

















#importing necessary libraries

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

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

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_squared\_error

from sklearn.metrics import mean\_absolute\_error,r2\_score

#importing dataset

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

#view first few rows

df.head()

#drop first column from dataframe

df = df.drop(df.columns[0],axis=1)

df.head()

#shape of dataframe

df.shape

#split df into X and y variables

x = df[['YearsExperience']].values

y = df['Salary'].values

#plot scatter plot between salary and yearsExperience

plt.figure(figsize=(8,6))

plt.scatter(df['YearsExperience'],df['Salary'])

plt.xlabel('Experience')

plt.ylabel('Salary')

plt.show()

#train test split

x\_train,x\_test,y\_train,y\_test=train\_test\_split(x,y,test\_size=0.2)

lr=LinearRegression()

#train the mode

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

#make predictions on train & test data

y\_pred1 = lr.predict(x\_train)

y\_pred2 = lr.predict(x\_test)

print(y\_pred1)

print(y\_train)

print(y\_pred2)

print(y\_test)

lr.coef\_

lr.intercept\_

#calculate MSE & R2 score of training data

print(mean\_squared\_error(y\_train,y\_pred1))

print(r2\_score(y\_train,y\_pred1))

#calculate MSE & R2 score of training data

print(mean\_squared\_error(y\_test,y\_pred2))

print(r2\_score(y\_test,y\_pred2))

#plotting a scatter plot of Experience vs Model Predicted Salary in Training Data

plt.figure(figsize=(8,8))

plt.scatter(x\_train,y\_train)

plt.plot(x\_train,y\_pred1,color = "red")

plt.title('Experience vs Model Predicted Salary(Train)')

plt.xlabel('Experience')

plt.ylabel('Salary')

plt.show()

#plotting a scatter plot of Experience vs Model Predicted Salary in Test Data

plt.figure(figsize=(8,8))

plt.scatter(x\_test,y\_test)

plt.plot(x\_test,y\_pred2,color = "red")

plt.title('EXPERIENCE vs Model Predicted Salary(Test)')

plt.xlabel('Experience')

plt.ylabel('Salary')

plt.show()

class SimpleLinearRegression:

def \_\_init\_\_(self):

self.m=None

self.b=None

def fit(self,x\_train,y\_train):

num=0

denum=0

for i in range(x\_train.shape[0]):

num=num+(x\_train[i]-x\_train.mean())\*(y\_train[i]-y\_train.mean())

denum=denum+(x\_train[i]-x\_train.mean())\*(x\_train[i]-x\_train.mean())

self.m=num/denum

self.b=y\_train.mean()-(self.m\*x\_train.mean())

print(self.m)

print(self.b)

def predict(self,x\_test):

return self.m\*x\_test+self.b

SLR=SimpleLinearRegression()

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