IS 6489 Final Project

Owen Horne, Michael Davis, Priyanka Mitra

Introduction

The aim of this project is to predict the final sales prices for homes in Ames,Iowa. The first part of the project required us to submit an interim report using only five predictors. For that prediction, we created a linear model based on the historical data of the houses in the area. The training dataset consisted of 1460 observations and 80 explanatory variables. The test data consisted of 1459 observations and all the predictor variables except Sales Price, the outcome variable.

For the final project we further evaluated the training dataset and through cleaning, combining of variables, and impuatation created a better set of training data. We then began to build models with which we were able to significantly improve our predictions over the interim model.

Exploratory Data Analysis and Cleaning

The dataset consists of both character and integer variables, where most of the character variables are actually factors. In total, there are 81 variables (character/integer), where the last variable is the Outcome variable(SalePrice).

The first part of the project consists of Data Pre-Processing. It is necessary to have clean training data so that our final model will have greater chances of having better performance. Data Pre-Processing can be divided into 3 major parts:

- Detecting and dealing with missing values
- Normalizing distribution of predictor variables
- Handling outliers

1. Detecting and Dealing with Missing Values

First of all , we detected all the missing values in each column and found that there are more missing values in categorical variables than numeric variables, and the highest missing value percentage is more than 80%. For those columns with highest amount of missing values , we decided to drop the variables and by doing so our data and model performance will not be affected. See Table 1.

Table 1: Table Displaying Variables having more than 80% NA values

Variable	Number_of_Missing_values
Alley	2721
PoolQC	2909
Fence	2348
MiscFeature	2814

