INT234

[Predictive Analytics]

CA-3

SUBMITTED TO : Mr. Sandeep Kumar

NAME:	Vinay Kumar
REGISTRATION NO.:	12104993
ROLL. NO.:	RK-21-BP-A-07
SECTION:	K-21-BP

INTRODUCTION :

This project focuses on analyzing user behavior patterns using machine learning techniques implemented in R. The goal is to categorize and segment users based on their behavior characteristics to gain insights for applications such as personalized marketing and targeted recommendations

---

About the Project :

The project uses a dataset called "user_behavior_dataset.csv" to explore and classify user behaviors. By leveraging both supervised and unsupervised learning methods, we aim to understand user segments and behavior classes within the data. The project examines the performance of different algorithms, assessing their ability to classify users and form meaningful clusters.

Data Used and Algorithms Applied :

The "user_behavior_dataset.csv" contains information on user attributes, including device model, operating system, and demographic details, which form the basis for analysis. Three algorithms are used:

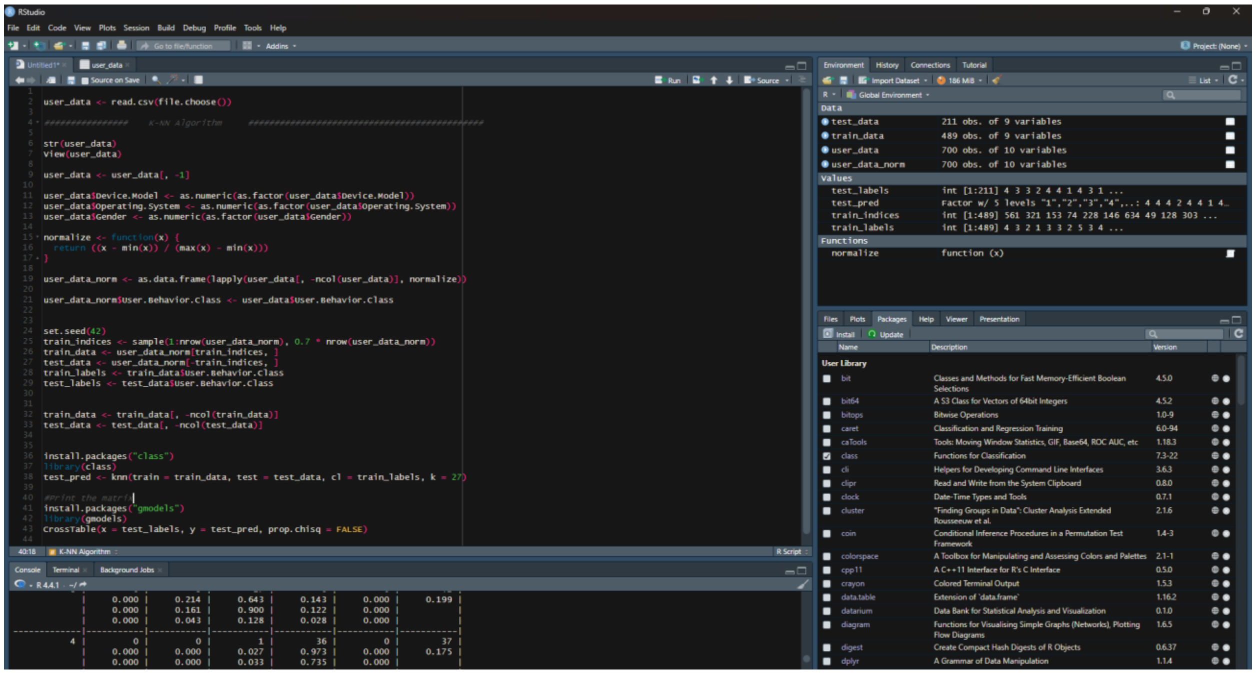
1. **K-Nearest Neighbors (K-NN):** A supervised learning algorithm that classifies users into predefined behavior classes by comparing each user to its nearest neighbors in the dataset.
2. **K-Means Clustering:** An unsupervised learning algorithm that groups users into clusters based on similarity, without any pre-labeled categories, to discover natural segments.
3. **Hierarchical Clustering:** Another unsupervised technique that builds a tree-like structure of user groups based on similarity, providing a visual interpretation of user segments at different levels of detail.

Performance Comparison :

The performance of each algorithm is evaluated based on its effectiveness in identifying distinct user behaviors. K-NN is assessed through classification accuracy, measuring how well it predicts user classes. K-Means and Hierarchical Clustering are compared based on cluster cohesion and interpretability, focusing on how consistently each method groups similar users. Through this comparison, we can identify the strengths and best applications of each approach, providing insights into their suitability for behavior analysis.

Algorithms Code's

K-Nearest Neighbors (K-NN):



