**OPIM – 5604 – PREDICTIVE MODELING**



**PREDICTING RUSSIAN HOUSE PRICES**

**Project White Paper**

**Aditya Nagori**

**Arijit Sinha**

**Rishitha Gajjala**

**Sai Gayatri Narne**

Table of Contents

[INTRODUCTION 3](#_Toc512634050)

[PROBLEM STATEMENT 3](#_Toc512634051)

[EXPLORE 4](#_Toc512634052)

[INSIGHTS INTO THE DATA 4](#_Toc512634053)

[DATA PRE-PROCESSING 6](#_Toc512634054)

[DATA CLEANING 6](#_Toc512634055)

[OUTLIER ANALYSIS 7](#_Toc512634056)

[DATA MODIFICATION 7](#_Toc512634057)

[DATA NORMALIZATION AND SCALING 8](#_Toc512634058)

[DATA REDUCTION 8](#_Toc512634059)

[Variable Inflation Factor (VIF) 8](#_Toc512634060)

[MODELLING 8](#_Toc512634061)

[MULTIPLE LINEAR REGRESSION 9](#_Toc512634062)

[Checking Assumptions of a Regression Model 10](#_Toc512634063)

[Adding Interaction Terms 10](#_Toc512634064)

[LASSO (LEAST ABSOLUTE SHRINKAGE AND SELECTION OPERATOR) 10](#_Toc512634065)

[PRINCIPAL COMPONENT ANALYSIS (PCA) 11](#_Toc512634066)

[DECISION TREE 12](#_Toc512634067)

[RANDOM FOREST 12](#_Toc512634068)

[BOOSTED TREE 13](#_Toc512634069)

[SUPPORT VECTOR MACHINE 14](#_Toc512634070)

[MODEL COMPARISON 14](#_Toc512634071)

[LEARNING CURVE 15](#_Toc512634073)

[INSIGHTS 16](#_Toc512634074)

[ENHANCEMENTS/ FUTURE WORK 17](#_Toc512634075)

[REFERENCES 17](#_Toc512634076)

# INTRODUCTION

**Domain Background**

Regression analysis is a form of predictive modeling which investigates the relationship between variables. More specifically, regression analysis helps one understand how the typical value of the dependent variable changes when any one of the independent variables is varied, while the other independent variables are held fixed.

It basically answers these questions: Which factors matter the most? Which predictors can we ignore? How do those factors interact with each other? And, perhaps most importantly, how certain are we about these factors and their predictions? The main factor that we’re trying to predict is a target (a dependent variable). The features (independent variables) are the factors supposed to have an impact on the dependent variable. Using this set of variables, we generate a function that maps inputs to outputs. The training process continues until the model achieves the desired level of accuracy. The project investigates supervised learning as a part of regression analysis that uses a known (training) dataset to make predictions. This dataset includes input data and response values. The supervised learning algorithms seek to build models which make predictions of the response values for a new dataset. A test dataset is used to validate the model.

# PROBLEM STATEMENT

**Sberbank Russian housing market**

[Sberbank](https://www.kaggle.com/sberbank), Russia’s oldest and largest bank, helps their customers by making predictions about realty prices so renters, developers, and lenders are more confident and well informed when they sign a lease or purchase a building. Sberbank provided a rich data set that included housing data (a total of 30000 observations and 292 variables)

Sberbank is the stakeholder involved and would be highly affected by the new sales made in the market.   
The bank profit indirectly depends on these right predictions and how many customers agree with these real time predictions. Prediction of prices is an ongoing process since the factors influencing the price keeps changing.

**DATASET and INPUTS**

The set of predictors includes internal housing characteristics collection of features about each property’s surrounding neighborhood and some features that are constant across each sub -area.

This data set contains around twelve predictors which convey the internal housing features like Total are, floors, year built etc. Around 279 predictors convey the supporting information of surroundings, neighborhood and general attributes of the area like population, income group etc. After going through the data dictionary to understand what each predictor implies, we have eliminated some columns which don’t directly contribute to the house price. This step has eased the further steps in modeling.

The input file name is train.csv. We have partitioned the data in training and test using the same set. The data dictionary file is also available with name data\_dictionary.txt.

# EXPLORE

Exploratory Data Analysis (EDA) is the first step in the data analysis process.

Statistical graphics is a collection of techniques--all graphically based and all focusing on one data characterization aspect.

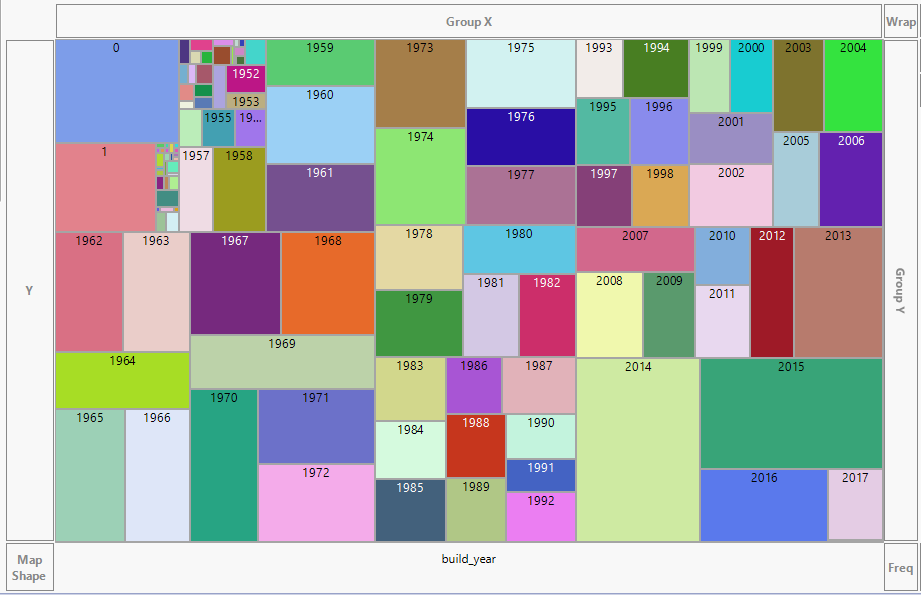
EDA encompasses a larger venue; EDA is an approach to data analysis that postpones the usual assumptions about what kind of model the data follow with the more direct approach of allowing the data itself to reveal its underlying structure and model

Exploratory Data Analysis (EDA) employs a variety of techniques (mostly graphical) to

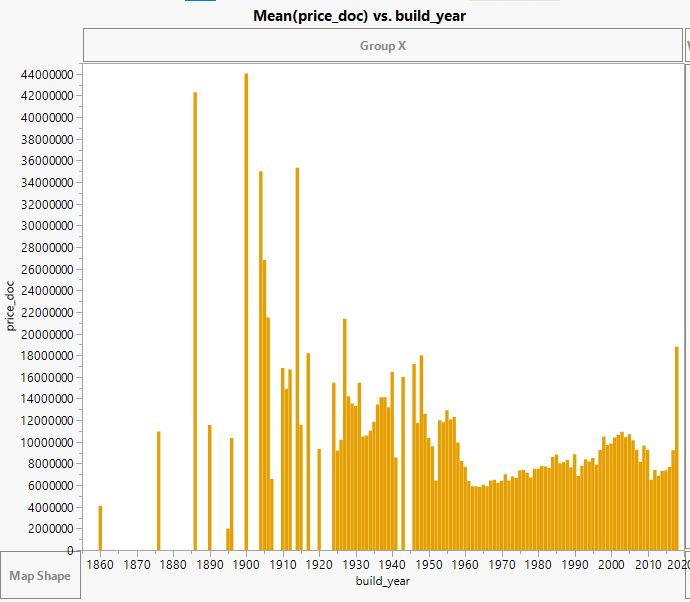
* Maximize insight into a data set;
* Uncover underlying structure;
* Extract important variables;
* Detect outliers and anomalies;
* Test underlying assumptions;
* Develop parsimonious models; and
* Determine optimal factor settings.