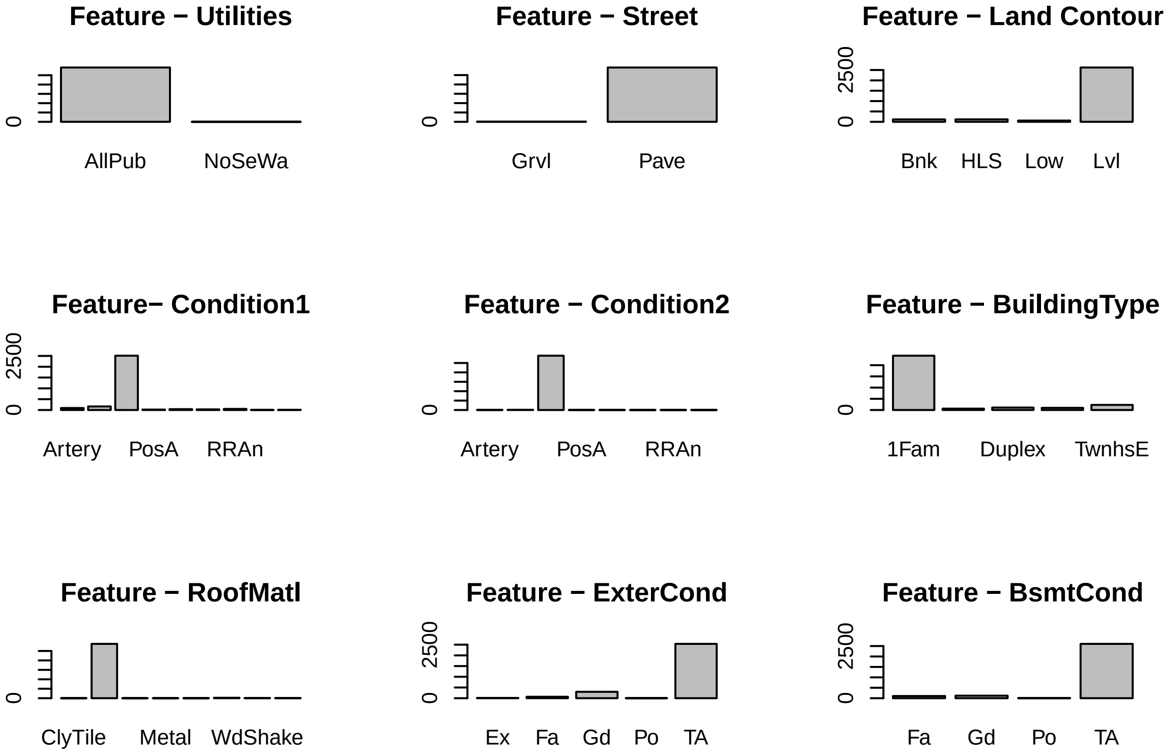
Filling NA Values

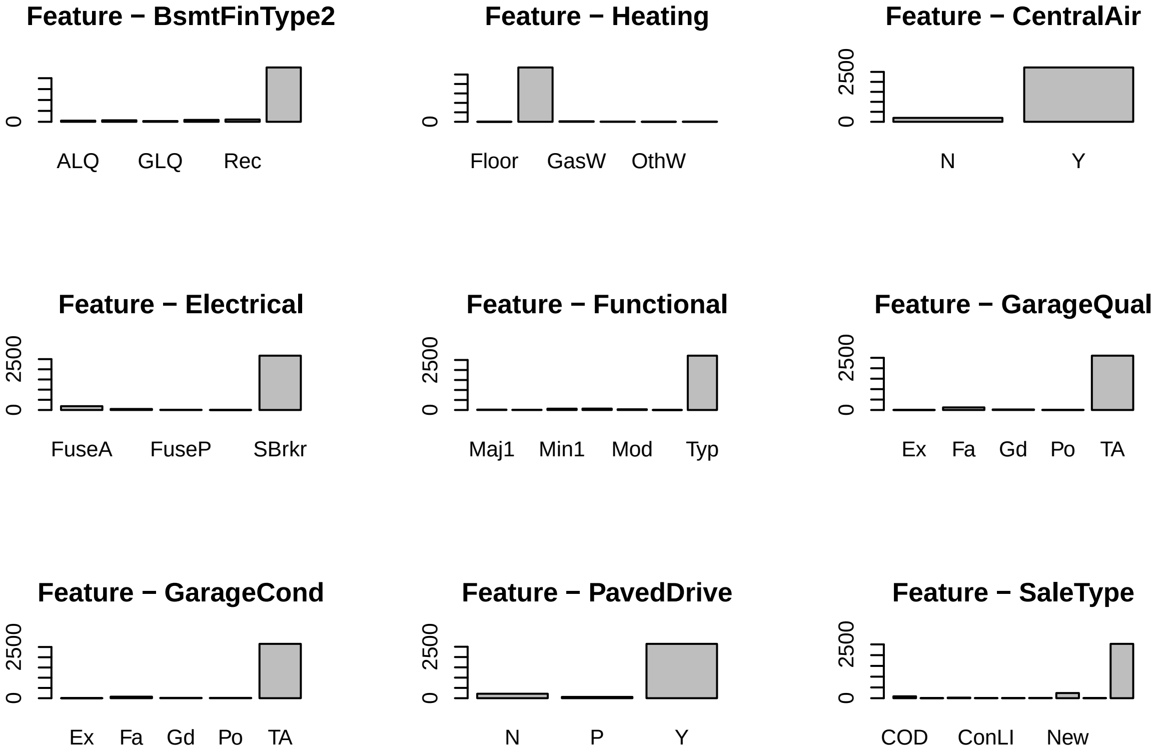
Many variables had NA values that needed to dealt with and were filled with values which made the most sense.For example, in categorical variable such as GarageType , the NAs represented an absence of garage in the property. Therefore, the NA values have been replaced with string 'None' indicating No Garage. Similarly, the NA values for other similar categorical features have been replaced with 'None'. For the numerical featureslike GarageYrBl, NA values will be replaced by 0.

Removing Variables with a Single Dominant Feature

On further analysis , we foud that there are many categorical variables which have one feature as the major feature i.e it dominates over more than 80% of data for that particular column. For example, consider the variable *Street* which has “Pavement” as the dominant feature and contributes to 99% of the data.

Keeping such variables in the model will not contribute much to the model and its performance and hence , we decided to remove such variables from model so that the number of variables in the final model is optimal and clean. The below plot shows the graphical representation of the all the variables which have a single dominant feature





Converting Numerical Variables to Factors

There are 3 variables that are recorded as numeric but actually should be categorical.

- Month and Year Sold : For both the variables, converting them into factors makes much more sense than keeping them as integers.Converting the Year Sold into factors will group the houses sold in that particular year together.This will make modelling easier and explain the variance in SalePrice for each year.Also, changing Month Sold into factors will describe the effect of seasonality on the SalePrice.
- MSSubClass: MSSubClass identifies the type of dwelling involved in the sale.These classes are coded as numbers ,but are really categories.Fo example, MSSubClass 80 represents "Split or Multi-Level" dwelling.

