import pandas as pd

import matplotlib.pylot as plt

data = pd,read\_scv("Salary\_Data.csv")x=

data['years Expereince'].values

y=data['salary'].values x =

x.reshape(len(x), I)y=y.reshape(len(y).1)

### Split the data into training & testing samplesfrom

sklearn.model\_selection.xtrain,xtest,ytrain,ytest = train\_test\_split(x,y,test\_size = 0.20.random\_state=0) xtrainxtest ytrainytestdata.isnull().sum()

###Train the model model.fit(xtrain,ytrain) ### predictionypred = model.predict(xtest)ypred

### Draw the line of regression (training sample)

plt.scatter(xtrain, ytrain.color

="blue" label="Actual Points") plt.scatter(xtrain.model.predict(xtrain).color="red",

label="Predicted Points") plt.plot(xtrain,model.predict(xtrain),color="purple", label"Line of Regression")

plt.legend(loc=2) plt.show()

Draw the line of regression (test sample)

plt.scatter(xtest,ytest.color="blue".label="Actual Points")

plt.scatter(xtest.ypred.color="red".label="predicted Point")

plt.plot(xtest.ypred.color="purple", label =" Line of Regression") plt.legend(loc=2) plt.show()

[6:52 pm, 23/09/2024] ᴍᴏʜᴀᴍᴍᴇᴅ ꜰᴀɪᴢ ᴋʜᴀɴ: ### Calculate the R-Square from sklerarn. metrics import r2\_scorer2 r2\_score(ytest.ypred)r2

### Calculate the MSE

from sklearn.metrics import mean\_sqaured\_errore

mean\_squared\_error(ytest, ypred)e ### Equation of linem model.coef\_c= model.intercept

### Saving the modeling from sklearn.externals

import joblibjoblib.dump(model, linear.pk1") ###

loading the model

from sklearn.externals import joblib my\_model= joblib.load(linear.pkl")my\_model.predict([25])

import numpy as np

### Function for Salary Predictiondef salary\_predict(): years int(input("Enter the Experience in years:"))salary = my\_model.predict([[years]])salary = np.round(salary)return "Salary of the Employee"+str(salary)salary\_predict()