Data Analytics Term Project

HOUSE PRICES: ADVANCED REGRESSION TECHNIQUES

(Using R)

Submitted By:

Ankur Joshi

RIN: 661883372

CONTENTS

Introduction

I have chosen to work with housing data from Ames, Iowa to predict the house prices based on 79 explanatory variables. As regression is the most widely used model in the industry, therefore I wanted to have a deeper understanding of regression model by implementing an end-to-end solution. This is the quintessential regression problem because the target variable is the sale price of a house, which is a continuous variable. Therefore, I decided to work on this problem.

I also got curious about this because recently I am in process of buying a house for myself back in India. It would be interesting to see which attributes impact the property price the most. We have some intuition regarding which variables affect the house prices but it's always safer to base our expectations on the hard facts derived from the data. I expect to find the usual suspects like property size, number of stories, type of property, locality etc. to have most impact on the price. Let's see, if our intuitions are in line with the data.

Data Description:

This data is taken from one of the Kaggle competitions. The data was original compiled by Dean De Cock for use in data science education. This dataset contains 79 explanatory variables for ~1,500 properties in Ames, Iowa along with their actual house price. The Ames Housing dataset is an incredible alternative for data scientists looking for a modernized and expanded version of the often-cited Boston Housing dataset.

Here is the list of the variables in this dataset:

50 categorical columns:

'MSSubClass'	'MSZoning'	'Street'	'Alley'	'LotShape'
'LandContour'	'Utilities'	'LotConfig'	'LandSlope'	'Neighborhood'
'Condition1'	'Condition2'	'BldgType'	'HouseStyle'	'OverallQual'
'OverallCond'	'RoofStyle'	'RoofMatl'	'Exterior1st'	'Exterior2nd'
'MasVnrType'	'ExterQual'	'ExterCond'	'Foundation'	'BsmtQual'
'BsmtCond'	'BsmtExposure'	'BsmtFinType1'	'BsmtFinType2'	'Heating'
'HeatingQC'	'CentralAir'	'Electrical'	'BsmtFullBath'	'BsmtHalfBath'
'KitchenAbvGr'	'KitchenQual'	'Functional'	'Fireplaces'	'FireplaceQu'
'GarageType'	'GarageFinish'	'GarageQual'	'GarageCond'	'PavedDrive'
'PoolQC'	'Fence'	'MiscFeature'	'SaleType'	'SaleCondition'

29 Numeric columns:

'TotRmsAbvGrd'	'GrLivArea'	'LotFrontage'	'LotArea'	'YearBuilt'
'YearRemodAdd'	'BsmtUnfSF'	'TotalBsmtSF'	'BsmtFinSF1'	'BsmtFinSF2'
'FullBath'	'HalfBath'	'X1stFlrSF'	'GarageArea'	'WoodDeckSF'
'BedroomAbvGr'	'GarageYrBlt'	'GarageCars'	'MoSold'	'YrSold'
'ScreenPorch'	'PoolArea'	'MiscVal'	'OpenPorchSF'	'MasVnrArea'
'X2ndFlrSF'	'LowQualFinSF'	'EnclosedPorch'	'X3SsnPorch'	'SalePrice')

The column names are intuitive of their meaning.

The data is broken into train and test set with ~1,100 and 340 records respectively. Train dataset was used to train the model and test dataset to test the accuracy of prediction.

Since it is a regression model, root mean square (RMSE) accuracy of predicted prices was used to assess the accuracy of prediction. RMSE calculates the average difference between the predicted and actual value of the target variable.

Out of 79 explanatory variables, 29 are numeric variables, 50 are categorical variables. All the categorical variables were converted into factor variables.

Analysis:

First step to any data science project is to deep dive into the data to understand the following:

1. Presence of Null values and outliers in the data because outliers have strong impact on the regression model results
2. The underlying distributions
3. Any possible correlations amongst various variables

To detect the presence of any null values in the data, summary command was run on the complete dataset (1,500 records). The summary gives various statistics about the distribution of data in each variables. For a numeric column it gives the mean, median, 1^{st}, 2^{nd}, 3^{rd} quartile, maximum value etc., and for categorical variables, it gives a frequency distribution of various levels.

