Prediction of House Prices: Exploratory and Predictive Analysis

Team Members	Contribution
Akshat Verma	20%
Amay Chopra	20%
Meghal Gupta	20%
Satya Akhil Reddy Bachugudam	20%
Shilpa Subramanian	20%

Table of Contents

Introduction ... ***3***

Importance of the Problem ... 3

Objective ... 3

Scope ... 3

Project Approach ... ***4***

Data description ... 4

Exploratory Data Analysis ... 4

Objective Flowchart ... 9

Reason for the chosen approach ... 10

Implementation Details ... ***10***

Regression ... **10**

1. Multiple Linear Regression ... 10
2. Shrinkage Methods ... 12

Tree Based Methods ... **14**

1. Bagging ... 14
2. Random Forest ... 15
3. Boosting ... 16

Comparison ... ***17***

Overall Comparison ... 18

Executive Summary ... ***18***

Importance in the Real World ... 18

Gap in Literature ... 18

Implication of the Result ... 19

Conclusions ... ***19***

Future Scope ... 20

References ... ***21***

Appendix ... ***22***

Introduction

Importance of the Problem

In the highly competitive real estate industry, which boasts a monopoly of large brokerage firms, the small players need a considerable competitive advantage to stay relevant. In fact, individual sellers often fail to put a price on their property and most buyers struggle to shortlist a house based on a favorable trade-off between the characteristics that they look for in an ideal home and the price they are willing to pay. Even for a given location and time of the year, predicting trends in house prices is tough at the best.

There are a number of parameters that may impact the final sale price of a house. With a dataset that explores about 80 probable variables that may impact the final negotiable price of a house in the city of Ames, we attempt to shortlist the most determining variables. Once these important predictors are shortlisted, we would use these to create a model that can successfully predict the final sale price of a house.

This predictive model would prove to be useful for both buyers and sellers in arriving at a prediction of the most suitable price for the property under consideration.

Objective

We would employ various data analysis techniques to predict the selling price of residential homes in Ames, Iowa. This can prove to be a good tool for sellers as they can analyze the prices of homes that they already own and for buyers who wish to purchase a new home. Currently, there are many options available to a buyer in the city and it is difficult to analyze the prices of homes. It is a cumbersome task to study every house on sale. Through this project, our aim is to predict the sales price of residential homes by considering the most important aspects (features) and determine the role they play in our decision.

Scope

All the methods employed in this project implement regression techniques in data analysis studied throughout the semester of this course to help in predicting sales price of residential homes which is not available easily.

The techniques that we use are different regression methods such as multiple linear regression, shrinkage methods including ridge regression and LASSO and regression trees including bagging, random forest and boosting. These regression models were evaluated on the basis of the training data.

Project Approach

Data description

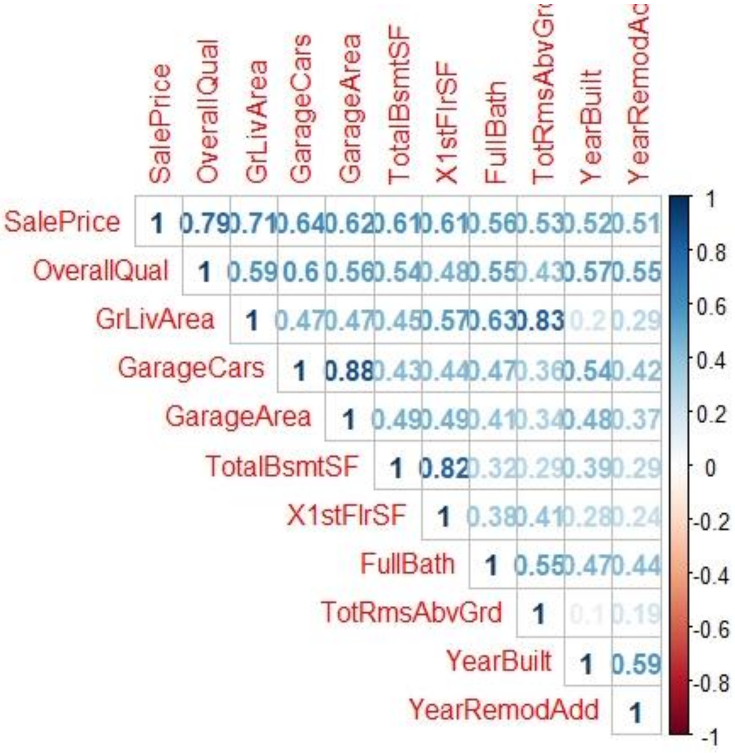
The dataset consists of character and integer variables. There are 1460 observations with 80 explanatory variables describing almost every aspect of residential homes in Ames, Iowa. There are 37 integer variables, and 43 character variables. We will use the subsets of this dataset according to our needs for descriptive analysis or quantitative analysis.

The main challenges in processing this data set is the limited data with respect to all the predictors of residential homes, missing feature values and feature values that belong to variable types.

Exploratory Data Analysis

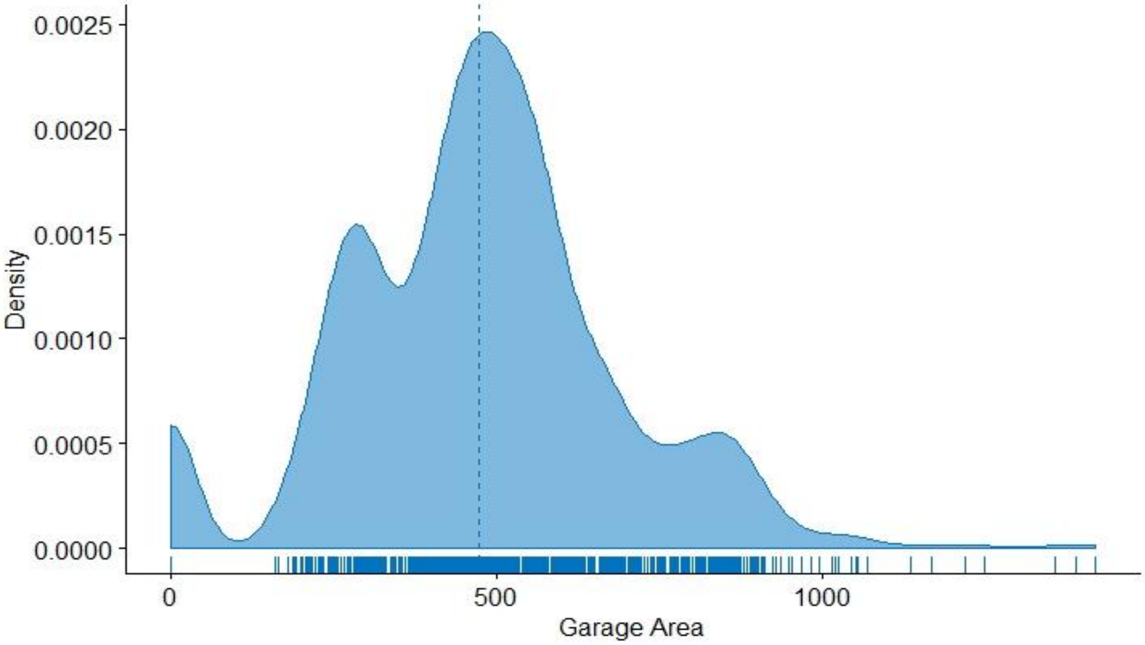
Here we study correlations of various predictors in our data which help us to further explore our understanding of how various attributes are linked.

We try to find the correlation between the predictors involved in our analysis. We only need predictors which have a strong correlation with sale price. A threshold of 0.5 was assumed and the important predictors were factored out. The below correlation plot shows the important predictors from our analysis arranged in a decreasing order.