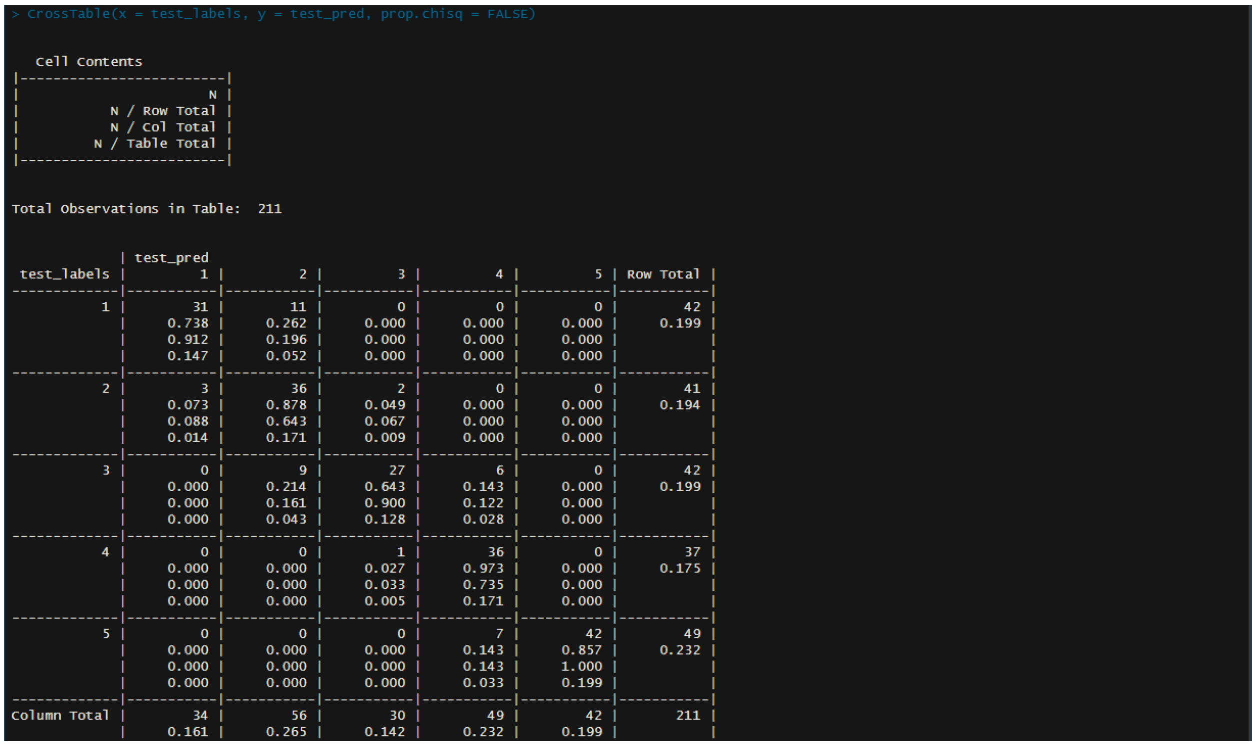
```r

user_data <- read.csv(file.choose())

#################    K-NN Algorithm    ############################################

str(user_data)
View(user_data)

user_data <- user_data[, -1]

user_data$Device.Model <- as.numeric(as.factor(user_data$Device.Model))
user_data$Operating.System <- as.numeric(as.factor(user_data$Operating.System))
user_data$Gender <- as.numeric(as.factor(user_data$Gender))

normalize <- function(x) {
  return ((x - min(x)) / (max(x) - min(x)))
}

user_data_norm <- as.data.frame(lapply(user_data[, -ncol(user_data)], normalize))

user_data_norm$User.Behavior.Class <- user_data$User.Behavior.Class


set.seed(42)
train_indices <- sample(1:nrow(user_data_norm), 0.7 * nrow(user_data_norm))
train_data <- user_data_norm[train_indices, ]
test_data <- user_data_norm[-train_indices, ]
train_labels <- train_data$User.Behavior.Class
test_labels <- test_data$User.Behavior.Class


train_data <- train_data[, -ncol(train_data)]
test_data <- test_data[, -ncol(test_data)]


install.packages("class")
library(class)
test_pred <- knn(train = train_data, test = test_data, cl = train_labels, k = 27)

#Print the matrix
install.packages("gmodels")
library(gmodels)
CrossTable(x = test_labels, y = test_pred, prop.chisq = FALSE)

```

```
             |      0.000 |      0.214 |      0.643 |      0.143 |      0.000 |      0.199 |
             |      0.000 |      0.161 |      0.900 |      0.122 |      0.000 |            |
             |      0.000 |      0.043 |      0.128 |      0.028 |      0.000 |            |
-------------|------------|------------|------------|------------|------------|------------|
           4 |          0 |          0 |          1 |         36 |          0 |         37 |
             |      0.000 |      0.000 |      0.027 |      0.973 |      0.000 |      0.175 |
             |      0.000 |      0.000 |      0.033 |      0.735 |      0.000 |            |
```

```
Data
test_data         211 obs. of 9 variables
train_data        489 obs. of 9 variables
user_data         700 obs. of 10 variables
user_data_norm    700 obs. of 10 variables
Values
test_labels       int [1:211] 4 3 3 2 4 4 1 4 3 1 ...
test_pred         Factor w/ 5 levels "1","2","3","4",..: 4 4 4 2 4 4 1 4…
train_indices     int [1:489] 561 321 153 74 228 146 634 49 128 303 ...
train_labels      int [1:489] 4 3 2 1 3 3 2 5 3 4 ...
Functions
normalize         function (x)
```

	Name	Description	Version
User Library			
	bit	Classes and Methods for Fast Memory-Efficient Boolean Selections	4.5.0
	bit64	A S3 Class for Vectors of 64bit Integers	4.5.2
	bitops	Bitwise Operations	1.0-9
	caret	Classification and Regression Training	6.0-94
	caTools	Tools: Moving Window Statistics, GIF, Base64, ROC AUC, etc	1.18.3
✓	class	Functions for Classification	7.3-22
	cli	Helpers for Developing Command Line Interfaces	3.6.3
	clipr	Read and Write from the System Clipboard	0.8.0
	clock	Date-Time Types and Tools	0.7.1
	cluster	"Finding Groups in Data": Cluster Analysis Extended Rousseeuw et al.	2.1.6
	coin	Conditional Inference Procedures in a Permutation Test Framework	1.4-3
	colorspace	A Toolbox for Manipulating and Assessing Colors and Palettes	2.1-1
	cpp11	A C++11 Interface for R's C Interface	0.5.0
	crayon	Colored Terminal Output	1.5.3
	data.table	Extension of `data.frame`	1.16.2
	datarium	Data Bank for Statistical Analysis and Visualization	0.1.0
	diagram	Functions for Visualising Simple Graphs (Networks), Plotting Flow Diagrams	1.6.5
	digest	Create Compact Hash Digests of R Objects	0.6.37
	dplyr	A Grammar of Data Manipulation	1.1.4

