Final Project: Final Report

Tatiana Kennedy, Hunter Moreton, Madeline Navigato

Introduction

The House Prices Kaggle competition challenges analysts to look at 79 different variables for residential homes in Ames, Iowa to most accurately predict the sale price for the home. These variables describe almost every aspect of a property to give a more detailed price estimate. Our goal is to find which variables have the biggest influence on sale price so that with new data we can accurately predict Sale Price. This can be valuable information for realtors, real estate investors, as well as current and future homeowners.

There are two datasets for this competition; train which has 1460 rows and test which has 1459 rows. The train and test datasets have 81 and 80 columns. The difference in the datasets is the response variable 'SalePrice' is not included in the test data. The datasets have null values that have to be redefined so that the variables can be used in the model. After cleaning the data we will use a multiple linear regression model to best predict Sale Price on new data.

Data Modeling and Cleaning

We discovered that there were a lot of missing factor levels and values within the test and train datasets. To overcome this issue, we joined the two datasets together using the rbind() function. We then went to work on cleaning up missing levels and N/A values within the joined dataset.

For the features representing the quality or condition of the house, within the scale from Poor to Excellent, we decided to set their scale to a variable called `ordered_levels`. We then assigned numbers to each level on that scale – 'None' = 0, 'Po' = 1, 'Fa' = 2, 'TA' = 3, 'Gd' = 4, 'Ex' = 5. We then revalued the feature's levels to be integers, assigning the `ordered_levels` value to each of their respective levels. We assigned a handful of other feature levels to be integers where it made the most sense, too. For the most part, we made sure that any categorical variable was set to factor.

For most of the features that had missing categories, we decided to set their N/A values to 'None' or the most common value. However, with data that had a lot of missing features, but were logically unable to be set to 'None', we had to think more about their fill value. For example, with some of the Garage variables: we found GarageType with a value other than 'None', but other Garage variables to be N/A. With intuition and help of commonality of other levels, we manually assigned the levels. We found this issue present itself again with the Basement variables. We handled it by applying the most common level to each of the missing variables.

For the features that had missing numeric values, we did fill most with 0. However, with certain variables that were N/A, we did fill with the median value, like with the variable LotFrontage.

We made two new variables, `TotalSF` and `TotalBa`, to represent the total square footage of the lot and the total bathrooms, respectively. `TotalSF` was calculated by adding the `GrLivArea` and `TotalBsmtSF` together. `TotalBa` was calculated by adding the `FullBath` to half of the quantity of `HalfBath` and then adding that to `BsmtFullBath` plus half the amount of `BsmtHalfBath`.

After we finished cleaning the joined dataset, we split the data back into their respective datasets, train and test, making sure to check that the N/A SalesPrice data went into the test data

Model and Model Development

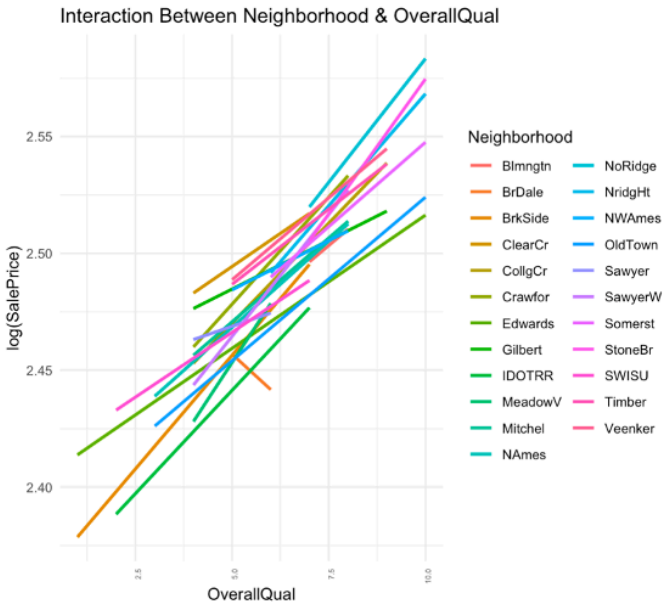
The modeling method we used for predicting Sale Price was a multiple linear regression This method is good for both descriptive and predictive goals. However, we wanted to focus on the predictive goal and use a multiple linear regression that can be used to predict Sale Prices on new data.

We ended up using all of the variables available in the data in hopes to maximize prediction efforts for our model. . In our Interim report we only used 5 predictors `OverallQual`, `GrLivArea`, `GarageCars`, `MSSubClass`, and `Neighborhood`. From our heat map below we can see that these variables had strong correlations to SalePrice. Our model using only 5 predictors resulted in an R-Squared of 0. 8327 on the test dataset, so we knew by adding more if not all of the variables after thorough data cleaning it would result in a higher R-Squared.

In our model, we logged the outcome variable which was Sale Price. Even though Sale Price is somewhat normally distributed, we logged the variable as price tends to have an exponential trend and a relatively large range.

The model contains two polynomials. These polynomials were added to the variables TotalBsmtSF (total square feet of basement area) and GrLivArea (above ground living area square feet). Both of these variables were plotted against Sale Price and had a nonlinear relationship. These two variables were also highly correlated with Sale Price. By adding polynomials to these predictor variables, model performance increased.

The model also contains an interaction. There is an interaction between Neighborhood (physical locations within Ames city limits) and OverallQual (rates the overall material and finish of the house). To determine if there was an interaction between these variables, they were conceptualized with the plot below:



Based on this plot, we can conclude that there is an interaction between OverallQual and Neighborhood. The slopes for each Neighborhood vary and are not the same. An interaction model is needed for these variables as it allows the slopes to vary and make the model more predictable. Both of these variables also have a significant impact on Sale Price.

Model Performance

The model had acceptable in-sample and out-of-sample performance metrics. The model performed great on the train set with an in-sample performance of an adjusted R-squared of 0.9355 and a log RMSE of 0.1015, which is an RMSE of 17556.55 in original units. In-sample, the model explains 93.55% of the variation in Sale Price.

For the test set, the out-of-sample performance measurements for the model was an adjusted R-square of 0.8519 and an estimated log RMSE of 0.1575. Out-of-sample, the model explains 85.19% of the variation in Sale Price. These out-of-sample performance statistics were derived using cross-validation. The model performed slightly better in the train set than it did in the test set. This means that the model is slightly suffering from overfitting but will still perform very well with new samples of data.

When submitting our model predictions into Kaggle, our model submission got a leaderboard rank of 1991 and a Kaggle score of 0.13406. The Kaggle score is also the returned log RMSE of 0.1341. Overall, the model did an exceptional job at predicting house prices for both the train and test sets. However, the model could be improved by slightly reducing overfitting.