Kaggle Challenge : Ames House Price Prediction

Sumedh R. Sankhe, 1st Year, Dec 2018

Shivayogi Biradar, 1st Year, May 2019

Sharyu Deshmukh, 1st Year, Dec 2018

1 Introduction

Understanding the customer needs and predicting the customer's purchase intents is the core for success for any business. From this dataset we are trying to accurately predict the final price of each property in the city of Ames Iowa, based on a number of predictor variables that we have been given. The goal of the problem is to utilize machine learning tools to make the best possible prediction of house prices. This is an interesting problem because most people will eventually buy/sell a home. This problem allows us, as data scientists, to learn more about the housing market and helps with making more informed decisions.

1.2 Data Description

The dataset represents residential properties in Ames, Iowa from 2006 to 2010. The dataset has been already partitioned for us into test and trainset each having 1460 and 1459 observations respectively. Both datasets contain 79 explanatory variables composed of 46 categorical and 33 continuous variables that describe house features such as neighborhood, square footage, number of full bathrooms, and many more. The test set has the response variable **SalePrice** which is what we will predict.

For a detailed view of the data dictionary :
https://www.kaggle.com/c/house-prices-advanced-regression-techniques/data

2 Methods

2.1 Data Preprocessing

Extensive data preprocessing was done so as to eliminated the missing values. The missing value for majority of the variables turned out to be "None" since the particular property did not have that feature. For instance PoolQC gives us information about whether a property has a swimming pool or no, hence we see most of the houses have missing values for this feature. Furthermore variables such are GarageType indicate existence of a garage, since Garage has a lot of other dependent variables they also become NA if a particular property does not have a garage.

Table 1: Missing Values

Features	Missing_Count
PoolQC	1453
MiscFeature	1406
Alley	1369
Fence	1179
FireplaceQu	690
LotFrontage	259
GarageType	81
GarageYrBlt	81
GarageFinish	81
GarageQual	81

On visualizing the response variable it was found out that the distribution was right skewed, using log transformation on the response variable helped removed the skewness.

3 Modeling

3.1 Elastic Net Regression

In statistics and, in particular, in the fitting of linear or logistic regression models, the elastic net is a regularized regression method that linearly combines the L1 and L2 penalties of the lasso and ridge methods. The target function of Lasso is given as:

$$min_w = \frac{1}{2n}||X^T w - y||_2^2 + \alpha||w||_1$$

where n is number of samples, w is the coefficient parameters to learn, $||w||_1$ term is L1 penalty, which promotes sparsity, reduces the redundancy and improves the accuracy and robustness of regression (it alleviates the over fitting in some degree). Also if there is a group of highly correlated variables, then the Lasso tends to select one variable from a group and ignore the others. However, the L1 norm in Lass has some limitations that, for example, in the "large p and small n" case (high-dimensional data with few examples), the Lasso selects at most n variables before it saturates. While the target function of Ridge is computed by:

$$min_w = \frac{1}{2}||X^T w - y||_2^2 + \frac{\alpha}{2}||w||_2$$

where $||w||_2^2$ term is L2 penalty, which constrains the module of w in to a L2 ball to alleviate the over fitting problem. However, L2 norm shrinkage the value of approximated parameters, but does not make it zero, which means it does not perform the function of parameter selection. For Elastic Net, it can be treated as a compromise between Lasso and Ridge, since it integrates the L1 and L2 regularization. Its target function is described as:

$$min_w = \frac{1}{2n}||X^T w - y||_2^2 + \alpha\rho||w||_1 + \frac{\alpha(1-\rho)}{2}||w||_2^2$$

where ρ is an adaptive hyper-parameter to control the contributions of L1 norm and L2 norm. With this compromise, Elastic Net remains part of the parameter selection function in Lasso and part of rotary stability property in Ridge. Meanwhile, compare to Lasso, Elastic Net not only randomly select one of the correlated variables, but also tends to obtain all of them and enhances their group effect.

3.2 Gradient Boosting Regression

Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. It builds the model in a stage-wise fashion like other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function. Like other boosting methods, gradient boosting combines weak "learners" into a single strong learner in an iterative fashion. It is easiest to explain in the least-squares regression setting, where the goal is to "teach" a model F to predict values in the form

$\hat{y} = F(x)$ by minimizing the mean squared error $\hat{y} - y^2$, averaged over some training set of actual values of the output variable y. Giving a training dataset $(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$, the goal is to find an approximation $\hat{F}(x)$ to a function $F(x)$ that minimizes the expected value of some specified loss function $L(y, F(x))$:

$$\hat{F} = arg\ min_F E_{x,y}[L(y, F(x))]$$

The gradient boosting method assumes a real-valued y and seeks an approximation of $\hat{F}(x)$ in the form of a weighted sum of functions $h_i(x)$ from some class H, called base (or weak)learners:

$$F(x) = \sum_{i=1}^{M} \gamma_i h_i(x) + const.$$

In accordance with the empirical risk minimization principle, the method tries to find an approximation $\hat{F}(x)$ that minimizes the average value of the loss function on the training set. It does so by starting with a model consisting of a constant function $F_0(x)$, and incrementally expanding it in a greedy fashion:

$$F_0(x) = arg\ min_\gamma \sum_{i=1}^{n} L(y_i, \gamma)$$

$$F_m(x) = F_{m-1}(x) + arg\ min_{h \in H} \sum_{i=1}^{n} L(y_i, F_{m-1}(x_i) + h(x_i))$$

where $h \in H$ is a base learner function. Unfortunately, choosing the best function h at each step for an arbitrary loss function L is a computationally unfeasible optimization problem in general. Therefore, we will restrict to a simplification. The idea is top apply a steepest descent step to this minimization problem. If we considered the continuous case, i.e. H the set of arbitrary differential function on R, we would update the model in accordance with the following equations:

$$F_m(x) = F_{m-1}(x) - \gamma_m \sum_{i=1}^{n} \bigtriangledown_{F_{m-1}} L(y_i, F_{m-1}(x_i))$$

$$\gamma_m = arg\ min_\gamma \sum_{i=1}^{n} L(y_i, F_{m-1}(x_i) - \gamma \frac{\delta L(y_i, F_{m-1}(x_i))}{\delta F_{m-1}(x_i)})$$

where the derivatives are taken with respect to the functions $F_i\ for\ i \in 1, ..., m$.

