**Name:** Jaydeep Solanki

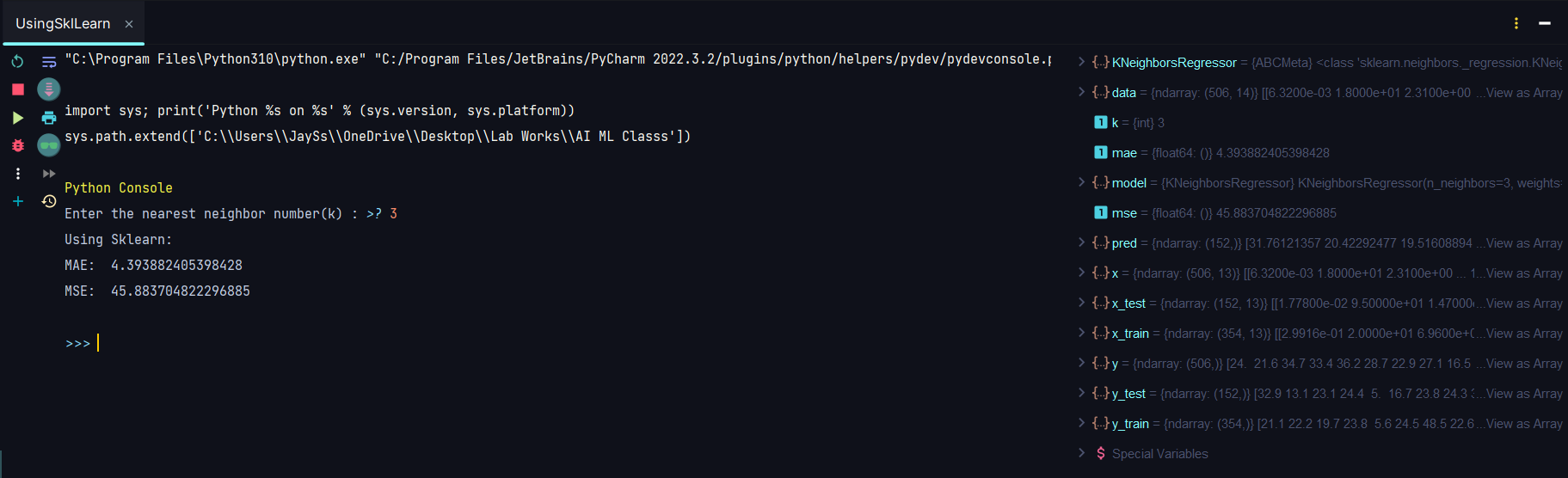
**Roll:** 22BEC059

**Practical 7:** Write a program for predicting selling price of houses in Boston dataset using KNN regressor

1. Using SkLearn Library.

import pandas as pd  
from sklearn.metrics import mean\_absolute\_error, mean\_squared\_error  
from sklearn.model\_selection import train\_test\_split  
from sklearn.neighbors import KNeighborsRegressor  
  
data = pd.read\_csv("BostonHousing.csv", header='infer').values  
  
x = data[:, 0:-1]  
y = data[:, -1]  
x\_train, x\_test, y\_train, y\_test = train\_test\_split(x, y, test\_size=0.3)  
k = int(input("Enter the nearest neighbor number(k) : "))  
  
model = KNeighborsRegressor(n\_neighbors=k, weights='distance')  
model.fit(x\_train, y\_train)  
pred = model.predict(x\_test)  
mae = mean\_absolute\_error(y\_test, pred)  
mse = mean\_squared\_error(y\_test, pred)  
print("Using Sklearn: ")  
print("MAE: ", mae)  
print("MSE: ", mse)

Output:



1. Without using SkLearn Library

import numpy as np  
import pandas as pd  
from sklearn.model\_selection import train\_test\_split  
from tabulate import tabulate

def MAE(pred, y\_test):  
 return np.mean(abs(pred - y\_test))  
  
def MSE(pred, y\_test):  
 return np.mean((pred - y\_test) \*\* 2)  
  
def RMSE(mse):  
 return np.sqrt(mse)  
  
def MAPE(pred, y\_test):  
 return np.mean(abs(pred - y\_test) / y\_test)  
  
def KNNRegressor(k):  
 data = pd.read\_csv("BostonHousing.csv", header='infer').values  
 x = data[:, 0:-1]  
 y = data[:, -1]  
 x\_train, x\_test, y\_train, y\_test = train\_test\_split(x, y, test\_size=0.3)  
 nClasses = np.unique(y\_train).shape[0]  
 distance = np.zeros(shape=x\_train.shape[0])  
 pred = np.zeros(shape=x\_test.shape[0])   
 for i in range(x\_test.shape[0]):  
 distance = np.sqrt(np.sum((x\_train - x\_test[i]) \*\* 2, axis=1))  
 kMinIndex = np.argpartition(distance, k)[0:k]  
 invDist = 1 / (distance + 10e-20)  
 Denom = sum(invDist[kMinIndex])  
 pred[i] = np.dot(invDist[kMinIndex] / Denom, y\_train[kMinIndex])  
 return [MAE(pred, y\_test), MSE(pred, y\_test), RMSE(MSE(pred, y\_test)), MAPE(pred, y\_test)]  
  
display = {}  
n = int(input("Enter the number of 'k' till which you have to compare: "))  
for i in range(1, n+1):  
 ans = KNNRegressor(i)  
 display[i] = ans  
  
print(tabulate(pd.DataFrame(display).T, tablefmt="pretty", headers=["MAE", "MSE", "RMSE", "MAPE"]))

Output:

