National College of Ireland

Research Project

on

A predictive Analysis of airline passengers demand between (Eco and Business class) during and post period of economic recession using machine leaning analysis

Ugwuanyi Arinze J
18139442

MSc Data Analytics

Submitted to:Vladimir Milosavljevic

National College of Ireland
Project Submission Sheet – 2017/2018
School of Computing

Student Name:	Forename Surname
Student ID:	1234567
Programme:	MSc Data Analytics
Year:	2018/9
Module:	Msc Research Project Manual Configuration
Lecturer:	Vladimir Milosavljevic
Submission Due Date:	7/08/2020
Project Title:	Insert Title Here

I hereby certify that the information contained in this (my submission) is information pertaining to my own individual work that I conducted for this project. All information other than my own contribution is fully and appropriately referenced and listed in the relevant bibliography section. I assert that I have not referred to any work(s) other than those listed. I also include my TurnItIn report with this submission.

ALL materials used must be referenced in the bibliography section. Students are encouraged to use the Harvard Referencing Standard supplied by the Library. To use other author's written or electronic work is an act of plagiarism and may result in disciplinary action. Students may be required to undergo a viva (oral examination) if there is suspicion about the validity of their submitted work.

Signature:	
Date:	September 25, 2020

PLEASE READ THE FOLLOWING INSTRUCTIONS:

1. Please attach a completed copy of this sheet to each project (including multiple copies).
2. **You must ensure that you retain a HARD COPY of ALL projects**, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer. Please do not bind projects or place in covers unless specifically requested.
3. Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration manual

Ugwuanyi Arinze J
18139442

September 25, 2020

1 Introduction

Configuration Manuel consists of a specification of soft and hardware employed for the implementation of this project. This involves the detailed step by step process and codes from the cleaning of data to the implementation of the models using the trained data and testing data to predict the major factor affecting passengers demand between (Eco and Business class) during and post period of economic recession using machine leaning analysis, four machine learning algorithm, Random Forest, Logistic Regression, K-Nearest Neighbour and Decision Tree (C50) was applied to predict and evaluation of their performance was conducted.

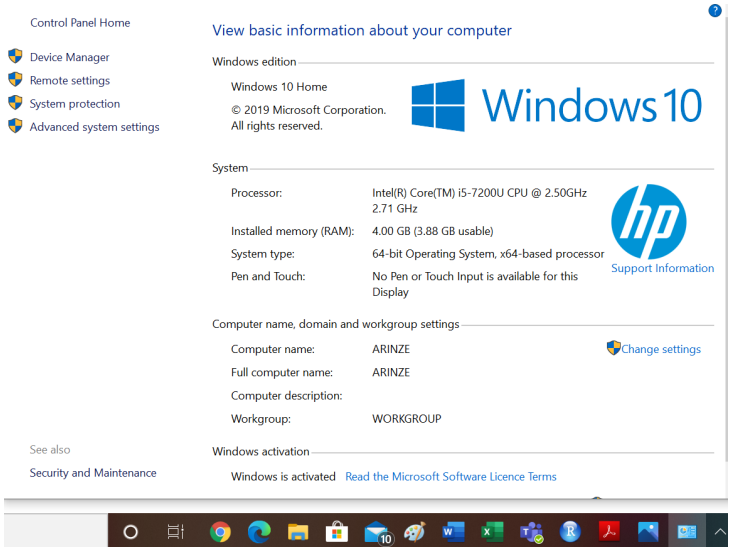


Figure 1: System specification

System Hardware
Processor: intel® core™ i5-7200u©2.50GHz 2.71GHz
RAM: 4 GB
System Type: Windows 10 OS, 64-bit operating System

1.1 Software Specification

Microsoft Excel 2018: This was used to store dataset in a flat-file as CVS (comma separated value)

