Andrew Seal

Capstone 2: Ames, Iowa – Housing Data

1. Introduction:

The US Census Department reports that 28.9% of US household wealth is in the form of home equity, making housing the second most significant store of wealth behind retirement accounts[1].

In fact, for most people in the United States, the purchase of a home is one of the largest financial transactions they will ever make[2]. Homes are not traded on a liquid market and unlike some other assets the market price of real estate is often difficult to determine. This can leave buyers without a reliable mechanism to determine if they are paying an appropriate sum for a given house.

The goal of this project is to create a model that will use the distinct features of homes to predict the Sales Price of a given house. This type of model could be used in a variety of business settings including by real estate agents who wanted a tool to advise clients or by institutional investors who were looking to purchase properties or by lenders who want to ensure the purchase price of the mortgages they are underwriting are reasonable. Although the dataset is unique to a specific location, this type of analysis could be performed for residential and commercial properties in any location where information from past sales is available.

2. Dataset

For this project, I utilized a publicly available dataset of recent home sales in the town of Ames, Iowa. The dataset consisted of 2,930 distinct sales within the town of Ames between 2006 and 2010. In total there were 82 columns in our data set, each representing a feature of the home. Features include quantifiable characteristics of the home such as:

- Total Square Footage
- Lot Size
- Number of Bedrooms
- Number of Fireplaces…etc.

Additionally, the features include categorical data such as:

- Zoning
- Neighborhood
- Condition of the home
- Whether the garage is attached or detached …etc

[1] US Census Bureau: https://www.census.gov/content/dam/Census/library/publications/2020/demo/p70br-170.pdf

[2] National Association of Realtors: https://www.nar.realtor/research-and-statistics/research-reports/highlights-from-the-profile-of-home-buyers-and-sellers

The original data was provided in a .CSV file and was imported into a pandas data frame called housing_data.

3. Data Cleaning and Data Wrangling

Each home in our dataset was assigned a ‘Parcel ID’ which is a unique identifier that I used to ensure no home sales showed up in our dataset more than one time.

With no duplicated data in our dataset, I graphed all of the numerical data into histograms in order to visually inspect the data for outliers or other anomalies. Some of these features showed outliers, for example the lot frontage on a few parcels was around 300 feet. While wider than most, a lot width of 300 feet seemed reasonable for a smaller town in Iowa and unlikely to be an error.

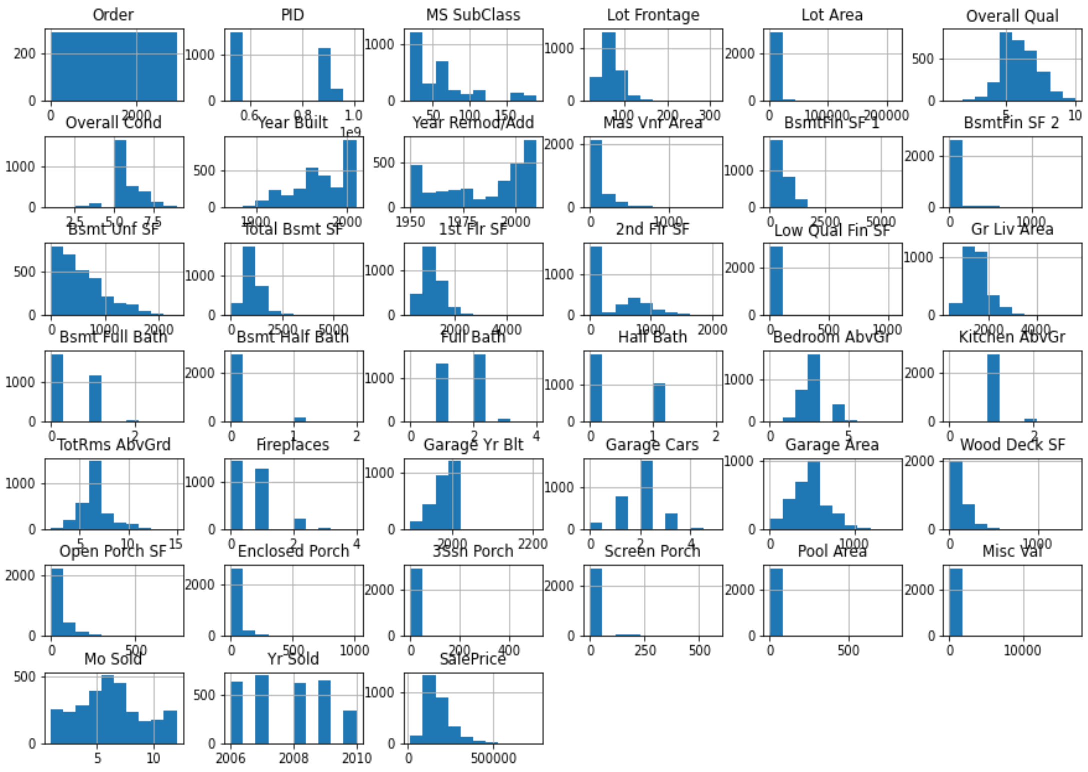


Figure 1. Histograms of numeric data created to visually inspect for outliers and anomalies.

During this step I also looked into missing data and into data that were mostly homogenous. The dataset included four variables where more than 80% of the data were missing. These were examined individually and one variable “Misc Feature” was deleted as it was determined that the value of each miscellaneous feature identified in this column was reported in a separate column.

Similarly, some variables were dropped because the data was exceedingly homogenous. For example, a column named ‘Utilities’ was dropped since all but three of the 2,930 sales had the same value indicating they were connected to public utilities.