3.3 Random Forest

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees habit of over fitting to their training set. Regression Trees are known to be very unstable, in other words, a small change in your data may drastically change your model. The Random Forest uses this instability as an advantage through bagging resulting in a very stable model. We simply estimate the desired Regression Tree on many bootstrap

samples (re-sample the data many times with replacement and re-estimate the model) and make the final prediction as the average of the predictions across the trees.
As one of the best use cases for random forest is feature selection, we examined which variables work best/worst in each tree.

$$max\ features = \sqrt{n\ feature}$$

4 Result

To measure the accuracy of our models the Root Mean Square error was calculated for them and compared, the RMSE was calculated as:

$$RMSE\ = \sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{n}}$$

Table 2: RMSE errors across models

Model	Test_Error	Train_Error
Ridge	0.12752	0.0875
Lasso	0.13341	0.0880
Elastic Net	0.13270	0.1100
GBM2	0.12781	0.0480
Random Forest	0.13210	0.0800

As a baseline, to begin our analysis we Implemented simple Linear regression and found out the RMSE errors to be: Train Error 0.13721 ,Test Error 0.1482.

Since our model contains a lot of correlated variables implementing regularization methods was the next obvious task.

We implemented Ridge, Keeping $\alpha = 0$ we found best lambda value, using cross validation = 0.069416. Similarly we found best lambda value for Lasso using cross validation for $\alpha = 1$.

Table 3: Model Parameters

Model	Alpha	Lambda
Ridge	0.0	0.0076183
Lasso	1.0	0.0034579
Elastic Net	0.5	0.0069097

Table 4: Model Parameters RF

Model	Random Forest
Fold_assignment	Modulo
ntrees	200
max_depth	20
nbins	20
min_rows	1
sample_rate	0.623
stopping_metric	rmse

We used H2O package in R to implement random forest. The random forest performed best on the training set with the following parameters.

Table 5: Model Parameters GBM2

Model	Gradient Boosting Machine
max_depth	12
sample_rate	0.65
col_sample_rate	0.63
col_sample_rate_per_tree	0.3
col_sample_rate_change_per_level	0.5
min_rows	10
nbins	10

We used H2O package in R to implement Gradient Boosting. The GBM performed best on the training set with the following parameters.

Table 6: Model Parameters GBM2

Model	Gradient Boosting Machine
max_depth	12
sample_rate	0.65
col_sample_rate	0.63
col_sample_rate_per_tree	0.3
col_sample_rate_change_per_level	0.5
min_rows	10
nbins	10

5 Discussion

Clearly our tree based models performed well on the training set and overfit on the test set. Regularization methods such as lasso ,elastic and ridge performed the best. Lasso which removes collinear variable pair performed the best on this data set. In the end the average of predictions of lasso, elastic and ridge had the least test error.

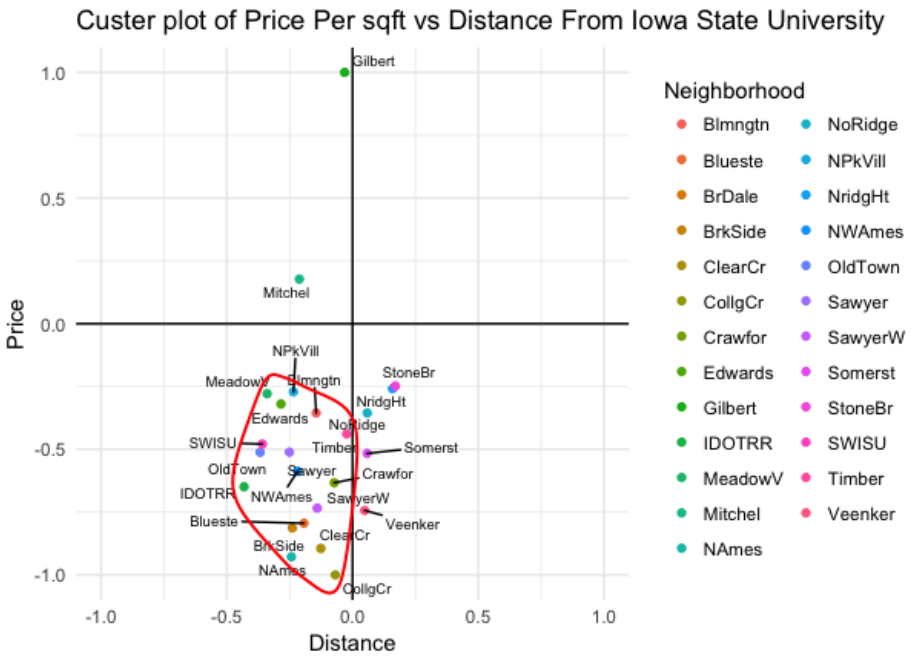
Data pre-processing plays an important part for any machine learning problem, handling the missing value correctly for each type of variable (categorical/continuous). Detecting outliers and deciding on whether to omit them or not. Applying multiple supervised machine learning techniques to the same problem and tweaking the parameters to obtain better predictive results. Better understanding of advanced machine learning alogrithms Random forest, Gradient boosting and working with packages such as H2O and others.

6 An Interesting Viewpoint

We took an additional task of shortlisting the neighborhood for customers interested in buying or renting a home based on listings we had.Customers often look for houses close to an area of interest/need such as office ,college,schools hospitals etc. Using the maps package and the geocode function we collected the coordinates for our target point of interest and the neighborhoods. The Eculidean distance between these points was calculated. We used Predicted saleprice from our best performing model and created a new variable price per square feet.Price per square feet remains pretty much constant throughout a neighborhood and is closely related with the rental price of the property

We scaled price per square feet between a scale of -1 to 1 and varied distance(which was also scaled between - 1 to 1) from prime locations such as hospital,college etc.

Case 1: Consider a student who is trying to look for a rental close to the university at the best price point availability.



By looking at the graph above we can see that the lower left quadrant has the best choice of property to live at a considerably moderate price aroudn the university. The same logic can be extended to other points of interest in the city.

