|  |
| --- |
| MA321 GROUP COURSEWORK |
| Srikanth Reddy (2004445)  MSc Data Science Placement  26-03-2021 |

**ABSTRACT**

Real Estate Market is the growing business nowadays. The agents need to predict the quality and the prices of the house. It needs more accurate method based on location, house type, size, build year, local amenities, and some other factors which could affect house demand and supply. With limited dataset and data features, a practical and composite data pre-processing, different classification, and regression methods are examined in this report.

**CONTENTS**

|  |  |  |
| --- | --- | --- |
| **Sr no.** | **Title** | **Page no.** |
| 1 | Numerical and graphical summaries: | 3 |
| 2 | Classification | 6 |
| 3 | Regression | 7 |
| 4 | Research questions | 8 |
| 5 | References | 10 |
| 6 | Appendix | 11 |

**WORD COUNT**

1137

**INTRODUCTION:**

The dataset “House Price Prediction” has been taken from the data science platform named Kaggle. Housing is a billion-dollar industry across the world. Emerging data science techniques have made the job easy for the Real Estate agents in selling the housing properties. The cost of a property varies on a various factor. They mainly depend on the location, size, area, age, neighborhood, etc. In this project we will use data science techniques to mainly predict two things, one, the condition of the house and second, price of the house based on a few features available. As a data scientist we need to find the trends and patterns in the data. Sometimes, we may not need to make predictive models to understand them, simple analysis lead us to fruitful findings.

**PRELIMINARY ANALYSIS:**

In this project, the first step will be to perform Descriptive Analysis and EDA. It is very important to understand the data and see its structure before proceeding further. This is like a planning phase where we understand our data and make few assumptions about it.

The next phase involves, data cleaning. This is the most important phase since it is crucial to clean the data before proceeding further. If we feed bad data to the statistical models, we will get garbage results. As a part of this analysis, we will check for the missing values, outliers, etc.

Once the data cleaning phase is done, we will proceed with data modelling. In this project we are required to use classification technique to predict the housing condition and regression techniques to predict the house selling prices. We will evaluate the ML models and check their performance.

**ANALYSIS:**

1. **NUMERICAL AND GRAPHICAL SUMMARIES:**

We have a house information dataset with 1460 rows and 51 columns (features). Before proceeding with the pre-processing there is a need to check for the missing values or null values. After getting the information about missing values we came to know that 4 of the features i.e. Alley, PoolQC, Fence and MiscFeature have more than 80% of the NAs, therefore we straight away eliminated these features from the data. However, 6 of the features i.e., LotFrontage, MasVnrArea, BsmtQual, BsmtCond, GarageType, GarageCond contain reasonable amount of NAs (<30%) which can be imputed.

After removing the non- required columns, there is a new data set named as “data1” with 46 features. For numerical summaries we identifies that 22 coulmns i.e. LotFrontage, LotArea, OverallQual, OverallCond, YearBuilt, MasVnrArea, TotalBsmtSF, X1stFlrSF, X2ndFlrSF, LowQualFinS, GrLivArea, FullBath, BedroomAbvGr , KitchenAbvGr, TotRmsAbvGrd, Fireplaces, GarageArea, PoolArea, MiscVal, MoSold, YrSold, SalePrice have only numeric values.

Now, we have found the correlation of each feature with respect to Sales Price and we draw a conclusion that usually big houses have a higher price and here it makes complete sense that a High Priced house will have a very good quality with big living area, garage area, a big basement area and finally a big First Floor in terms of area.

We have done some detail analysis and tried to find the relationship between different parameters. Such as how the house price (sale price) depends on neighborhood and it has been concluded that area known as “NoRidge” have highest house prices whereas “Somerst”, “Timber” and “Veenker” have nearly same prices. The following graph shows the top six areas with expensive houses.

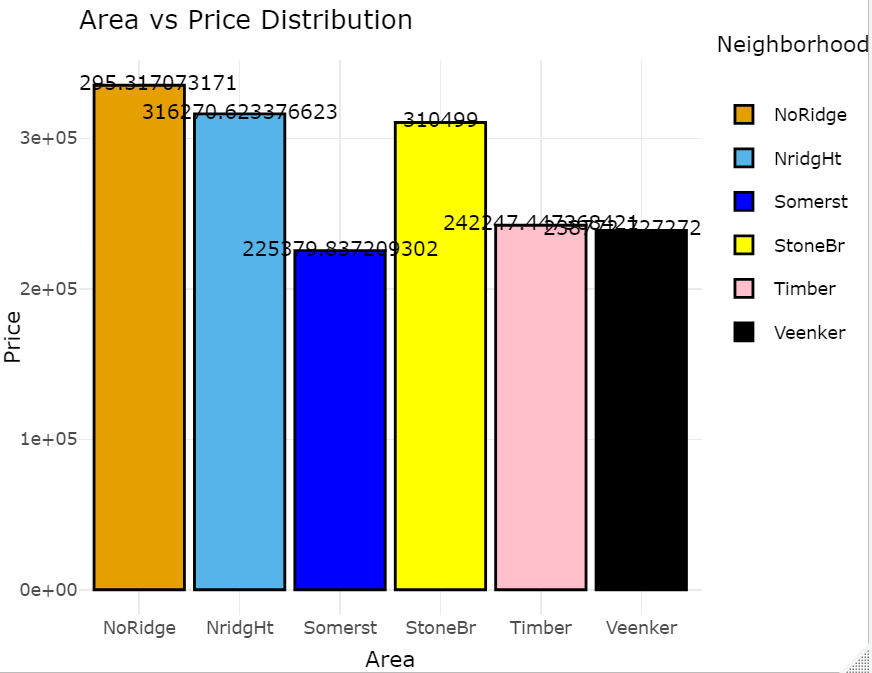


Figure 1: Area vs Price plot

Similarly, after getting the relationship between some other variables say Foundation type and House Price, we concluded that houses with Foundation type "Poured Concrete” tend to cost more than other foundation as shown in the figure below:

