Thorough Evaluation of Data Mining Techniques

What is data mining?

Data mining is a multifaceted process of extracting meaningful patterns, insights, and knowledge from large datasets using a combination of statistical, machine learning, and database techniques. It involves the systematic analysis of data to uncover hidden relationships, trends, and structures that can be valuable for decision-making, prediction, and understanding complex phenomena. Through data mining, organizations can sift through vast amounts of data to identify patterns that may not be immediately apparent, allowing them to make informed decisions, optimize processes, and gain competitive advantages. Techniques such as association rule mining, classification, clustering, regression analysis, and anomaly detection are commonly employed in data mining to uncover valuable insights from data. It is a fundamental component of modern data-driven approaches, playing a crucial role in fields ranging from business and finance to healthcare, marketing, and scientific research. Overall, data mining facilitates the extraction of actionable knowledge from data, enabling organizations to derive valuable insights and drive innovation.

TASK 01

Applying K-Nearest Neighbors Classifier on Titanic Dataset

Table Of Content

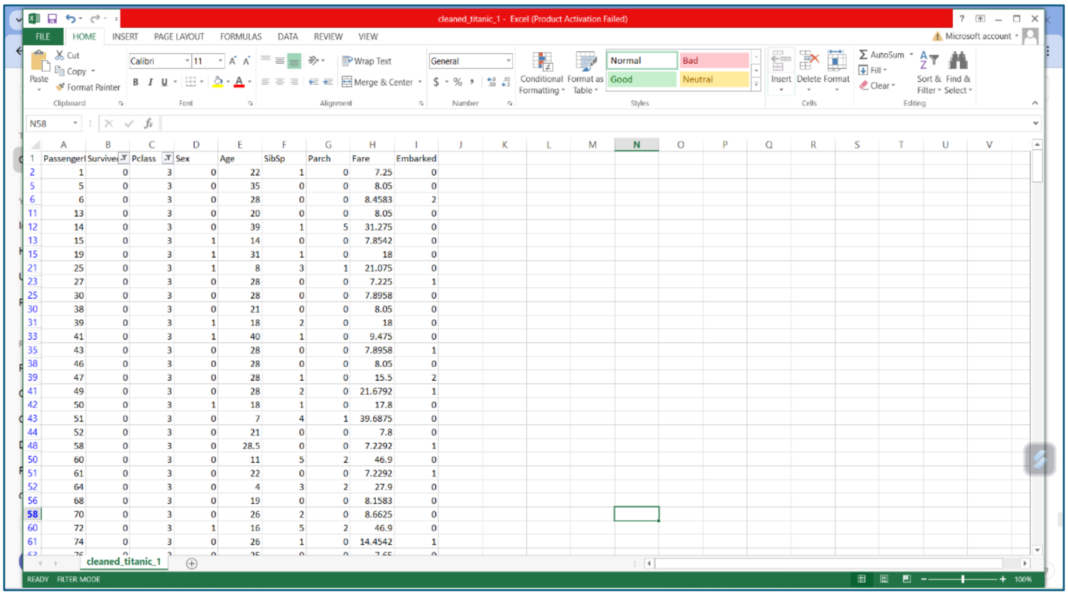
1.Introduction

In this document, we investigate the use of the R programming language and the K-Nearest Neighbour (KNN) machine learning algorithm on the Titanic dataset. A well-known dataset that is frequently used in data science and machine learning is the Titanic dataset.

It includes details about the passengers who were on board the Titanic, such as age, gender, ticket class, and whether or not they survived the disaster that occurred. This analysis's main goal is to use the KNN classifier to predict, from the available features, whether or not a passenger survived. KNN is an easy-to-understand algorithm that uses most of them class of its closest neighbour to classify data points. We hope to illustrate the process of developing and accessing a machine learning model using actual data by applying KNN to the Titanic dataset.