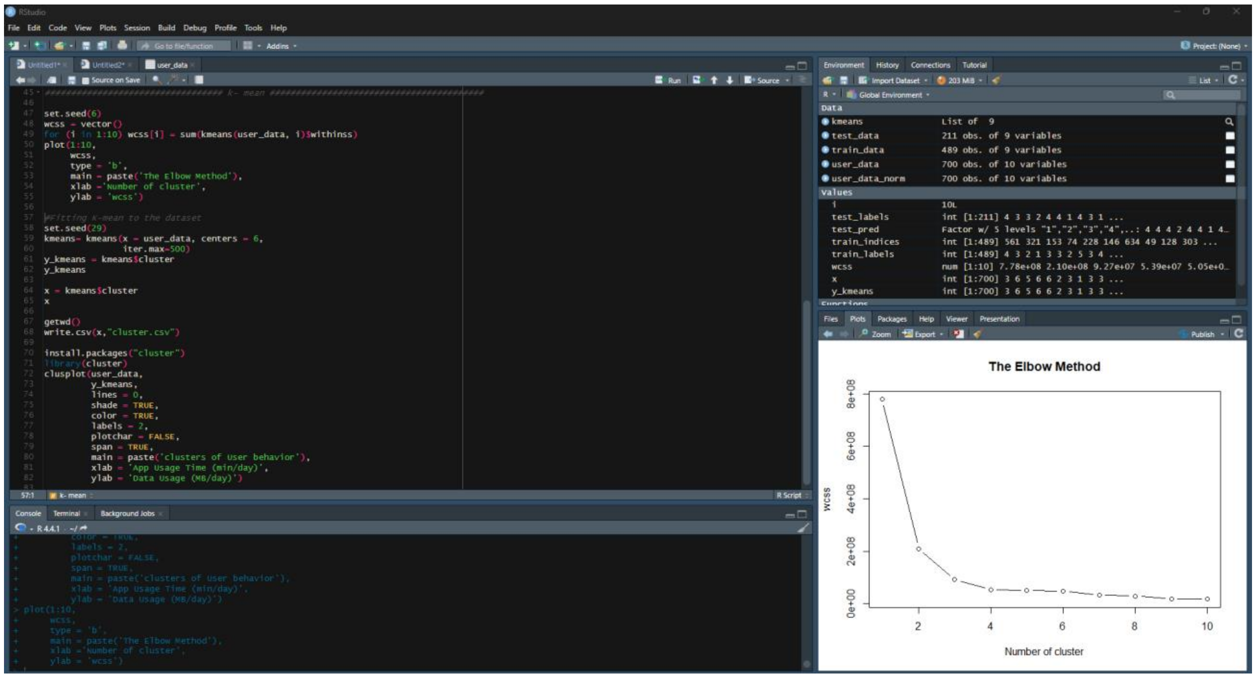
This code performs K-Nearest Neighbors (K-NN) classification on a dataset (user_data) to predict user behavior based on features such as device model, operating system, and gender. The dataset is first loaded and preprocessed by removing any non-informative columns (like an ID column) and converting categorical variables (e.g., Device.Model, Operating.System, and Gender) to numeric form. Then, all features except the target variable (User.Behavior.Class) are normalized to a 0-1 scale to ensure equal weighting in the K-NN algorithm. The data is split into training (70%) and testing (30%) sets, with a seed set for reproducibility. Using the class library's knn function, the model is trained with k=27 and tested on the test set. Finally, the gmodels package is used to generate a confusion matrix, allowing for an assessment of classification accuracy by comparing predicted and actual classes.



```
> CrossTable(x = test_labels, y = test_pred, prop.chisq = FALSE)


   Cell Contents
|-------------------------|
|                       N |
|           N / Row Total |
|           N / Col Total |
|         N / Table Total |
|-------------------------|


Total Observations in Table:  211


             | test_pred
 test_labels |         1 |         2 |         3 |         4 |         5 | Row Total |
-------------|-----------|-----------|-----------|-----------|-----------|-----------|
           1 |        31 |        11 |         0 |         0 |         0 |        42 |
             |     0.738 |     0.262 |     0.000 |     0.000 |     0.000 |     0.199 |
             |     0.912 |     0.196 |     0.000 |     0.000 |     0.000 |           |
             |     0.147 |     0.052 |     0.000 |     0.000 |     0.000 |           |
-------------|-----------|-----------|-----------|-----------|-----------|-----------|
           2 |         3 |        36 |         2 |         0 |         0 |        41 |
             |     0.073 |     0.878 |     0.049 |     0.000 |     0.000 |     0.194 |
             |     0.088 |     0.643 |     0.067 |     0.000 |     0.000 |           |
             |     0.014 |     0.171 |     0.009 |     0.000 |     0.000 |           |
-------------|-----------|-----------|-----------|-----------|-----------|-----------|
           3 |         0 |         9 |        27 |         6 |         0 |        42 |
             |     0.000 |     0.214 |     0.643 |     0.143 |     0.000 |     0.199 |
             |     0.000 |     0.161 |     0.900 |     0.122 |     0.000 |           |
             |     0.000 |     0.043 |     0.128 |     0.028 |     0.000 |           |
-------------|-----------|-----------|-----------|-----------|-----------|-----------|
           4 |         0 |         0 |         1 |        36 |         0 |        37 |
             |     0.000 |     0.000 |     0.027 |     0.973 |     0.000 |     0.175 |
             |     0.000 |     0.000 |     0.033 |     0.735 |     0.000 |           |
             |     0.000 |     0.000 |     0.005 |     0.171 |     0.000 |           |
-------------|-----------|-----------|-----------|-----------|-----------|-----------|
           5 |         0 |         0 |         0 |         7 |        42 |        49 |
             |     0.000 |     0.000 |     0.000 |     0.143 |     0.857 |     0.232 |
             |     0.000 |     0.000 |     0.000 |     0.143 |     1.000 |           |
             |     0.000 |     0.000 |     0.000 |     0.033 |     0.199 |           |
-------------|-----------|-----------|-----------|-----------|-----------|-----------|
Column Total |        34 |        56 |        30 |        49 |        42 |       211 |
             |     0.161 |     0.265 |     0.142 |     0.232 |     0.199 |           |
```

Here we have the Confusing Matrix For our data set with Prediction.