4. Exploratory Data Analysis

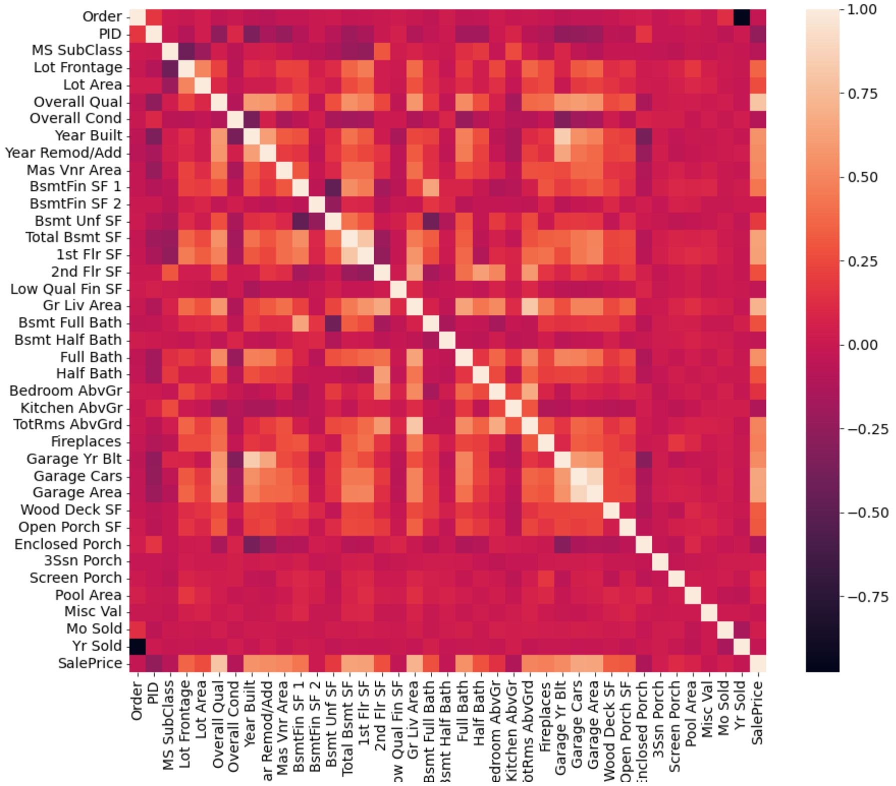


Figure 2. Heatmap of correlations between variables. For this analysis particular attention is paid to the correlation between sales price and other variables.

In the exploratory data analysis step, the relationship between variables is assessed. Of particular importance is the relationship between Sales Price and other variables. In the above heatmap we can see a positive correlation between Sales Price and a few variables including Overall Qual, Gr Liv Area, Full Bath, Garage Cars and others.

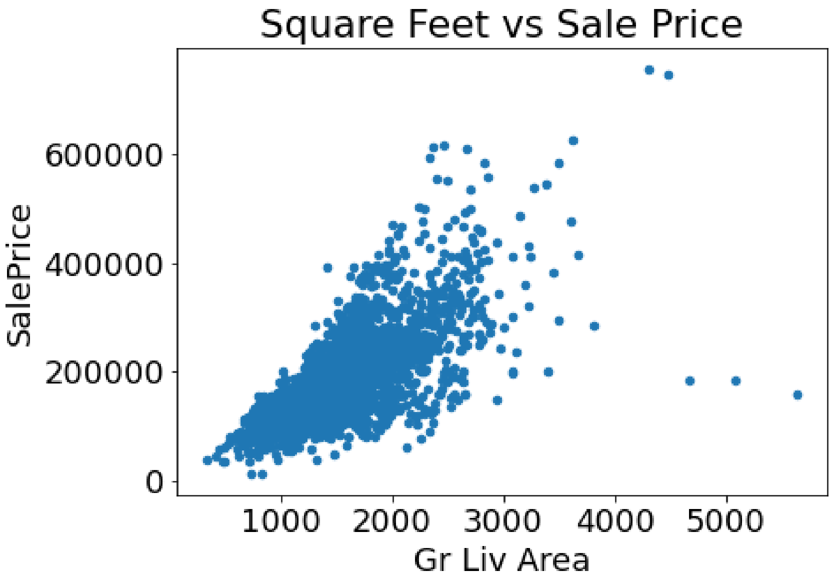


Figure 3. Scatterplot of square footage vs sales price

A strong positive relationship between Gross Livable Area (square feet) and sales price is apparent in Fig 3 above. The Pearson correlation coefficient between square feet and sales price is 0.714.

Likewise, Fig 4 below demonstrates that a logical relationship is found between sales price and the number of cars the garage will hold where homes without garages sell for a discount while those with three-car garage command a higher sales price. The Pearson correlation coefficient between these two variables is 0.661.