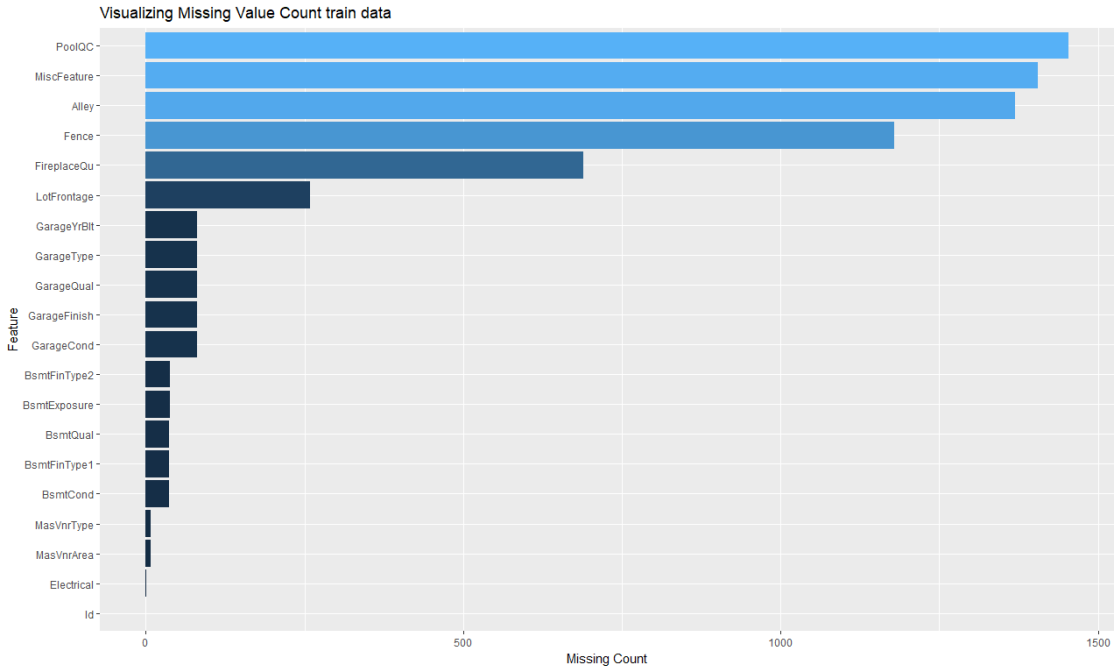
References

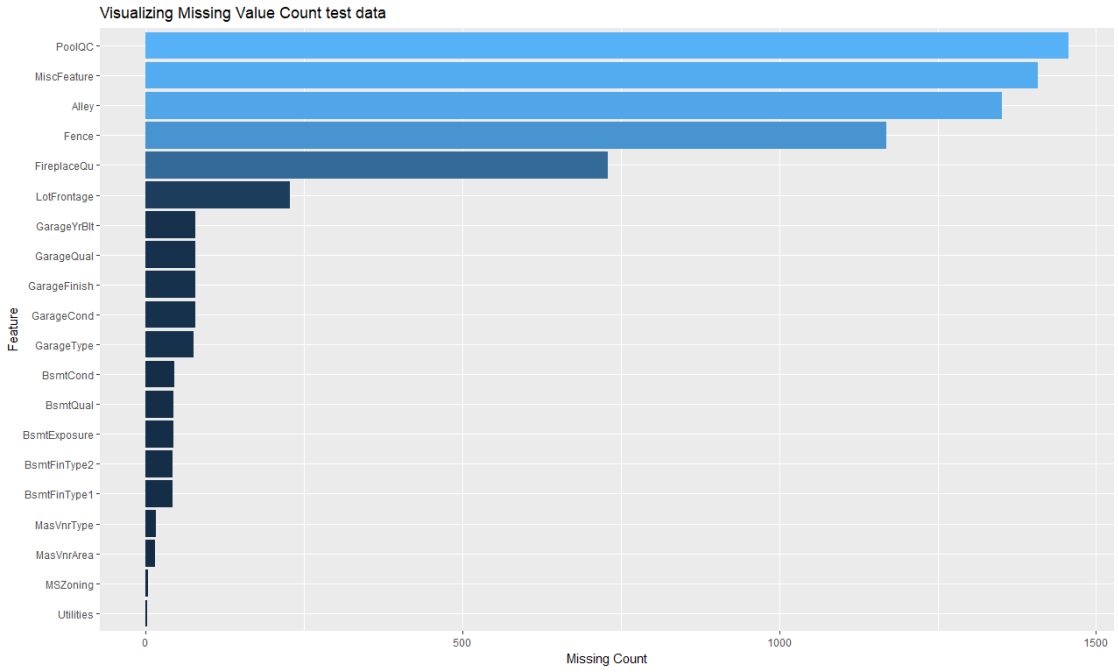
[1] https://www.kaggle.com/c/house-prices-advanced-regression-techniques

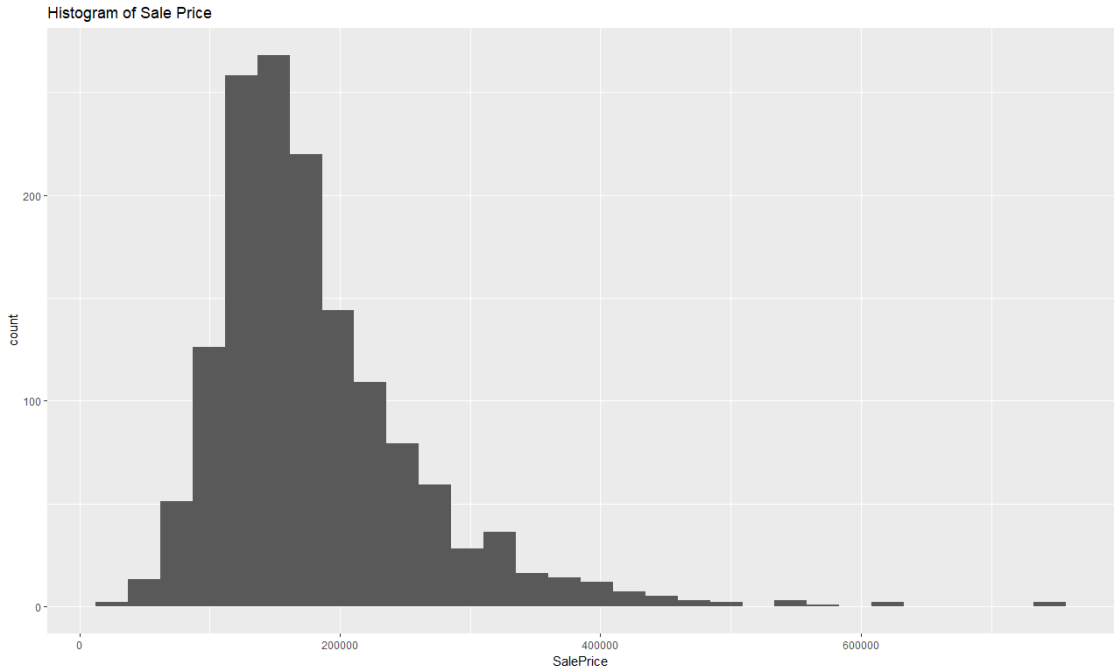
[2] http://ww2.amstat.org/publications/jse/v19n3/decock.pdf

