AMES HOUSING PROJECT

Project Title: **Predictive Analysis on Ames Housing Dataset using R**

ABSTRACT

The goal of this project is to analyse the given Housing data and predict the prices of houses precisely. The data was divided into two parts i.e. the training and testing dataset respectively.

Models were developed based on training dataset excluding some observations which were kept for validation purpose, and then applied on the test dataset after finding accuracy of particular models to predict the respective values.

Based on the prices predicted, by each individual model we can determine best fitting model as per our objective.

Submitted by:

ROHAN DONGARE
SWAPNIL BAGKAR
ASTHA KAUSHIK
NAHEED ANJUM
RISHIKESH GAIKWAD

CONTENTS

INTRODUCTION

Ames Housing Authority is a public housing agency that serves the city of Ames, Iowa, US. It helps provide decent and safe rental housing for eligible low-income families, the elderly, and persons with disabilities. The housing authority has collected 79 assessment parameters which describes every aspect of residential homes in Ames. These variables focus on the quality and quantity of the physical attributes of a property. Most of the variables are exactly the type of information that a typical home buyer would want to know about a potential property.

The project uses Feature selection method of Random Forest Selector (RFS) by application of package Boruta. To predict the house prices in city Ames, The machine learning algorithms used are Linear Regression, Decision Tree, Random Forest Model and SVM.

With the help of analytics on this real-world housing data, one can easily get familiar with features those are most important in determining housing prices in city of Ames.

Number of features in the given dataset: 80

Number of observations in the given dataset: 1460

SUMMARY/REVIEW OF DATASET

1) Training dataset that contains data about houses and their prices,

2) Test dataset which contains data about a different set of houses, for which we have to predict Sale Price.

STRUCTURE OF DATA

>Str (data)

```
'data.frame':   2919 obs. of  81 variables:
 $ Id           : int  1 2 3 4 5 6 7 8 9 10 ...
 $ MSSubClass   : int  60 20 60 70 60 50 20 60 50 190 ...
 $ MSZoning     : Factor w/ 5 levels "C (all)","FV",..: 4 4 4 4 4 4 4 4 5 4 ...
 $ LotFrontage  : int  65 80 68 60 84 85 75 NA 51 50 ...
 $ LotArea      : int  8450 9600 11250 9550 14260 14115 10084 10382 6120 7420 ...
 $ Street       : Factor w/ 2 levels "Grvl","Pave": 2 2 2 2 2 2 2 2 2 2 ...
 $ Alley        : Factor w/ 2 levels "Grvl","Pave": NA NA NA NA NA NA NA NA NA NA ...
 $ LotShape     : Factor w/ 4 levels "IR1","IR2","IR3",..: 4 4 1 1 1 1 4 1 4 4 ...
 $ LandContour  : Factor w/ 4 levels "Bnk","HLS","Low",..: 4 4 4 4 4 4 4 4 4 4 ...
 $ Utilities    : Factor w/ 2 levels "AllPub","NoSeWa": 1 1 1 1 1 1 1 1 1 1 ...
 $ LotConfig    : Factor w/ 5 levels "Corner","CulDSac",..: 5 3 5 1 3 5 5 1 5 1 ...
 $ LandSlope    : Factor w/ 3 levels "Gtl","Mod","Sev": 1 1 1 1 1 1 1 1 1 1 ...
 $ Neighborhood : Factor w/ 25 levels "Blmngtn","Blueste",..: 6 25 6 7 14 12 21 17 18 4 ...
 $ Condition1   : Factor w/ 9 levels "Artery","Feedr",..: 3 2 3 3 3 3 3 5 1 1 ...
 $ Condition2   : Factor w/ 8 levels "Artery","Feedr",..: 3 3 3 3 3 3 3 3 3 1 ...
 $ BldgType     : Factor w/ 5 levels "1Fam","2fmCon",..: 1 1 1 1 1 1 1 1 1 2 ...
 $ HouseStyle   : Factor w/ 8 levels "1.5Fin","1.5Unf",..: 6 3 6 6 6 1 3 6 1 2 ...
 $ OverallQual  : int  7 6 7 7 8 5 8 7 7 5 ...
 $ OverallCond  : int  5 8 5 5 5 5 5 6 5 6 ...
 $ YearBuilt    : int  2003 1976 2001 1915 2000 1993 2004 1973 1931 1939 ...
 $ YearRemodAdd : int  2003 1976 2002 1970 2000 1995 2005 1973 1950 1950 ...
 $ RoofStyle    : Factor w/ 6 levels "Flat","Gable",..: 2 2 2 2 2 2 2 2 2 2 ...
 $ RoofMatl     : Factor w/ 8 levels "ClyTile","CompShg",..: 2 2 2 2 2 2 2 2 2 2 ...
 $ Exterior1st  : Factor w/ 15 levels "AsbShng","AsphShn",..: 13 9 13 14 13 13 13 7 4 9 ...
 $ Exterior2nd  : Factor w/ 16 levels "AsbShng","AsphShn",..: 14 9 14 16 14 14 14 7 16 9 ...
 $ MasVnrType   : Factor w/ 4 levels "BrkCmn","BrkFace",..: 2 3 2 3 2 3 4 4 3 3 ...
 $ MasVnrArea   : int  196 0 162 0 350 0 186 240 0 0 ...
 $ ExterQual    : Factor w/ 4 levels "Ex","Fa","Gd",..: 3 4 3 4 3 4 3 4 4 4 ...
 $ ExterCond    : Factor w/ 5 levels "Ex","Fa","Gd",..: 5 5 5 5 5 5 5 5 5 5 ...
 $ Foundation   : Factor w/ 6 levels "BrkTil","CBlock",..: 3 2 3 1 3 6 3 2 1 1 ...
 $ BsmtQual     : Factor w/ 4 levels "Ex","Fa","Gd",..: 3 3 3 4 3 3 1 3 4 4 ...
 $ BsmtCond     : Factor w/ 4 levels "Fa","Gd","Po",..: 4 4 4 2 4 4 4 4 4 4 ...
 $ BsmtExposure : Factor w/ 4 levels "Av","Gd","Mn",..: 4 2 3 4 1 4 1 3 4 4 ...
 $ BsmtFinType1 : Factor w/ 6 levels "ALQ","BLQ","GLQ",..: 3 1 3 1 3 3 3 1 6 3 ...
 $ BsmtFinSF1   : int  706 978 486 216 655 732 1369 859 0 851 ...
 $ BsmtFinType2 : Factor w/ 6 levels "ALQ","BLQ","GLQ",..: 6 6 6 6 6 6 6 2 6 6 ...
 $ BsmtFinSF2   : int  0 0 0 0 0 0 0 32 0 0 ...
 $ BsmtUnfSF    : int  150 284 434 540 490 64 317 216 952 140 ...
 $ TotalBsmtSF  : int  856 1262 920 756 1145 796 1686 1107 952 991 ...
 $ Heating      : Factor w/ 6 levels "Floor","GasA",..: 2 2 2 2 2 2 2 2 2 2 ...
 $ HeatingQC    : Factor w/ 5 levels "Ex","Fa","Gd",..: 1 1 1 3 1 1 1 1 3 1 ...
 $ CentralAir   : Factor w/ 2 levels "N","Y": 2 2 2 2 2 2 2 2 2 2 ...
```

```
 $ Electrical   : Factor w/ 5 levels "FuseA","FuseF",..: 5 5 5 5 5 5 5 5 2 5 ...
 $ X1stFlrSF    : int  856 1262 920 961 1145 796 1694 1107 1022 1077 ...
 $ X2ndFlrSF    : int  854 0 866 756 1053 566 0 983 752 0 ...
 $ LowQualFinSF : int  0 0 0 0 0 0 0 0 0 0 ...
 $ GrLivArea    : int  1710 1262 1786 1717 2198 1362 1694 2090 1774 1077 ...
 $ BsmtFullBath : int  1 0 1 1 1 1 1 1 0 1 ...
 $ BsmtHalfBath : int  0 1 0 0 0 0 0 0 0 0 ...
 $ FullBath     : int  2 2 2 1 2 1 2 2 2 1 ...
 $ HalfBath     : int  1 0 1 0 1 1 0 1 0 0 ...
 $ BedroomAbvGr : int  3 3 3 3 4 1 3 3 2 2 ...
 $ KitchenAbvGr : int  1 1 1 1 1 1 1 1 2 2 ...
 $ KitchenQual  : Factor w/ 4 levels "Ex","Fa","Gd",..: 3 4 3 3 3 4 3 4 4 4 ...
 $ TotRmsAbvGrd : int  8 6 6 7 9 5 7 7 8 5 ...
 $ Functional   : Factor w/ 7 levels "Maj1","Maj2",..: 7 7 7 7 7 7 7 7 3 7 ...
 $ Fireplaces   : int  0 1 1 1 1 0 1 2 2 2 ...
 $ FireplaceQu  : Factor w/ 5 levels "Ex","Fa","Gd",..: NA 5 5 3 5 NA 3 5 5 5 ...
 $ GarageType   : Factor w/ 6 levels "2Types","Attchd",..: 2 2 2 6 2 2 2 2 6 2 ...
 $ GarageYrBlt  : int  2003 1976 2001 1998 2000 1993 2004 1973 1931 1939 ...
 $ GarageFinish : Factor w/ 3 levels "Fin","RFn","Unf": 2 2 2 3 2 3 2 2 3 2 ...
 $ GarageCars   : int  2 2 2 3 3 2 2 2 2 1 ...
 $ GarageArea   : int  548 460 608 642 836 480 636 484 468 205 ...
 $ GarageQual   : Factor w/ 5 levels "Ex","Fa","Gd",..: 5 5 5 5 5 5 5 5 2 3 ...
 $ GarageCond   : Factor w/ 5 levels "Ex","Fa","Gd",..: 5 5 5 5 5 5 5 5 5 5 ...
 $ PavedDrive   : Factor w/ 3 levels "N","P","Y": 3 3 3 3 3 3 3 3 3 3 ...
 $ WoodDeckSF   : int  0 298 0 0 192 40 255 235 90 0 ...
 $ OpenPorchSF  : int  61 0 42 35 84 30 57 204 0 4 ...
 $ EnclosedPorch: int  0 0 0 272 0 0 0 228 205 0 ...
 $ X3SsnPorch   : int  0 0 0 0 0 320 0 0 0 0 ...
 $ ScreenPorch  : int  0 0 0 0 0 0 0 0 0 0 ...
 $ PoolArea     : int  0 0 0 0 0 0 0 0 0 0 ...
 $ PoolQC       : Factor w/ 3 levels "Ex","Fa","Gd": NA NA NA NA NA NA NA NA NA NA ...
 $ Fence        : Factor w/ 4 levels "GdPrv","GdWo",..: NA NA NA NA NA 3 NA NA NA NA ...
 $ MiscFeature  : Factor w/ 4 levels "Gar2","Othr",..: NA NA NA NA NA 3 NA 3 NA NA ...
 $ MiscVal      : int  0 0 0 0 0 700 0 350 0 0 ...
 $ MoSold       : int  2 5 9 2 12 10 8 11 4 1 ...
 $ YrSold       : int  2008 2007 2008 2006 2008 2009 2007 2009 2008 2008 ...
 $ SaleType     : Factor w/ 9 levels "COD","Con","ConLD",..: 9 9 9 9 9 9 9 9 9 9 ...
 $ SaleCondition: Factor w/ 6 levels "Abnorml","AdjLand",..: 5 5 5 1 5 5 5 5 1 5 ...
 $ SalePrice    : num  208500 181500 223500 140000 250000 ...
```

Summary of the data/ Review of Literature :

The data types of the columns are mixed: we have integers, numeric data & factors (levels). So, it's clear that features come in fundamentally different types:

Nominal	Ordinal	Continuous/Discrete
• Lotshape • LandSlope • Utilities • Land Slope • OverallQual • OverallCond • ExterQual • ExterCond • BsmtQual • BsmtCond • BsmtExposure • BsmtFinType1 • BsmtFinType2 • HeatingQC • Electrical • KitchenQual • Functional • FireplaceQu • Garage Finish • Garage Qual • GarageCond • PaveDrive • PoolQC • Fence	• MSSubclass • MSZoning • Street • Alley • LandContour • LandConfig • Neighbourhood • Condition1 • Condition2 • BldgType • HouseType • RoofStyle • RoofMatl • Exterior1 • Exterior2 • MasVnr • Foundation • Heating • CentralAir • GarageType • MiscFeature • SaleType • SaleCondition	• LotFrontage • LotArea • YearBuilt • YearRemodAdd • MasVnrArea • BsmtFinSF1 • BsmtFinSF2 • BsmtUnfSF • TotalBsmtSF • X1stFlrSF • X2ndFlrSF • LowQualFinSF • GrLivArea • BsmtFullBath • BsmtHalfBath • FullBath • HalfBath • Bedroom • Kitchen • TotalRmsAbrGrd • Fireplaces • GarageYrBlt • GarageCars • GarageArea • WoodDeckSF • OpenPorchSF • EnclosedPorch • 3-SsnPorch • ScreenPorch • PoolArea • MiscVal • MoSold • YrSold • SalePrice

1. Some features are inherently **NUMERICAL**. They are quantities that we can measure or count. Some of these are continuous, such as the total living area (GrLivArea), while others are discrete, such as the number of rooms (TotRmsAbvGrd).
2. Other features are **NOMINAL**. They are qualitative or descriptive in nature. For example, this includes the neighbourhood in which the house is located (Neighbourhood), and the type of foundation the house was built on (Foundation). There is no inherent ordering to these features.

3. Yet others are **ORDINAL**. They comprise categories with an implicit order. Examples of this include the overall quality rating (OverallQual) or the irregularity of the lot (LotShape). We can think of them as representing values on an arbitrary scale.

Missing values in Train Dataset:

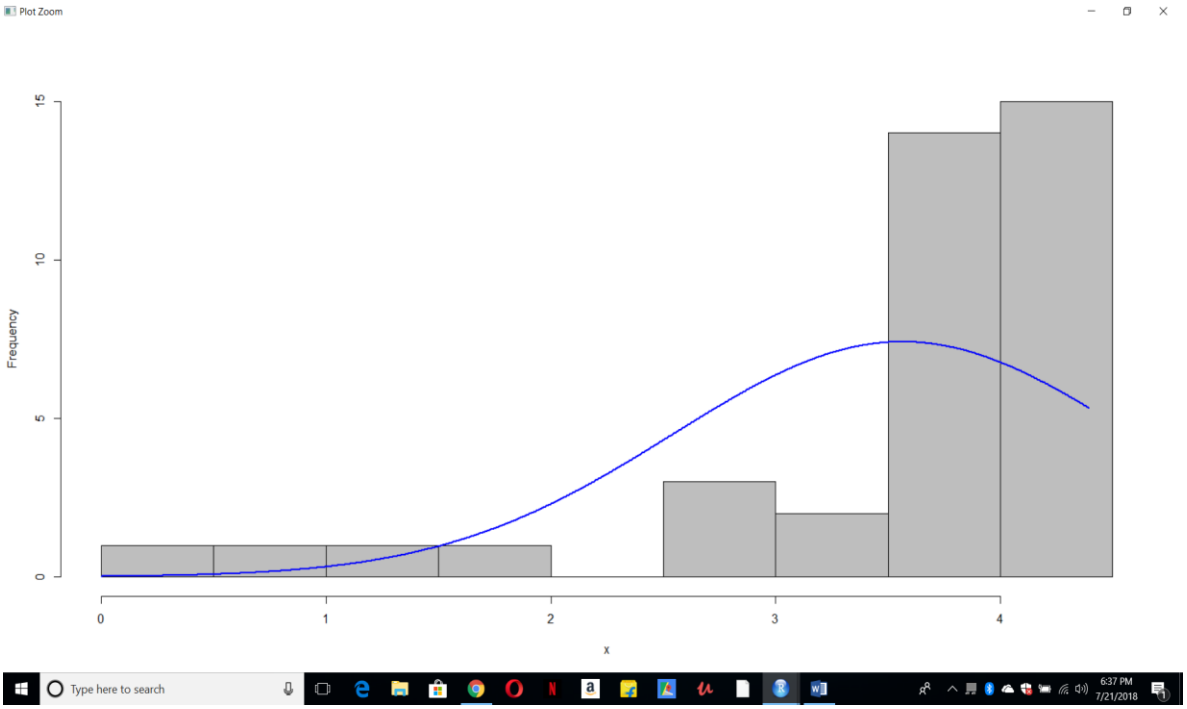
```
> #Checking for missing Values in Train
> NAcol= which(colSums(is.na(TrainDs)) > 0)
> sort(colSums(sapply(TrainDs[NAcol], is.na)), decreasing = TRUE)
LotFrontage GarageYrBlt  MasVnrType  MasVnrArea  Electrical
        259          81           8           8           1
> cat('There are', length(NAcol), 'columns with missing values')
There are 5 columns with missing values
```

Missing values in Test Dataset:

```
> #Checking for missing Values in Test
> NAcol= which(colSums(is.na(TestDs)) > 0)
> sort(colSums(sapply(TestDs[NAcol], is.na)), decreasing = TRUE)
 LotFrontage  GarageYrBlt   MasVnrType   MasVnrArea     MSZoning    Utilities BsmtFullBath
         227           78           16           15            4            2            2
BsmtHalfBath   Functional  Exterior1st  Exterior2nd   BsmtFinSF1   BsmtFinSF2    BsmtUnfSF
           2            2            1            1            1            1            1
 TotalBsmtSF  KitchenQual   GarageCars   GarageArea     SaleType
           1            1            1            1            1
> cat('There are', length(NAcol), 'columns with missing values')
There are 19 columns with missing values
```

HISTOGRAM:

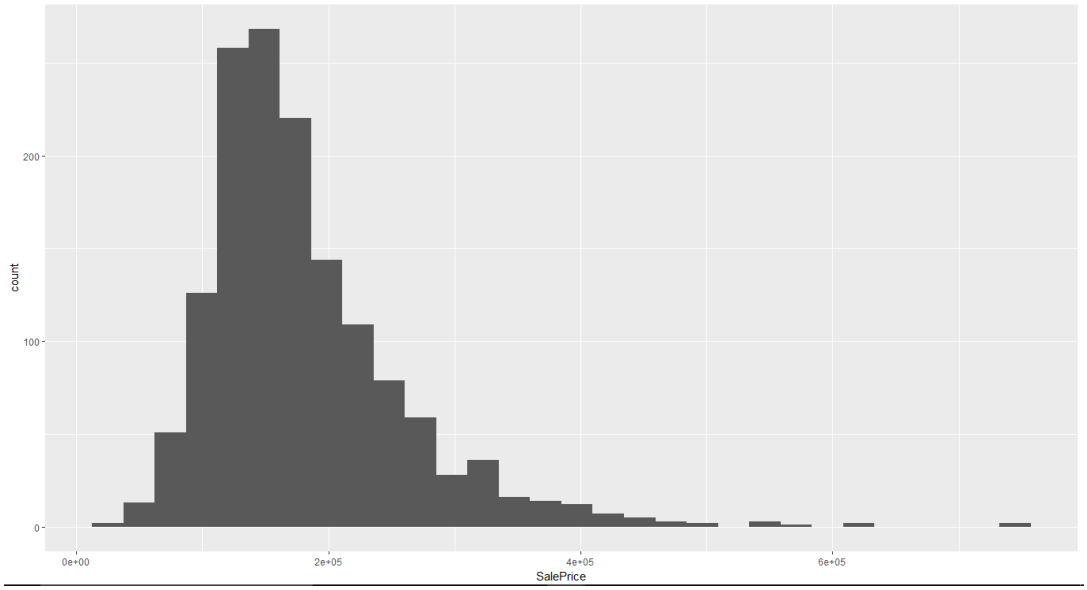
Histogram to show the distribution of data after transformation has been performed which is Normal.



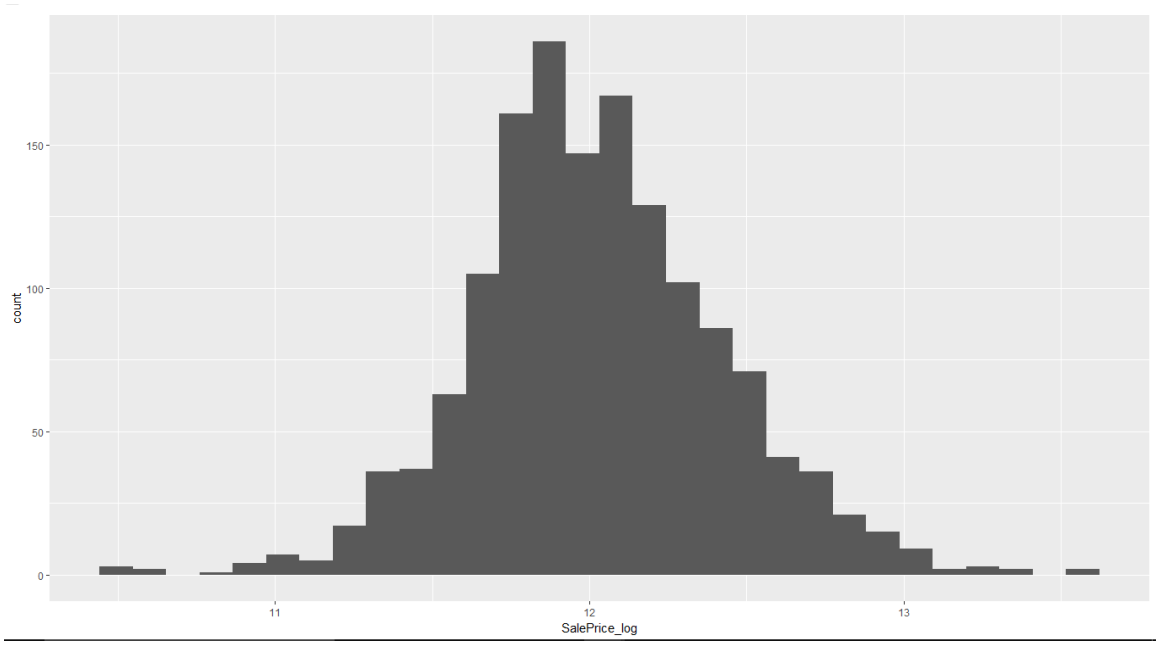
VISUALIZATION :

Univariate Analysis on Numerical Variables:

Histograms are perfect for visualizing distribution of numeric variables. Below is the distribution plotted on target variable 'SalePrice'.

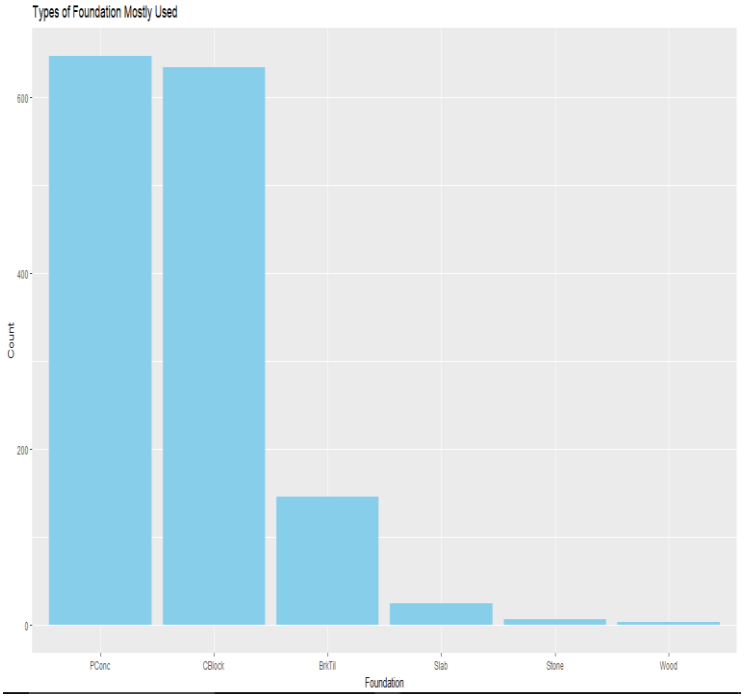


We notice that the distribution is skewed towards cheaper homes, with a relatively long tail at high prices. To make the distribution more symmetric, we can try taking its logarithm.



Besides making the distribution more symmetric, working with the log of the sale price will also ensure that relative errors for cheaper and more expensive homes are treated on an equal footing. As such, we can think of taking 'log(SalePrice)' as our true target variable.

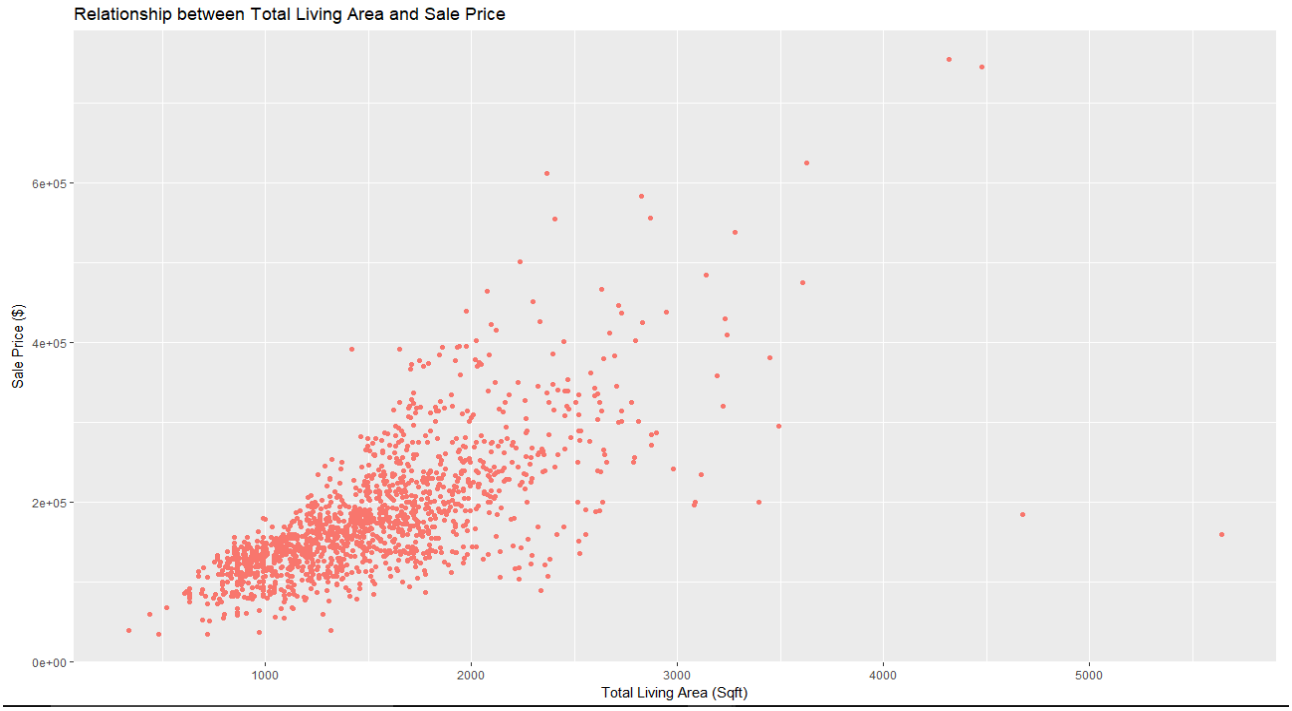
Univariate Analysis on Categorical Variables:



Here is a bar chart representing an important feature **'Foundation'** which contains types of materials used in construction such as, 'Brick & Tile', 'Cinder Block', 'Poured Contrete', 'Slab', 'Stone, Wood'.From the bar chart we can say, Stone and wood are very rarely used materials. Whereas Poured Concrete & Cinder Blocks are extensively used.