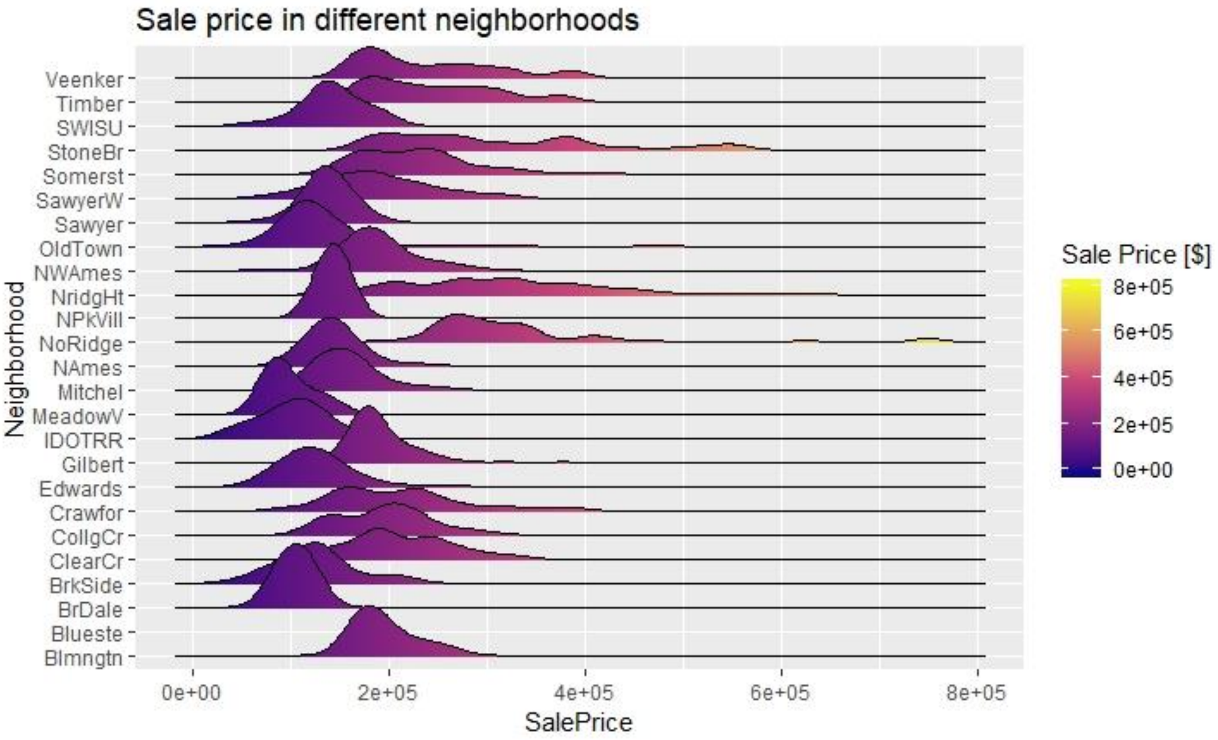


	SalePrice	OverallQual	GrLivArea	GarageCars	GarageArea	TotalBsmtSF	X1stFlrSF	FullBath	TotRmsAbvGr	YearBuilt	YearRemodAc
SalePrice	1	0.79	0.71	0.64	0.62	0.61	0.61	0.56	0.53	0.52	0.51
OverallQual		1	0.59	0.6	0.56	0.54	0.48	0.55	0.43	0.57	0.55
GrLivArea			1	0.47	0.47	0.45	0.57	0.63	0.83	0.2	0.29
GarageCars				1	0.88	0.43	0.44	0.47	0.36	0.54	0.42
GarageArea					1	0.49	0.49	0.41	0.34	0.48	0.37
TotalBsmtSF						1	0.82	0.32	0.29	0.39	0.29
X1stFlrSF							1	0.38	0.41	0.28	0.24
FullBath								1	0.55	0.47	0.44
TotRmsAbvGrd									1	0.1	0.19
YearBuilt										1	0.59
YearRemodAdd											1

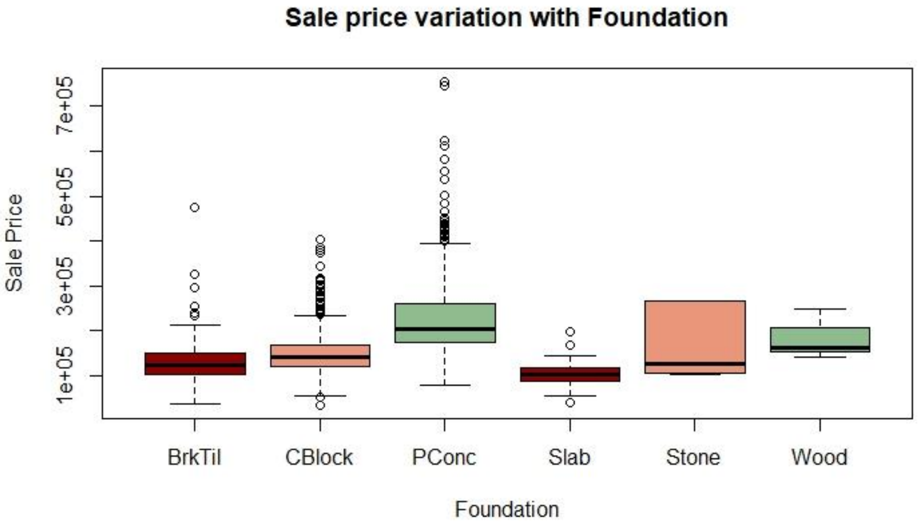
Description: We find that overall quality of the house has the highest correlation with sale price. Then, predictors like ground living area, garage area etc. come into the picture. This can be observed as the starting point in our attempt to separate out the important variables for fitting the models. Let us take a look at the variation of the first 3-4 highly correlated variables with sale price.



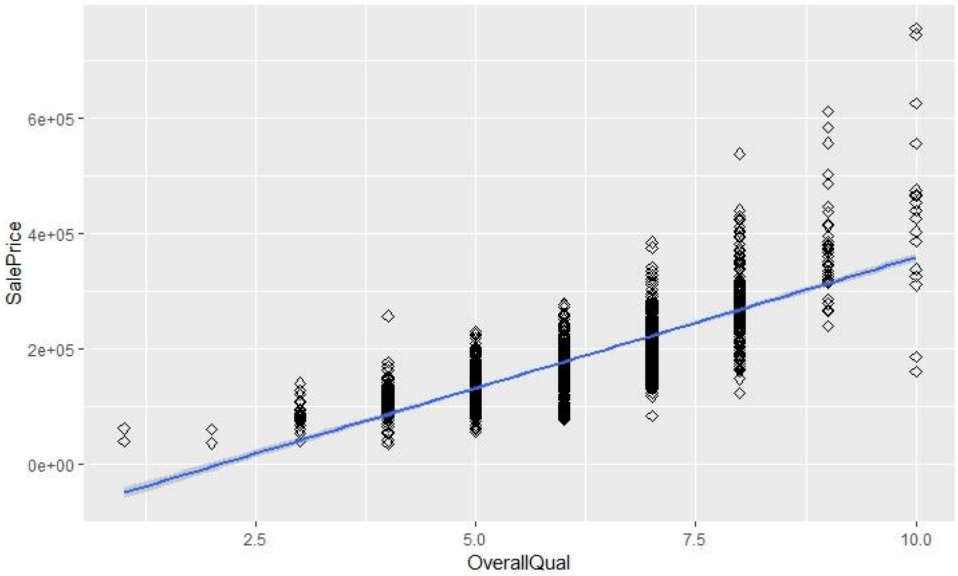
Description: When buying a house, people prefer to buy the one which provides optimal garage space. In Ames, the houses for sale have an average garage area of 500 sq. ft. We can make certain observations from this graph. First, the houses with a garage area around the mean maybe centered in urban areas, where the availability of land is very less. Since such houses are present in an urban area, their selling price may be on the higher side. But from our previous analysis, we saw that most of the neighborhoods have almost the same selling price. Thus, from these observations, we can say that garage area alone doesn’t seem fit in making an initial guess on the sale price. We need to combine it with other parameters to better understand its importance.



