Dream House Finder
Ames, IA

BUDT758T – Team 2
Lorenzo, Abhishek Shinde,
Rohan Shetty, Ashish, &
Ngoc-Quyen
Spring 2019

Table of Contents

BUDT758T: Data Mining and Predictive Analysis

Data Mining for Business (BUDT758T)

Project Title

Dream House Finder Ames, IA

Team Members

Abhishek Shinde	-	116226011
Rohan Shetty	-	116194303
Ngoc-Quyen Vu	-	115080119
Ashish Verma	-	116220194
Lorenzo Bartolini	-	113348273

Original Work Statement

We the undersigned certify that the actual composition of this proposal was done by us and is original work.

Contact Author	Typed Name	Signature
abhishek.shinde@rhsmith.umd.edu	Abhishek Shinde	
rohan.shetty@rhsmith.umd.edu	Rohan Shetty	
ngoc-quyen.vu@rhsmith.umd.edu	Ngoc-Quyen Vu	
ashish.verma@rhsmith.umd.edu	Ashish Verma	
lorenzo.bartolini@rhsmith.umd.edu	Lorenzo Bartolini	

I- Executive Summary

Ask a homeowner to describe their dream house, they probably won't begin with the height of the basement ceiling or the proximity to an east-west railroad. There is a lot more that influences price negotiations than the number of bedrooms or if it has a white picket fence. Our analytical solution will provide value to home buyers and sellers be better placed in pricing discussions by enabling them to predict the price of a house based on its characteristics.

With 79 explanatory variables describing (almost) every aspect of residential homes in Ames, Iowa, we as a team performed creative data and feature engineering techniques and ran advanced regression techniques to provide a solution that is currently ranked in the top 15% of all solutions on Kaggle.

We implemented 5 different machine learning algorithms and below is the performance of each them:

Algorithm	Pros	Cons	Cross-Validated RMSE Score	Kaggle Score
Multiple Linear Regression	Simple, easy to implement	Models only linear relationship, susceptible to outliers	0.12339	0.12947
Random Forest	Lower variance, decorrelates data, scale invariant	High bias, difficult to interpret	0.12889	0.14014
Lasso Regression	Easily interpretable, computationally inexpensive, enables feature selection	Drops grouped variables that are highly correlated	0.11374	0.12241
Ridge Regression	Easily interpretable, computationally inexpensive	Requires scaled variables and numeric variables	0.1004	0.12741
Ensemble Method	Can improve accuracy	Loss of interpretability	-	0.12246

The dataset contained variables that have a good amount of multicollinearity. As expected, Lasso Regression dropped a substantial number of variables from the model, performing the best with a cross validated RMSE of 0.11374.

The ensemble model performed as well as the lasso model, which is again expected, as it was largely driven by the predictions made by the lasso model as it has the highest weight of 0.7 assigned.

We built the models by getting a very good understanding of the data by performing detailed Exploratory Data Analysis (EDA) with different kinds of visualizations.

II- Data Description

1. Data Source Information

Source: Kaggle

Link: https://www.kaggle.com/c/house-prices-advanced-regression-techniques

2. Sample Information

Total variables: 79

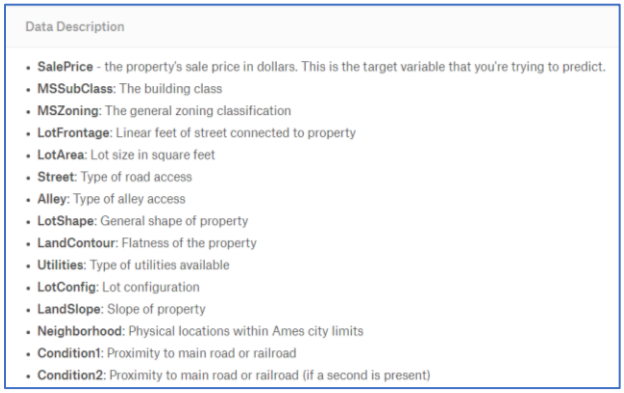
Sample Size: Train data set – 1460, Test data set - 1460

Sample Data:

	A	B	C	D	E	F	G	H	I	J	K
1	Id	MSSubCla	MSZoning	LotFronta	LotArea	Street	Alley	LotShape	LandCont	Utilities	LotConfig
2	1461	20	RH	80	11622	Pave	NA	Reg	Lvl	AllPub	Inside
3	1462	20	RL	81	14267	Pave	NA	IR1	Lvl	AllPub	Corner
4	1463	60	RL	74	13830	Pave	NA	IR1	Lvl	AllPub	Inside
5	1464	60	RL	78	9978	Pave	NA	IR1	Lvl	AllPub	Inside
6	1465	120	RL	43	5005	Pave	NA	IR1	HLS	AllPub	Inside
7	1466	60	RL	75	10000	Pave	NA	IR1	Lvl	AllPub	Corner
8	1467	20	RL	NA	7980	Pave	NA	IR1	Lvl	AllPub	Inside
9	1468	60	RL	63	8402	Pave	NA	IR1	Lvl	AllPub	Inside
10	1469	20	RL	85	10176	Pave	NA	Reg	Lvl	AllPub	Inside

[See Variable Table in Appendix]

3. Variable Type Information



Data Description

- **SalePrice** - the property's sale price in dollars. This is the target variable that you're trying to predict.
- **MSSubClass**: The building class
- **MSZoning**: The general zoning classification
- **LotFrontage**: Linear feet of street connected to property
- **LotArea**: Lot size in square feet
- **Street**: Type of road access
- **Alley**: Type of alley access
- **LotShape**: General shape of property
- **LandContour**: Flatness of the property
- **Utilities**: Type of utilities available
- **LotConfig**: Lot configuration
- **LandSlope**: Slope of property
- **Neighborhood**: Physical locations within Ames city limits
- **Condition1**: Proximity to main road or railroad
- **Condition2**: Proximity to main road or railroad (if a second is present)

Data Description

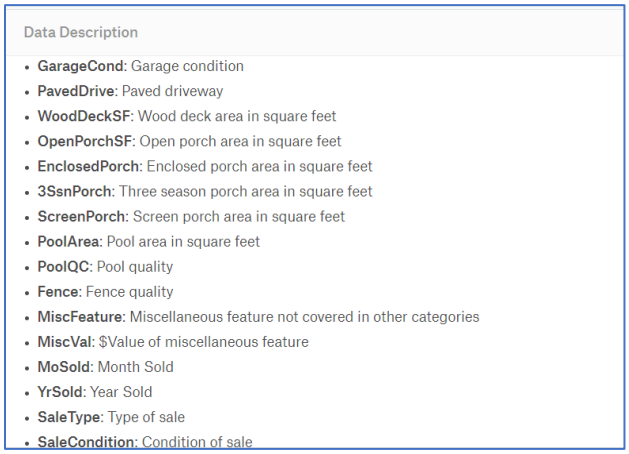
- **BldgType**: Type of dwelling
- **HouseStyle**: Style of dwelling
- **OverallQual**: Overall material and finish quality
- **OverallCond**: Overall condition rating
- **YearBuilt**: Original construction date
- **YearRemodAdd**: Remodel date
- **RoofStyle**: Type of roof
- **RoofMatl**: Roof material
- **Exterior1st**: Exterior covering on house
- **Exterior2nd**: Exterior covering on house (if more than one material)
- **MasVnrType**: Masonry veneer type
- **MasVnrArea**: Masonry veneer area in square feet
- **ExterQual**: Exterior material quality
- **ExterCond**: Present condition of the material on the exterior
- **Foundation**: Type of foundation
- **BsmtQual**: Height of the basement

Data Description

- **BsmtCond**: General condition of the basement
- **BsmtExposure**: Walkout or garden level basement walls
- **BsmtFinType1**: Quality of basement finished area
- **BsmtFinSF1**: Type 1 finished square feet
- **BsmtFinType2**: Quality of second finished area (if present)
- **BsmtFinSF2**: Type 2 finished square feet
- **BsmtUnfSF**: Unfinished square feet of basement area
- **TotalBsmtSF**: Total square feet of basement area
- **Heating**: Type of heating
- **HeatingQC**: Heating quality and condition
- **CentralAir**: Central air conditioning
- **Electrical**: Electrical system
- **1stFlrSF**: First Floor square feet
- **2ndFlrSF**: Second floor square feet
- **LowQualFinSF**: Low quality finished square feet (all floors)
- **GrLivArea**: Above grade (ground) living area square feet