K-Means Clustering :



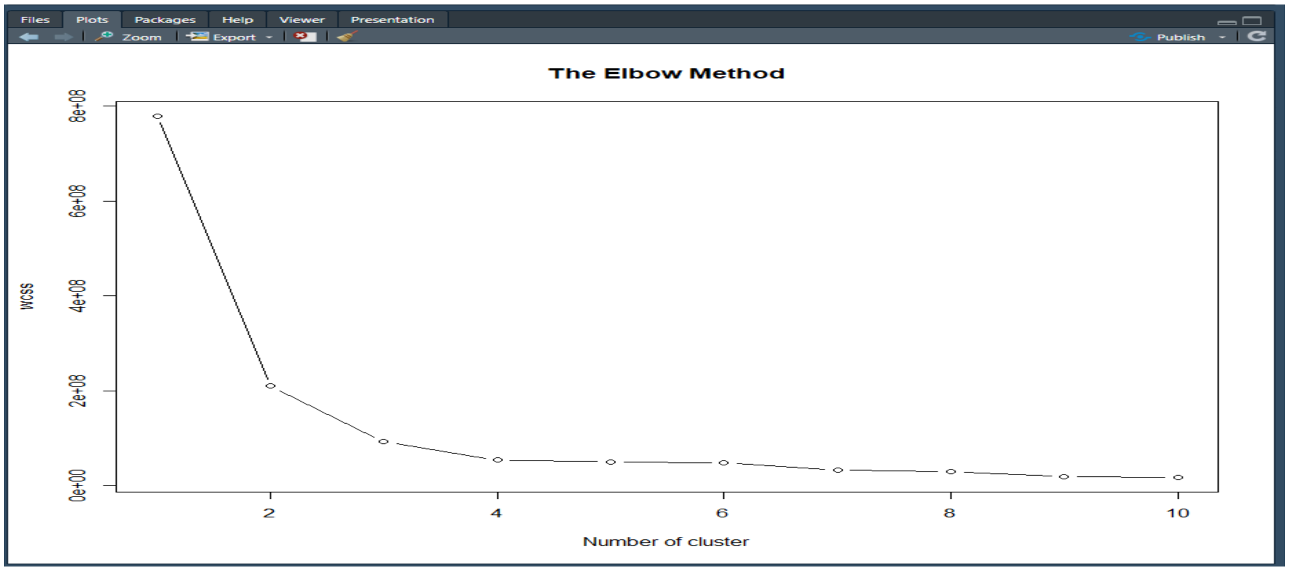
```r
################################### k- mean ###########################################

set.seed(6)
wcss = vector()
for (i in 1:10) wcss[i] = sum(kmeans(user_data, i)$withinss)
plot(1:10,
     wcss,
     type = 'b',
     main = paste('The Elbow Method'),
     xlab ='Number of cluster',
     ylab = 'wcss')

#Fitting k-mean to the dataset
set.seed(29)
kmeans= kmeans(x = user_data, centers = 6,
               iter.max=500)
y_kmeans = kmeans$cluster
y_kmeans

x = kmeans$cluster
x

getwd()
write.csv(x,"cluster.csv")

install.packages("cluster")
library(cluster)
clusplot(user_data,
         y_kmeans,
         lines = 0,
         shade = TRUE,
         color = TRUE,
         labels = 2,
         plotchar = FALSE,
         span = TRUE,
         main = paste('clusters of User behavior'),
         xlab = 'App Usage Time (min/day)',
         ylab = 'Data Usage (MB/day)')
```

```
Data
kmeans             List of  9
test_data          211 obs. of 9 variables
train_data         489 obs. of 9 variables
user_data          700 obs. of 10 variables
user_data_norm     700 obs. of 10 variables
Values
i                  10L
test_labels        int [1:211] 4 3 3 2 4 4 1 4 3 1 ...
test_pred          Factor w/ 5 levels "1","2","3","4",..: 4 4 4 2 4 4 1 4…
train_indices      int [1:489] 561 321 153 74 228 146 634 49 128 303 ...
train_labels       int [1:489] 4 3 2 1 3 3 2 5 3 4 ...
wcss               num [1:10] 7.78e+08 2.10e+08 9.27e+07 5.39e+07 5.05e+0…
x                  int [1:700] 3 6 5 6 6 2 3 1 3 3 ...
y_kmeans           int [1:700] 3 6 5 6 6 2 3 1 3 3 ...
```