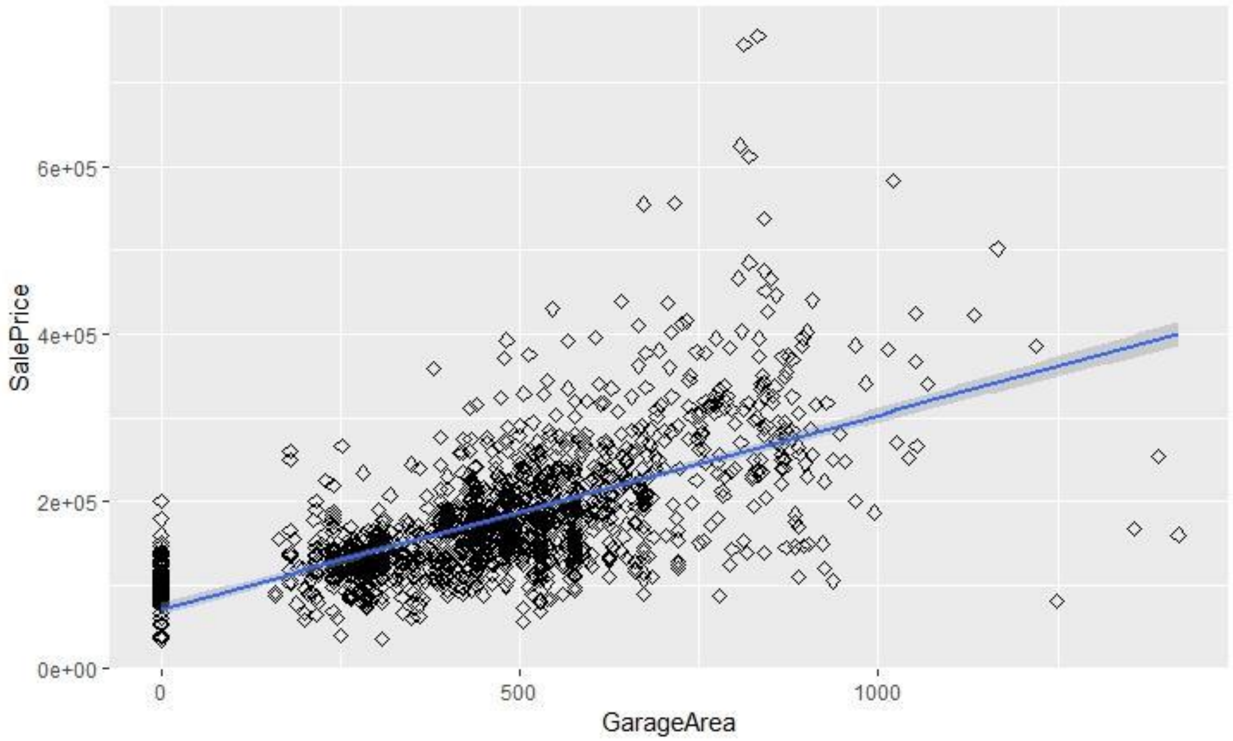
Description: In any city, we have neighborhoods with different standards of living. The standard of living in turn will reflect on the sale price of the houses in that neighborhood. From the plot between Sale Price and Neighborhood, we find that most of the localities have a similar sale price when other factors are considered constant. Among them, only Northridge and Northridge Heights standout. The mean sale price in these localities is almost 50% greater when compared to other neighborhoods. Also, the sale price has a higher spread in Stone Brook, Northridge and Northridge Heights relatively.



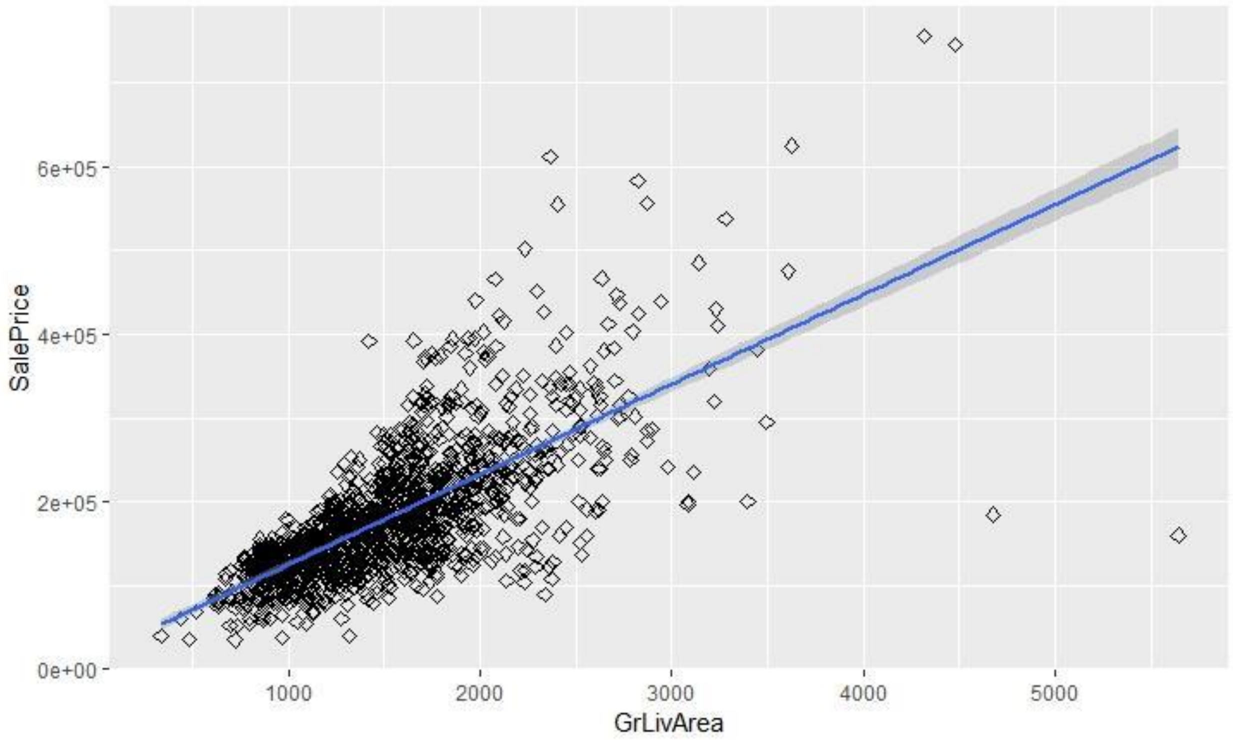
Description: Foundation is one of the main indicators which buyers consider when trying to buy a house. From the plot between Sale Price and Foundation in Ames, we find that there is no clear distinction based on foundation, which is in contrast to generic assumption. Though poured concrete type foundation has a higher sale value, the pairwise difference with other types of foundation is very small. In such a case, we can assume the effect of foundation to be negligible on sale price.



Description: The above plot shows the variation between sale price and overall quantity. It is obvious that, higher the quality, higher is the sale price. But the effect of sale price is limited by the influence of other predictors. For instance, the sale price for any quality of the house is limited by the kind of neighborhood the house is located in.