RStudio: R version 3.3.3 (2017-03-06) was used implementation of all algorithm used in this project. The step by step process of implementing this project from data extraction to Evaluation of the models was carried out using RStudio. This is the link: https://cran.r-project.org/bin/windows/base/

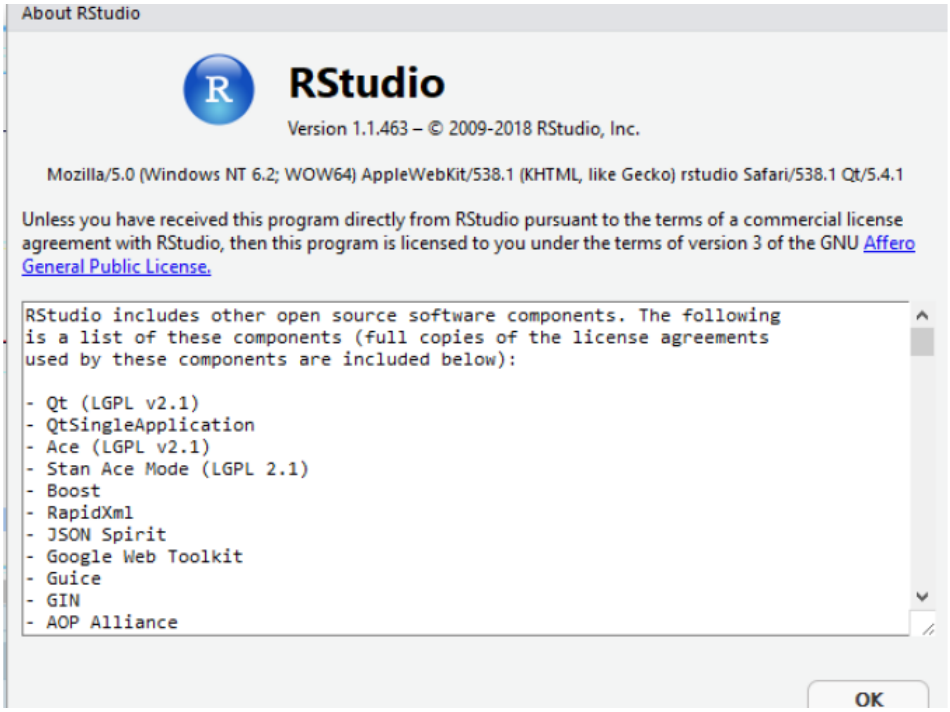


Figure 2: Rstudio versionS

2 Step-wise Implementation of the project

a) Downloaded the R language and R studio.
(b) Installation of R Studio 3.3.3
(c) Installation of the libraries.
(d) Setting up a working directory.
(e) Cleaning and exploratory Analysis.
(f) Implementation of the four models.

3 Data Preparation

Setting the working directory: a working directory was set in Rstudio to "C:/users/Arinze/Document" this where all the working files and dataset in CSV was saved.
Setwd ("c://users/Arinze/Document")
Getwd()

4 Library Installation

This is a process of identifying and installing the package in R that is essential for a particle code or model that will help to archive your objective. The library is run before

executing the code command. Below is the list of libraries used in this project.

```
### loading of the library using used for the project
library(readxl)
library(caret)
library(randomForest)
library(lm)
library (e1071)
library(rpart)
library(ROCR)
library(ROSE)
library(C50)
library(gplots)
library(caTools)
library(Matrix)
library(readxl)
```

