General Sir John Kotelawala Defense University

Faculty of Management, Social Sciences and Humanities

Department of Languages

BSc in Applied Data Science Communication

Clustering and Classification using R

1.D/ADC/23/0010 - C.S.Rathnasekara

2.D/ADC/23/0011 - A.M.K.D.Bandara

3.D/ADC/23/0019 - B.K.M.M.Hansanee

4.D/ADC/23/0027- P.G.D.Rodrigo

Fundamentals of Data Mining/LB2114

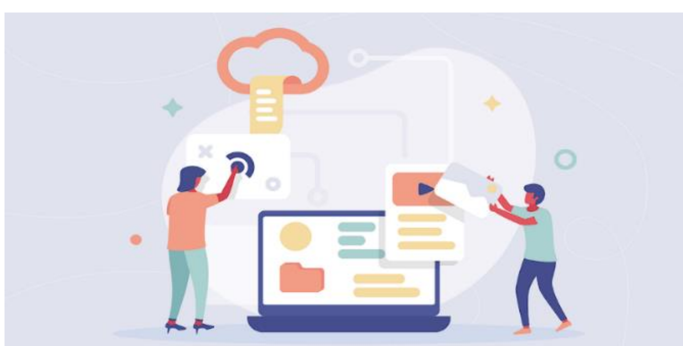
Year 2: Semester 1

Assignment 1

Table of Contents

01.Classification

Maternal Health Risk Prediction Classification



1.Introduction

Risks to the mother's health during pregnancy, childbirth and the post-natal period are included in this category. These risks include maternal mortality, obstetric complications such as eclampsia and obstructed labor, maternal morbidity such as gestational diabetes and postpartum depression, and difficulties in accessing prenatal care due to socioeconomic disparities. These risks are influenced by factors such as maternal age, health status and access to health services.

In addition to measures to address socioeconomic issues and promote women's rights and education, addressing these hazards calls for comprehensive health care programs, including increased access to prenatal care, skilled birth attendance, and emergency obstetric services. We can reduce the burden of preventable maternal mortality and morbidity and improve the health of mothers and babies by addressing maternal health issues holistically.

2.Datasets

This dataset was taken from https://archive.ics.uci.edu/dataset/863/maternal+health+risk on the "OCI" website.

The dataset's aim is to predict the risk of the maternal health. The dataset consists of 7 columns and 1015 rows. The character variable of the dataset is risk level. It is a character variable that tells the outcome of the prediction.

Data has been collected from different hospitals, community clinics, maternal health cares from the rural areas of Bangladesh.

3.Explanation and Preparation of the datasets

1. Age – Age in years
2. Systolic Blood Pressure as SystolicBP – Blood pressure is measured using two ways. First one is systolic blood pressure. It measures pressure in your arteries when your heart beats.
3. DiastolicBP- It measures pressure in your arteries when your heart rests between beats.
4. Blood Sugar as BS – Main sugar found in the blood. If you had a fasting blood glucose test, level between 70 and 100 mg is the normal.
5. Body Temperature as BodyTemp – The average body temperature is 98.6F
6. HeartRate – The number of times the heart beats within a certain time period.
7. RiskLevel

The risk level column shows "high risk", "low risk", "mid risk" as outcomes. These are main outputs as well as the main classes of the dataset. To predict the maternal health risk of mothers who are in this dataset, we use the classification method which is very popular in data mining to categorize the distinct classes of the respective dataset.

4.Data Mining

The process of removing patterns and insights from large data sets using statistical and computational methods is called data mining. It is applied in many different domains to uncover patterns and hidden correlations that can be used for strategic planning and decision-making. Methods of data mining that we frequently employ include regression analysis, association rule mining, clustering, and classification.

The application of Data mining techniques

Classification

The goal is to apply classification to the crop recommendation data set. Classification is the categorization of distinct elements into different classes. There are two types of classification issues: binary (only two classes are feasible) and multi-class. (With more than two possible classes).

K-Nearest Neighbor classifier (KNN)

K nearest neighbors is a machine learning approach for regression analysis and classification. While there are a variety of metrics that can be used to identify the nearest neighbors, some common ones are the Minkowski, Manhattan and Euclidean distances.

$$d(x,y) = \sqrt{\sum_{i=1}^{n} (y_i - x_i)^2}$$

x,y – two points in Euclidean n-space

yi, xi – Euclidean vectors, starting from the origin of the space.

n – n space

5.Implementation in R

Data preprocessing

Before stepping in to the data preprocessing, as the first step, we must install and load 6 packages which are mentioned below.

```r
#Install required packages and libraries
install.packages('caTools')
install.packages('dplyr')
install.packages('ggplot2')
install.packages('class')
install.packages('caret')
install.packages('corrplot')

library(caTools)
library(dplyr)
library(ggplot2)
library(caret)
library(class)
library(corrplot)
```

caTools() - A wide range of utility functions, including as those for data splitting, generating statistical data, and model evaluation, are included in the R caTools package.

Dplyr() - Filtering, summarizing, modifying, and organizing data frames is made simple and straightforward with the help of dplyr, a crucial R program for data manipulation.

Ggplot2() – with ggplot2, an adaptable R software, you can use a layered syntax of graphics to produce publication-ready, high-quality visualizations.

Class() - R's class package offers functions to identify and classify R objects, making object-oriented programming and data manipulation jobs easier.

Caret() - Model-building is made easier using the caret package in R, which offers a uniform interface for training and assessing various machine learning models. Its consistent syntax and capabilities make designing models easier.

Corrplot() - The R corrplot package offers tools for visualizing correlation matrices with graphical parameters and color schemes that may be easily customized, facilitating the exploration and interpretation of correlations between variables.

Import the Maternal health risk dataset in to R.

```r
#Import the dataset into R
MH<- read.csv("Maternal Health Risk Data Set.csv",header=TRUE,sep=","
              , col.names = c("Age","SystolicBP","DiastolicBP","BP",
               "BodyTemp","HeartRate","RiskLevel"))
```

Inspect dataset

View the summary of the imported data set

```
> summary(MH)
      Age          SystolicBP     DiastolicBP          BP        
 Min.   :10.00   Min.   : 70.0   Min.   : 49.00   Min.   : 6.000  
 1st Qu.:19.00   1st Qu.:100.0   1st Qu.: 65.00   1st Qu.: 6.900  
 Median :26.00   Median :120.0   Median : 80.00   Median : 7.500  
 Mean   :29.87   Mean   :113.2   Mean   : 76.46   Mean   : 8.726  
 3rd Qu.:39.00   3rd Qu.:120.0   3rd Qu.: 90.00   3rd Qu.: 8.000  
 Max.   :70.00   Max.   :160.0   Max.   :100.00   Max.   :19.000  
    BodyTemp        HeartRate      RiskLevel        
 Min.   : 98.00   Min.   : 7.0   Length:1014       
 1st Qu.: 98.00   1st Qu.:70.0   Class :character  
 Median : 98.00   Median :76.0   Mode  :character  
 Mean   : 98.67   Mean   :74.3                     
 3rd Qu.: 98.00   3rd Qu.:80.0                     
 Max.   :103.00   Max.   :90.0                     
```

Explore the dataset

Using the below codes the names,first rows values,last rows values, summary of the data set,and structure of the data set can be identified.

```r
#Explore the dataset
names(MH)
head(MH)
tail(MH)
summary(MH)
str(MH)
#Removing outliers
```

Use ‘name()’ function to understand the main categories of this dataset.

```
> names(MH)
[1] "Age"          "SystolicBP"  "DiastolicBP" "BP"          "BodyTemp"
[6] "HeartRate"    "RiskLevel"
> head(MH)
  Age SystolicBP DiastolicBP    BP BodyTemp HeartRate RiskLevel
1  25        130          80 15.00       98        86 high risk
2  35        140          90 13.00       98        70 high risk
3  29         90          70  8.00      100        80 high risk
4  30        140          85  7.00       98        70 high risk
5  35        120          60  6.10       98        76  low risk
6  23        140          80  7.01       98        70 high risk
> tail(MH)
     Age SystolicBP DiastolicBP BP BodyTemp HeartRate RiskLevel
1009  48        120          80 11       98        88 high risk
1010  22        120          60 15       98        80 high risk
1011  55        120          90 18       98        60 high risk
1012  35         85          60 19       98        86 high risk
1013  43        120          90 18       98        70 high risk
1014  32        120          65  6      101        76  mid risk
```

Use the ‘summary()’ function to take the summary of this dataset.

```
> summary(MH)
      Age          SystolicBP      DiastolicBP           BP        
 Min.   :10.00   Min.   : 70.0   Min.   : 49.00   Min.   : 6.000  
 1st Qu.:19.00   1st Qu.:100.0   1st Qu.: 65.00   1st Qu.: 6.900  
 Median :26.00   Median :120.0   Median : 80.00   Median : 7.500  
 Mean   :29.87   Mean   :113.2   Mean   : 76.46   Mean   : 8.726  
 3rd Qu.:39.00   3rd Qu.:120.0   3rd Qu.: 90.00   3rd Qu.: 8.000  
 Max.   :70.00   Max.   :160.0   Max.   :100.00   Max.   :19.000  
    BodyTemp         HeartRate      RiskLevel        
 Min.   : 98.00   Min.   : 7.0   Length:1014       
 1st Qu.: 98.00   1st Qu.:70.0   Class :character  
 Median : 98.00   Median :76.0   Mode  :character  
 Mean   : 98.67   Mean   :74.3                     
 3rd Qu.: 98.00   3rd Qu.:80.0                     
 Max.   :103.00   Max.   :90.0                     
```

Use the str() function to get the names of the columns, class of each column.

```
> str(MH)
'data.frame':   1014 obs. of  7 variables:
 $ Age        : int  25 35 29 30 35 23 23 35 32 42 ...
 $ SystolicBP : int  130 140 90 140 120 140 130 85 120 130 ...
 $ DiastolicBP: int  80 90 70 85 60 80 70 60 90 80 ...
 $ BP         : num  15 13 8 7 6.1 7.01 7.01 11 6.9 18 ...
 $ BodyTemp   : num  98 98 100 98 98 98 98 102 98 98 ...
 $ HeartRate  : int  86 70 80 70 76 70 78 86 70 70 ...
 $ RiskLevel  : chr  "high risk" "high risk" "high risk" "high risk" ...
>
```

Removing outliers

The process of standardizing data so that analysts and others can use, examine, and research it is known as data standardization. The process of placing various variables on the same scale so that scores from various variable types can be compared is known as standardization in statistics. Scale () is used .

```
#Removing outliers
standard.features <- scale(MH[,1:6])
standard.features
```

When the outliers are removed ,the interpretation of the standard features.

