TITLE:

Housing Price Dataset

Contents

[**Introduction** 3](#_Toc99359493)

1[**.** **Preliminary Analysis** 1](#_Toc99359496)

[**1.** **Introduction** 3](#_Toc99359495)

[**2.** **Preliminary Analysis of the dataset** **.**](#_Toc99359496)

[**3** **Data Importation** **Error! Bookmark not defined.**](#_Toc99359497)

[**3** **Dealing with missing values** **Error! Bookmark not defined.**](#_Toc99359497)

[**3** **Handling Collinearity** **Error! Bookmark not defined.**](#_Toc99359497)

[**3.** **Analysis** **Error! Bookmark not defined.**](#_Toc99359498)

[**4.** **Selecting Best Model** **Error! Bookmark not defined.**](#_Toc99359499)

[**5.** **Conclusion** **Error! Bookmark not defined.**](#_Toc99359501)

[**6.** **References** **Error! Bookmark not defined.**](#_Toc99359502)

[**Appendices** **Error! Bookmark not defined.**](#_Toc99359503)

[**Appendix I: R-code with Comments** **Error! Bookmark not defined.**](#_Toc99359504)

[**Appendix II: New Variable Name and Indicator Name** **Error! Bookmark not defined.**](#_Toc99359505)

[**Appendix III: Acronyms/Abbreviations** **Error! Bookmark not defined.**](#_Toc99359506)

Figures

[**Figure 1. .**](file:///E:\Project\Reports\Final%20Report.docx#_Toc448163830)

[**Figure 2.**  **Error! Bookmark not defined.**](#_Toc448163831)

[**Figure 3. .** **Error! Bookmark not defined.**](file:///E:\Project\Reports\Final%20Report.docx#_Toc448163832)

**Abstract**

In recent times, one of the most valuable digital commodities is “Data” and a huge amount of this resource, is produced every millisecond and captured in both structured and unstructured forms. As we capture this stream of data it is imperative to explore and exploit the precious content therein through statistical analyses, machine and deep learning algorithms and Computer Science techniques to enable us to automate these processes. In view of this, of the myriads of datasets available to us, in this study a housing dataset will be explored both to carryout inferential and predictive analysis (using some powerful machine learning algorithms). It is worth noting that all analysis and predictions have been implemented in R software environment. Consequently, we have deployed and carried out interesting things on the housing dataset. In this case study, we have explored and performed Exploratory Data Analysis (EDA) on the dataset by investigating the trends of distributions in the features (categorical/non-numerical and numerical) employing readily available data visualization tools in R. We have checked for and handled missing values using proper statistical intuition and techniques. Having said that, the housing dataset embeds interesting features, hence, we have deployed machine learning models to predict the overall housing conditions of the houses as well as the house prices. In addition, the test errors after the machine learning algorithm have been fitted on the training data set have been estimated using appropriate statistical re-sampling methods.

1. **Introduction**

Available for this case study is the “housing dataset” containing valuable information, in terms of feature variables collected and customized for a housing data. This dataset consists of 1460 observation or records (of different houses captured) and 51 feature set of the houses. The data set house different data types, in other words there are 28 features with character strings while the other 23 features are of type doudble. A panoramic view on the dataset and using a special function in R it is suggestive that the dataset contains quite several missing values, however details on how the missing values are handled is discussed in the Exploratory Data Analysis sections below. To gain some useful insight and information about the dataset before applying machine learning algorithms we have analysed the dataset using descriptive statistical and visualization tools on the some of the features as well as check for correlation of these features with one of the response variables (which in this case, the “Sales Price” of the houses). Going a step further, the ranks recorded in the overall condition feature is transformed into three categories: “Poor” if it ranks between 1 to 3, “Average” if rank is between 4 and 6, and “Good” for ranks in the range from 7 and 10. On top of that, appropriate machine learning algorithms were chosen to perform predictions on the overall condition of the houses as well as the house prices. Subsequently, two re-sampling methods were used to estimate the errors of predictions on the test data after applying appropriate algorithms to fit the training data. Full details on the predictions and model evaluation are fully discussed in the subsequent sections below.

**Note:** The statistical summary of the dataset as outputted by R software using the function is in the Figure A1 in the appendix section.

1.1 **Exploratory Data Analysis**

Before feeding some of our features as predictor variables into the machine learning algorithms, it is expedient to carry out some descriptive analysis to check for some interesting trends and patterns in the data set using some data visualization packages in R.

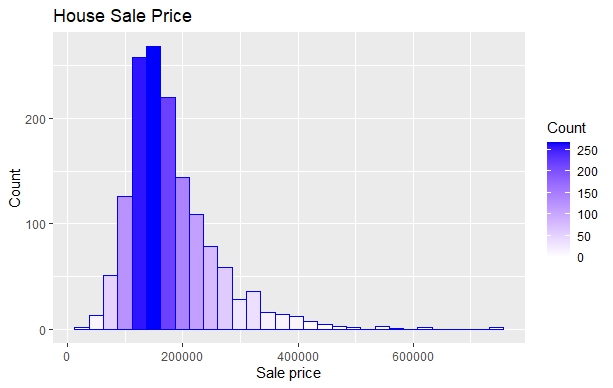
In Figure 1.0 below, we can visually and easily observe the features with missing values. The features in the figure have been ranked relative to how their respective missing values can impact statistical analyses and machine learning prediction of the response variables. Furthermore, for the continuous variables in the dataset the missing values were filled with zeros since the number of missing values associated with these features are very minimal as depicted in the image below.

Bar chart

Description automatically generated with low confidence

**Figure 1.0: Missing values**

Interestingly, most of the house prices are within the range of £100,000 – £250,000 while very few houses are above £400,000 assuming that the prices are in GBP currency. This is quite intuitive as only very few rich people (with high-income status) can afford houses with higher prices in contrast to many people who are in the middle-income cadre. This is obvious in figure 1.2.