We have incorporated the graphical representations using JMP and RStudio tools.

## INSIGHTS INTO THE DATA

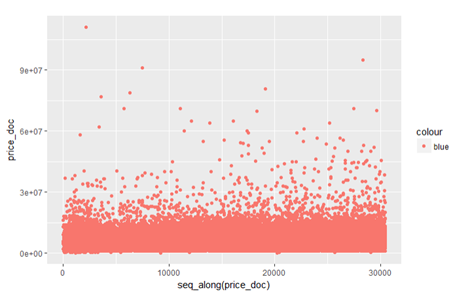
* **Build year distribution**

This is the frequency distribution of build year across all the observations. The data is spread across many years i.e. 1860 to 2018. We could also observe some wrong entries in build year like 0,1,3 etc. There are many null values for this column too. As per the domain understanding, build year is an important factor which influences the pricing of the house.

* **Mean (price\_doc) vs. build\_year**

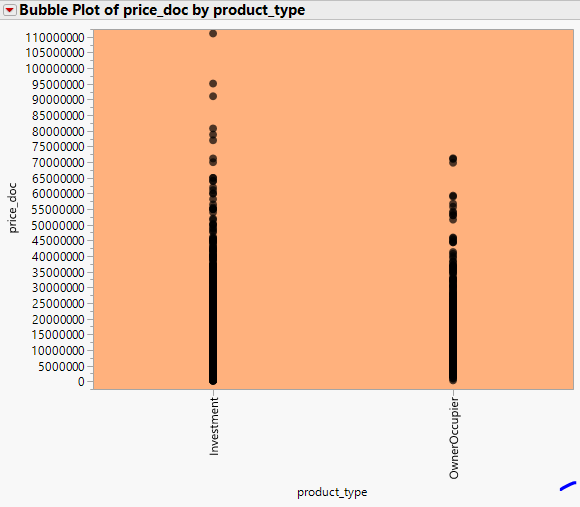
The graph besides shows how the average house prices vary with respect to the year in which the houses are built. In the given data the houses are as old as 100 years. The build year ranges from approximately 1915 to the year 2015. It is observed that the data is less concentrated on the houses built between the years 1915 and 1955. However, the average price of the same houses seems to be very high when compared to the houses built recently in the past 50 years (i.e. from 1955 to 2015). It is also observed that amongst the houses built in the past 50 years, those built between the years 1995 to 1960 and 2000 to 2003 seem to have seem to have relatively higher average prices than the other houses built in the same period.

* **Target variable -Price Disribution**



The below graph shows how price\_doc, target variable distribution in the observations. This shows that we have uniformly distributed data set across the price ranges. There are few outliers in the price which could be imputed.

* **Price vs Product Type**



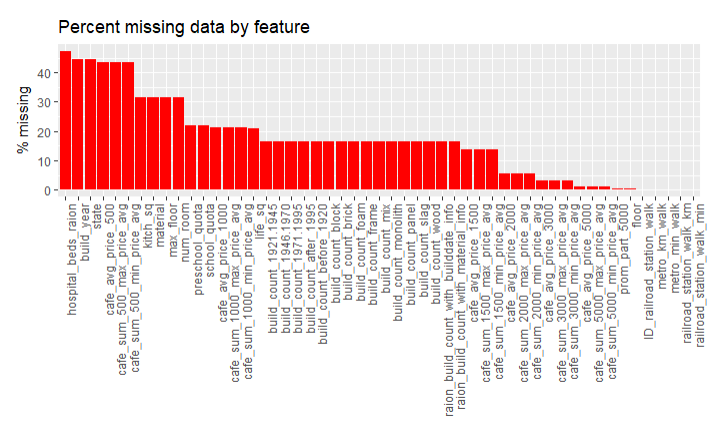
Distribution of prices for homes bought by an owner-occupier or homes bought for investment? It's clear from the below graph that homes sold for investment sell more than homes sold to owner-occupiers.

## DATA PRE-PROCESSING

Our dataset went through a series of steps during preprocessing. The sub steps under this are mentioned below in four sections

## DATA CLEANING

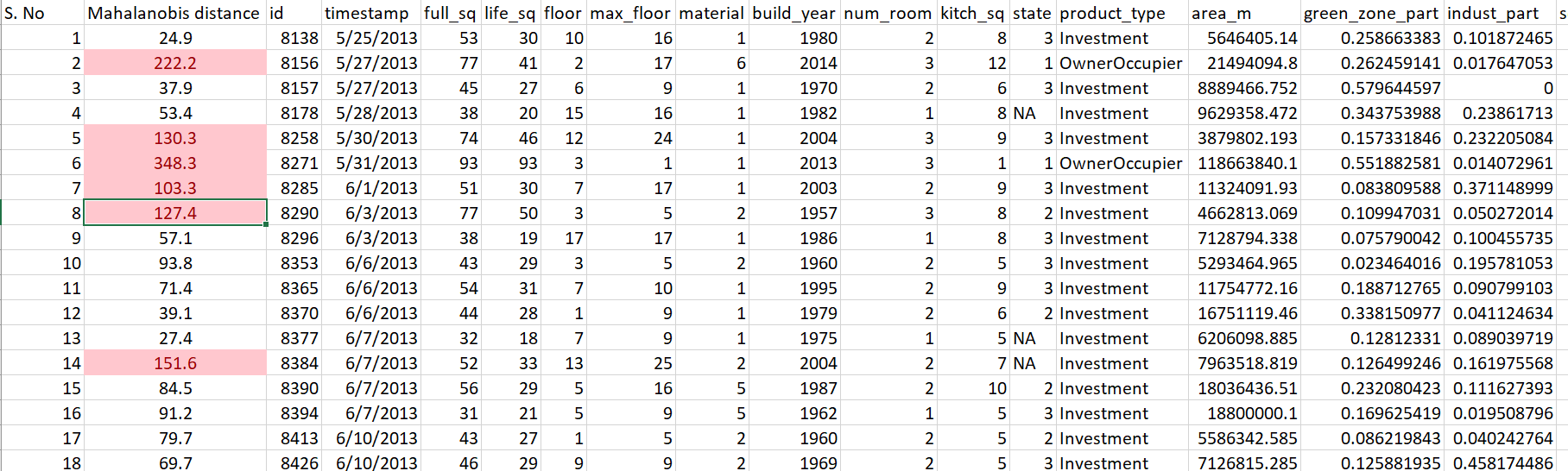
* Studying the data keenly, we understood that there were few inconsistencies in the data. We removed some observations and brought the number of rows to nearly down to nearly ~16K. Approach took:
  + For many columns such as build year, hospital beds raion etc. many rows have NA values as shown in the missing value percentages below
  + Imputed some inconsistent values in columns such as State, Build year
  + Calculated the Mahalanobis distance for each observation and removed the observations with z scores >3 and <-3



* State variable has mainly 4 levels. There was an additional number which we have removed considering it as wrong data entry
* Build year also has a lot of unmeaningful years
* The feature full\_sq is defined in the data dictionary as 'total area in square meters, including loggias, balconies and other non-residential areas' and the life\_sq is defined as 'living area in square meters, excluding loggias, balconies and other non-residential areas.' So, it should be the case that life\_sq is always less than full\_sq. There are 37 observations where life\_sq is greater than full\_sq