Console   Terminal   Background Jobs

R 4.3.2 · C:/Users/Midara/Downloads/maternal+health+risk/

```
> standard.features <- scale(MH[,1:6])
> standard.features
              Age  SystolicBP DiastolicBP          BP    BodyTemp
 [1,] -0.361559706  0.91294582  0.25489700  1.90495017 -0.48497618
 [2,]  0.380589164  1.45630853  0.97505739  1.29769930 -0.48497618
 [3,] -0.064700158 -1.26050504 -0.46526338 -0.22042787  0.97340415
 [4,]  0.009514729  1.45630853  0.61497720 -0.52405331 -0.48497618
 [5,]  0.380589164  0.36958310 -1.18542377 -0.79731620 -0.48497618
 [6,] -0.509989480  1.45630853  0.25489700 -0.52101705 -0.48497618
 [7,] -0.509989480  0.91294582 -0.46526338 -0.52101705 -0.48497618
 [8,]  0.380589164 -1.53218640 -1.18542377  0.69044843  2.43178448
 [9,]  0.157944503  0.36958310  0.97505739 -0.55441585 -0.48497618
[10,]  0.900093373  0.91294582  0.25489700  2.81582647 -0.48497618
[11,] -0.509989480 -1.26050504 -1.18542377 -0.52101705 -0.48497618
[12,] -0.806849028  0.36958310  0.25489700 -0.52405331 -0.48497618
[13,] -0.361559706 -0.17377961  0.90304135 -0.52101705 -0.48497618
[14,] -0.732634141  0.36958310 -0.10518319 -0.52101705  0.97340415
[15,]  1.345382696  0.36958310  0.25489700  0.69044843 -0.48497618
[16,] -1.103708576  0.36958310  0.25489700 -0.52101705 -0.48497618
```

Rename the dataset

Rename the data new dataset.

```
#Renaming the dataset as MH1
MH1<- cbind(standard.features,MH[7])
MH1
head(MH1)
summary(MH1)
```

```
> summary(MH1)
      Age               SystolicBP        DiastolicBP            BP         
 Min.   :-1.4748   Min.   :-2.3472   Min.   :-1.9776   Min.   :-0.8277  
 1st Qu.:-0.8068   1st Qu.:-0.7171   1st Qu.:-0.8253   1st Qu.:-0.5544  
 Median :-0.2873   Median : 0.3696   Median : 0.2549   Median :-0.3722  
 Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000  
 3rd Qu.: 0.6774   3rd Qu.: 0.3696   3rd Qu.: 0.9751   3rd Qu.:-0.2204  
 Max.   : 2.9781   Max.   : 2.5430   Max.   : 1.6952   Max.   : 3.1195  
    BodyTemp         HeartRate         RiskLevel        
 Min.   :-0.485   Min.   :-8.3205   Length:1014       
 1st Qu.:-0.485   1st Qu.:-0.5318   Class :character  
 Median :-0.485   Median : 0.2099   Mode  :character  
 Mean   : 0.000   Mean   : 0.0000                     
 3rd Qu.:-0.485   3rd Qu.: 0.7045                     
 Max.   : 3.161   Max.   : 1.9408                     
> 
```

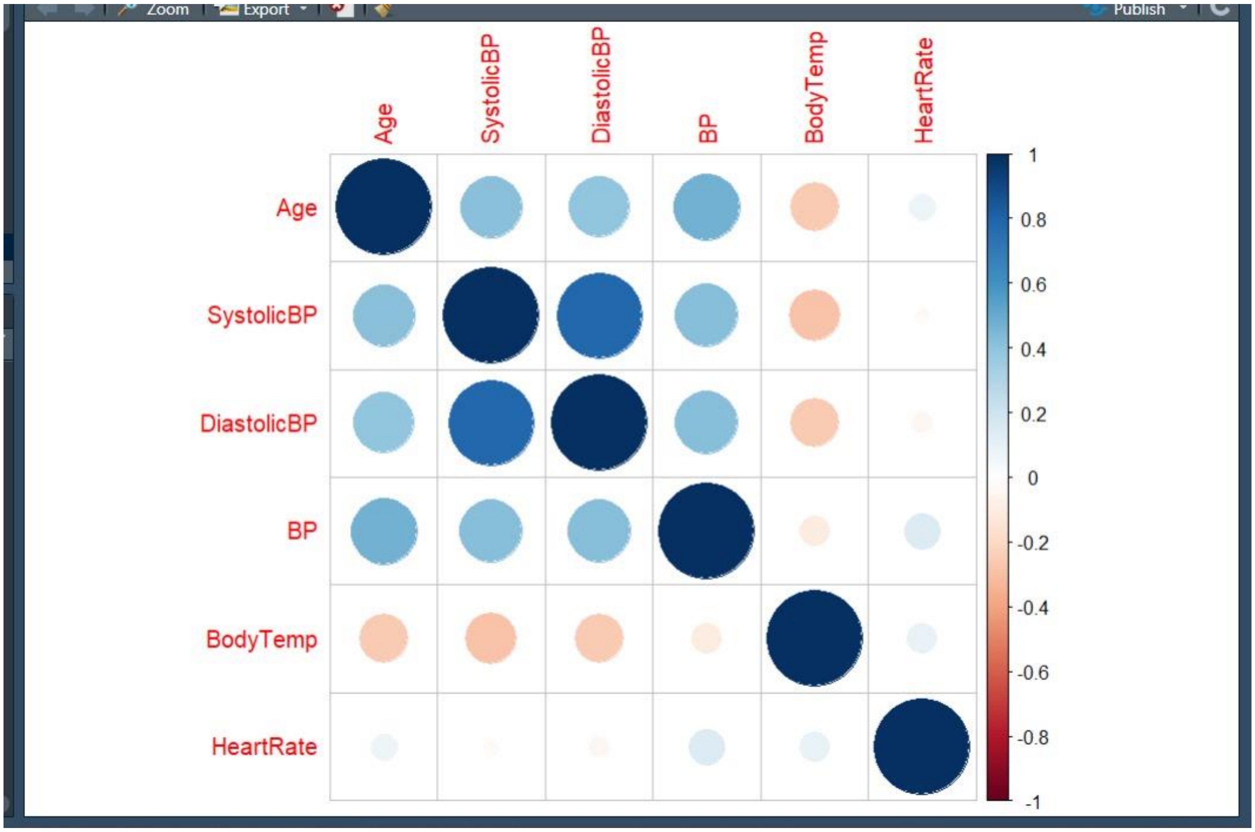
Check if there are any null values to input

```
> anyNA(MH1)
[1] FALSE
> 
```

Create the corrplot

Explains the relationship between the attributes

```
#Creating the corrplot
#load corrplot function
corrplot(cor(MH1[,-7]))
```



<u>Split the data</u>

as train and test data using sample.split function with a splitRatio=0.70

```
#split the data set
set.seed(101)
#Train the sample dataset
#load catool function
sample <- sample.split(MH1$RiskLevel, SplitRatio = 0.70)
train<- subset(MH1 , sample== TRUE )
dim(train)
#Testing the dataset
test <- subset(MH1,sample==FALSE)
dim(test)
```

Then we use dim function to get dimensions

```
> dim(train)
[1] 709   7
> dim(test)
[1] 305   7
>
```

<u>Predict our target variable risk level of the test dataset with k=1.using KNN model</u>.

The knn () function needs to be used to train the model for which structured by installing the package ‘class’. The knn() function identifies the k-nearest neighbours using Euclidean distance where k is a user-specified number.

```
#Improving the model performance
predicted.type <- knn(train[1:6],test[1:6],train$RiskLevel,k=1)
predicted.type
#Error in prediction
error <- mean(predicted.type!=test$RiskLevel)
error
```

```
> error <- mean(predicted.type!=test$RiskLevel)
> error
[1] 0.1180328
>
```

Confusion matrix

```
error
#Confusion matrix
confusionMatrix(predicted.type, as.factor(test$RiskLevel),mode = "everything")
#Testing the alternative k values
```

```
R 4.3.2 · C:/Users/Midara/Downloads/maternal+health+risk/

           Reference
Prediction  high risk low risk mid risk
  high risk        78        5        2
  low risk          1      103        7
  mid risk          3       14       92

Overall Statistics

               Accuracy : 0.8951
                 95% CI : (0.8551, 0.9271)
    No Information Rate : 0.4
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.8413

 Mcnemar's Test P-Value : 0.1577
```

```
Statistics by Class:

                     Class: high risk Class: low risk Class: mid risk
Sensitivity                    0.9512          0.8525          0.8614
Specificity                    0.9776          0.9290          0.9118
Pos Pred Value                 0.9398          0.8889          0.8286
Neg Pred Value                 0.9820          0.9043          0.9300
Prevalence                     0.2689          0.4000          0.3311
Detection Rate                 0.2557          0.3410          0.2852
Detection Prevalence           0.2721          0.3836          0.3443
Balanced Accuracy              0.9644          0.8907          0.8866
>
```

The above results reveal that our model achieved an accuracy of 89.51%. Let’s try different values of k and assess our model.