From summary command, following variables were found to have null values:

Numerical columns with Null values:

1. GarageYrBlt
2. LotFrontage
3. MasVnrArea

These null values in the numeric columns were replaced with the median value of respective variables.

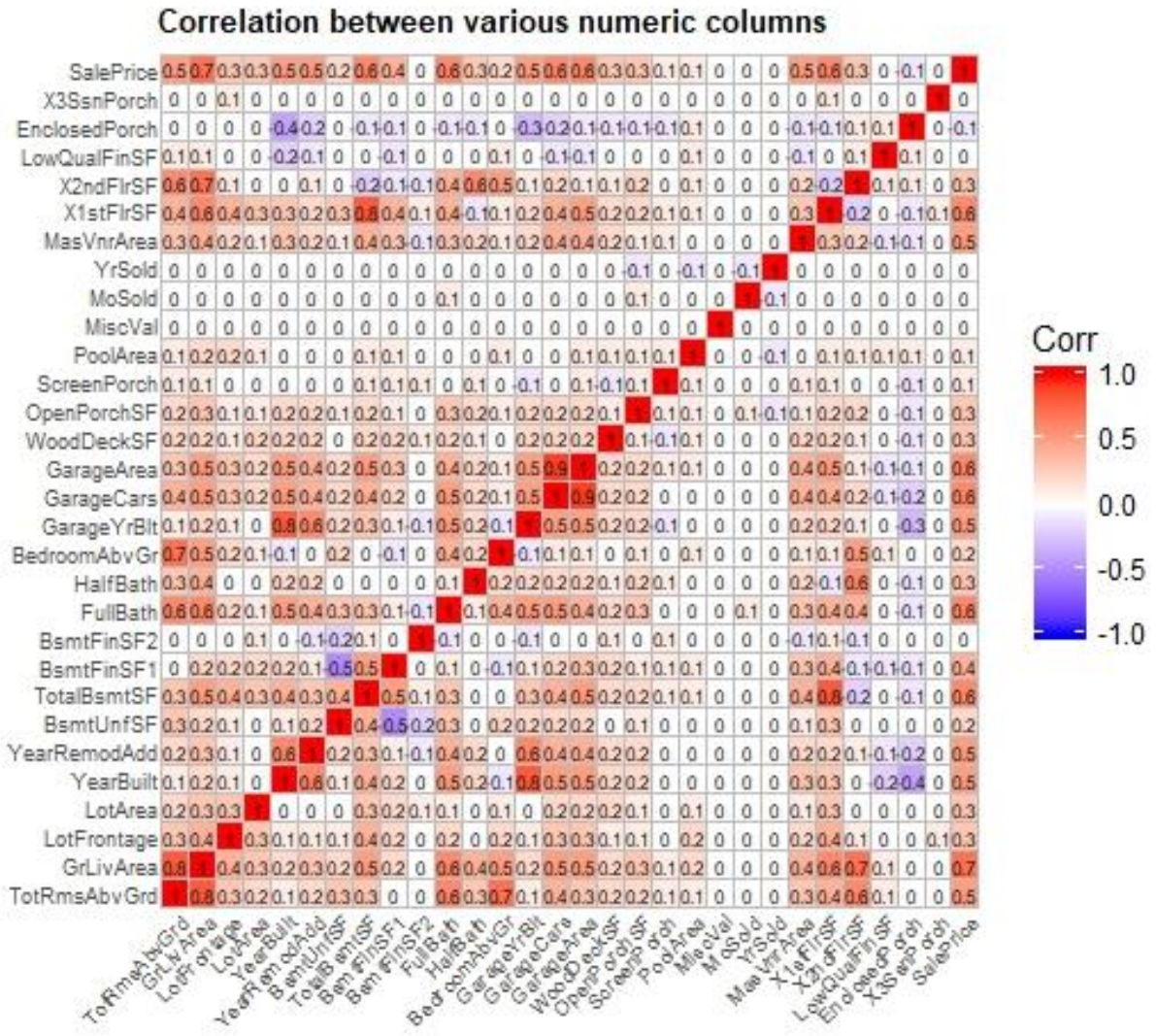
Categorical Columns with Null values:

Categorical Columns			
PoolQC	GarageQual	GaraYrBuilt	Alley
Fence	GarageCond	BsntFinType1	
MiscFeature	GarageCond	BsntFinType2	
GarageQual	GarageType	BsmtExposure	

These null values in the categorical columns were replaced with 'None'.

Checking for correlation of explanatory variables and the Sale Price:

As a next step, I wanted to check the correlation between all the numerical variables and house Sale Price. Here is a nice visualization of that:



From the chart above, we can infer that Sale Price is significantly correlated with following variables:

#	Variable2	Correlation Coefficient
6	GrLivArea	71%
9	GarageCars	64%
11	GarageArea	62%
14	TotalBsmtSF	61%
16	X1stFlrSF	61%
19	FullBath	56%
22	TotRmsAbvGrd	53%
23	YearBuilt	52%
26	YearRemodAdd	51%

Besides, there are many variables, which were significantly correlated with each other like:

#	Variable1	Variable2	Correlation Coefficient
1	TotRmsAbvGrd	TotRmsAbvGrd	100%
2	GarageArea	GarageCars	88%
3	GrLivArea	TotRmsAbvGrd	83%
4	X1stFlrSF	TotalBsmtSF	82%
5	GarageYrBlt	YearBuilt	78%
7	X2ndFlrSF	GrLivArea	69%
8	BedroomAbvGr	TotRmsAbvGrd	68%
10	FullBath	GrLivArea	63%
12	GarageYrBlt	YearRemodAdd	62%
13	X2ndFlrSF	TotRmsAbvGrd	62%
15	X2ndFlrSF	HalfBath	61%
17	YearRemodAdd	YearBuilt	59%
18	X1stFlrSF	GrLivArea	57%
20	FullBath	TotRmsAbvGrd	55%
21	GarageCars	YearBuilt	54%
24	BsmtFinSF1	TotalBsmtSF	52%
25	BedroomAbvGr	GrLivArea	52%
27	X2ndFlrSF	BedroomAbvGr	50%

