General Sir John Kotelawala Defence University

Faculty of Management, Social Sciences and Humanities

Department of Languages

BSc in Applied Data Science Communication

Fundamentals of data mining / LB 2114

Group Assignment

P Laksia – D/ADC/23/0013

PRM Perera – D/ADC/23/0031

MM Jayasinghe – D/ADC/23/0035

RYN Sanduprabha – D/ADC/23/0046

Fundamentals of data mining / LB 2114

Year 2: Semester 1

Assignment number 1

04.03.2024

Intention of Online Shoppers

Content

1. Introduction

Online shopping is a form of e-commerce which is involved the process of researching and purchasing and selling of goods and services via internet and online media through a web browser or a mobile app.

Online shopping has become an important aspect in the lives of people due to its easy convenience to user since the user can browse and purchase the desired product any place any time as long as he has access to the internet, saves cost and time, the ability to purchase a wider variety of products and easy comparison of products and product prices.

The intention of online shoppers is based on a real-time online shopper behaviour analysis system which consists of two modules where one module predicts the visitor's shopping intent while the other simultaneously predicts the website abandonment likelihood of each visitor.

2. Dataset

The online shoppers purchasing intention is a multivariate dataset which consists of 12,330 rows and 18 columns where 17 columns contain feature variable which are considered as the independent variables while 1 column is a target variable which is considered as a dependant variable.

The 18 variables are as follows;

1. **Administrative:** represents the number of visitors visited the administrative page during a session. Feature variable. Data type – integer.
2. **Administrative_Duration:** the total time spent in the administrative page. Feature variable. Data type – integer.
3. **Informational:** represents the number of visitors visited the informational page during a session. Feature variable. Data type – integer.
4. **Informational_Duration:** the total time spent in the informational page. Feature variable. Data type – integer.
5. **ProductRelated:** represents the number of visitors visited the product related page during a session. Feature variable. Data type – integer.
6. **ProductRelated_Duration:** the total time spent in the product related page. Feature variable. Data type – continuous.
7. **BounceRates:** refers to the percentage of visitors who enter the site from that page and then leave ("bounce") without triggering any other requests to the analytics server during that session. Feature variable. Data type – continuous.
8. **ExitRates:** represents a value for a specific web page is calculated as for all pageviews to the page, the percentage that were the last in the session. Feature variable. Data type – continuous.
9. **PageValues:** represents the average value for a web page that a user visited before completing an e-commerce transaction. Feature variable. Data type – integer.
10. **SpecialDay:** indicates the closeness of the site visiting time to a specific special day (e.g. Mother's Day, Valentine's Day) in which the sessions are more likely to be finalized with transaction. Feature variable. Data type – integer.
11. **Month:** represents the 12 months of the year not including January and April. Feature variable. Data type – categorical.
12. **OperatingSystems:** represents the number of operating systems used by a single visitor. Feature variable. Data type – integer.
13. **Browser:** represents the number of browser engines used by a single visitor. Feature variable. Data type – integer.
14. **Region:** represents the position of the region where the total number of regions is 9. Feature variable. Data type – integer.
15. **TrafficType:** represents the type of traffic to understand where online shoppers are coming from. Feature variable. Data type – integer.
16. **VisitorType:** refers the visitor type, whether he is a returning visitor or new. Feature variable. Data type – categorical.

17. **Weekend:** indicates whether the date of the visit is weekend. Feature variable. Data type – binary.
18. **Revenue:** indicates whether the particular website generates a revenue or not. Target variable. Data type – binary.

The dataset consists of 10 numerical and 8 categorical attributes.

The target variable, "Revenue" is used as the class label.

There are no missing values.

Both classification and clustering are used in the dataset.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	Administrative	Administrative_Duration	Informational	Informational_Duration	ProductRelated	ProductRelated_Duration	BounceRates	ExitRates	PageValues	SpecialDay	Month	OperatingSystems	Browser	Region	TrafficType	VisitorType	Weekend	Revenue	
2	0	0	0	0	1	0	0.2	0.2	0	0	Feb	1	1	1	1	Returning_V	FALSE	FALSE	
3	0	0	0	0	2	64	0	0.1	0	0	Feb	2	2	1	2	Returning_V	FALSE	FALSE	
4	0	0	0	0	1	0	0.2	0.2	0	0	Feb	4	1	9	3	Returning_V	FALSE	FALSE	
5	0	0	0	0	2	2.666666667	0.05	0.14	0	0	Feb	3	2	2	4	Returning_V	FALSE	FALSE	
6	0	0	0	0	10	627.5	0.02	0.05	0	0	Feb	3	3	1	4	Returning_V	TRUE	FALSE	
7	0	0	0	0	19	154.2166667	0.015789474	0.024561	0	0	Feb	2	2	1	3	Returning_V	FALSE	FALSE	
8	0	0	0	0	1	0	0.2	0.2	0	0.4	Feb	2	4	3	3	Returning_V	FALSE	FALSE	
9	1	0	0	0	0	0	0.2	0.2	0	0	Feb	1	2	1	5	Returning_V	TRUE	FALSE	
10	0	0	0	0	2	37	0	0.1	0	0.8	Feb	2	2	2	3	Returning_V	FALSE	FALSE	
11	0	0	0	0	3	738	0	0.022222	0	0.4	Feb	2	4	1	2	Returning_V	FALSE	FALSE	
12	0	0	0	0	3	395	0	0.066667	0	0	Feb	1	1	3	3	Returning_V	FALSE	FALSE	
13	0	0	0	0	16	407.75	0.01875	0.025833	0	0.4	Feb	1	1	4	3	Returning_V	FALSE	FALSE	
14	0	0	0	0	7	280.5	0	0.028571	0	0	Feb	1	1	1	3	Returning_V	FALSE	FALSE	
15	0	0	0	0	6	98	0	0.066667	0	0	Feb	2	5	1	3	Returning_V	FALSE	FALSE	
16	0	0	0	0	2	68	0	0.1	0	0	Feb	3	2	3	3	Returning_V	FALSE	FALSE	
17	2	53	0	0	23	1668.285119	0.008333333	0.016313	0	0	Feb	1	1	9	3	Returning_V	FALSE	FALSE	
18	0	0	0	0	1	0	0.2	0.2	0	0	Feb	1	1	4	3	Returning_V	FALSE	FALSE	
19	0	0	0	0	13	334.9666667	0	0.007692	0	0	Feb	1	1	1	4	Returning_V	TRUE	FALSE	
20	0	0	0	0	2	32	0	0.1	0	0	Feb	2	2	1	3	Returning_V	FALSE	FALSE	
21	0	0	0	0	20	2981.166667	0	0.01	0	0	Feb	2	4	4	4	Returning_V	FALSE	FALSE	
22	0	0	0	0	8	136.1666667	0	0.008333	0	1	Feb	2	2	5	1	Returning_V	TRUE	FALSE	
23	0	0	0	0	2	0	0.2	0.2	0	0	Feb	3	3	1	3	Returning_V	FALSE	FALSE	
24	0	0	0	0	3	105	0	0.033333	0	0	Feb	3	2	1	5	Returning_V	FALSE	FALSE	
25	0	0	0	0	2	15	0	0.1	0	0.8	Feb	2	4	1	3	Returning_V	FALSE	FALSE	
26	0	0	0	0	1	0	0.2	0.2	0	0	Feb	2	2	4	1	Returning_V	TRUE	FALSE	
27	0	0	0	0	5	156	0	0.04	0	0	Feb	1	1	9	3	Returning_V	FALSE	FALSE	
28	4	64.6	0	0	32	1135.444444	0.002857143	0.009524	0	0	Feb	2	2	1	3	Returning_V	FALSE	FALSE	
29	0	0	0	0	4	76	0.05	0.1	0	0	Feb	1	1	1	3	Returning_V	FALSE	FALSE	
30	0	0	0	0	4	63	0	0.05	0	0.2	Feb	2	6	1	3	Returning_V	FALSE	FALSE	
31	1	6	1	0	45	1582.75	0.043478261	0.050821	54.179764	0.4	Feb	3	2	1	1	Returning_V	FALSE	FALSE	
32	0	0	0	0	2	35	0	0.1	0	0	Feb	1	1	6	3	Returning_V	FALSE	FALSE	
33	0	0	0	0	3	78	0	0.066667	0	0	Feb	1	2	6	6	Returning_V	TRUE	FALSE	
34	0	0	0	0	8	209.5	0	0.025	0	0	Feb	2	2	1	1	Returning_V	FALSE	FALSE	
35	0	0	0	0	10	183.6666667	0.04	0.08	0	0	Feb	1	1	3	1	Returning_V	FALSE	FALSE	
36	0	0	0	0	14	380.5	0.014285714	0.028571	0	0	Feb	2	2	1	1	Returning_V	FALSE	FALSE	

