CS 480 Spring 2022

Assignment #5

# Due: Friday, 12:00 AM (midnight) Beijing Time

**May 6, 2022**

Submit before 11:59 PM May 5th (Thursday) Beijing time. Points: **100**

# Problem : Linear Regression

Attached is Python code for linear regression (as a .py and .txt file). You may start with this code for the assignment, or you may add / modify the methods you feel fit. All places labeled as TODO are to be modified.

# Instructions :

1. Make sure your code is clear and well-documented. Points will be deducted for poorly-structured, undocumented, or impossible-to-read code.
2. Feel free to change/add methods as you see fit.
3. Your code MUST run in Python 3.
4. Rename your python file as Anumber\_LinearRegression.py. The python file must run with the command

python Anumber\_LinearRegression.py

1. Feel free to include a README.md or README.txt or README.docx file if any explanations are necessary. Explanations can also be given as comments in the python file.
2. Edit this document to have the answers for questions 1(a, b, c, d, e) and 2(b, f). Rename this file to Anumber\_LinearRegression.docx and submit along with your modified python code.
3. Submit all files in a zipped folder named FirstName\_LastName\_ANumber\_CS480\_Assignment5.

**Task :** Do linear regression on Boston Housing Prices dataset and score the model

Question 1. The code uses Boston housing dataset from the sklearn library. Understand the data. You are free to use any data descriptive methods including plots. [25 pts]

1. How many records are there in the dataset? [5 pts]
2. How many features are present? [5 pts]
3. What are the feature names? [5 pts]
4. What is the mean of all features? [5 pts]
5. What is the standard deviation of all features? [5 pts]

**Solution:**

1. There are *506* records.
2. *13* features are present.
3. They are *CRIM, ZN, INDUS, CHAS, NOX, RM, AGE, DIS, RAD, TAX, PTRATIO, B, LSTAT*.

- CRIM per capita crime rate by town

- ZN proportion of residential land zoned for lots over 25,000 sq.ft.

- INDUS proportion of non-retail business acres per town

- CHAS Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)

- NOX nitric oxides concentration (parts per 10 million)

- RM average number of rooms per dwelling

- AGE proportion of owner-occupied units built prior to 1940

- DIS weighted distances to five Boston employment centres

- RAD index of accessibility to radial highways

- TAX full-value property-tax rate per $10,000

- PTRATIO pupil-teacher ratio by town

- B 1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town

- LSTAT % lower status of the population

1. The mean of all features is

*[3.61352356e+00, 1.13636364e+01, 1.11367787e+01, 6.91699605e-02, 5.54695059e-01, 6.28463439e+00, 6.85749012e+01, 3.79504269e+00, 9.54940711e+00, 4.08237154e+02, 1.84555336e+01, 3.56674032e+02, 1.26530632e+01]*

1. The standard deviation of all features is

*[8.59304135e+00, 2.32993957e+01, 6.85357058e+00, 2.53742935e-01, 1.15763115e-01, 7.01922514e-01, 2.81210326e+01, 2.10362836e+00, 8.69865112e+00, 1.68370495e+02, 2.16280519e+00, 9.12046075e+01, 7.13400164e+00]*

Question 2. Do linear regression on the dataset from scratch, using Normal Equation. [75 pts]

Equations :

y\_pred = XW + b

For bias, concatenate a column of ones in X as the 0th column :

y\_pred = XW Loss(Mean Squared Error / MSE) = 1/M sum(y\_true\_i - y\_pred\_i)^2

W = (XT**.** X)-1 **.** XT y (all dot products)

1. Train a linear regression model. Compute the weights / model coefficients

(w) by implementing the Normal Equation. Modify the fit() method.

Hint : Utilize numpy library functions to calculate the inverse, dot products and concatenate bias value [15 pts]

1. What is the dimension of the weight matrix (W)? [10 pts]
2. Do predictions using the trained model. Modify the predict() method. [15 pts]
3. Evaluate the model using Mean Square Error (MSE). Modify the loss() method. [15 pts]
4. Fit, predict and evaluate the linear model by calling the fit(), predict() and loss() methods. [10 pts]
5. What is the MSE score for your regression model? [10 pts]

**Solution:**

1. Using the Normal Equation , it is easy to translate into numpy, but at first X needs to append ones

one\_arr = np.ones((m, 1))

X = np.hstack((one\_arr, X))

np.dot(np.linalg.inv(np.dot(X.T, X)), np.dot(X.T, y))

1. The dimension of the weight matrix(W) is (14,1)
2. The method predict() is simply, just dot product of X and W

        # Appends a column of ones to X for the bias term.

        one\_arr = np.ones((m, 1))

        X = np.hstack((one\_arr, X))

        # Do the prediction dot product

        y\_pred = np.dot(X, self.W)

1. The loss function is simple, just calculate mean square of y\_pred and y\_test

        mse = np.square(y\_pred - y\_test).mean()

1. Adds all the methods calls to the main function.

    # TODO Fit, predict and evaluate using linear regression model

    my\_linear\_model = LinearRegression()

    my\_linear\_model.fit(X\_train, y\_train)

    y\_predict = my\_linear\_model.predict(X\_test)

    mse = my\_linear\_model.loss(y\_test, y\_predict)

1. The MSE is around 20.72.