Figure 3: Rstudio versionS

5 Data Transformation

This is the process of transforming variable into the desired form, 7 variables Gender, Type of travel, Destination State, Airline Status, Flight cancelled and Airline Name including the target Class variable was transformed from the categorical variable into numerical variables, the missing and null values were omitted. Figure 4 Below is the code for the transformation.

```
###############coversion to numeric ############
Survey_data$Class<- factor(Survey_data$Class, levels = c("Business", "Eco"), labels = c(0,1))
Survey_data$Type_of_Travel <- factor(Survey_data$Type_of_Travel, levels =
        c("Business travel", "Mileage tickets", "Personal Travel"), labels = c(1,2,3))
Survey_data$Airline_Status <- factor(Survey_data$Airline_Status, levels =
                      c("Blue", "Gold", "Platinum","Silver"), labels = c(1,2,3,4))
Survey_data$Gender <- factor(Survey_data$Gender, levels = c("Female", "Male"),
                             labels = c(0,1))
Survey_data$Destination_State <- factor(Survey_data$Destination_State,levels =
            c("Alabama", "Alaska", "Arizon", "Arkanasas", "California", "Colorado",
              "Connecticut", "Delaware", "Florida", "Georgia", "Hawaii", "Idaho", "Illinois",
              "Indiana", "Lowa", "Kanasas", "Kentucky", "Louisiana", "Maine", "Maryland",
              "Massachusetts", "Michigan", "Minnesota", "Missisispps", "Missouri", "Montana",

            "Nebraska", "Nevada", "New Hamspshire", "New Jersey", "New Mexico", "New York",
            "North Carolina", "North Dakota", "Ohio", "Oklahoma", "Oregon", "Pennsylvania",
            "Puerto RIco", "Rhode Island", "South Carolina", "south Dakolian", "Tennessee",
            "Texas", "U.S Pacific Trust Territories and Possesion", "Utah", "Vermont",
            "Virginia", "Washington", "West Virginia", "Wisconsin", "wyoming"),
              labels = c(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,
     23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52))
Survey_data$Flight_cancelled <- factor(Survey_data$Flight_cancelled, levels = c("Yes", "No"),
                                       labels = c(1,0))
Survey_data$Airline_Name <- factor(Survey_data$Airline_Name, levels =
                  c("Cheapseats Airlines Inc.", "Cool&Young Airlines Inc.",
                    "EnjoyFlying Air Services", "FlyFast AirWays Inc.", "FlyHere AirWays",
                    "FlyToSun Airlines Inc.", "GoingNorth Airlines INc.",
                   "NorhWest Business Airlines Inc.", "OnlyJets Airlines Inc.",
              "oursin Airlines Inc.", "Paul Smith Airlines Inc.", "Sigma Airlines Inc.",
              "Southeast Airlines co.", "West Airline Inc."),
              labels = c(1,2,3,4,5,6,7,8,9,10,11,12,13,14))

recessionclean <- Survey_data[, -c(5,9,10,17,19,20,21,29)]

View(recessionda)
sum(is.na(recessionclean))
sapply(recessionclean,function(x) sum(is.na(x)))
write.csv(recessionclean, "surveycleaned.csv", row.names = FALSE)
```

Figure 4: Rstudio version

6 Data Cleaning

The data was downloaded in xls format, it contained the year when there was a recession and when there was no recession. It was loaded into RStudio for cleaning, encoding and save in CSV. Exploratory data analysis was also perform using cleaned CSV data.

```
######## data cleaning ###############
library(readxl)
Survey_data <- read_excel("c:/users/Arinze/Document")
View(Survey_data)
str(Survey_data)
is.na(Survey_data)
sum(is.na(Survey_data))

View(Survey_data)
summary(Survey_data)

sum(is.na(overbacleaned))
na.omit(overbacleaned)
any(is.na(Survey_data))
sapply(overbacleaned,function(x) sum(is.na(x)))
write.csv(Survey_data, "surveycleaned.csv", row.names = FALSE)
```