## OUTLIER ANALYSIS

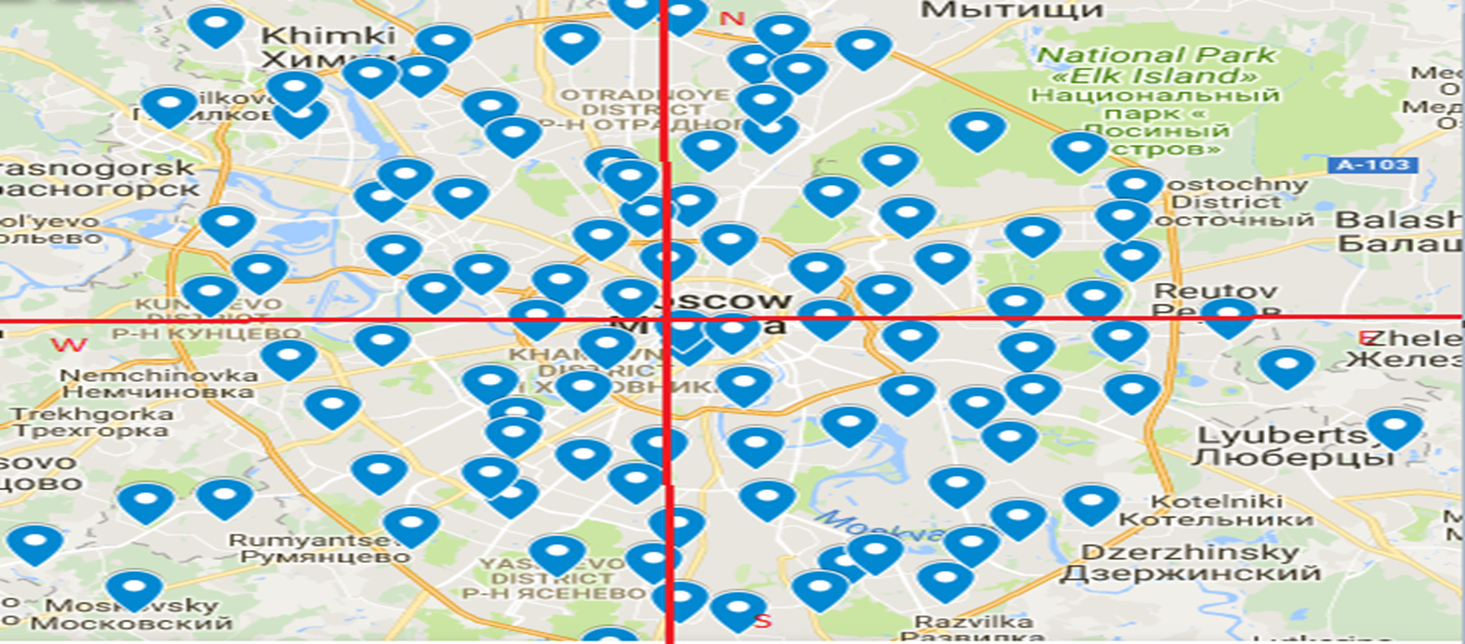
We have studied the outliers in the data using Mahanalobis distance which is an outlier detection method for multivariate outliers. It is the distance of a data point from the calculated centroid of the other cases where the centroid is calculated as the intersection of the mean of the variables being assessed. We found the z score values of the Mahanalobis distance for all the rows and removed the ones having z scores greater than 3 or less than -3. Below is a screenshot of the range of Mahanalobis distance values the rows were taking when arranged in decreasing order of distance.



There is a direct function in R named Mahalanobis which we used to find the distance

## DATA MODIFICATION

We have a variable named sub\_area in our original dataset which have ~150 unique names of Russia locations. Since, based on business understanding geography would be a very important factor in determining house prices, we manually searched all the locations on Google maps (screenshot below) and binned the sub area variable into 5 region categories ex. North, South, East, West, Centre

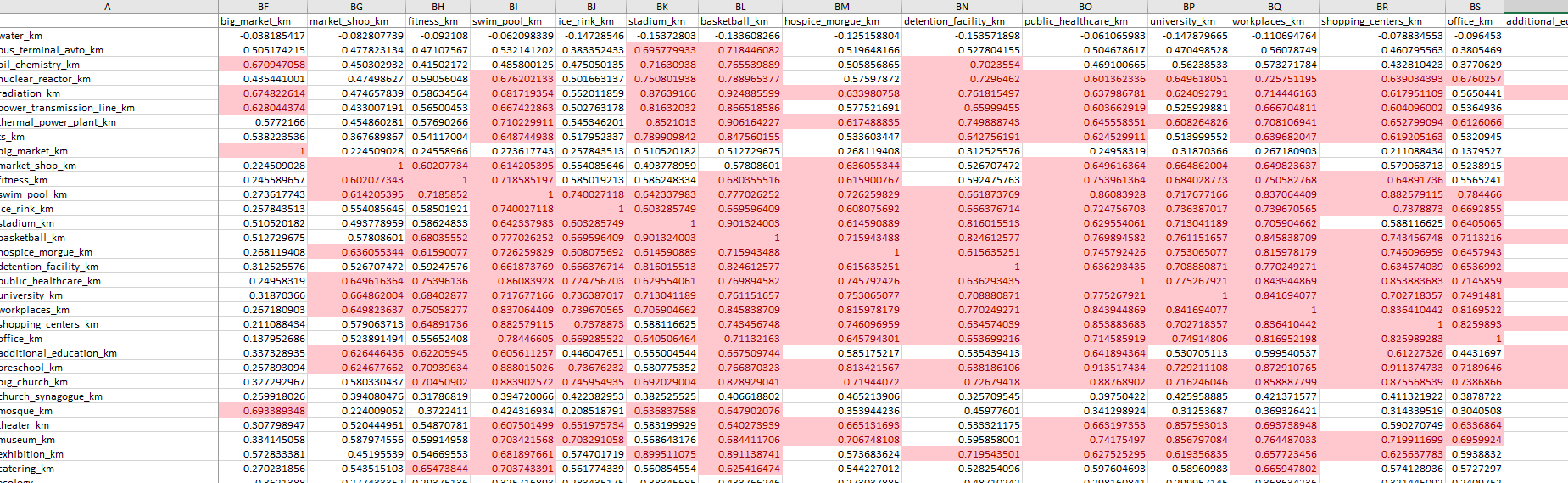


## DATA NORMALIZATION AND SCALING

We have scaled the target variable price by 100,000. Since most of the values are above that, we chose this scale for Price. For unsupervised learning techniques such as PCA or some of the advanced machine learning algorithms (ex. SVM), we normalized the independent variables to remove unwanted bias since these algorithms worked upon finding the Euclidean distance between data points and scaling of variables might skew the results

## DATA REDUCTION

We had ~290 variables in our initial dataset which are a lot of variables to work with. We have used multifold approach to reduce this number to a reasonable number. First and foremost, based on business understanding we eliminated ~50 variables which were very vague. Next, we found the correlation values between each variable. Below is a screenshot of a section of correlation matrix we created highlighting the ones which are having a correlation value of greater than 0.6 and less than - 0.6.



### Variable Inflation Factor (VIF)

We also looked at the VIF for all the independent variables and identified the ones which have this value higher than 5. This is an indicator which variables are causing multi collinearity in the model. It provides an index that measures how much the [variance](https://en.wikipedia.org/wiki/Variance) of an estimated regression coefficient is increased because of collinearity. We used the ‘usdm’ package in R to find the VIF values for each variable with respect to the other independent variables.

By using a combination of the above two parameters (correlation matrix and VIF values), p values obtained after running regression on the entire dataset in JMP initially we identified variables which are not contributing significantly in predicting the correct house prices due to correlation and collinearity with other independent variables and thus removed those variables from the dataset taking our final number of predictors in the dataset to 55.

