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

Ans: import pandas as pd

import numpy as np

import seaborn as sns

import statsmodels.formula.api as smf

data=pd.read\_csv('E:\\delivery\_time.csv')

data.head()

data.info()

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

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

data=data.rename({'Delivery Time':'delivery\_time', 'Sorting Time':'sorting\_time'},axis=1)

data.head()

data.corr()

sns.regplot(x=data['sorting\_time'],y=data['delivery\_time'])

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

model.params

model.tvalues , model.pvalues

model.rsquared , model.rsquared\_adj

delivery\_time = (6.582734) + (1.649020)\*(5)

delivery\_time

new\_data=pd.Series([5,8])

new\_data

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

data\_pred

model.predict(data\_pred)

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

Ans: import pandas as pd

import numpy as np

import seaborn as sns

import statsmodels.formula.api as smf

data=pd.read\_csv('E:\\salary\_data.csv')

data.head()

data.info()

sns.distplot(data['YearsExperience'])

sns.distplot(data['Salary'])

data.corr()

sns.regplot(x=data['YearsExperience'],y=data['Salary'])

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

model.params

model.tvalues , model.pvalues

model.rsquared , model.rsquared\_adj

delivery\_time = (25792.200199) + (9449.962321)\*(3)

delivery\_time

new\_data=pd.Series([3,5])

new\_data

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

data\_pred

model.predict(data\_pred)

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