Figure 5: Data cleaning

7 Implementation of oversample Process

The data was highly imbalance data it contain more of Economy class records more than the Business class, data imbalance effect the performance of model. however, over-sampling method of balancing imbalance data was applied that is using synthetic Minority Oversampling Techniques (SMOTE) figure 6 is the code used

```
##############balancing of data set ############################
library(ROSE)
str(surveycleaned)
prop.table(table(surveycleaned$class))
barplot(prop.table(table(surveycleaned)),
        colours=rainbow(2),
        ylim = c(0, 0.8),
        main = "class mainframe")

overbacleaned <- ovun.sample(Class~., data = surveycleaned,
                             method = "over", N = 10956)$data
table(overbacleaned$Class)
write.csv(overbacleaned, "surveycleaned1", row.names = FALSE)
```

Figure 6: Overbalancing of Data

8 Variable Significant

checking for variable significant was conducted using Logistic Regression, the variable that are significant at 95% confident interval was mark with (“”***””) variable with p-value 0.05 or less, secondly the Random Forest, the technique it arranging the variable in order of their important and last Decision Tree (C50) ranking the variables by the order of their Percentage of contribution.

```
##### selection of performing variable###

rftrain <- randomForest(Class~., data = train)
varImpPlot(rftrain)
######for logistic regression ###
logit <- glm(Class~., family = binomial(link = "logit"), data = training)
summary(logit)

### for c50 ##
# Train
cFifty <- C5.0(Class~., data = training)
summary(cFifty)
```

Figure 7: code for variable significant

9 Implementation of Random Forest

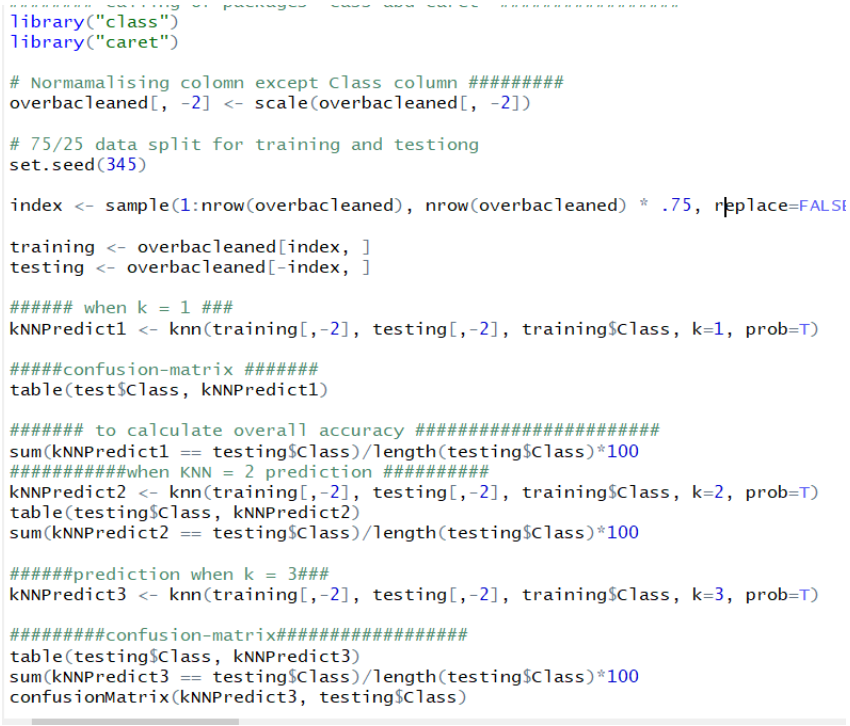
the dataset for implementation of Random forest model was divided 75/25 for training and testing data. varImpPlot(rftraining) was used to get the order of variable important to the prediction. the model performs very well but it took a longer time to run. eating and drinking, price-sensitive, and Satisfaction had the highest contribution and the accuracy prediction was 79.8%.

```
#############################################################################################

overbacleaned$Class <- factor(overbacleaned$Class, levels=c(0,1), labels = c("Eco", "Business"))

summary(overbacleaned)
prop.table(table(overbacleaned$Class))
barplot(prop.table(table(overbacleaned$Class)),
        colours=rainbow(2),
        ylim = c(0, 0.8),
        main = "class mainframe")
str(overbacleaned)
set.seed(345)

ind <- sample(2,  nrow(overbacleaned), replace = TRUE,prob = c(0.7,0.3))
train <- overbacleaned[ind==1,]
test <- overbacleaned[ind==2,]
table(train$Class)
prop.table(table(overbacleaned))
summary(train)

####conversion to factor model random forest ###
library(randomForest)
train$Class <- as.character(train$Class)
train$Class <- as.factor(train$Class)
Randftrain <- randomForest(Class~., data = train)
varImpPlotDat <- varImpPlot(Randftrain)
##############prediction Business usig test data evaluation ######
library(caret)
library(e1071)
confusionMatrix(predict(Randftrain, test), test$Class, positive = "Business")
```