# MODELLING

To predict house prices of the Russia houses we created multiple regression models with differing tuning parameters for each model as per the need. Finally, the best model with the lowest test set RMSE and highest test set R square is selected as the best model. We used the R programming language to make different machine learning algorithms

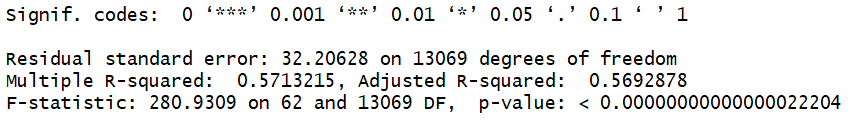
For linear models such as multiple linear regression, we divided our dataset into two parts, one is the training dataset used to train the regression model and the other is test set to check the accuracy of the models created. For non-linear models involving tuning parameters such as decision trees, random forest, boosted trees, support vector machines etc. we used the cross-validation approach with 3-fold cross validation. Initially, we tried higher number of folds (5) and more number of iterations (3) in cross validation to enhance model performance but the models were taking a long time to run due to system limitations and dataset complexity (large number of predictors and adding interaction terms above them). Hence finally we decided to stick with 3-fold approach. The training test split ratio was kept at 80:20 and kept this ratio constant throughout. Also, the test set is kept same by adding same seed in R throughout the model building algorithms to check the performance of different models on same test set enabling model comparisons. In case of cross validation, the training set is further split into training and validation set where the validation set is used to prune the models

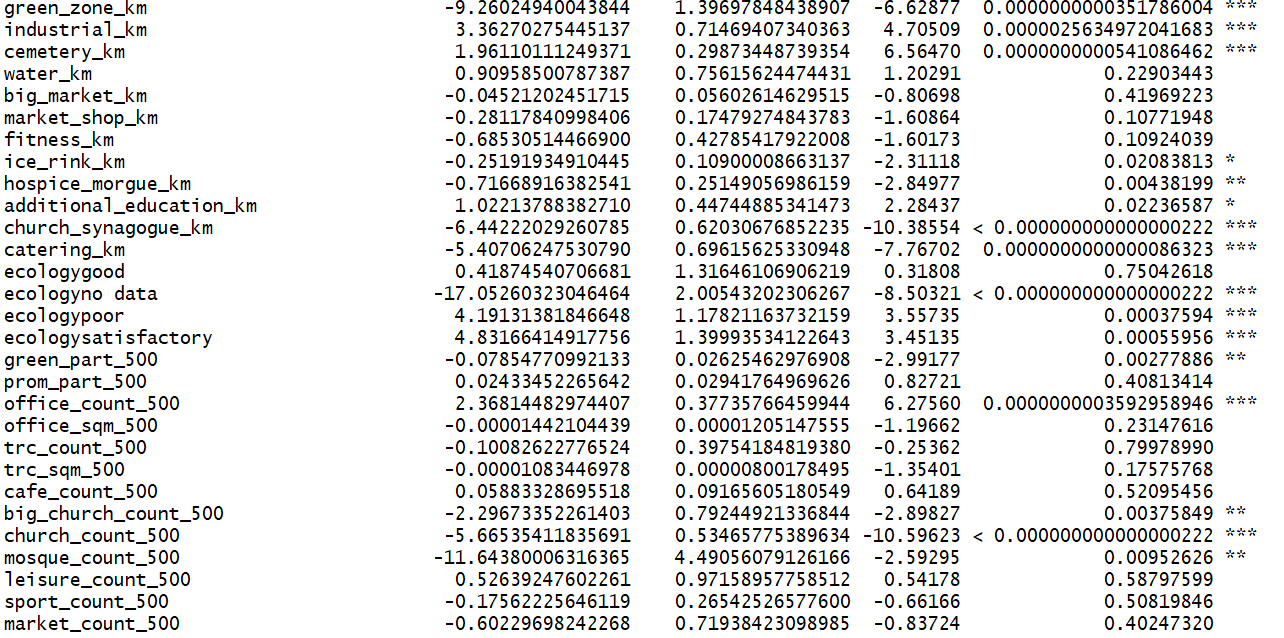
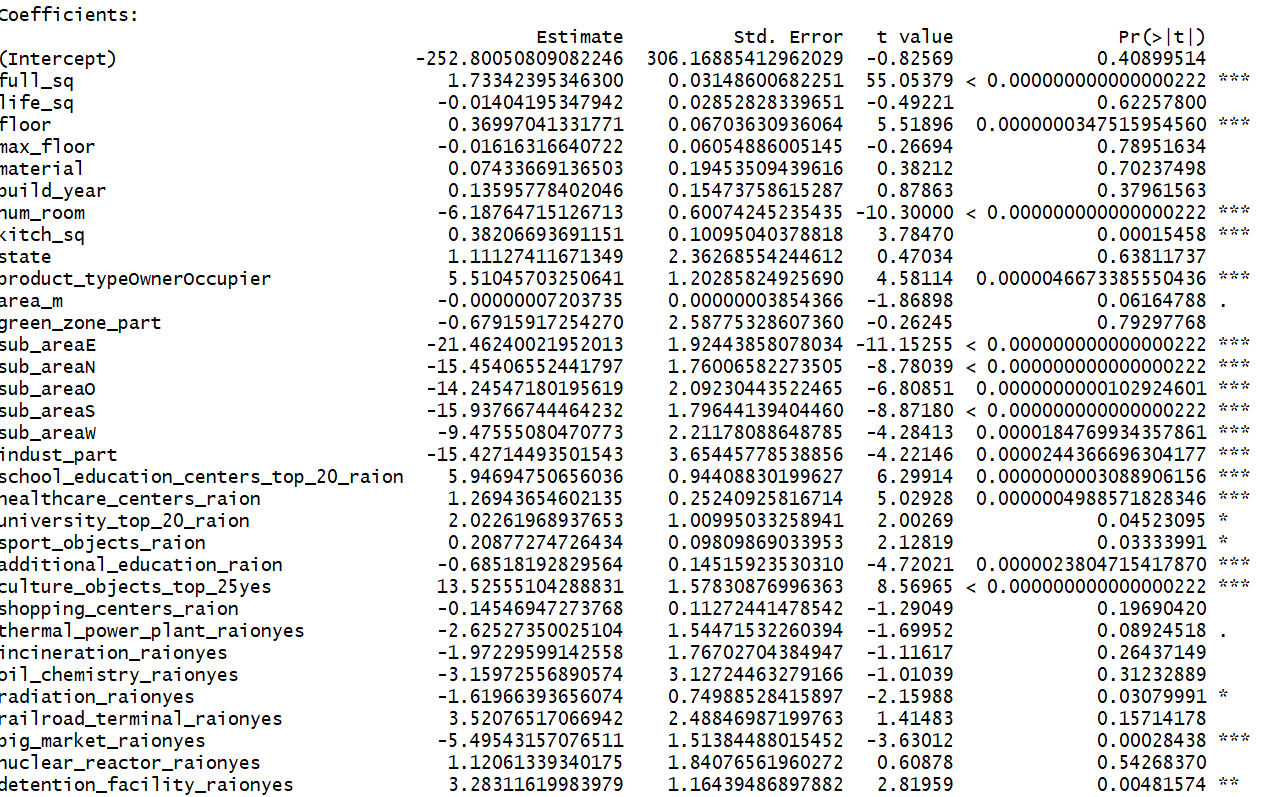
The following models were created in R to solve this regression problem:

* Multiple Linear regression (with and without interactions)
* Lasso
* Decision Tree
* Random Forest
* Boosting
* SVM

## MULTIPLE LINEAR REGRESSION

We started our model creation stage with linear regression model in R using the “lm” package and got the following statistics on the training set.



