**Predict which Brand of Products Customers Prefer**

**#loading the library#####**

library(readr)

library(ggplot2)

library(dplyr)

library(caret)

library(corrplot)

library(corrgram)

library(e1071)

library(C50)

**#loading the dataset####**

existingproductattributes2017 <- read\_csv("existingproductattributes2017.csv")

existingproductattributes2017

**#dummyfy the data####**

exisitingproductattributes <- dummyVars("~ .", data = existingproductattributes2017)

newexistingproducts <- data.frame(predict(exisitingproductattributes, newdata = existingproductattributes2017))

#newexistingproducts

**#checking the missing value####**

#any(is.na(newexistingproducts))

newexistingproducts$BestSellersRank[is.na(newexistingproducts$BestSellersRank)] <- median(newexistingproducts$BestSellersRank,na.rm = TRUE)

#any(is.na(newexistingproducts))

**#correlation####**

corr.data <- cor(newexistingproducts)

#corr.data

corrplot(corr.data, tl.cex = 0.4, type = "upper")

corrplot (corr.data, tl.cex = 0.4, method = c('circle'), type= 'full')

**#droplist#####**

droplist <- c("x5StarReviews","x4StarReviews","x3StarReviews","ShippingWeight")

mynewexistingproducts <- select(newexistingproducts, -droplist)

mynewexistingproducts

head(mynewexistingproducts)

attributes(mynewexistingproducts)

**#data visulaization existing dataset####**

#Histogram

Histogram<- ggplot(mynewexistingproducts, aes(x=Volume, fill= Price)) + geom\_histogram(binwidth =1000)

Histogram

plothisto<- ggplot(mynewexistingproducts,aes(x=Volume)) + geom\_histogram(binwidth =1000, aes(fill = ..count..))

plothisto

#scatterplot

scatter <- ggplot (mynewexistingproducts, aes(x= Volume, y= Price)) + geom\_point()

scatter

#scatterplot <- ggplot(mynewexistingproducts, aes(x= Volume, y= Price))

#scatterplot

print( scatterplot + geom\_point(alpha= 0.5, size=5))

scatterplot <- scatterplot + geom\_point(size=5, color = "red")

scatterplot

**# simple linear regression####**

#sampling the data

set.seed(123**)**

**c**

trainSize<-round(nrow(mynewexistingproducts)\*0.7)

testSize<-nrow(mynewexistingproducts)-trainSize

trainSize

testSize

training\_indices<-sample(seq\_len(nrow(mynewexistingproducts)),size =trainSize)

trainSet<- mynewexistingproducts[training\_indices,]

testSet<- mynewexistingproducts[-training\_indices,]

**#Modelling**

linearModel <- lm(Volume ~ ., trainSet)

linearModel

summary(linearModel)

**#Caret Model####**

#sampling data

set.seed(998)

mynewexistingproducts <- mynewexistingproducts[sample(1:nrow(mynewexistingproducts), 80,replace=FALSE),]

**#splitting data**

inTraining <- createDataPartition(mynewexistingproducts$Volume, p = .75, list = FALSE)

training <- mynewexistingproducts[inTraining,]

testing <- mynewexistingproducts[-inTraining,]

**# LinearModel 10 fold cross validation**

fitControl <- trainControl(method = "repeatedcv", number = 10,repeats = 2)

linearModel1 <- train(Volume ~., data = training, method = "lm", trControl = fitControl)

linearModel1

varImp(linearModel1)

**#Prediction**

predictlmmodel <- predict(linearModel1,testing)

predictlmmodel

postResample(predictlmmodel, testing$Volume)

summary(predictlmmodel)

**#Support Vector Machine(SVM)####**

set.seed(998)

# cross validation

fitControl <- trainControl(method = "repeatedcv", number = 10,repeats = 2)

SVMmodel <- train(Volume ~., data = training, method = "svmLinear", trControl = fitControl)

SVMmodel

varImp(SVMmodel)

#Prediction

predictSVMmodel <- predict(SVMmodel,testing)

predictSVMmodel

postResample(predictSVMmodel, testing$Volume)

summary(predictSVMmodel)

**#Random Forest(Manuel Grid)####**

#crossvalidation

fitControl <- trainControl(method = "repeatedcv", number = 10, repeats = 1, search = 'random')

rfGrid <- expand.grid(mtry=c(1,2,3))

rfManuel <- train(Volume ~., data = training, method = "rf", trControl=fitControl, tuneGrid=rfGrid)

rfManuel

#predictions

predictrfManuel <- predict(rfManuel,testing)

predictrfManuel

**#Random forest Model####**

#crossvalidation

fitControl <- trainControl(method = "repeatedcv", number = 10, repeats = 1, search = 'random')

rfModel <- train(Volume ~., data = training, method = "rf", trControl=fitControl, tuneLength = 1)

rfModel

summary(rfModel)

**#predictions**

predictrfModel <- predict(rfModel,testing)

predictrfModel

**#kNN####**

#crossvalidation

knnmodel <- train(Volume ~ ., data = training, method = "knn", trControl=fitControl, tunelength = 1)

knnmodel

#predictions

predictknnmodel <- predict(knnmodel,testing)

predictknnmodel

**#Predictionnewproduct attributes###**

#read and preprocess the data

#Assign the data set name

newproducts <- read\_csv("newproductattributes2017.csv")

newproducts

**#dummify the data####**

salesproducts <- dummyVars("~ .", data = newproducts)

newsalesproducts <- data.frame(predict(salesproducts, newdata = newproducts))

newsalesproducts

**#droplist**

droplist1 <- c("x5StarReviews","x4StarReviews","ShippingWeight")

newsp <- select(newsalesproducts, -droplist)

newsp

head(newsp)

attributes(newsp)

**#Add predictions to the new product data set####**

newsp.predictions <- predict(rfModel,newsp)

newsp.predictions

**#adding new columns to the newproduct attribute set**

newsp$Volume <- newsp.predictions

newsp

newsp$profitnew <- newsp$Volume \* newsp$Price \* newsp$ProfitMargin

newsp

View(newsp)

**#Correlation newproduct attribute data set ####**

corr.data <- cor(newsp)

corr.data

corrplot(corr.data, tl.cex = 0.4, type = "upper")

corrplot (corr.data, tl.cex = 0.4, method = c('circle'), type= 'full')

**#data visulaization####**

#Histogram

Histogram<- ggplot(newsp, aes(x=Volume, fill= Price)) + geom\_histogram(binwidth =1000)

Histogram

pl<- ggplot(newsp,aes(x=Volume)) + geom\_histogram(binwidth =1000, aes(fill = ..count..))

pl

#scatterplots

scatter1 <- ggplot (newsp, aes(x= Volume, y= x2StarReviews)) + geom\_point()

scatter1

sc <- ggplot(newsp, aes(x= Volume, y= profitnew))

sc

print( sc + geom\_point(alpha= 0.5, size=5))

scat <- sc + geom\_point(size=5, color = "blue")

scat

sc2 <- ggplot(newsp, aes(x= Volume, y= PositiveServiceReview))

sc2

print( sc2 + geom\_point(alpha= 0.5, size=5))

pl2 <- sc2 + geom\_point (size = 5, color = 'blue')

pl2

attributes(newsp)

**#creating a csv file####**

write.csv(newsp, file="profitability.csv", row.names = TRUE)

write.csv(newsp, file = "profit.csv", row.names = TRUE )