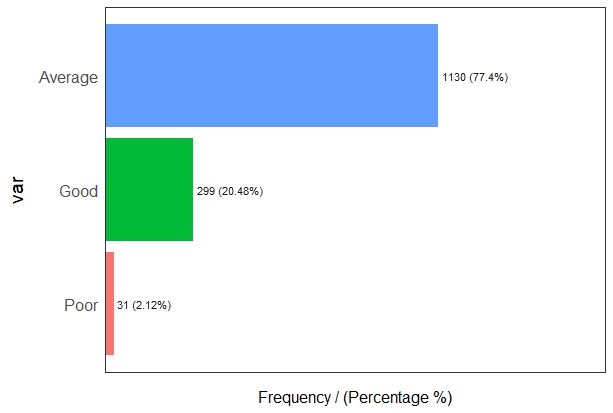


**Figure 1.1: Frequency distribution of the house prices**

In addition, we also observed that the sales price of the 1story and 2story house style are high when compared to the other styles and there are no outliers in 1.5Unf house style and it has low sales price as illustrated in the boxplots in figure A2, in the Appendix section. Other relationships of the predictor variable in relation to the house sales price are illustrated in the images in the Appendix section.

**2.0 Model Selection and Overall Condition Prediction:**

Having dealt with the missing values and caried out descriptive analysis on some of the features in the dataset to see how they are distributed with the sales price of the houses. The feature, Overall condition, which ranks the general status of the houses on a scale of 1 to 10 was further transformed to “poor” for conditions between 1 to 3, “Average” for conditions between 4 to 6, and “Good” for conditions between 7 to 10 using the if/else code in the appendix section of Question 2. The distribution of the conditions is summarized below in figure 2.0.



**Figure 2.0: Frequency Distribution of the Overall Condition of the Houses**

2.1a Logistic Regression Model

Having trained and tested (used in prediction on the house dataset) the Logistic regression model, the prediction accuracy on the training data was estimated at 0.7842466 (about 78%) which indicates how well the model has learned from the training process and the measure of strength on how well the overall house condition (response variable) variance is explained by the house features in the data set. Also, the (explained variance) is over 81% which gives a high proportion of variance as well as the response strength of the response variable (overall house condition). The figure below shows closeness of fitness in terms of the test accuracy. The Details of the summary are given in the Question 2 appendix section on Figure B5 and B6 respectively.



**Figure 2.1: Values obtained after fitting the Logistic Regression Model**

2.1b Decision Tree

We have also opted for Decision Tree algorithm in predicting the overall conditions of the houses as this model can be deployed for both classification and regression tasks. While Decision tree has been deployed on different data sets in various domains with high-rate prediction on medium sized data sets, it is also capable of handling numeral and categorical data, and multi-output problems with little preparation on the datasets. Amongst the decision tree in the pictorial form can provide interpretability of the model. That said, we have thought it wise to exploit its advantage in this study. It will suffice that we compared the model accuracy of Linear Discriminant Analysis (LDA) and SVM, but the SVM proved better. Details of the codes are in the Question 2 appendix section for reference.

**Prediction Result for Overall Condition for the Decision Tree**

From the figure (figure 2.2) below, it can be observed that the decision tree model performed well as the misclassification error rate of the model is quite low of 0.1955 and a relatively high residual mean deviance of a little above 78% which is suggestive of how well the model predicted the response variable which in this case, the Overall Condition of the houses. The confusion matrix also gives an illustrative perspective of the model’s performance. The confusion matrix and the decision tree plot are figures B1 and B2 respectively in the appendix section of Question 2.

Text

Description automatically generated with medium confidence

**Figure 2.1: Classification Result/Report of the Decision Tree Model**

**3.0 House Price Prediction and Model Selection**

3.1a First Model Justification: Random Forest

In this instance we have chosen random forest as one of the algorithms to predict the house prices. Random forest algorithms have the capacity to be applied to both regression and classification problems, hence, a choice in our selection. Overtime, random forest model has also proven its higher prediction power and better accuracy over decision tree models as well as reducing potential overfitting in decision tree. Moreover, the results have high interpretability. In other words, the results obtained from this model is easy to interpret.

3.1b Prediction Result on the House Prices

Having trained and fitted the random forest model using the training dataset after splitting on a ratio of (80% for training and 20% for testing), the fitted model was then deployed on the test and achieved (explained variance) value of 0.8580437. This values proves that over 85% of the house sales prices variance is explained by the random forest model’s feature variables. The values of the RMSE (Root Mean Square Error), MAE (Mean Absolute Error) including the value is in the Question 3 appendix section of figure C1 as well as the R codes for the model. In the figure below (figure 3.0), the fitted line passes through a lot of points which shows the accuracy of the predictions as outputted by the model.

**Chart, scatter chart

Description automatically generated**

**Figure 3.0: Prediction accuracy graph of the fitted Random Forest model**

3.2a Second Model Justification: Support Vector Machines (SVM)

We have also selected the Support Vector Machine Model (SVM) of the various classification models available. We have chosen this model by its vantage point in using high dimensionality power for classification and regression. In comparison with other classes SVM provides a clear separation of classes by using distinct margins among classes as well as flexible support vectors for regression tasks. Moreover, SVM is memory efficient and works very well on medium sized dataset such as the housing data set.

3.2b Prediction Result on the House Prices

When the SVM model was applied accordingly the strength and performance of the model has been pictorially detailed in the figure (figure 3.1) below. It can be observed that most of the points passes through the fitted line as well as close it. Hence, the (explained variance) was 0.780994 (which is estimated at about 78%) gives a string indication on how well the response variable’s variance is explained and determined by the house features when this model is been fitted to the data set. The estimated values of the (explained variance), RMSE and MAE are on figure C3 in the Question 3 appendix section.

Chart, scatter chart

Description automatically generated

**Figure 3.1: Prediction accuracy graph of the fitted SVM model**

3.3 Resampling Methods

Resampling Methods used to test the random forest model are Cross validation and Bootstrap. The Root Mean Squared Error (RMSE) for both methods of resampling test error are quite close (Cross validation: 30480.81 and Bootstrap: 30984.7). These estimations are indicative of the fact that the model will generalize well on other unseen data. Included in the Question 3 Appendix section are the codes and figure (Figure C2) for the results of resampling model error test for reference.

Secondly, the resampling method used for the support vector machine is RMSE resampling method. The (explained variance) on how well the SVM model performed is over 78% which is suggestive that the model performed well in its prediction of the response variable. Details of the values of RMSE and MAE are in the appendix section below.

**4.0 Research Question**

Research Question: “Effect of Stratified Sampling method on House Price Value”

Taking a panoramic view on the dataset and having applied different machine learning algorithms on the housing dataset, it would be interesting to further investigate how different sampling methods (Simple Random Sampling, Systemic Sampling, Stratified Sampling, Cluster Sampling) can enhance the performance of the machine learning algorithms deployed so far or if it will otherwise impact the performance negatively.

