ORIE 4741 Midterm Report

Philip Ayoub (pja66) and Kevin Cushing (kxc4)

November 2, 2021

1 Initial Data Analysis

1.1 Overview

For our ORIE 4741 Final project we are using the data set 'house-prices-advanced.csv' to gain insight into what determines the "price-tag" of a house. The data set consists of 81 features and 1406 instances ranging in values and data-type. Among these features there are 31 real valued columns, 26 categorical columns, 18 ordinal columns, 1 Boolean and 1 Regression column (Y-Values).

1.1.1 Real valued columns

These features consist of columns such as 'LotArea', PoolArea', and 'GarageArea'. All the data in these columns are integers. We embed these numerical features as is in the training of our models.

1.1.2 Categorical columns

These features consist of columns such as 'Utilities', 'SaleCondition', and 'Heating'. These columns all have string values that describe the qualitative values of the house.

1.1.3 Ordinal valued Columns

These features consist of columns such as 'OverallQual', 'GarageQual', and 'GarageCond'. The features in this category are strings that mainly describe the conditions of different features of the house.

1.1.4 Boolean Valued Columns

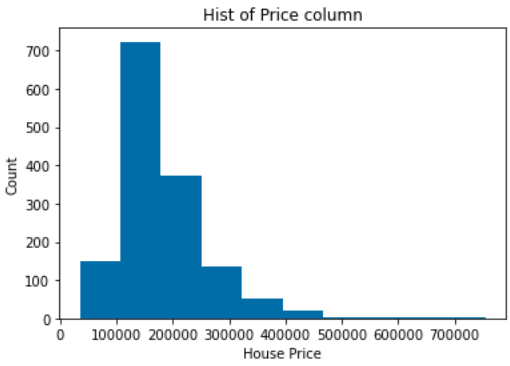
There is only one feature that falls into this category, 'CentralAir', which carries values of 'Y' and 'N' representing yes or no.

1.1.5 Corrupted/Missing Values

There where a small but note worthy amount of missing values in our initial data set. However, there were no corrupted values which helps verify our confidence in the data set. We classified missing values as Nan or inf values and corrupted values as non-sense values such as negative numbers in a feature that describes area.

1.1.6 Y-Column

We did a number of checks to determine the validity of our Y-axis column being used for regression but the main sign that the data is trust worth is the normal distribution generated when you plot a histogram of the values.



1.2 Data Cleaning/Preproccessing

1.2.1 Handling Missing Data

We encountered many missing entries throughout our data set. For numerical valued features, we replaced missing entries with 0, since these occurred for features like 'PoolArea' for houses without pools. Another possible way to do this would have been to add another feature to indicate whether the house had a pool or not, however, this is unnecessary since there are corresponding categorical features like 'PoolQual' that are also NA and our one hot encoding of categorical features essentially does this already.

1.2.2 Categorical Values

For the features in our data set that fell under this category we choose to use one-hot encoding to transform these features of originally datatype string to datatype integer.

1.2.3 Ordinal Values

For the features in our data set that fell under this category we choose to give integer values based on the order of the value in question. For instance, many of the instances in this category of features describe the condition of an aspect of the house, ['nan','Po', 'Fa', 'TA', 'Gd', 'Ex']. Thus we transformed these values into [0,1,2,3,4,5] in order to transform these strings into integers.

1.2.4 Boolean Values

For the features in our data set that fell under this category we choose to change the values of true to 1, and values of false too 0.