Addition of New Variables

On further exploration of the housing data ,we decided to add 7 more variables in the dataset. This variables depict an important feature , based on the combination of variables already present in the dataset. The variables added in the dataset can be explained as follows:

1. TotalSqFeet : The new variable created is one of the most important features of the house which gives the total squarefeet of the house and makes it much easier to be compared with other house areas.
2. TotBathrooms : Instead of comparing each variable related to feature Bathroom of the house, a variable Totbathrroms has been added which denote the Total Number of Bathrooms in the house.This variable is made of the Bathrooms of the house and the ones present in the Basement.
3. TotalPorchSF : Similar to TotalSqFeet , this variable provides information about the Total Porch Area of a house.
4. PropAge : Calculated from YearSold and YearBuilt, the PropAge(Property Age) will help in determining whether the value of a property should increase or decrease based on its value.
5. Remod : This is a boolean variable which will give output 1 if the property was remodelled and 0 if the property was not remodelled.

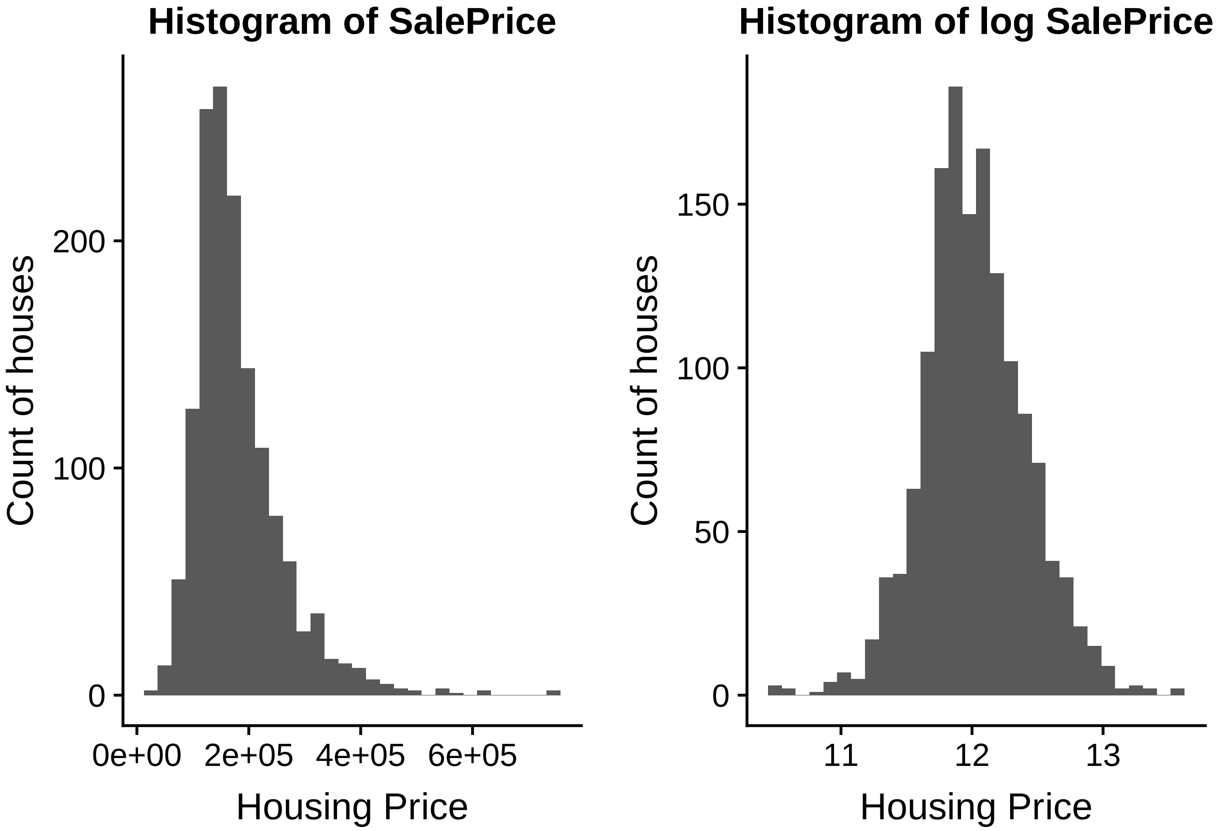


Figure 1: Comparison of distribution between SalePrice and log(SalePrice)

6. LastRemod : This variable will provide information that if a particular property was remodelled , then which year it was re-modelled. This feature will also affect the SalePrice of the house.
7. IsNew : For our data, we have considered that if the difference between YearSold and YearBuilt is less than 15 years , then the property is a new property (denoted as 1) or else it will be old (denoted as 0).