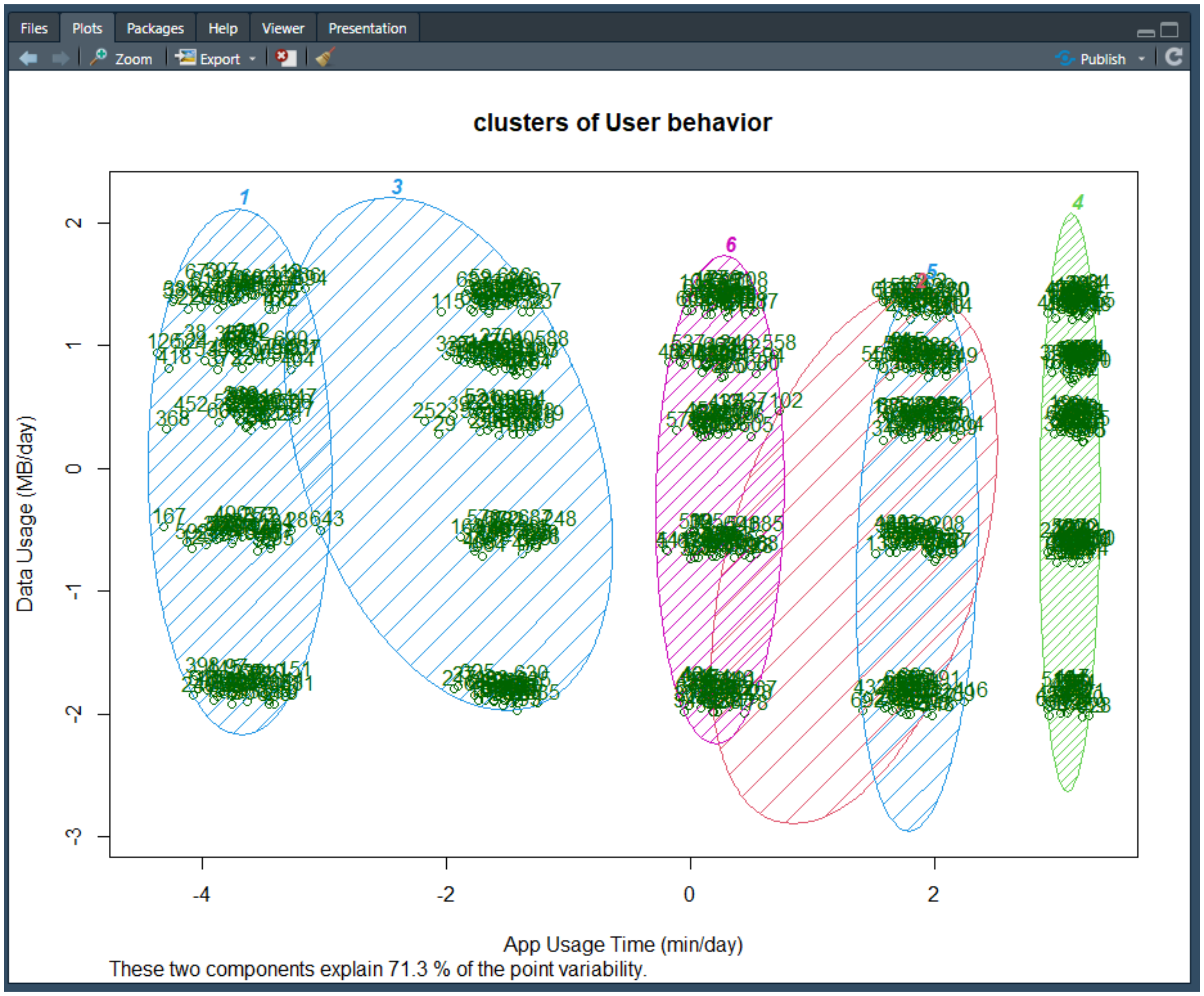
```
+          color = TRUE,
+          labels = 2,
+          plotchar = FALSE,
+          span = TRUE,
+          main = paste('clusters of User behavior'),
+          xlab = 'App Usage Time (min/day)',
+          ylab = 'Data Usage (MB/day)')
> plot(1:10,
+      wcss,
+      type = 'b',
+      main = paste('The Elbow Method'),
+      xlab ='Number of cluster',
+      ylab = 'wcss')
```

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Untitled1* Untitled2* user_data
Source on Save
Run Source
57:1 k- mean R Script
Console Terminal Background Jobs
R 4.4.1 ~/
Environment History Connections Tutorial
Import Dataset 203 MiB List
Global Environment
Files Plots Packages Help Viewer Presentation
Zoom Export Publish
The Elbow Method
wcss
0e+00
2e+08
4e+08
6e+08
8e+08
2
4
6
8
10
Number of cluster

This code uses K-Means clustering in R to analyze user behavior patterns in user_data. First, the Elbow Method is applied to find the optimal number of clusters by plotting the within-cluster sum of squares (WCSS) for clusters ranging from 1 to 10. The "elbow" in this plot suggests the optimal cluster count, where adding more clusters provides diminishing returns in WCSS reduction. With the optimal number identified (6 clusters in this case), the K-Means algorithm is run with centers = 6 and a maximum of 500 iterations to generate clusters. The resulting cluster assignments are saved to a CSV file for further analysis.

