**Assignment**

Build the linear regression model using scikit learn in boston data to predict 'Price' based on other dependent variable.

Here is the code to load the data:

import numpy as np

import pandas as pd

import scipy.stats as stats

import matplotlib.pyplot as plt

import sklearn

from sklearn.datasets import load\_boston

boston = load\_boston()

bos = pd.DataFrame(boston.data)

**Task:**Deploy this assignment in any cloud platform.(Try to look for free cloud platform)

**Assignment:** Submit assignment’s deployable link only.

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

Ans:-

**import pickle**

**from flask import Flask, render\_template, request**

**from flask\_cors import cross\_origin**

**app=Flask(\_\_name\_\_)**

**@app.route("/",methods=['GET', 'POST'])**

**@cross\_origin()**

**def home():**

**return render\_template('index.html')**

**@app.route("/predict",methods=['POST','GET'])**

**@cross\_origin()**

**def pred():**

**if request.method=='POST':**

**try:**

**CRIM=float(request.form['crim'])**

**ZN = float(request.form['zn'])**

**INDUS = float(request.form['indus'])**

**CHAS = float(request.form['chas'])**

**NOX = float(request.form['nox'])**

**RM = float(request.form['rm'])**

**AGE = float(request.form['age'])**

**DIS = float(request.form['dis'])**

**RAD = float(request.form['rad'])**

**PTRATIO = float(request.form['pt-ratio'])**

**B = float(request.form['b'])**

**LSTAT = float(request.form['LSTAT'])**

**x\_predict=[[CRIM,ZN,INDUS,CHAS,NOX,RM,AGE,DIS,RAD,PTRATIO,B,LSTAT]]**

**filename='final\_model.pkl'**

**# model=pickle.load(open(config.FINAL\_MODEL,'rb'))**

**model=pickle.load(open(filename,'rb'))**

**prediction=model.predict(x\_predict)**

**return render\_template('result.html', prediction=prediction)**

**except Exception as e:**

**return e**

**else:**

**return render\_template('index.html')**

**if \_\_name\_\_== "\_\_main\_\_":**

**app.run(debug=True)**

**import pickle**

**x\_predict=[[2,2,7,1,0,2,6,5,2,5,4,2]]**

**filename='final\_model.pkl'**

**model=pickle.load(open(filename,'rb'))**

**prediction=model.predict(x\_predict)**

**print(prediction)**