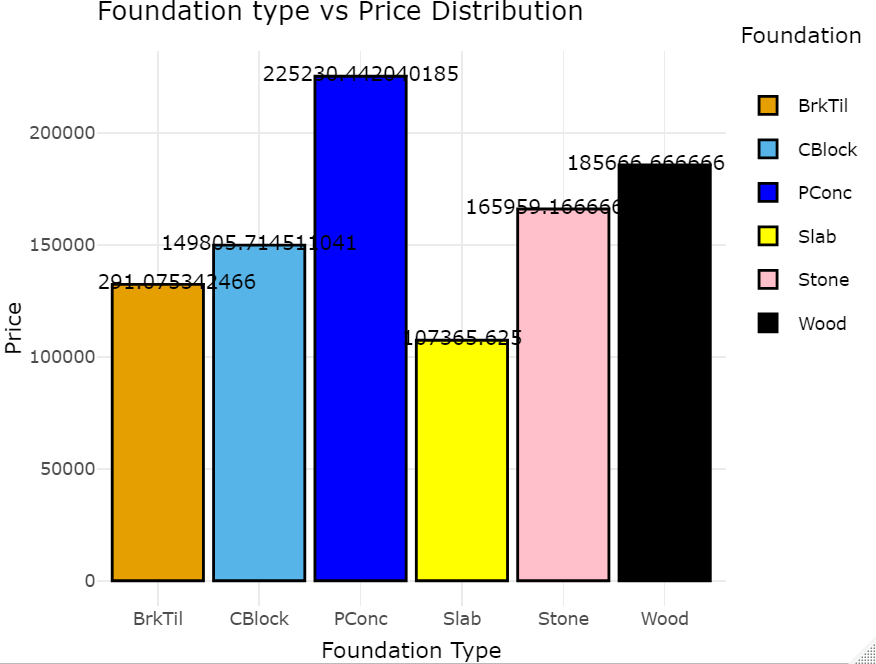


Figure 2: Foundation type vs Price Distribution Plot

Another pair that is taken into consideration is Floor vs Foundation type. From the figure below, it can be clearly observed that only Poured Concrete and Cinder Block have been used for as foundation in SLvl House Type. Also, all kinds of Foundations have been used in the construction of 1.5Fin and 2Story buildings.

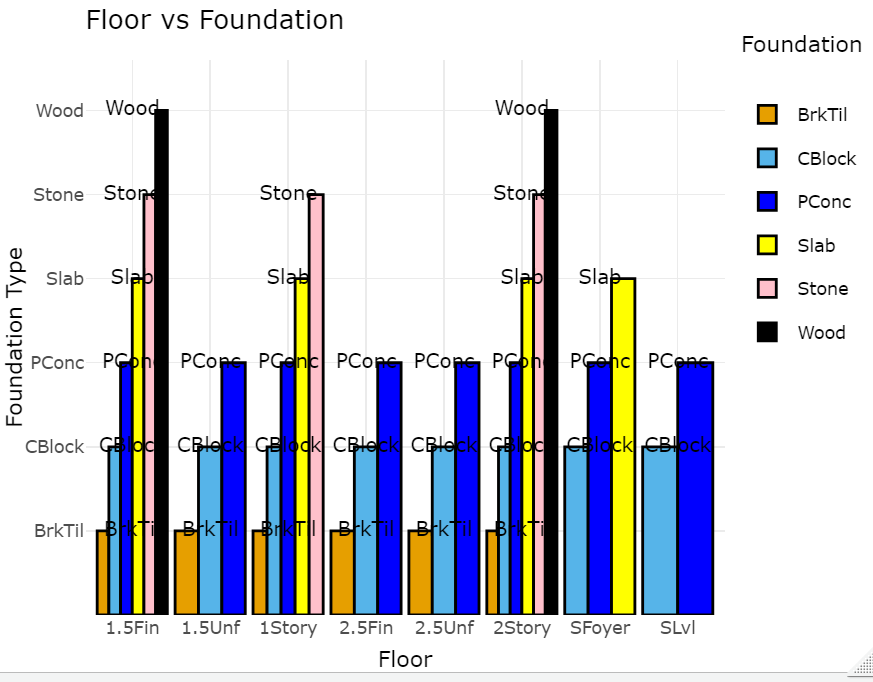


Figure 3: Floor type vs foundation plot

The last relation is between construction year and foundation. From the graph it is clear that the houses that constructed in the recent years are the most expensive ones. Also, a few houses made in early 1800s and late 1800s are also values very high, they might me antique houses.