1.1 Dataset

This dataset was taken from Titanic Disaster Dataset ***(https://data.world/nrippner/titanic-disaster-dataset)*** in 'data. World' website. A well-known dataset, the Titanic dataset includes details about the people who were on board the ship, including whether they survived. Numerous features, including age, sex, class, fare, and embarked port, are included in the dataset. The preprocessed dataset was divided into training and testing sets. There are 418 instances in the testing set compared to 891 instances in the training set.



1	PassengerI	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
2	1	0	3	0	22	1	0	7.25	0
5	5	0	3	0	35	0	0	8.05	0
6	6	0	3	0	28	0	0	8.4583	2
11	13	0	3	0	20	0	0	8.05	0
12	14	0	3	0	39	1	5	31.275	0
13	15	0	3	1	14	0	0	7.8542	0
15	19	0	3	1	31	1	0	18	0
21	25	0	3	1	8	3	1	21.075	0
23	27	0	3	0	28	0	0	7.225	1
25	30	0	3	0	28	0	0	7.8958	0
30	38	0	3	0	21	0	0	8.05	0
31	39	0	3	1	18	2	0	18	0
33	41	0	3	1	40	1	0	9.475	0
35	43	0	3	0	28	0	0	7.8958	1
38	46	0	3	0	28	0	0	8.05	0
39	47	0	3	0	28	1	0	15.5	2
41	49	0	3	0	28	2	0	21.6792	1
42	50	0	3	1	18	1	0	17.8	0
43	51	0	3	0	7	4	1	39.6875	0
44	52	0	3	0	21	0	0	7.8	0
48	58	0	3	0	28.5	0	0	7.2292	1
50	60	0	3	0	11	5	2	46.9	0
51	61	0	3	0	22	0	0	7.2292	1
52	64	0	3	0	4	3	2	27.9	0
56	68	0	3	0	19	0	0	8.1583	0
58	70	0	3	0	26	2	0	8.6625	0
60	72	0	3	1	16	5	2	46.9	0
61	74	0	3	0	26	1	0	14.4542	1

Figure 1: Data Set

2.Explanation and preparation of datasets

The Titanic dataset contains several features, including age, sex, class, fare, , and the target variable survived. Age is a continuous variable, while sex, class, and embarked port are categorical variables. We will convert the categorical variables into dummy variables to make them suitable for use in the KNN algorithm.

We will split the dataset into training and testing sets, using 70% of the data for training and 30% for testing. We will then standardize the training and testing sets to ensure that all the features have zero mean and unit variance.

Description of Variables:

- Passenger: Unique identifier for each passenger.
- Survived: Binary variable indicating whether the passenger survived (0 = No, 1 = Yes).
- Pclass: Ticket class (1st, 2nd, or 3rd).
- Sex: Passenger's gender (male or female).
- Age: Passenger's age in years.
- SibSp: Number of siblings/spouses aboard.
- Parch: Number of parents/children aboard.
- Fare: Ticket fare.
- Embarked: Port of embarkation (C = Cherbourg, Q = Queenstown, S = Southampton).

Source of dataset: The dataset is sourced from Dataworld. Specifically, it is available at the following link: Titanic Dataset.: ***(https://data.world/nrippner/titanic-disaster-dataset)***

Number of instances: The Titanic dataset contains information about 891

Number of attributes: The dataset comprises a total of 9 attributes,

Number of classes: For the classification task, the dataset has two classes for the target variable "Survived": 0 (did not survive) and 1 (survived).

Number of missing values: The dataset may contain missing values that need to be handled during data preprocessing. Specific details about the number of missing values in each attribute can be obtained by analyzing

Related task: The primary task associated with the Titanic dataset is classification. The objective is to predict whether a passenger survived or not based on the available attributes.

3.Data mining

3.1 Implementation in R

3.1.1 Data Preprocessing

We used the R programming language to prepare this dataset. In this section, we'll discuss the codes and functions of the R software. As a result, we selected a few functions to get this data set ready. The data packages that we install in the R language to carry out classification in data mining will now be covered.