Test the alternative k values

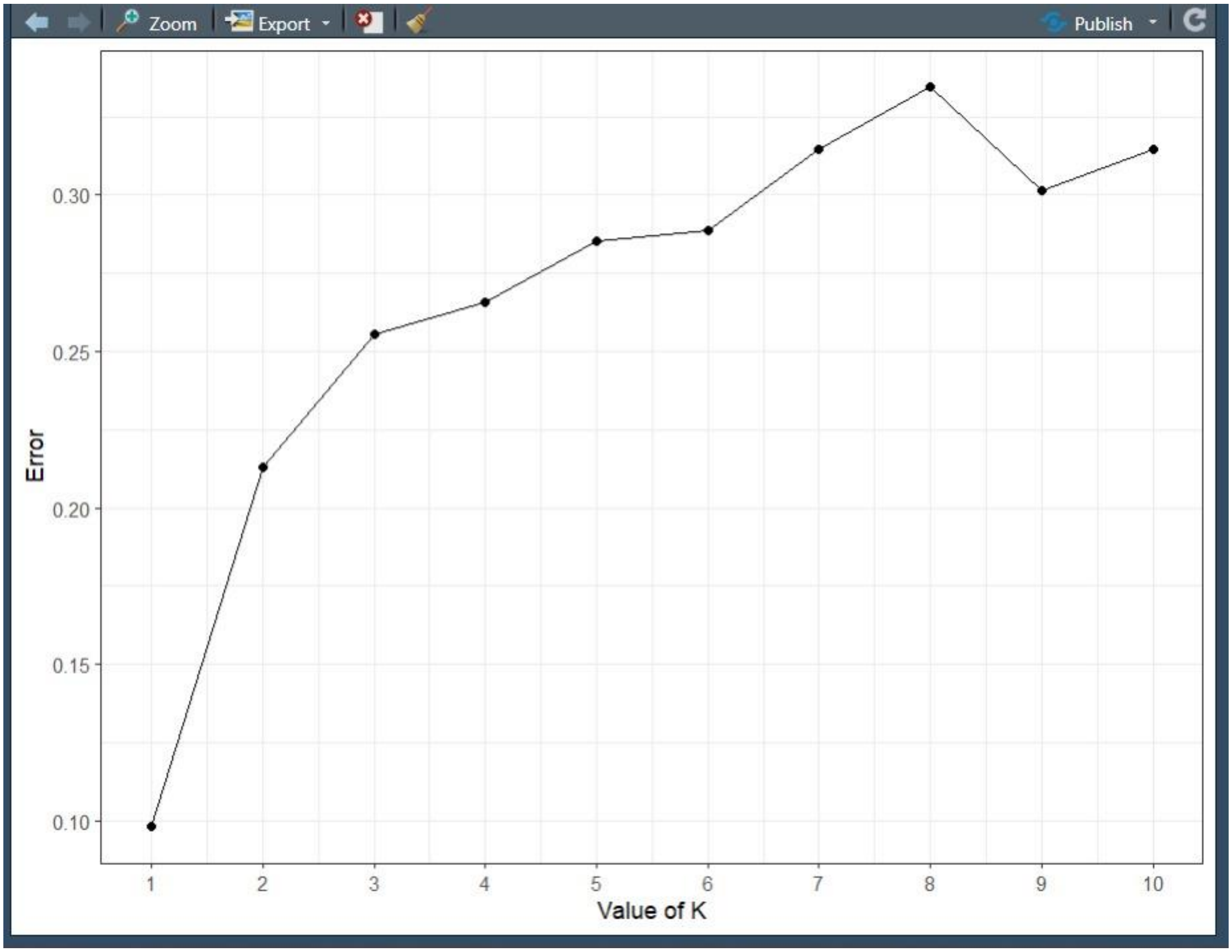
```r
#Testing the alternative k values
predicted.type <- NULL
error.rate <- NULL
for (i in 1:10) {
  predicted.type <- knn(train[1:6],test[1:6],train$RiskLevel,k=i)
  error.rate[i] <- mean(predicted.type!=test$RiskLevel)
}
knn.error <- as.data.frame(cbind(k=1:10,error.type =error.rate))
knn.error
```

```
> knn.error
    k error.type
1   1  0.1180328
2   2  0.2000000
3   3  0.2426230
4   4  0.2557377
5   5  0.2918033
6   6  0.3016393
7   7  0.3278689
8   8  0.3344262
9   9  0.3049180
10 10  0.3180328
>
```

Use the ‘ggplot2’ library to plot the chart.

```r
#Creating the ggplot for the K values
#load ggplot2 function
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')
```

Choose K Value by Visualization. Predict the unique k value for the accuracy of the model.



Improving the model performance

```
#load class function
#Predict our target variable risklevel of the test dataset with k=1.using KNN model
predicted.type <-knn(train[,1:6], test [, 1:6], train$RiskLevel, k=1)
#Error in prediction
error<- mean(predicted.type!=test$RiskLevel)
error
```

```
> predicted.type <- knn(train[1:6],test[1:6],train$RiskLevel,k=1)
> predicted.type
  [1] high risk low risk  low risk  low risk  mid risk  low risk  low risk  low risk
  [9] low risk  low risk  low risk  low risk  low risk  mid risk  mid risk  mid risk
 [17] mid risk  mid risk  mid risk  mid risk  mid risk  low risk  mid risk  mid risk
 [25] mid risk  mid risk  mid risk  mid risk  mid risk  high risk high risk high risk
 [33] high risk high risk high risk high risk high risk high risk high risk high risk
 [41] high risk mid risk  high risk mid risk  high risk mid risk  high risk low risk
 [49] low risk  low risk  high risk mid risk  low risk  high risk high risk low risk
 [57] low risk  low risk  high risk high risk low risk  mid risk  mid risk  mid risk
 [65] low risk  high risk low risk  mid risk  mid risk  high risk low risk  high risk
 [73] low risk  high risk high risk high risk high risk high risk low risk  high risk
 [81] high risk mid risk  low risk  low risk  mid risk  mid risk  high risk low risk
 [89] high risk mid risk  low risk  low risk  low risk  high risk low risk  low risk
 [97] mid risk  low risk  low risk  low risk  low risk  low risk  low risk  mid risk
[105] high risk low risk  low risk  high risk low risk  low risk  mid risk  mid risk
[113] mid risk  low risk  low risk  low risk  mid risk  low risk  high risk mid risk
[121] low risk  mid risk  high risk high risk low risk  high risk high risk mid risk
[129] low risk  low risk  mid risk  mid risk  low risk  low risk  mid risk  mid risk
```

Error in prediction

```
> error <- mean(predicted.type!=test$RiskLevel)
> error
[1] 0.1180328
>
```

Confusion matrix

```
error
#Confusion Matrix
confusionMatrix(predicted.type,as.factor(test$RiskLevel))
```

```
            Reference
Prediction  high risk low risk mid risk
  high risk        78        3        2
  low risk          1      104       12
  mid risk          3       15       87

Overall Statistics

               Accuracy : 0.882
                 95% CI : (0.8404, 0.9159)
    No Information Rate : 0.4
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.821

 Mcnemar's Test P-Value : 0.6746

Statistics by Class:
```

Improving the model performance

```
Statistics by Class:

                     Class: high risk Class: low risk Class: mid risk
Sensitivity                    0.9512          0.8443          0.9109
Specificity                    0.9686          0.9563          0.9167
Pos Pred Value                 0.9176          0.9279          0.8440
Neg Pred Value                 0.9818          0.9021          0.9541
Precision                      0.9176          0.9279          0.8440
Recall                         0.9512          0.8443          0.9109
F1                             0.9341          0.8841          0.8762
Prevalence                     0.2689          0.4000          0.3311
Detection Rate                 0.2557          0.3377          0.3016
Detection Prevalence           0.2787          0.3639          0.3574
Balanced Accuracy              0.9599          0.9003          0.9138
>
```

These statistics shows 80% accuracy when k =1,this means we can implement this model for future data models to analyze the maternal health risk level of pregnant mothers by internet of things.

6.Result analysis and Discussion

When analyzing the above results, we have confirmed that our classification of data mining results was accurate. The result of our model was evaluated by using KNN model. The confusion matrix shows the incidence of the model as follows. The maternal health of the pregnant mothers’ can be predicted as,

- Low risk
- Mid risk
- High risk

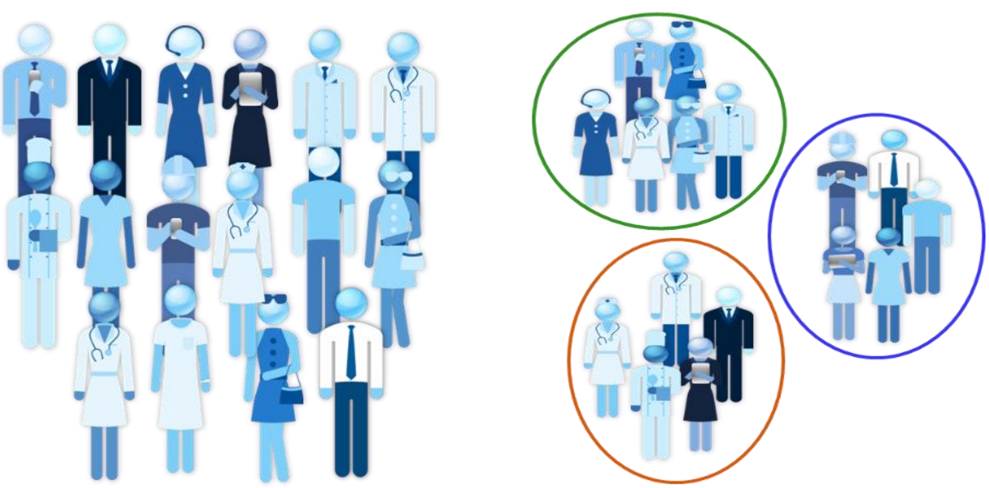
Machine learning models can be developed further to categorize maternal health risks groups. Healthcare practitioners can identify and intervene on behalf of pregnant women who may be at risk of difficulties during pregnancy or childbirth earlier by using this classification model.