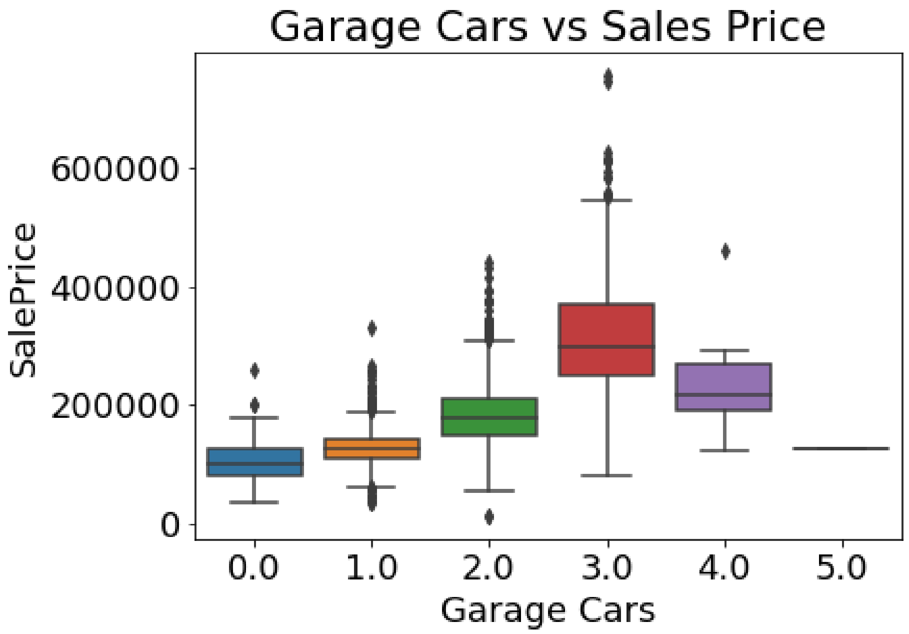
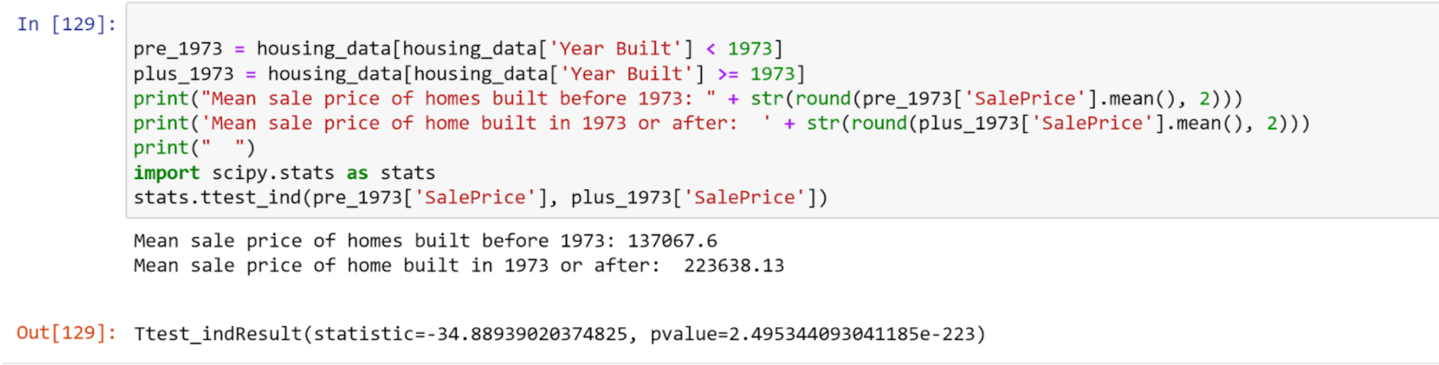


Figure 4. Boxplot showing sales price vs capacity of garage (number of cars)

One additional variable of note is the year built. As shown in Figure 6 below, recently constructed homes appear to command a high sales price. The median year in which homes in our dataset were constructed was 1973. Homes constructed prior to 1973 commanded a mean sales price of $137,068 whereas homes constructed in 1973 or later sold for an average (mean) price of $223,638. An independent t-test can check for statistically significant differences in the mean of two groups. In order to evaluate whether this difference in means is statistically significant, an independent t-test was conducted comparing the sales price of homes constructed prior to 1973 and the sales price of homes constructed in 1973 or later. The p-value was very near to zero meaning we can reject the null hypothesis that the mean sales price of homes constructed before 1973 are equal to the mean sales price of homes constructed in 1973 or later .



```
In [129]:  pre_1973 = housing_data[housing_data['Year Built'] < 1973]
           plus_1973 = housing_data[housing_data['Year Built'] >= 1973]
           print("Mean sale price of homes built before 1973: " + str(round(pre_1973['SalePrice'].mean(), 2)))
           print('Mean sale price of home built in 1973 or after:  ' + str(round(plus_1973['SalePrice'].mean(), 2)))
           print("  ")
           import scipy.stats as stats
           stats.ttest_ind(pre_1973['SalePrice'], plus_1973['SalePrice'])

           Mean sale price of homes built before 1973: 137067.6
           Mean sale price of home built in 1973 or after:  223638.13

Out[129]:  Ttest_indResult(statistic=-34.88939020374825, pvalue=2.495344093041185e-223)
```

Figure 5. Code snippet of t-test comparing the mean sales price of homes built before 1973 to those built in 1973 or after.

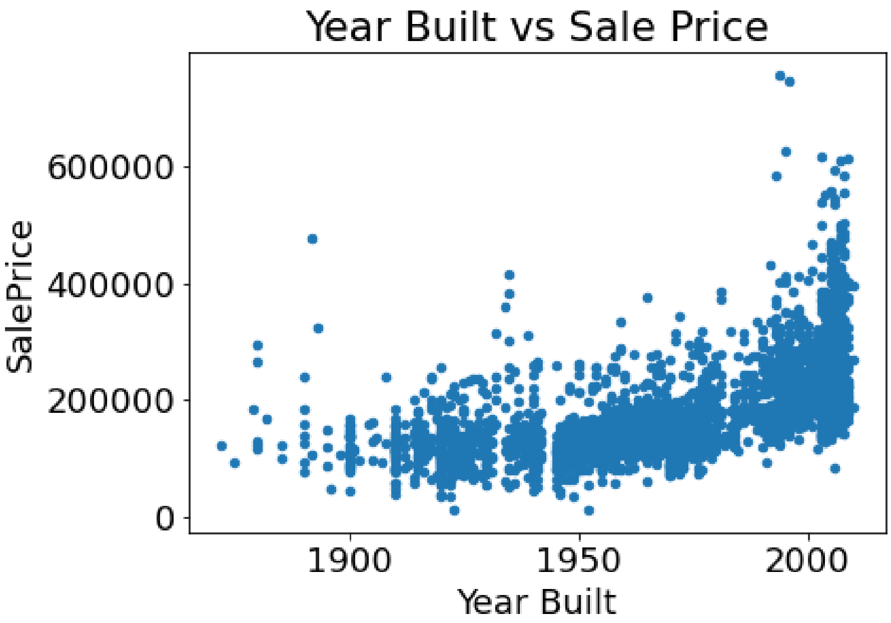


Figure 6. Scatterplot showing relationship between Year Built and Sales Price.

Finally, during this step I ran a Random Forest Feature Importance analysis.