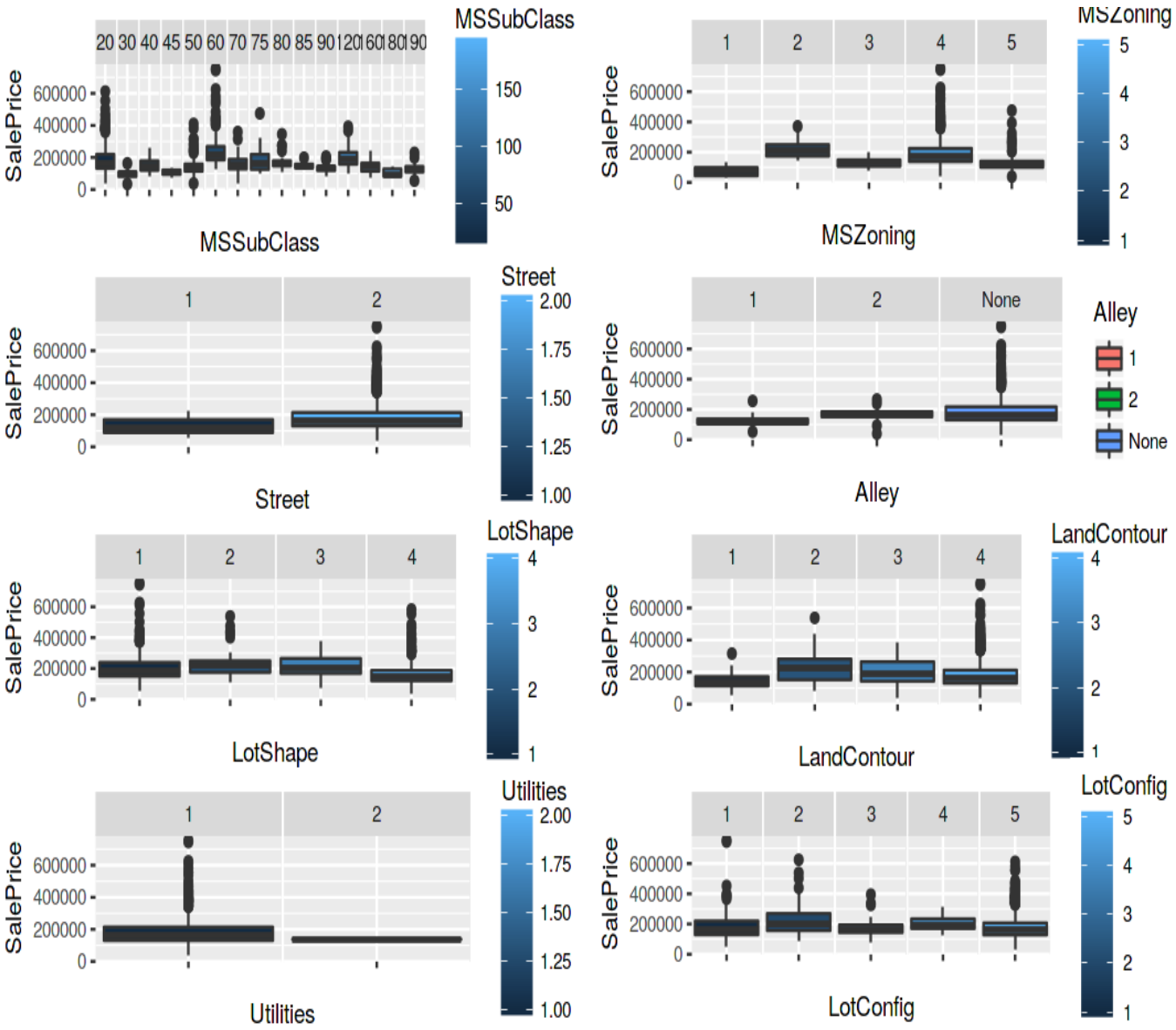
Only one of these highly correlated variables should be used in the regression model because highly correlated variables affect the overall accuracy of the model. These correlated variables were not dropped right away, but after the first run of the base linear regression model. Variables, which were found to have insignificant regression coefficients, were dropped from the analysis.

This gives us an intuition of what can we expect from our regression model.

I also plotted box plots of Sale Price for various levels in categorical columns. Following are some examples of box plots. Such plots were created for all the categorical variables. Box plots help us identify any trend that exists between various levels of categorical variables and the target variables. In this case – Sale Price.

If we see that the boxplots of Sale Price corresponding to each levels of a categorical variable have no variation in median and mean values, then we can infer that the Sale price is not dependent on that categorical columns. I f we see significant variation in Median and Mean, we can infer that that particular categorical variable has significant impact on the Sale Price.

Below are some example boxplots for few of the 50 categorical variables present in the data:

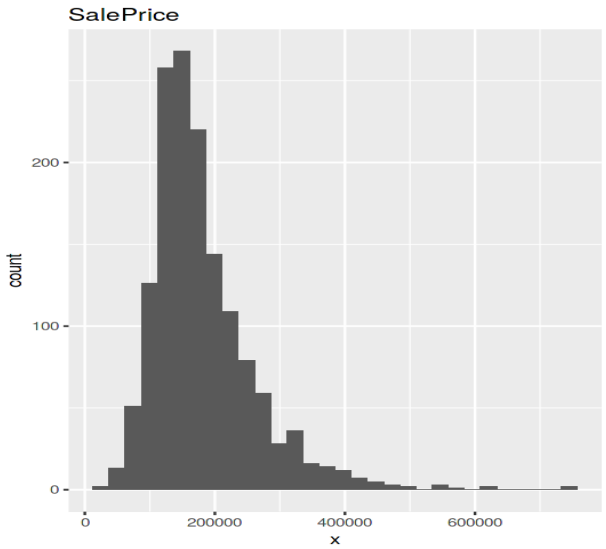


From the charts above it was clear that some categorical variables had significant impact on the house prices. Eg. if you look at the first chart for variable 'MSSubClass', there is