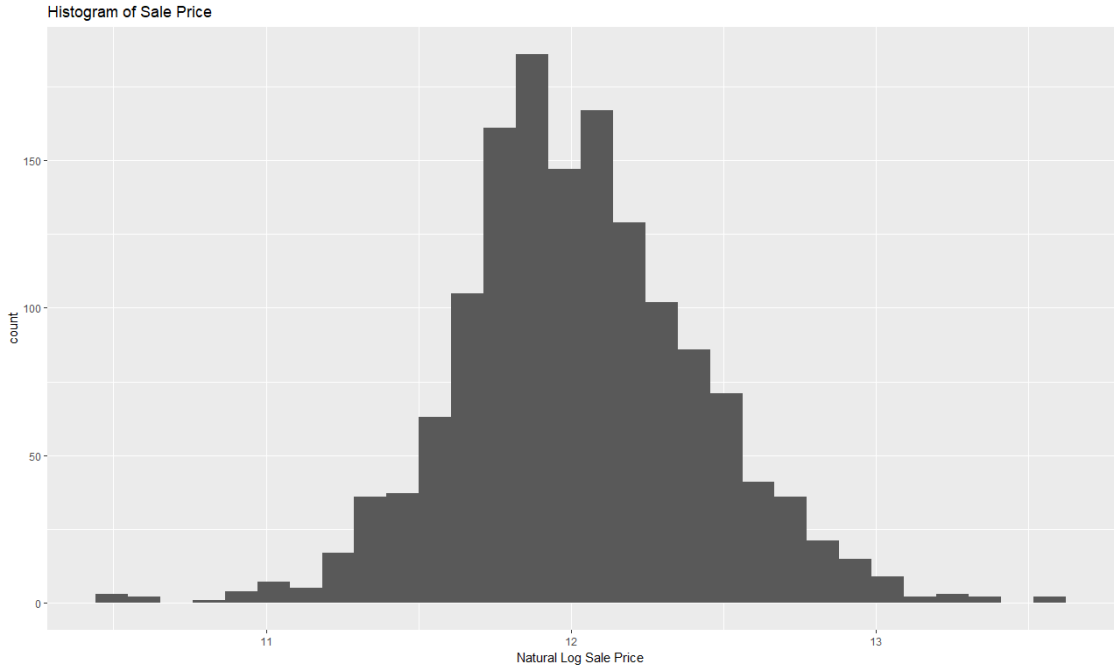
[3]https://github.com/h2oai/h2o-3/blob/master/h2o-docs/src/product/tutorials/gbm/gbmTuning.Rmd

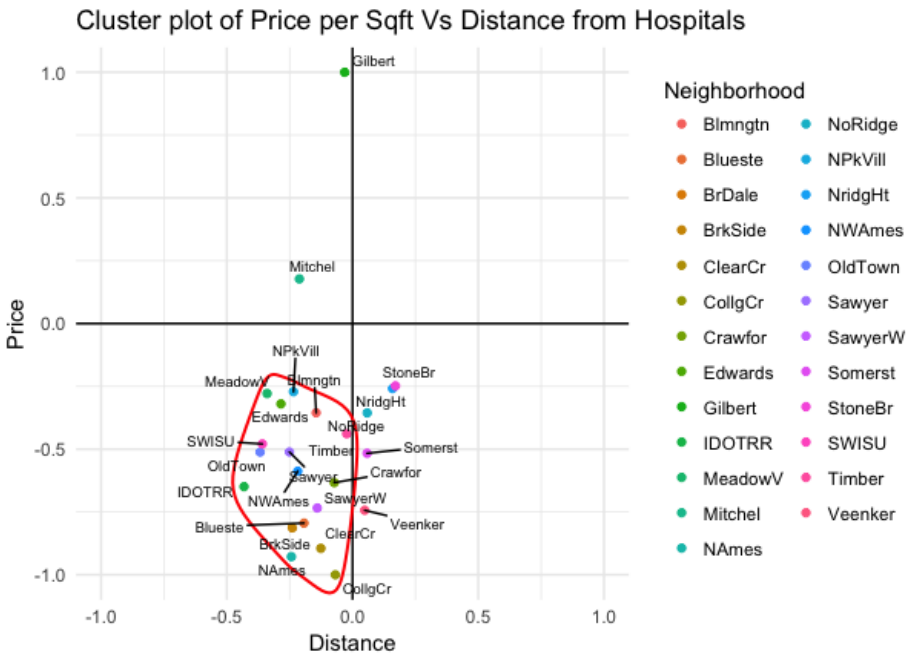
Appendix

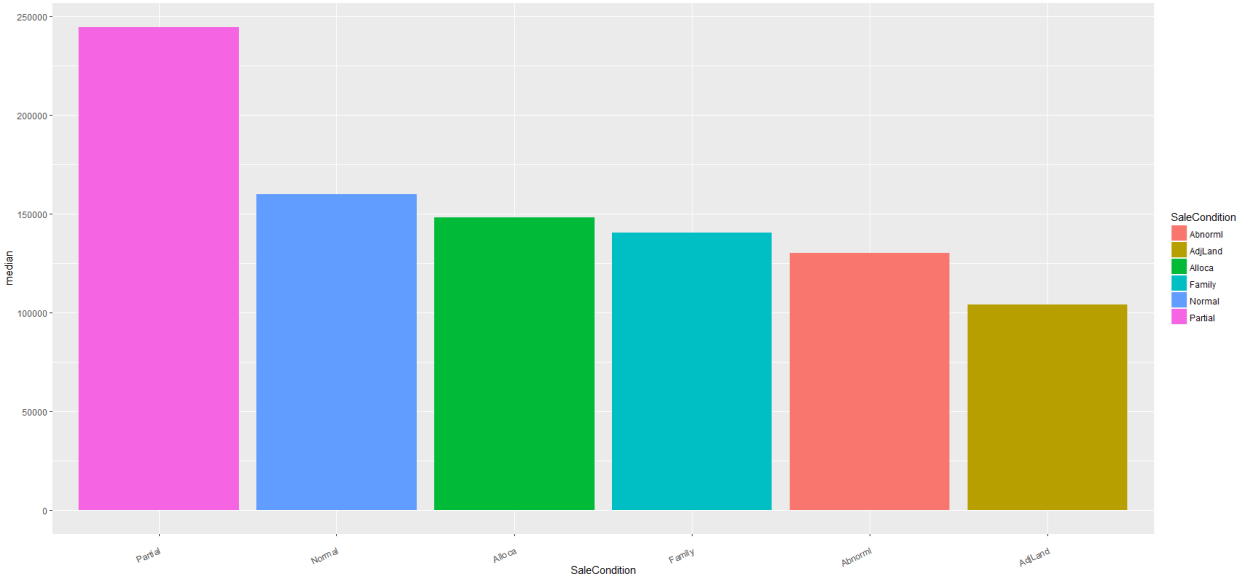

Visualizing Missing Value Count train data
PoolQC
MiscFeature
Alley
Fence
FireplaceQu
LotFrontage
GarageYrBlt
GarageType
GarageQual
GarageFinish
GarageCond
BsmtFinType2
BsmtExposure
BsmtQual
BsmtFinType1
BsmtCond
MasVnrType
MasVnrArea
Electrical
Id
Feature
0
500
1000
1500
Missing Count