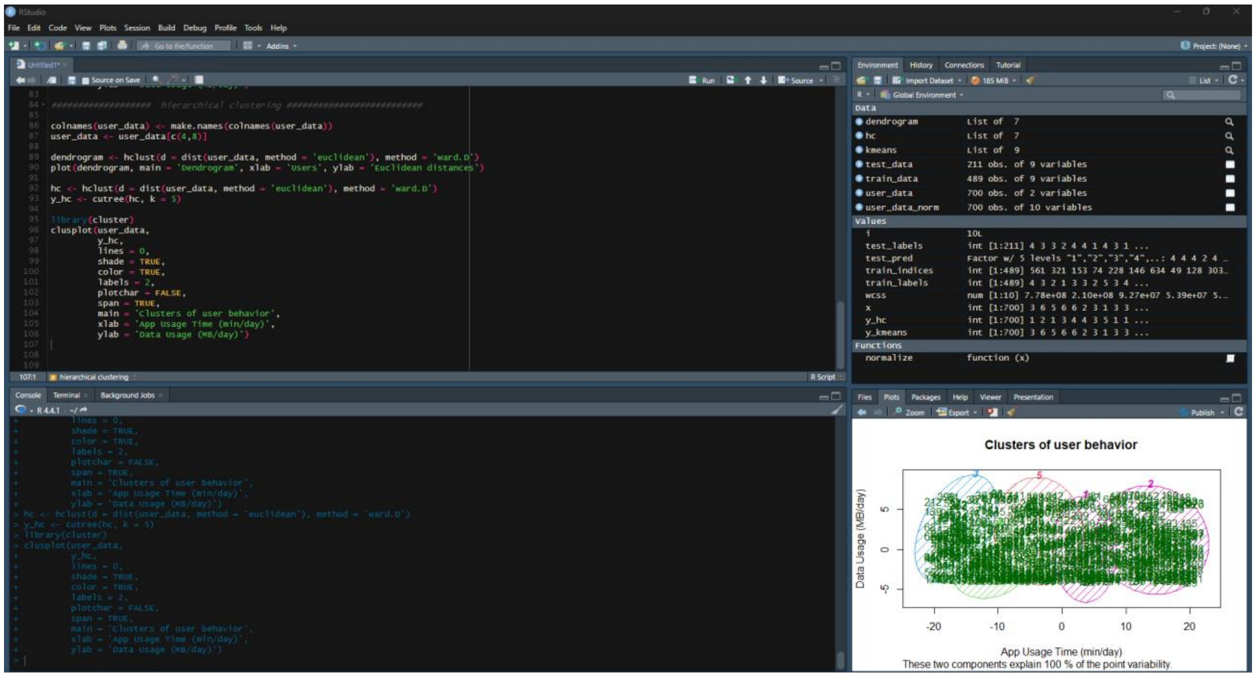


Finally, the clusplot function visualizes the clusters with shading and color differentiation, providing a clear, 2D view of the cluster distribution. This plot helps in understanding the cohesion and separation between clusters based on two attributes, labeled here as "App Usage Time" and "Data Usage." Overall, this workflow identifies distinct user behavior segments, aiding in interpreting and analyzing clustering effectiveness within the dataset.



This plot shows the clusters of user behavior, segmented by the K-Means algorithm based on "App Usage Time (min/day)" and "Data Usage (MB/day)." Each color and shaded ellipse represents a distinct cluster, helping to visualize the separation and distribution of user groups.

Hierarchical Clustering:



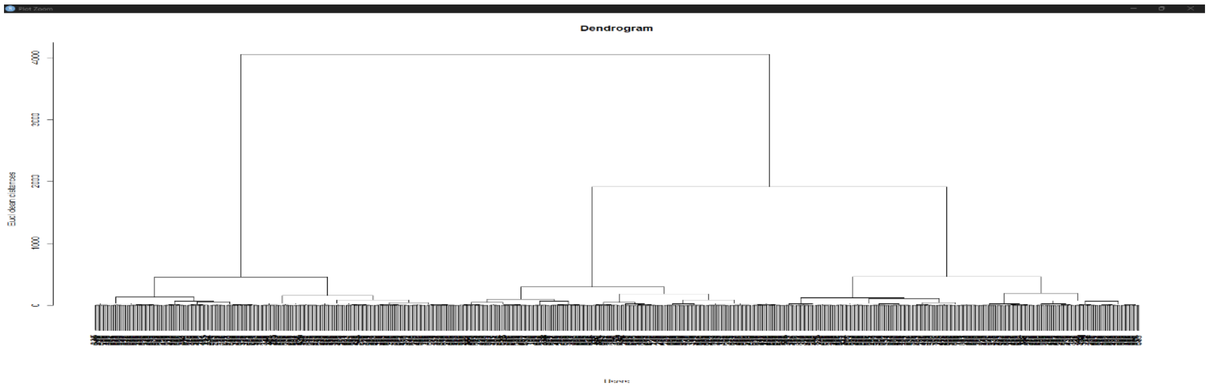
```
#####################  hierarchical clustering ###########################

colnames(user_data) <- make.names(colnames(user_data))
user_data <- user_data[c(4,8)]

dendrogram <- hclust(d = dist(user_data, method = 'euclidean'), method = 'ward.D')
plot(dendrogram, main = 'Dendrogram', xlab = 'Users', ylab = 'Euclidean distances')

hc <- hclust(d = dist(user_data, method = 'euclidean'), method = 'ward.D')
y_hc <- cutree(hc, k = 5)

library(cluster)
clusplot(user_data,
         y_hc,
         lines = 0,
         shade = TRUE,
         color = TRUE,
         labels = 2,
         plotchar = FALSE,
         span = TRUE,
         main = 'Clusters of user behavior',
         xlab = 'App Usage Time (min/day)',
         ylab = 'Data Usage (MB/day)')
```

```
Data
dendrogram        List of  7
hc                List of  7
kmeans            List of  9
test_data         211 obs. of 9 variables
train_data        489 obs. of 9 variables
user_data         700 obs. of 2 variables
user_data_norm    700 obs. of 10 variables
Values
i                 10L
test_labels       int [1:211] 4 3 3 2 4 4 1 4 3 1 ...
test_pred         Factor w/ 5 levels "1","2","3","4",..: 4 4 4 2 4 _
train_indices     int [1:489] 561 321 153 74 228 146 634 49 128 303_
train_labels      int [1:489] 4 3 2 1 3 3 2 5 3 4 ...
wcss              num [1:10] 7.78e+08 2.10e+08 9.27e+07 5.39e+07 5._
x                 int [1:700] 3 6 5 6 6 2 3 1 3 3 ...
y_hc              int [1:700] 1 2 1 3 4 4 3 5 1 1 ...
y_kmeans          int [1:700] 3 6 5 6 6 2 3 1 3 3 ...
Functions
normalize         function (x)
```

