivery\_time~sorting\_time",data=data).fit()

sns.regplot(x="sorting\_time",y="delivery\_time",data=data)

**

*model.params*

*print(model.tvalues,'/n',model.pvalues)*

*(model.rsquared,model.rsquared\_adj)*

*(0.6822714748417231, 0.6655489208860244)*

*newdata=pd.Series([7,10,12,5]) {Predicting Delivery Time}*

*data\_pred=pd.DataFrame(newdata,columns=['sorting\_time'])*

*model.predict(data\_pred)*

***Output:***

*0 18.125873*

*1 23.072933*

*2 26.370973*

*3 14.827833*

2) Salary\_hike -> Build a prediction model for Salary\_hike

Python Code-

import pandas as pd

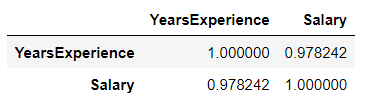
import seaborn as sns

import statsmodels.formula.api as smf

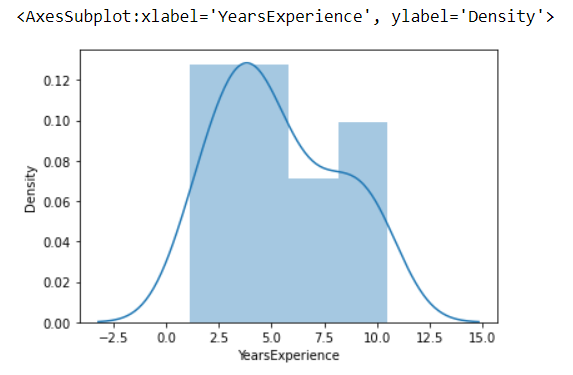
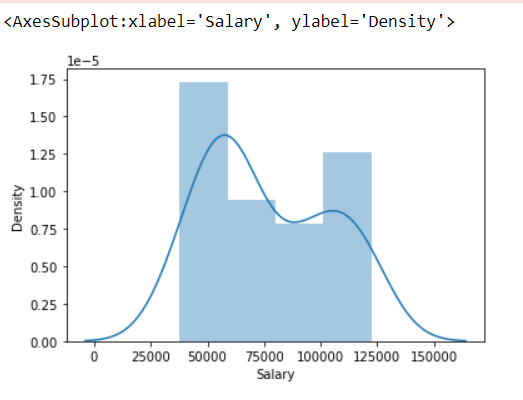
data=pd.read\_csv("C:\\Users\\advay\\Downloads\\Assignment 4\\Salary\_Data.csv")

data.info()

data.corr()

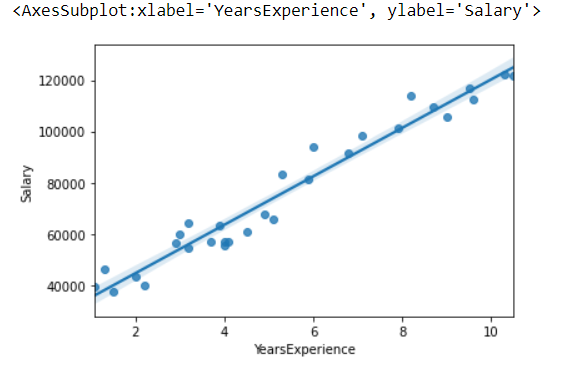


sns.distplot(data['YearsExperience']) sns.distplot(data['Salary'])

model=smf.ols("Salary~YearsExperience",data=data).fit()

sns.regplot(x="YearsExperience",y="Salary",data=data)



model.params

print(model.tvalues,'/n',model.pvalues)

(model.rsquared, model.rsquared\_adj)

(0.9569566641435086, 0.9554194021486339)

#predicting Salary

newdata=pd.Series([15,10,18,19,0])

data\_pred=pd.DataFrame(newdata,columns=['YearsExperience'])

model.predict(data\_pred)

**Output-**

0 167541.635020

1 120291.823413

2 195891.521985

3 205341.484306

4 25792.200199