7.Conclusion

The KNN algorithm's study of the UCI maternal health risk dataset, in conclusion, emphasizes the value of machine learning methods for recognizing and comprehending intricate health patterns. We were also able to identify relationships between a number of variables and their effects on maternal health outcomes, including age, location, nutritional intake, socioeconomic status, and access to prenatal care. These relationships were obtained by applying KNN. Using KNN's capacity to group data points according to how close they are to other similar occurrences. we were able to understand the complex interactions between these factors. To effectively address maternal health hazards, the results highlight the significance of focused treatments guided by data-driven approaches. In the future, developing customized therapies and more accurate risk assessment will be made possible by using machine learning techniques like KNN into public health research. This will ultimately lead to better maternal health outcomes globally.

02.Clustering

Cluster Analysis of various types of points of interests in South India



1.Introduction

Understanding customer preferences and user interests is essential for creating individualized experiences and successful marketing of the places in the travel and tourism industry. Travelers from all over the world looking for uncommon and fascinating experiences are drawn to South India because of it’s rich in cultural legacy, varied landscapes, and lively customs.

We examine user reviews from holidayiq.com, a well-known website for travelers to exchange experiences and ideas, in our clustering analysis. The reviews in the dataset were collected by 249 travel enthusiasts up to October 2014, with a concentration on South Indian destinations. The information specifically groups reviews into six categories that reflect the wide range of interests and experiences that visitors have while visiting the area.

The collection of data highlights into the diverse range of travel experiences available in South India, including historical sites, immaculate beaches, verdant hill stations, wildlife reserves, cultural celebrations, and wellness centers. To reveal underlying trends and preferences and highlight the most intriguing features of South India's area of interest, we count the number of reviews in each category for each reviewer.
We used clustering algorithms to identify different groups according to interests and preferences. We aim to explore hidden insights that can shape destination marketing aspects, improve tourist experiences, and promote sustainable tourism practices by delving into the complexities of travel reasons through the analysis of user-generated material.
This clustering analysis aims to provide useful insights for travel and tourism industry stakeholders by revealing the highlights of user interest in South India, fostering a deeper understanding of traveler behavior and preferences in one of India's most captivating regions.

2.Dataset

This dataset was derived from the famous data repository called UCI: UCI Machine Learning Repository

249 destination reviews of user interest data regarding the different kinds of South Indian points of interest up to October 2014 from travelers that have posted in holidayiq.com were contributed to this the dataset.
250 rows and 7 columns are consisted in this data set.

The link to further references about this data set is provided below.
Dataset: BuddyMove Data Set - UCI Machine Learning Repository

3.Explanation and preparation

1. User- Unique user id
2. Sports- Number of reviews on stadiums, sports complex, etc.
3. Religious- Number of reviews on religious institutions Number of reviews on religious institutions
4. Nature - Number of reviews on beach, lake, river, etc.
5. Theatre- Number of reviews on theatres, exhibitions, etc.
6. Shopping- Number of reviews on malls, shopping places, etc.
7. Picnic- Number of reviews on parks, picnic spots, etc.

4.Implementation of R

set the working directory.

R studio->Session-> Set working directory

In the **Choose working Directory** dialogue, navigate and select the folder where you saved your data file.

Import the data set

```
#Import the data set
Holiday_Interests<-read.csv("buddymove_holidayiq.csv",header=TRUE,
          col.names=c("User","Sports","Religious","Nature",
                      "Theatre","Shopping","Picnic"))
View(Holiday_Interests)
```

Data processing

Since the data set contains a greater number of rows to improve the clear visualization of the clustering, a subset of the dataset containing a substantial number of entries has been extracted, consisting of the top 100 rows.

The new dataset with 100 rows is as follows.

The following code is used to explore the structure of the data set

```
#filter first 100 rows for clustering
Holiday_Interests_1<- Holiday_Interests[1:100, ]
str(Holiday_Interests_1)
```

```
> str(Holiday_Interests_1)
'data.frame':   100 obs. of  7 variables:
 $ User     : chr  "User 1" "User 2" "User 3" "User 4" ...
 $ Sports   : int  2 2 2 2 2 3 3 3 3 3 ...
 $ Religious: int  77 62 50 68 98 52 64 54 64 86 ...
 $ Nature   : int  79 76 97 77 54 109 85 107 108 76 ...
 $ Theatre  : int  69 76 87 95 59 93 82 92 64 74 ...
 $ Shopping : int  68 69 50 76 95 52 73 54 54 74 ...
 $ Picnic   : int  95 68 75 61 86 76 69 76 93 103 ...
```

Explore the data set.

Identify column names, first data rows, last data rows, summary of data.

```r
#inspect the data set
names(Holiday_Interests_1)
head(Holiday_Interests_1)
tail(Holiday_Interests_1)
summary(Holiday_Interests_1)
```

```
R 4.3.2 · C:/Users/Midara/Downloads/buddymove+data+set/
> names(Holiday_Interests_1)
[1] "User"      "Sports"    "Religious" "Nature"    "Theatre"   "Shopping"  "Picnic"
> head(Holiday_Interests_1)
    User Sports Religious Nature Theatre Shopping Picnic
1 User 1      2        77     79      69       68     95
2 User 2      2        62     76      76       69     68
3 User 3      2        50     97      87       50     75
4 User 4      2        68     77      95       76     61
5 User 5      2        98     54      59       95     86
6 User 6      3        52    109      93       52     76
```

```
> tail(Holiday_Interests_1)
        User Sports Religious Nature Theatre Shopping Picnic
95   User 95      6        84    108     133       98     81
96   User 96      6        99     84     138       69     83
97   User 97      8       113     94      84      109    128
98   User 98      6       148     64      74      138    128
99   User 99      8        84    138     113       64    114
100 User 100      6        98     79     138       79     79
```

```
> summary(Holiday_Interests_1)
     User               Sports         Religious          Nature          Theatre
 Length:100         Min.   : 2.00   Min.   : 50.00   Min.   : 52.00   Min.   : 59.00
 Class :character   1st Qu.: 4.00   1st Qu.: 74.00   1st Qu.: 76.00   1st Qu.: 76.00
 Mode  :character   Median : 5.00   Median : 87.50   Median : 94.00   Median : 89.00
                    Mean   : 5.47   Mean   : 89.62   Mean   : 96.96   Mean   : 93.93
                    3rd Qu.: 6.50   3rd Qu.:101.50   3rd Qu.:115.75   3rd Qu.:110.00
                    Max.   :14.00   Max.   :148.00   Max.   :155.00   Max.   :148.00
    Shopping          Picnic
 Min.   : 50.00   Min.   : 61.00
 1st Qu.: 67.50   1st Qu.: 79.00
 Median : 80.50   Median : 90.50
 Mean   : 88.18   Mean   : 93.87
 3rd Qu.:103.00   3rd Qu.:102.25
 Max.   :208.00   Max.   :143.00
>
```

check the dimensions and number of points in Holiday_Interests_1

Number of rows and columns, dimensions can be identified.

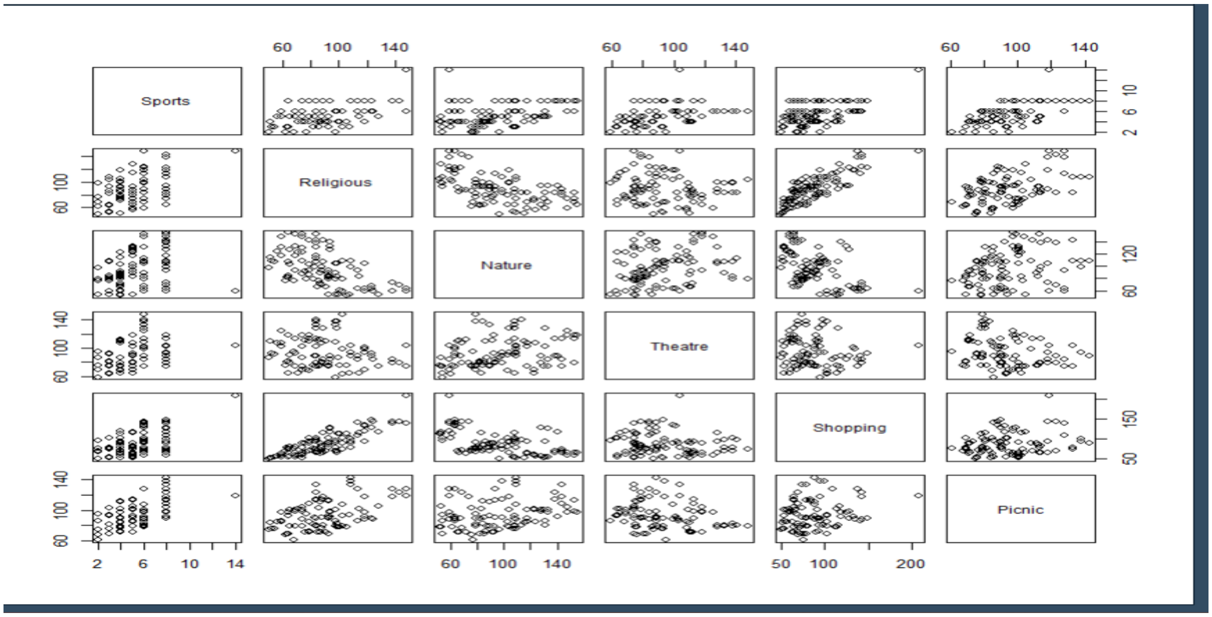
```
nrow(Holiday_Interests_1)
ncol(Holiday_Interests_1)
dim(Holiday_Interests_1)
str(Holiday_Interests_1)
```

```
> nrow(Holiday_Interests_1)
[1] 100
> ncol(Holiday_Interests_1)
[1] 7
> dim(Holiday_Interests_1)
[1] 100   7
```

Pair the columns.

Install the required packages and remove character column to contrast the variables. Deleting the first column user in coded in below.