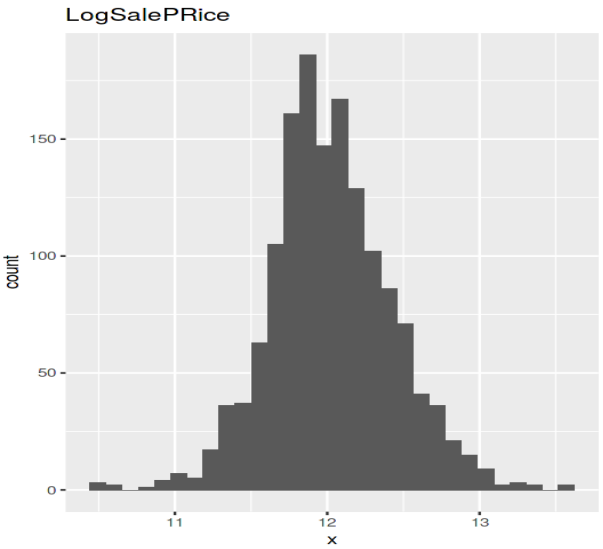
a significant variation in sales price for different levels of this category while Sale Prices are relatively stable across 'Utilities' variable.

Some variables which had skewed distributions (right or left), were transformed by taking log (1+feature) to make them normal. Distribution of these variables was checked after transformation. Charts above show the original and transformed distribution of the Target Variable 'SalePrice'.

For example, target variables SalePrice has a skewed distribution as shown below in the histogram. The skewness was removed after taking a log transformation. A normally distributed dependent variables gives better results with regression models

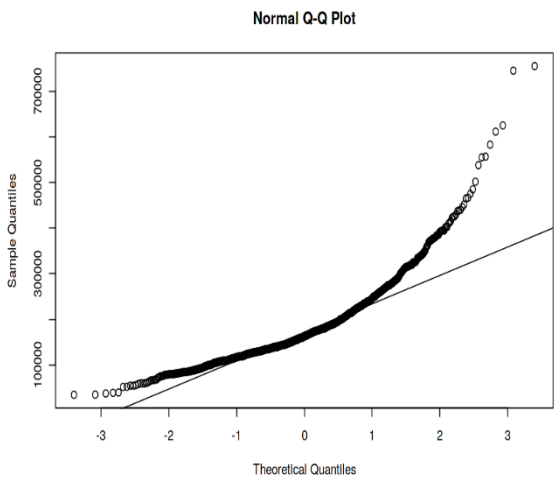


SalesPRice

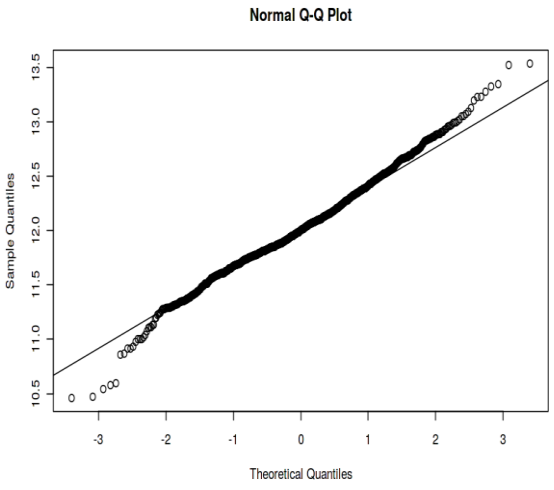


Log(SalesPRice)

We can also check for the normality of the distribution using quintile-quintile plot:



SalesPRice



Log(SalesPRice)

It is clear from the charts above that the distribution of the SalePrice became closer to normal after taking log transformation.