In this report, we detail the implementation of classification using the Titanic dataset in R programming language. The Titanic dataset, obtained from data world, consists of information regarding passengers aboard the Titanic, including various attributes such as age, gender, ticket class, and survival status. The main objective of this implementation is to predict passenger survival based on the provided attributes.

Step 01:

Before proceeding with data preprocessing, we must install and load the required packages. The following packages are essential for our analysis:

- caTools: Provides functions for data splitting and sampling.
- dplyr: Offers a wide range of functions for data manipulation and transformation.
- ggplot2: Used for data visualization and exploratory data analysis.
- class: Contains functions for building and training the K-Nearest Neighbors (KNN) classifier.
- caret: Provides a unified interface for training and evaluating machine learning models.
- corrplot: Enables visualization of correlation matrices.

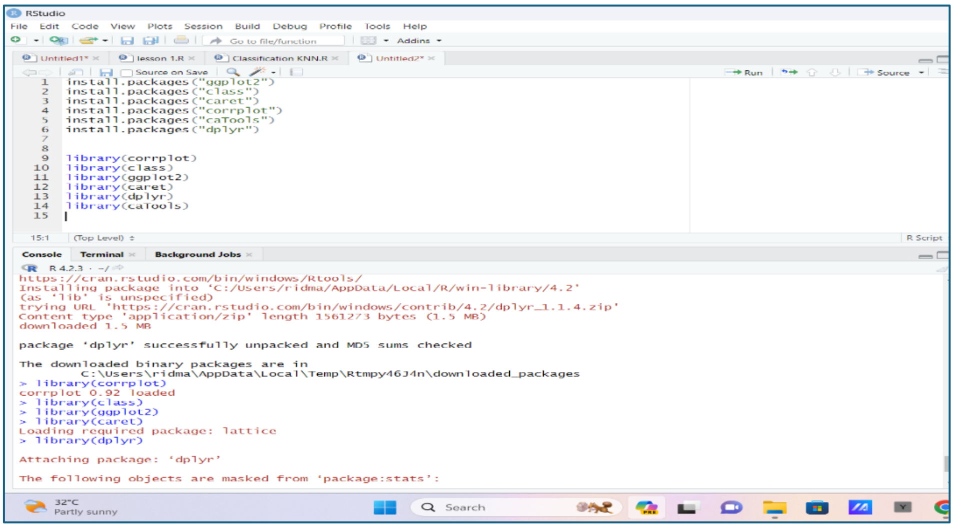


Figure 2: install and load the required libraries.

Step 02:

We set the working directory and import the comma-separated (CSV) data file to the R studio and assigned it as “cleaned-titanic” and import the heart disease dataset to R studio.

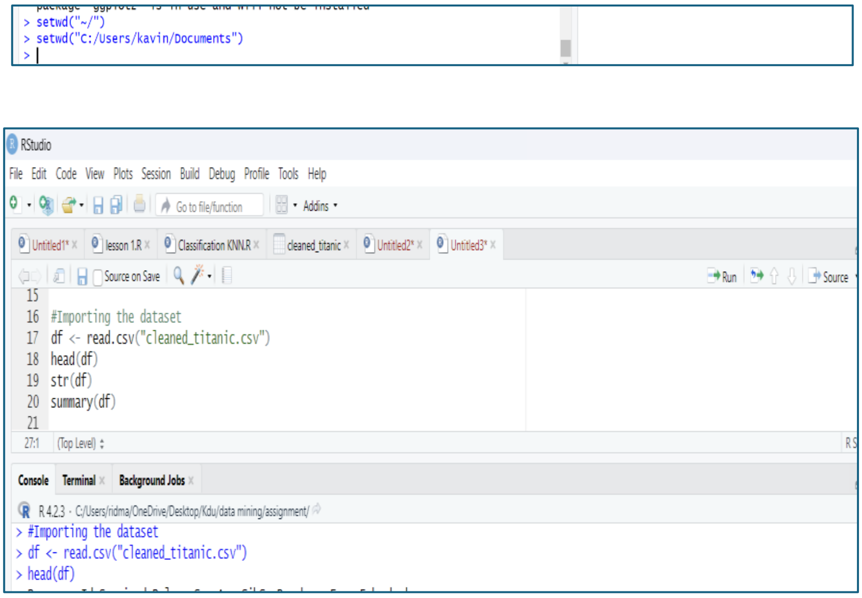


Figure 3: Import the data.