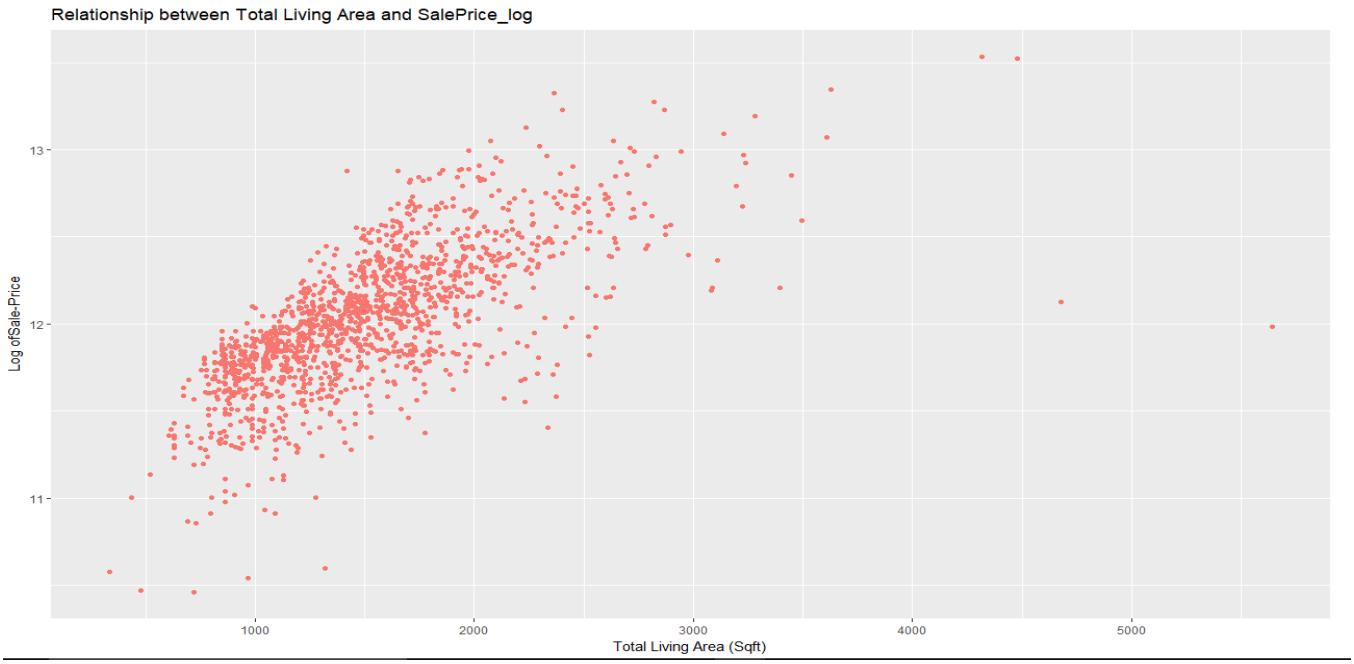
Bivariate Analysis on Numerical Variables:

Since, the total living area of a house is likely to be an important factor in determining its price. Here is a Scatter plot explaining the relationship between 'GrLivArea' and 'SalePrice'.

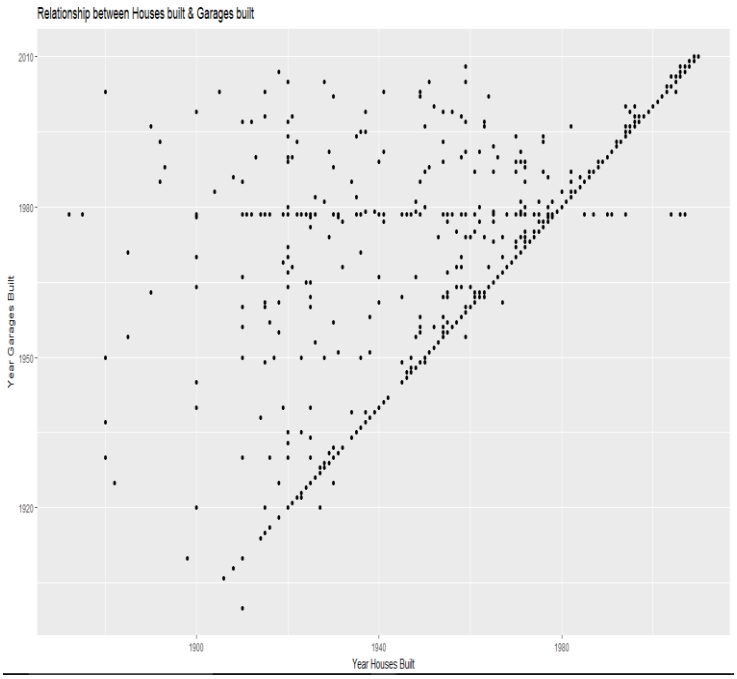


There is indeed a strong dependence of 'Saleprice' on the 'Total living area', as expected. The larger the house, the more expensive it tends to be.

When we take the log in the second plot, the distribution looks notably more balanced.

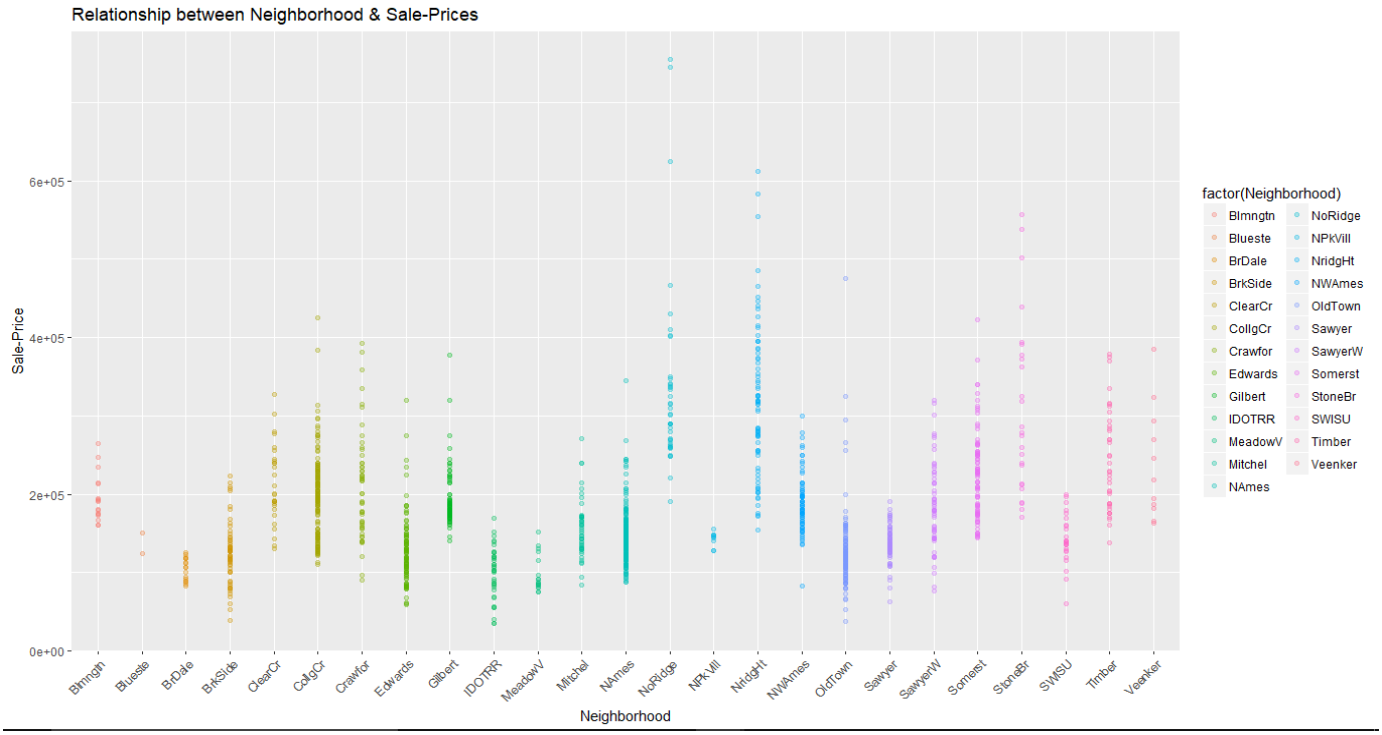


There are two points that don't seem to fit in with the rest. Towards the lower right part of the plot, there are two very large houses (bigger than 4500 sqft) with unusually low sale prices**.** Hence, we conclude there are **OUTLIERS** present in the dataset.