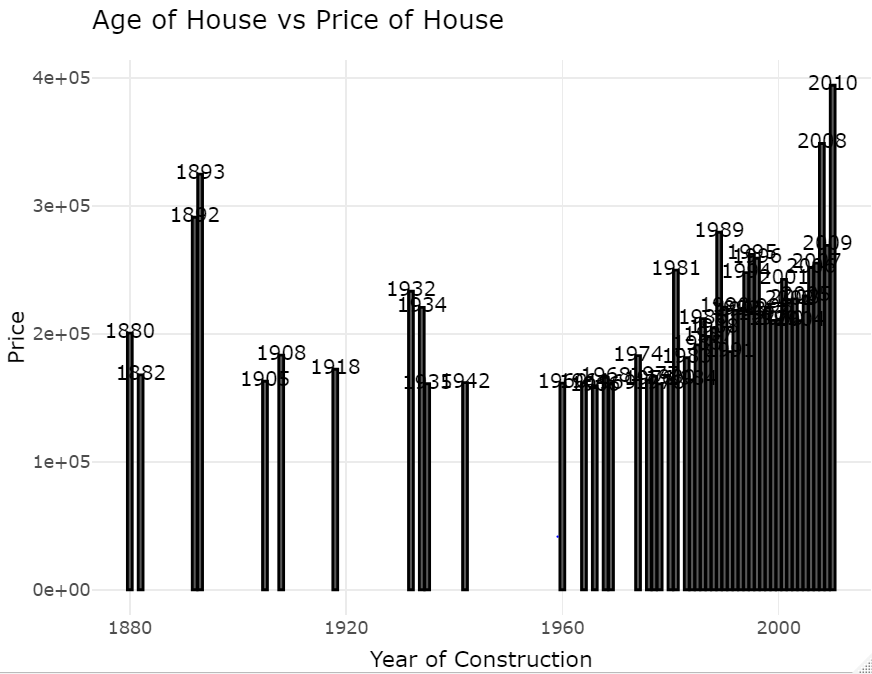


Figure 4: year of construction vs price plot.

1. **MISSING VLAUE IMPUATION:**

In this task, we have done missing value imputation differently for Numeric and Categorical data. For numeric data, we have grouped the corresponding missing value column with a categorical column. Instead of performing blind mean, median and mode imputation, we have grouped the numeric data with the categorical features, then we have performed the groupwise mean imputation. When it comes to categorical features, we have imputed them with the feature’s corresponding mode values. For numeric features it was easy to group by categories and perform the mean imputation and it is the best techniques as we will not be imputing the features with any random values. However, we didn’t have any categories that are dependent on other categories, hence we had to impute them with the mode value.

1. **CLASSIFICATION:**

The first task was to impute the missing values which have been performed separately on numerical features and categorical features. As specified by the assignment, we have grouped the data depending on overall condition in Good, Average and Poor houses. Moreover, a model cannot work with the categorical values, so they need to be converted into factors. The undesired columns such as years do no contribute anything in modeling a system so have removed that columns also. Now the data is all set for modeling named as “data2” with “out” as a response variable and rest of them are predicting variables. Further this data is split into Training and testing data using (80-20 split). This 80-20 split is one of the resampling methods.

For classification algorithms, two methods have been used here.

1. Logistic Regression
2. State Vector Machine classifier.

**Logistic Regression**

Starting with multiclass Logistic Regression there is a need to set a reference level and we have selected “poor” as a reference level and we decided to compute P-values to see which features are contributing more to model and the P-values should be less than 5%, so we have directly eliminated the features which shows grater p- values. Eliminating the features with high p values is not extremely mandatory if the R Square is not increasing. But the training cost increases which is ideally not a great idea when working in a commercial setup.

For Logistic Regression we draw the following conclusions:

**Error**: =0.007

**Confusion matrix:**

|  |  |  |  |
| --- | --- | --- | --- |
| actual  Predicted | Poor | Average | Good |
| Poor | 11 | 0 | 1 |
| Average | 0 | 205 | 1 |
| Good | 0 | 0 | 62 |

The above matrix shows that we get quite accurate results for poor and average classes and there is some error in predicting class Good.

**State Vector Machine**

We have used two different SVM kernels i.e. Rbf and sigmoid for this algorithm and we got fairly same results for both of them given below:

**Error:** 0.04642857

**Confusion matrix:**

|  |  |  |  |
| --- | --- | --- | --- |
| actual  Predicted | Poor | Average | Good |
| Poor | 0 | 0 | 0 |
| Average | 11 | 205 | 2 |
| Good | 0 | 0 | 62 |

The machine successfully spots all the average houses, but fails to recognize the poor ones.

1. **REGRESSION:**

Now we need to predict the house prices and it can only be done by regression algorithms. Like classification, the first task is to handle the categorical features. For this we can create dummy and include them in the dataset, But we decided to select the highly correlated numerical variables with "SalePrice" as the Predictor variables and these are “OverallQual”, “GrLivArea”, “GarageArea”, “TotalBsmtSF”,”X1stFlrSF” and “FullBath”. However, last two have p value greater than 5%, so we removed these features also. Hence, we decide to proceed with rest of the four features to train our model. We have used two training algorithms.