**Conclusion**

Finally, having performed various analysis on the dataset set using descriptive statistics as well as selecting and fitting different models on the dataset, we can make general comments regarding how the models performed as well as suggestions for future study in a similar study. In the first part of the prediction exercise for the overall condition of the houses, the performances of the two models suggest that either of the models would do the job if proper data preparation were done and the model parameters are properly tuned. However, in the case of the house price prediction, it is clear that the random forest performed better by its predictions than the support vector machine. Hence, for model deployment for house price prediction based on this case study the random forest model will be advised. Moreover, if we could gather more records for this dataset, we could also fit a neural networks model to see how well the predictions will be. This could be explored in future similar case study.

**Group Contributions:**

Abiodun Araba:Exploratory Data Analysis and Machine Learning model selection and implementation

Adeniji Rahmon Olusegun**:** Exploratory Data Analysis, Machine Learning Model Testing and report compilation.

Alegimenlen Abdumalik**:** Report review, research question and approach methodology.

Appendix

**Question 1**

#installing necessary packages for plotting, machine learning algorithms, missing plots

install.packages("ggplot2")

install.packages("GGally") #this

install.packages("DataExplorer")

install.packages("GGally")

install.packages("funModeling") #for plotting categorical variables

install.packages("roperators") #provides reassignment of logical operators

install.packages("caret") #for regression and classification model training

install.packages("tree") #for decision tree classifier

#the plot\_mising function is in the domain if this library

library(DataExplorer)

#helps with the graphics for data visualization

library("ggplot2")

#helps with the platform for the pair plots

library("GGally")

#this is a powerful library that helps with data manupulation

library(dplyr)

#powerful library that houses data science packages

library(tidyverse)

#helps to improve code readability and maintenance. Also has direct implication to the pipe operator

library(magrittr)

#for neural networks

library(nnet)

provides reassignment of logical operators

library(roperators)

#for decision tree classifier

library(tree)

#it aids the fast reading of different file format

library(readr)

#this provides consistent tools in working with functions and vectors

library(purrr)

#useful and provides strong tools required for classification model predictions

library(ipred)

library(ISLR)

#aids the functionality of data interaction, whilst providing improved shiny-based front-end

library(ExPanDaR)

#helps to gain access to functions during R session

library(MASS)

#powerful library for machine learning classification and regression

library(rpart)

#this library provides access to the random forest machine learning classification tools

library(randomForest)

#this also provides a powerful package for classification and well suited for small to moderate sized data

library(ada)

#aidsplotting categorical variables

library(funModeling)

#loading the data into R

df <- df <- read\_csv("C:\\Users\\840-G2\\Downloads\\house-data.csv")

#this function gives an overview, statistics and data type of the features

summary(df)

**Figures and respective codes**

#this function pplots the missing values in the dataset by assigning percentages

plot\_missing(

df,

group = list("Excellent" = 0.0, "Good" = .01, "Ok" = .05, "Bad" = 1),

missing\_only = FALSE,

geom\_label\_args = list(),

title = NULL,

)

#this is the code for the sales price frequency distribution in for figure 1.1

Code for Figure 1.1

ggplot(data=df, aes(SalePrice)) +

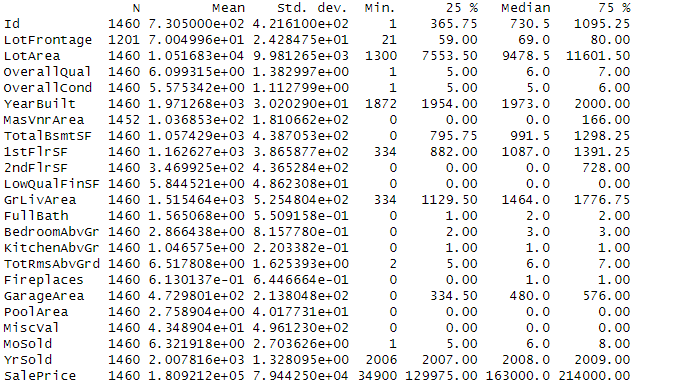
geom\_histogram(col="blue", aes(fill=..count..)) +

scale\_fill\_gradient("Count", low="white", high="blue") +

labs(y = "Count", x = "Sale price", title = "House Sale Price")

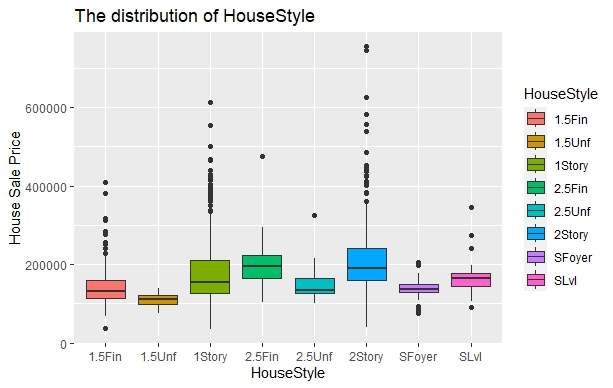
summary(SalePrice)

Figure A1:



Code A1

Figure A2:



Code A2

#this is the code for The distribution of House Style vs. House Sale Price using the ggplot and gg boxplot function

ggplot(data=df, aes(y= SalePrice, x=HouseStyle, fill=HouseStyle) ) +

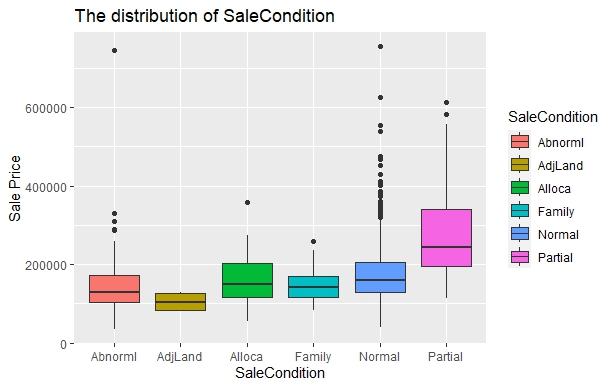
geom\_boxplot() +

ggtitle("The distribution of HouseStyle") +

ylab("House Sale Price") +

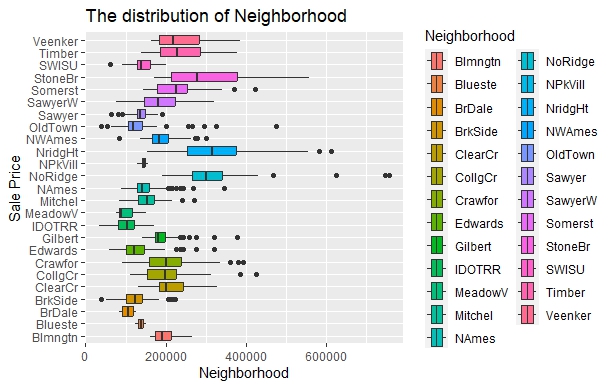
xlab("HouseStyle")