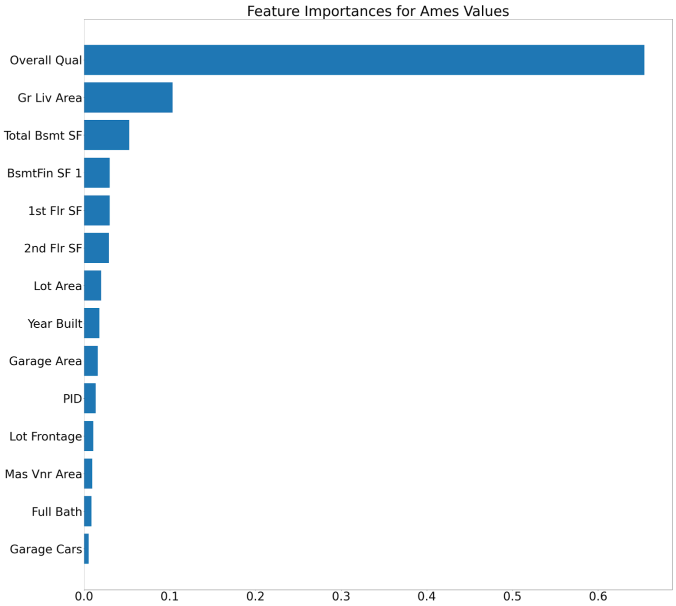


Figure 7. Random Forest Feature Importance

The results of the feature analysis, shown above, indicate that by far the most important variable in the analysis was the 'Overall Quality' Variable. Unfortunately, there was limited information available regarding how this quality was assessed or measured and by whom. Unlike square footage and lot size I had concerns about whether this variable would be consistent among various datasets including using this model on new and unseen information. Ultimately, I determined that it would be best to remove this variable altogether.

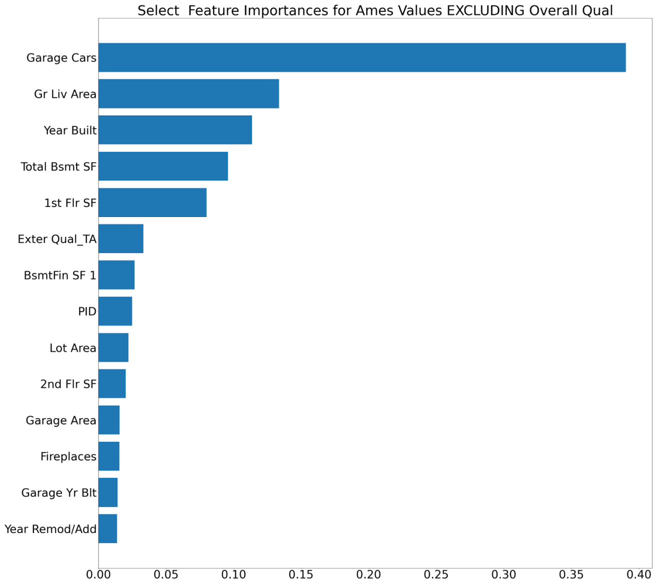


Figure 8. Random Forest Feature Importance Excluding Overall Quality

Once the variable was removed, the Random Forest Feature Importance analysis was run again. What is interesting is that when we included the 'Overall Quality' variable, the 'Garage Cars' variable was only our 15th most important variable. When the 'Overall Quality' variable was removed the 'Garage Cars' feature was identified as the most important variable. The top results of the Random Forest Feature Importance Excluding the' Overall Quality' variable are shown in Figure 8 above.

5. Pre-Processing

During this stage some missing values were imputed. A total of 490 home sales (16.7% of the total) were missing the 'Lot Frontage' variable. Instead of removing this column, the missing data for this one variable was imputed with the mean.

At this stage, I was careful not to impute data for garages, pools, air conditioning or other features that may not be present in every property.

After imputing data and creating dummy variables, I dropped any missing data and the resulting data frame had 2,747 rows (unique sales) and 244 variables (columns).

6. Regression Analysis of Feature Importance

Addressing multicollinearity

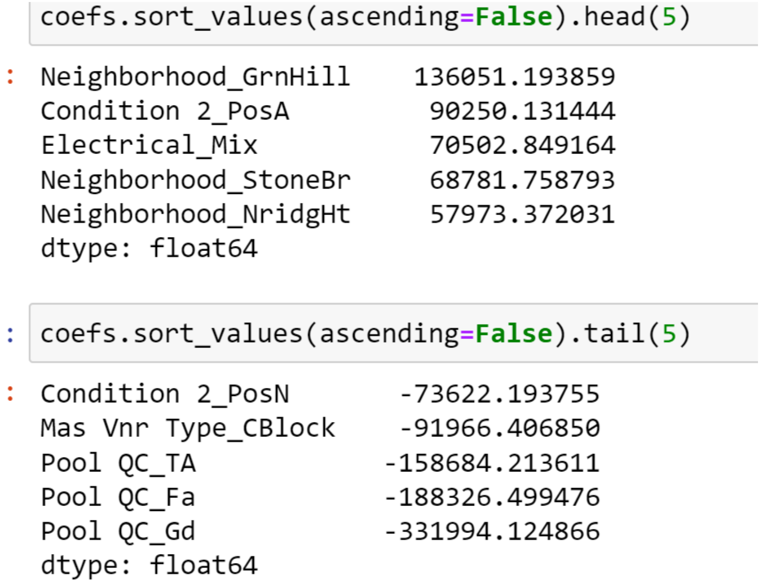
The first step was addressing simple collinearity among my variables. To do so, I examined all variables with a correlation coefficient of 0.95 or greater with any other variable and then simply eliminated one of the variables. In this step I eliminated four variables.

Next, I examined a measure of multicollinearity called the Variance Inflation Factor (VIF). In this step I used some value judgements to eliminate variables with an emphasis on retaining variables who were shown to be important in the Random Forest Feature Importance exercise. The first variables to be eliminated were redundant dummy variables. For example, condition good and condition fair convey similar information. At this point I eliminated many of the zoning variables as I suspected they may be correlated with other location variables such as 'neighborhood' and because they were not shown to be important in our Feature Importance analysis. This process was repeated using my best judgment until the remaining variables all had a VIF score below 10.