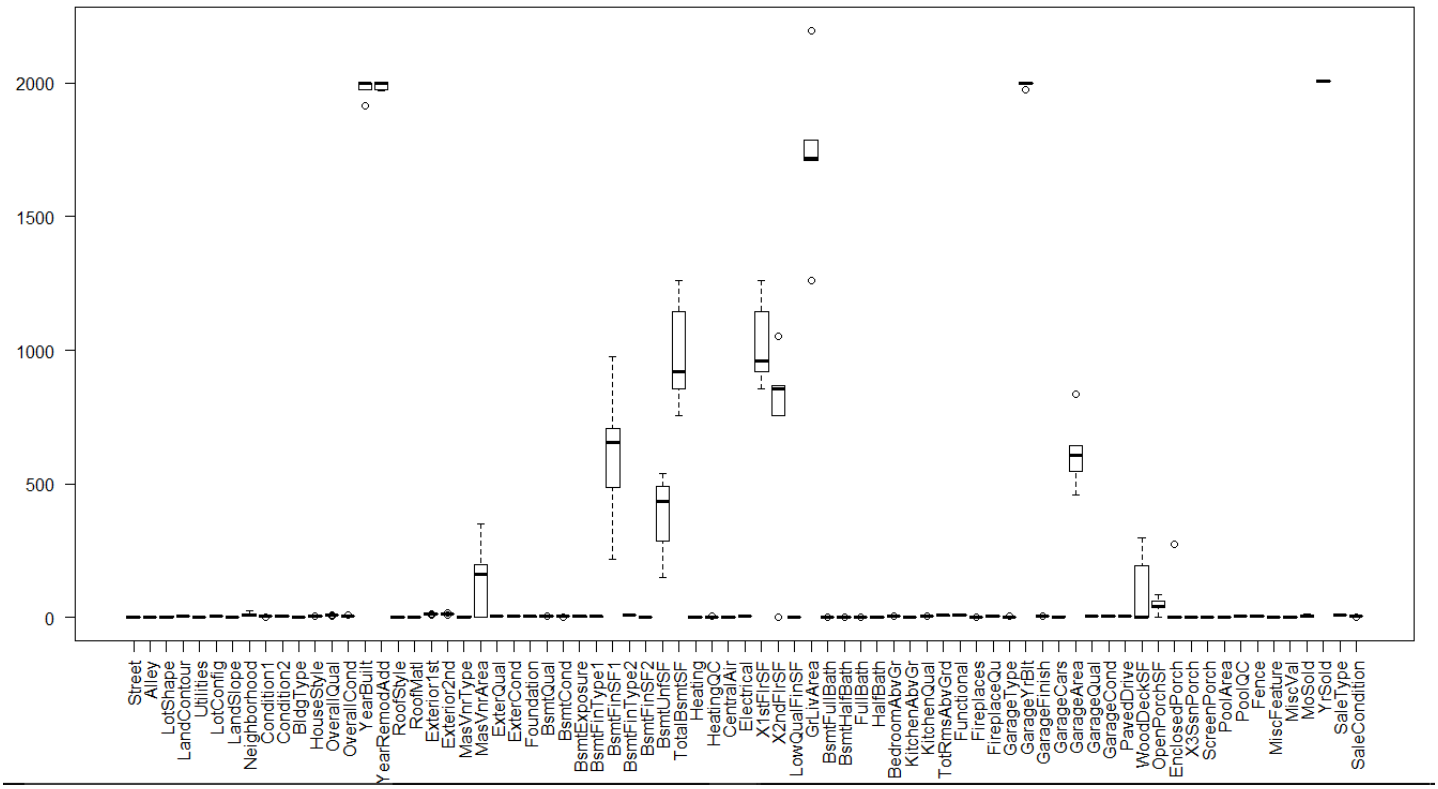
Scatter plot of 'YearBuilt' and 'GarageYrBlt' tells us that the majority of garages were built at the same time as the houses they belong to; these forms the diagonal line that runs across the plot. A significant number were also added later, these are the points above the line.

Bivariate Analysis on Categorical Variables:



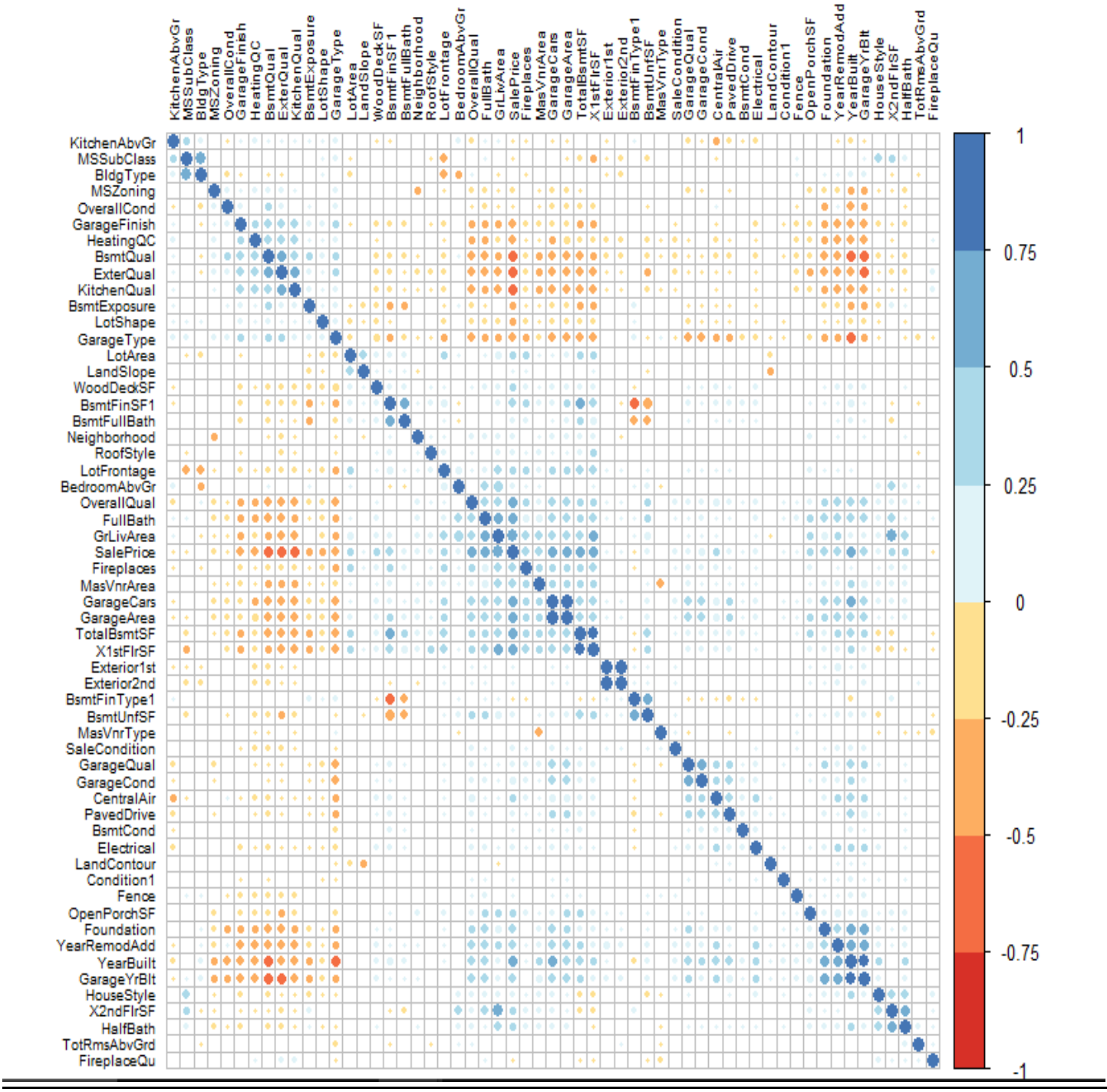
From a graph above we can say, there is considerable variation in price between neighbourhoods. The figure allows to get an idea of how different areas compare to each other at a glance.

Boxplot of Data:



CORRELATION:

The Following figure shows the CORRELATION MATRIX HEATMAP in R using package corrplot. This gives us an idea about which features are highly correlated and are considerable or not.



Feature Selection :

Feature selection is considered one of the important steps in the predictive modelling.