Figure 01

https://archive.ics.uci.edu/dataset/468/online+shoppers+purchasing+intention+dataset

The purpose of this dataset is to identify whether the online shopping website in question generates a revenue or not is the problem identified after analysing the dataset. According to the target variable, if a revenue is generated it will show "TRUE" while vice-versa if a revenue is not generated it will show "FALSE".

Through this analysis we can identify that there is a high trend where a revenue is not generated, since this is also important for the online store owners to understand where online shoppers are coming from where they can further analyse and apply various marketing and price strategies in order to generate a future revenue to the website.

3. Classification

3.1. Explanation and preparation of datasets

The dataset utilized for the classification is the purchasing intention of online shoppers which consists of 12,330 data records.

Since the dataset is already a clean dataset, no missing values were found. Therefore, no preparatory tasks were carried on the dataset.

The dependant variable is the “Revenue” column which is also considered to be the class label of the dataset while the independent variables are the rest of the columns such as “Administrative”, “BounceRates”, “TrafficType” and “Region” are some of them.

The classification of this dataset is predicted using the KNN model using the target variable, “Revenue” where the dataset is split to train and test the dataset.

3.2. Data mining

The data mining technique utilized for the dataset is classification using R programming language.

KNN model, K – Nearest Neighbour is the technique used for the classification of the dataset.

Using the KNN model, we split the dataset where 70% is used to train the dataset and 30% of the dataset is used to test the dataset where the model will be able to understand, evaluate and access the dataset.

Visualization tools used:

1. corrplot to find the performance using the confusion matrix.

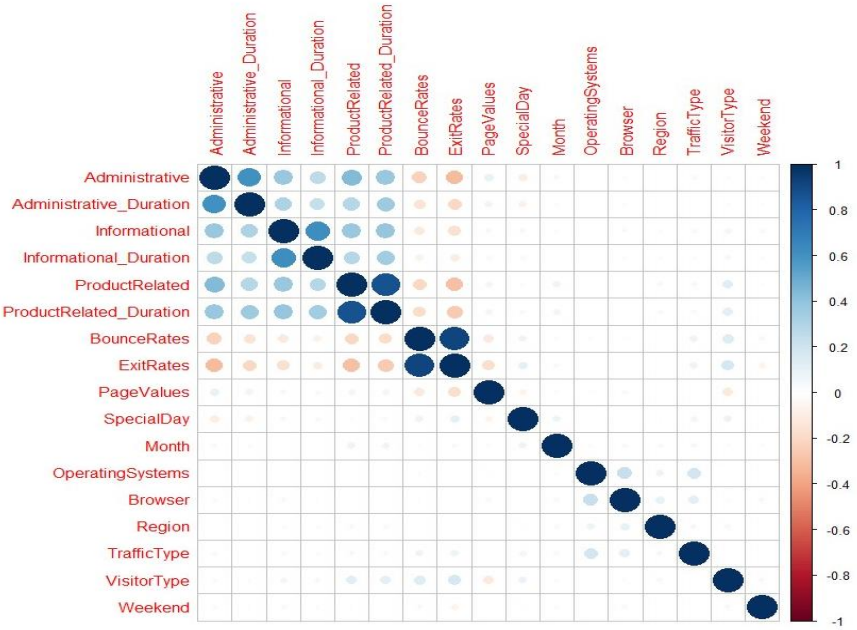

Figure 02

2. ggplot to find the K value.