Step 03

After importing the dataset to the R studio we Inspect dataset the dataset by using the following codes.

- Names: Displays the column names of the dataset.
- Head: Shows the first 6 rows of the dataset.
- Tail: Shows the last few rows of the dataset.
- Summary: Provides summary statistics for numerical variables in the dataset.
- Str: Displays the structure of the dataset, including variable names, data types, and a summary of the first few rows.

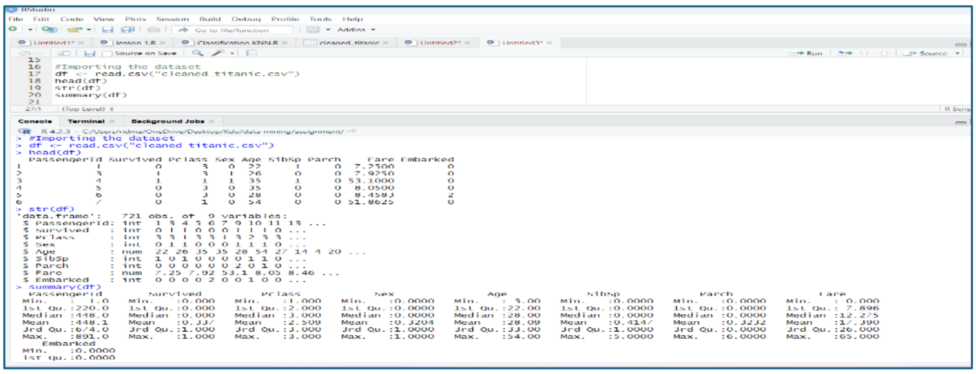
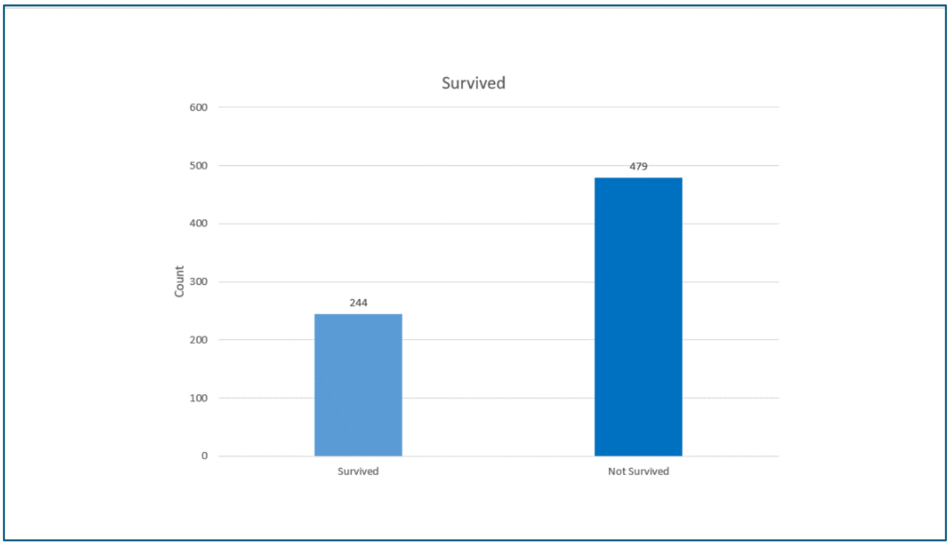
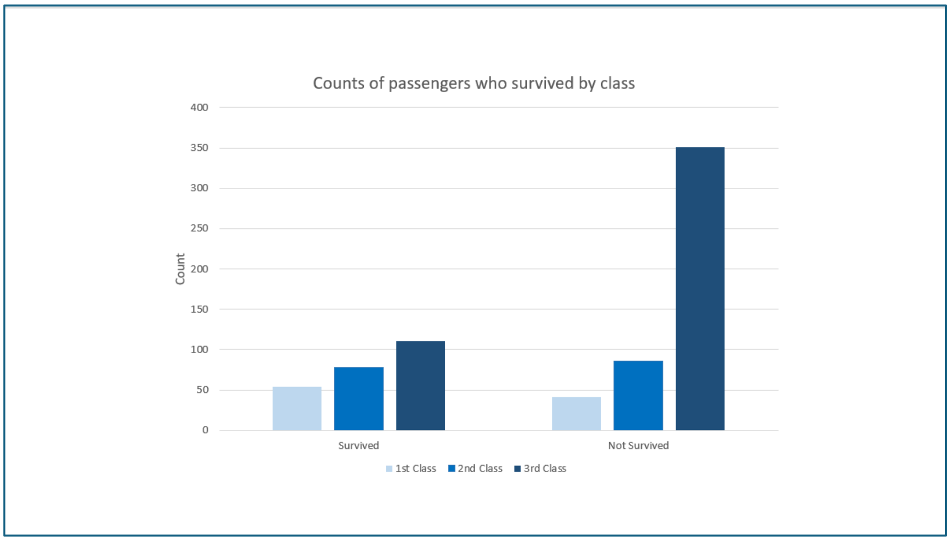