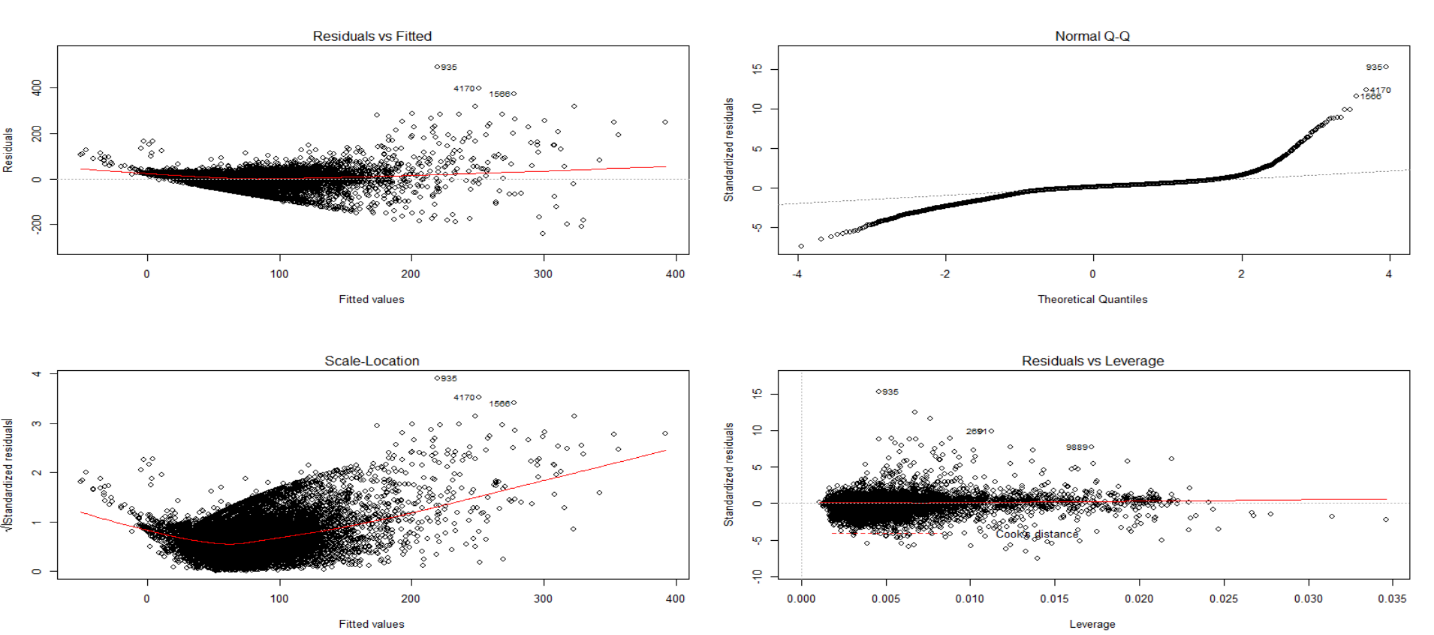


The fitness of our model in case of linear regression is not great as the Adjusted R square value comes to be around **~57%** while that on the test set is **~45.4%**. We should compare this with other non-linear models to conclude whether our independent variables have a good linear relationship with the dependent variable.

Another conclusion we can draw from this is the fact that since our R square and adjusted R square values are almost similar. There are not many predictors which are not at all significant otherwise the model would have penalized R square due to the presence of non-significant variables

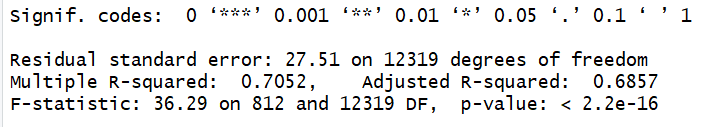
## Checking Assumptions of a Regression Model

R has an in-built regression diagnostic tool where one can find if your regression model is reasonably meeting the assumptions of a linear model. From the below graphs, we can interpret that model has signs of being heteroscedastic and the residuals are not perfectly normally distributed.



### Adding Interaction Terms

We added all the possible two-way interaction terms of all the numerical variables with all the categorical variables in our regression model in hope further improving the accuracy of our model. There were 12 categorical variables and remaining 43 numerical variables in our final dataset.

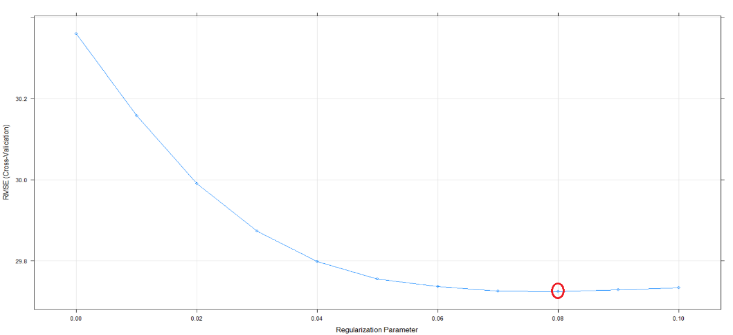
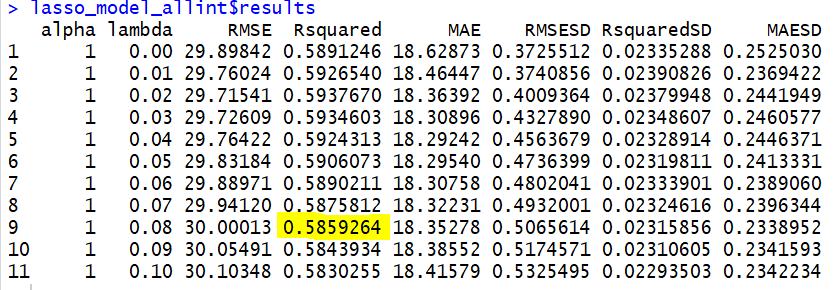


The results were quite interesting, the training R square value has improved significantly from ~57% in case of model without interactions to ~70% but the performance of the model on test set remains similar, indicating that model has become overfitted and suffering from the issue of high variance

### LASSO (LEAST ABSOLUTE SHRINKAGE AND SELECTION OPERATOR)

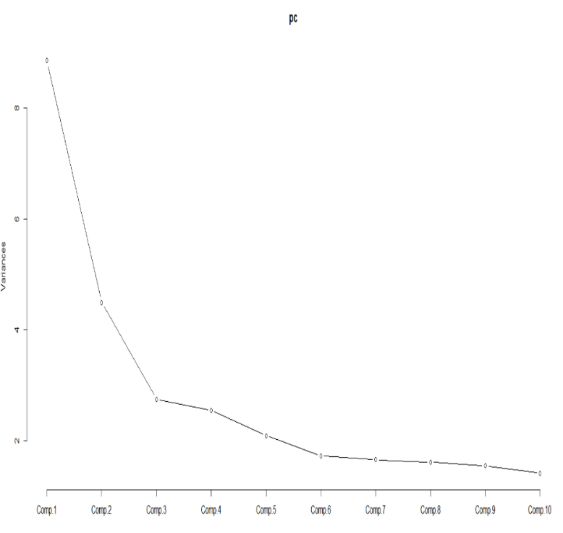
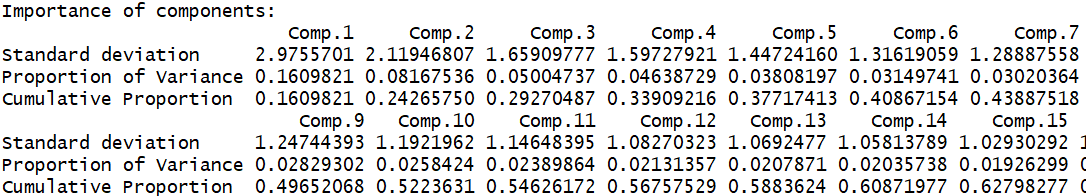
In order to overcome the problem of overfitting coming in case above model of linear regression with interactions we tried LASSO (least absolute shrinkage and selection operator). It is a [regression analysis](https://en.wikipedia.org/wiki/Regression_analysis) method that performs both [variable selection](https://en.wikipedia.org/wiki/Variable_selection) and [regularization](https://en.wikipedia.org/wiki/Regularization_(mathematics)) in order to enhance the prediction accuracy and interpretability of the [statistical model](https://en.wikipedia.org/wiki/Statistical_model) it produces. We used the “*glmnet”* package in R to perform LASSO.

