Case Study #4: Predicting Boston Housing Prices with Neural Network

For this case, apply the same data set as for case study #1, *BostonHousing.csv.* It 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 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 neural network modeling. (Part will not be graded as all questions below have been already done in case study #1).
    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.

2. Develop a neural network model for Boston Housing and use it for predictions.
    a. Develop in Python the outcome and predictor variables, partition the data set (60% for training and 40% for validation partitions), display in Python and present in your report the first five records of the training partition. Then, using the *StandardScaler()* function,

develop the scaled predictors for training and validation partitions. Display in Python and provide in your report the first five records of the scaled training partition. Present a brief explanation of what the scaled values mean and how they are calculated.

b. Train a neural network model using *MLPRegressor()* with the scaled training data set and the following parameters: *hidden_layer_sizes=9, solver='lbfgs', max_iter=10000*, and *random_state=1*. Identify and display in Python the final intercepts and network weights of this model. Provide these intercepts and weights in your report and briefly explain what the values of intercepts in the first and second arrays mean. Also, briefly explain what the values of weights in the first and second arrays mean.

c. Using the developed neural network model, make in Python predictions for the outcome variable (*MVALUE*) using the scaled validation predictors. Based on these predictions, develop and display in Python a table for the first five validation records that contain actual and predicted median prices (*MVALUE*), and their residuals. Present this table in your report.

d. Identify and display in Python the common accuracy measures for training and validation partitions. Provide and compare these accuracy measures in your report and assess a possibility of overfitting. Would you recommend applying this neural network model for predictions? Briefly explain.

3. Develop an improved neural network model with grid search.

   a. Use in Python *GridSearchCV()* function to identify the best number of nodes for the hidden layer in the Boston Housing neural network model. For that, consider the *hidden_layer_sizes* parameter in a range from 2 to 20. Provide in your report the best score and best parameter value.

   b. Train an improved neural network model using *MLPRegressor()* with the scaled training data set and the best identified value of the parameter from the previous question. The rest of the parameters remain the same as in model developed in 2b. Present in your report the final intercepts and network weights of the improved neural network model.

   c. Identify and display in Python the common accuracy measures for the training and validation partitions with the improved neural network model. Provide and compare these accuracy measures in your report and assess a possibility of overfitting. Would you recommend applying this neural network model for predictions? Briefly explain.

   d. Present and compare the accuracy measures for the validation partition from the Exhaustive Search model for multiple linear regression in case study #1 and the validation partition for the improved neural network model in this case. Which of the models would you recommend for predictions? Briefly explain.