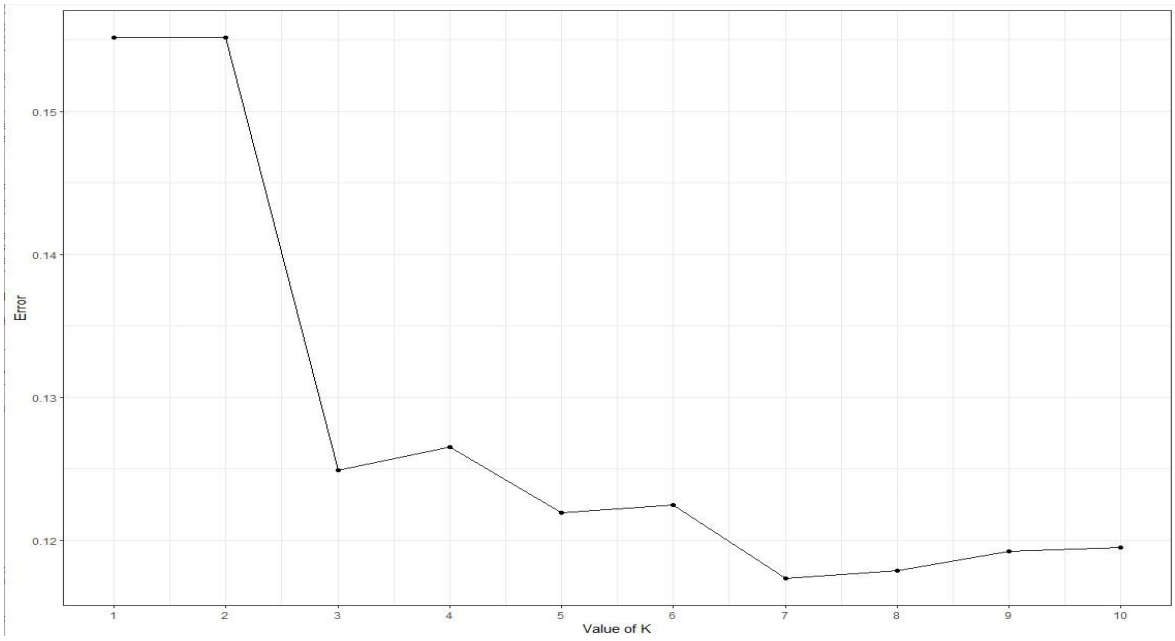

Figure 03

3.3. Implementation in R

- Install packages to R.
- The R packages installed are as follows:
  1. **caTools (caretTools)** – this is used to split the dataset to train and test the dataset since the KNN model will use this model to learn and evaluate the dataset.
  2. **dplyr** – this is used to manipulate the data since the KNN model is required to clean and process the dataset. dplyr allows the user to achieve these tasks in a user-friendly manner.
  3. **ggplot2** – this is used for data visualization where it allows to visualize the data before and after modelling the dataset.
  4. **class** – this implements core KNN functionality which allows the model to calculate the distances between data points, identify the K nearest neighbours for each test point, and classifies them.
  5. **caret** – this is a machine learning framework which offers a comprehensive toolkit for building, tuning, and evaluating machine learning models, including KNN. This also streamlines the modelling process.
  6. **corrplot** – this is a correlation plot generation which helps in visualizing the relationships between features in the data of the dataset and aids in understanding feature relationships of the dataset.

```
1  install.packages('caTools') #for train and test data split
2  install.packages('dplyr') #for Data Manipulation
3  install.packages('ggplot2') #for Data Visualization
4  install.packages('class') #KNN
5  install.packages('caret') #Confusion Matrix
6  install.packages('corrplot') #Correlation Plot
```

Figure 04

- Calling out the required libraries for the classification of the dataset.

```
8  library(caTools)
9  library(dplyr)
10 library(ggplot2)
11 library(caret)
12 library(class)
13 library(corrplot)
```

Figure 05

- Import and view the dataset. For our convenience, checking the head (which shows the 1st 6 rows of the dataset), the str (string) and the summary of the dataset and the dim (which shows the dimension, the number of rows and the columns of the dataset).

```
15 #import the dataset
16 data<-read.csv("online_shoppers_intention.csv")
17 head(data)
18 str(data)
19 summary(data)
20 dim(data)
```

Figure 06

- Converting the datatype from character to integer.

```
22  #Exchanging character to Integer
23  data$Month<-as.factor(data$Month)
24  data$Month<-as.integer(data$Month)
25  table(data$Month)
26
27  data$VisitorType<-as.factor(data$VisitorType)
28  data$VisitorType<-as.integer(data$VisitorType)
29  table(data$VisitorType)
30
31  head(data)
```

Figure 07

- Checking if there are any null values in the dataset.

```
33  #checking for any not available in the dataset
34  anyNA(data)
```

Figure 08

- Extracting the labels from the dataset.

```
36  #Extracting the labels
37  table(data$Revenue)
38  round(prop.table(table(data$Revenue))*100,digits = 1)
```

Figure 09

- Removing outliers.

```
40  #Removing outliers
41  standard.features <- scale(data[,1:17])
42  standard.features
43
```

Figure 10

- Renaming the dataset after removing the outliers.

```
44  #Renaming our dataset after removing outliers
45  data1 <- cbind(standard.features,data[18])
46  head(data1)
47  anyNA(data1)
48
```

Figure 11

- Creating the corrplot to check the performance using the confusion matrix.

```
49  #creating the corplot
50  corrplot(cor(data1[,-18]))
```

Figure 12

- Splitting to train and test the dataset.

```
52  #splitting data
53  set.seed(101)
```

Figure 13

- Training the dataset where 70% of the dataset which was split was used to train the dataset.

```
55  #Training the data
56  sample <- sample.split(data1$Revenue,SplitRatio = 0.70)
57  train <- subset(data1,sample==TRUE)
58  dim(train)
```

Figure 14

- Testing the dataset where 30% of the dataset which was split was used to test the dataset.

```
60  #Testing the data
61  test <- subset(data1,sample==FALSE)
62  dim(test)
```

Figure 15

```
> #splitting data
> set.seed(101)
> #Training the data
> sample <- sample.split(data1$Revenue,SplitRatio = 0.70)
> train <- subset(data1,sample==TRUE)
> dim(train)
[1] 8631   18
> #Testing the data
> test <- subset(data1,sample==FALSE)
> dim(test)
[1] 3699   18
> #using the KNN model
> predicted.type <- knn(train[,1:17], test[,1:17], train$Revenue,k=1)
> predicted.type
```

Figure 16

- Using the KNN model for the classification of the dataset.

```
64  #using the KNN model
65  predicted.type <- knn(train[,1:17], test[,1:17], train$Revenue,k=1)
66  predicted.type
```

Figure 17

- Checking the error value in the prediction when K = 1.

```
68  #Error in prediction
69  error <- mean(predicted.type!=test$Revenue)
70  error
```

Figure 18

```
> #Error in prediction
> error <- mean(predicted.type!=test$Revenue)
> error
[1] 0.1551771
> #Confusion Matrix
> confusionMatrix(predicted.type,as.factor(test$Revenue),mode="everything")
Confusion Matrix and Statistics

          Reference
Prediction FALSE TRUE
     FALSE  2861  308
     TRUE    266  264

               Accuracy : 0.8448
                 95% CI : (0.8327, 0.8564)
    No Information Rate : 0.8454
    P-Value [Acc > NIR] : 0.54730

                  Kappa : 0.3881

 Mcnemar's Test P-Value : 0.08702

            Sensitivity : 0.9149
            Specificity : 0.4615
         Pos Pred Value : 0.9028
         Neg Pred Value : 0.4981
              Precision : 0.9028
                 Recall : 0.9149
                     F1 : 0.9088
             Prevalence : 0.8454
         Detection Rate : 0.7735
   Detection Prevalence : 0.8567
      Balanced Accuracy : 0.6882

       'Positive' Class : FALSE
```