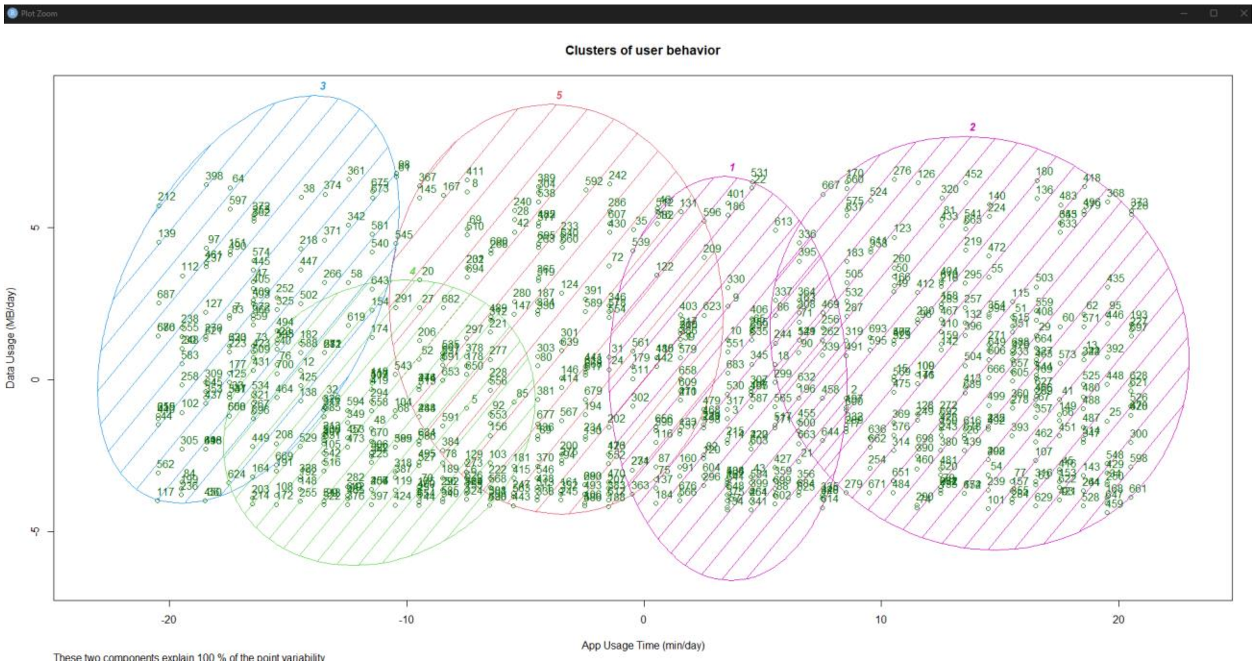
```
+          lines = 0,
+          shade = TRUE,
+          color = TRUE,
+          labels = 2,
+          plotchar = FALSE,
+          span = TRUE,
+          main = 'Clusters of user behavior',
+          xlab = 'App Usage Time (min/day)',
+          ylab = 'Data Usage (MB/day)')
> hc <- hclust(d = dist(user_data, method = 'euclidean'), method = 'ward.D')
> y_hc <- cutree(hc, k = 5)
> library(cluster)
> clusplot(user_data,
+          y_hc,
+          lines = 0,
+          shade = TRUE,
+          color = TRUE,
+          labels = 2,
+          plotchar = FALSE,
+          span = TRUE,
+          main = 'Clusters of user behavior',
+          xlab = 'App Usage Time (min/day)',
+          ylab = 'Data Usage (MB/day)')
>
```

Clusters of user behavior
Data Usage (MB/day)
App Usage Time (min/day)
These two components explain 100 % of the point variability.

This code performs **Hierarchical Clustering** on the user_data dataset in R to group users based on selected attributes, followed by visualization with a dendrogram and cluster plot. First, it ensures all column names are syntactically valid and selects two key features (e.g., "App Usage Time" and "Data Usage") for analysis. Using **Euclidean distance** and **Ward's method**, it computes hierarchical clusters that minimize within-cluster variance. The dendrogram displays hierarchical relationships and helps determine the optimal number of clusters, with larger merges indicating greater dissimilarity. After determining the number of clusters (k), the data is segmented accordingly, and the results are visualized in a 2D cluster plot using clusplot(). This plot shows clusters as distinct, color-coded regions, allowing for easy interpretation of how users are grouped based on the selected behavioral features.



This dendrogram represents the hierarchical clustering of users based on Euclidean distances. Two main clusters are visible, with higher distances indicating greater dissimilarity between the groups.