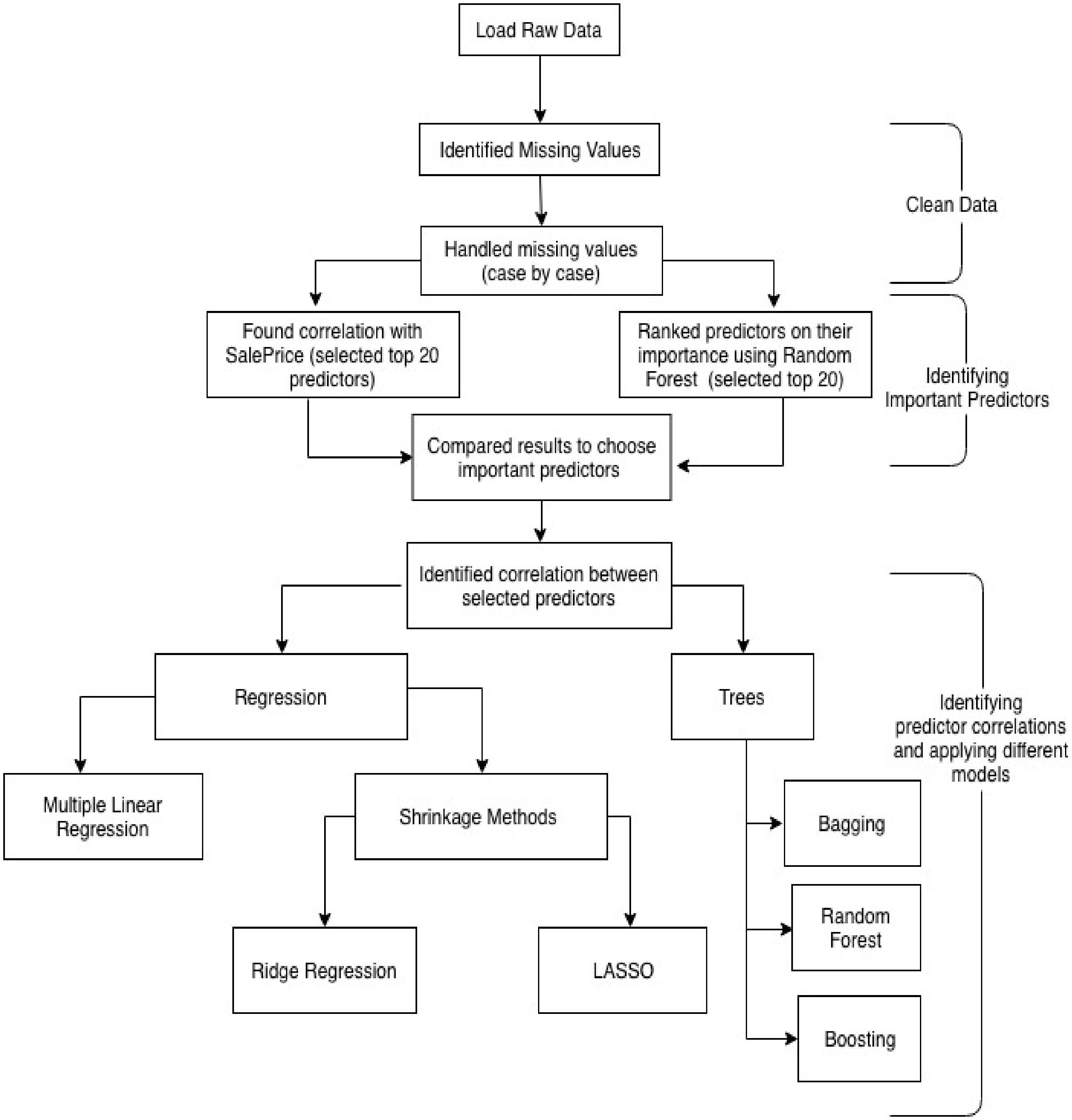


Description: From the above plot, we find that there is a linear effect of garage area on the sale price. But we find that houses with a low garage area also have a higher selling price. Not to be repetitive, this point was already discussed in one of the observations of density plot of garage area.



Description: The above plot shows the effect of ground living area on the sale price. The sale price increases almost linearly with the ground living area. We can discuss the influence of certain predictors on this association as well. A house with large living area but with less overall quality will tend to be in the market at a lower selling price. Thus, such nuances have to be accounted for, which is why different models will be developed in the coming sections to address the shortcomings in visual observations.

Objective Flowchart


Load Raw Data
Identified Missing Values
Handled missing values (case by case)
Clean Data
Found correlation with SalePrice (selected top 20 predictors)
Ranked predictors on their importance using Random Forest (selected top 20)
Identifying Important Predictors
Compared results to choose important predictors
Identified correlation between selected predictors
Regression
Trees
Identifying predictor correlations and applying different models
Multiple Linear Regression
Shrinkage Methods
Bagging
Random Forest
Ridge Regression
LASSO
Boosting

Reason for the chosen approach

There is no single statistical model which would promise to give consistent accuracy or prediction performance across a varied set of real world problems. The modelling methods applied to any specific problem depends a great deal on the nature of the problem at hand - classification or regression. Even after this is known, of all the available techniques for classification and regression, arriving at a one with the least prediction error is possible only by performing all possible modelling activities and then comparing the results to identify the best possible method.

Since the objective of our project is to predict the sale price of houses, which is a numeric quantity, we eliminated the classification methods from the list of possible approaches to achieve the objective at hand. Following this, of all the available regression methods we chose to perform Multiple Linear Regression, Ridge, LASSO, Bagging, Random Forest and Boosting on the features that we selected based on their importance ranking obtained using Random Forest. To visualize the results, methods like ggplot were used. So, for arriving at our prediction of the most suitable sale price for houses, we have utilized various approaches and then used cross-validation in an attempt to identify the best approach.

Implementation Details

Regression

1. Multiple Linear Regression

Running multiple linear regression is a good starting point after identifying the 20 most important predictors.

```
mlr.housing_data <- lm(SalePrice ~., data = final_data)
```

The summary for the multiple linear regression fit is:

```
        Call:
lm(formula      =      SalePrice      ~      .,      data      =      final_data)

Residuals:
    Min                  1Q      Median                   3Q                 Max
-4.1396       -0.1314             0.0027             0.1132              2.2525

Coefficients:      (1      not      defined      because      of      singularities)
                      Estimate      Std.      Error      t      value      Pr(>|t|)
(Intercept)                        -1.276e+00      1.437e+00      -0.888   0.374714
```

```
NeighborhoodBlueste     -7.925e-02          2.409e-01           -0.329     0.742196
NeighborhoodBrDale         -9.631e-02         1.256e-01         -0.767     0.443336
NeighborhoodBrkSide     -4.748e-02          1.059e-01           -0.448     0.653988
NeighborhoodClearCr        1.124e-01        1.037e-01             1.085    0.278288
NeighborhoodCollgCr     -2.852e-02          8.478e-02           -0.336     0.736658
NeighborhoodCrawfor        1.598e-01        9.934e-02             1.609    0.107853
NeighborhoodEdwards    -2.221e-01         9.329e-02        -2.380     0.017440    *
NeighborhoodGilbert     -4.251e-02          8.936e-02           -0.476     0.634388
NeighborhoodIDOTRR         -1.262e-01         1.218e-01         -1.036     0.300320
NeighborhoodMeadowV     -1.005e-01          1.226e-01           -0.819     0.412715
NeighborhoodMitchel     -1.486e-01          9.542e-02           -1.557     0.119708
NeighborhoodNAmes            -1.381e-01        8.964e-02         -1.540    0.123766
NeighborhoodNoRidge       5.416e-01      9.771e-02          5.543   3.55e-08   ***
NeighborhoodNPkVill     -1.186e-01          1.348e-01           -0.880     0.379032
NeighborhoodNridgHt       2.447e-01       8.892e-02          2.752    0.006000   **
NeighborhoodNWAmes         -1.463e-01         9.177e-02         -1.594     0.111092
NeighborhoodOldTown     -1.633e-01          1.085e-01           -1.505     0.132517
NeighborhoodSawyer         -1.156e-01         9.435e-02         -1.225     0.220744
NeighborhoodSawyerW     -7.373e-02          9.096e-02           -0.811     0.417769
NeighborhoodSomerst        8.838e-02        1.055e-01             0.838    0.402419
NeighborhoodStoneBr       4.790e-01      1.007e-01          4.758   2.16e-06    ***
NeighborhoodSWISU           -2.145e-01       1.138e-01       -1.886    0.059529    .
NeighborhoodTimber           3.280e-02        9.553e-02           0.343    0.731349
NeighborhoodVeenker       3.000e-01       1.241e-01           2.418   0.015715    *
GrLivArea                        4.620e-04    5.162e-05      8.951    <  2e-16  ***
X2ndFlrSF                            -2.520e-05     5.217e-05     -0.483   0.629220
OverallQual                     1.065e-01    1.200e-02      8.874     <  2e-16  ***
TotalBsmtSF                         -4.620e-06     4.985e-05      -0.093   0.926173
BsmtFinSF1                         5.430e-05    3.171e-05       1.712   0.087063  .
YearBuilt                            9.460e-04     7.137e-04       1.326   0.185216
BsmtFinType1BLQ                 -2.923e-02       3.454e-02       -0.846    0.397520
BsmtFinType1GLQ                  7.805e-03       3.098e-02         0.252   0.801116
BsmtFinType1LwQ               -1.057e-01       4.409e-02      -2.398   0.016607   *
BsmtFinType1None            -5.585e-01       9.173e-02      -6.088   1.47e-09   ***
BsmtFinType1Rec               -6.811e-02       3.611e-02      -1.886   0.059437   .
BsmtFinType1Unf              -1.297e-01      3.425e-02      -3.787   0.000159   ***
GarageArea                        3.767e-04    6.532e-05      5.768  9.88e-09  ***
MSZoningFV                         2.813e-01    1.463e-01       1.923   0.054736  .
MSZoningRH                         2.594e-01    1.468e-01       1.767   0.077445  .
MSZoningRL                         3.029e-01    1.221e-01       2.481   0.013207  *
MSZoningRM                         2.565e-01    1.149e-01       2.232   0.025789  *
KitchenQualFa                 -3.429e-01      7.427e-02     -4.617  4.26e-06   ***
KitchenQualGd                 -2.469e-01      4.426e-02     -5.578  2.91e-08   ***
KitchenQualTA                 -3.093e-01      4.897e-02     -6.316  3.60e-10   ***
ExterQualFa                       -2.093e-01     1.164e-01     -1.799  0.072273   .
ExterQualGd                     -2.276e-01     5.864e-02     -3.881  0.000109   ***
ExterQualTA                     -2.352e-01     6.536e-02     -3.599  0.000331   ***
BsmtQualFa                       -3.704e-01     7.681e-02    -4.822   1.58e-06  ***
BsmtQualGd                       -3.064e-01     4.194e-02    -7.305   4.65e-13  ***
```

```
BsmtQualNone                  NA          NA        NA         NA
BsmtQualTA            -3.374e-01   5.056e-02    -6.673   3.61e-11 ***
FireplaceQuFa         -1.214e-01   8.934e-02    -1.359   0.174243
FireplaceQuGd         -8.346e-02   6.885e-02    -1.212   0.225626
FireplaceQuNone       -1.837e-02   8.039e-02    -0.228   0.819302
FireplaceQuPo         -1.126e-01   1.002e-01    -1.124   0.261046
FireplaceQuTA         -1.173e-01   7.158e-02    -1.638   0.101624
GarageTypeAttchd       2.638e-01   1.356e-01     1.946   0.051867 .
GarageTypeBasment      2.131e-01   1.551e-01     1.374   0.169532
GarageTypeBuiltIn      3.127e-01   1.413e-01     2.214   0.027001 *
GarageTypeCarPort      1.602e-01   1.720e-01     0.932   0.351742
GarageTypeDetchd       2.309e-01   1.350e-01     1.711   0.087374 .
GarageTypeNone         3.201e-01   1.463e-01     2.188   0.028804 *
Fireplaces             1.253e-01   3.212e-02     3.900   0.000101 ***
FullBath               6.244e-02   2.390e-02     2.612   0.009090 **
MSSubClass            -1.899e-03   2.645e-04    -7.178   1.15e-12 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3104 on 1395 degrees of freedom
Multiple R-squared:  0.854,   Adjusted R-squared:  0.8473
F-statistic: 127.5 on 64 and 1395 DF,  p-value: < 2.2e-16
```