Figure 19

- Confusion matrix.

```
#Confusion Matrix
confusionMatrix(predicted.type,as.factor(test$Revenue),mode="everything")

```

Figure 20

- Testing the dataset for alternative K values.

```
#Testing alternative k values
predicted.type <- NULL
error.rate <- NULL
for (i in 1:10) {
  predicted.type <- knn(train[1:17],test[1:17],train$Revenue,k=i)
  error.rate[i] <- mean(predicted.type!=test$Revenue)
}
knn.error <- as.data.frame(cbind(k=1:10,error.type =error.rate))
knn.error

```

Figure 21

- Creating a ggplot for K values.

```
#creating a ggplot for k values
ggplot(knn.error,aes(k,error.type))+
  geom_point()+
  geom_line() +
  scale_x_continuous(breaks=1:10)+
  theme_bw() +
  xlab("Value of K") +
  ylab('Error')
```

Figure 22

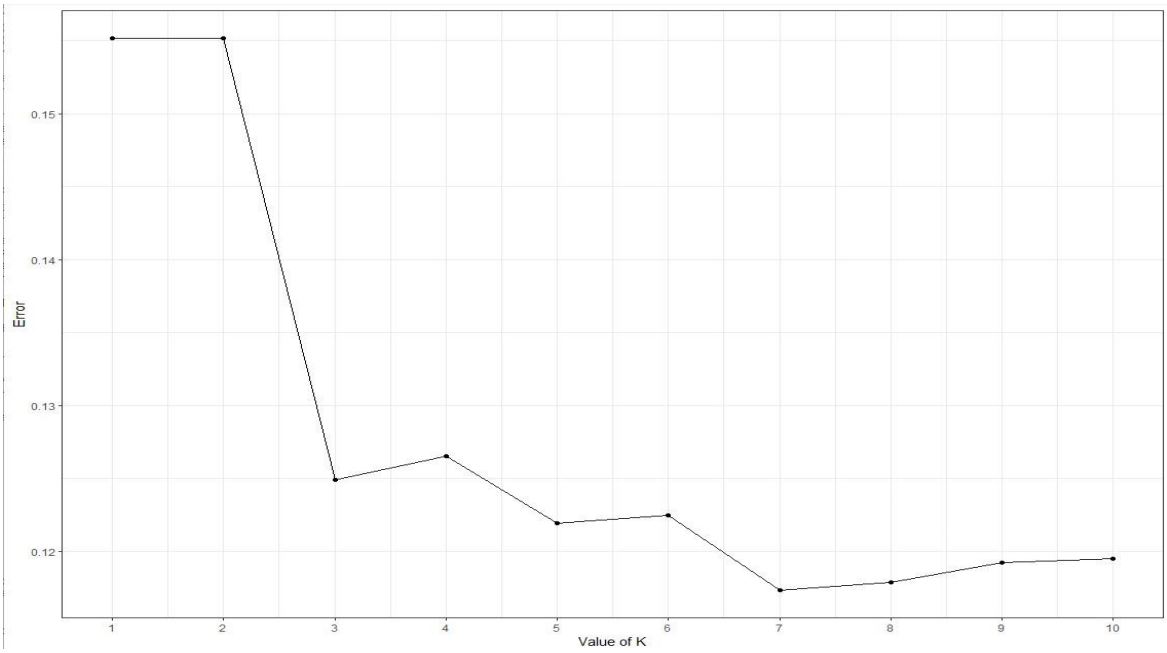

Figure 03

- Improving the performance of the model.

```
94  #improving the model performance
95  data_test_prediction<-knn(train[,-1],test[,-1],train$Revenue,k=7)
96  data_test_prediction
97
```

Figure 23

- Checking the error value of the prediction when K = 7.

```
98  #checking the error value for k=7
99  error<-mean(data_test_prediction!=test$Revenue)
100 error
```

Figure 24

```
> #checking the error value for k=7
> error<-mean(data_test_prediction!=test$Revenue)
> error
[1] 0.06163828
> #Rechecking the accuracy
> library(caret)
> confusionMatrix(data_test_prediction,as.factor(test$Revenue),mode="everything")
Confusion Matrix and Statistics

          Reference
Prediction FALSE TRUE
     FALSE  3096  197
     TRUE     31  375

               Accuracy : 0.9384
                 95% CI : (0.9301, 0.9459)
    No Information Rate : 0.8454
    P-Value [Acc > NIR] : < 0.00000000000000022

                  Kappa : 0.7325

 Mcnemar's Test P-Value : < 0.00000000000000022

            Sensitivity : 0.9901
            Specificity : 0.6556
         Pos Pred Value : 0.9402
         Neg Pred Value : 0.9236
              Precision : 0.9402
                 Recall : 0.9901
                     F1 : 0.9645
             Prevalence : 0.8454
         Detection Rate : 0.8370
   Detection Prevalence : 0.8902
      Balanced Accuracy : 0.8228

       'Positive' Class : FALSE
```

Figure 25

- Rechecking the accuracy of the prediction.

```
102 #Rechecking the accuracy
103 library(caret)
104 confusionMatrix(data_test_prediction,as.factor(test$Revenue),mode="everything")
```

Figure 26

3.4. Results analysis and discussion

The classification of the dataset is predicted by using the class label, the target variable "Revenue". The KNN model is used for the classification of the dataset to predict the dataset by training and testing the already split dataset.

The dataset is split where 70% of the dataset is used to train the dataset where the model learns about the variables and learns how each independent feature variable is used to predict the dependant target variable while the rest of the 30% of the dataset is used to evaluate the unseen data.

Accordingly, after splitting, training and testing the dataset using the KNN model, we assign a value to K to get the rate of accuracy and rate of error of the prediction. Since, we do not know which K value gives the highest rate of accuracy and lowest rate of error, we initially assigned the K value as 1 and got the accuracy rate of 0.8448 which is 84.48% as the result.
Then, a ggplot was created to get the K values in order to optimize and improve the performance of the model and then we later assign the K value as 7 after analysing the ggplot curve where we got the accuracy rate of 0.9384 which is 93.84% as the result.

4. Clustering

4.1. Explanation and preparation of datasets

The dataset utilized for the classification is the purchasing intention of online shoppers which consists of 12,330 data records.

Since the dataset is already a clean dataset, no missing values were found. Therefore, no preparatory tasks were carried on the dataset.

The dependant variable is the “Revenue” column which is also considered to be the class label of the dataset while the independent variables are the rest of the columns such as “Administrative”, “BounceRates”, “TrafficType” and “Region” are some of them.

For the clustering of the dataset, the target variable, “Revenue” is removed where only the feature variables such as “Administrative”, “Informational Duration”, “BounceRates”, “Region” and etc. are used for the clustering of the dataset. The dataset is clustered into clusters where the quality of the clusters is evaluated.