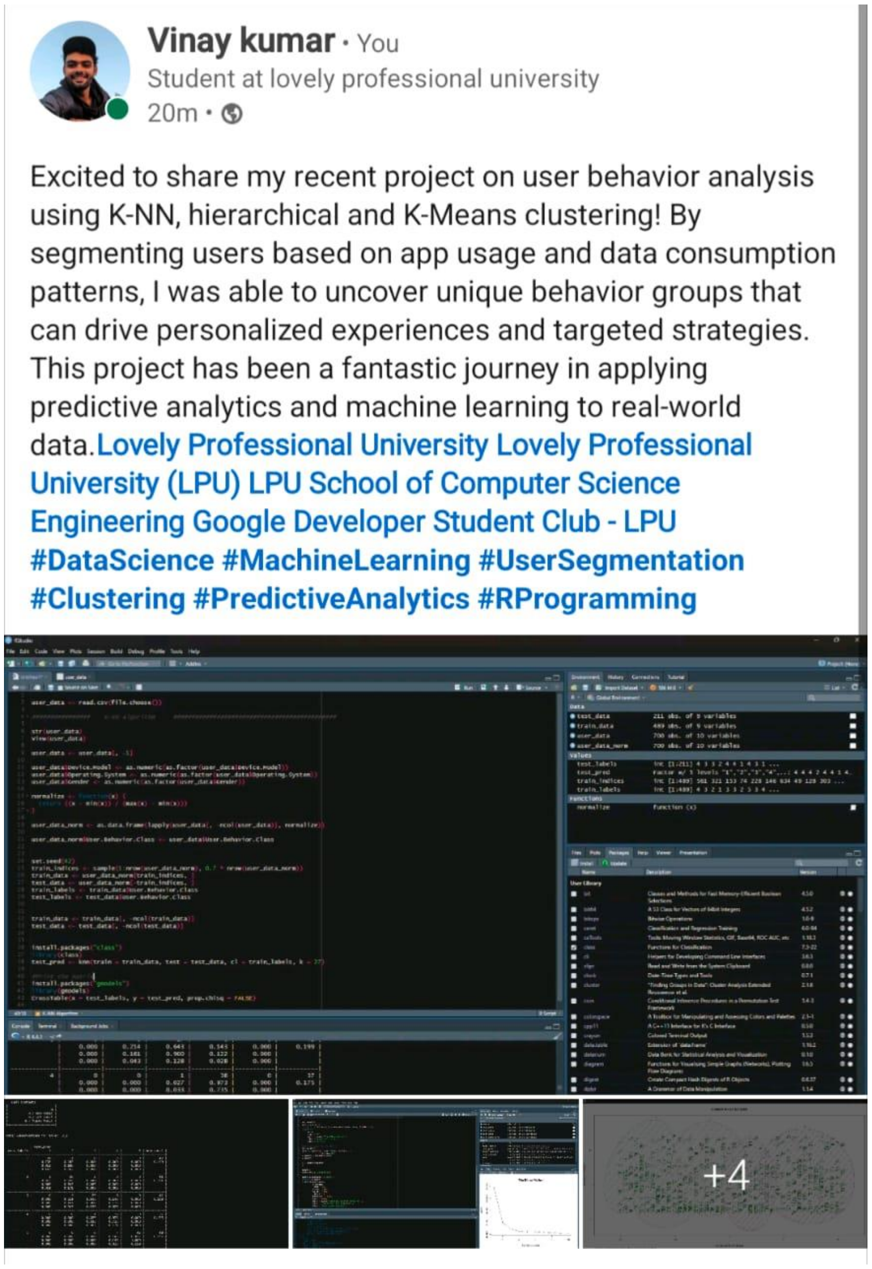


This cluster plot visualizes user behavior by grouping users based on "App Usage Time" (x-axis) and "Data Usage" (y-axis). There are five distinct clusters, each represented by a shaded ellipse with a unique color for clear separation. Cluster numbers are labeled at the center of each group. Data points within each cluster are displayed as green numbers, indicating individual users.

The clusters are structured as follows:

1. Cluster 3 (blue) contains users with low app usage time and data usage.
2. Cluster 5 (red) includes users with moderate app usage and data usage levels.
3. Cluster 1 (purple) has users with higher data usage and slightly higher app usage.
4. Cluster 2 (magenta) includes users with the highest app usage and moderately high data usage.
5. Cluster 4 (green) represents a middle range of app usage and data usage.

LINKEDIN POST



Vinay kumar • You
Student at lovely professional university
20m •

Excited to share my recent project on user behavior analysis using K-NN, hierarchical and K-Means clustering! By segmenting users based on app usage and data consumption patterns, I was able to uncover unique behavior groups that can drive personalized experiences and targeted strategies. This project has been a fantastic journey in applying predictive analytics and machine learning to real-world data.Lovely Professional University Lovely Professional University (LPU) LPU School of Computer Science Engineering Google Developer Student Club - LPU #DataScience #MachineLearning #UserSegmentation #Clustering #PredictiveAnalytics #RProgramming

```
user_data <- read.csv(file.choose())

str(user_data)
View(user_data)

user_data <- user_data[, -1]

user_data$Device.Model <- as.numeric(as.factor(user_data$Device.Model))
user_data$Operating.System <- as.numeric(as.factor(user_data$Operating.System))
user_data$Gender <- as.numeric(as.factor(user_data$Gender))

normalize <- function(x) {
  return ((x - min(x)) / (max(x) - min(x)))
}

user_data_norm <- as.data.frame(lapply(user_data[, -ncol(user_data)], normalize))

user_data_norm$User.Behavior.Class <- user_data$User.Behavior.Class


set.seed(42)
train_indices <- sample(1:nrow(user_data_norm), 0.7 * nrow(user_data_norm))
train_data <- user_data_norm[train_indices, ]
test_data <- user_data_norm[-train_indices, ]
train_labels <- train_data$User.Behavior.Class
test_labels <- test_data$User.Behavior.Class


train_data <- train_data[, -ncol(train_data)]
test_data <- test_data[, -ncol(test_data)]


install.packages("class")
library(class)
test_pred <- knn(train = train_data, test = test_data, cl = train_labels, k = 27)

install.packages("gmodels")
library(gmodels)
CrossTable(x = test_labels, y = test_pred, prop.chisq = FALSE)
```

```
         |     0.000 |     0.214 |     0.643 |     0.143 |     0.000 |     0.199 |
         |     0.000 |     0.161 |     0.900 |     0.122 |     0.000 |           |
         |     0.000 |     0.043 |     0.128 |     0.028 |     0.000 |           |
---------|-----------|-----------|-----------|-----------|-----------|-----------|
       4 |         0 |         0 |         1 |        36 |         0 |        37 |
         |     0.000 |     0.000 |     0.027 |     0.973 |     0.000 |     0.175 |
         |     0.000 |     0.000 |     0.033 |     0.735 |     0.000 |           |
```

+4

LINK :

https://www.linkedin.com/posts/vinay-kumar-9b27411b9_datascience-machinelearning-usersegmentation-activity-7259306141975797760-CPgJ?utm_source=screenshot_social_share&utm_medium=android_app&utm_campaign=copy_link