There were some variables which had numerical values but represented categorical values. Such variables were converted into factors. To do this, I converted all the variables except the numeric ones into factors. Now, even if any of these factor variables have numeric values, R would consider them as levels of a categorical variable.

Model Development and Application of model(s)

Training and Testing Sets:

There are total 1,460 records in the dataset, which were split into train_set (1,100 records) and test_set (340 records).

Following models were used to predict the house prices based on the explanatory variables available.

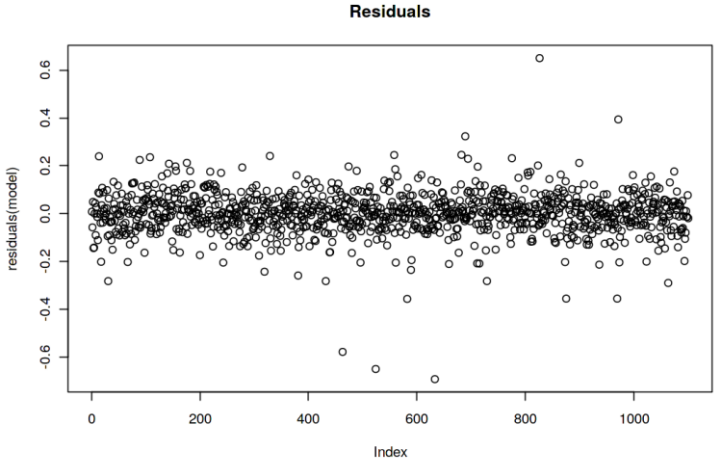
1. Linear Regression
2. Ridge Regression
3. Lasso Regression
4. XGBoost

Root Mean Square Error (RMSE) was used as an accuracy measure to compare the performance of different models. RMSE is the square root of sum of squares of difference between actual and the predicted house price. Here are the details of individual models:

1. Linear Regression Model:

Simple linear regression was the first model that was built. It acted as the baseline model to compare subsequent advanced models with. Linear regression model gave a good fit with R^2 value in range of 93% plus and overall p-value of fit ~0.

Below is the corresponding residual plot:



Residuals seem to have normally distributed, which is one of the requirements of a linear regression model.

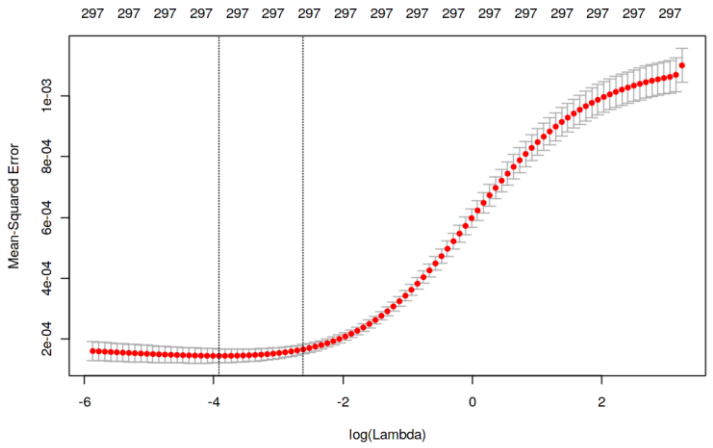
Overall **RMSE** of linear regression model = **101,625**

2. Ridge Regression:

As a next step, Ridge regression model used. It uses regularization of regression coefficients to avoid over fitting on training datasets. Following is the regression equation for ridge regression. Ridge regression introduces a parameter (λ) that introduces a penalty on the square of regression coefficient, and in turn avoids over fitting. This is a hyper-parameter, which is chosen by permutation and combination approach. R has a library named 'glmnet' which has inbuilt functionality to develop ridge regression model. It automatically finds out an optimum value of λ from a given range that maximizes the accuracy of prediction.

For this problem, λ ranged between 0.05 and 75 was used to get the optimal value from. The glmnet calculated the optimum value of λ to be 0.23.