4.2. Data mining

The data mining technique used for the dataset is clustering using R programming language.

The clustering technique used for the clustering of the dataset is K means clustering technique.

Using K means clustering, the model splits the dataset into the optimum number of clusters the dataset can be clustered to.

Visualization tools used:

1. stats and ggplot2 are used to create the wss plot to choose the maximum number of clusters in the dataset.

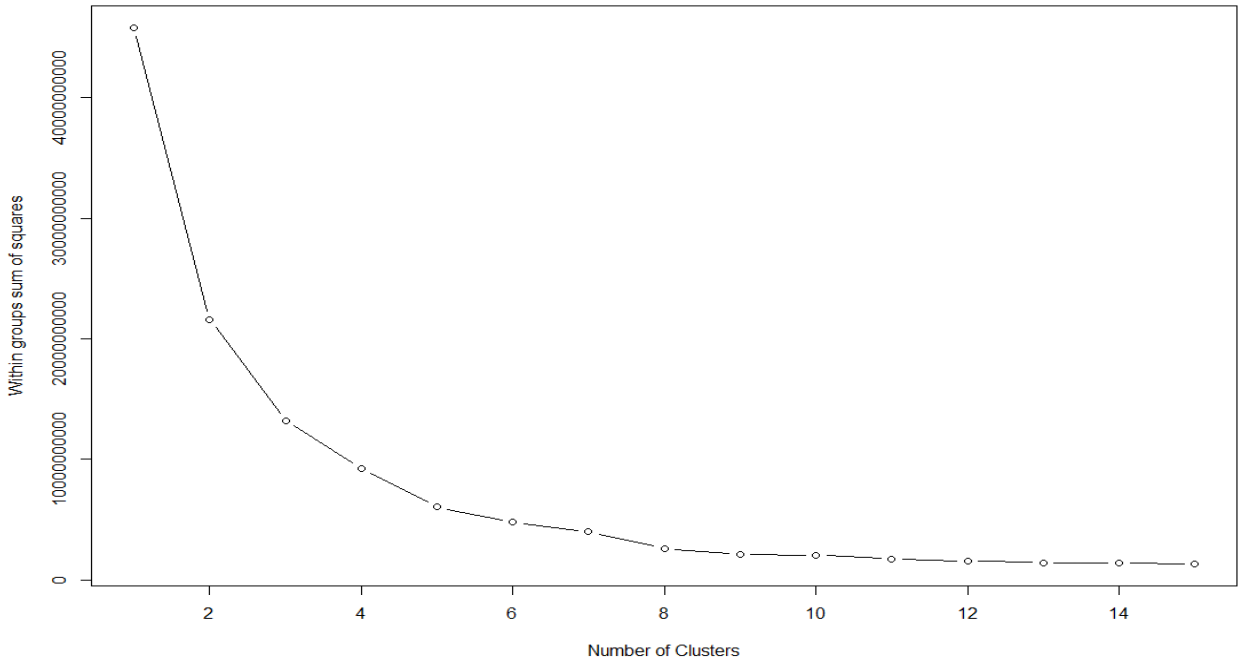

Figure 27

2. ggplot2, factoextra and ggfortify are used to create the cluster analysis.

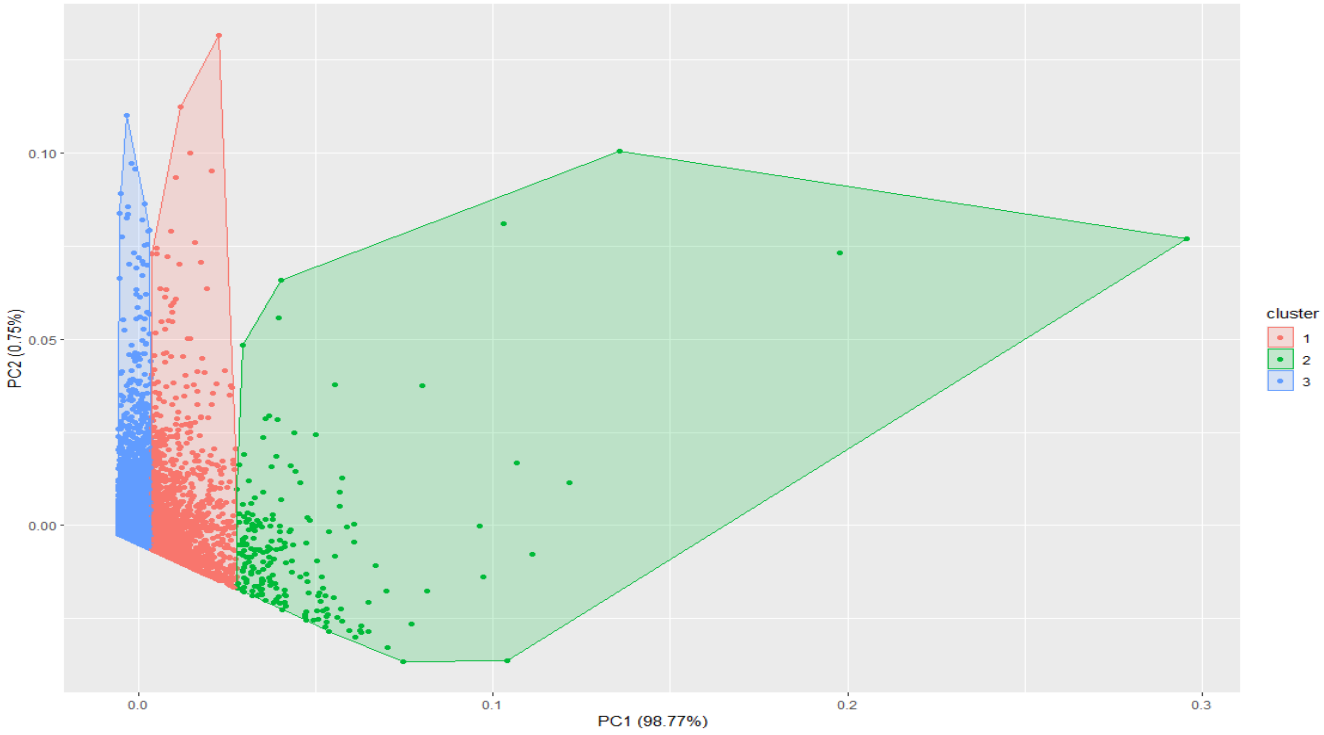

Figure 28

4.3. Implementation in R

- Install packages to R.
- The R packages installed are as follows:
  1. **ggfortify** – this is an extended version of ggplot2 which is used to create advanced forms of cluster visualizations.
  2. **cluster** – this provides the core K means functionality for various clustering algorithms.
  3. **factoextra** – focuses on multivariate analysis and exploratory data analysis which helps the model to determine the optimal number of clusters. This is also a data visualization tool which highlights the potential clusters.
  4. **dplyr** – this helps to manipulate the data like cleaning, filtering, and transforming the data before clustering.
  5. **ggplot2** – this allows to create informative visualizations before and after clustering the data. this is used to explore the distribution of data, identify potential clusters visually and assess the effectiveness of the clusters.
  6. **stats** – this helps to prepare the data by providing various statistical functions such as calculating the summary statistics or distance measures used in the algorithm.

```
#install pre-required packages
install.packages("ggfortify")
install.packages("cluster")
install.packages("factoextra")

#load required libraries
library(dplyr)
library(ggplot2)
library(stats)
library(ggfortify)
library(cluster)
library(factoextra)