Data Description

- **BsmtFullBath**: Basement full bathrooms
- **BsmtHalfBath**: Basement half bathrooms
- **FullBath**: Full bathrooms above grade
- **HalfBath**: Half baths above grade
- **Bedroom**: Number of bedrooms above basement level
- **Kitchen**: Number of kitchens
- **KitchenQual**: Kitchen quality
- **TotRmsAbvGrd**: Total rooms above grade (does not include bathrooms)
- **Functional**: Home functionality rating
- **Fireplaces**: Number of fireplaces
- **FireplaceQu**: Fireplace quality
- **GarageType**: Garage location
- **GarageYrBlt**: Year garage was built
- **GarageFinish**: Interior finish of the garage
- **GarageCars**: Size of garage in car capacity
- **GarageArea**: Size of garage in square feet



Data Description

- **GarageCond**: Garage condition
- **PavedDrive**: Paved driveway
- **WoodDeckSF**: Wood deck area in square feet
- **OpenPorchSF**: Open porch area in square feet
- **EnclosedPorch**: Enclosed porch area in square feet
- **3SsnPorch**: Three season porch area in square feet
- **ScreenPorch**: Screen porch area in square feet
- **PoolArea**: Pool area in square feet
- **PoolQC**: Pool quality
- **Fence**: Fence quality
- **MiscFeature**: Miscellaneous feature not covered in other categories
- **MiscVal**: $Value of miscellaneous feature
- **MoSold**: Month Sold
- **YrSold**: Year Sold
- **SaleType**: Type of sale
- **SaleCondition**: Condition of sale

The train dataset consisted of character and integer variables. Most of the character variables were ordinal factors. We chose to read them in R as character strings because most of those required cleaning and/or feature engineering. In total, there were 81 variables, of which the last one is the response variable (SalePrice).

III- Research Questions

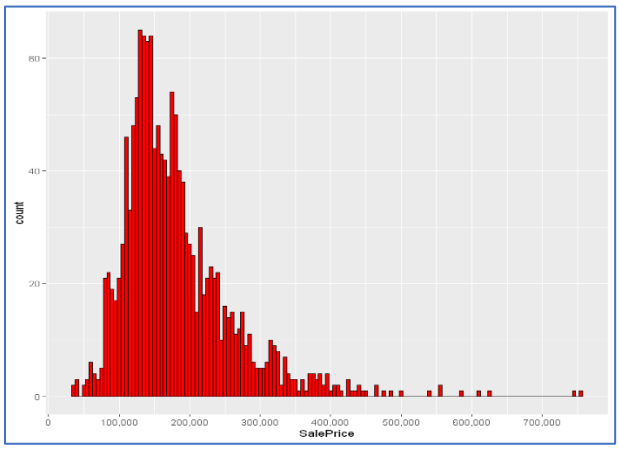
The following were the most important research questions for our study:

1. How can we predict sales prices of a house based on its features?
2. What are the most important features that impact house prices?
3. How can the variables be improved through feature engineering?

IV- Methodology

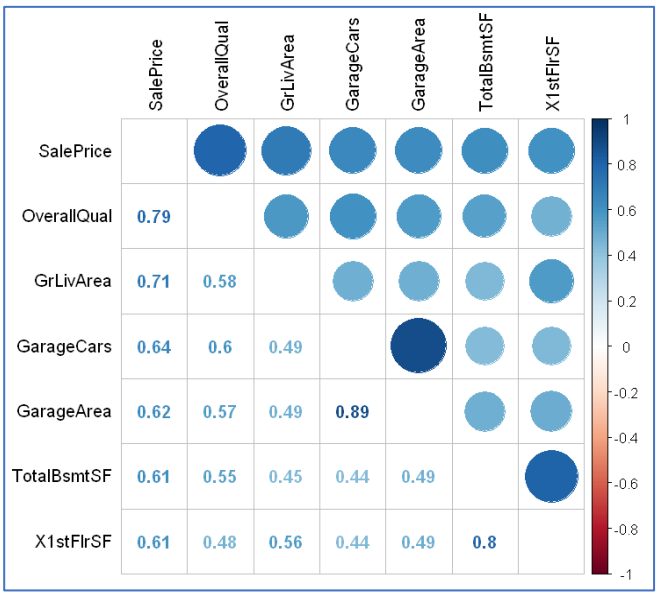
1. Variable Exploration

Figure 1: The response variable, Sales Price



The dependent variable SalePrice is right skewed because the data represents that fewer people can afford expensive houses. We kept this in mind before modeling.

Figure 2: Numeric Predictors



We looked at the most important numeric variables that had the highest correlation with the dependent variable, SalePrice.

SalePrice was highly correlated with OverallQual and GrLivArea variables that corresponded to the area above ground, 0.79 and 0.71 respectively.