Figure A3:



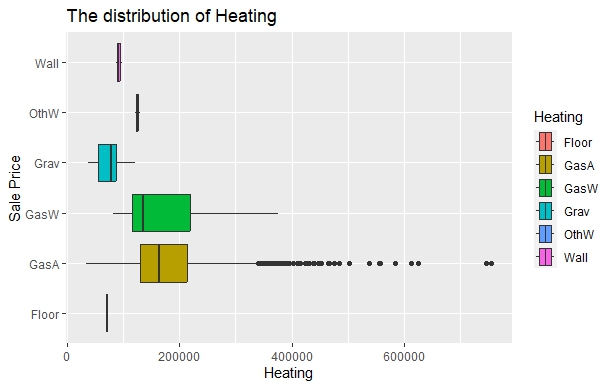
Code A3:

Figure A4:



Code A4

Figure A5:



Code A5

#this is the code for The distribution of Heating vs. House Sale Price using the ggplot and gg boxplot function

ggplot(data=df, aes(x= SalePrice, y=Heating, fill=Heating ) ) +

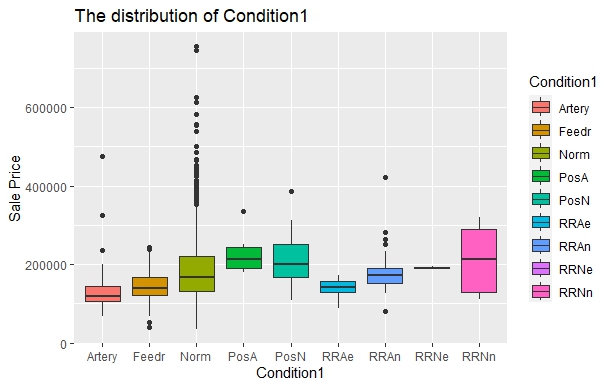
geom\_boxplot() +

ggtitle("The distribution of Heating") +

ylab("Sale Price") +

xlab("Heating")

Figure A6:



Code A6

#this is the code for The distribution of Condition1 vs. House Sale Price using the ggplot and gg boxplot function

ggplot(data=df, aes(y= SalePrice, x=Condition1, fill=Condition1) ) +

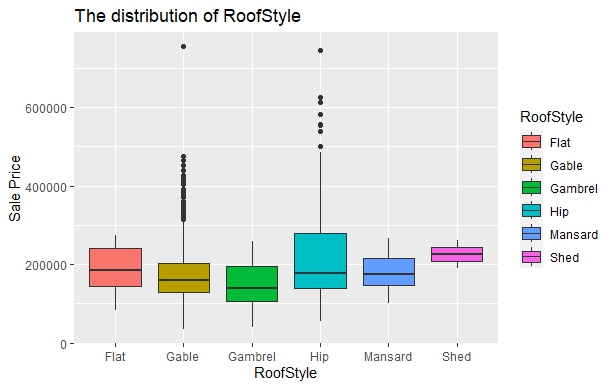
geom\_boxplot() +

ggtitle("The distribution of Condition1") +

ylab("Sale Price") +

xlab("Condition1")

Figure A7:



Code A7

#this is the code "The distribution of Roof Style" vs. House Sale Price using the ggplot and gg boxplot function

ggplot(data=df, aes(y= SalePrice, x=RoofMatl, fill=RoofMatl ) ) +

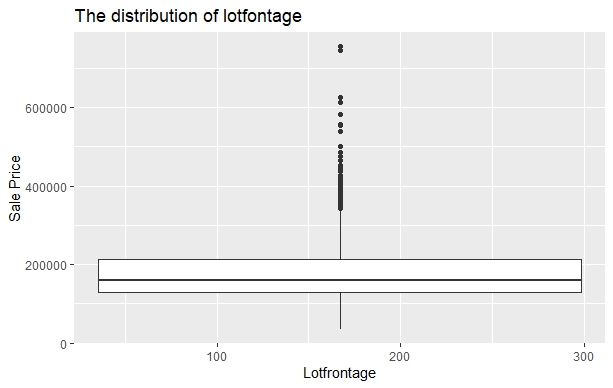
geom\_boxplot() +

ggtitle("The distribution of RoofMatl") +

ylab("Sale Price") +

xlab("RoofMatl")

Figure A8:



Code A8

#this is the code "The distribution of lotfontage " vs. House Sale Price using the ggplot and gg boxplot function

ggplot(data=df, aes(y= SalePrice, x=LotFrontage, fill=LotFrontage) ) +

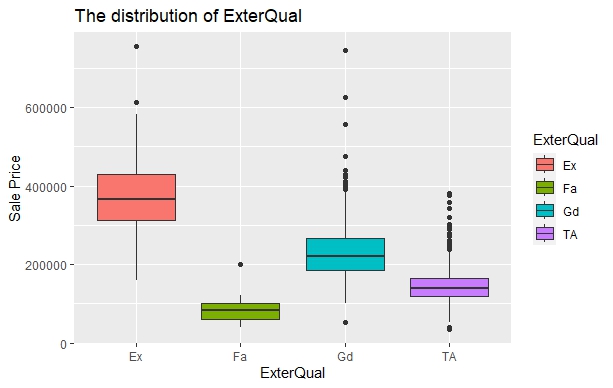
geom\_boxplot() +

ggtitle("The distribution of lotfontage") +

ylab("Sale Price") +

xlab("Lotfrontage")

Figure A9:



Code A9

#this is the code The distribution of ExterQual vs. House Sale Price using the ggplot and gg boxplot function

ggplot(data=df, aes(y= SalePrice, x=ExterQual, fill=ExterQual ) ) +

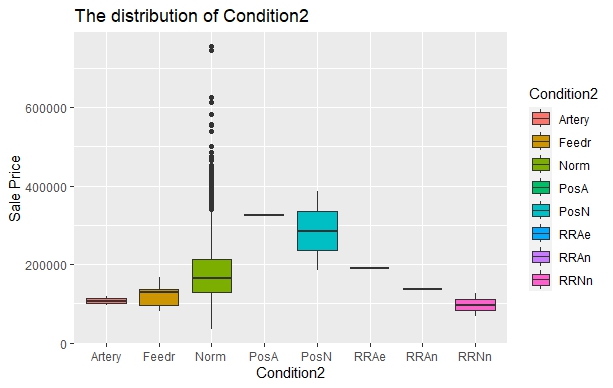
geom\_boxplot() +

ggtitle("The distribution of ExterQual") +

ylab("Sale Price") +

xlab("ExterQual")