```

Figure 29

- Calling out the required libraries for the clustering of the dataset.

```
#Import the dataset
my_data<-read.csv("online_shoppers_intention.csv")

view(my_data)
head(my_data)
str(my_data)
summary(my_data)

dim(my_data)
```

Figure 30

- Importing the dataset and checking the head, string, the summary and the dimension of the dataset.

```
25 #Convert the data type : char to int
26 my_data$Month<-as.factor(my_data$Month)
27 my_data$Month<-as.integer(my_data$Month)
28 table(my_data$Month)
29
30 my_data$VisitorType<-as.factor(my_data$VisitorType)
31 my_data$VisitorType<-as.integer(my_data$VisitorType)
32 table(my_data$VisitorType)
33
34 head(my_data)
```

Figure 31

- Changing the data type from character to integer in the "Month" and the "VisitorType" variables

```
25 #Convert the data type : char to int
26 my_data$Month<-as.factor(my_data$Month)
27 my_data$Month<-as.integer(my_data$Month)
28 table(my_data$Month)
29
30 my_data$VisitorType<-as.factor(my_data$VisitorType)
31 my_data$VisitorType<-as.integer(my_data$VisitorType)
32 table(my_data$VisitorType)
33
34 head(my_data)
```

Figure 32

- Checking the dataset for null values.

```
36 #checking for any not available in the dataset
37 anyNA(my_data)
```

Figure 33

- Removing the target variable, "Revenue" from the dataset for the clustering of the dataset.

```
39 #Removing the labeled column
40 my_data<-my_data[1:17]
```

Figure 34

- Using the stats and ggplot2 to create the wss plot to choose the maximum number of clusters in the dataset.

```
#wss plot to choose maximum number of clusters
wssplot <- function(data, nc=15, seed=1234){
  wss <- (nrow(data)-1)*sum(apply(data,2,var))
  for (i in 2:nc){
    set.seed(seed)
    wss[i] <- sum(kmeans(data, centers=i)$withinss)}
  plot(1:nc, wss, type="b", xlab="Number of Clusters",
       ylab="within groups sum of squares")
  wss
}
wssplot(my_data)
```

Figure 35

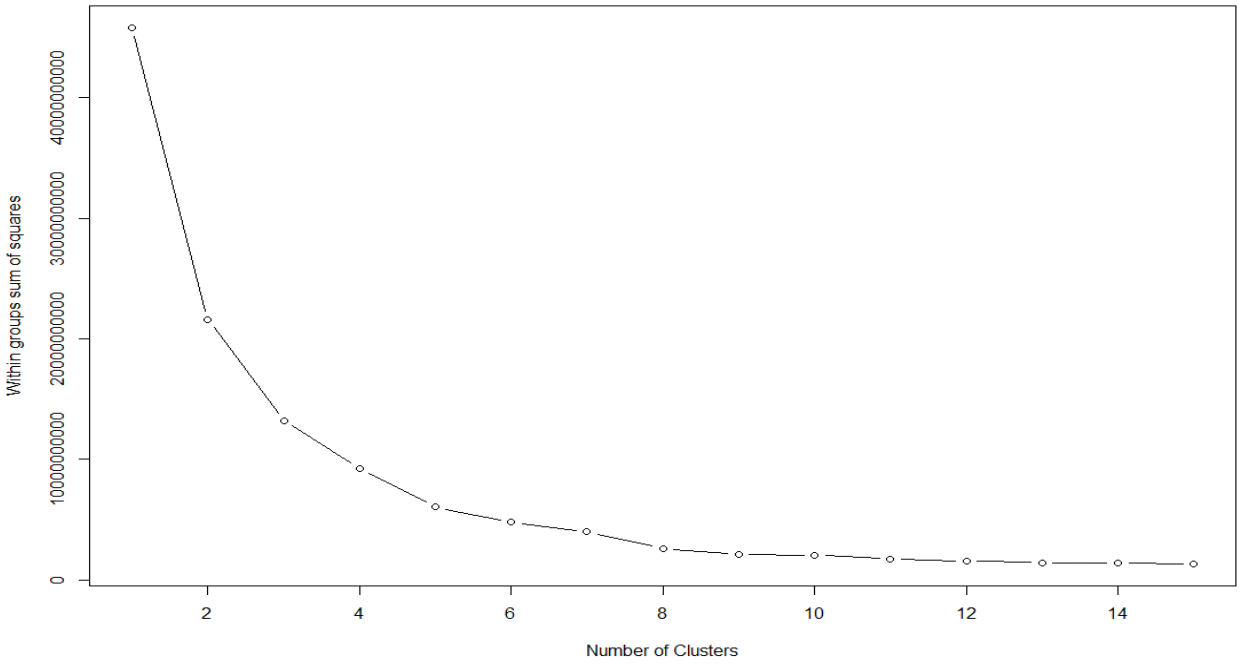

Figure 27

- Spotting the k means in the curve in order to choose the optimum number of clusters, where the no. of clusters = 3. This is where ggplot2, factoextra and ggfortify are used to create the cluster analysis

```
#Spotting the k means in the curve in order to choose the optimum number of cluster=3
KM=kmeans(my_data,3)
KM
```

