# Case Study #1: Predicting Boston Housing Prices

The data set *BostonHousing.csv* contains information collected by the U.S. Census Bureau concerning housing in the area of Boston. The goal is to predict the median house price in new areas based on information such as crime rate, pollution, and number of rooms, etc. The dataset contains 13 predictors, and the outcome (response variable) is the median house value/price (*MVALUE*). The table below describes each of the predictors and the outcome.

| ***Variables*** | ***Description of Variables*** |
|---|---|
| CRIME | Per capita crime rate by town. |
| ZONE | Proportion of residential land zoned for lots over 25,000 square feet. |
| INDUST | Percentage of nonretail business acres per town. |
| CHAR RIV | “Y” if house tract bounds Char river, and “N” if otherwise. |
| NIT OXIDE | Nitric oxide concentration (parts per 10 million). |
| ROOMS | Average number of rooms per dwelling. |
| AGE | Proportion of owner-occupied units built prior to 1940. |
| DISTANCE | Weighted distance to five Boston employment centers. |
| RADIAL | Index of accessibility to radial highways. |
| TAX | Full-value property tax rate per $10,000 (does not depend on the median value of homes). |
| ST RATIO | Student/teacher ratio by town. |
| LOW STAT | Percentage of lower status of the population. |
| MVALUE | Median value (price) of owner-occupied homes in $10000s. |
| C MVALUE | “Yes” for houses with the price equal or greater than $300K, and “No” for houses with the price of less than $300K. |

## Questions

1. Upload, explore, clean, and preprocess data for multiple linear regression.
    a. Create a *boston_df* data frame by uploading the original data set into Python. Determine and present in this report the data frame dimensions, i.e., number of rows and columns.
    b. Display in Python the column titles. If some of them contain two (or more) words, convert them into one-word titles, and present the modified titles in your report.
    c. Display in Python column data types. If some of them are listed as “object’, convert them into dummy variables, and provide in your report the modified list of column titles with dummy variables.
    d. Display in Python the descriptive statistics for all columns in the modified *boston_df* data frame (after converting to one-word titles and dummy variables). Check if there are missing records (values) in the columns. Present the table with descriptive statistics in your report, and comment about the missing values. You don’t need to comment on the values of outliers (min/max) or their extreme values.

2. Develop multiple linear regression with all 13 predictors.
    a. Develop in Python outcome and predictor variables, partition the data set (60% for training and 40% for validation partitions), and train the multiple linear regression model using *LinearRegression()* with the training data set. Identify and display in Python intercept and regression coefficients of this model. Provide these coefficients in your report and present the mathematical equation of this linear regression model.
    b. Using the multiple regression model, identify in Python predictions for validation and training predictors (*valid_X* and *train_X*). Based on these predictions, identify and display in Python $R^2$ and *adjusted* $R^2$ performance measures for training and validation partitions. Present and compare these performance measures in your report and explain if there is a possibility of overfitting.
    c. Identify and display in Python the common accuracy measures for training and validation data set (predictions). Provide and compare these accuracy measures in your report and assess again a possibility of overfitting.

3. Develop multiple linear regression with reduced number of predictors.
    a. Use the *Exhaustive Search* algorithm in Python to identify the best predictors for the multiple linear regression model. Based on these predictors, train a new multiple linear regression model using the respective training data set predictors. Identify and display in Python the intercept and regression coefficients of this model and the common accuracy measures for validation partition. Provide these coefficients in your report and present the mathematical equation of the respective multiple linear regression model.
    b. Use the *Forward Selection* algorithm in Python exactly as discussed in 3a. Provide the same results in your report as discussed in 3a. Also, explain the differences between the best predictors (number and specific predictors used) in the models in 3a and 3b.
    c. Present and compare in your report the common accuracy measures for validation data set of the three linear regression models: with *all predictors*, based on the *Exhaustive Search* algorithm, and based on *Forward Selection* algorithm. Using the value of *RMSE* and the number of variables in each model, which model would you recommend using for making predictions in this case? Briefly explain your answer.