Boruta is a wrapper algorithm for all relevant feature selection, capable of working with any classification method that output variable importance measure (VIM); by default, Boruta uses Random Forest. The method performs a top-down search for relevant features by comparing original attributes' importance with importance achievable at random, estimated using their permuted copies, and progressively eliminating irrelevant features to stabilise that test.

	meanImp	medianImp	minImp	maxImp	normHits	decision
MSSubClass	9.7933	9.79902	7.04921	11.74539	1.0000	Confirmed
MSZoning	9.3518	9.45322	6.94020	11.23135	1.0000	Confirmed
LotFrontage	4.2583	4.18198	1.38902	7.20142	0.9394	Confirmed
LotArea	10.6938	10.62954	8.43777	13.41521	1.0000	Confirmed
Street	0.3481	0.46420	-1.61326	1.21757	0.0000	Rejected
Alley	2.5059	2.47631	-0.26176	4.84958	0.5051	Confirmed
LotShape	3.1660	3.24438	-0.07542	5.35060	0.6970	Confirmed
LandContour	3.3768	3.42784	1.07600	5.23818	0.8384	Confirmed
Utilities	0.0000	0.00000	0.00000	0.00000	0.0000	Rejected
LotConfig	0.1073	0.20008	-1.71078	1.89019	0.0000	Rejected
LandSlope	2.3934	2.36082	0.26342	4.02294	0.4444	Confirmed
Neighborhood	8.5941	8.62766	6.29580	9.93929	1.0000	Confirmed
Condition1	2.2214	2.25507	-0.95939	4.65137	0.3838	Rejected
Condition2	-1.1613	-1.43355	-2.28239	0.51922	0.0000	Rejected
BldgType	7.3371	7.49104	4.95386	10.42132	1.0000	Confirmed
HouseStyle	7.0627	7.05379	4.75448	8.76019	1.0000	Confirmed
OverallQual	17.5630	17.80139	14.28315	19.65221	1.0000	Confirmed
OverallCond	5.0011	4.50698	2.74445	9.08928	1.0000	Confirmed
YearBuilt	12.6493	12.61279	11.09284	14.67791	1.0000	Confirmed
YearRemodAdd	10.8016	10.88807	8.23829	12.35715	1.0000	Confirmed
RoofStyle	2.4350	2.47788	0.05044	4.56849	0.4646	Confirmed
RoofMatl	1.1946	1.40122	-0.80725	2.72941	0.0202	Rejected
Exterior1st	4.3878	4.37383	2.60527	7.03167	0.9697	Confirmed
Exterior2nd	4.2515	4.25417	1.80087	5.94607	0.9293	Confirmed
MasVnrType	3.0857	3.00320	1.38184	5.27520	0.7273	Confirmed
MasVnrArea	6.2860	6.48309	3.37661	8.42566	1.0000	Confirmed
ExterQual	12.2440	12.29478	9.37513	14.18108	1.0000	Confirmed
ExterCond	1.1958	1.33576	-1.20048	2.98923	0.0404	Rejected
Foundation	7.0526	7.08525	5.47847	8.45899	1.0000	Confirmed
BsmtQual	9.1337	9.46793	4.99792	11.82157	1.0000	Confirmed
BsmtCond	2.2050	2.18849	-0.12784	5.03705	0.3636	Confirmed
BsmtExposure	3.1980	3.19868	0.52674	5.42958	0.7071	Confirmed

Output showing feature importance

STEPS PERFORMED :

1. Loading the data into the Rstudio.

```
TrainDs = data.frame(fread(file.choose()), stringsAsFactors = FALSE)
TestDs=data.frame(fread(file.choose()),stringsAsFactors=FALSE)
```

   Making the necessary conversions to make the data more usable.

   a. Introduced level "NO" to features where NA's are not supposed to consider missing values.
   b. Log transformed the target feature in order to achieve normalcy.
   c. Converted ordinal variables such as 'OverallQual','OverallCond','TotRmsAbvGrd' into the categorical variables.
   d. Took a Subset of important features selected by algorithm to build a model on.

2. Cleaning data.
   a. Treated missing values by imputing Means and Modes in Numeric variables and Categorical variables respectively.

3. Visualized data, by developing various plots for exploratory analysis.

4. Creating a Training and Validation dataset with 300 rows being the validation data and rest being the Training data.

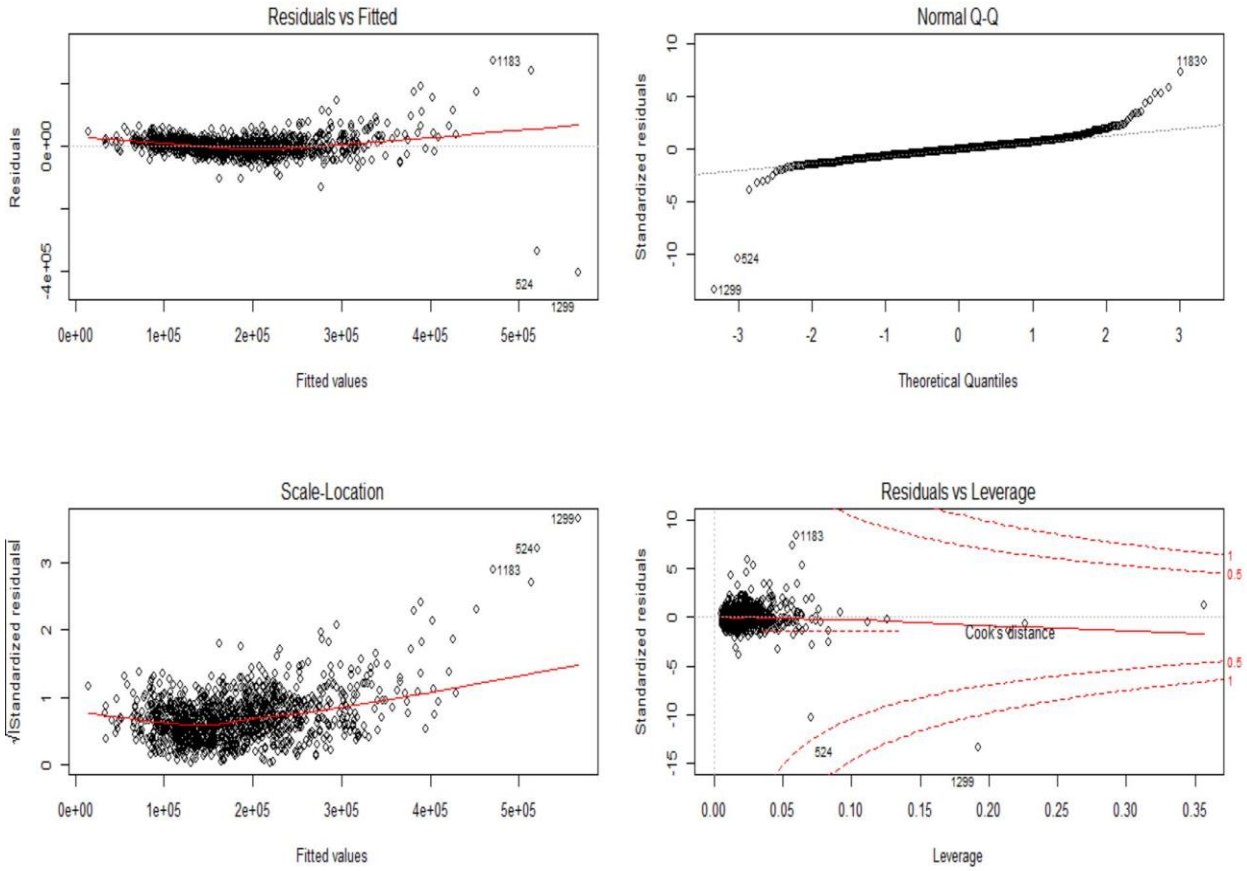
5. Building Models.
   a. **Model 1:** Linear regression algorithm. It establishes linear relationship between the predictor variables(x) and response variable(y) to predict for the unknown values in response variable when the predictor values are known.
   b. **Model 2:** Tree classifier model algorithm. The decision tree classifiers organize a series of test questions and conditions in a tree structure.
   c. **Model 3:** Support Vector Machines algorithm. It is a data classification method that separates data using hyper planes.
   d. **Model 4:** Random Forest algorithm. It is a tree-based algorithm which involves building several trees (decision trees), then combining their output to improve generalization ability of the model.

6. Validating builder models on the validation set and creating the actual confusion matrix based on predicted values and analysing the results.

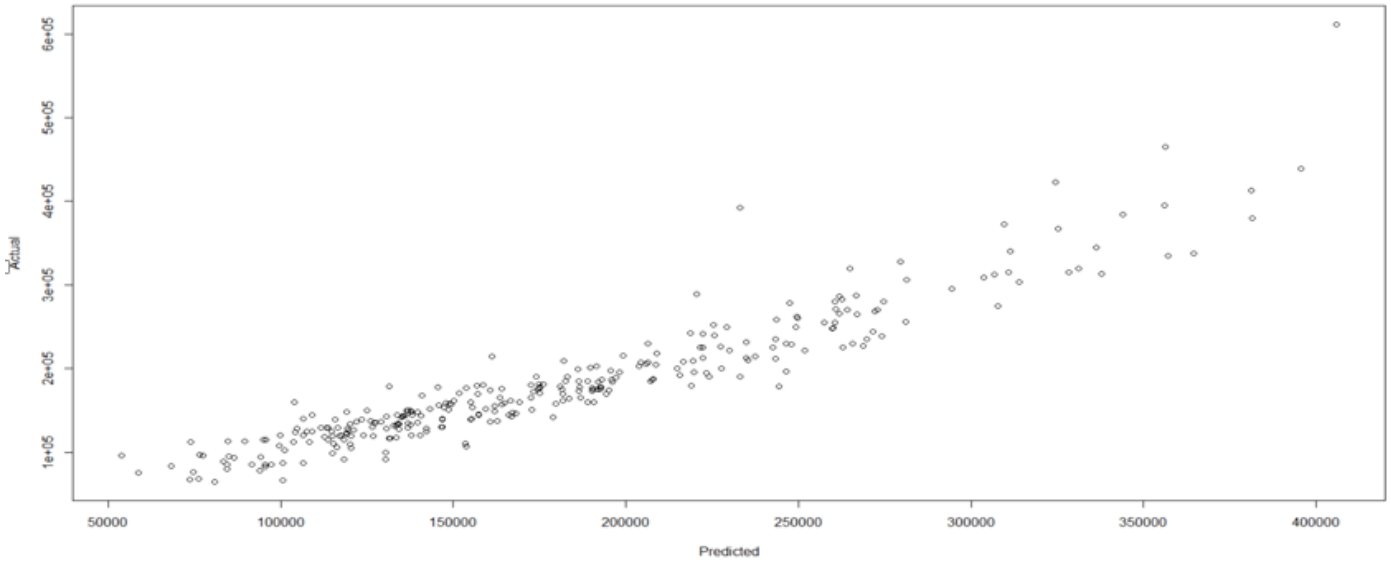
7. Predicting values for the testing dataset and Interpreting results.