There was a strong multicollinearity between GarageArea and GarageCars (which is the size of the garage

in terms of how many cars it can hold) with 0.89. Furthermore, there was a strong correlation between X1stFlSF (square footage on the 1st floor) and TotalBmstSF (Total Basement Square Footage) because basements typically have a very similar square footage to the 1st floor.

2. Missing Data and Factors/Ordinals

We observed that approximately 7% of the entire data available had NA's which was accounted for 34 out of the 79 columns available. We dealt with missing values starting with the columns with most missing values. Some values were easily imputed, whereas others required deeper data processing. Furthermore, depending on the variable, there were some character variables that were encoded as ordinal.

NOTE: These imputations were carried out after studying the data dictionary and the necessary documentation provided on the competition website.

The following activities were done as part of the data cleaning process:

1. Correlation analysis of variables with SalePrice was performed
2. Quantified NA's and missing data
3. Treated PoolQC for NA as 'No Pool'. The corresponding PoolArea was set to zero
4. Treated MiscFeature for NA as 'No Feature' and make it as a factor since there is no order
5. Treated Alley for NA as 'No Alley'. In this case, we weren't sure whether the variable was ordinal or not
6. Treated Fence for NA as 'No Fence' and converted Fence variable into a factor
7. Treated FireplaceQu and Fireplaces for NA as 'No Fireplace'. FireplaceQu was treated as an ordinal variable
8. NAs in LotFrontage were imputed by the median per neighborhood
9. Treated Garage Finish/quality/condition/type for NA as 'No Garage'
10. Fixed discrepancy in Houses 2127 and 2577 with respect to GarageType and GarageFinish
11. Treated 5 of the Basement variables for NA
12. Fixed discrepancy in House 2611 for Masonry variables
13. After treating all the NAs, we converted all the character variables to factors
14. Few of the numeric variables like YearSold, MonthSold and, MSSubClass were converted to factors
15. Correlation analysis performed to figure out a total of 8 variables with a correlation > 0.6
16. Ran Random Forest to understand the importance of various variables. Numeric variables have the most impact on predicting the house prices
17. Feature engineering on TotalSqFt and bathroom variables was performed

Please refer the attached Jupyter notebook for more information on each of these tasks.

3. Analytical Data Set Preparation

Finally, the following activities were performed for preparation of the ADS for modeling:

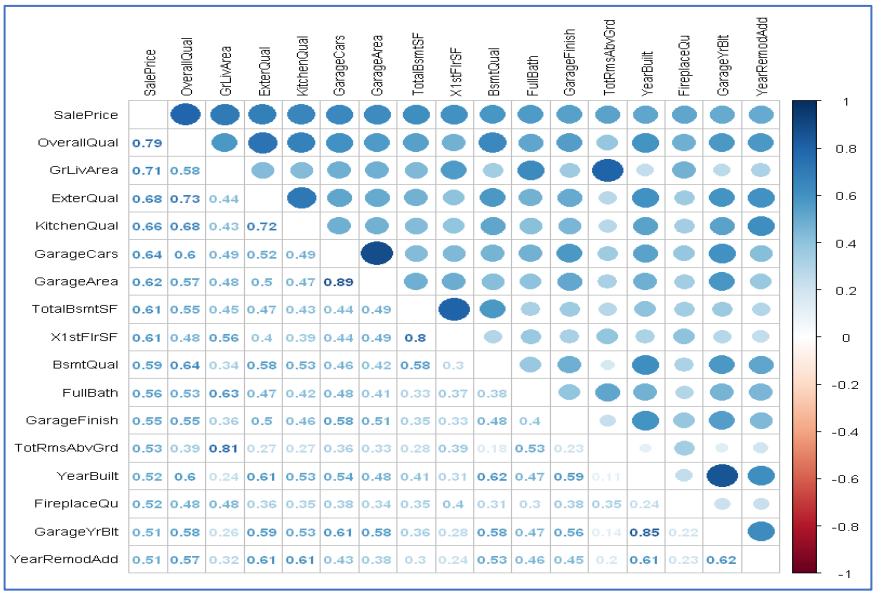
1. Removal of outliers
2. Dealing with multicollinearity
3. Standardization of variables
4. Encoding categorical variables
5. Merger of standardized and encoded variables

The exhaustive and manual approach that we took towards data cleaning and exploration helped us understand the different variables in depth. This helped us to reduce the number of iterations we took to come up with a good model.