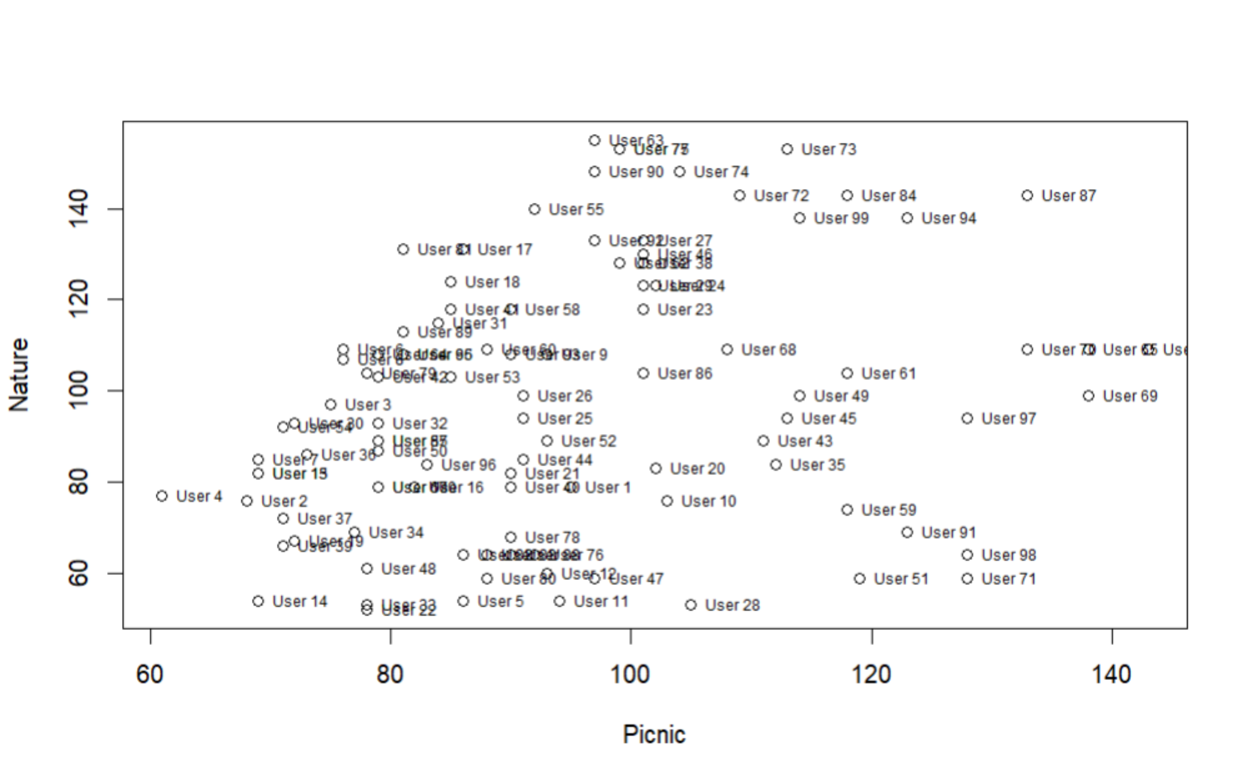
```
#Install and activate cluster package
install.packages("cluster")
library(cluster)
#create scatterplot matrix to compare the variables
x=Holiday_Interests_1[, -c(1)]
View(x)
str(x)
pairs(x)
```



The plot of Picnic vs Nature variable as follows.

Using the following line of code to plot can understand the relationship between the interests of users about Nature and Picnics can be understood.

```
#create the polt and understand the relationship between nature and picnics
plot(Nature~Picnic, data =Holiday_Interests_1 )
#label the data points with User_ID variables
with(Holiday_Interests_1,text(Nature~Picnic,labels=User))
#Remove the overlap of user names
plot(Nature~Picnic, data =Holiday_Interests_1)
with(Holiday_Interests_1,text(Nature~Picnic,labels=User,pos=4,cex=.6))
```



The relationships between several attributes can be displayed using a scatterplot matrix. The matrix can display relationships between variables after two-way combinations of the user interests (nature & picnic) are shown to illustrate which groupings are probably going to be important.

5.Data Mining.

The practice of normalizing the data in an unstructured database is called data normalization. Data redundancy can be reduced, and overall data integrity can be enhanced by doing this. The data set consists of different kinds of values. By normalizing a stable format of data
Can interpret.

Normalization

```
with(Holiday_Interests_1,text(Nature~Picnic,labels=User,pos=4,cex=.6))
#normalise function
str(Holiday_Interests_1)
head(Holiday_Interests_1)
normalize <- function(df) {
  return(((df - min(df, na.rm = TRUE)) / (max(df, na.rm = TRUE) - min(df, na.rm = TRUE)) * (1 - 0)) + 0)
}
interests <- Holiday_Interests_1[, 2:7]
View(interests)
Interests_n <- as.data.frame(lapply(interests, normalize))
View(Interests_n)
head(Interests_n)
```

```
> normalize <- function(df) {
+   return(((df - min(df, na.rm = TRUE)) / (max(df, na.rm = TRUE) - min(df, na.rm = T
RUE)) * (1 - 0)) + 0)
+ }
> interests <- Holiday_Interests_1[, 2:7]
> View(interests)
> Interests_n <- as.data.frame(lapply(interests, normalize))
> View(Interests_n)
> head(Interests_n)
      Sports  Religious     Nature   Theatre   Shopping     Picnic
1 0.00000000 0.27551020 0.26213592 0.1123596 0.11392405 0.41463415
2 0.00000000 0.12244898 0.23300971 0.1910112 0.12025316 0.08536585
3 0.00000000 0.00000000 0.43689320 0.3146067 0.00000000 0.17073171
4 0.00000000 0.18367347 0.24271845 0.4044944 0.16455696 0.00000000
5 0.00000000 0.48979592 0.01941748 0.0000000 0.28481013 0.30487805
6 0.08333333 0.02040816 0.55339806 0.3820225 0.01265823 0.18292683
> 
```

Euclidean Distance

After that, Euclidean distance is used to verify the length. Frequently, the default approach for clustering is the Euclidean distance method.

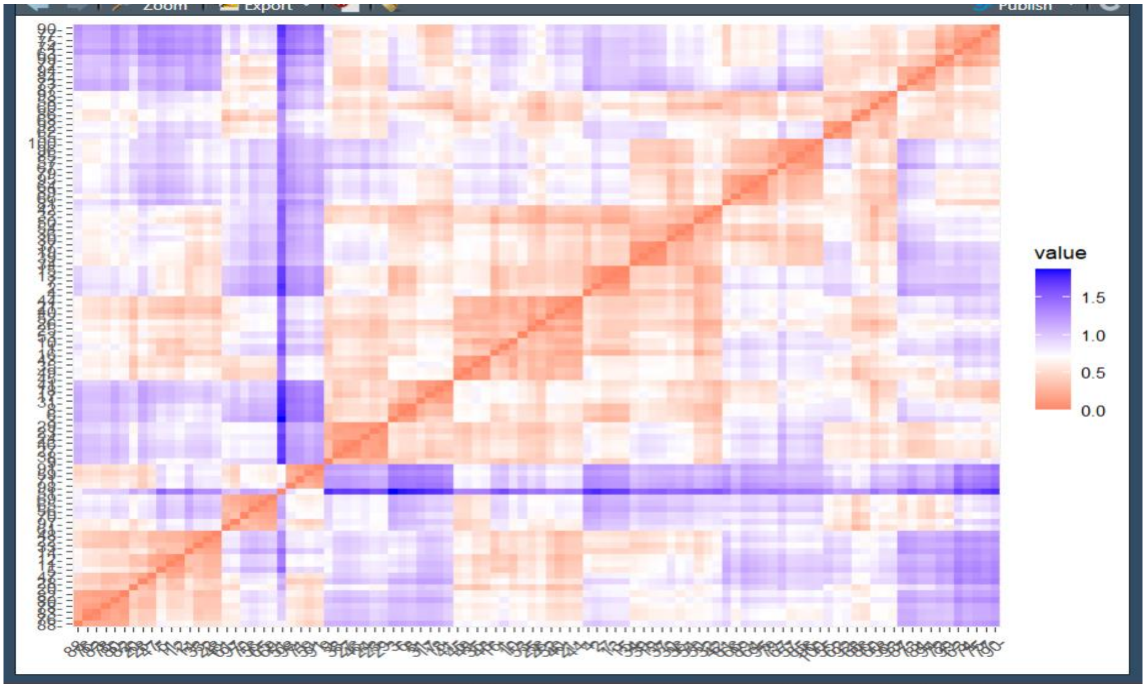
```r
install.packages("factoextra") # install "factoextra" package
library(factoextra) # activate "factoextra" package
#choose distance method and create distance matrix
distance<- dist(Interests_n,method = "euclidean",)
max(distance)
min(distance)
print(distance)
print(distance, digits = 3)
fviz_dist(distance)
```

```
> max(distance)
[1] 1.860247
> min(distance)
[1] 0.02430821
```

```
 [ reached getOption("max.print") -- omitted 89 rows ]
> print(distance, digits = 3)
        1      2      3      4      5      6      7      8      9     10     11     12     13     14
2   0.3727
3   0.4688 0.3059
4   0.5183 0.2422 0.3684
5   0.3984 0.5407 0.7825 0.6439
6   0.5418 0.4214 0.1606 0.4326 0.8645
7   0.3883 0.1426 0.2658 0.2141 0.5891 0.3309
8   0.5141 0.3938 0.1483 0.4076 0.8325 0.0329 0.3007
9   0.3397 0.4733 0.3929 0.6263 0.6928 0.4055 0.4369 0.3905
10  0.1742 0.5006 0.5869 0.5991 0.3957 0.6296 0.4879 0.6021 0.4364
11  0.4604 0.6546 0.8675 0.7264 0.1750 0.9266 0.6731 0.8948 0.7508 0.3872
       15     16     17     18     19     20     21     22     23     24     25     26     27     28
```

Visualization of the distance

The color level is proportional to the value of the dissimilarity between observations: Red: high similarity / Blue: low similarity.