Following chart shows the mean square error for different values of log(λ). Log plot is used because it can show a large range of values in a concise manner:



Optimal value of **λ ~ 0.26.**

The **RMSE** value for the optimum value of λ was **73,309**

3. Lasso Regression:

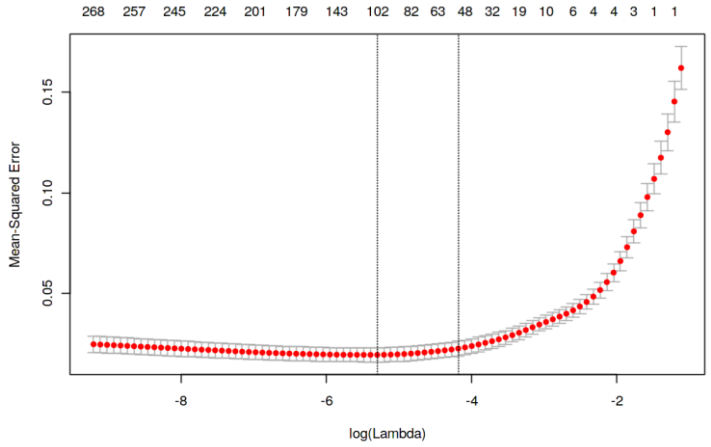
Another form of regression model is Lasso. It is in line with ridge regression but with small twist. It introduces a parameter (λ) that introduces a penalty on the absolute value of regression coefficients rather that the square of regression coefficient used in ridge regression.

This is a hyper-parameter which is chosen by permutation and combination approach. R has a library named 'glmnet' which has inbuilt functionality to develop ridge regression model. It automatically finds out an optimum value of λ from a given range that maximizes the accuracy of prediction.

For this problem, λ range between 0.05 and 75 was used to get the optimal value from. The glmnet calculated the optimum value of λ to be 0.0045.

The RMSE value for the optimum value of λ was 105,445.

Following chart shows the mean square error for different values of λ:



Optimal value of **λ ~ 0.018**

Lasso regression model (**RMSE = 105,445**) has better accuracy compared to Ridge regression model (**RMSE=73,309).**

4. XGB Model:

XGB Model or Extreme gradient boosted model is an ensemble of many weaker decision tree model which are weak predictors individually, but when combined together, they yield a better prediction accuracy. There are many hyper parameters for XGB model like step size, maximum number of variables to be included in individual decision trees, maximum depth of DTs, sample size etc. I played around with various combinations of these hyper-parameters to find the optimal combination.

The RMSE accuracy from XGB model was the best = 28,818.

This is not surprising because XGB model is the "rockstart" of classification model and has consistently outperformed other models in many data science competitions. This model utilizes the power of democracy to come up with the best predictions.

Conclusions & Discussion

Here is the summary of results from the 3 models used:

Comparison Results:

Model	Train Records (75%)	Test Records (25%)	RMSE
Linear Regression	1,100	340	105,445
Ridge Regression	1,100	340	73,309
Lasso Regression	1,100	340	101,626
XGB Model	1,100	340	28,818

It is evident from the above table that XGB model is the best performing model for this regression task. It is significantly better than the other regression models. Amongst regression models, Ridge regression has a better performance than the models built on regular linear regression and Lasso regression models.

One of the other way we can try to further improve our model is to change the way we handle some of the ordinal categorical columns. Right now, I have converted all the nominal and ordinal variables into factors, which means there is no order to different levels of these variables after converting them to factors. Instead, if we could encode these variables into numeric codes, which reflect their relative order we could hope to, improve the performance of the models.

There is also a possibility to transform some numerical variables to make their distribution more normal. Sometimes such transformations improve the accuracy of the models.

There is also a possibility to use Principle Component Analysis (PCA) to reduce the dimensionality of the problem.

Appendix:

Here is a link to R notebook on my GitHub repository:

https://github.com/joshia4/Data-Analytics-Term-Project