V- Results and Findings

1. Data Exploration

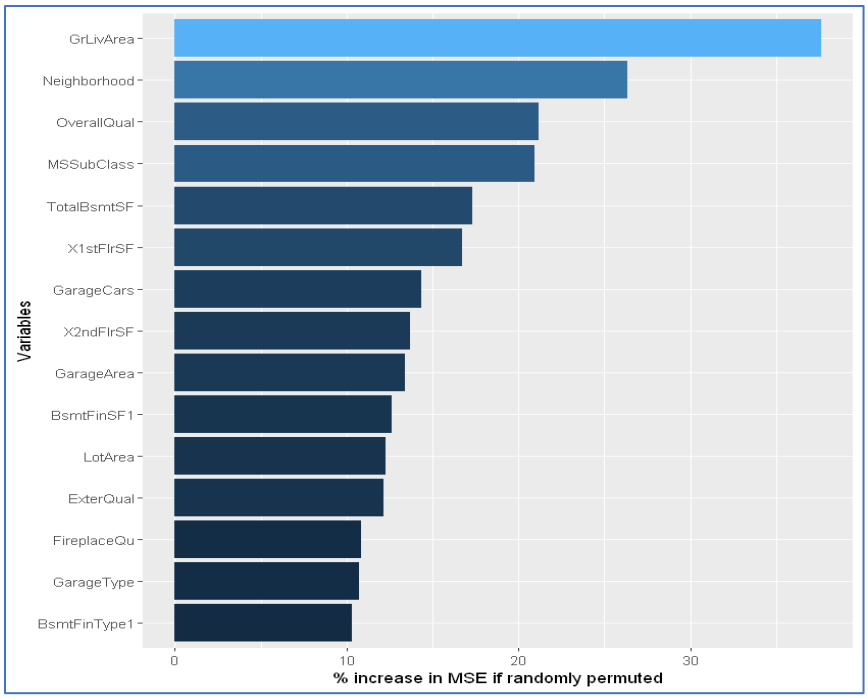
Figure 3: Most Important variables



Compared to Figure 2, we got 2 extra variables with a high correlation. A total of 8 numeric variables with a correlation > 0.6 with the dependent variable SalePrice.

OverallQual (Overall Quality) of the house and GrLivArea (General Living Area) remained the 2 most correlated variables with SalePrice.

Figure 4: Finding variable importance with a quick Random Forest



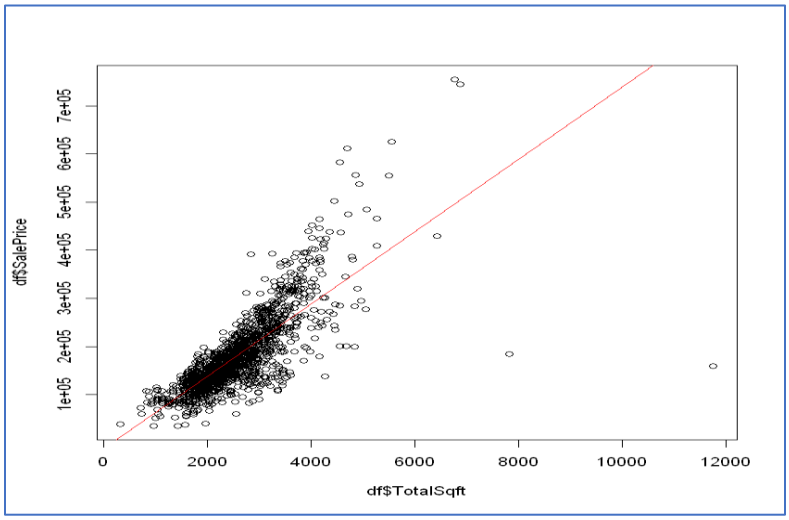
Although the correlations gave a good overview of the most important numeric variables and multicollinearity among those variables before moving on with modeling.

We considered the first 15 variables only, showing by how much the MSE would increase if the variables were randomly moved, i.e., how important the variables were in explaining the model. This was a rough Random Forest that showcased how the variables interact with each other. We flipped the axis for the ease of readability.

Out of the most important 15 variables, only 3 were categorical - Neighborhood, MSSubClass and GarageType, in this order. According to the results of Random Forest, the numeric variables were the most important in determining the sale price of a house. As observed previously GrLivArea (General Living Area) and OverallQual (Overall Quality) still had the biggest impact on house price predictions.