Hopkins statistics

The evaluation of the tendency for clustering. It establishes whether the dataset can produce meaningful clusters. The Hopkins statistics can be used to assess how accurate a class cluster is. An important cluster is one where the Hopkins statistics are near to one.

```
#Get cluster tendency
tendency <- get_clust_tendency(Interests_n, 30, graph = TRUE)
tendency$hopkins_stat
# plot the Within Cluster Sum of Squares and the
```

```
> tendency <- get_clust_tendency(Interests_n, 30, graph = TRUE)
> tendency$hopkins_stat
[1] 0.8123846
```

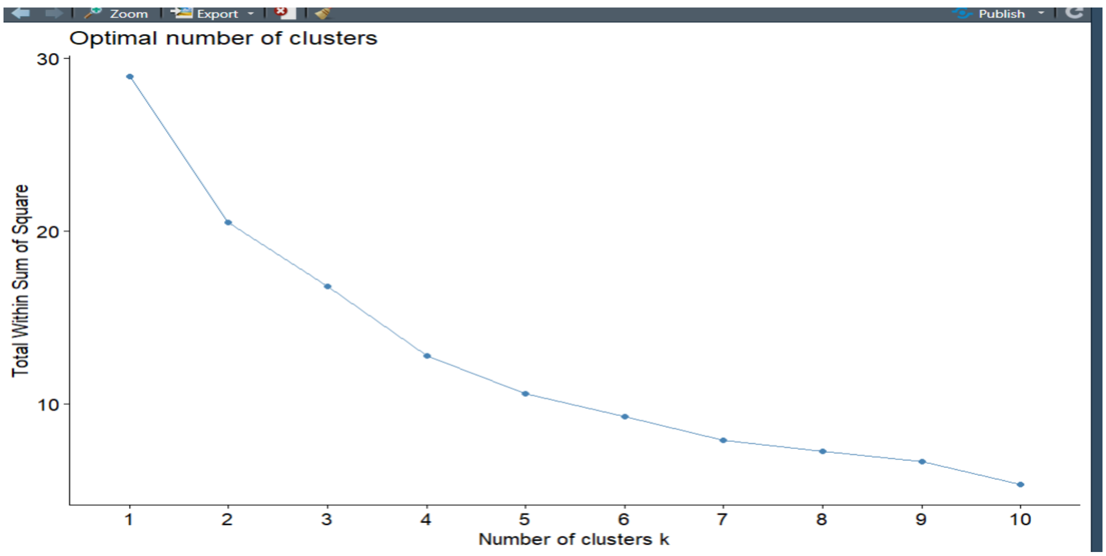
A value that is almost one has been attained by our model. It predicts that the dataset can visualize meaningful clusters.

Determine the optimal number of clusters.

The Elbow Method

```
# plot the Within Cluster Sum of Squares and the
#number of clusters to find the location of a bend or a knee in the plot
fviz_nbclust(Interests_n, kmeans, method = "wss")
#Perform k-means clustering on a Interests_n data set with k=3
```

The elbow-type location of this plot is revealing that number 2 is the optimal value for k.so 2 for k value should be interpreted to cluster the data set.

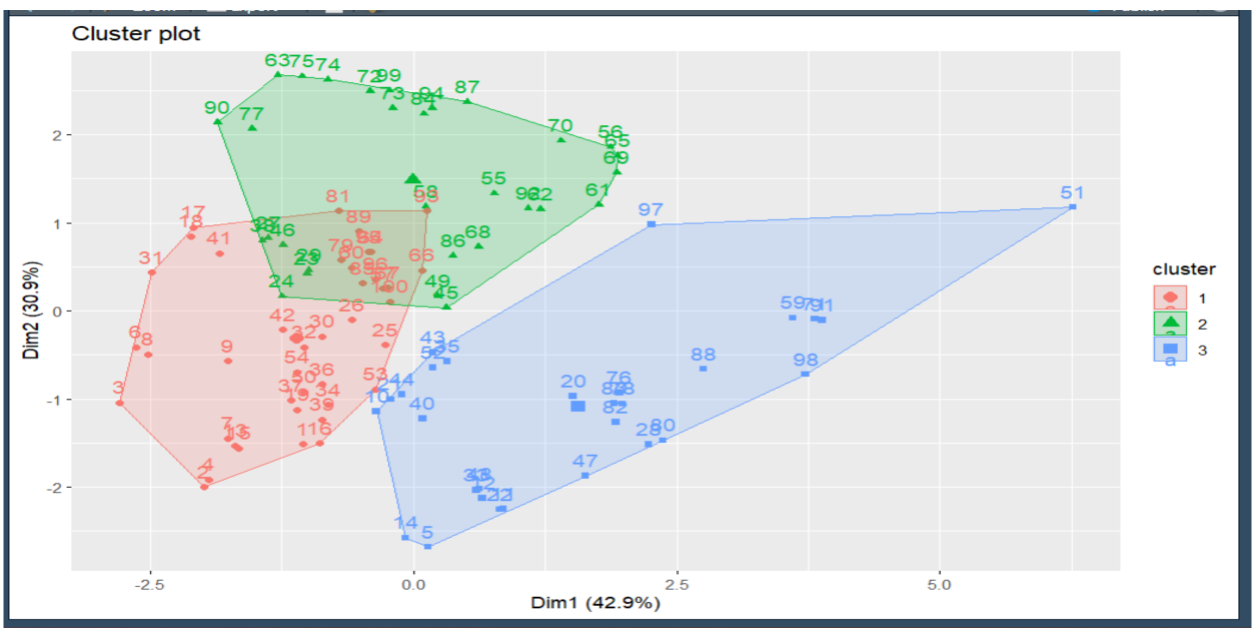


K Means clustering

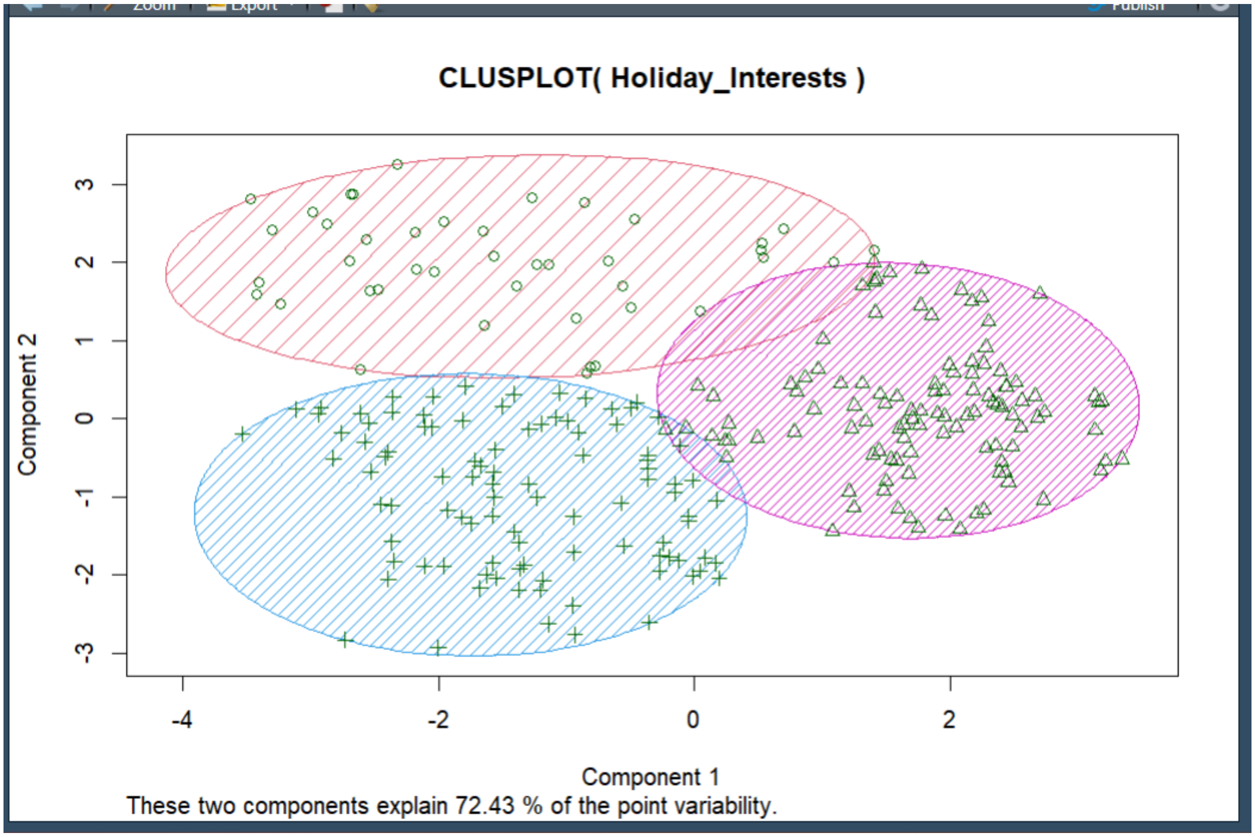
After determining the best value for cluster classes, we created the fit model and visualized the cluster plot.

When we have unlabeled data, or data without clearly defined categories or groupings, we can use unsupervised learning techniques like K-means clustering. This algorithm's objective is to identify groups within the data; the variable K indicates how many groups there are. Using the given features, the algorithm iteratively assigns each data point to one of the K groups. Based on feature similarity, data points are grouped.
Any kind of grouping can be achieved with the K-Means clustering technique.