Here the tuning parameter is lambda which automatically ‘de-emphasize’ variables that are not important. It will make the coefficients for unimportant variables very small, effectively suggesting we do know need to care too much about those predictors**.** This allows us to get a sense of what variables are more important than others**.** Following are the statistics we are getting upon running the lasso model.

****

The model performance has improved substantially from normal regression with R square value of 58% using cross validation. Also, the model is getting optimized at the tuning parameter value of 0.08.

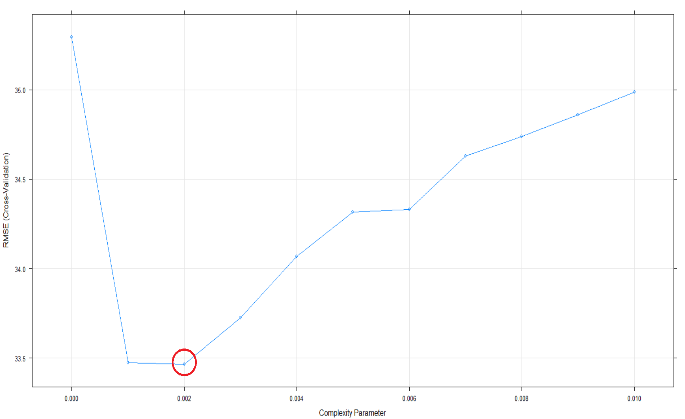
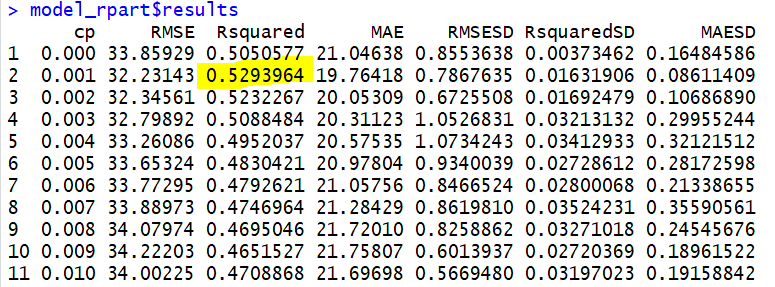
## PRINCIPAL COMPONENT ANALYSIS (PCA)



We tried the PCA technique to reduce the number of predictors by finding a low-dimensional representation of the dataset. As we can see from the scree plot above, after the 6th PCA, the other PCA’s does not contribute significantly in explaining dataset variance. Also, the cumulative variance explained by the first 6 PCA’s is only ~40% which is not significant enough. We still ran the multiple linear regression suing the first 10 PCA’s as independent variables and the results were quite similar to the earlier regression. Thus, we can say that the PCA technique is not very helpful for our dataset in model creation

## DECISION TREE

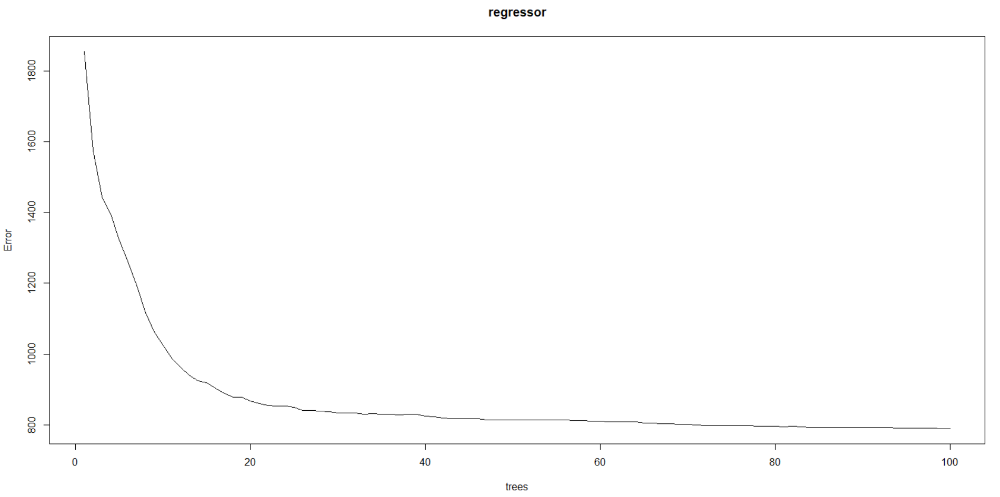
Next, we model a decision tree using cross validation via the caret library in R. This library has an in-built cost tuning parameter for which you can give a set of values for the cost tuning parameter, in case of tree this tuning parameter is cp, the complexity parameter which is used to control the size of the decision tree and selects the optimal tree size. If the cost of adding another variable to the decision tree from the current node is above the value of cp, then tree building does not continue. The model identified the best value of cp at which the model performance is optimum. We also included the significant interaction terms coming from regression as independent variables for all the non-linear model created. Below are a few screenshots of the optimum Rsquare value model gives on different value of cp and the optimum tuning parameter curve which minimizes the RMSE.

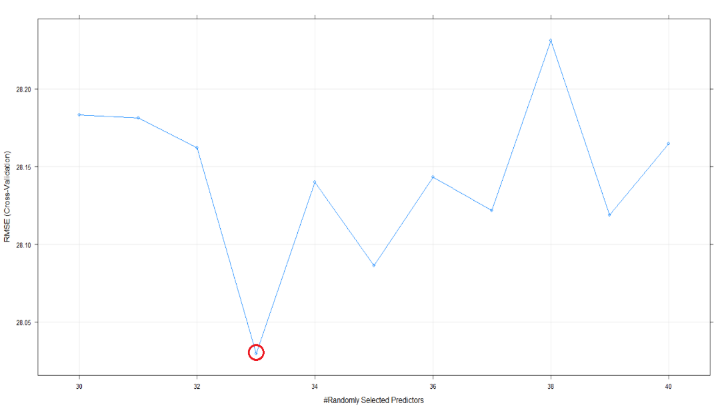
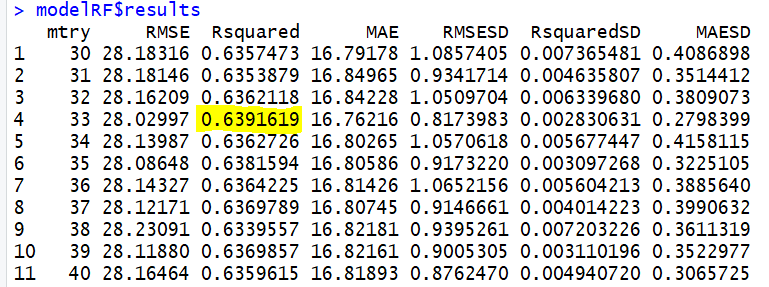


The R square value of the model which indicates the fitness of the model has improved from the linear regression model. This could be due to the fact our data has a mon linear shape and decision trees have done a better job at capturing the non-linearity in the data by dividing the space into smaller sub-spaces