Visualizing Missing Value Count test data
PoolQC
MiscFeature
Alley
Fence
FireplaceQu
LotFrontage
GarageYrBlt
GarageQual
GarageFinish
GarageCond
GarageType
BsmtCond
BsmtQual
BsmtExposure
BsmtFinType2
BsmtFinType1
MasVnrType
MasVnrArea
MSZoning
Utilities
Feature
0
500
1000
1500
Missing Count

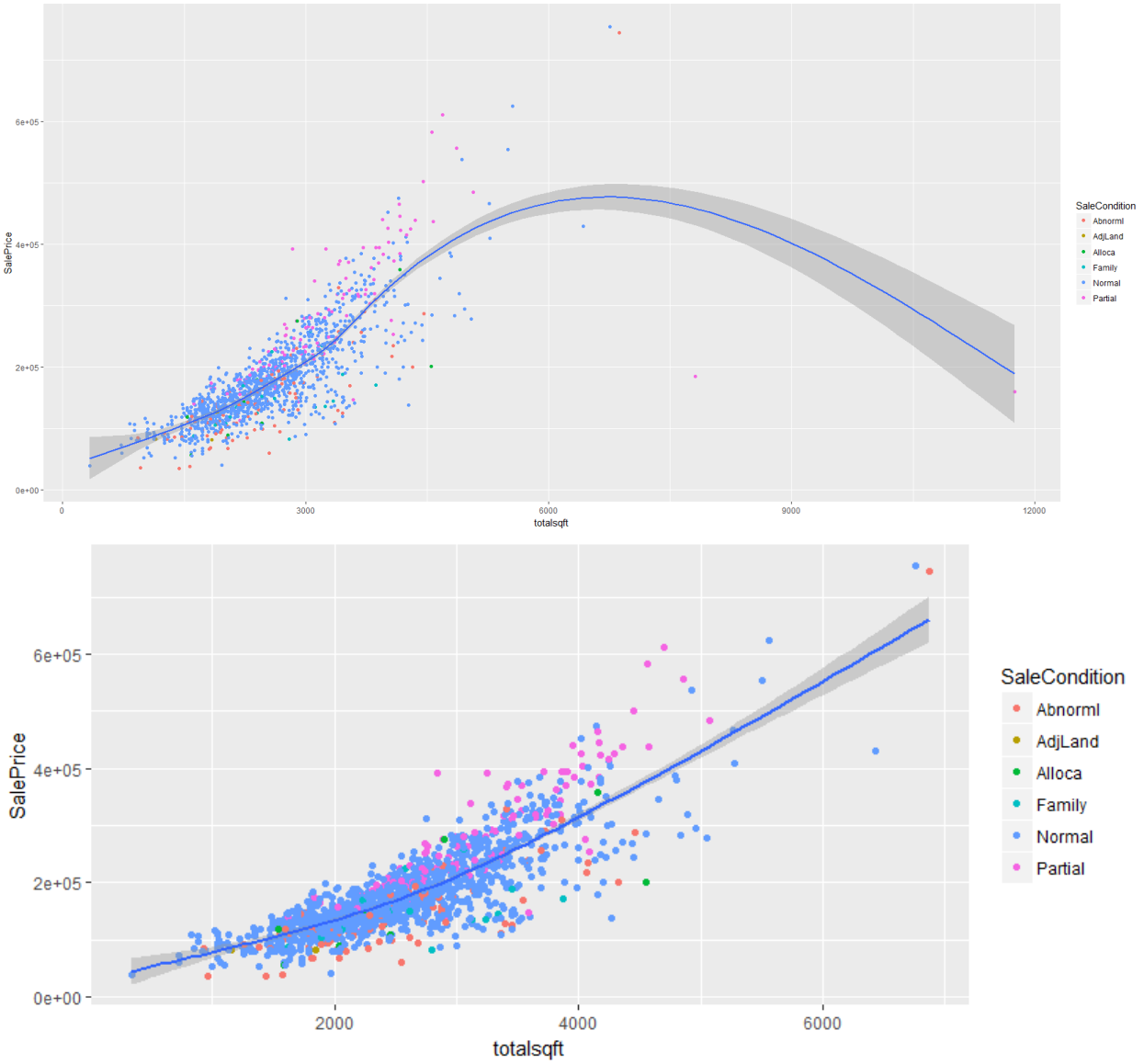

Histogram of Sale Price
count
200
100
0
0
200000
400000
600000
SalePrice

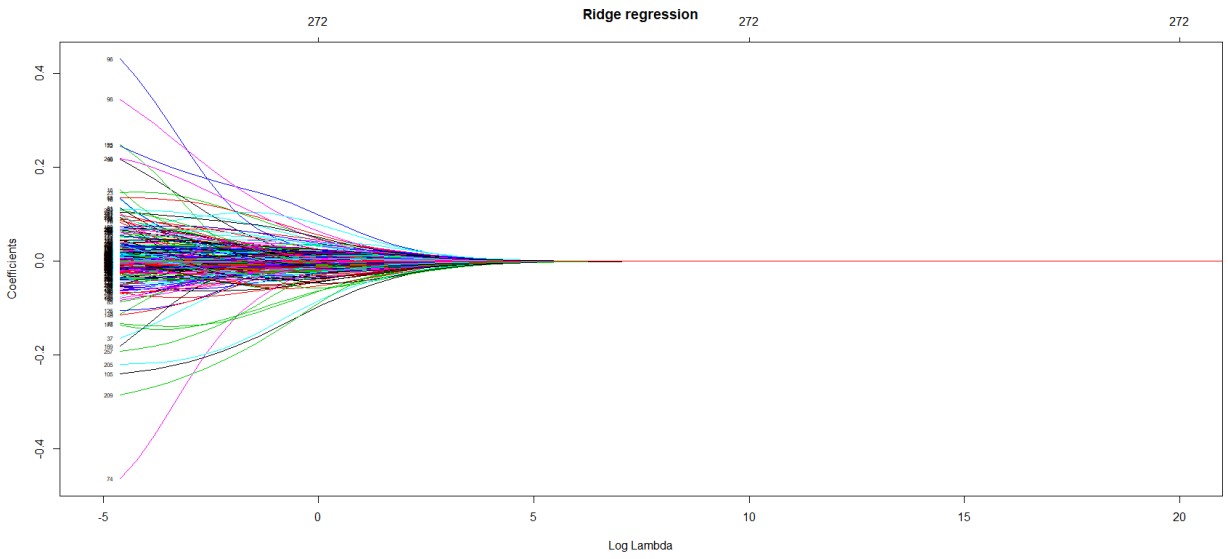

Histogram of Sale Price
count
150
100
50
0
11
12
13
Natural Log Sale Price

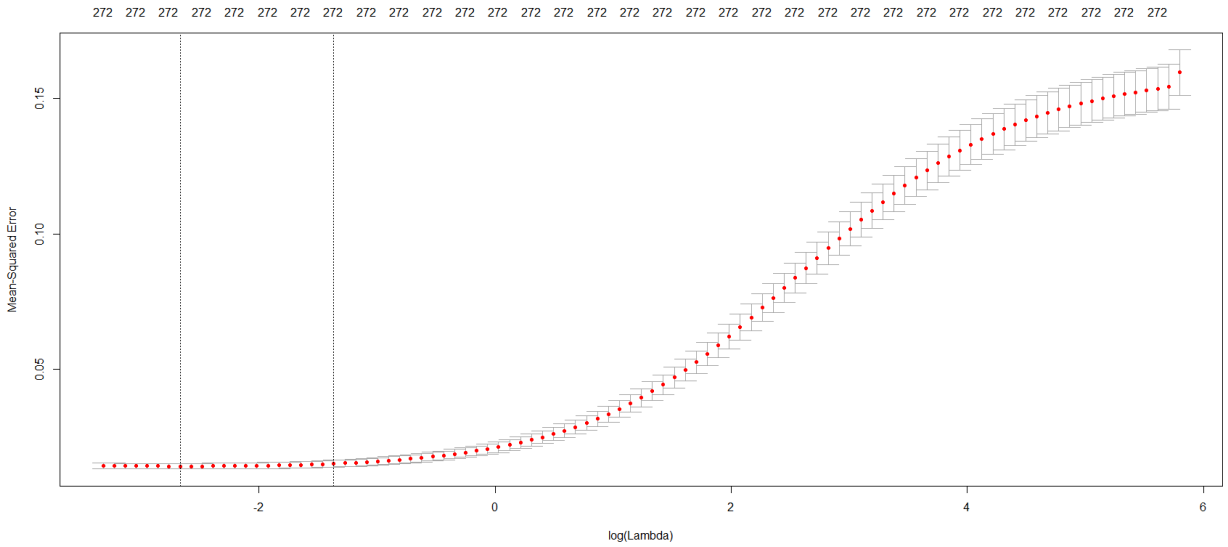

Cluster plot of Price per Sqft Vs Distance from Hospitals
Price
Distance