The test MSE obtained using this fit is **0.8540823**

2. Shrinkage Methods

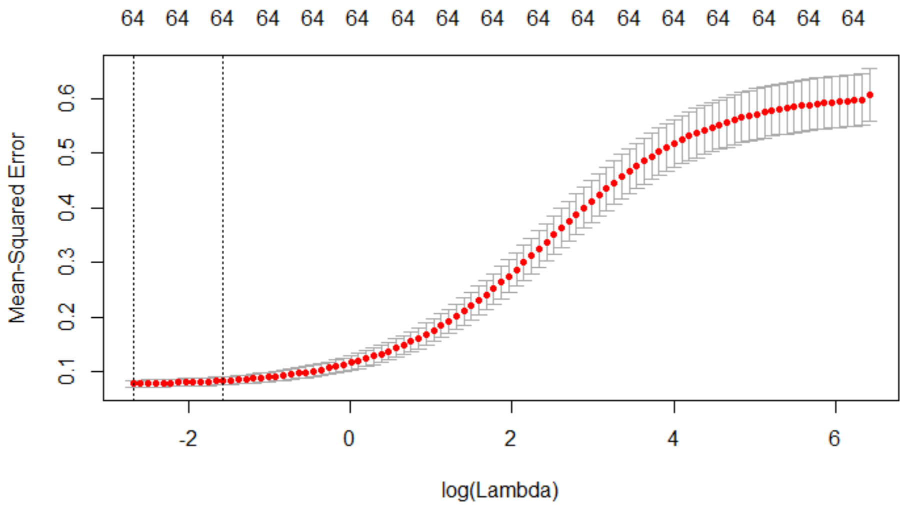
a. The Ridge

We used cross validation to find the best shrinkage parameter (lambda), of which we have obtained a value of 0.06766699.

```
grid=10^seq(10,-2,length=100)
ridge.house_data=glmnet(X[train,],y[train],alpha=0,lambda=grid)
```

Next, we have predicted response values (SalePrice) on the test data using the best lambda value.

```
yhat_ridge=predict(ridge.house_data,s=cv_ridge$lambda.min,newx=X[
-train,])
```



Lastly, we have computed the test MSE i.e. **0.1483397** and plotted the MSE vs log(lambda) graph.

b. LASSO

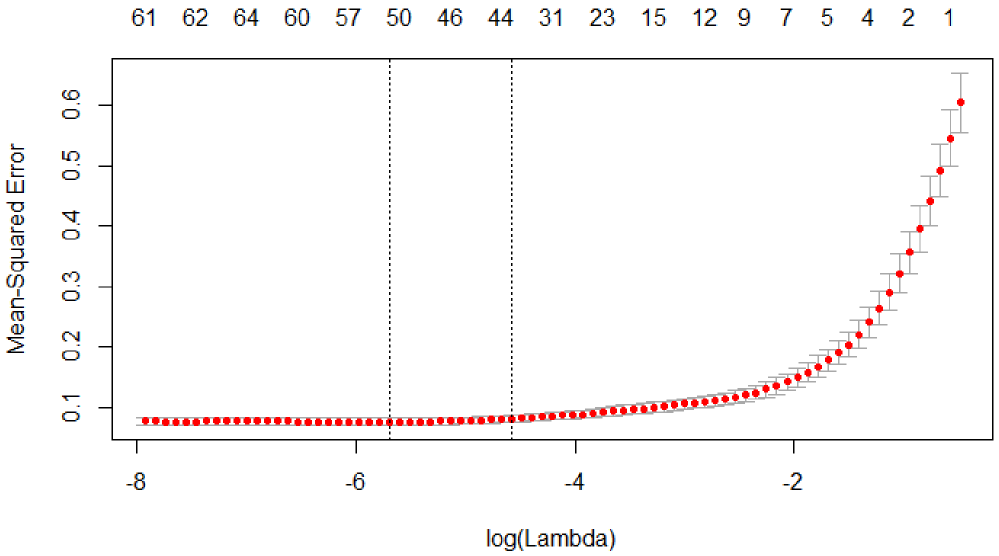
Again, like the Ridge Regression, we use cross validation to find the best shrinkage parameter, of which we obtained a value of 0.003367802.

```
cv_lasso=cv.glmnet(X[train,],y[train],alpha=1)
```

Next, we have predicted response values (SalePrice) on the test data using the best lambda value.

```
yhat_lasso=predict(lasso.house_data,s=cv_lasso$lambda.min,newx=X[
-train,])
```

Lastly, we have computed the test MSE i.e. **0.1547769** and plotted the MSE vs log(lambda) graph.