Figure A10:



Code A10

#this is the code The distribution of Condition2 vs. House Sale Price using the ggplot and gg boxplot function

ggplot(data=df, aes(y= SalePrice, x=Condition2, fill=Condition2) ) +

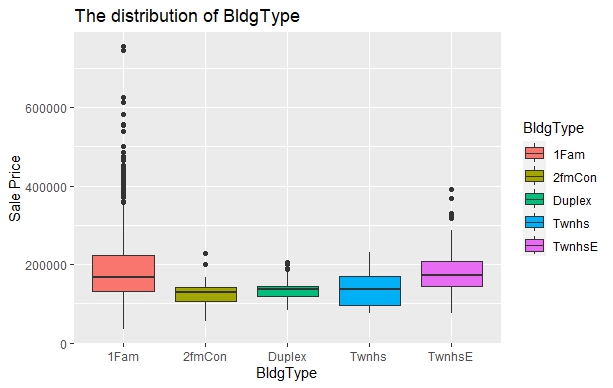
geom\_boxplot() +

ggtitle("The distribution of Condition2") +

ylab("Sale Price") +

xlab("Condition2")

Figure A11:



Code A11

#this is the code The distribution of BldgType vs. House Sale Price using the ggplot and gg boxplot function

ggplot(data=df, aes(y= SalePrice, x=BldgType, fill=BldgType) ) +

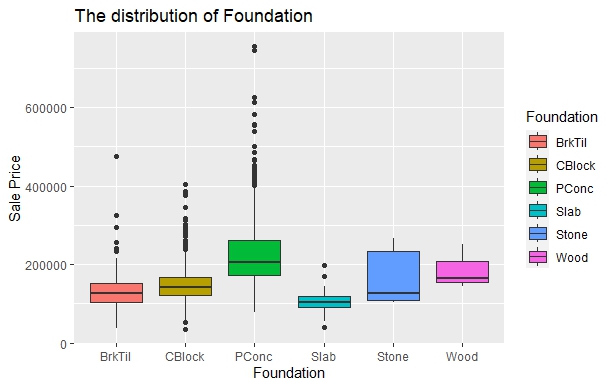
geom\_boxplot() +

ggtitle("The distribution of BldgType") +

ylab("Sale Price") +

xlab("BldgType")

Figure A12:



Code A12

#this is the code The distribution of Foundation vs. House Sale Price using the ggplot and gg boxplot function

ggplot(data=df, aes(y= SalePrice, x=Foundation, fill=Foundation) ) +

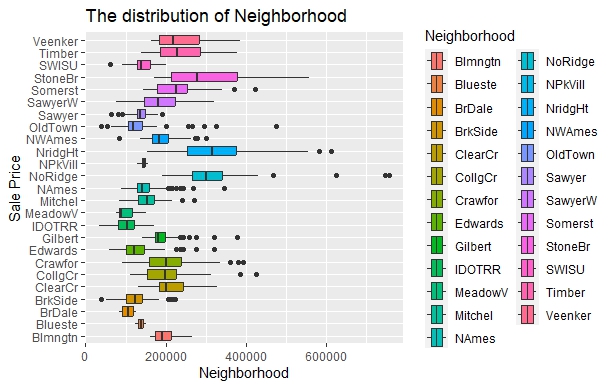
geom\_boxplot() +

ggtitle("The distribution of Foundation") +

ylab("Sale Price") +

xlab("Foundation")

Figure A13:



Code A13

#this is the code The distribution of Neighbourhood vs. House Sale Price using the ggplot and gg boxplot function

ggplot(data=df, aes(y= Neighborhood, x=SalePrice, fill=Neighborhood) ) +

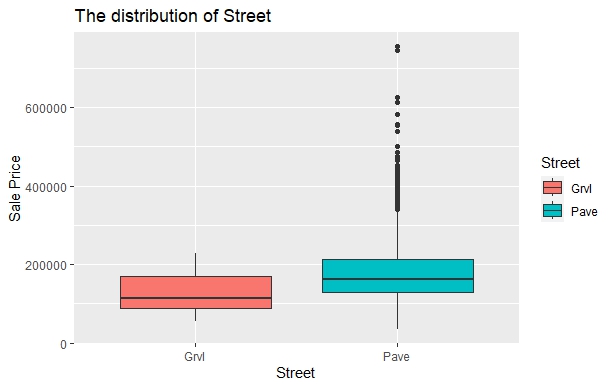
geom\_boxplot() +

ggtitle("The distribution of Neighborhood") +

ylab("Sale Price") +

xlab("Neighborhood")

Figure A14:



Code A14

#this is the code The distribution of Street vs. House Sale Price using the ggplot and gg boxplot function

ggplot(data=df, aes(y= SalePrice, x=Street, fill=Street) ) +

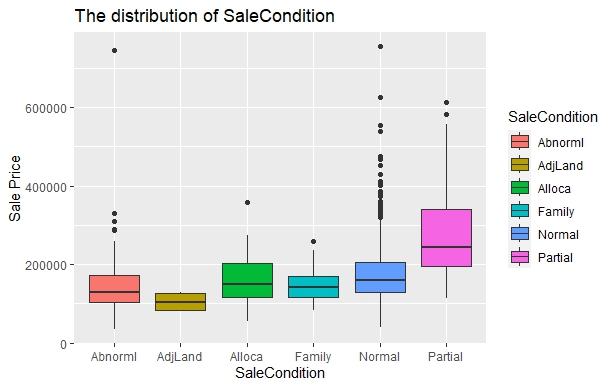
geom\_boxplot() +

ggtitle("The distribution of Street") +

ylab("Sale Price") +

xlab("Street")

Figure A15:



Code A15

#this is the code The distribution of Sale Condition vs. House Sale Price using the ggplot and gg boxplot function

ggplot(data=df, aes(y= SalePrice, x=SaleCondition, fill=SaleCondition) ) +

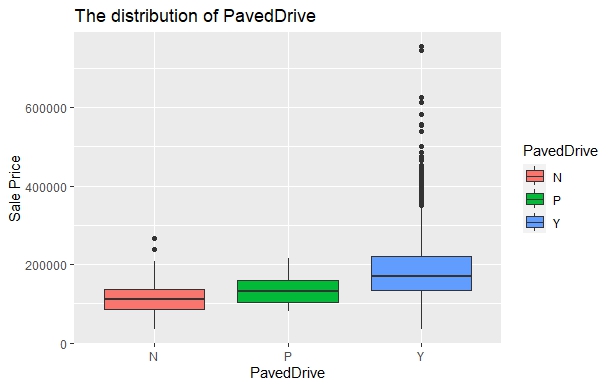
geom\_boxplot() +

ggtitle("The distribution of SaleCondition") +

ylab("Sale Price") +

xlab("SaleCondition")

Figure A16:



Code A16

#this is the code The distribution of Pave Drive vs. House Sale Price using the ggplot and gg boxplot function

ggplot(data=df, aes(y= SalePrice, x=PavedDrive, fill=PavedDrive) ) +

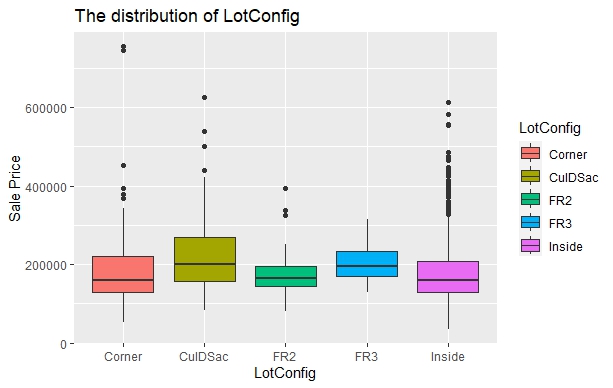
geom\_boxplot() +

ggtitle("The distribution of PavedDrive") +

ylab("Sale Price") +

xlab("PavedDrive")

Figure A17:



Code A17

#this is the code The distribution of LotConfig vs. House Sale Price using the ggplot and gg boxplot function

ggplot(data=df, aes(y= SalePrice, x=LotConfig, fill=LotConfig) ) +

geom\_boxplot() +

ggtitle("The distribution of LotConfig") +

ylab("Sale Price") +

xlab("LotConfig")

**Question 2:**

**#**Organised the OverallCond to Average, Good and Poor using the if/else statement

house\_df$OverallCond <- as.factor(ifelse(house\_df$OverallCond >= 1 & house\_df$OverallCond <= 3, 'Poor',

ifelse(house\_df$OverallCond >= 4 & house\_df$OverallCond <= 6, 'Average', 'Good')))

#gives the number of houses corresponding to the the overall condition in terms of Average, Good, Poor

summary(house\_df$OverallCond)

#produces the frequency distribution plot of the overall conditions of the houses

freq(house\_df$OverallCond)

**Question 2a:**

#dataset loading

house\_df\_new<-read\_csv(file.choose("C:/Users/ADENIJI RAHMON O/Download/house-data.csv"))

#structure of data

dim(house\_df\_new)

class(house\_df\_new)

str(house\_df\_new)

#the count of missing values in the data

table(is.na(str(house\_df\_new)))

#replacing NA values according to description

house\_df\_new$Alley[is.na(house\_df\_new$Alley)] <- "No alley"

house\_df\_new$Fence[is.na(house\_df\_new$Fence)] <- "No fence"

house\_df\_new$BsmtQual[is.na(house\_df\_new$BsmtQual)] <- "No basement"

house\_df\_new$GarageCond[is.na(house\_df\_new$GarageCond)] <- "No garage"

house\_df\_new$GarageType[is.na(house\_df\_new$GarageType)] <- "No garage"

house\_df\_new$MiscFeature[is.na(house\_df\_new$MiscFeature)] <- "None"

house\_df\_new$PoolQC[is.na(house\_df\_new$PoolQC)] <- "No pool"

house\_df\_new$BsmtCond[is.na(house\_df\_new$BsmtCond)] <- "No basement"

table(is.na(house\_df\_new))

house\_df\_new<-na.omit(house\_df\_new)

dim(house\_df\_new[sapply(house\_df\_new, is.character)])

dim(house\_df\_new[sapply(house\_df\_new, is.integer)])

data[sapply(house\_df\_new, is.character)] <- lapply(house\_df\_new[sapply(house\_df\_new, is.character)], as.factor)

class(house\_df\_new$OverallCond)

#replacing the numerical values in "OverallCond" as factors based on conditions

house\_df\_new$OverallCond\_predict<-ifelse(house\_df\_new$OverallCond >=1 & house\_df\_new$OverallCond <=3, "Poor", "Good")

house\_df\_new$OverallCond\_predict<-ifelse(house\_df\_new$OverallCond >=4 & house\_df\_new$OverallCond <=6, "Average",house\_df\_new$OverallCond\_predict)

house\_df\_new$OverallCond\_predict<-as.factor(house\_df\_new$OverallCond\_predict)

house\_df\_new$FullBath<-as.factor(house\_df\_new$FullBath)

house\_df\_new$BedroomAbvGr<-as.factor(house\_df\_new$BedroomAbvGr)

house\_df\_new$KitchenAbvGr<-as.factor(house\_df\_new$KitchenAbvGr)

house\_df\_new$Fireplaces<-as.factor(house\_df\_new$Fireplaces)

house\_df\_new<-house\_df\_new[,-14]

table(house\_df\_new$OverallCond\_predict)

prop.table(table(house\_df\_new$OverallCond\_predict))

library(ggplot2)

ggplot(house\_df\_new, aes(x=OverallCond\_predict))+geom\_bar(fill="#7666dd")+

labs(caption ="Figure 2 The number of observations for each class",x = "Overall Condition",y = "Count")+

theme(plot.caption = element\_text(hjust=0.5, size=rel(1)))

library(tree)

#Decision trees for classification

#The train-test to be split before fitting the model

set.seed(101)

#we are considering 80% of data as the train data

train<-sample(1:nrow(house\_df\_new), 987)

test<-house\_df\_new[-train,]

tree\_house <- tree(OverallCond\_predict ~ ., data=house\_df\_new, subset=train, method="class")

tree\_house

#summary of the fitted model

summary(tree\_house)

#plotting the tree

plot(tree\_house)

text(tree\_house)