1. Linear Regression
2. State vector machine regressor

Table1: Comparison between Linear Regression and SVM

|  |  |  |
| --- | --- | --- |
|  | R Score | RMSE |
| Linear Regression | 0.5445303 | 51661.44 |
| SVM Regressor | 0.8099098 | 33374.64 |

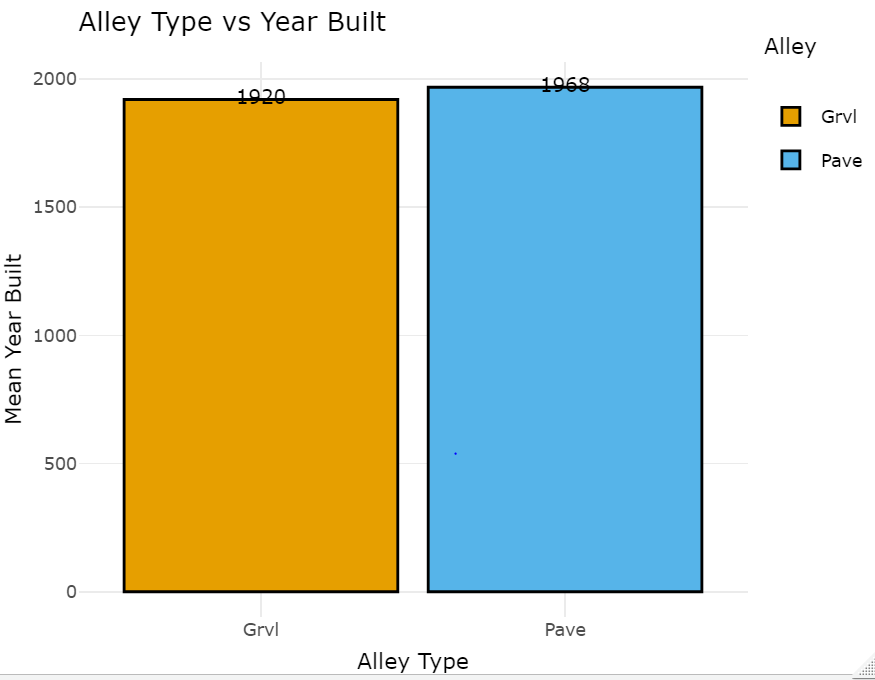
Results can clearly tell that SVM is producing better results as compare to linear regression.

**RESEARCH QUESTIONS:**

After careful analysis of the data set provided, we have come across different factors and relationships among the features. Depending on this, we have derived three questions.

**Question 1: What can we say about the feature Alley if the year of construction is known?**

Analysis: From the analysis we found that majority of the houses that have been built in the early 1900s still have an alley made of Gravel and the houses made in 1970s have Pavements. The following graph shows the relationship between alley type and year of construction of houses.

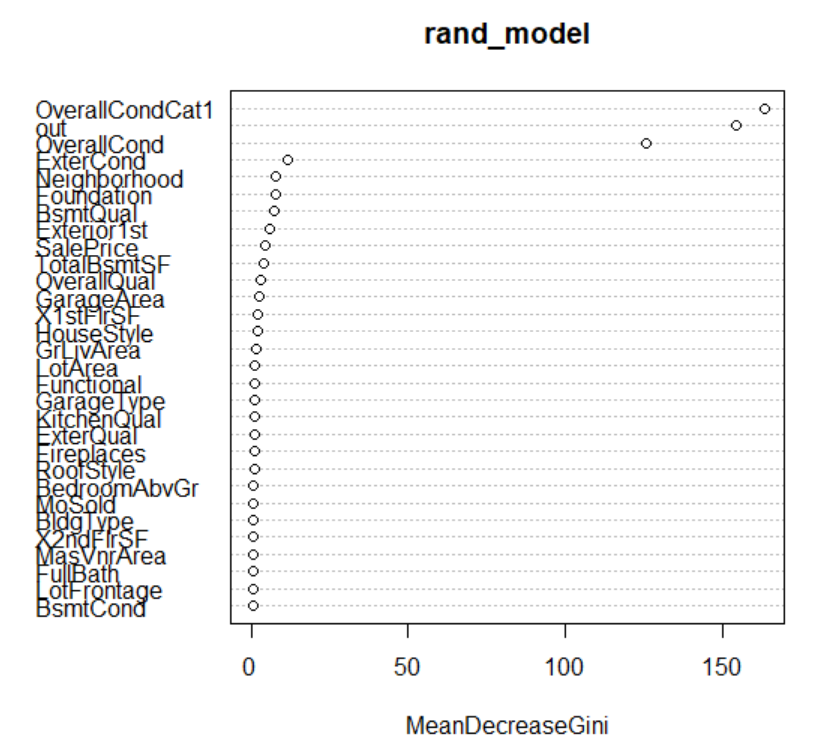


**Question 2: How the house prices depend on the pool quality?**

Analysis: Pool quality is classified among three categories i.e. Excellent, Fair and Good and we concluded that the house has an excellent pool has a very huge selling price. This is obvious since excellent pool quality can only be seen in new houses and, therefore new houses tend to have a high selling price too. Whereas, houses with fair and good pool quality have nearly equal prices.

**Question 3: Which feature is/are important in determining the price of the house?**

Analysis: To determine this we have followed the random forest method and calculated the variable importance i.e. which feature is contributing more in determining the house price. We can see that properties like Overall Condition, Exterior Condition, Neighborhood and Height of the Basement play a major role in deciding the price of the house.