Figure 8: Random Forest code

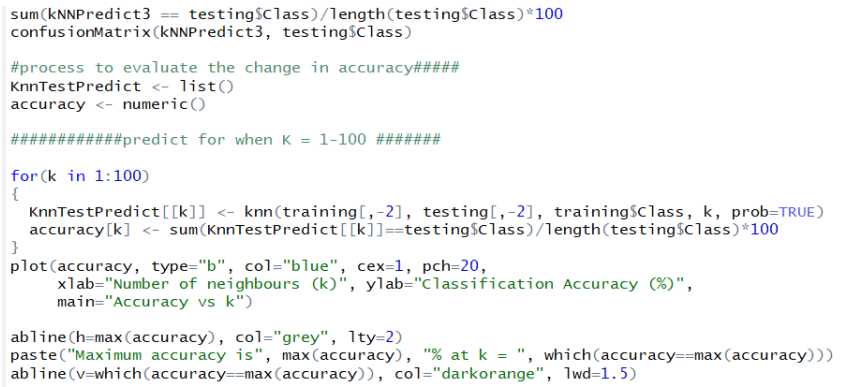
10 implementation KNN

Implementing K-Nearest Neighbour the dataset was divided into two 75/25% the training and testing section respectively. Training data was used for training of the model, while the testing data was used for testing of the model the Accuracy of the prediction was plotted against the number of k (1:100) to get visualise the best k-value.



```
library("class")
library("caret")

# Normamalising colomn except Class column #########
overbacleaned[, -2] <- scale(overbacleaned[, -2])

# 75/25 data split for training and testiong
set.seed(345)

index <- sample(1:nrow(overbacleaned), nrow(overbacleaned) * .75, replace=FALSE

training <- overbacleaned[index, ]
testing <- overbacleaned[-index, ]

###### when k = 1 ###
kNNPredict1 <- knn(training[,-2], testing[,-2], training$Class, k=1, prob=T)

#####confusion-matrix #######
table(test$Class, kNNPredict1)

####### to calculate overall accuracy ######################
sum(kNNPredict1 == testing$Class)/length(testing$Class)*100
###########when KNN = 2 prediction ##########
kNNPredict2 <- knn(training[,-2], testing[,-2], training$Class, k=2, prob=T)
table(testing$Class, kNNPredict2)
sum(kNNPredict2 == testing$Class)/length(testing$Class)*100

######prediction when k = 3###
kNNPredict3 <- knn(training[,-2], testing[,-2], training$Class, k=3, prob=T)

#########confusion-matrix##################
table(testing$Class, kNNPredict3)
sum(kNNPredict3 == testing$Class)/length(testing$Class)*100
confusionMatrix(kNNPredict3, testing$Class)
```