#predict on test data using the fitted model

tree\_pred = predict(tree\_house, house\_df\_new[-train,], type="class")

tree\_pred

with(house\_df\_new[-train,], table(tree\_pred, OverallCond\_predict))

#The prediction accuracy

(test\_accuracy1<-(153 + 19 + 0) /208)

confusionMatrix(table(tree\_pred, test$OverallCond\_predict)

#prune method

cv\_house = cv.tree(tree\_house, FUN = prune.misclass)

cv\_house

#plotting to show cross-validation

plot(cv\_house)

#pruning the tree

prune\_house = prune.misclass(tree\_house, best = 10)

plot(prune\_house)

text(prune\_house)

tree\_pred = predict(prune\_house, house\_df\_new[-train,], type="class")

with(house\_df\_new[-train,], table(tree\_pred, OverallCond\_predict))

#The accuracy computation

(test\_accuracy2<-(153 + 19 + 0) /208)

library(caret)

library(e1071)

confusionMatrix(table(tree\_pred, test$OverallCond\_predict))

Figure B1:

**A picture containing text

Description automatically generated**

Figure B2:

**Diagram

Description automatically generated**

#for LDA model trial

house\_OverallConf\_lda <- lda(house\_df$OverallCond ~ house\_df$OverallQual

+ house\_df$SalePrice + house\_df$YearBuilt, data=house\_df)house\_OverallConf\_lda

lda\_predicted <- predict(house\_OverallConf\_lda)$class

confusionMatrix(table(lda\_predicted,house\_df$OverallCond))

#dataset loading

house\_df\_new<-read\_csv(file.choose("C:/Users/ADENIJI RAHMON O/Download/house-data.csv"))

#structure of data

dim(house\_df\_new)

class(house\_df\_new)

str(house\_df\_new)

#the count of missing values in the data

table(is.na(str(house\_df\_new)))

#replacing NA values according to description

house\_df\_new$Alley[is.na(house\_df\_new$Alley)] <- "No alley"

house\_df\_new$Fence[is.na(house\_df\_new$Fence)] <- "No fence"

house\_df\_new$BsmtQual[is.na(house\_df\_new$BsmtQual)] <- "No basement

house\_df\_new$GarageCond[is.na(house\_df\_new$GarageCond)] <- "No garage"

house\_df\_new$GarageType[is.na(house\_df\_new$GarageType)] <- "No garage"

house\_df\_new$MiscFeature[is.na(house\_df\_new$MiscFeature)] <- "None"

house\_df\_new$PoolQC[is.na(house\_df\_new$PoolQC)] <- "No pool"

house\_df\_new$BsmtCond[is.na(house\_df\_new$BsmtCond)] <- "No basement"

table(is.na(house\_df\_new))

house\_df\_new<-na.omit(house\_df\_new)

dim(house\_df\_new[sapply(house\_df\_new, is.character)])

dim(house\_df\_new[sapply(house\_df\_new, is.integer)])

data[sapply(house\_df\_new, is.character)] <- lapply(house\_df\_new[sapply(house\_df\_new, is.character)], as.factor)

class(house\_df\_new$OverallCond)

#replacing the numerical values in "OverallCond" as factors based on conditions

house\_df\_new$OverallCond\_predict<-ifelse(house\_df\_new$OverallCond >=1 & house\_df\_new$OverallCond <=3, "Poor", "Good")

house\_df\_new$OverallCond\_predict<-ifelse(house\_df\_new$OverallCond >=4 & house\_df\_new$OverallCond <=6, "Average",house\_df\_new$OverallCond\_predict)

house\_df\_new$OverallCond\_predict<-as.factor(house\_df\_new$OverallCond\_predict)

house\_df\_new$FullBath<-as.factor(house\_df\_new$FullBath)

house\_df\_new$BedroomAbvGr<-as.factor(house\_df\_new$BedroomAbvGr)

house\_df\_new$KitchenAbvGr<-as.factor(house\_df\_new$KitchenAbvGr)

house\_df\_new$Fireplaces<-as.factor(house\_df\_new$Fireplaces)

house\_df\_new<-house\_df\_new[,-14]

table(house\_df\_new$OverallCond\_predict)

prop.table(table(house\_df\_new$OverallCond\_predict))

library(ggplot2)

ggplot(house\_df\_new, aes(x=OverallCond\_predict))+geom\_bar(fill="#7666dd")+

labs(caption ="Figure 2 The number of observations for each class",x = "Overall Condition",y = "Count")+

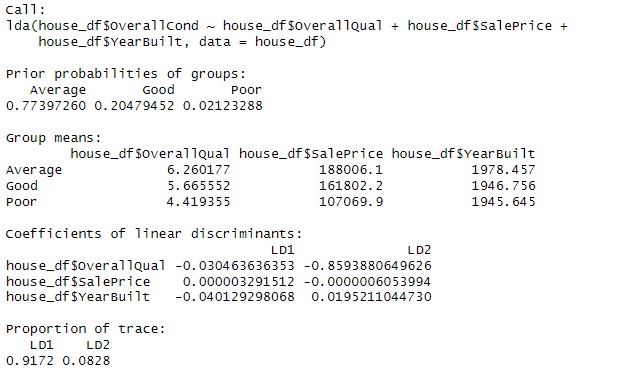
theme(plot.caption = element\_text(hjust=0.5, size=rel(1)))

Figure B3:

A picture containing text

Description automatically generated

Figure B4:



**Figure B5:**

**A screenshot of a computer

Description automatically generated with medium confidence**

**Figure B6:**

**Graphical user interface, text, application

Description automatically generated**

#Logistic regression model

house\_df<- read\_csv(file.choose("C:/Users/ADENIJI RAHMON O/Download/house-data.csv"))

summary(house\_df)

str(house\_df)

is.factor(house\_df$OverallCond)

house\_df$OverallCond <- as.factor(ifelse(house\_df$OverallCond >= 1 & house\_df$OverallCond <= 3, 'Poor',

ifelse(house\_df$OverallCond >= 4 & house\_df$OverallCond <= 6, 'Average', 'Good')))

summary(house\_df$OverallCond)

freq(house\_df$OverallCond)

house\_df[is.na(house\_df)]<- 0

table(is.na(house\_df))

LR.house <- data.frame(house\_df$OverallQual, house\_df$SalePrice,house\_df$YearBuilt,house\_df$OverallCond)

LR.house

attach(LR.house)