2. Normalizing distribution of target variable

First, the regression requires all variables in the model to have normal distributions.Therefore, we created a histogram for our predictor Sales Price to analyze the distribution. See Figure 1. From the plot we could see the distribution of sale price is clearly skewed and one way to deal with skewed data is to log-transform the variable.

The plot on the right shows after the sales price is log-transformed and its distribution is almost the shape of the normal distribution. Before we fit the regression model , we pre-processed with log-transform of Sale Price and we are going to use the logged Sale Price for our final model.

3. Other Cleaning and Imputation

One important aspect we considered in cleaning our dataset was looking for outlier values that may impact our model in a negative way. These outlier values could be due to many different things, but we are mostly concerned with values that don't quite make sense in the context of the dataset. Meaning values that may have been a result of a data entry error (e.g. values that are not possible) or values that are too extreme to be explained.

In our exploratory data analysis, we focused on ensuring all the numerical values made sense. As explained earlier, we found many NA values that really meant 'None' or zero, but in this analysis we are mostly

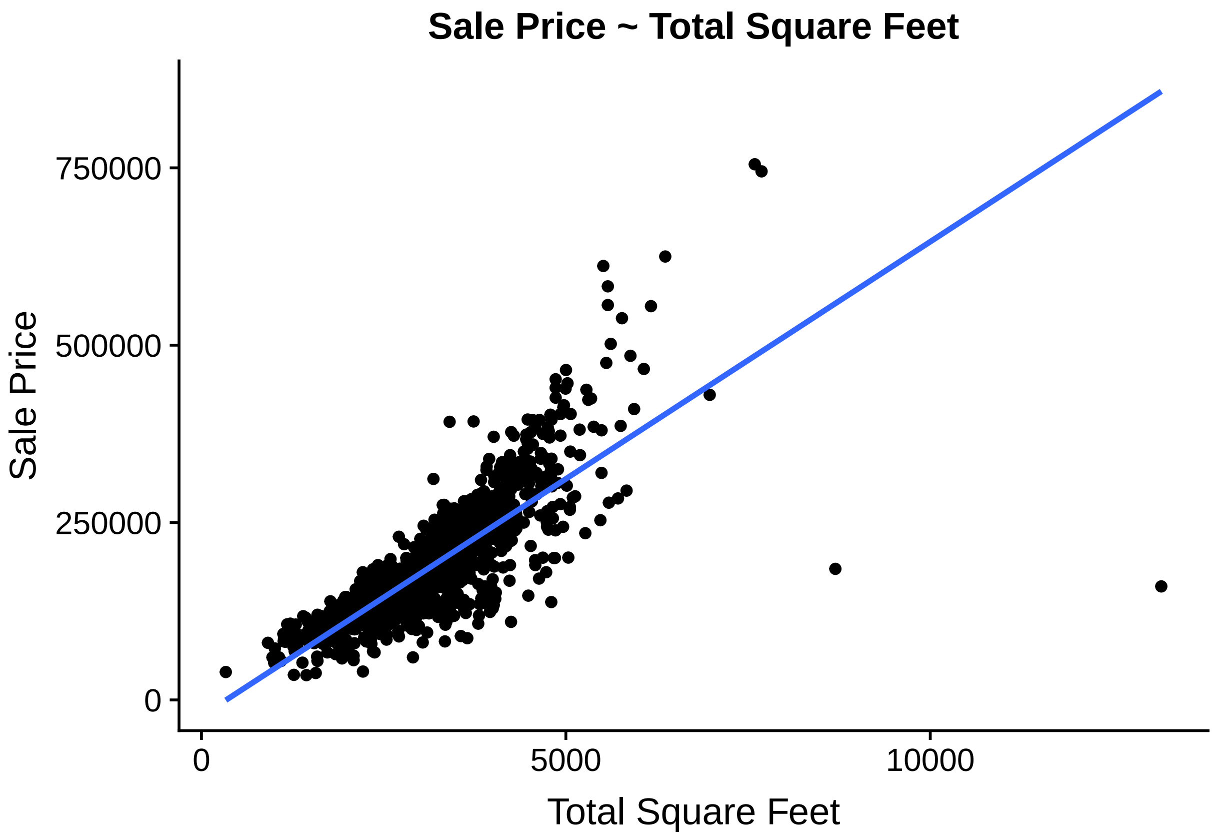


Figure 2: Plot showing relationship between Sale Price and Total Square Feet

concerned that the numerical values are possible. To do this we wanted to take a closer look at the remaining numerical values.