Figure 9: knn code



```
sum(kNNPredict3 == testing$Class)/length(testing$Class)*100
confusionMatrix(kNNPredict3, testing$Class)

#process to evaluate the change in accuracy#####
KnnTestPredict <- list()
accuracy <- numeric()

############predict for when K = 1-100 #######

for(k in 1:100)
{
  KnnTestPredict[[k]] <- knn(training[,-2], testing[,-2], training$Class, k, prob=TRUE)
  accuracy[k] <- sum(KnnTestPredict[[k]]==testing$Class)/length(testing$Class)*100
}
plot(accuracy, type="b", col="blue", cex=1, pch=20,
     xlab="Number of neighbours (k)", ylab="Classification Accuracy (%)",
     main="Accuracy vs k")

abline(h=max(accuracy), col="grey", lty=2)
paste("Maximum accuracy is", max(accuracy), "% at k = ", which(accuracy==max(accuracy)))
abline(v=which(accuracy==max(accuracy)), col="darkorange", lwd=1.5)
```

Figure 10: knn code

11 Implementation of Logistic Regression

From figure 11 show Logistic Regression model implementation, the dataset was split into training 70% and testing 30% and the variables with ("***") are significant. when implementing the model I observed that price-sensitive, loyalty card which was significant

during the random forest and c50 was not significant, and that affected the accuracy of the prediction it performed lower.

```
##################prediction with logistic regression########
overbacleaned11 <- overbacleaned[, -c(5,7,9)]

str(overbacleaned11)
View(overbacleaned11)
set.seed(345)
index <- sample(1:nrow(overbacleaned11), nrow(overbacleaned11) * .70,
                replace=FALSE)

training <- overbacleaned11[index, ]
testing <- overbacleaned11[-index, ]
training$Class <- as.character(training$Class)
training$Class <- as.factor(training$Class)
##predicting with general linear model and variable contribution

logit <- glm(Class~., family = binomial(link = "logit"),
             data = training)
summary(logit)
########## predicting logistic regression#########################

logit.prediction <- predict(logit, newdata = testing, type = "response")


results.logit <- ifelse(logit.prediction > 0.5,"Business","Eco")
results.logit

# table confusion matrix
table(testing$Class, logit.prediction > 0.5)


# ploting ROC and AUC for logistic prediction
library(ROCR)
ROCRpred <- prediction(logit.prediction, testing$Class)
ROCRperf <- performance(ROCRpred, 'tpr','fpr')
plot(ROCRperf, colorize = TRUE, text.adj = c(-0.2,1.7))
auc <- performance(ROCRpred, measure = "auc")
auc <- auc@y.values[[1]]
auc
```

Figure 11: Logistic Regression code

12 Implementation of Decision Tree

Figure 12 shows the implementation of Decision Tree, the dataset was divided 75/25 Training and Testing accordingly, 80% accuracy prediction was archived. however, the variable significant show that customer satisfaction, Eating and Drinking and price-sensitive were significant also in the Random forest. summary(Dcc5) was used to get the percentage of each variable contribution..

```
################# C5.0 Decision Tree ##################
overbacleaned$Class <- factor(overbacleaned$Class, labels=c(0,1),
                              levels=c("Eco", "Business"))
sapply(overbacleaned,function(x) sum(is.na(x)))
str(overbacleaned)
library(C50)
set.seed(345)
# decision trees can use numerical values so let's import


# Create 75/25 split train-test set



indexc5 <- sample(1:dim(overbacleaned)[1],
                  dim(overbacleaned)[1] * .70, replace = F)
trainc5 <- overbacleaned[indexc5, ]
testc5<- overbacleaned[-indexc5, ]

trainc5$Class<- as.character(trainc5$Class)
trainc5$Class <- as.factor(trainc5$Class)
Dcc5 <- C5.0(Class~., data = trainc5)
summary(Dcc5)
predc5 <- predict(Dcc5, testc5[, -2])
caret::confusionMatrix(testc5$Class, predc5, positive="Business")
```

Figure 12: Decision tress code

References

YouTube Video presentation link: https://youtu.be/uLLHbMLlC7Q
Using R and R studio: link: https://cran.r-project.org/bin/windows/base/,
https://rstudio.com/products/rstudio/download/
https://microsoft-paint
https://www.stackoverflow.com