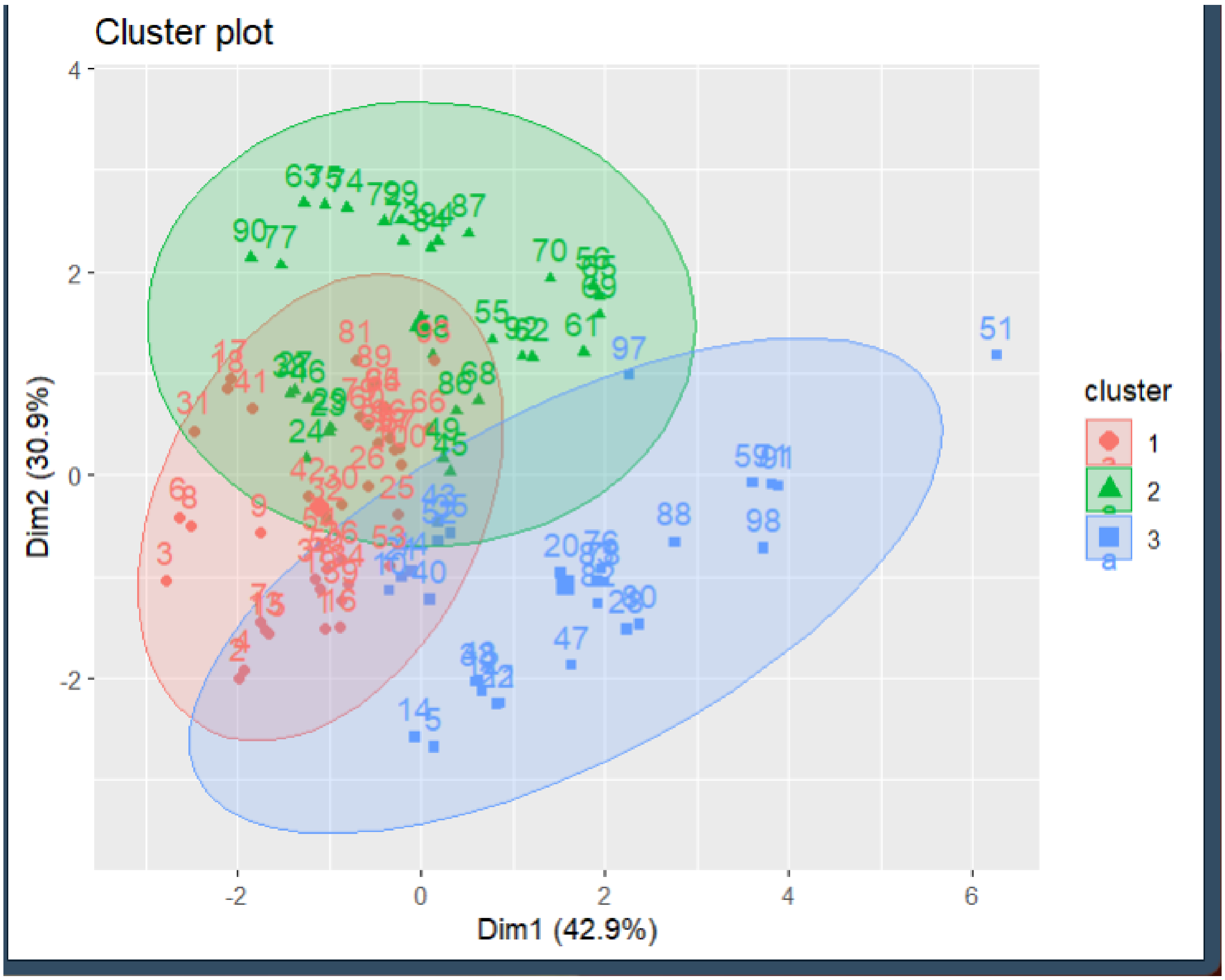
```r
set.seed(123)
km.fit <- kmeans(Interests_n, 3, nstart = 30)
km.fit$cluster
km.fit$size
# Visualize clusters using fviz_cluster() function in facto extra package
fviz_cluster(km.fit,Interests_n)
```



```r
#kmeans function perform k-means clustering on a data matrix.
kc<-kmeans(Holiday_Interests[,-1],3) #k=3
kc
#clusplot() function
clusplot(Holiday_Interests, kc$cluster, color=TRUE, shade=TRUE,  lines=0)
```




```r
fviz_cluster(km.fit,Interests_n,ellipse.type = "norm")
```


Cluster plot
Dim2 (30.9%)
Dim1 (42.9%)
cluster
1
2
3

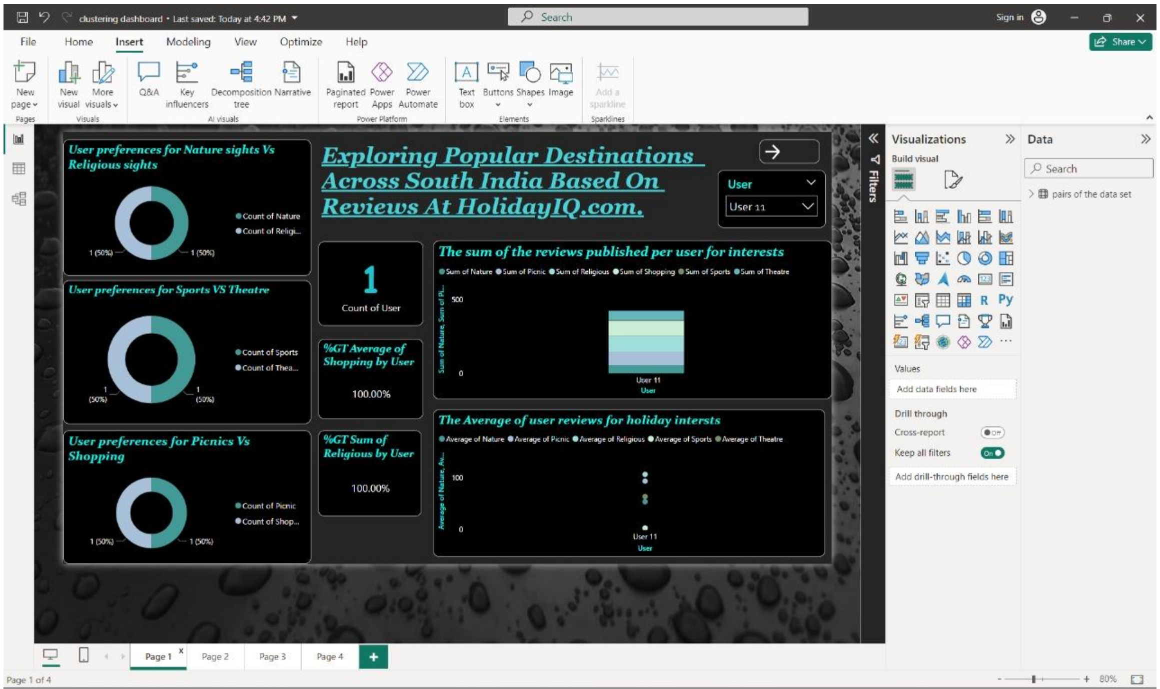
6.Result Analysis and Discussion.

Our goal of our assignment is to divide the dataset into discrete groups according to similarity.
the underlying patterns and groups within the dataset is possible to identify.
Euclidean distance using distance ()
The distance between the rows. The distance between the users can identified.
Visualization of distance matrix shows the dissimilarities between the holiday interests of users.
The cluster plot observations. The k means algorithm was used to identify the homogenous subgroups in holiday interests reviews in holidayiq.com using the Hopkins statistics cluster tendency is accessed to identify whether the dataset is clusterble or not. Since the cluster tendency is 0.8 which is close to 1. From this data set a clear visualization of clustering can be interpreted. fviz_nbclust () function is used to predict the optimal number of clusters using the Elbow method. The elbow line falls around 3.the k means algorithm fin the ideal groups for the unlabeled data the user interests' reviews can be categorized under 3 different clusters.
The clusters provide a detailed breakdown of the user interests reviews on 6 main holiday interests in south India. The clusters consist of these number of users:
Cluster 1:41 users, Cluster2:30 users, Cluster3:29 users.
People who are highly engaged are classified as High-Interest Users (Cluster 1).
Moderate-Interest Users (cluster 2): are individuals who participate in moderate levels of activity.
User interaction is low among: "Low-Interest Users" (cluster 3). cluster 01 is a triangle clustered data points spread as a triangle. Scattered Cluster (Cluster 2): Points spread out. Compact Cluster" (Cluster 3): Points densely packed.
Consequently, coherent clusters of users with similar interests and activity patterns can be identified. Through the visualization of the cluster assignments and centroids, we were able to obtain a more comprehensive understanding of user segmentation and preferences by gaining insights into the distribution and features of the user reviews. Businesses and researchers seeking to improve user experiences across a range of areas, personalize suggestions, and optimize marketing strategies may find important insights in this K-means cluster analysis of this Holiday Interests dataset.

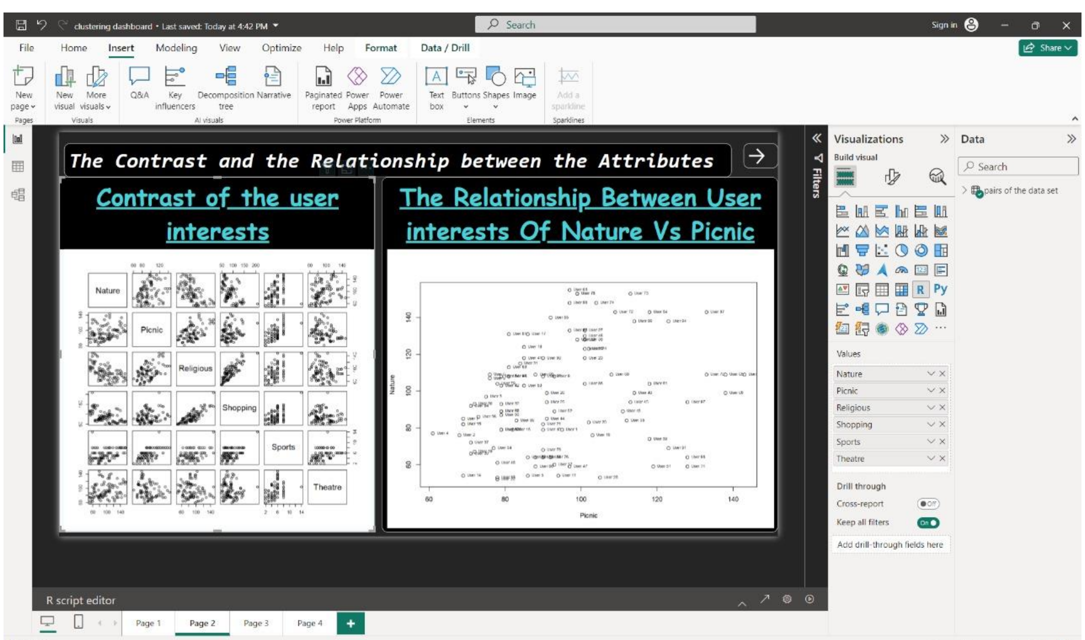
7.Conclusion

Ultimately, the Buddy Move dataset from UCI has yielded insightful information about user preferences and behaviors across a range of categories based on user reviews on holidayiq.com. With the K-means cluster analysis of the dataset. Remarkable patterns and similarities between user reviews of religious, shopping, picnic and entertainment establishments were found through the identification of unique clusters by this study. By grouping users into cohesive groups, the K means clustering technique made it easier for academics and businesses to customize advice, develop marketing campaigns, and improve user experiences. It has become possible to target interventions and enhance service delivery by comprehending the underlying structure of user interactions in the dataset. To get deeper insights and streamline decision-making processes across a variety of industries, including social networking and e-commerce, further investigation and improvement of clustering techniques like K means would be necessary in the future.