Figure 5: Relationship of engineered variables with SalePrice

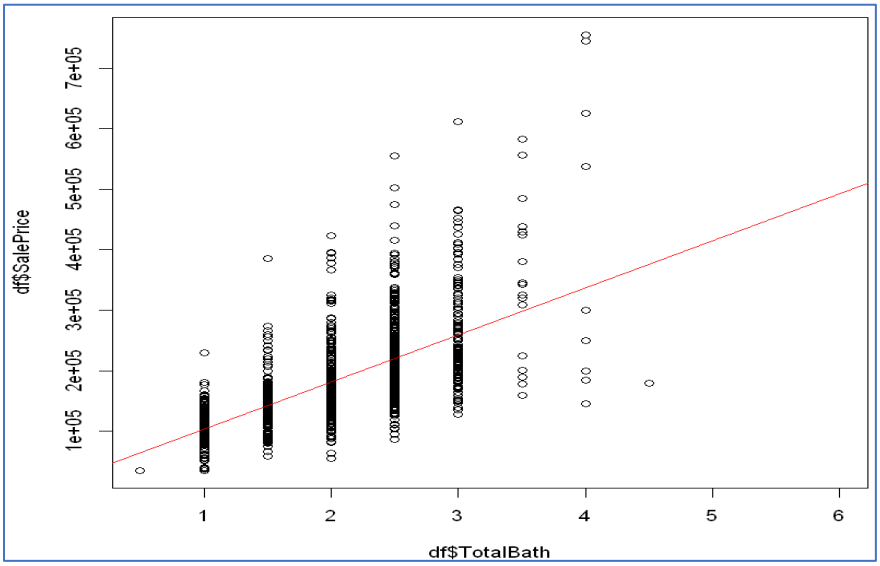
Total Square Footage: df$TotalSqft <- df$GrLivArea + df$TotalBsmtS



Since total living space is a very important point of consideration when people buy houses, we created a predictor that added up the living space above and below ground.

This new variable had a strong correlation of 0.78 with the SalePrice. Thus, we added it in the model.

Figure 6: Bathroom variables



There were four bathroom variables: FullBath (full bathroom), HalfBath (half bathroom), BsmtFullBath (basement full bathroom) and BsmtHalfBath (basement half bathroom). We added these, weighing based on where the bathroom is located. In this section, we weighed HalfBathroom the same as FullBathroom, and then with a weight of 0.5. The correlation was stronger when half bathrooms were weighted down. We also worked with the weight for full bathroom in the above and below ground living area. The strongest correlation appeared when the full bathroom in the basement was weighted down to 0.5. SalePrice and the new bathroom variable had a strong correlation of 0.65.

2. Data Modeling: Advanced Regression Techniques

Figure 7: Multiple Linear Regression with cross validation

```
set.seed(123)
lm_mod <- train(SalePrice~.,
                  data = train1,
                  method = "lm",
                  trControl=trainControl(
                    method = "cv",
                    number=5,
                    savePredictions = TRUE,
                    verboseIter = TRUE)
)
```

```
+ Fold1: intercept=TRUE
- Fold1: intercept=TRUE
+ Fold2: intercept=TRUE
- Fold2: intercept=TRUE
+ Fold3: intercept=TRUE
- Fold3: intercept=TRUE
+ Fold4: intercept=TRUE
- Fold4: intercept=TRUE
+ Fold5: intercept=TRUE
- Fold5: intercept=TRUE
Aggregating results
Fitting final model on full training set
```

We started off with a linear regression model with cross validation to set the baseline to compare the other models perform.

We execute a K-fold cross validated multiple linear regression model with k = 5 that gave an RMSE of 0.1233 and Adjusted R-square of 0.934.

Additionally, after studying the coefficients of the regression we observed that 1-unit change in standardized TotalSqFt (Total Square Feet) had a 10% impact on the SalePrice. Similarly, OverallQual (Overall Quality) had a 5% impact on the SalePrice.

Figure 8: Lasso Regression with cross validation

We used caret cross validation to find the best value of lambda, which is the only hyperparameter that needs to be tuned for the lasso model. The tuning parameter lambda controlled the overall strength of the penalty. It was known that the ridge penalty shrinks the coefficients of correlated predictors towards each other while the lasso tends to pick one of them and discard the others.

```
In [134]: set.seed(123)
          my_control <-trainControl(method="cv", number=5)
          lassoGrid <- expand.grid(alpha = 1, lambda = seq(0.001,0.1,by = 0.0005))

          lasso_mod <- train(x=train2, y=dfmodel$SalePrice[!is.na(dfmodel$SalePrice)],
                             method='glmnet', trControl= my_control, tuneGrid=lassoGrid)
          lasso_mod$bestTune
```