## RANDOM FOREST

After decision trees the next obvious model we built is the random forest which is an aggregation of multiple trees and finally takes mean of the sub sections of all the trees. As per the screenshot below, after ~100 trees in the forest, the error is getting saturated and adding more trees would not improve the model performance. Hence, we used 100 trees in in our model and the tuning parameter changed was mtry. When forming each split in a tree, the algorithm selects mtry variables from the set of predictors available. Hence when forming each split a different random set of variables is selected within which the best split point is chosen to find the best performing forest whose specifications are mentioned below:

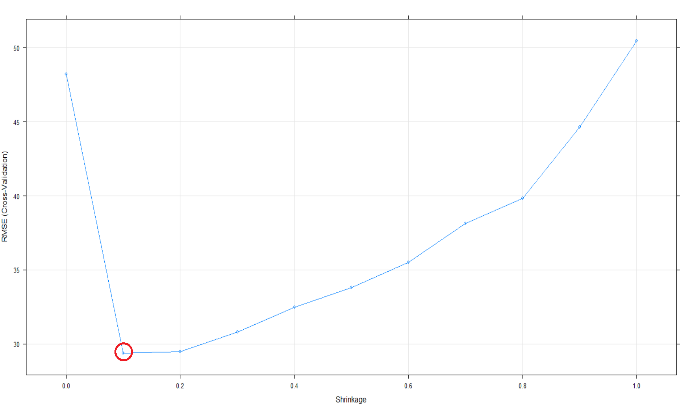
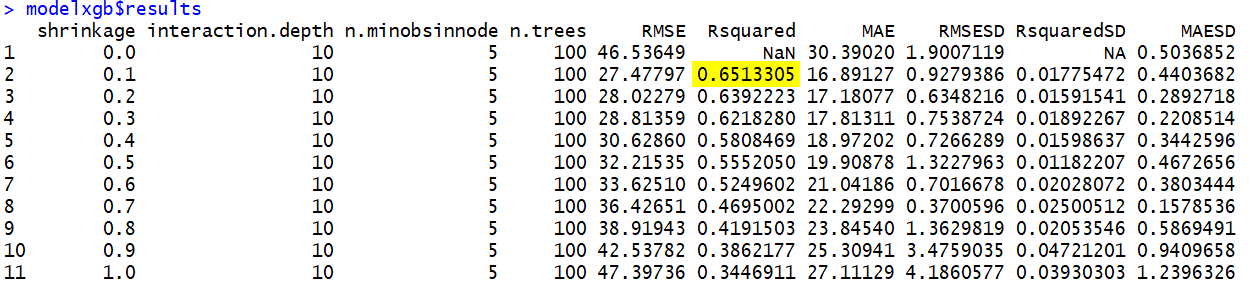




As expected, the R square value has improved from ~53% in case of decision tree to ~64% and the sweet spot of mtry is 33 at which the model performance is getting optimized.

## BOOSTED TREE

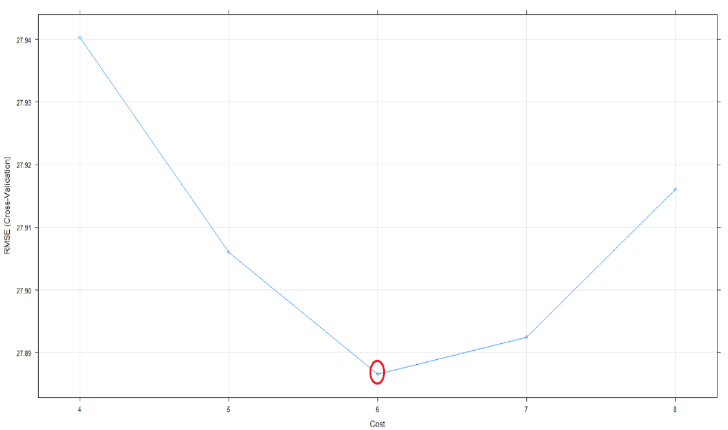
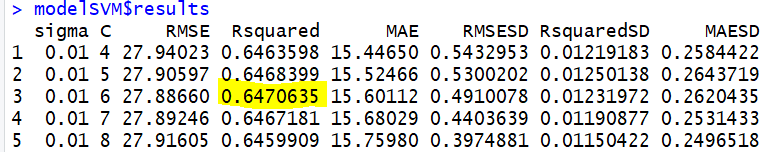
Following the random forest model, we created a boosted tree model to predict the house prices. Since boosting generate trees in a sequential manner, one following the other, based on the residuals of the preceding tree, we were hoping to see an improvement in performance when compared with previous model. We created multiple boosted trees keeping the other parameters such as number of trees = 100, interaction depth = 10, maximum number of splits at each node (n.minobsinnode) = 5 and vary the learning rate using it as a tuning parameter. The shrinkage parameter is used to control the rate of the growth of trees avoiding overfitting in the process. It basically downplays the importance of each tree in the step thus we reach to our final conclusion slowly. We noticed that as the learning rate increased beyond 0.1 the performance of the model starts degrading. Following results were obtained upon running the gradient boosting algorithm.



The performance of the boosted tree is at par with Random Forest in predicting the house prices

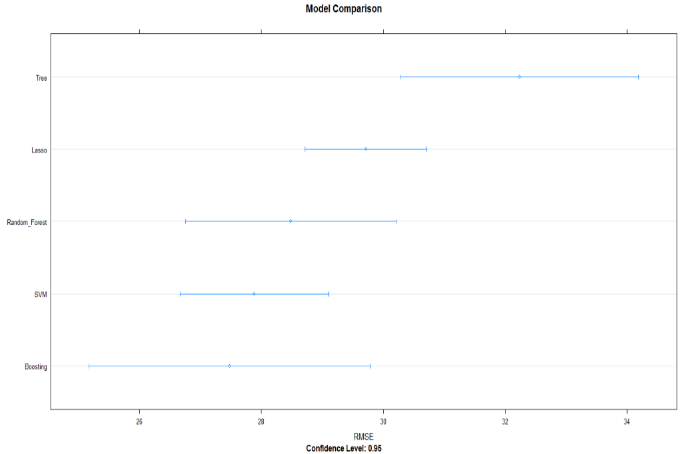
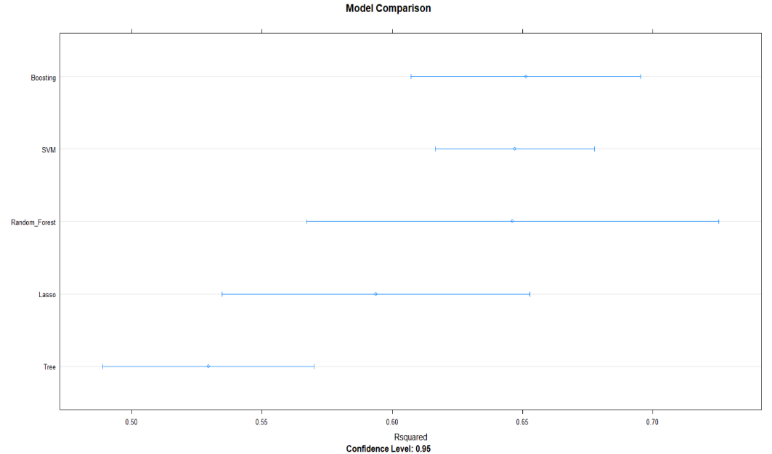
## SUPPORT VECTOR MACHINE

At last, we tried the widely used support vector machine algorithm. Used the radial basis function as a kernel to find the linearly separable hyperplane in higher dimensions. The cost parameter which decides the margin width is used as a tuning parameter and multiple models are created by varying C. The results are shown below:

The optimum model performance is coming at C=6 and the performance of the model is comparable with Random Forest and Boosting