**REFERENCES:**

[1] <https://ademos.people.uic.edu/Chapter12.html>

[2] <https://stats.stackexchange.com/questions/33240>

**Contributions:**

Every person has worked very hard in this project and have contributed equally. There were few tasks that required more thinking and less coding, and few required more programming, but the complexity was average. It would be great if we receive equal marks. Thank You.

1. Srikanth Reddy Gudibandi:

* Multiple Regression
* SVR

1. Anchal:

* Logistic Regression
* Missing Value Imputation

1. Sai Vinay:

* SVM and Resampling methods and EDA

1. Obinna:

* Data Cleaning, correlation analysis and part of Missing Value Imputation

1. Antash:

* Data Analysis and EDA

1. Bhima Dasari:

* EDA and Research Questions

**Appendix:**

##################### Importing the required packages ########################

library(ggplot2)

library(dplyr)

library(PerformanceAnalytics)

library(corrplot)

library(plot3D)

library(plotly)

library(caret)

################### Loading the dataset to R #######################

data=read.csv("C://Users//lenovo//Desktop//MA321-AS//house-data.csv",header = TRUE,sep = ",")

summary(data)

#This tells us about the missing value information for each variable in the dataframe.

NaDetails<-colSums(sapply(data,is.na))

# It can be observed that 4 features(Alley, PoolQC, Fence, MiscFeature) contain a lot of NAs(>80%), it is better to eliminate these features

# Since it is useless to impute them. However, 6(LotFrontage, MasVnrArea, BsmtQual, BsmtCond, GarageType, GarageCond) features contain

# Reasonable amount of NAs which can be imputed.

#We may not include the features containing huge missing values but we can definitely get some important information

#Remove the 4 features.

data1<-data[,-c(1,5,43,44,45)]

###### 1. Provide Numerical and Graphical summary of the data-set and comment on them. ########

##Getting the numerical columns

numeric\_vars<-names(data1)[which(sapply(data1,is.numeric))]

data1\_num<-data1[,c(numeric\_vars)]

##Finding the Correlations among the numerical variables.

cor\_num<-cor(data1\_num,method = "pearson",use = "complete.obs")

#Plot

#corr\_plot<- corrplot(cor\_num)

library(GGally)

ggcorr(cor\_num,label=TRUE)

# Correlation of all the variables w.r.t to the Sales Price.

corr\_mat\_wrt\_price<-as.matrix(sort(cor\_num[,"SalePrice"],decreasing = TRUE))

corr\_mat\_wrt\_price[2:6,] #Top 5 features that has a high positive correlation with the House Price.

#Usually big houses have a higher price and here it makes complete sense that a High Priced house will have a very good Quality with Big living area and

#garage area, a big basement area and finally a big First Floor in terms of area.

######### Neighborhood vs SalePrice #################

Neighborhood\_group<- data1 %>%

group\_by(Neighborhood) %>%

summarize(SalePrice=mean(SalePrice),.groups = 'drop')

Price\_wrt\_Neighbor<-data.frame(Neighborhood\_group)

Price\_wrt\_Neighbor=Price\_wrt\_Neighbor[order(-Price\_wrt\_Neighbor$SalePrice),] #ordering from High SalePrice to low.

df\_area\_price<-ggplot(data=Price\_wrt\_Neighbor[1:6,], aes(x=Neighborhood,y=SalePrice,fill=Neighborhood)) +

geom\_bar(stat="identity",color='black',position=position\_dodge())+

geom\_text(aes(label=SalePrice), vjust=-0.3, size=3.5)+

labs(x="Area ", y= "Price", title="Area vs Price Distribution")+

theme\_minimal()+

scale\_fill\_manual(values=c('#E69F00','#56B4E9',"blue","yellow","pink","black"))

ggplotly(df\_area\_price) #top 6 areas where the house prices are expensive.

############################ House Prices according to Foundation Type ########################

Foundation\_group<- data1 %>%

group\_by(Foundation) %>%

summarize(SalePrice=mean(SalePrice),.groups = 'drop')

Price\_wrt\_foundation<-data.frame(Foundation\_group)

Price\_wrt\_foundation=Price\_wrt\_foundation[order(-Price\_wrt\_foundation$SalePrice),]

df\_foundation\_price<-ggplot(data=Price\_wrt\_foundation[1:6,], aes(x=Foundation,y=SalePrice,fill=Foundation)) +

geom\_bar(stat="identity",color='black',position=position\_dodge())+

geom\_text(aes(label=SalePrice), vjust=-0.3, size=3.5)+

labs(x="Foundation Type ", y= "Price", title="Foundation type vs Price Distribution")+

theme\_minimal()+

scale\_fill\_manual(values=c('#E69F00','#56B4E9',"blue","yellow","pink","black"))

ggplotly(df\_foundation\_price) # Houses with Foundation type "Poured Concrete" tend to cost more that other foundation.