Tree Based Methods

1. Bagging

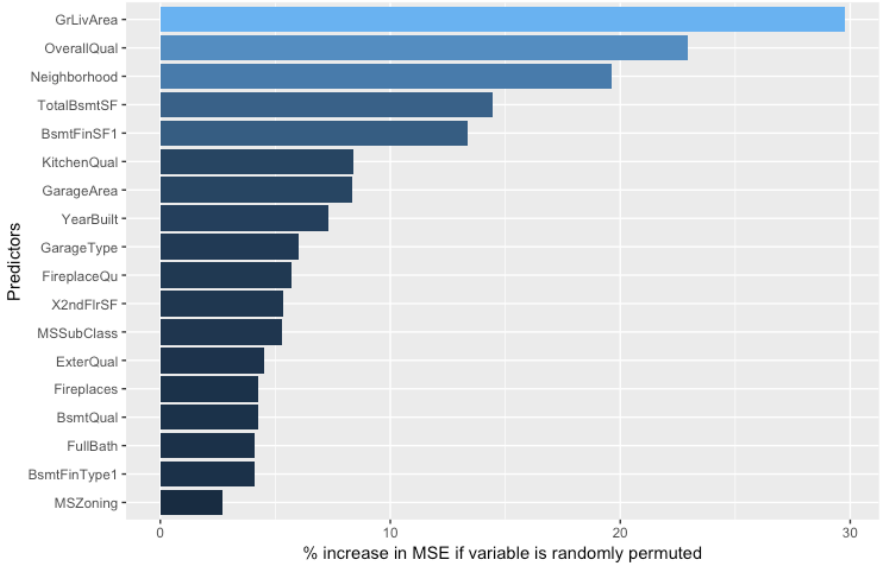
After dividing our data into training and test data, using mtry = 18 and number of trees = 100, we applied bagging on our training data.

```
bag.house_data <- randomForest(x=house_data.train[,-19],
y=house_data.train$SalePrice, ntree=100,importance=TRUE, mtry = 18)
```

Also, we created a plot for important variables found using Bagging.

```
imp.bag <- importance(bag.house_data)
imp.DataFrame <- data.frame(Variables = row.names(imp.bag), MSE =
imp.bag[,1])
imp.DataFrame <- imp.DataFrame[order(imp.DataFrame$MSE, decreasing =
TRUE),]

ggplot(imp.DataFrame, aes(x=reorder(Variables, MSE), y=MSE,
fill=MSE),col='red') + geom_bar(stat = 'identity') + labs(x =
'Predictors', y= '% increase in MSE if variable is randomly permuted')
+ coord_flip() + theme(legend.position="none")
```



Finally, we predicted the MSE on test data which came out as **0.1268441**

2. Random Forest

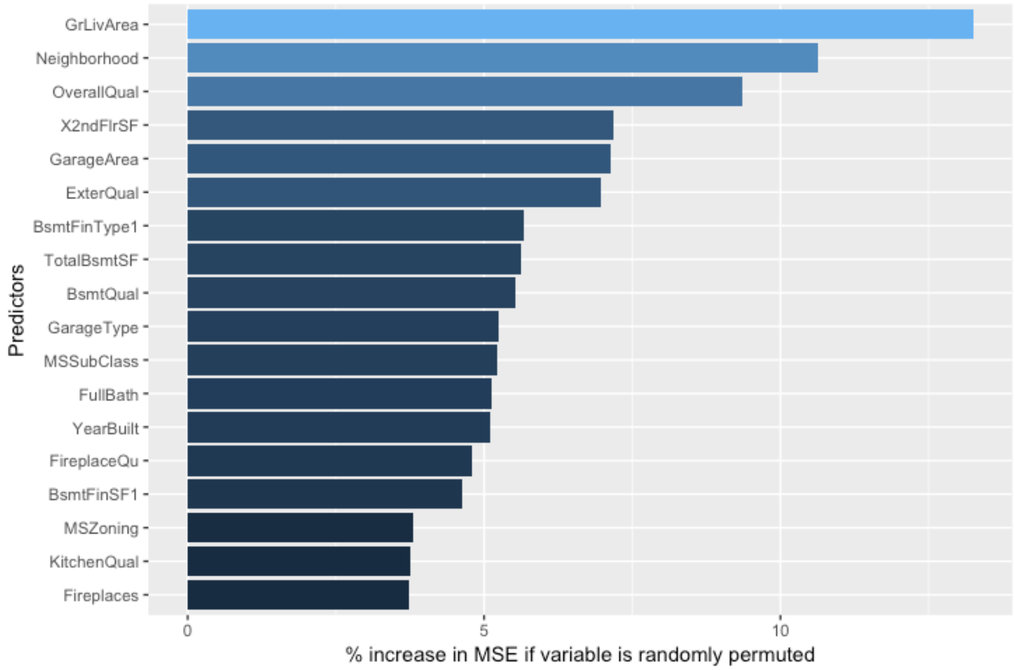
After dividing our data into training and test data like in Bagging, using mtry = 4 (square root of the number of predictors) and number of trees = 100, we applied random forest method on our training data.

```
rf.house_data          <-          randomForest(x=house_data.train[,-19],
y=house_data.train$SalePrice, ntree=100,importance=TRUE, mtry = 4)
```

Also, we created a plot for important variables found using Random Forest.

```
imp.rf <- importance(rf.house_data)
imp.DataFrame   <-   data.frame(Variables   =   row.names(imp.rf),   MSE   =
imp.rf[,1])
imp.DataFrame  <-  imp.DataFrame[order(imp.DataFrame$MSE,  decreasing  =
TRUE),]

ggplot(imp.DataFrame,     aes(x=reorder(Variables,     MSE),     y=MSE,
fill=MSE),col='red')   +   geom_bar(stat   =   'identity')   +   labs(x   =
'Predictors', y= '% increase in MSE if variable is randomly permuted')
+ coord_flip() + theme(legend.position="none")
```



Finally, we predicted the MSE on test data which came out as **0.1134977**

3. Boosting

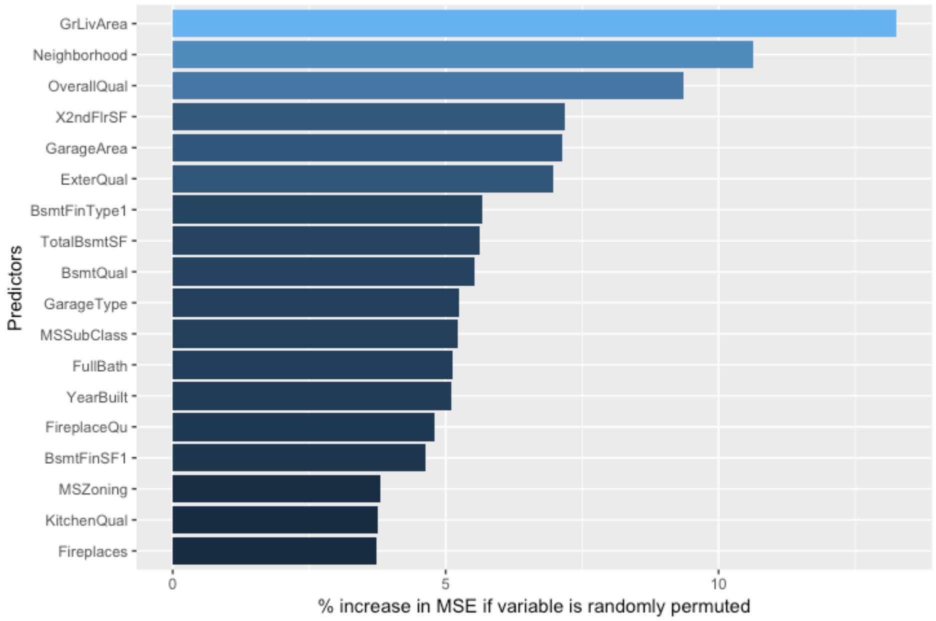
Choosing number of trees = 100, shrinkage parameter = 0.1 and interaction depth = 1 (optimal), we applied boosting on our training data.

```
boost.house_data           =gbm(SalePrice~.,data=house_data.train,
distribution="gaussian",n.trees    =100    ,shrinkage    =    0.1,
interaction.depth =1)
```

Also, we created a plot for important variables found using Boosting.

```
imp.boost <- importance(boost.house_data)
imp.DataFrame <- data.frame(Variables = row.names(imp.rf), MSE =
imp.rf[,1])
imp.DataFrame      <-      imp.DataFrame[order(imp.DataFrame$MSE,
decreasing = TRUE),]

ggplot(imp.DataFrame,   aes(x=reorder(Variables,   MSE),   y=MSE,
fill=MSE),col='red')  + geom_bar(stat  =  'identity')  + labs(x  =
'Predictors',  y=  '% increase  in  MSE  if  variable  is  randomly
permuted') + coord_flip() + theme(legend.position="none")
```



Finally, we predicted the MSE on test data which came out as **0.11619**

Comparison

By applying 6 different models on our training data and predicting its accuracy on the test data, we had a comprehensive performance evaluation of each of the models with respect to each other.

Before starting the application of models, we predicted that the multiple linear regression would perform poorly, and boosting would provide the best results. This was influenced by our theoretical knowledge about the accuracy of the methods and after examining the data.

