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

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

------------------------------------------------------------

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

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

#Import Libraries

import pandas as pd

import seaborn as sns

import statsmodels.formula.api as smf

import matplotlib.pyplot as plt

#import dataset

df=pd.read\_csv("delivery\_time.csv")

df

df.info()

#Renaming Columns

df1=df.rename({'Delivery Time':'del\_tm','Sorting Time':'sort\_tm'},axis=1)

df1

#Correlation Analysis

df1.corr()

#Splitting Variables

X=df1[['sort\_tm']]

Y=df1['del\_tm']

from sklearn.linear\_model import LinearRegression

LR = LinearRegression()

LR.fit(X,Y)

Bo=LR.intercept\_ #Bo

B1=LR.coef\_

# Predictions

Y\_pred=LR.predict(X)

#EDA and Data Visualization

plt.scatter(x=df1['sort\_tm'],y=df1['del\_tm'],color='blue')

plt.scatter(x=df1['sort\_tm'],y=Y\_pred,color='red')

plt.plot(df1['sort\_tm'],Y\_pred,color='black')

plt.xlabel("Sorting Time")

plt.ylabel("Delivery Time")

plt.show()

sns.regplot(x=df1['sort\_tm'],y=df1['del\_tm'])

Y-Y\_pred

model=smf.ols("del\_tm~sort\_tm",data=df1).fit()

model.summary()

#Model Testing

#Finding Coeficient Parameter

model.params

#Finding TValue and PValue

model.tvalues

model.pvalues

#Finding RSquared Values

model.rsquared

model.rsquared\_adj

#Model Predictions manually for sorting time 5

del\_tm = (6.582734) + (1.649020)\*(5) # Coeficient Paramenter Value

del\_tm

#Automatic Prediction for say sorting time 5, 8

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

new\_df

df\_pred=pd.DataFrame(new\_df,columns=['sort\_tm'])

df\_pred

model.predict(df\_pred)

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

#Import Libraries

import pandas as pd

import seaborn as sns

import statsmodels.formula.api as smf

import matplotlib.pyplot as plt

#import dataset

df=pd.read\_csv("Salary\_Data.csv")

df

df.info()

#Correlation Analysis

df.corr()

#Splitting Variables

X=df[['YearsExperience']]

Y=df['Salary']

from sklearn.linear\_model import LinearRegression

LR = LinearRegression()

LR.fit(X,Y)

Bo=LR.intercept\_ #Bo

B1=LR.coef\_

# Predictions

Y\_pred=LR.predict(X)

#EDA and Data Visualization

plt.scatter(x=df['YearsExperience'],y=df['Salary'],color='blue')

plt.scatter(x=df['YearsExperience'],y=Y\_pred,color='red')

plt.plot(df['YearsExperience'],Y\_pred,color='black')

plt.xlabel("YearsExperience")

plt.ylabel("Salary")

plt.show()

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

Y-Y\_pred

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

model.summary()

#Model Testing

#Finding Coeficient Parameter

model.params

#Finding TValue and PValue

model.tvalues

model.pvalues

#Finding RSquared Values

model.rsquared

model.rsquared\_adj

#Model Predictions manually for 3 years Experience

Salary = (25792.200199) + (9449.962321)\*(3) # Coeficient Paramenter Value

Salary

#Automatic Prediction for say Years of Experience 5, 8

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

new\_df

df\_pred=pd.DataFrame(new\_df,columns=['YearsExperience'])

df\_pred

model.predict(df\_pred)