	alpha	lambda
4	1	0.0025

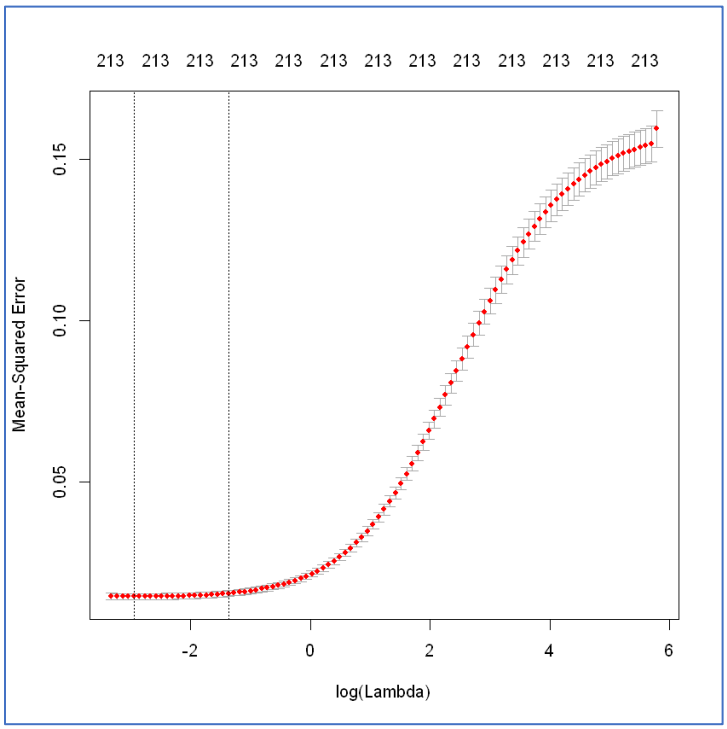
```
In [135]: min(lasso_mod$results$RMSE)
          min(lasso_mod$results$Rsquared)
          min(lasso_mod$results$MAE)
```

```
0.113738283115503

0.830879200867707

0.0796113087326443
```

The Lasso Regression model did a pretty good job dealing with multicollinearity by using only 104 out of 213 variables available to it and resulted in an RMSE value of 0.11374.

Figure 9: Ridge Regression by K-Fold cross validation

Ridge regression performed shrinkage without exclusion of predictors. We first created a matrix of all predictors and a vector of the response before we passed it into the glmner function.



The best lambda value that resulted in the smallest cross validation error was 0.0522611783512238. The RMSE value associated with this lambda value was 0.1004.

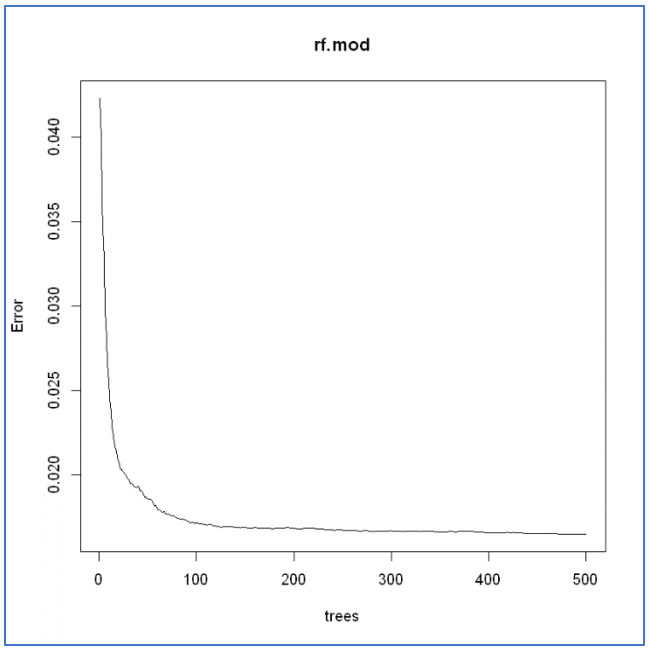
Figure 10: Random Forest

Random forests were built on the same fundamental principles as decision trees and bagging. Since the algorithm selected a random bootstrap sample to train, predictors to use at each split and, tree correlation was lessened beyond bagged trees.

```
In [158]: rf.mod <- randomForest(
            formula = SalePrice~.,
            data    = train1)

          rf.mod


          Call:
           randomForest(formula = SalePrice ~ ., data = train1)
                         Type of random forest: regression
                               Number of trees: 500
          No. of variables tried at each split: 71

                    Mean of squared residuals: 0.01649832
                              % Var explained: 89.67
```

We started with a basic model before trying to tune the different parameters available.