1.0
0.5
0.0
-0.5
-1.0
-1.0
-0.5
0.0
0.5
1.0
Gilbert
Mitchel
NPkVill
Blmngtn
MeadowV
Edwards
StoneBr
NridgHt
NoRidge
SWISU
Timber
Somerst
OldTown
Sawyer
Crawfor
IDOTRR
NWAmes
SawyerW
Veenker
Blueste
ClearCr
BrkSide
NAmes
CollgCr
Neighborhood
Blmngtn
Blueste
BrDale
BrkSide
ClearCr
CollgCr
Crawfor
Edwards
Gilbert
IDOTRR
MeadowV
Mitchel
NAmes
NoRidge
NPkVill
NridgHt
NWAmes
OldTown
Sawyer
SawyerW
Somerst
StoneBr
SWISU
Timber
Veenker

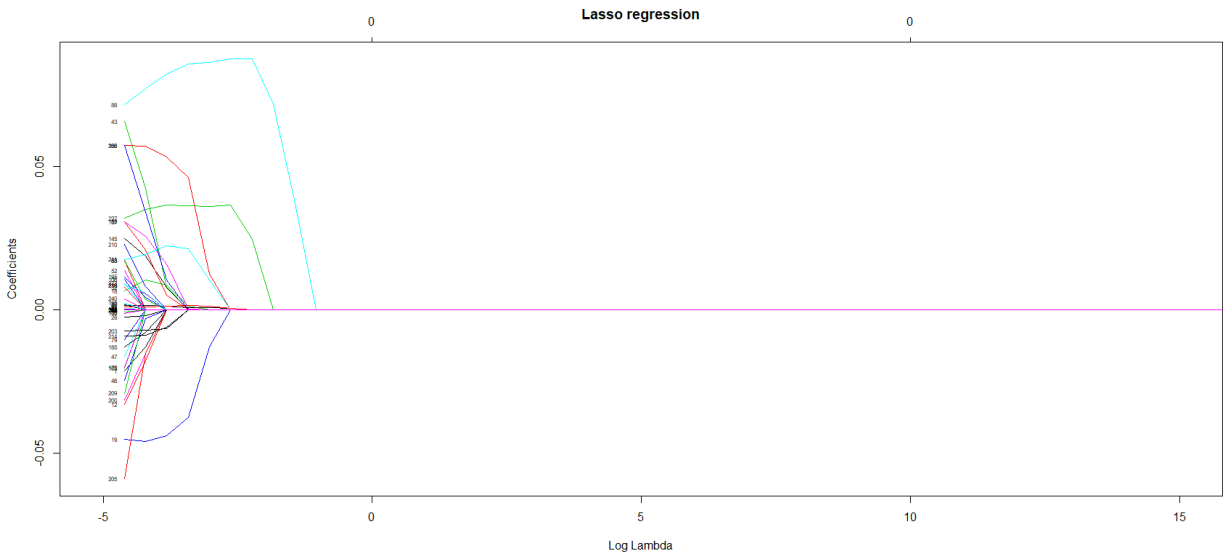

250000
200000
150000
100000
50000
0
median
Partial
Normal
Alloca
Family
Abnorml
AdjLand
SaleCondition
SaleCondition
Abnorml
AdjLand
Alloca
Family
Normal
Partial