Figure 36

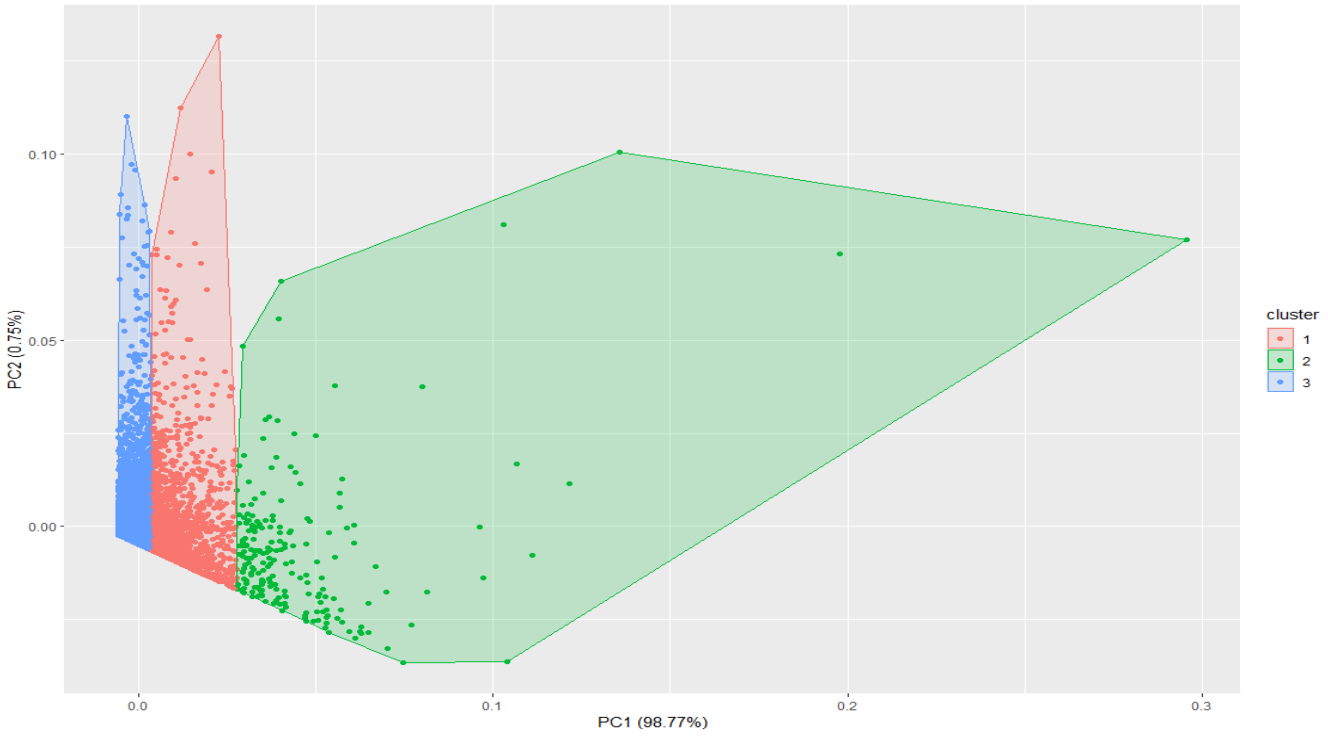

Figure 28

- Evaluating the cluster analysis.

```
59  #Evaluating cluster Analysis
60  autoplot(KM,my_data,frame=TRUE)  #cluster plot
61
```

Figure 37

```
> #Spotting the k means in the curve in order to choose the optimum number of cluster=3
> KM=kmeans(my_data,3)
> KM
K-means clustering with 3 clusters of sizes 10148, 1975, 207

Cluster means:
  Administrative Administrative_Duration Informational Informational_Duration ProductRelated ProductRelated_Duration
1       1.819472                62.02896     0.3272566                19.96497       18.61047                576.6107
2       4.325063               154.87748     1.1741772                84.67262       77.72962               3355.1075
3       7.439614               295.36489     2.7487923               266.72309      236.10628              10886.1744
  BounceRates  ExitRates PageValues SpecialDay    Month OperatingSystems  Browser   Region TrafficType VisitorType
1 0.025454467 0.04771509   5.503715 0.06413086 6.086618         2.121108 2.367166 3.169196    4.148699    2.677178
2 0.007128234 0.02167056   8.013634 0.05073418 6.496709         2.136203 2.310380 3.094177    3.710380    2.901772
3 0.005939756 0.01968831   4.521370 0.03091787 6.782609         2.149758 2.309179 2.584541    3.618357    2.985507
    Weekend
1 0.2348246
2 0.2192405
3 0.2512077
```

Figure 38 (a)

```
Clustering vector:
   [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1
  [56] 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 2
 [111] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [166] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 2 1 2 1 1 1 2 2 2 1 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1
 [221] 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 2 2 2 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1
 [276] 1 1 2 1 1 1 1 1 1 1 1 1 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 2 1 1
 [331] 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 2 1 1 1 1 2 1 1 1 2 1 1 1 1 1 2 1 1 1 1 1
 [386] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1
 [441] 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 2 1 1 1 1 1 2 1 1 1 1 1 2 2 1 1 2 1 1 1 1 1 1 2 1 1
 [496] 1 1 1 1 2 1 1 2 2 1 1 1 1 1 1 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [551] 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 2 1 1
 [606] 1 1 1 1 1 2 2 2 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1
 [661] 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 2 1 1 2 1 1 1
 [716] 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 2 2 1 1 1 2 2 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 2 1 1
 [771] 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 3 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1
 [826] 2 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1
 [881] 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 1 1 1 1 2 1 1 2 1 1 1 1 1 1 1 2 1 1 1 1 1 2
 [936] 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 2 1 1 1 1 1 1 1 1 3 1 1 1 1 1 1
 [991] 1 1 1 1 1 1 2 1 1 1
 [ reached getOption("max.print") -- omitted 11330 entries ]

Within cluster sum of squares by cluster:
[1] 3100362409 3291367158 6823243412
 (between_SS / total_SS =  71.2 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss" "betweenss"    "size"
[8] "iter"         "ifault"
```

Figure 38 (b)

- Getting the cluster centres from the dataset. They are the central points formed from each cluster and these cluster points show the average values of each cluster.

```
62  #cluster centers
63  KM$centers
```

Figure 39

```
> #cluster centers
> KM$centers
  Administrative Administrative_Duration Informational Informational_Duration ProductRelated ProductRelated_Duration BounceRates  ExitRates
1       4.325063              154.87748     1.1741772              84.67262       77.72962              3355.1075 0.007128234 0.02167056
2       7.439614              295.36489     2.7487923             266.72309      236.10628             10886.1744 0.005939756 0.01968831
3       1.819472               62.02896     0.3272566              19.96497       18.61047               576.6107 0.025454467 0.04771509
  PageValues SpecialDay    Month OperatingSystems  Browser   Region TrafficType VisitorType   Weekend
1   8.013634 0.05073418 6.496709         2.136203 2.310380 3.094177    3.710380    2.901772 0.2192405
2   4.521370 0.03091787 6.782609         2.149758 2.309179 2.584541    3.618357    2.985507 0.2512077
3   5.503715 0.06413086 6.086618         2.121108 2.367166 3.169196    4.148699    2.677178 0.2348246
```

Figure 40

4.4. Results analysis and discussion

The clustering technique in data mining allows us to identify patterns and analyse them in large datasets where we will be able to identify trends and outliers easily which leads to effective decision making.

Here, the clustering technique utilized for this dataset is the K means clustering technique where only the feature variables are utilized in identifying and analysing the trends while the target variable is removed during the process.

