# Lab Assignment #1

* **Select any data set (No toy Datasets) with dimensions >= (500,11):**

I have selected **The Boston Housing Dataset** for this assignment because of below mentioned reasons;

* + This dataset has dimension (505, 14)
  + This dataset is well suited for Linear Regression model as its **target variable** is **continuous** in nature, which the median value of owner occupied homes in $1000.

This dataset is about the property data of Boston and in its suburbs, it contains 14 attributes related to demographic, economical and geographical data. Purpose is to predict the median value of owner-occupied homes in $1000 which depends on above mentioned attributes.

* **CODE:**

**#%% Importing all the libraries for; splitting data, running linear regression and plotting graphs**

import random

import pandas as pd

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

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_squared\_error

import matplotlib.pyplot as plt

**#%% Loading Datasets**

boston\_data = pd.read\_csv(r"C:\Users\Ahsan\Desktop\AIcodes\housing.csv")

**#%% Seperating independent and dependent variables respectively (X=features(13), #y=target(median value of home in $1000))**

X = boston\_data.iloc[:, :-1]

y = boston\_data.iloc[:,-1]

print(boston\_data.shape)

**#%% Converting to list and print its dimensions**

tuples = list(zip(X, y))

print(boston\_data.shape)

**#%% Creating seed values, and lists for MSE and Regression score**

seeds = [1,2,3,4,5,6,7,8,9,10]

mse\_list = []

r2\_scores = []

**#%% Shuffling & Spitting data 10 times, running linearRegression and predicting with R2 and MSE**

for seed in seeds:

**# Setting seed values and shuffling**

random.seed(seed)

random.shuffle(tuples)

**# Splitting data in ratio of 0.7 for training and 0.3 for testing**

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=(seed))

print("Seed", seed, ": Train set size =", len(X\_train), ", Test set size =", len(X\_test))

**# Running LinearRegression**

model = LinearRegression()

model.fit(X\_train, y\_train)

y\_pred = model.predict(X\_test)

**# getting R2 score for each sample of data**

score = model.score(X\_test, y\_test)

r2\_scores.append(score)

print("Seed", seed, "Regression score:", score)

**# getting MSE score for each sample of data**

mse = mean\_squared\_error(y\_test, y\_pred)

mse\_list.append(mse)

print("Seed", seed, "MSE:", mse)

**#%% Plotting for MSE**

plt.bar(seeds, mse\_list)

plt.title("Mean Squared Error (MSE) for 10 random seeds")

plt.xlabel("Random seed")

plt.ylabel("MSE")

plt.show()

**#%% Plotting for R^2 scores**

plt.bar(seeds, r2\_scores)

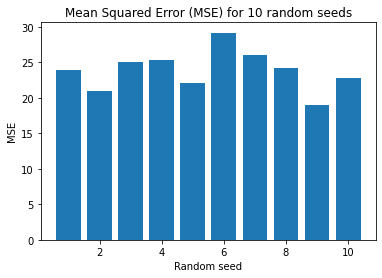
plt.title("Regression Score (R^2) for 10 random seeds")

plt.xlabel("Random seed")

plt.ylabel(("R^2"))

plt.show()

* **Plot for MSE**



* **Plot for R^2 scores**

