Homework Assignment # 1: Data Preprocessing and Evaluation

Due: Wednesday, February 24, 2021, 11:59 p.m.
Total Points: 100

This assignment is designed to give you practical programming experience with the data preprocessing and evaluation concepts that were discussed in class.

You have been assigned a partner to work with on this homework assignment, and this information is shown at the end of the assignment. A spreadsheet can be found on the Syllabus page on Canvas with the username for each student. Use this information to contact your partner. Reach out to your partner as soon as possible, to establish your working relationship. You will also be given time to meet each other during lecture. Submit a post to Piazza if you have any issues connecting or working with your partner. Your partner will change for each assignment.

Each individual within the pairing should submit a solution to the assignment, to avoid miscommunication issues with your partner. Be sure to specify in the file, whose submission should be graded, as only one will be graded for the pairing. The submitted homework must include your names, as well as all resources that were used to solve the problem (e.g. web sites, books, research papers, other people, etc.). Academic honesty is taken seriously; for detailed information see Indiana University Code of Student Rights, Responsibilities, and Conduct.

Your assignment must be submitted by committing your code to your IU Github repository before the deadline. Please create a folder called 'hw1' (without quotes) and commit all files for this assignment to this folder. You are strongly encouraged to submit early and often to avoid any last minute issues and to confirm that you are committing your work correctly. The submission should be in the form of a Python 3 Jupyter Notebook file, along with any special instructions for running the program. **Be sure to run the file before committing, so that we can directly see your results**. Programs that fail to run will not be graded and will result in a zero.

Late homework submission policy: We expect all work to be finished and submitted on time. However, we do understand that there might be times when something unexpected comes up which delays you. Hence, as a late submission policy, we will allow late submissions up to 3 days late, each day carrying a 5 point penalty. If you submit 70 hours after the original deadline, the highest possible score for that assignment will be 85 points, out of 100. If the assignment is submit 1 minute after the deadline, then a 5 point penalty will be assessed. Late assignments submitted at 72:01 hours after the original deadline will not be accepted. Please, make the arrangements to start and finish your assignments early. This means you should push to Github well before the time stated on the homework. This means you must start the homework as soon as you can and continually commit.

Question 1. [100 POINTS]

The goal of this problem is for you to estimate the house price based on other features/attributes. Before doing the estimation, you will need to perform data pre-processing first. You will use the housing dataset that is in your IU GitHub repository for this course. **Perform the following steps, in order**, and answer the questions directly in your Jupyter Notebook:

- Create a Python function to load the data from your local directory. Display the top few rows of the data.

- How many data samples are present in the dataset? What attributes/features are continuous valued? Which attributes are categorical?
- Remove the date, street and country attributes.
- Display the statistical values for each of the attributes, along with visualizations of the distributions for each attribute. Explain noticeable traits for key attributes. Are there any attributes that might require special treatment? If so, what special treatment might they require?
- Notice that some of the instances have a value of zero for the house price. Write two functions. One that computes the mean house price from the non-zero instances. One that computes the median house price for the non-zero instances. Note that you should not use the built-in median and mean functions to do this, but should compute it yourself. You can, however, compare the result of your functions to the built-in functions to verify your implementation.
- Create three instances of the dataset: (1) that removes all instances of houses with prices of zero, (2) that replaces the zero-priced houses with the median price value from the non-zero priced houses, using your median function from above and (3) that replaces the zero-priced houses with the mean price value from the non-zero priced houses, using your mean value function above. Generate histogram plots of the house prices for the four cases (e.g. original, $0 priced homes removed, median replaced, and mean replaced).
- For each of the above instances, visualize the dependency of the target on each continuous-valued feature (e.g. 2D scatter plot). What features seem to be linearly correlated with the target? What features do not have a correlation with the target?
- Write your own function that computes the correlation coefficient between the target and the continuous valued features. Do the resulting correlation coefficients agree with your visual inspection of linear correlation? Why or why not? How do the correlation coefficients differ for each set (e.g. original, $0 priced homes removed, median replaced, and mean replaced)?
- Use OneHotEncoder to encode each variable that you deem should be converted, for each data set (e.g. original, $0 priced homes removed, median replaced, and mean replaced). For each variable that you encode, provide a rationale or justification for why it needs to be encoded. Also, provide a rationale for the variables that you choose not to encode.
- For the continuous variables and the ones where one-hot encoding is not applied, separately perform normalization (min-max scaling) and standardization for each of these variables, for each dataset. Hence, you should have eight different versions of the dataset after this step. Show and verify that the data has been converted correctly in each case.
- For each data set (eight in total), us Scikit-learns KFold object to generate 10 folds of training and testing sets for cross validation. Be sure to set the random state variable. For each fold, train a Linear Regression model (see chapter 2 of "Hands-on Machine Learning...") using the training data. Then make predictions of the housing prices of the test data, using the trained model.
- Write a function to compute the mean absolute error and mean square error for each fold of each data set. Report the results and discuss how different factors may have contributed to the final results.