Accordingly, using the K means technique the maximum optimum number of clusters for the dataset is chosen can be identified which is 3 as per the "elbow point" of the wss plot. Through the cluster analysis, we can identify the size of the 3 clusters which are as follows; the size of Cluster 01 is 1975, the size of Cluster 02 is 207 and the size of the Cluster 03 is 10148 where the total of the cluster sizes is equal to 12330 which is equivalent to the amount of data in the dataset.

Cluster 01 with the cluster size of 1975, have high values for "Administrative", "Administrative_Duration", "Informational", "Informational_Duration", "ProductRelated", "ProductRelated_Duration" and "PageValues" while "BounceRates" and "ExitRates" have relatively low values which represents a specific group of customers with common characteristics since they spend a significant amount of time on the website and visit various pages resulting in higher page values.

Cluster 02 with the cluster size of 207, have very high values for "Administrative", "Administrative_Duration", "Informational", "Informational_Duration", "ProductRelated", "ProductRelated_Duration" and "PageValues" while "BounceRates" and "ExitRates" have relatively low values where this cluster may be seen similar to the Cluster 01, however, Cluster 02 represents a smaller group of customers with distinct behaviours and characteristics which indicates that they are a more engaged group of customers.

Cluster 03 with the cluster size of 10148, have a relatively lower values for "Administrative", "Administrative_Duration", "Informational", "Informational_Duration", "ProductRelated," "ProductRelated_Duration, PageValues" while "BounceRates", "ExitRates" have a relatively higher value for "BounceRates", "ExitRates" and higher value to look forward for "SpecialDay" which a specific group of customers represents a group of children with more general or rational characteristics as they spend less time on the website and visit fewer pages resulting in lower page values.

5. Conclusions

This dataset about the purchasing intention of online shoppers which consists of 12,330 data. The dataset consists of 18 variables where 17 of them are feature variables while the last variable is a target variable which is also considered to be as the class label.

Both classification and clustering data mining techniques are used in this dataset.

The target variable, "Revenue" is chosen to predict the dataset. The K – Nearest Neighbour (KNN) model is used for the classification of the dataset where, 70% of the dataset is split up and was used to train the dataset while, the remaining 30% of the dataset is used to test, evaluate and assess the dataset.

Accordingly, after the splitting, training and testing of the dataset, we will predict a revenue where when a revenue is generated from the online purchase, the revenue will be shown as "TRUE" while when there is no revenue generated from each purchase, the revenue will be shown as "FALSE".

However, when it comes to the clustering technique used on the dataset, the target variable is removed where only the feature variables are used to identify and analyse the trends in the dataset by sub grouping the dataset into separate clusters, in this scenario the maximum optimum number of clusters is 3.

The purpose of selecting this dataset is to identify the trends and to identify whether online shopping websites generates a revenue or not since revenue is essential for the continuation of the business.

Through the data mining techniques utilized, we can analyse and predict that the online shopping websites do not generate much revenue to ensure the continuous survival of the business. The fact that the revenue of the online websites has no significant increase, a high trend of revenue is not generated.

Therefore, it is essential that the online shop owners analyse and understand the behaviour and the intentions of the customers and the online visitors and take necessary measures and actions such as implementing various marketing strategies and pricing strategies in order to maximise and optimise the revenue generated by the online stores since analysing the data related to the online stores and the customers and their intentions is important to be understood by the store owners in order to contribute to the accumulation of the store's earnings and the revenue.

6. References

1. archive.ics.uci.edu. (n.d.). *UCI Machine Learning Repository*. [online] Available at: https://archive.ics.uci.edu/dataset/468/online+shoppers+purchasing+intention+dataset.

2. Sakar, C.O., Polat, S.O., Katircioglu, M. and Kastro, Y. (2018). Real-time prediction of online shoppers' purchasing intention using multilayer perceptron and LSTM recurrent neural networks. *Neural Computing and Applications*, [online] 31(10), pp.6893–6908. doi:https://doi.org/10.1007/s00521-018-3523-0.

7. Appendices

***Figure 01*:** Online Shoppers Purchasing Intention Dataset.
***Figure 02*:** corrplot to find performance using the confusion matrix.
***Figure 03*:** ggplot to find the K value using classification.
***Figure 04*:** installing packages from R for the classification of the dataset.
***Figure 05*:** calling out the required libraries for the classification of the dataset.
***Figure 06*:** importing and viewing the datasets head, string, summary and the dimension of the dataset.
***Figure 07*:** converting the datatype from character to integer.
***Figure 08*:** checking if there are any null values in the dataset.
***Figure 09*:** extracting the labels from the dataset.
***Figure 10*:** removing the outliers from the model.
***Figure 11*:** renaming the dataset after removing the outliers.
***Figure 12*:** creating the corrplot to check the performance using the confusion matrix.
***Figure 13*:** splitting the dataset.
***Figure 14*:** training 70% of the dataset.
***Figure 15*:** testing 30% of the dataset.
***Figure 16*:** results from splitting, training and testing the K model.
***Figure 17*:** using the KNN model.
***Figure 18*:** checking the error value in the prediction when K = 1.
***Figure 19*:** the result from checking the error value when K = 1.
***Figure 20*:** using the confusion matrix for the prediction.
***Figure 21*:** testing the dataset for alternative K values.
***Figure 22*:** creating a ggplot for K values.
***Figure 23*:** improving the performance of the model.
***Figure 24*:** checking the error value of the prediction when K = 7.
***Figure 25*:** the result from checking the error value when K = 7.
***Figure* 26:** rechecking the accuracy of the prediction.
***Figure 27*:** wss plot to choose the maximum number of clusters in the dataset.
***Figure 28*:** cluster analysis.
***Figure 29*:** installing packages from R for the clustering of the dataset.
***Figure 30*:** calling out the required libraries for the clustering of the dataset.
***Figure 31*:** importing the dataset and checking the head, string, the summary and the dimension of the dataset.
***Figure 32*:** changing the data type from character to integer in the "Month" and the "VisitingType" variables of the dataset.
***Figure 33*:** checking the dataset for null values.
***Figure 34*:** removing the target variable, "Revenue" from the dataset for the clustering of the dataset.
***Figure 35*:** using the stats and ggplot2 to create the wss plot to choose the maximum number of clusters in the dataset.

***Figure 36*:** spotting the k means in the curve in order to choose the optimum number of clusters, where the no. of clusters = 3.
***Figure 37*:** evaluating the cluster analysis.
***Figure 38 (a)*:** the result from evaluating the cluster analysis.
***Figure 38 (b)*:** the result from evaluating the cluster analysis.
***Figure 39*:** getting the cluster centres from the dataset.
***Figure 40*:** the result after getting the cluster centres.