Plotting the model illustrated the error rate as we averaged across more trees and showed that the error rate stabilized with approximately 100 trees but, continued to decrease slowly until approximately 300 trees.



Random forests were easy to tune since there were only a handful of tuning parameters. The primary concern when starting out was to tune the number of candidate variables to select from at each split. However, there were a few additional hyperparameters that we tuned with the help of a grid.

The best random forest model was found with mtry value set to 70, terminal node size set to 5 observations and, sample size set to 70%. The resulting RMSE was the best across all the models, with a value of 0.12873.

Figure 11: Ensemble Method

Ridge regression, lasso regression and, linear regression models were combined using ensemble method with the highest weight of 0.7 assigned to lasso regression. On the test data set, the ensemble model scored a 0.1224 on Kaggle that put the model in the top 15% of the total submissions till date.

VI- Conclusion

In conclusion, the Lasso Regression model with cross validation performed the best to predict the sales prices of houses. In the process of designing the best model, we learnt about the most 15 most important features that have the maximum impact in determining the house prices. This was a crucial finding in terms of the business value that these features bring to the model that we have built. The results of the extensive data cleaning and feature engineering we carried out helped us come up with 2 of the most important features that were implicit in the data set. Additionally, we have been able to identify which variables that are most important while predicting the Sales Price like Total Square Feet and Overall Quality of the Home using Random Forests and then estimating their impact using the coefficients generated in our Multiple Linear Regression Model.

Finally, this project work was a great learning experience. Following were the key takeaways from this project:

- Data cleaning is the most important characteristic of data mining.
- The scope of data mining is limited by the number of records available for model building.
- Accuracy is not always the most significant aspect of data mining. Instead, understanding the most impact of variables helps solve the business problem at hand.

VII- Appendix

1. Jupyter Notebook

Capstone - House Price Prediction v0.2.h

2. Variable Type Information Table

Variable Name	Variable Type
Id	Numerical
MSSubClass	Categorical
MSZoning	Categorical
LotFrontage	Numerical
LotArea	Numerical
Street	Categorical
Alley	Categorical
LotShape	Categorical
LandContour	Categorical
Utilities	Categorical
LotConfig	Categorical
LandSlope	Categorical
Neighborhood	Categorical
Condition1	Categorical
Condition2	Categorical
BldgType	Categorical
HouseStyle	Categorical
OverallQual	Ordinal
OverallCond	Ordinal
YearBuilt	Numerical
YearRemodAdd	Numerical
RoofStyle	Categorical
RoofMatl	Categorical
Exterior1st	Categorical
Exterior2nd	Categorical
MasVnrType	Categorical
MasVnrArea	Numerical
ExterQual	Ordinal
ExterCond	Ordinal
Foundation	Categorical
BsmtQual	Ordinal
BsmtCond	Ordinal

BsmtExposure	Ordinal
BsmtFinType1	Ordinal
BsmtFinSF1	Numerical
BsmtFinType2	Ordinal
BsmtFinSF2	Numerical
BsmtUnfSF	Numerical
TotalBsmtSF	Numerical
Heating	Categorical
HeatingQC	Ordinal
CentralAir	Categorical
Electrical	Categorical
1stFlrSF	Numerical
2ndFlrSF	Numerical
LowQualFinSF	Numerical
GrLivArea	Numerical
BsmtFullBath	Numerical
BsmtHalfBath	Numerical
FullBath	Numerical
HalfBath	Numerical
BedroomAbvGr	Numerical
KitchenAbvGr	Numerical
KitchenQual	Ordinal
TotRmsAbvGrd	Numerical
Functional	Categorical
Fireplaces	Numerical
FireplaceQu	Ordinal
GarageType	Categorical
GarageYrBlt	Numerical
GarageFinish	Categorical
GarageCars	Numerical
GarageArea	Numerical
GarageQual	Ordinal
GarageCond	Ordinal
PavedDrive	Categorical
WoodDeckSF	Numerical
OpenPorchSF	Numerical
EnclosedPorch	Numerical
3SsnPorch	Numerical
ScreenPorch	Numerical
PoolArea	Numerical
PoolQC	Ordinal
Fence	Ordinal
MiscFeature	Categorical

MiscVal	Numerical
MoSold	Categorical
YrSold	Numerical
SaleType	Categorical
SaleCondition	Categorical
SalePrice	Numerical