Rick Lattin

Lydia Malone

Armin Charkhkar

Fall 2023

OREM 7331 Project 2

Team Name: Rick_Lydia_Armin

Project 2 Report

In this project, we were tasked with entering into a Kaggle competition that required us to predict the dollar value of various houses, based off of a dataset of features. We were instructed to make said predictions through an implementation of creative feature engineering and regression techniques. Our prediction submissions were then graded based off of the Root Mean Squared Error between the logarithm of the predicted value and the logarithm of the actual sales price. Furthermore, these submissions got ranked amongst the other competing teams.

The first step we took towards cleaning and processing the training dataset was to drop the columns that had a substantially high number of missing values. With such a high percentage of unavailable data, these columns would have been insignificant in helping build our prediction model. The columns we took out were: PoolQC, MiscFeature, Alley, and Fence. The next step in cleaning and processing the training dataset was imputing any missing values. For the categorical variables like FireplaceQu and garage-related attributes, we chose to input ‘none’ for the missing data, which suggests those houses do not feature these attributes. For GarageYrBlt, where the absence of data implies the absence of a garage, we entered a 0, keeping the numerical data type of the data and distinguishing these houses from those with garage years. For the features where there were next to no missing values, such as MasVnrType and MasVnrArea, we replaced the few missing points with the mean from the data, in an attempt to preserve the integrity of this section of data. For LotFrontage, we filled in missing entries with the median value from the available data. The median is a good fit for this purpose because it's less influenced by extreme values, providing a balanced approach for a variable that can have a wide range. Lastly, we imputed missing Electrical data using the mode, since this is a categorical field with few missing entries. This method keeps the houses with missing electrical data within the most frequently occurring category, preserving the existing distribution of the dataset. After all the data was cleaned and ran through the linear model there remained a few missing values in the final output. We replaced these with an estimation of the mean house price from the dataset before sending the output off to be scored by Kaggle.

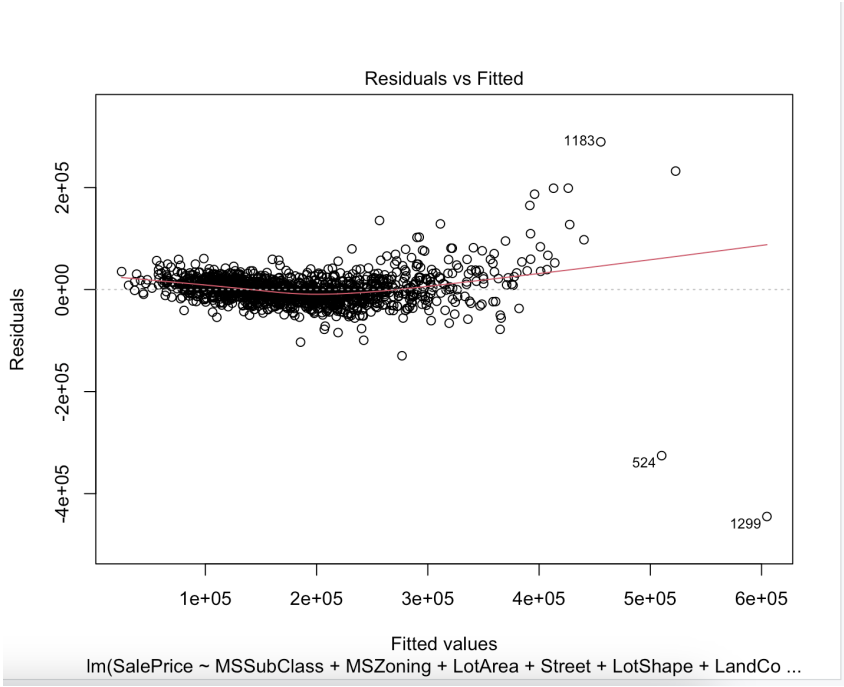
Our initial approach to predicting house prices was to construct a linear regression model. We used the training data set, where the target variable was the Sale Price of the house, to train the model. All the other variables in the dataset served as predictors, aiming to estimate their influence on the Sale Price. We also attempted to implement a neural network in order to train a prediction model, but that proved much more difficult then we were expecting. The data that we

were working with, despite all of the cleaning, did not seem to be sufficient for training with a neural network, as the network would often just refuse to train, or would take an unreasonably long time to train as we tested different hyperparameters. Even when it would return a trained model, the values that the model would return would be the same for every house. This indicates that although we were able to create a model from the neural network, the model did not function properly and was unable to predict any housing prices.

We believe that linear modeling is much better at predicting prices for the data set we were given to work with than neural network modeling. We believe this is due to the amount of outliers and imputed data members that were contained within the data. Although we did a fair amount of data cleaning, it was not enough to adjust for the variety in meaning in the different data columns, the outliers and the data that could have been irrelevant that we included. The linear model was better able to cut through the noise and find the trends in the data where the neural network fell short.

One of the main issues that came with this project was the cleaning and processing of the data. For the most part in this class, the data we have used for in class assignments, as well as the previous project, was fairly clean. This project had the extra steps of not only doing a thorough cleanse of the data, but we also had to impute the data corresponding to various factors. Additionally, we needed to decide what the best model would be, in order to best carry out our predictions. There are a myriad of ways to approach this problem, and we decided a linear regression model would be the most straightforward way to go.

We did not end up implementing any major packages from outside of class in our final code, but we did experiment with the *data.table* and *tidyr* packages when working with the data in how to clean it and manipulate it. We ended up creating a new file to clean up our code and decided we did not need to use those functions for what we needed to do as far as data cleaning, and therefore the packages were removed. We did have to change how we used the neural network model implementation that was given to us in class in order to accommodate for the fact that we were not doing classification for this assignment, but that took nothing more than changing one variable and a google search for clarification.



This is a model of the results of the training of our linear regression model, and it very clearly shows how accurate the model is in its predictions through how close the dots are clustered to the line. Although this only displays the model in respect to the training data, it still lends to the relative proficiency of the linear model in its prediction task.