Talking of trees, interestingly, Random Forest model provided the least MSE on the test data (0.1135) and predicted the SalePrice value on the test data accurately. Though, boosting and bagging didn't fall much behind either. Boosting ended up providing a test error rate of 0.1161, a bit over that of random forest and bagging produced a test error rate of 0.1268.

Also, we could get a glimpse of the computation slowness and overfit of the model when we applied boosting with number of trees =1000. Using a similar value of number of trees for other models didn't affect the test error rate in case of bagging and random forest.

Without any surprise, the multiple linear regression produced the worst MSE since a few predictors were categorical and discrete while some were numeric and continuous. Thus, the model wasn't flexible enough to account for all kinds of relationships between the response and different predictors. Thus, failing to fit the data and produced a high test error rate of 0.854.

Next, we compare the 2 shrinkage methods Ridge Regression and LASSO, where we obtained a test error rate of 0.1483 and 0.1547 respectively. Ridge Regression was faster to compute and having the property to take into effect all predictors (without zeroing out coefficients) helped us in our case and produced the best results between the 2 shrinkage methods.

Overall Comparison

According to our data which had both quantifiable and qualitative predictors, multiple linear regression didn't fit the data well. Shrinkage methods did apply fairly good. But tree models produced the most accurate results and Random Forest is the way to go with its test error rate valued at 0.1161.

The comparison of the models seems a logical step at this point, but sometimes to produce an even more accurate result and considering a much larger set of observations, we might take an average of the results obtained by the shrinkage and the tree methods (excluding multiple linear regression, which is an obvious drop out).

Executive Summary

Importance in the Real World

Buying and selling homes is a big hassle for people who are involved in any of these activities and have a significant importance in an individual's life. It has been observed that 87% of buyers purchased their home through a real estate agent or broker—a share that has steadily increased from 69 percent in 2001. There are various platforms available for sellers as well as buyers to avoid a middleman and explore various options at ease. Among the buyers surveyed, majority of them responded that they tend to compromise on one of the features in lieu of another as they are not aware of vast options across the city. Through this project, we have developed a solution in order to facilitate the process of selling and buying a residential home.

Gap in Literature

In order to achieve our project objective, we decided not to rely on the most popular or the most advocated method for prediction. We followed a systematic approach of devising various prediction models and then compared the results obtained from these techniques to arrive at the house sale price. Analysing data by employing several methods gave us a wider perspective on the trends that we have observed and helped us make the most efficient decision at each step.

The fact that we have considered a comprehensive list of possible factors that may contribute to the final sale price of a house adds to the credibility of our findings.

We were able to conclude that, for the dataset at hand and the objective described above, Random Forest method has proven to be more accurate as opposed to our initial assumption that Boosting would prove to be better.

Implication of the Result

Our aim is to propose sales price of residential homes through regression analysis by studying various features that a buyer desires in a home and a seller has to offer. Through this model, we are trying to predict the prices based on important features that stand out among others. In our data analysis, we tried to compare all the methods based on the test error rates.

We select the model with minimum test error as the best model. We emphasize on test error even if the training error for that model is slightly high as we have a real world scenario to implement or solution on. To preserve the purpose of our project, we have to be very specific about sales prices by considering all the factors involved in determining it as people would spend a considerable amount of money based on the model.

This method would help us to predict the sales price of residential homes but we do not achieve this with 100% accuracy. Practically, it is not possible to achieve that accuracy with any of the methods that we have available with us,

Conclusions

After analyzing the data set and removing the missing values, we applied various exploratory data analysis techniques on the data to identify the data patterns and to rank the predictors according to their importance levels. This was followed by the steps taken to identify the predictor correlations in an attempt to choose the ones that would contribute the most to the reduction of the prediction error. In order to obtain a comprehensive set of models from various strata of statistical modelling, we applied regression, shrinkage and tree methods after which the obtained results were compared. We conclude that applying Random Forest on the data set gives the most accurate prediction for the house sale prices. Some of the interesting facts which we could conclude from our analysis include:

1. We were able to predict the house prices around the Ames, Iowa with approximately 89-90% accuracy.
2. Concluding from various models applied on the data, house prices depend the most on ground living area and the neighborhood of the property.
3. House prices in areas Stone Brook, Northridge and Northridge Heights begin from low rates and range up to the costliest house prices in the city. Thus, we could conclude that

these areas are rich part of the city and a seller could set a higher price if selling a house in these areas.

4. Due to a large number of attributes which were quantifiable and categorical at times, we obtained the best results from the tree models (error rate for random forest ~0.1135) and second-best results from shrinkage methods.

Future Scope

To achieve a greater prediction accuracy, we need more observations about the features listed in our data and cleaner data set so that data is as accurate as possible. We can employ further resampling strategies to better attain the problem of missing and limited data. A good technique can be SMOTE(Synthetic Minority Over-Sampling Technique). The model can be trained better if we also accommodate the choice of buyers about several features that they look for in a residential form. Based on their choices, we can assign weights to the features and better predict sales price. Neural networks is another method that can be explored to uncover several hidden layers as they have high computation power and thus can facilitate us in training our model in a better manner.

References

[1] *https://www.kaggle.com/c/house-prices-advanced-regression-techniques*

[2] *https://www.nar.realtor/research-and-statistics/quick-real-estate-statistics*

[3] *https://www.kaggle.com/shaoyingzhang/data-exploration-and-prediction-of-house-price*

[4] *http://www.sthda.com/english/articles/38-regression-model-validation/157-cross-validation-essentials-in-r/*

[5] *https://www.kaggle.com/c/house-prices-advanced-regression-techniques/kernels?sortBy=%20hotness&group=everyone&pageSize=20&language=R&competitionId=5407*