LR.house$house\_df.OverallCond

LR.house <- na.omit(LR.house)

training.obs <- caret::createDataPartition(LR.house$house\_df.OverallCond, p = 0.8, list = FALSE)

train.house <- LR.house[training.obs, ]

test.house <- LR.house[-training.obs, ]

### The Multinomial Logistic Regression#####

multi\_log <- nnet::multinom(train.house$house\_df.OverallCond ~ train.house$house\_df.OverallQual

+ train.house$house\_df.SalePrice + train.house$house\_df.YearBuilt, data = train.house, family="binomial")

multi\_log

summary(multi\_log)

as.data.frame(test.house)

as.data.frame(multi\_log)

predict.condition <- (predict(multi\_log, test.house, type = "class"))

head(predict.condition)

mean(predict.condition == test.house$house\_df.OverallCond)

class(test.house)

exp(coef(multi\_log)

head(probability <- fitted(multi\_log))

house\_df$predicted <- predict(multi\_log, newdata = house\_df, "class")

house\_df$predicted

confusionMatrix(table(house\_df$predicted,house\_df$OverallCond))

**Question 3:**

#Loading the dataset

houseprice\_df <- read.csv((file.choose("C:/Users/ADENIJI RAHMON O/Download/house-data.csv")))

str(df)

#summary of the data

summary(houseprice\_df)

#checking for null entries

sum(is.na(houseprice\_df))

houseprice\_df$GarageCond <- chr(houseprice\_df$GarageCond)

houseprice\_df$BsmtCond <- chr(houseprice\_df$BsmtCond)

houseprice\_df$Alley <- chr(houseprice\_df$Alley)

houseprice\_df$Fence <- chr(houseprice\_df$Fence)

houseprice\_df$PoolQC <- chr(houseprice\_df$PoolQC)

houseprice\_df$BsmtQual <- chr(houseprice\_df$BsmtQual)

houseprice\_df$GarageType <- chr(houseprice\_df$GarageType)

houseprice\_df$PoolQC %na<-% "No Pool"

houseprice\_df$LotFrontage %na<-% 0

houseprice\_df$GarageCond %na<-% "No Garage"

houseprice\_df$Alley %na<-% "No alley"

houseprice\_df$Fence %na<-% "No Fence"

houseprice\_df$BsmtCond %na<-% "No Basement"

houseprice\_df$BsmtQual %na<-% "No Basement"

houseprice\_df$GarageType %na<-% "No Garage"

houseprice\_df$MasVnrArea %na<-% 0

#Now, lets convert back to factors

houseprice\_df$Alley <- factor(houseprice\_df$Alley)

houseprice\_df$Fence <- factor(houseprice\_df$Fence)

houseprice\_df$PoolQC <- factor(houseprice\_df$PoolQC)

houseprice\_df$GarageCond<- factor(houseprice\_df$GarageCond)

houseprice\_df$BsmtCond <- factor(houseprice\_df$BsmtCond)

houseprice\_df$BsmtQual <- factor(houseprice\_df$BsmtQual)

houseprice\_df$GarageType <- factor(houseprice\_df$GarageType)

#lets check for which variable with most null values

sum(is.na(houseprice\_df$MiscFeature))

cols.dont.want <- c("MiscFeature", "Id")

#dropping "miscfeature" variable and "Id"

houseprice\_df <- houseprice\_df[, ! names(houseprice\_df) %in% cols.dont.want, drop = F]

map(houseprice\_df, ~sum(is.na(.)))

dim(houseprice\_df)

attach(houseprice\_df)

library(randomForest)

#Splitting the dataset

set.seed(1000)

library(caret)

library(purrr)

in\_train <- createDataPartition(y = houseprice\_df$SalePrice, p= 0.8, list = FALSE)

train\_set <-houseprice\_df[in\_train,]

test\_set <- houseprice\_df[-in\_train,]

# Creating random forest for regression

randf\_salesPrice <- randomForest(SalePrice ~ . ,data = train\_set, importance = T)

randf\_salesPrice

#creating a feature importance scale

importance(randf\_salesPrice)

#predicting the model using test data

randf\_salesPrice\_pred <-predict(randf\_salesPrice, newdata = test\_set)

randf\_salesPrice\_pred

importance\_var <- importance(randf\_salesPrice)

var\_Importance <- data.frame(Variables = row.names(importance\_var),

Importance = round(importance\_var[ ,'%IncMSE'],2))

var\_Importance

data.frame(

R2 = R2(randf\_salesPrice\_pred, test\_set$SalePrice),

RMSE = RMSE(randf\_salesPrice\_pred , test\_set$SalePrice),

MAE = MAE(randf\_salesPrice\_pred , test\_set$SalePrice)

)

plot(randf\_salesPrice\_pred,test\_set$SalePrice,

xlab="The predicted",ylab="The actual", xaxt="n", main = "The Random Forest Regression")

abline(a=0,b=1)

Figure C1:



Figure C1: The values obtained from the random forest model.

#Resampling

#Cross validation on random forest model

mypredict\_randomForest <- function(object, newdata)

predict(object, newdata = newdata, type = c("response"))

errorest(SalePrice ~ ., data=houseprice\_df, model=randomForest,

estimator = "cv", predict= mypredict\_randomForest)

#bootstrap method on the random forest model

errorest(SalePrice ~ ., data=houseprice\_df,model=randomForest,

estimator = "boot", est.para=control.errorest(nboot = 25), predict= mypredict\_randomForest)

Figure C2:

Graphical user interface, text, application, email

Description automatically generated

Figure C3:



**References:**

<https://cran.r-project.org/web/packages/caret/vignettes/caret.html>

<https://www.rdocumentation.org/packages/roperators/versions/1.0.1>

<https://www.r-bloggers.com/2020/09/how-to-explore-data-dataexplorer-package/>

<https://boxuancui.github.io/DataExplorer/reference/plot_missing.html>

<https://www.tidyverse.org/#:~:text=The%20tidyverse%20is%20an%20opinionated,%2C%20grammar%2C%20and%20data%20structures>.

<https://cran.r-project.org/web/packages/magrittr/vignettes/magrittr.html>

<https://visualstudiomagazine.com/articles/2016/11/01/using-the-r-nnet-package.aspx>

<https://www.section.io/engineering-education/introduction-to-random-forest-in-machine-learning/#:~:text=Advantages%20of%20random%20forest,over%20the%20decision%20tree%20algorithm>.

<https://towardsdatascience.com/everything-about-svm-classification-above-and-beyond-cc665bfd993e>

<https://scikit-learn.org/stable/modules/tree.html>