Mario Jelev, mvj140030

CS 6375.502

11/25/18

Project Final Report

**Introduction**

For my project, I selected to participate in the ongoing Kaggle project titled ‘House Prices: Advanced Regression Techniques’. As its name suggests, the dataset offers a large repository of housing data, including foundation types, roof styles, and kitchen quality for a number of homes in various neighborhoods. The goal of the project is to utilize these attributes in predicting the sale price of a house; based on the Kaggle leaderboard page for the competition, prediction accuracy is calculated using the root mean squared logarithmic error.

**Dataset Description**

The training dataset in its original format contains 80 attributes and 1,460 training instances. The majority of the variables are qualitative/categorical in nature and represent various information about a particular housing or residential entity; for example, the shape of the land the building is found on, the type of pavement that constitutes the nearby infrastructure, and the amount of stories the property has. Some of the quantitative variables include the square footage of each floor and the area of the lot frontage. Kaggle offers a data description file that goes more into detail about the variables, and lists all of the unique values for each variable. The fact that most of these attributes are easily understandable means that a deep domain knowledge in real estate and residential science is not necessary to perform machine learning on this data set.

Both the training and test sets include missing data. Some of this missing data appears to occur due to simple lack of information (in both datasets, lot frontage values are among the most commonly mussing values); however, due to the nature of some of the attributes, such as MiscFeature, PoolQC, and BasementFinType2, missing data can sometimes be read as ‘not applicable’. For example, a house without a pool will not have any value for PoolQC, and since most of the residences in the datasets do not offer pools, most of the instances have a null value for this column.

**Preprocessing**

Data preprocessing for this project has been done with the Pandas and Sklearn external Python libraries. Initially, all columns that are missing more than 25% of their values in the training set are identified and then dropped from both the training and test sets. This is to prevent imputation that will ruin the data integrity. Next, constant and quasi constant variables (using a threshold of 95%) are removed to prevent data redundancy.

Imputation for columns missing a relatively significant amount of values (ex: LotFrontage) are imputed using the grouping functionality in pandas; that is, lot frontage values are imputed based on the values of some other columns. It can be assumed the paving of the nearby infrastructure and building type affect the amount of land allocated to the property; for example, a one story property with gravel pavement is most likely in a more rural location and thus probably has a larger lot frontage. Another imputation technique is simple column replacement. For example, the garage quality can be inferred from the kitchen quality, since both the garage and kitchen are typically found on the first floor. Most of the categorical columns that are missing data are missing a very small amount ( < 5%), so in these cases data is imputed using simply the mode of the column.

Correlation matrices are also used during the preprocessing of the data set. Columns that have an absolute value of correlation that exceeds 80% are dropped to avoid data redundancy.

Value ‘bucketing’ was used during preprocessing to lessen the chance of dimensionality increase during categorical encoding. For example, there is a column denoting the month the house is sold; in the default case, one hot encoding would increase dataset dimensionality by 12 (one new column for each month); so, value mapping was used to map each month to a season (months 12- 2 to winter, 3- 6 to spring, etc.). This ultimately increases dimensionality by only 4 columns. Neighborhoods were also analyzed to find the different general price tiers and were ultimately replaced by their representative tier (some neighborhoods were a part of ‘expensive’ communities; others ‘poor’.) This was done for several columns that had a significant amount of unique values; columns with only a handful of values were encoded as is.

In order to standardize the continuous columns, a standard scaler from the Sklearn library is used; this ensures the variables have zero mean and unit variance, which is necessary for the successful fitting of many models.

**Results**

After the preprocessing phase, GridSearchCV is used to try out instances of different algorithms. The algorithms used in the project included Bayesian Ridge Regressors, decision trees, bagging, random forest, Adaboost, and gradient boosting. Different levels of PCA were also done on the dataset, although best results were obtained without altering the components of the dataset.

The below results compare the mean squared log error for some of the algorithms mentioned. These were obtained during GridSearchCV during a run that used 5 fold CV and no PCA (all 307 features were used).