[6] *An Introduction to Statistical Learning ISBN 978-1-4614-7138-7 (eBook)*

Appendix

Please find below code snippets of the important parts of the code:

```
# Importing the data

house_data=read.table("train.csv",header=TRUE,sep=",")

# Removing the column with name 'ID'
house_data=subset(house_data,select=-c(Id))
head(house_data)

# Size of the dataset
dim(house_data)

# Exploratory Data Analysis

# Variable Information and trends
library(ggplot2)
ggplot(data=house_data,aes(SalePrice))+geom_histogram(bins=100,col='re
d')+xlab('Sale Price')+ylab('Count')

# Summary of sale price
print('Summary of Sale Price : ')
summary(house_data$SalePrice)

# Analyzing Neighborhood predictor
ggplot(house_data,aes(x=Neighborhood,y=SalePrice))+geom_boxplot()+them
e(axis.text.x = element_text(angle = 45, hjust = 1))

library(ggridges)
ggplot(
  house_data,
  aes(x = SalePrice, y = Neighborhood)
  ) +
  geom_density_ridges_gradient(
    aes(fill = ..x..), scale = 3, size = 0.3
    ) +
  scale_fill_gradientn(
    colours = c("#0D0887FF", "#CC4678FF", "#F0F921FF"),
    name = "Sale Price [$]"
    )+
  labs(title = 'Sale price in different neighborhoods')

# Analyzing Garage Area predictor
library(ggpubr)
ggdensity(house_data, x = "GarageArea",
          fill = "#0073C2FF", color = "#0073C2FF",
          add = "mean", rug = TRUE)+xlab('Garage
Area')+ylab('Density')

# Analyzing Foundation predictor
boxplot(SalePrice~Foundation,data=house_data,col=colors()[100:102],mai
n='Sale price variation with Foundation',xlab='Foundation',ylab='Sale
Price')

# Finding Correlations

library(corrplot)
numeric_variables <- unlist(lapply(house_data, is.numeric))
num_data=house_data[,numeric_variables]
Cor=cor(num_data,use='pairwise.complete.obs')

col=colorRampPalette(c("red", "white", "blue"))(20)
corrplot(Cor, type="upper", order="hclust", col=col,tl.cex=0.5)

# Finding the factors having correlation greater than 0.5
cor_sorted=as.matrix(sort(Cor[,'SalePrice'], decreasing = TRUE))
cor_high=names(which(apply(cor_sorted, 1, function(x) abs(x)>0.5)))
cor_final=Cor[cor_high,cor_high]

corrplot(cor_final, type="upper", method='number')

# Sctterplots for most correlated predictors with SalePrice
ggplot(house_data, aes(x=OverallQual, y=SalePrice)) +
  geom_point(size=2, shape=23)+geom_smooth(method='lm')

ggplot(house_data, aes(x=GrLivArea, y=SalePrice)) +
  geom_point(size=2, shape=23)+geom_smooth(method='lm')

ggplot(house_data, aes(x=GarageArea, y=SalePrice)) +
  geom_point(size=2, shape=23)+geom_smooth(method='lm')

ggplot(house_data, aes(x=YearBuilt, y=SalePrice)) +
  geom_point(size=2, shape=23)+geom_smooth(method='lm')

# Check for missing data

# Total number of missing values grouped by column names
colSums(sapply(house_data,is.na))

# Arranging in decreasing order
sort(colSums(sapply(house_data, is.na)), decreasing = TRUE)

# Filling the missing data (only a few predictors shown here)

# PoolQC

temp=as.character(house_data$PoolQC)
temp[which(is.na(house_data$PoolQC))]='None'
house_data$PoolQC=as.factor(temp)

# MiscFeature

temp=as.character(house_data$MiscFeature)
temp[which(is.na(house_data$MiscFeature))]='None'
house_data$MiscFeature=as.factor(temp)

# Alley

temp=as.character(house_data$Alley)
temp[which(is.na(house_data$Alley))]='None'
house_data$Alley=as.factor(temp)

# Fence (Qualitative)

temp=as.character(house_data$Fence)
temp[which(is.na(house_data$Fence))]='None'
house_data$Fence=as.factor(temp)


# GarageType (Qualitative)

temp=as.character(house_data$GarageType)
temp[which(is.na(house_data$GarageType))]='None'
house_data$GarageType=as.factor(temp)

# Important predictors from Random Forest
```

```
library(randomForest)

set.seed(2018)
row.has.na <- apply(house_data, 1, function(x){any(is.na(x))})
datatest <- house_data[!row.has.na,]
quick_RF <- randomForest(x=datatest[,-80], y=datatest$SalePrice,
ntree=100,importance=TRUE, mtry = 9)
imp_RF <- importance(quick_RF)
imp_DF <- data.frame(Variables = row.names(imp_RF), MSE = imp_RF[,1])
imp_DF <- imp_DF[order(imp_DF$MSE, decreasing = TRUE),]

ggplot(imp_DF[1:20,], aes(x=reorder(Variables, MSE), y=MSE,
fill=MSE),col='red') + geom_bar(stat = 'identity') + labs(x =
'Predictors', y= '% increase in MSE if variable is randomly permuted')
+ coord_flip() + theme(legend.position="none")
```

# Final important variables after validating results from Correlation plot and important predictors obtained using Random Forest

```
imp_var=c('Neighborhood','GrLivArea','X2ndFlrSF','OverallQual','TotalB
smtSF','BsmtFinSF1','YearBuilt','BsmtFinType1','GarageArea','MSZoning'
,'KitchenQual','ExterQual','BsmtQual','FireplaceQu','GarageType','Fire
places','FullBath','MSSubClass','SalePrice')

final_data=house_data[,imp_var]
```

# Application of Models

```
# Scaling the sales data to 100,000 to standardize the predictors and
response to one level before applying any model on the data
final_data$SalePrice <- final_data$SalePrice/100000

# Create train and test data
train <- sample(1:nrow(final_data), nrow(final_data) / 2)
house_data.train <- final_data[train, ]
house_data.test <- final_data[-train, ]
```

# Multiple linear regression

```
set.seed(1)
mlr.housing_data <- lm(SalePrice ~., data = house_data.train)
mlr.predictions <- predict(mlr.housing_data,
house_data.test[!(house_data.test$Neighborhood=="Blueste"),])

# Test Error Rate
```

```
mean((mlr.predictions-house_data.test$SalePrice)^2)

# Ridge Regression (Shrinkage Method)

library(glmnet)

# Creating X and y tensors for input
X=model.matrix(SalePrice~.,data=final_data)
y=final_data$SalePrice

# Ridge regression model
grid=10^seq(10,-2,length=100)
ridge.house_data=glmnet(X[train,],y[train],alpha=0,lambda=grid)
dim(coef(ridge.house_data))

# Using CV to find the best lambda
set.seed(1)
cv_ridge=cv.glmnet(X[train,],y[train],alpha=0)
plot(cv_ridge)


# Best lambda
cat('The best lambda obtained using Cross Validation for ridge is
:',cv_ridge$lambda.min)

# Prediction using best lambda
yhat_ridge=predict(ridge.house_data,s=cv_ridge$lambda.min,newx=X[-
train,])

# Test Error Rate
mean((yhat_ridge-y[-train])^2)

# LASSO (Shrinkage Method)

lasso.house_data=glmnet(X[train,],y[train],alpha=1,lambda=grid)
dim(coef(lasso.house_data))

# Using CV to find the best lambda
set.seed(1)
cv_lasso=cv.glmnet(X[train,],y[train],alpha=1)
plot(cv_lasso)

# Best lambda
cat('The best lambda obtained using Cross Validation for lasso is
:',cv_lasso$lambda.min)
```

```
# Prediction using best lambda
yhat_lasso=predict(lasso.house_data,s=cv_lasso$lambda.min,newx=X[-
train,])

# Test Error Rate
lasso_err=mean((yhat_lasso-y[-train])^2)

# Bagging (Tree Method)

set.seed(1)
bag.house_data <- randomForest(x=house_data.train[,-19],
y=house_data.train$SalePrice, ntree=100,importance=TRUE, mtry = 18)

# Important Variables
imp.bag <- importance(bag.house_data)
imp.DataFrame <- data.frame(Variables = row.names(imp.bag), MSE =
imp.bag[,1])
imp.DataFrame <- imp.DataFrame[order(imp.DataFrame$MSE, decreasing =
TRUE),]

ggplot(imp.DataFrame, aes(x=reorder(Variables, MSE), y=MSE,
fill=MSE),col='red') + geom_bar(stat = 'identity') + labs(x =
'Predictors', y= '% increase in MSE if variable is randomly permuted')
+ coord_flip() + theme(legend.position="none")

yhat.bag = predict(bag.house_data, newdata = house_data.test)

# Test Error Rate
mean((yhat.bag-house_data.test$SalePrice)^2)

# Random Forest (Tree Method)

set.seed(1)
rf.house_data <- randomForest(x=house_data.train[,-19],
y=house_data.train$SalePrice, ntree=100,importance=TRUE, mtry = 4)

# Important Variables
imp.rf <- importance(rf.house_data)
imp.DataFrame <- data.frame(Variables = row.names(imp.rf), MSE =
imp.rf[,1])
imp.DataFrame <- imp.DataFrame[order(imp.DataFrame$MSE, decreasing =
TRUE),]
```

```
ggplot(imp.DataFrame, aes(x=reorder(Variables, MSE), y=MSE,
fill=MSE),col='red') + geom_bar(stat = 'identity') + labs(x =
'Predictors', y= '% increase in MSE if variable is randomly permuted')
+ coord_flip() + theme(legend.position="none")

yhat.rf = predict(rf.house_data, newdata = house_data.test)

# Test Error Rate
mean((yhat.rf-house_data.test$SalePrice)^2)
```

# Boosting (Tree Method)

```
library (gbm)
set.seed (1)
boost.house_data =gbm(SalePrice~.,data=house_data.train,
distribution="gaussian",n.trees =100 ,shrinkage = 0.1,
interaction.depth =1)

# Important Variables
imp.boost <- importance(boost.house_data)
imp.DataFrame <- data.frame(Variables = row.names(imp.rf), MSE =
imp.rf[,1])
imp.DataFrame <- imp.DataFrame[order(imp.DataFrame$MSE, decreasing =
TRUE),]

ggplot(imp.DataFrame, aes(x=reorder(Variables, MSE), y=MSE,
fill=MSE),col='red') + geom_bar(stat = 'identity') + labs(x =
'Predictors', y= '% increase in MSE if variable is randomly permuted')
+ coord_flip() + theme(legend.position="none")

yhat.boost = predict(boost.house_data, newdata = house_data.test,
n.trees = 100)

# Test Error Rate
mean((yhat.boost-house_data.test$SalePrice)^2)
```