# Data Exploration Assignment

Spring 2018 CSCI490/680

## Introduction

For each task, you should describe how you transformed the data to the format you can perform your analysis on. Describe the issues you faced, how you resolved these issues, and justify why you decided to handle it that way. These explanations should either be Python inline comments or Jupyter Notebook Markdown cells. Additionally, use comments or cells to clearly mark where your code for each task begins.

Ensure that you use Python3 for this assignment. The dataset used in this assignment is the Boston house price dataset from sklearn. It can be imported into Python in this way:

```
from sklearn import datasets
boston = datasets.load_boston()
```

If you do not have sklearn or any other package installed on your machine, you can install it from the terminal with your Python package manager. Below are two examples. This may be different for you depending on operating system, your Python version, and your Python package manager.

```
sudo pip3 install packagename
OR
sudo conda install packagename
```

## Housing Price Analysis (2 points: 1 point for correct stats, 1 point for correct features)

Use the usual summary statistics (count, min, q1, median, q3, max, mean, and standard deviation) to give a brief overview of housing price statistics. Display these in an organized table. Include the summary stats for the price (MEDV or `boston.target`), and the features: age (AGE or data attribute 7), nitric oxide concentration (NOX or data attribute 5), distance from employment (DIS or attribute data 8), and any additional attributes if you wish to. `boston.data` is an ndarray, so you can use 2 indices: the first for house, and the second for attribute. Finding the index for attribute names and accessing them can be done like this:

```
boston.feature_names
# ['CRIM' 'ZN' 'INDUS' 'CHAS' 'NOX' 'RM' 'AGE' 'DIS' 'RAD' 'TAX' 'PTRATIO' 'B' 'LSTAT']
boston.data[0, 2]  # access INDUS from first house
# 2.31
boston.data[:, 2]  # access an entire attribute column (here INDUS) for all houses
# [  2.31   7.07   ...   11.93  11.93  ]
```

## Plotting and Correlation (2 points: 1 for plots, 1 for Pearson value)

For the features you analyzed above, generate: a histogram of the values of each feature, a scatter plot of each feature VS price (you should have six figures in total), and report the Pearson coefficient of the feature with price. The figures should be clearly labeled and titled.

## Correlation Evaluation (1 point for identifying feature performance)

Identify which feature you analyzed relates to price the best, and which relates the worst.

## Home Similarity (3 points: 1 for for correct output, 1 for methodology, 1 for explanation)

Using the four features mentions above, including price, build a distance matrix for these 4 homes using the `euclidean` distance function included in `scipy.spatial.distance.euclidean` or `scipy.spatial.distance.cdist`. For the latter, make sure to check the default distance method in the documentation. Print this matrix with row and column labels (the house index is fine). Identify the most similar pair of houses and explain what the results mean.

| index | MEDV |
|---|---|
| 99 | 33.2 |
| 154 | 17.0 |
| 203 | 48.5 |
| 485 | 21.2 |

## Augmenting House Price Data (2 points: 1 for a link+description, 1 for critical analysis of information and issue)

Finally, you should identify a new dataset that could be potentially integrated with similar housing price data. You should provide a link and description of the new dataset, describe what valuable information the data might add, and identify any issues you might have in integrating the new data.

The dataset does not necessarily need to be for the exact same houses in the Boston dataset. You do not need to actually integrate the two datasets; for this homework, a description is sufficient.

## Hints

- Pandas may be a useful Python package for this assignment. Pandas DataFrames share interoperability with NumPy ndarrays in many cases, and have dozens of useful methods, including methods to compute summary statistics and plot data.
- matplotlib.pyplot may be a useful Python package.
- Remember what a negative correlation means.
- You can use `dir(boston)` to see what the available attributes and methods of `boston` are.
- If you have trouble, check online first, post on the Blackboard discussion forum for help,then finally ask the TA z1754188@students.niu.edu.