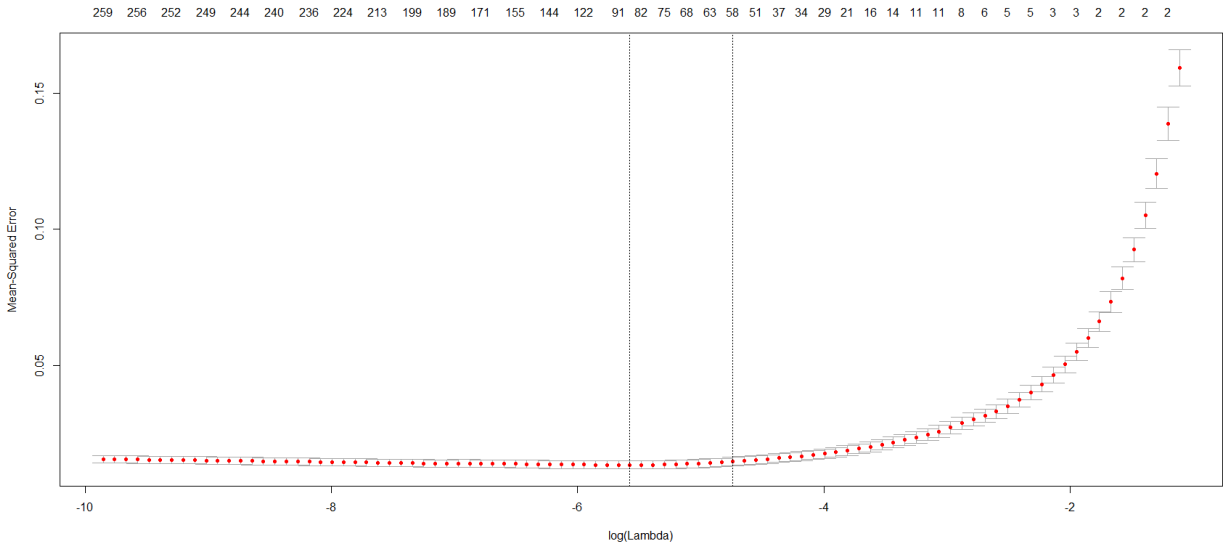
Handling Outliers in the SalePrice

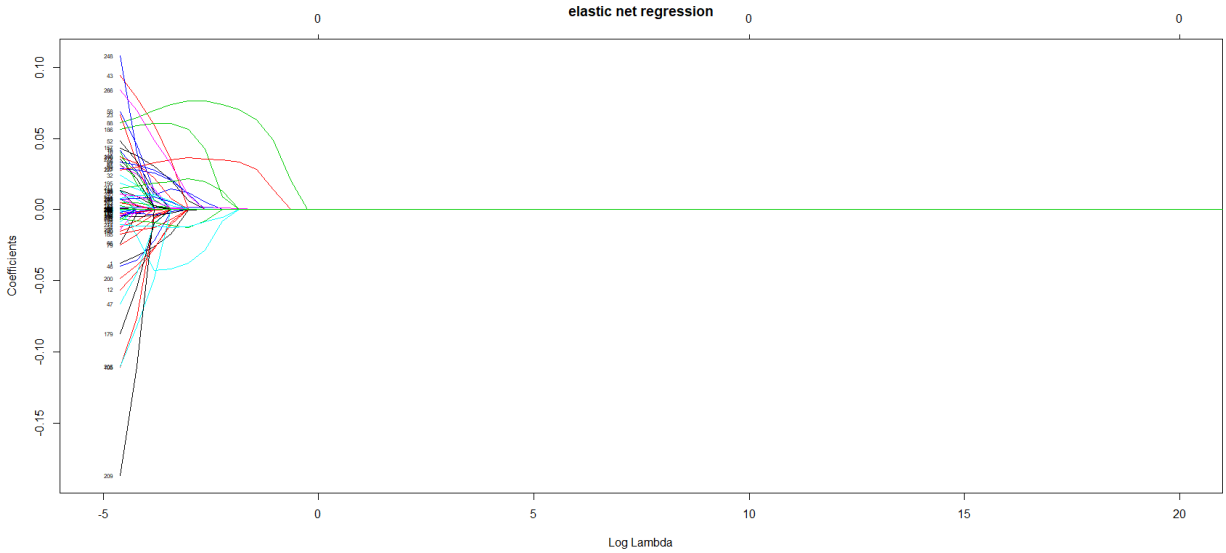


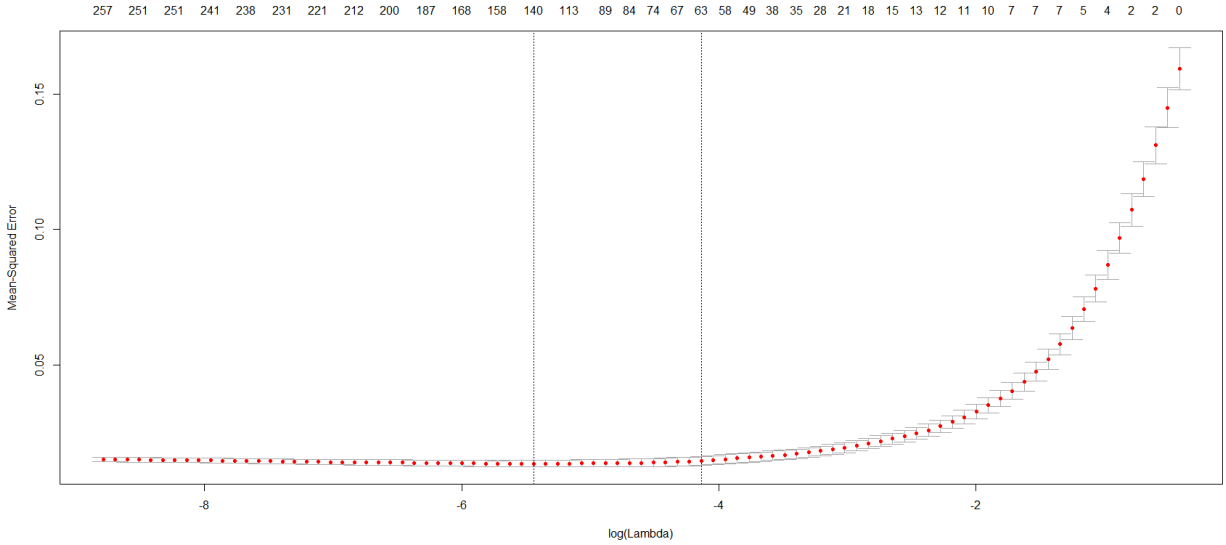

Ridge regression
272
272
272
Coefficients
0.4
0.2
0.0
-0.2
-0.4
-5
0
5
10
15
20
Log Lambda