########################## Floor vs Foundation ##############################

FoundationVSFloor\_group<- data1 %>%

group\_by(HouseStyle,Foundation) %>%

summarize(count=n(),.groups = 'drop')

foundationVSfloor<-data.frame(FoundationVSFloor\_group)

#foundationVSfloor=Price\_wrt\_foundation[order(-Price\_wrt\_foundation$SalePrice),]

df\_foundationVSfloor<-ggplot(data=foundationVSfloor, aes(x=HouseStyle,y=Foundation,fill=Foundation)) +

geom\_bar(stat="identity",color='black',position=position\_dodge())+

geom\_text(aes(label=Foundation), vjust=-0.3, size=3.5)+

labs(x="Floor", y= "Foundation Type", title="Floor vs Foundation")+

theme\_minimal()+

scale\_fill\_manual(values=c('#E69F00','#56B4E9',"blue","yellow","pink","black"))

ggplotly(df\_foundationVSfloor)

#Obervations:

#1.It can be clearly observed that only Poured Concrete and Cinder Block have been used for as foundation in SLvl HouseType.

#2. All kinds of Foundations have been used in the construction of 1.5Fin and 2Story buildings.

#################################### Construction Year vs Price ##########################

Construction\_Year<- data1 %>%

group\_by(YearBuilt) %>%

summarize(SalePrice=mean(SalePrice),.groups = 'drop')

Const\_year<-data.frame(Construction\_Year)

Const\_year=Const\_year[order(-Const\_year$SalePrice),]

df\_yearvsPrice<-ggplot(data=Const\_year[1:50,], aes(x=YearBuilt,y=SalePrice)) +

geom\_bar(stat="identity",color='black',position=position\_dodge())+

geom\_text(aes(label=YearBuilt), vjust=-0.3, size=3.5)+

labs(x="Year of Construction ", y= "Price", title="Age of House vs Price of House")+

theme\_minimal()+

scale\_fill\_manual(values=c('#E69F00','#56B4E9',"blue","yellow","pink","black"))

ggplotly(df\_yearvsPrice)

#Observations:

#1. It can be observed that most of the houses made after 1980s have a high selliing price, and prices tend to increase higher for the houses made after 2005.

#2.A few houses made in early 1800s and late 1800s are also values very high, they might me antique houses.

##################################### 2. Divide the house Based on Overall Condition ################################

# Creating the new feature "OverallCondCat" with values as a "Poor", "Average" and "Good".

data1["OverallCondCat"]<-ifelse(data1["OverallCond"]>=1 & data1["OverallCond"]<4,"Poor",

ifelse(data1["OverallCond"]>=4 & data1["OverallCond"]<7,"Average",

ifelse(data1["OverallCond"]>=7 & data1["OverallCond"]<11,"Good","Unknown")))

table(data1["OverallCondCat"])# We can observe that 1130 Houses are Average, 299 are Good and 31 houses are in Poor Condition.

################################## Missing Value Imputation ########################

#Now it is time to impute the missing values(LotFrontage, MasVnrArea, BsmtQual, BsmtCond, GarageType, GarageCond).

#We will not perform a blind mean/median/mode imputation since the dataset also contains categorical values. We will perform a category wise imputation.

Lot\_data<-subset(data1,(!is.na(data1[,1]))) # Creating a new dataset where there are no NAs in the LotFrontage.

#Grouping the data accoring to the Neighborhood, and taking the mean values so that we can impute them with the NAs in the LotFrontage column.

LotFrontage<- Lot\_data %>%

group\_by(Neighborhood) %>%

summarize(lot\_mean = mean(LotFrontage), .groups = 'drop')

LotFrontage = data.frame(LotFrontage)

#In the below step we will fill all the NAs in the LotFrontage column with the mean values of the corresponding category.

data1$LotFrontage[is.na(data1$LotFrontage)]<-LotFrontage$lot\_mean[match(data1$Neighborhood,LotFrontage$Neighborhood)][which(is.na(data1$LotFrontage))]

#Reference: stackoverflow.com/questions/34697032/fill-in-missing-values-nas-with-values-from-another-dataframe-in-r

################################## Performing the same operation to all the numerical variables that have NAs #############

MasVnrArea\_data<-subset(data1,(!is.na(data1[,17])))

MasVnrArea\_group<- MasVnrArea\_data %>%

group\_by(Neighborhood) %>%

summarize(mas\_mean = mean(MasVnrArea), .groups = 'drop')

MasVnrArea\_group = data.frame(MasVnrArea\_group)

data1$MasVnrArea[is.na(data1$MasVnrArea)]<-MasVnrArea\_group$mas\_mean[match(data1$Neighborhood,MasVnrArea\_group$Neighborhood)][which(is.na(data1$MasVnrArea))]

############## Mode Imputation for Categorical Features ###############

table(data1["BsmtQual"]) # "TA" is the most common category

data1$BsmtQual[is.na(data1$BsmtQual)]<-"TA" # Mode imputation

#########################

table(data1["BsmtCond"]) # "TA" is the most common category

data1$BsmtCond[is.na(data1$BsmtCond)]<-"TA" # Mode imputation

#####################

table(data1["GarageType"]) # "Attchd" is the most common category

data1$GarageType[is.na(data1$GarageType)]<-"Attchd" # Mode imputation

#######################

table(data1["GarageCond"]) # "TA" is the most common category

data1$GarageCond[is.na(data1$GarageCond)]<-"TA" # Mode imputation

#write.csv(data1,"M:\\pc\\downloads\\Applied Stats" ,row.names = TRUE)

################## 2.(a): Fit a Logistic Regression that predicts the Over Condition ###################

#Check if the dataframe contains any duplicate values.

cat("The number of duplicate rows in the dataset are: ",nrow(data1)-nrow(unique(data1))) #No duplicate rows found

#str(data1)

data1$OverallCondCat1<-ifelse(data1$OverallCondCat=="Poor",1,ifelse(data1$OverallCondCat=="Average",2,ifelse(data1$OverallCondCat=="Good",3,4)))

#Converting the character columns(categorical variables) to as.factors.

data2<-data1%>%mutate\_if(is.character,as.factor)

#str(data2)

data2$OverallCondCat1<-as.factor(data2$OverallCondCat1)

data2<-data2[,-c(13,43)] # Eliminating the "Year Built" and "Year Sold" columns as they are not needed for Model Building.