Linear Regression Feature Coefficients

At this stage a significant amount of multicollinearity had been removed from our dataset so another feature importance analysis was conducted, this time using OLS regression.

The first step was to evaluate the coefficients of the OLS regression of our unscaled data. The unscaled coefficients are in their original units and the interpretation is therefore straightforward. The summary table includes around 200 variables but the top and bottom 5 variables are presented in figure 9 below.



```
coefs.sort_values(ascending=False).head(5)
```

```
:  Neighborhood_GrnHill    136051.193859
   Condition 2_PosA         90250.131444
   Electrical_Mix           70502.849164
   Neighborhood_StoneBr     68781.758793
   Neighborhood_NridgHt     57973.372031
   dtype: float64
```

```
:  coefs.sort_values(ascending=False).tail(5)
```

```
:  Condition 2_PosN        -73622.193755
   Mas Vnr Type_CBlock     -91966.406850
   Pool QC_TA             -158684.213611
   Pool QC_Fa             -188326.499476
   Pool QC_Gd             -331994.124866
   dtype: float64
```

Figure 9 Top 5 and Bottom 5 coefficients in unscaled OLS regression

Some of these coefficients are not reasonable. The variable with the greatest impact on our price is the dummy variable indicating a swimming pool with quality assessed as ‘Good’. This coefficient indicates that the presence of a good quality pool detracts $331,994 from the sales price. While a pool may detract from the sales price of a home, the magnitude seems unreasonable since a pool could be filled-in for much less than $300K. It is also worth noting that only four homes in our dataset had a good quality swimming pool.

The next step was to standardize the data. Using a Robust Scaler, all of the independent variables were standardized. The standardized data was fit into an ordinary least squares regression and the coefficients were extracted from the regression. Given the data were standardized, the coefficients should represent the relative impact of the features on the Sales Price.

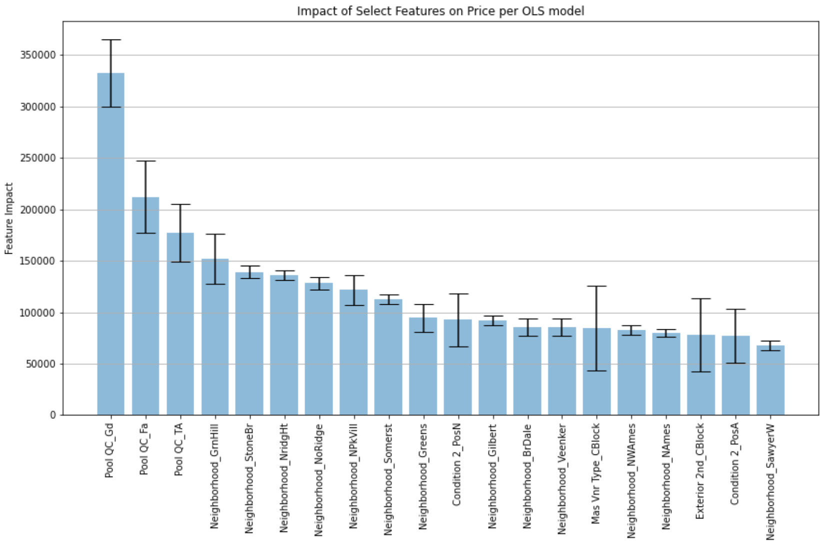


Figure 10. Impact of Select Features on Price per Scaled OLS model

Interestingly, the features identified as most important on the OLS model did not match what we had previously identified as important in the Random Forest Feature Importance. In the OLS model, many of the variables identified as important were relatively rare features like the swimming pool variables and dummy variables for specific neighborhoods. The variables identified by the OLS model differ from those identified in the Random Forest Feature importance because the OLS regression does not take into consideration how many properties are affected by the variable, and only considers the magnitude of the effect.

It was determined that it may not make sense to build a model around these fairly rare features so a second Feature Importance analysis was conducted.

7. SHAP Feature Importance

The second Feature Importance Analysis that I used was the Shapley Additive Explanations (SHAP) which is a measure of the average marginal contribution of a feature across all the possible combinations of features. This analysis, presented below, was more consistent with both logic and the Random Forest analysis previously conducted.

