APPENDIX A

PATTERN DETECTION OF IRIS FLOWER DATASET WITH PREDICTIVE MODEL

##KNN ON IRIS DATASET

```
# Install and load the required packages for this project
library(tidyverse) #For data manipulation,transformation and visualization
library(matrixStats) #Provides methods for operating on rows and columns
library(caret) #Mostly used for predictive modeling
library(e1071) #For evaluating functions
library(rpart.plot) #Provides a simplified interface
library(dplyr) #Mostly used for data manipulation
library(readr) #For reading rectangular data
library(DataExplorer) #For data reporting
library(ggplot2) #For creating graphics
library(ggtitle) #For adding overall plot title
library(knitr) #Mostly used for research

# Loading my dataset to R Dataframe
data<-read.csv("iris-data.csv")
# Exploring the dataset in R
names(data)
head(data)
tail(data)
str(data)
summary(data)
view(data)

# Checking the dimension of the dataset
dim(data)
```

```
# Factoring the dependent variable
data$specie<-as.factor(data$specie)


# Visualization of the Sepal length with class
data %>%
  ggplot(aes(x=specie, y=sepal.length, fill = specie)) +
  geom_boxplot() +theme_bw()+
  ggtitle("Sepal length box plot with specie")
# Visualization of the Sepal width with class
data %>%
  ggplot(aes(x=specie, y=sepal.width, fill = specie)) +
  geom_boxplot() +theme_bw()+
  ggtitle("Sepal width box plot with specie")
# Visualization of the Petal length with class
data %>%
  ggplot(aes(x=specie, y=petal.length, fill = specie)) +
  geom_boxplot() +theme_bw()+
  ggtitle("Petal length box plot with specie")
# Visualization of the Petal width with class
data %>%
  ggplot(aes(x=specie, y=petal.width, fill = specie)) +
  geom_boxplot() +theme_bw()+
  ggtitle("Petal width box plot with specie")
## MODEL BUILDING USING KNN Algorithm
```

```
# Splitting the data set for train and test
set.seed(1234)
ind <- sample(2, nrow(data), replace = T, prob = c(0.6, 0.4))
train <- data[ind == 1,]
test <- data[ind == 2,]
trControl <- trainControl(method  = "cv",
                          number  = 10)
Knn_model<- train(specie ~ .,
                  method     = "knn",
                  tuneGrid   = expand.grid(k = 1:5),
                  trControl  = trControl,
                  metric     = "Accuracy",
                  data       = train)
Knn_model
# Plotting observations
plot(Knn_model)

##MAKING PREDICTION FROM THE MODEL BUILT
# Predicting the test set
knn_predict <- predict(Knn_model, test)
# Predicting the train set
knn_predict_train <- predict(Knn_model, train)
#Get the confusion matrix to see accuracy value and other parameter values
misclass <- 1 - sum(diag(cm$table))/sum(cm$table)
```

```
cm_train <- confusionMatrix(knn_predict_train, train$specie )

knn_predict <- predict(Knn_model, test, type="prob")

knn_predict

#Train and Test misclassification

misclass <- 1 - sum(diag(cm$table))/sum(cm$table)

misclass_train <- 1 - sum(diag(cm_train$table))/sum(cm_train$table)

misclass

misclass_train
```

APPENDIX B

GROCERY ITEM ANALYSIS WITH ASSOCIATION RULES

```
## Association Rule with Groceries dataset in R

# Install and load the required packages for this project
library(arules) #For representing, manipulating and analyzing transaction data and patterns

library(fpp2) #To load the required data

library(arulesViz) #For visualizing association rules and frequent itemsets

library(dplyr) #Mostly used for data manipulation

library(pander) #To provide a minimal and easy tool for rendering R objects into Pandoc's markdown

library(Rcpp) #For high performance computing


# Loading my dataset to R Dataframe
grocery<-read.transactions("groceries.csv", sep = ",", format = "basket")

# Exploring the dataset in R
names(grocery)
```

```
head(grocery)
tail(grocery)
str(grocery)
view(grocery)
summary(grocery)

# Checking the dimension of the dataset
dim(grocery)

##Using the following codes to create the Association rules
itemFrequencyPlot(grocery, topN = 20, main = "Top 20 items purchased")
# The association algorithm
grocery_rules<-apriori(grocery, control=list(verbose=FALSE), parameter
                =list(support=0.001, confidence = 0.25, minlen=2))
grocery_rules_uplift<-sort(grocery_rules, by = "lift", decreasing = FALSE)[1:10]
grocery_rules_support<-sort(grocery_rules, by = "support", decreasing = TRUE)[1:10]
grocery_rules_confidence<-sort(grocery_rules, by = "confidence", decreasing = TRUE)[1:10]
inspect(grocery_rules_uplift)
inspect(grocery_rules_support)
inspect(grocery_rules_confidence)

# Showing the items sold with soda
rule_soda<- apriori(grocery, parameter = list(support=0.01,
                                    confidence=.01,
                                    minlen=2,
                                    target='rules'),
              appearance = list(default='rhs',lhs='soda'), control = list(verbose=FALSE))
```

```
inspect(sort(rule_soda, by = "support", decreasing = T))
#Plotting the analysis
plot(rule_soda, method="graph", iteractive=FALSE)
plot(grocery_rules, method = "graph", measure = "confidence", shading = "lift")
plot(grocery_rules, measure=c("support", "confidence"), shading="lift", interactive=FALSE)
#Generating the rules
data$class[data$class == 'Iris-virginica'] <- 'virginica'
data$class[data$class == 'Iris-setosa'] <- 'setosa'
data$class[data$class == 'Iris-versicolor'] <- 'versicolor'
cor(data[,c(1:4)],use="complete")
correlate <- cor(data[,1:4]) #makes correlations for the 1st through 4th columns of the data iris
corrplot(correlate, method="number")
pairs(data[1:4], main="Iris Data",
      pch=21, bg=c("red","green3","blue")[unclass(data$Species)])
```