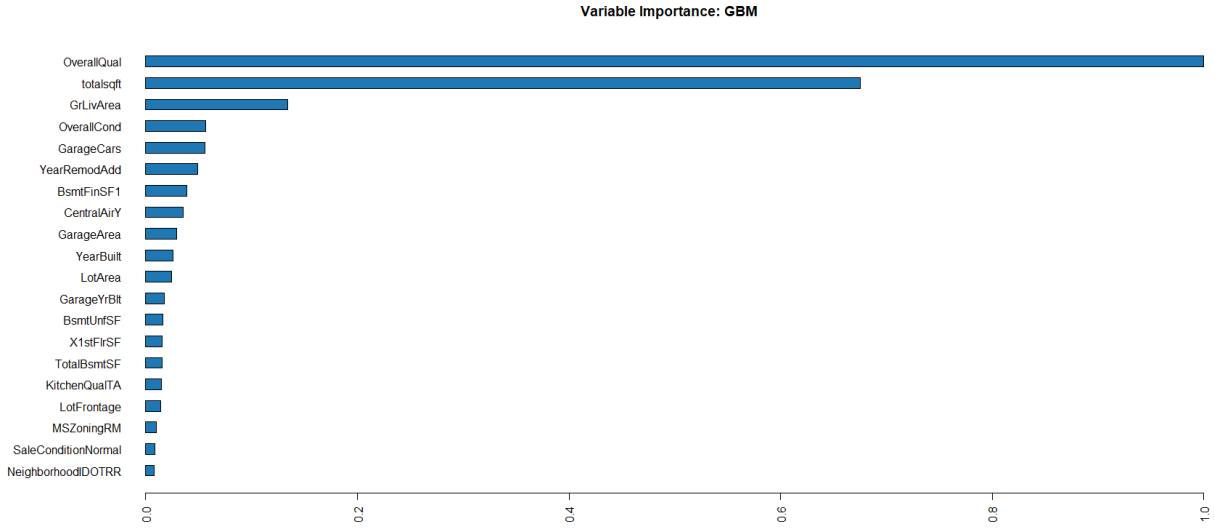

272 272 272 272 272 272 272 272 272 272 272 272 272 272 272 272 272 272 272 272 272 272 272 272 272 272 272 272 272 272 272
Mean-Squared Error
0.15
0.10
0.05
-2
0
2
4
6
log(Lambda)

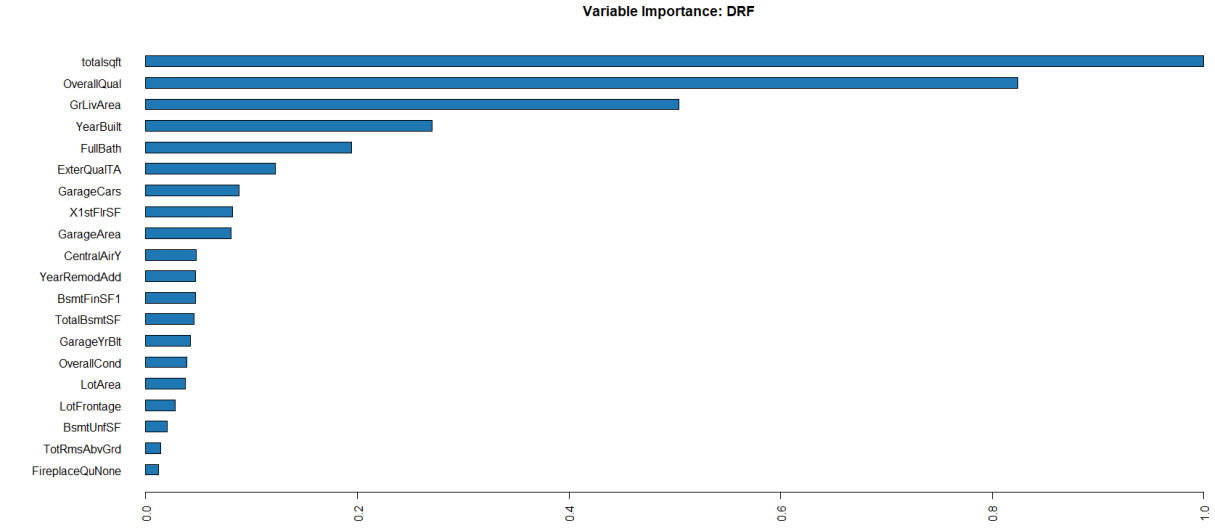

Lasso regression
0
0
Coefficients
0.05
0.00
-0.05
-5
0
5
10
15
Log Lambda


259 256 252 249 244 240 236 224 213 199 189 171 155 144 122 91 82 75 68 63 58 51 37 34 29 21 16 14 11 11 8 6 5 5 3 3 2 2 2 2
Mean-Squared Error
0.15
0.10
0.05
-10
-8
-6
-4
-2
log(Lambda)


elastic net regression
0
0
0
0.10
0.05
0.00
-0.05
-0.10
-0.15
Coefficients
-5
0
5
10
15
20
Log Lambda


257 251 251 241 238 231 221 212 200 187 168 158 140 113 89 84 74 67 63 58 49 38 35 28 21 18 15 13 12 11 10 7 7 7 5 4 2 2 0
0.15
0.10
0.05
Mean-Squared Error
-8
-6
-4
-2
log(Lambda)


Variable Importance: GBM
OverallQual
totalsqft
GrLivArea
OverallCond
GarageCars
YearRemodAdd
BsmtFinSF1
CentralAirY
GarageArea
YearBuilt
LotArea
GarageYrBlt
BsmtUnfSF
X1stFlrSF
TotalBsmtSF
KitchenQualTA
LotFrontage
MSZoningRM
SaleConditionNormal
NeighborhoodIDOTRR
0.0
0.2
0.4
0.6
0.8
1.0


Variable Importance: DRF
totalsqft
OverallQual
GrLivArea
YearBuilt
FullBath
ExterQualTA
GarageCars
X1stFlrSF
GarageArea
CentralAirY
YearRemodAdd
BsmtFinSF1
TotalBsmtSF
GarageYrBlt
OverallCond
LotArea
LotFrontage
BsmtUnfSF
TotRmsAbvGrd
FireplaceQuNone
0.0
0.2
0.4
0.6
0.8
1.0

Statement of Contributions

Sumedh R. Sankhe – Independent data processing, Simple Linear Modelling, Mapping & Plotting graphs & Latex.
Shivayogi Biradar – Independent data processing, Gradient Boosting, Elastic Net Regression, Data Modeling.
Sharyu Deshmukh – Independent data processing