## MODEL COMPARISON

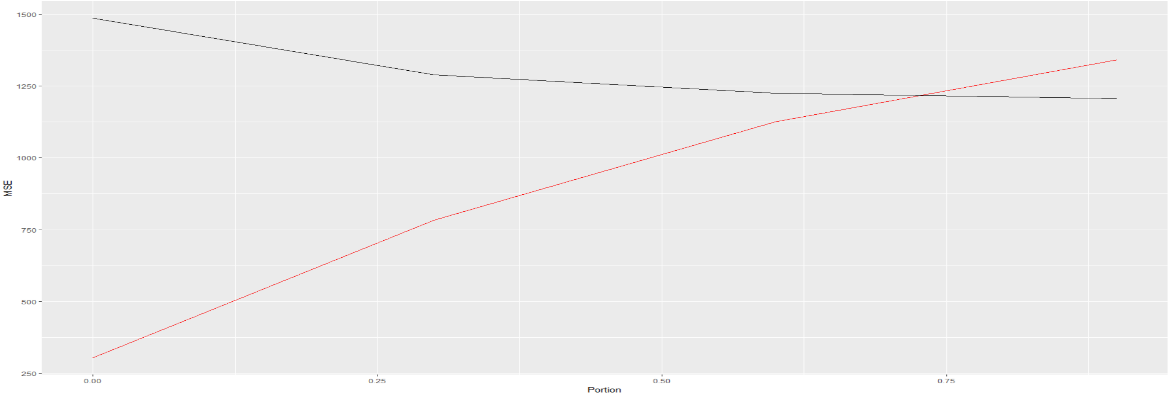


Above is a picture of comparison of all the nonlinear models we created including LASSO. As we can see the R square value is highest for Random Forest and boosting and we can’t really say which one among them is doing a better job followed by SVM.

|  |  |
| --- | --- |
| **Model** | **R Squared (Test Set)** |
| Linear Regression | ~42% |
| LASSO | ~61% |
| Decision Tree | ~54% |
| **Random Forest** | **~65.5%** |
| **Boosting** | **~64.5%** |
| SVM | ~63% |

# The above table shows the performance of models on test set but since in case of cross validation there is not much difference in the performance of model on training and test set we are getting similar results with Random Forest and Boosting emerging as the top models

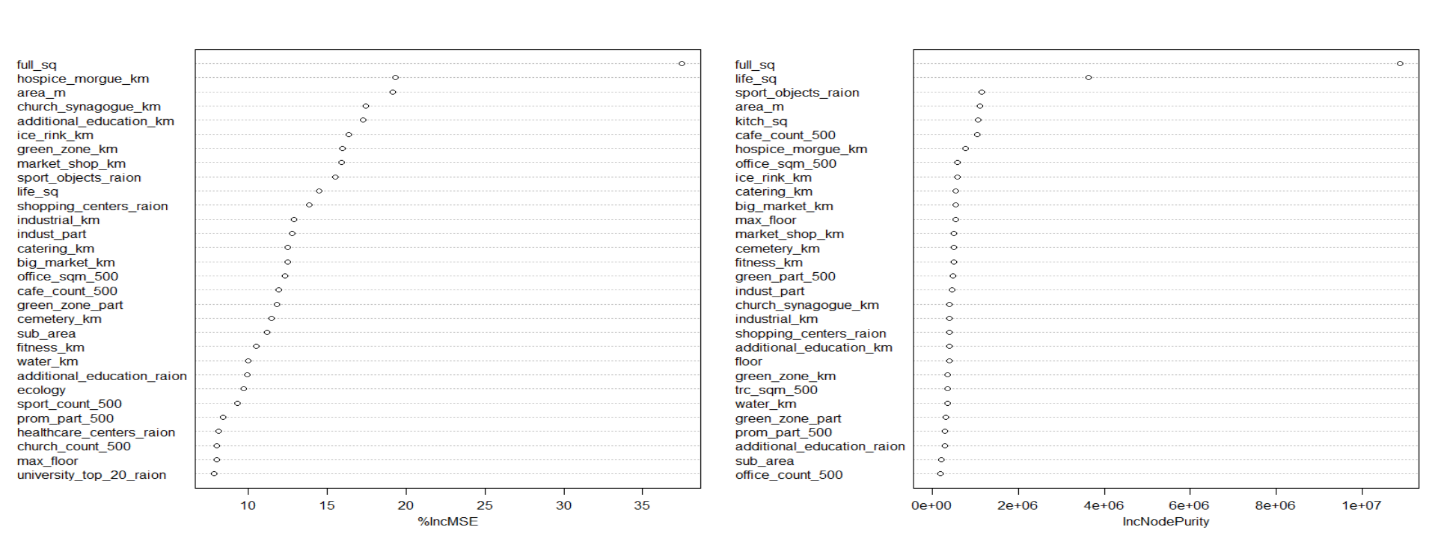
## LEARNING CURVE



Above is a plot of the learning curve created on Random Forest model mentioned above. It’s a plot of training data size on X axis and Error on Y axis. We can interpret from the curve that the test set error decreases till ~70% portion of the data. The rest 30% of the training data is redundant as it is not helping the model in improving the accuracy

## INSIGHTS

**Variable Importance:**

****

Above is a plot of variables which are contributing the most in reducing the model RSS averaged across all the trees for random Forest. Variables which were expected and also important include full area, market distance, ecology rating, green zone distance etc. Unexpected variables which are coming as important from our model includes distance to Hospice, church, cemetery, number of cafes around etc.

*Sberbank could use the prediction of prices in multiple ways:*

In Russia, housing is one of the biggest investment for most people. It is considered a high-risk investment as it involves huge money and people contribute most or all their savings. Price fluctuations in residential property will have higher wealth impact when compared to financial assets. Bank loans, mortgage pricing, and the credit cycles are strongly linked with these prices. So, it is important to continuously monitor and predict the price trends of the housing property considering various factors.

**Applications of The Model:**

* **Grant housing loans to customers:**

When a customer approaches the bank for a housing loan, the maximum loan amount that can be granted needs to be determined based on the earning potential, creditworthiness, and value of the property. While the customer discloses the purchase price, it might be worthwhile for the bank to perform its own analysis to predict and analyze the current pricing trends. Sberbank can use the developed model to predict and analyze property price trends and make an informed decision on the loan amount, repayment term and APR.

* **House mortgage/ HELOC:**

When the customer approaches the bank for mortgaging a housing property or apply for home equity line, the bank needs to evaluate the current price of that property. Even in this case, the model helps in predicting the accurate price. If the customer is interested to buy a new house mortgaging a property, the same model could be used to predict the price of the new property.

**What Happens After the Model Prediction?**

Bank would be informed about the current evaluation of the house and they can use this information, along with customers credibility, in processing the loan further. It also aids the bank to decide on factors such as:

* Maximum amount of loan

Bucket customers based on the financial credibility, mortgage evaluation if exists, into below groups:

* 80% of the Housing Price
* 70% of the Housing Price
* 60% of the Housing Price
* Percentage of interest
* Payback period

# ENHANCEMENTS/ FUTURE WORK

**What could be done additionally?**

* Since the Russian economy is highly volatile, including macroeconomic factors such as GDP and the country's economic factors in the model would improve the accuracy of predictions
* This also calls for the model to be continuously learning from real-time data
* Having historical data on same property over the years can also make the bank informed about the pricing trends location wise and could indirectly help in formulating their policies
* We would like to gain access to some higher configuration systems as this time due to the limited computational speed of our systems we could not run models like neural networks and multi fold cross validations with multiple iterations which could further boost the model performance

# REFERENCES

<https://www.sisense.com/blog/exploratory-data-analysis/>

<https://www.itl.nist.gov/div898/handbook/eda/section1/eda11.html>

https://www.techopedia.com/definition/14650/data-preprocessing