MODEL PLOTS:

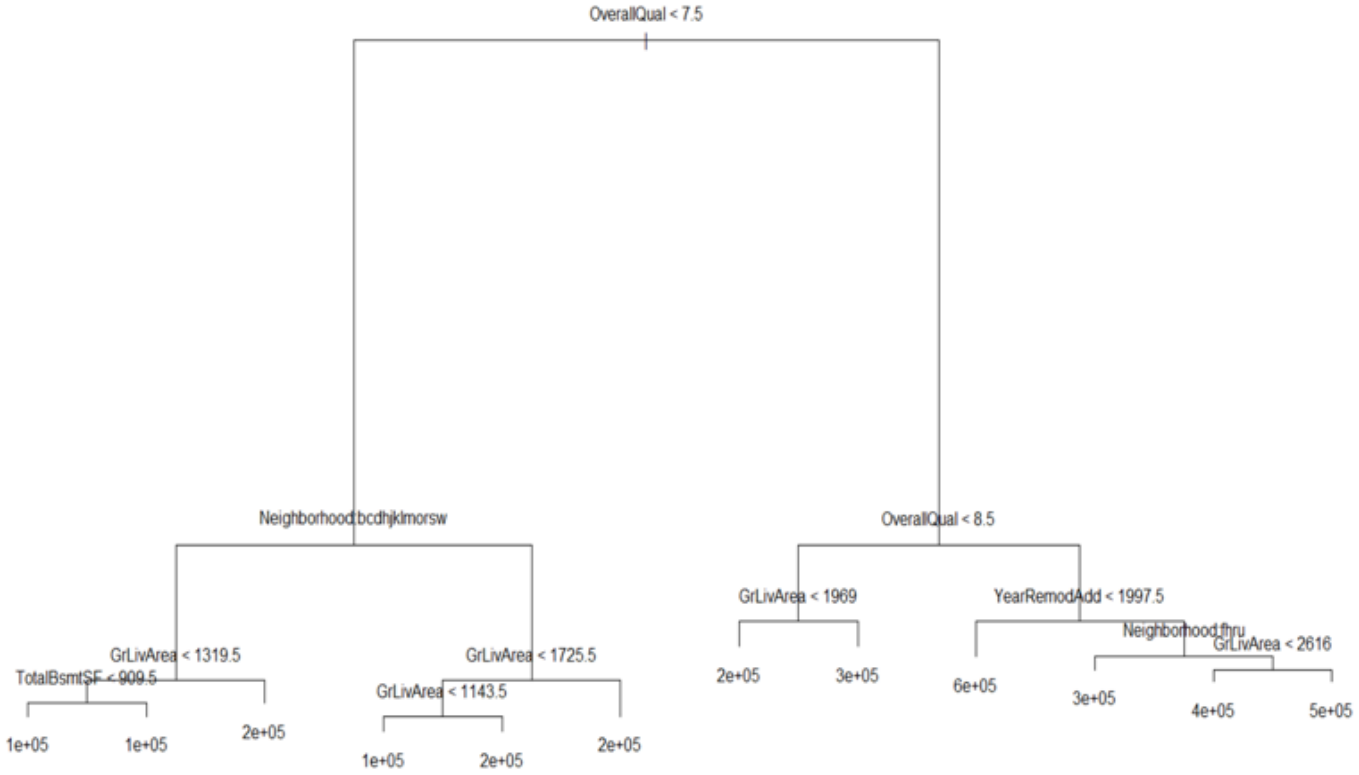
1. Linear Regression Model:
    A. Residual VS Fitted and Normality Plot



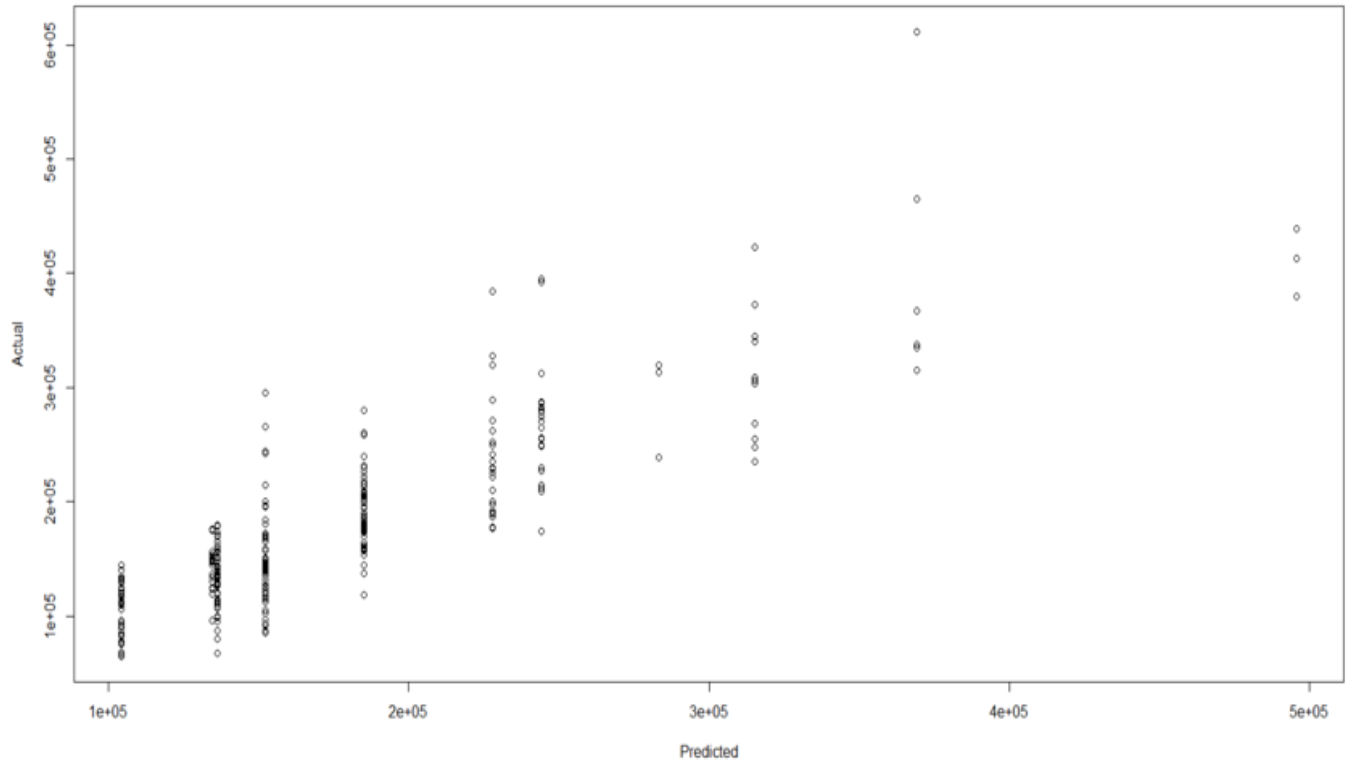
B. ACTUAL Vs. PREDICTED PLOT:



2. Decision Tree Classifier:
   TREE PLOT:

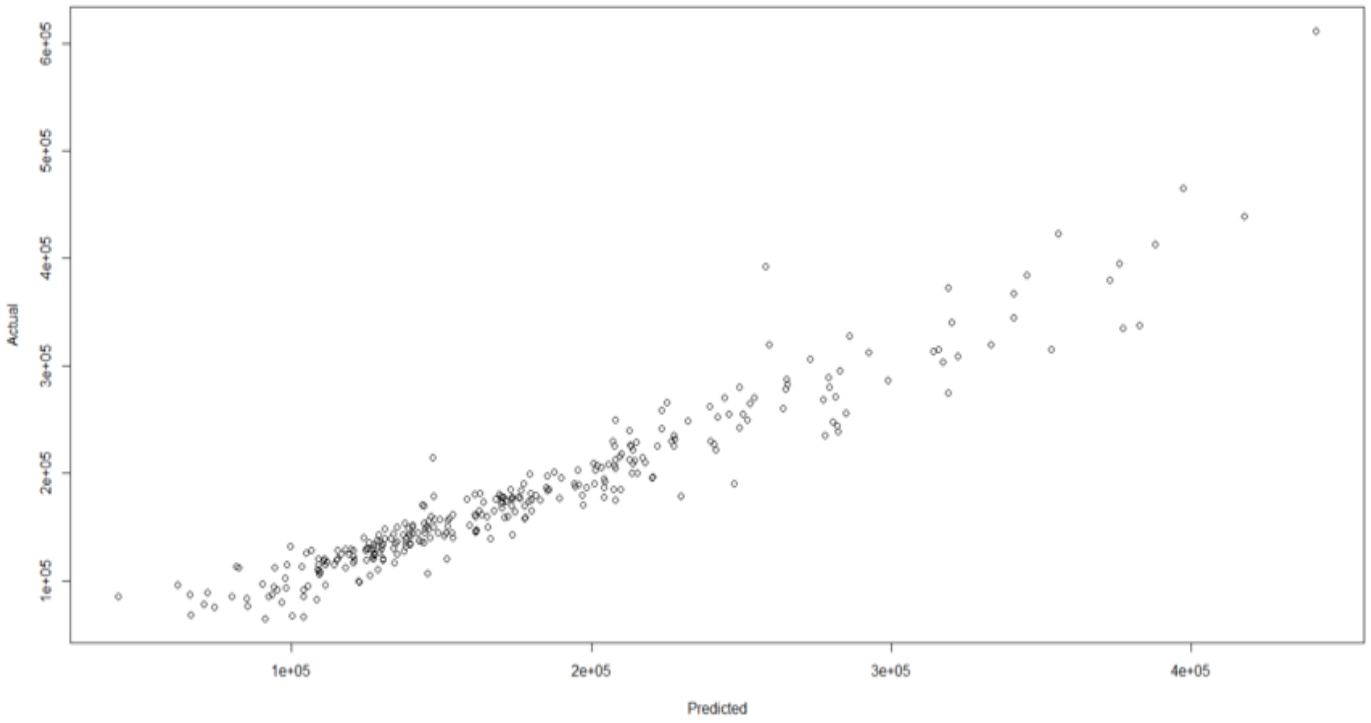


ACTUAL VS. PREDICTED GRAPH:



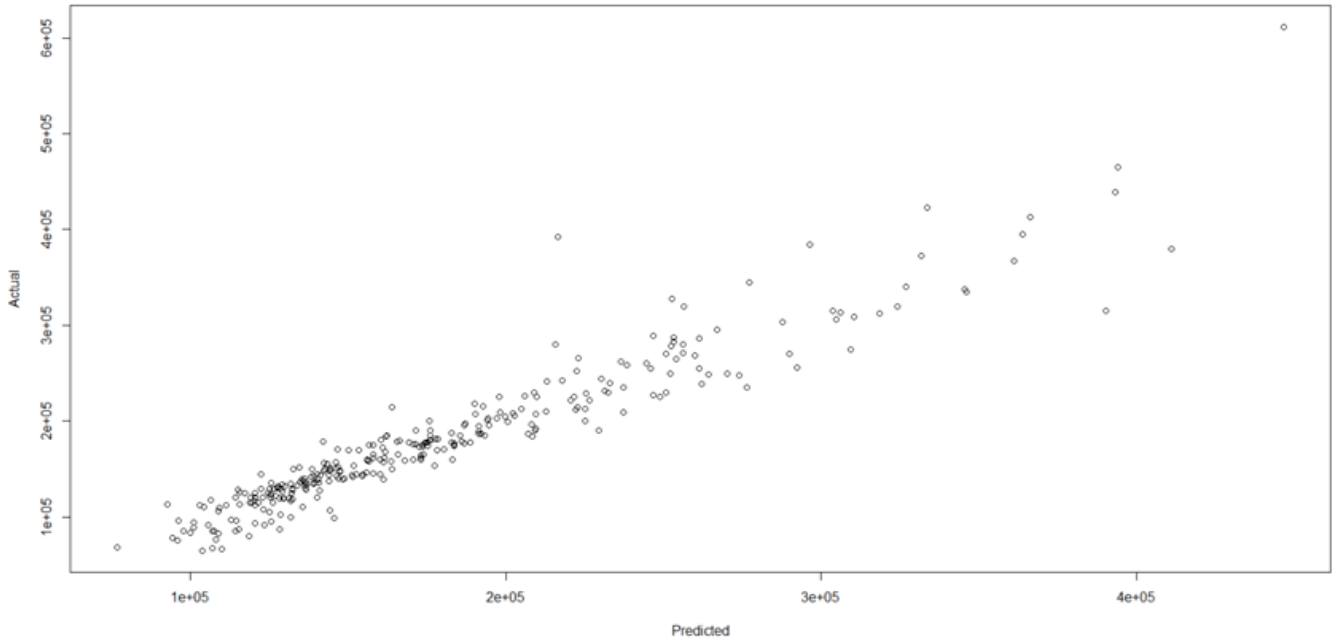
3. Support Vector machine:

   ACTUAL VS. PREDICTED GRAPH:



4. Random Forest:

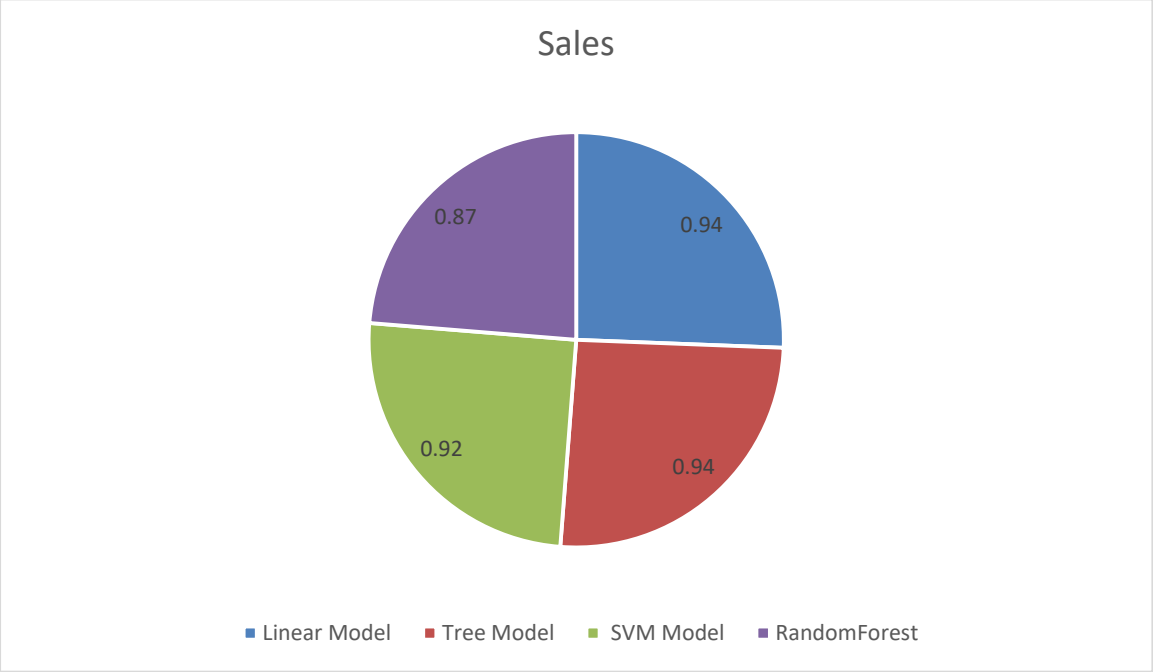
   ACTUAL VS. PREDICTED GRAPH:



MODEL EVALUATION:

Model	Median Prediction Error in '$'	% Difference than Correct Value
Linear Model	62058	7.889
Tree Classifier Model	78860	13.1
Support Vector Machine	52634	5.396
Random Forest Model	55470	5.624

ACCURACY COMPARISON OF ALL MODELS:



As we can see that the

Support Vector Machine Model (Model 3) and Random Forest Model (Model 4) are nearly same and highly accurate in predicting precise house prices i.e. 94 % and seems to be the best of all among models we have trained. However, Linear Model house prices prediction accuracy is 92% and Tree classifier Model is the worst among all in predicting house prices with 87% accuracy. The Prediction accuracy can be further improved by using ensemble modelling.

Conclusion :

The relationship between house prices and the economy is an important factor For predicting house prices. As per buyer and sellers concern Housing prices trends are very important to study before making an investment, Hence it is directly or indirectly related to current economic situation.

Therefore it is important to predict housing prices without bias to help both buyers and sellers make their decisions.

The data contains list of helpful features as well as the unnecessary or Luxurious features of which probabilities of occurrence are very less. The data is more concentrated or helpful for the middle class people or a broker-seller. Since, huge amount of data belongs to cheaper or affordable house prices with critical or important amenities.

With the help of Data science and R we have managed to develop a machine learning algorithm to predict housing prices using given features with fair accuracy of around 94% and can be further improved with different Possible approaches and ensemble modelling.

REFERNCES :

1. www.cran.r-project.org (Information on R packages)
2. www.stackoverflow.com (Information on Code optimisation)
3. www.edx.org (Information about visualizing data)
4. www.rapidtables.com (Graphical representation)