#W have 3 levels in response variables, so we have to set a reference level, we are setting "Poor" as a the reference level

data2$out<-relevel(data2$OverallCondCat1,ref = "1")

#https://stats.stackexchange.com/questions/33240/

library(nnet)

set.seed(123)

splitting<-sample(2,nrow(data2), #### Re-sampling Method. Taking 80% as train and 20% as test.

replace=TRUE,

prob = c(0.8,0.2))

train\_lr<-data2[splitting==1,]

test\_lr<-data2[splitting==2,]

lr\_model<-multinom(out~.,data=train\_lr)

#-LotFrontage -LotArea -Condition1RRNe -Condition2RRAn -TotalBsmtSF -X1stFlrSF -X2ndFlrSF -GrLivArea -FunctionalSev -GarageArea -SalePrice

summary(lr\_model)

z<-summary(lr\_model)$coefficients/summary(lr\_model)$standard.errors

p<-(1-pnorm(abs(z),0,1))\*2

p ##P-value of the models

#Re-training after removing less significant variables p>5%

lr\_model<-multinom(out~.-LotFrontage -LotArea -TotalBsmtSF -X1stFlrSF -X2ndFlrSF -GrLivArea -GarageArea -SalePrice

,data=train\_lr)

summary(lr\_model)

#### Predict on Test Sample(Set) #####

prediction<-predict(lr\_model,test\_lr)

########## Confusion Matrix #############

confusion<-table(prediction,test\_lr$out)

# prediction 1 2 3

# 1 11 0 1

# 2 0 205 1

# 3 0 0 62

Error<-1-sum(diag(confusion))/sum(confusion)

#### Model Assessment ###########

Class\_wise<-confusion/colSums(confusion)

# prediction 1 2 3

# 1 1.000000000 0.000000000 0.090909091

# 2 0.000000000 1.000000000 0.004878049

# 3 0.000000000 0.000000000 0.968750000

#Our data is imbalanced, Class 2 is in majority and class 3 is minority.

#We can see that our model is performing good for class 1 and 2 but there

# is a slight mis-classification issues with class 3 but it is acceptable since the error is only 10%.

#

# library(ROCR)

# library(gplots)

#

# eval<-preformance(predicition(predict(lr\_model,train\_lr,type="prob"),train\_lr$OverallCondCat),"acc")

#

# predict(lr\_model,train\_lr,type="prob")

# predicition(predict(lr\_model,train\_lr,type="prob"),train\_lr$OverallCondCat)

#

######################## 2b. Support Vector Machine #######################

library(e1071)

svm\_model<-svm(OverallCondCat~.,data=train\_lr)

summary(svm\_model)

########## Prediction #############

svm\_predict<-predict(svm\_model,test\_lr)

#### Confusion Matrix ###########

svm\_cm<-table(svm\_predict,test\_lr$OverallCondCat)

# svm\_predict\_sig Average Good Poor

# Average 205 2 11

# Good 0 62 0

# Poor 0 0 0

svm\_Error<-1-sum(diag(svm\_cm))/sum(svm\_cm) # 4% error, but all the "Poor" Conditioned houses are being classified as "Average".

###### Sigmoid Kernal #########

#Lets change the kernal to Sigmoid and see the results

svm\_model\_sig<-svm(OverallCondCat~.,data=train\_lr,

kernal="sigmoid",cv=10) # Resampling Method: K-fold Cross validation.

summary(svm\_model\_sig)

########## Prediction #############

svm\_predict\_sig<-predict(svm\_model\_sig,test\_lr)

#### Confusion Matrix ###########

svm\_cm\_sig<-table(svm\_predict\_sig,test\_lr$OverallCondCat)

# svm\_predict\_sig Average Good Poor

# Average 205 2 11

# Good 0 62 0 #### No change in result, the Poor category is performing bad.

# Poor 0 0 0

################## 3(a). Multiple Linear Regression ###########################

library(caTools)

# The data what we have pose have categorical variables, we can create dummy and include them in the dataset.

#But we would rather select the highly correlated numerical variables with "SalePrice" as the Independant variables.

corr\_mat\_wrt\_price[2:7,] # this gives us the top 6 variables that are correlated with the "SalePrice".

lin\_model<-lm(SalePrice~ OverallQual+GrLivArea+GarageArea+TotalBsmtSF, data = train\_lr)

summary(lin\_model)

#X1stFlrSF and FullBath have been removed since the p value for these variables are higher than 5%.

############## Multi-Collinearity ##########

library("car")

vif(lin\_model) # Theoretically it is okay to to have VIF<10. We have less than 5 here.