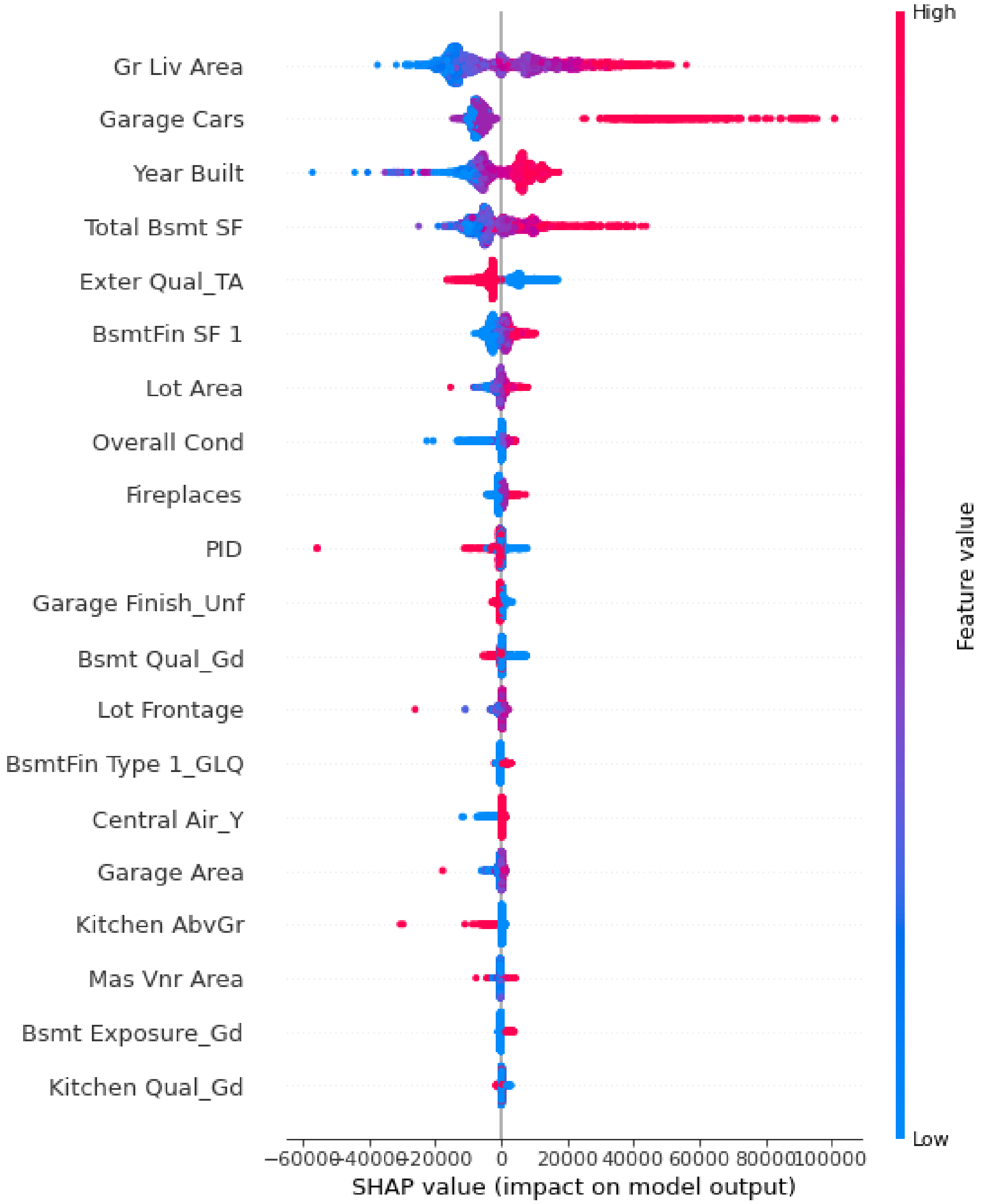


Figure 11. SHAP Feature Importance

8. Train Test Split

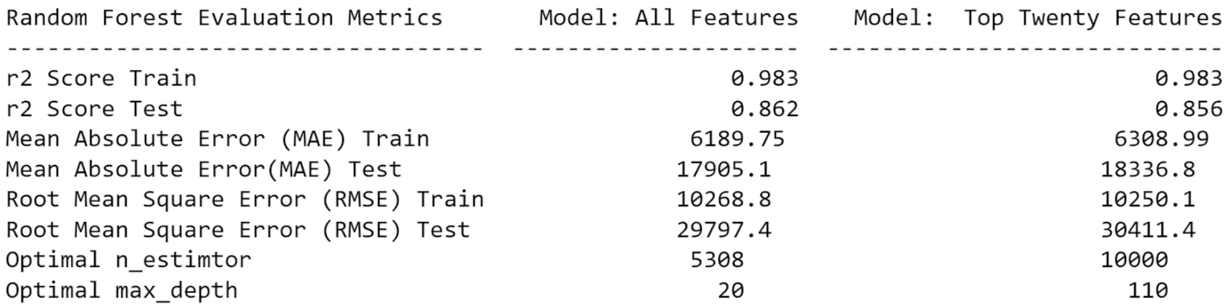
In advance of modeling, I split the data into a training set consisting of 75% of the recent sales and a training set consisting of the remaining 25% of the data. Each of the models was fit and optimized on the training set and then tested on the test data in order to evaluate the performance of the model.

9. Modeling

The housing data was evaluated using three different models, Random Forest , Ridge Regression and Gradient Boosting. Each of the three models was run first using all of the features in our dataset and then utilizing only the top twenty features as identified in our SHAP Feature Importance analysis. For both feature sets the relevant hyperparameters were tuned and the results of each model were compared in order to select the best model.

Random Forest

The first model employed was Random Forest. For this algorithm I elected to optimize both the n_estimators and the max_depth hyperparameters. I then used a random search to optimize the hyperparameters. When modeling the entire feature set, it was determined the optimal hyperparameters were n_estimator of 5,308 and max_depth of 20. When modeling only the top 20 features the optimal parameters were n_estimator of 10,000 and max_depth of 110.



```
Random Forest Evaluation Metrics      Model: All Features    Model:  Top Twenty Features
-----------------------------------  ---------------------  -----------------------------
r2 Score Train                                      0.983                          0.983
r2 Score Test                                       0.862                          0.856
Mean Absolute Error (MAE) Train                   6189.75                        6308.99
Mean Absolute Error(MAE) Test                    17905.1                        18336.8
Root Mean Square Error (RMSE) Train              10268.8                        10250.1
Root Mean Square Error (RMSE) Test               29797.4                        30411.4
Optimal n_estimtor                                5308                           10000
Optimal max_depth                                   20                             110
```

Figure 12. Random Forest Evaluation Metrics

For the Random Forest model, using all of the features resulted in a slightly more robust model with both a higher r-squared and a lower root mean square error. This model shows a strong relationship between actual and predicted values.

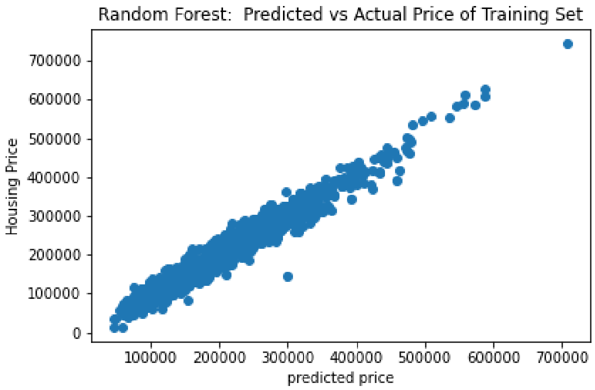
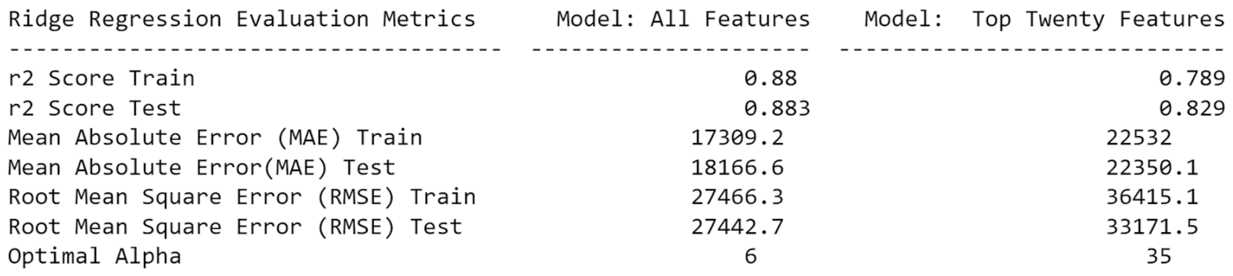