1.2.5 Standardizing Data

We decided to use Z-score normalization to further clean our data set:

```
df_scaled = finalDF
for column in df_scaled.columns:
    df_scaled[column] = (df_scaled[column] - df_scaled[column].mean()) / (df_scaled[column].std()+1)
finalDF = df_scaled
```

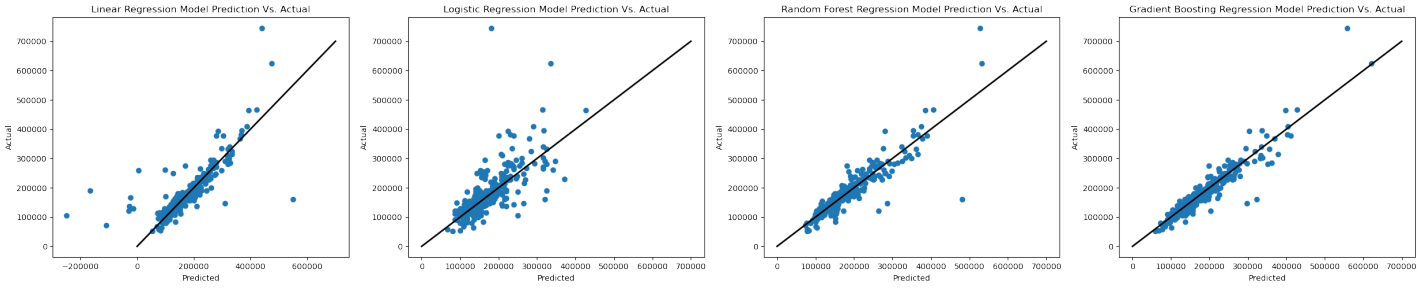
2 Plan to avoid over (and under-)fitting

We are attempting to prevent over-fitting by using k-fold cross validation on each model in order to select the model that generalizes best as opposed to the model that performs the best on the data it was trained on. Using k-fold cross validation allows us to train our model on different subsets of our data, and then validate it using the remainder of our data. This gives our model selection an advantage over simply using a single validation set since a single validation set could be slightly biased, causing the selection process to favor certain models that may not truly be the best model of the set. Using k-fold cross validation allows us to validate with multiple validation sets in order to give a better estimate of how the models will perform on unseen data. We plan to avoid under-fitting

my training our models on as much data as possible. We will also examine certain features that may be making our models perform worse and exclude them in order for our models to fit better.

3 Preliminary Analysis

3.0.1 Simple Regression Model Performance



3.0.2 Linear Regression

The first basic regression model we tried was a Least Squares Linear Regression. As is evident by the graph, this was the worst model at predicting the sale price of houses,and the only model that predicted negative sale prices for some houses. There are also many outliers for which this model drastically mispredicted the sale price. The root log MSE (the loss function we are using to measure our models' performance) for this model using 5-fold cross validation was 0.34384317849549184.

3.0.3 Logistic Regression

The next regression model we tested the performance on our data set was a logistic regression. As can be seen from how tight the points on each graph fit the ground truth line, this model seems to predict prices that tend to be more inaccurate than the other models in general, however, there appear to be fewer outliers than in the linear regression and no negative values, making this model overall perform slightly better than the linear regression model. The root log MSE for this model using 5-fold cross validation was 0.2494546835909405

3.0.4 Random Forest Regressor

Judging by the graphs, it is clear that this model performed better than the previous two, in that most data points fit tightly around the ground truth line and there seem to be far fewer outliers than both the linear and logistic regression models. The root log MSE for this model using 5-fold cross validation was 0.1439212138327585

3.0.5 Gradient Boosting Regressor

Again, referring to the graphs, this appears to be our best performing model at a glance since almost all of the data points are positioned tightly around the ground truth line and there are fewer outliers than each of the other models. The root log MSE for this model using 5-fold cross validation was 0.1290686149651774

4 Next Steps

Going forward, we will continue to experiment with new types of models, as well as improve our feature embeddings in order to achieve a model that performs better than our current model. We will also try selectively excluding some features from our models that we believe may not have an impact on the sale price of a house, or features that are highly correlated with other features in order to reduce the likelihood of over-fitting. Once we achieve an acceptably low validation error, we will go on to create a model that generalizes well over time, training it on a subset of data from a given range of years, and then validating it on data from a separate range of years. In order to make this model, we may need to add in some kind of economic data so the model can deal with changes over time like inflation. Creating a model that generalizes well over time will be challenging due to the large fluctuations that occur in real estate prices throughout time.