########## Other Assumptions ##########

plot(lin\_model)

############### The residual vs fitted line- Plot1: ##########

# The first plot shows us the residuals vs fitted line plot. The red line is horizontal to the 0 on y axis.

#The data points shouldn't have any patterns like a sign curve or a parabola curve, in this case, we have a

#linear pattern. All the points are evenly distributed around the zero and is homoscedastic. Our first assumption passed.

############## The Normality Assumption- Plot2: ##########

#Most of the residuals are normally distributed along the line at 45degrees. Our second assumption has passed.

############## The scale location plot ###############

#The red line is supposed to be straight and points should be equally distributed around it. Here the line is not straight

#but the points are evenly distributed. We can say that the data is not homoscedastic. The plot of the predicted values

#and the square rooted standardized residuals looks good. Hence our assumption has passed.

############ The Cooks Distance ########################

#Influencial outliers destroy the models, but here we have 3 major outliers, they cannot impact the model much.

########### Prediction ################

linear\_pred<-predict(lin\_model,test\_lr)

#R\_Square<-R2(as.integer(linear\_pred),(test\_lr$SalePrice))

R\_Square<- 1-((sum((linear\_pred-test\_lr$SalePrice)^2))/(sum((test\_lr$SalePrice-mean(test\_lr$SalePrice))^2)))

R\_Square

RMSE\_score<-RMSE(linear\_pred,test\_lr$SalePrice)

RMSE\_score

#

# #################### SVR #####################

#

SVR\_model<-svm(SalePrice~ OverallQual+GrLivArea+GarageArea+TotalBsmtSF,data = train\_lr)

summary(SVR\_model)

svr\_predict<-predict(SVR\_model,test\_lr)

################# Evaluations #############

R\_Square\_svm<- 1-((sum((svr\_predict-test\_lr$SalePrice)^2))/(sum((test\_lr$SalePrice-mean(test\_lr$SalePrice))^2)))

R\_Square\_svm

svr\_MSE<-(1/nrow(test\_lr))\*sum((svr\_predict-test\_lr$SalePrice)^2)

Rmse\_svr=(svr\_MSE)^0.5

Rmse\_svr

############################## 3B Resampling Methods ####################################

#1. Validation Set 80-20 Split

#2. K-Fold Crossvalidation.

#Both of the methods have been incorporated in the classification techniques in question 2.

######################### 4. Research Question #######################

# alley<-data[which(data["Alley"]=="Grvl" | data["Alley"]=="Pave"),]

# PoolQC<-data[which(data["PoolQC"]=="Ex" | data["PoolQC"]=="Fa" | data["PoolQC"]=="Gd" |data["PoolQC"]=="TA") ,]

# Fence<-data[which(data["Fence"]=="GdPvr" | data["Fence"]=="MnPrv" | data["Fence"]=="GdWo"| data["Fence"]=="MnWw") ,]

# MiscFeature<-data[which(data["MiscFeature"]=="Gar2" | data["MiscFeature"]=="Othr" | data["MiscFeature"]=="Shed" | data["MiscFeature"]=="Tenc"),]

#

##### 1. It is quite evident that majority of the house that have been built in the early 1900s still have an alley made of Gravel and the houses made in 1970s have Pavements.

alley\_group<- alley %>%

group\_by(Alley) %>%

summarize(YearBuilt=mean(YearBuilt),.groups = 'drop')

alley\_group<-data.frame(alley\_group)

alley\_plot<-ggplot(data=alley\_group, aes(x=Alley,y=round(YearBuilt,0),fill=Alley)) +

geom\_bar(stat="identity",color='black',position=position\_dodge())+

geom\_text(aes(label=round(YearBuilt,0)), vjust=-0.3, size=3.5)+

labs(x="Alley Type ", y= "Mean Year Built", title="Alley Type vs Year Built")+

theme\_minimal()+

scale\_fill\_manual(values=c('#E69F00','#56B4E9'))

ggplotly(alley\_plot)

##### 2. It can be seen that has a house that has an excellent pool has a very huge selling price. This is obvious since excellent pool quality can only be seen

## in new houses and, new houses tend to have a high selling price too.

PoolQC\_group<-PoolQC%>%

group\_by(PoolQC)%>%

summarize(SalePrize=mean(SalePrice),.groups="drop")

PoolQC\_group<-data.frame(PoolQC\_group)

PoolQC\_plot<-ggplot(data=PoolQC\_group, aes(x=PoolQC,y=SalePrize,fill=PoolQC)) +

geom\_bar(stat="identity",color='black',position=position\_dodge())+

geom\_text(aes(label=SalePrize), vjust=-0.3, size=3.5)+

labs(x="Pool QUALITY ", y= "Mean House Selling Price", title="Pool qualuty vs House Selling Estimate")+

theme\_minimal()+

scale\_fill\_manual(values=c('orange','yellow','blue'))

ggplotly(PoolQC\_plot)

#################### Which feature are important in determining the quality/price of the house. #############################

library(randomForest)

rand\_model<-randomForest(OverallCondCat~.,data=data2,ntree=30)

#importance(rand\_model)

varImpPlot(rand\_model)

# We can see that properties like Overall Condition, Exterior COndition, Neighborhood, Height of the Basement play a major role in deciding the quality/price of the house.