Figure 13. Random Forest Scatter plot of Actual vs. Predicted Sales Price (Training Set)

Ridge Regression

Ridge Regression is useful when multicollinearity still exists between the variables. Given the large number of features in our model, a ridge regression was used to model the data set.

The hyperparameter of alpha was optimized using Grid Search cross validation and a ridge regression was run using the optimized alpha.



```
Ridge Regression Evaluation Metrics      Model: All Features    Model:  Top Twenty Features
------------------------------------  ---------------------  -----------------------------
r2 Score Train                                         0.88                          0.789
r2 Score Test                                          0.883                         0.829
Mean Absolute Error (MAE) Train                    17309.2                       22532
Mean Absolute Error(MAE) Test                      18166.6                       22350.1
Root Mean Square Error (RMSE) Train                27466.3                       36415.1
Root Mean Square Error (RMSE) Test                 27442.7                       33171.5
Optimal Alpha                                          6                            35
```

Figure 14. Ridge Regression Evaluation Metrics

Once again, if we compare r-squared and RMSE, then using the full feature set resulted in a better model than using the top twenty features from our SHAP feature importance.

For the ridge regression I plotted the residuals to visually ensure linearity, and a constant variance. As shown below, nothing unusual was detected in the plot of the residuals.

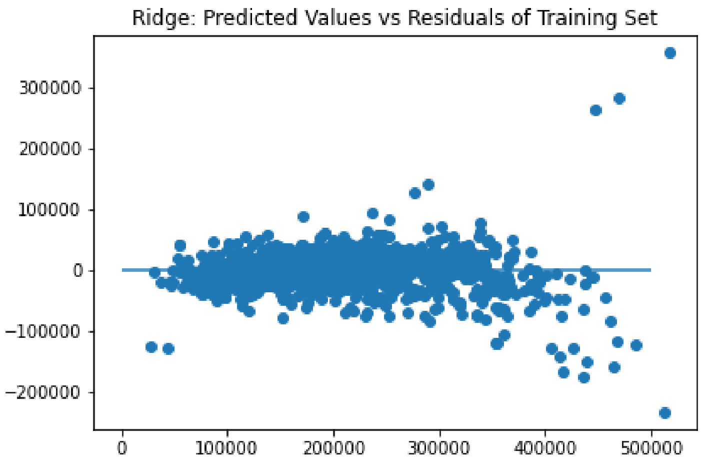
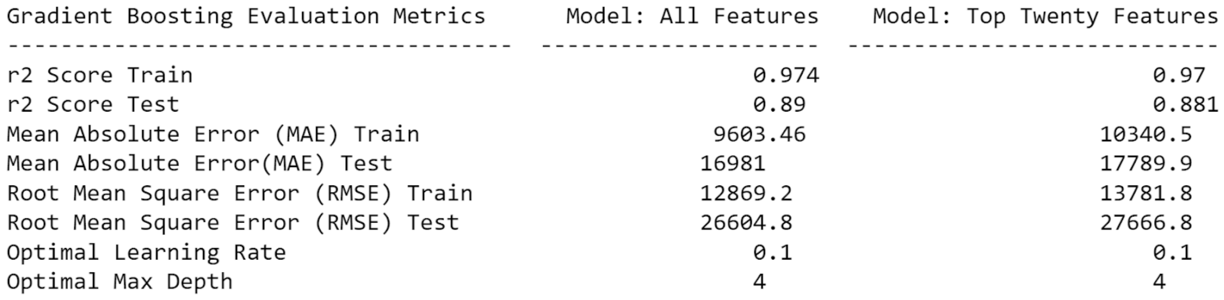


Figure 15. Predicted Values vs Residuals Ridge Regression (Training Set)

Again, this model produced a satisfactory R-squared and a slightly lower RMSE than the Random Forest model.

Gradient Boosting

The final algorithm was a Gradient Boosting model. Utilizing a randomized search cross validation method, I optimized the learning rate and the max depth hyperparameters. As with our other models, using the full feature set resulted in the most robust model. This model performed slightly better than our other model when measuring both the R-squared and the RMSE. Some evaluation metrics of the Gradient Boosting model are shown below.



```
Gradient Boosting Evaluation Metrics      Model: All Features    Model: Top Twenty Features
--------------------------------------  ---------------------  ----------------------------
r2 Score Train                                          0.974                          0.97
r2 Score Test                                           0.89                           0.881
Mean Absolute Error (MAE) Train                      9603.46                       10340.5
Mean Absolute Error(MAE) Test                       16981                          17789.9
Root Mean Square Error (RMSE) Train                 12869.2                        13781.8
Root Mean Square Error (RMSE) Test                  26604.8                        27666.8
Optimal Learning Rate                                   0.1                            0.1
Optimal Max Depth                                       4                              4
```