03.Power BI Dashboard



The contrast and the relationship between the attributes



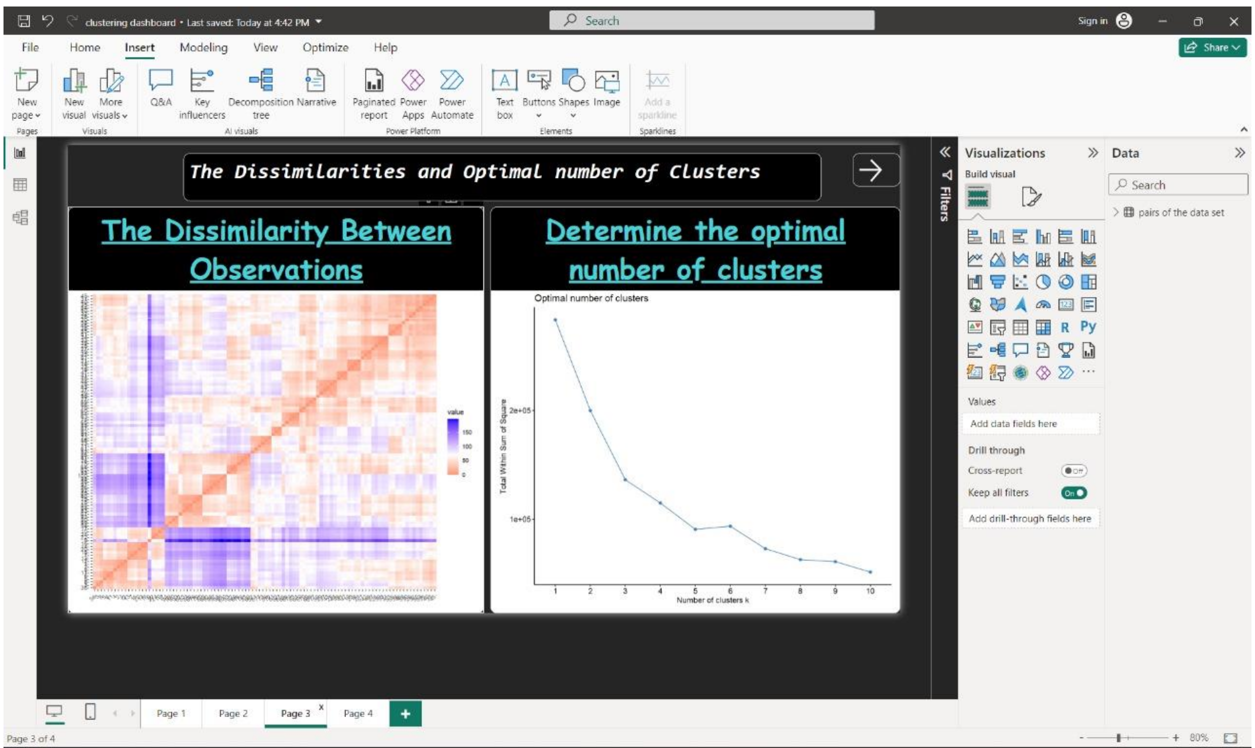
⚠ Duplicate rows will be removed from the data.

```
# The following code to create a dataframe and remove duplicated rows is always executed and acts as a preamble for your script:

# dataset <- data.frame(Theatre, Shopping, Sports, Religious, Picnic, Nature)
# dataset <- unique(dataset)

# Paste or type your script code here:
plot(Nature~Picnic,data=dataset)
with(dataset,text(Nature~Picnic,labels=User ,pos=4,cex=0.6))
```

The dissimilarities and optimal number of clusters



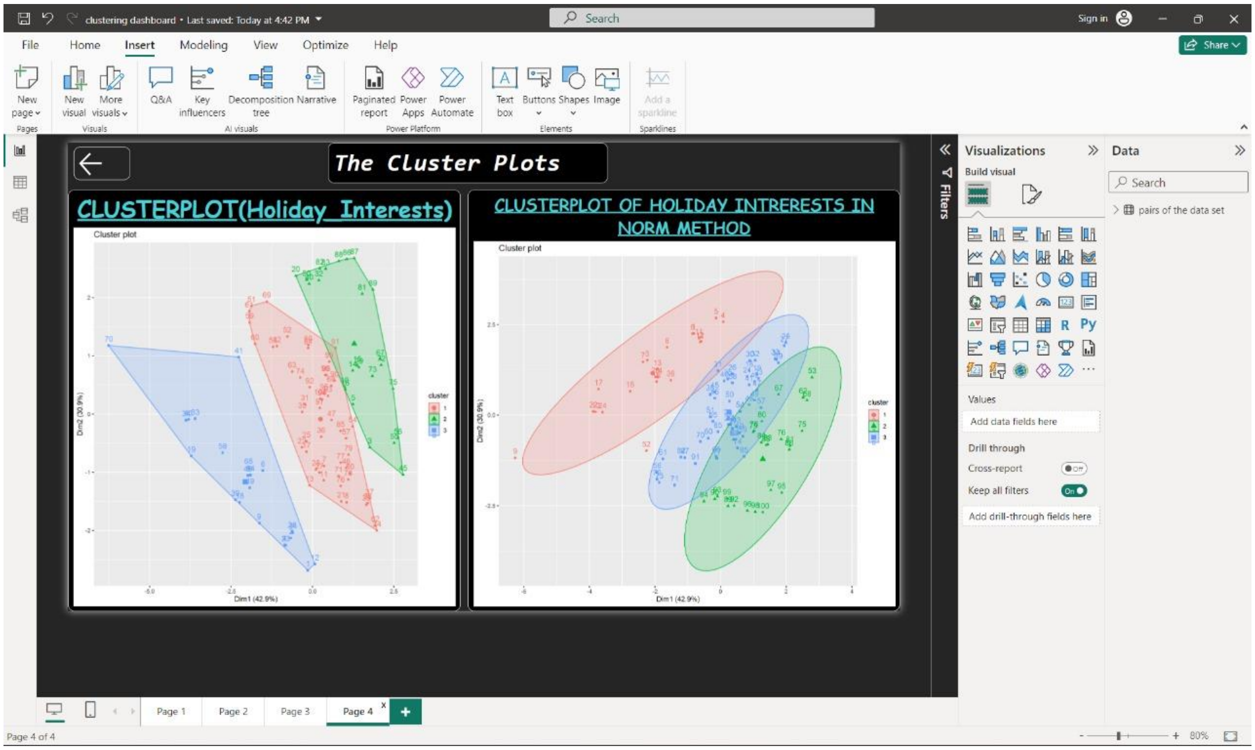
⚠ Duplicate rows will be removed from the data.

```
# The following code to create a dataframe and remove duplicated rows is always executed and acts as a preamble for your script:

# dataset <- data.frame(Nature, Picnic, Religious, Shopping, Sports, Theatre)
# dataset <- unique(dataset)

# Paste or type your script code here:
distance<-dist(dataset,method="euclidean")
library(factoextra)
fviz_dist(distance)
```

The cluster plots



⚠ Duplicate rows will be removed from the data.

```
# The following code to create a dataframe and remove duplicated rows is always executed and acts as a preamble for your script:

# dataset <- data.frame(Theatre, Sports, Shopping, Religious, Picnic, Nature)
# dataset <- unique(dataset)

# Paste or type your script code here:
library(factoextra)
set.seed(123)
km.fit <- kmeans(dataset,3,nstart=30)
km.fit$cluster
km.fit$size
fviz_cluster(km.fit,dataset)
```

Conclusion

We took the same dataset for power BI dashboard implementation that we used in Clustering. But for visualization purposes, we had to add more additional datasets, which adds more insightful details to the report. Then, we linked the dashboard to the R studio and connected the R scripts. It aims to help travelers make informed decisions about their trips by offering reviews, ratings and recommendations. Finally, this report summarizes the methods that we have used to create the dashboard.

04.References

1.Classification

Data set link:

archive.ics.uci.edu. (n.d.). *UCI Machine Learning Repository*. [online] Available at: https://archive.ics.uci.edu/dataset/863/maternal+health+risk.

KNN model:

IBM (2023). *What is the k-nearest neighbors algorithm? | IBM*. [online] www.ibm.com. Available at: https://www.ibm.com/topics/knn.

www.youtube.com. (n.d.). *CROP RECOMMENDATION ASSISTANT USING MACHINE LEARNING (K-NEAREST ALGORITHM)*. [online] Available at: https://www.youtube.com/watch?v=e99LZoii3ng&t=336s [Accessed 3 Mar. 2024].

2.Clustering

Data set link:BuddyMove Data Set - UCI Machine Learning Repository

K means clustering: TowardsMachineLearning. (2021). *K-Means*. [online] Available at: https://towardsmachinelearning.org/k-means/.

Normalization: www.digitalocean.com. (n.d.). *How to Normalize data in R [3 easy methods] | DigitalOcean*. [online] Available at: https://www.digitalocean.com/community/tutorials/normalize-data-in-r.

Euclidean Distance: Liberti, L., Lavor, C., Maculan, N. and Mucherino, A. (2014). Euclidean Distance Geometry and Applications. *SIAM Review*, 56(1), pp.3–69. doi:https://doi.org/10.1137/120875909.