APPENDIX C

CLUSTERING ANALYSIS IN IRIS FLOWERS CLASSIFICATION

```
##K means Clustering Analysis in R
#Install and load the required packages
library(factoextra) #For visualizing the contribution of rows/columns
library(ggplot2)  #For data exploration and visualization
library(gridExtra) #For summary statistics visualisation
library(cluster)  #For cluster algorithm
```

```
# Loading my dataset to R Dataframe
data<-read.csv("iris-data.csv")

# Exploring the dataset in R
names(data)
head(data)
tail(data)
str(data)
summary(data)
view(data)

# Checking the dimension of the dataset
dim(data)
##Using the following codes to create K-Means Clustering
data<-data[1:4]
data_scale<-scale(data)
data<-dist(data_scale)
fviz_nbclust(data_scale, kmeans, method='wss') + labs(subtitle="Elbow_Method")
km_iris<-kmeans(data_scale, centers=3)
#Print the result
print(km_iris)
clusplot(data_scale, km_iris$cluster, color=TRUE, shade = TRUE, label=2)
```

APPENDIX D

SENTIMENT ANALYSIS OF HOTEL REVIEWS FOR BUSINESS DECISIONS

```
##Sentiment Analysis
#Install and load the required packages
library(tm) #For cleaning the Corpus
```

```
library(syuzhet) #Extracts sentiment and sentiment-derived plot arcs from text
library(wordcloud) #For analyzing texts and to quickly visualize the keywords
library(skimr) ##for data exploration
library(readxl) # read excel into r dataframe
library(ggplot2) #For creating graphics


# Loading my dataset to R Dataframe
data <- read.csv("Second_Last_Hotel_1.csv")
#Exploring the data
names(data)
head(data)
tail(data)
summary(data)
str(data)
dim(data)
skim(data)


# Converting to UTF-8 format
corpus<-iconv(data$Review, to = "UTF-8")
corpus<-Corpus(VectorSource(corpus))
inspect(corpus[1:5])
# The cleaning of text
corpus<-tm_map(corpus,removePunctuation)
inspect(corpus[1:5])
corpus<-tm_map(corpus, tolower)
corpus<-tm_map(corpus, removeNumbers)
inspect(corpus[1:5])
```

```
corpus<-tm_map(corpus, removeWords, stopwords("english"))
inspect(corpus[1:5])
corpus<-tm_map(corpus, removeWords, c("fufuudfufufubbfufuudfufufubb",
"fufuuafufuuafufuuafufuua"))
inspect(corpus[1:5])
corpus<-tm_map(corpus, stripWhitespace)
inspect(corpus[1:5])

# Obtaining the Term Document Frequency
tdm<-TermDocumentMatrix(corpus)
tdm<-as.matrix(tdm)
g<-sort(rowSums(tdm), decreasing = TRUE)
b<-data.frame(word=names(g), freq=g)
tdm[1:10,1:5]
#Computing the wordcloud
set.seed(12345)
wordcloud(b$word,b$freq, max.words = 50, random.order = FALSE, rot.per = 0.3
        ,colors = brewer.pal(8, "Dark2"), scale = c(4, 0.5))

# Sentiment analysis with Syuzhet
corpus<-as.character(corpus)
sentiment<-get_nrc_sentiment(corpus)
sent<-data.frame(colSums(sentiment))
SentimentScores<-data.frame(colSums(sentiment[,]))
names(SentimentScores) <- "Score"
```

```
SentimentScores <- cbind("sentiment" = rownames(SentimentScores), SentimentScores)

rownames(SentimentScores) <- NULL

ggplot(data = SentimentScores, aes(x = sentiment, y = Score)) + geom_bar(aes(fill = sentiment), stat = "identity") + theme(legend.position = "none") + xlab("Sentiment") + ylab("Score") + ggtitle("Hotel Reviews Analysis")
```