After carefully obvserving these values, especially paying attention the the minimum and maximum values, we discovered one error. In the GarageYrBlt variable (Garage Year Built), there is a maximum value of 2207, meaning the garage for that property was built in year 2207, which is not possible.

After closer inspection of this property, specifically of the year built values, our best guess was the GarageYrBlt value for this property was entered incorrectly and should be the year 2007.

Next, we searched for outlier values. We started this search by looking at a very strong predictor, one of our created features, TotalSqFeet in relation to SalePrice.

In Figure 2 we see a very strong linear relationship between TotalSqFeet and SalePrice, but do notice two properties that do not fit the relationship at all. These properties have a very high square footage, but were sold for an average price. As we dug deeper into the outliers there is not much explanation but we did notice that both these properties are located in the Edwards neighborhood which may offer more insight. See Table 2.

Table 2: Table Displaying Size and Price in Edwards

	TotalSqFeet	SalePrice
176	3936	243000
446	3605	127500
524	8698	184750
725	4164	320000

	TotalSqFeet	SalePrice
1169	3775	235000
1299	13170	160000

Still, after comparing to other large Edwards neighborhood properties, the '524' and '1299' properties sell for about the same price as properties one-half or even one-quarter the square footage. In addition, both of these properties are the only properties with a total square foot value above 8000 in the entire training dataset. We decided to remove these values as they seem to be extreme in relation and cannot be explained, which figures to be detrimental to the model performance.

Imputation

Missing values in data sets are problematic in data analysis, model training and prediction. Hence, we will impute missing values before going further in model training. For our project, we have imputed missing values using the MissForest package which will do single imputations using random forest.We haven't used other imputation methods such as knnImpute or medianImpute as they work for only numeric variables and we will have to convert the factors to Integers.

EDA and Cleaning Summary

Post-model development, we compared a lasso model performance of a model with the outlier values and a model without the outlier values, with all other aspects of the model the same. The lasso model performed much better after removing the outlier values and so we proceeded with that training data set.

Model without deleting outlier values: RMSE = .1716971, r-squared = .8331531

Model after deleting outlier values: RMSE - .1151516, r-squared = .9174832

Model Development and Variable Selection

Interim Model

The interim model was limited to five variables. We noticed many statistically significant variables, but limited to five we chose: LotArea, OverallQual, TotRmsAbvGrd, YearBuilt and GrLivArea. After choosing our variables, we developed a linear regression model.

Final Model Development

Due to the amount of variables in the data and a high rate of suspected variable collinearity we decided that a regularized regression model would work well. We hoped that a regularization model would help in simplifying the model by removing variables and increasing prediction.

We fit and tested lasso, ridge, and glmnet models. We also tested a SVM (Support Vector Machine) model, which is different from the lasso, ridge and glmnet models as it is a machine-learning algorithm, typically used for classification problems but is very useful for regression problems. We fit this model and using the e1071 package and found that it also did not perform better than the lasso model on it's own.

In order to prevent overfitting out-of-sample as well as reduce the variance of a single model, we decided to combine, or stack, the four models we developed into a singular model. We did this by assigning equal weight to each model to make a singular prediction. This helps reduce the variance because with combining, we are essentailly taking the average of the variance for each indidividual model. We found this method to improve our model performance, over the performance of the individual lasso model, which was our best individual predictive model. Below, we will explain and compare our performance metrics with each of our models as well as our performance of our final model: the stacked model combining all four.

Prediction Results and Fit Metrics

Interim Model

- Estimated RMSE - .1678447
- Estimated Rsquared - .8241553
- In sample RMSE -.1680826
- In sample Rsquared - .82

Lasso

The lasso model performed as we had hoped in simplying the model and dealing with collinearity. It selected less than 40% of the variables.

- Estimated RMSE - .1151145
- Estimated Rsquared - .9175369
- In sample RMSE - .1078874
- In sample Rsquared - .9270983
- Kaggle rank:1274 and score:0.12324

Ridge

- Estimated RMSE - 0.11856
- Estimated Rsquared - .9125161
- In sample RMSE - .103214
- In sample Rsquared - .9332773
- Lasso estimated better

Glmnet

- Estimated RMSE - 0.1151592
- Estimated Rsquared - .9175396
- In sample RMSE - .1082117
- In sample Rsquared - .9266594
- Lasso estimated better

Stacked

- Kaggle Rank 1179 Score .12184

While the lasso model removed many variables simplifying the model this was a prediction projest. By taking the mean prediction of each model the stacked model created the best prediction and score.