Figure 16. Gradient Boosting Evaluation Metrics

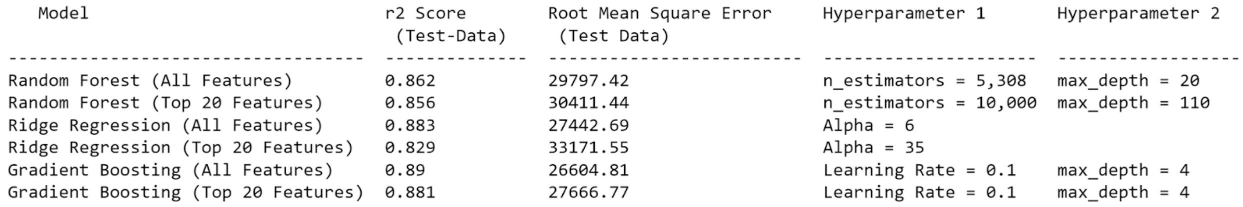
Selecting a Model

For this project I used three different regression algorithms to model the Ames, Iowa housing data with the objective of finding the best model to predict a dependent variable (Sales Price) based on the features of the property. Each of these three algorithms was evaluated using both the full feature set (all of the variables) and a reduced feature set with the top 20 variables as identified in our SHAP feature importance.

For each model I used a grid (or randomized) search cross-validation to optimize the hyperparameters.

A key metric used to compare these three models was their R-squared, which is the extent to which the variance in the independent variables influences the variance in the dependent variable. This is often referred to as the 'Goodness of Fit' and is an indication of how well the data fits the regression model. A higher R-squared will indicate a model that creates a better fit of the data.

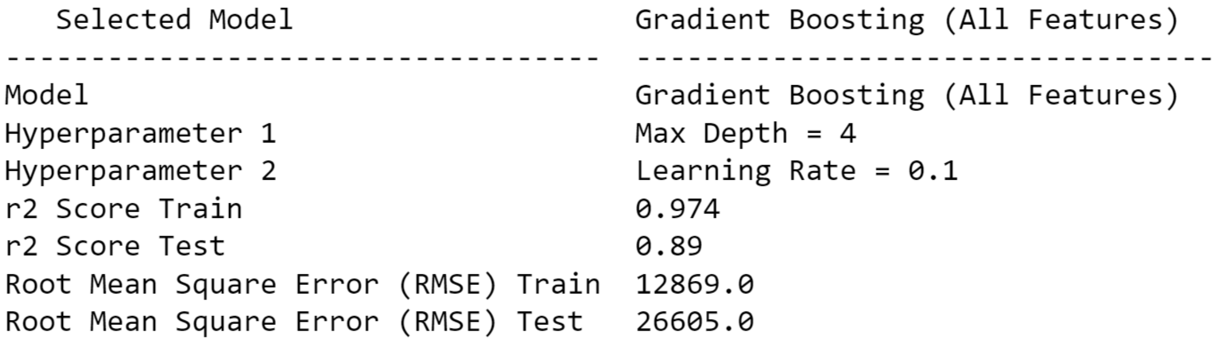
Of our models, the Gradient Boosting (All Features) yielded the best R-squared on the test data.



```
   Model                              r2 Score       Root Mean Square Error     Hyperparameter 1       Hyperparameter 2
                                       (Test-Data)    (Test Data)
----------------------------------  -------------  ------------------------  --------------------  -----------------
Random Forest (All Features)        0.862          29797.42                  n_estimators = 5,308   max_depth = 20
Random Forest (Top 20 Features)     0.856          30411.44                  n_estimators = 10,000  max_depth = 110
Ridge Regression (All Features)     0.883          27442.69                  Alpha = 6
Ridge Regression (Top 20 Features)  0.829          33171.55                  Alpha = 35
Gradient Boosting (All Features)    0.89           26604.81                  Learning Rate = 0.1    max_depth = 4
Gradient Boosting (Top 20 Features) 0.881          27666.77                  Learning Rate = 0.1    max_depth = 4
```

Figure 18. Comparison Metrics of the three models

Since the objective of the model is to determine the price of a real estate offering where the sales price is unknown, the r2 Score on the test data (not the Training Data) was used for comparison. Given the superior performance of the Gradient Boosting (All Features) model both in terms of its R-Squared and RMSE on the test data, I would select the Gradient Boosting Model (All Features) and expect it to most precisely predict the Sales Price.



```
   Selected Model                    Gradient Boosting (All Features)
----------------------------------  ----------------------------------
Model                               Gradient Boosting (All Features)
Hyperparameter 1                    Max Depth = 4
Hyperparameter 2                    Learning Rate = 0.1
r2 Score Train                      0.974
r2 Score Test                       0.89
Root Mean Square Error (RMSE) Train 12869.0
Root Mean Square Error (RMSE) Test  26605.0
```

Figure 19. Select Metrics of Selected Model - Gradient Boosting (All Features)

10. Running a Test Case

To illustrate the functionality of the model constructed I randomly selected one property from our test data and used the gradient boosting model to predict the sales price of the home.

On this particular home, the model predicted a house price of $247,968. When I looked up the actual sales price of this home, it in fact sold for $226,000. While a single sale is not a good indicator of the performance of the model, it is nonetheless fun to put the model to use in a manner that simulates its real-world application!

11. Conclusion

The goal of this project was to build a model that would use the features of recently sold homes to predict the sales price of a home in that same market. In the end, the tuned model was able to predict house price within an average error of ~$26k. This seems to be well within a range that could be useful though further conversations with potential stakeholders could shed light on how great of an issue this error would be.

Such stakeholders might include: :

- Real Estate agents advising their clients
- Investors valuing potential purchases
- Mortgage Lenders conducting due diligence prior to lending etc

Although our data were limited to single family residential real estate transactions in Ames, Iowa, this type of analysis could be conducted for multifamily, condominium, or commercial real estate in any number of geographical locations.

12. Future Work

A significant amount of time was spent optimizing the hyperparameters and each of our models included either one or two hyperparameters to tune. However, it is likely that including additional parameters in a future analysis could result in a more robust model.

Furthermore, while we assessed each model using two feature sets, additional feature sets i.e. top 50 variables may provide a more robust model.

Finally, the data available is more than a decade old, so future analysis could include revisiting with more updated data or evaluating different types of real estate including condominiums, or commercial buildings.