Figure 4: explore the dataset

- Let’s have an idea about the dataset and the targeted classes by building simple plots to identify the relationships.



- The bar chart you sent appears to show the distribution of survivors and non-survivors from the Titanic disaster.



- It shows the number of passengers who survived and did not survive the Titanic disaster, broken down by passenger class.

3.2 Data standardization

Standardization, also known as z-score normalization, is a common preprocessing technique used in machine learning to rescale numerical features to have a mean of 0 and a standard deviation of 1. Here's how you can standardize numerical variables in the Titanic dataset:

Step 04

select the features you want to standardize. In this case, let's select the (age, sibsp, parch, fare, pclass, and embarked variables)

We then use the scale() function to standardize these features, and store the result in the standard.features data frame.

```
standard.features <- scale(df[,1:8])
```

Figure 5: standard.features data frame.

Step 05

Preprocess the dataset by converting categorical variables such as sex, class, and embarked port into dummy variables.

The NULL and NA values are one more important item to check before moving with the further steps.

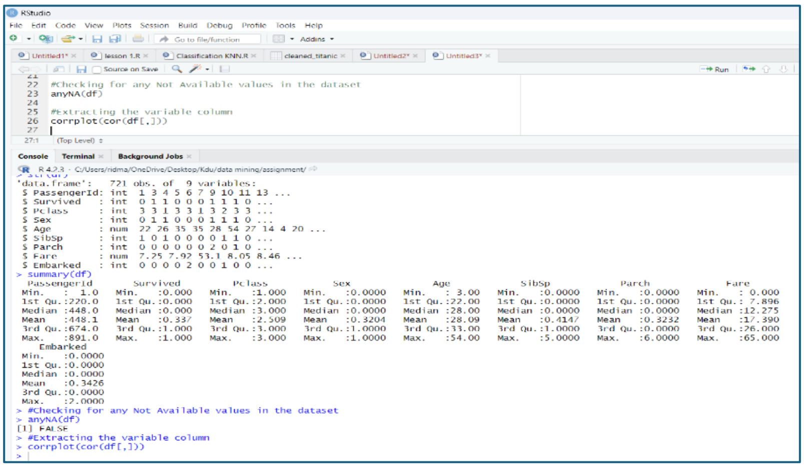


Figure 6: Check if there are any missing values to impute

Step 06

we use the cbind() function to join the standardized features with the Survived column from the titanic data frame. This creates a new data frame data that includes the standardized features and the target variable.

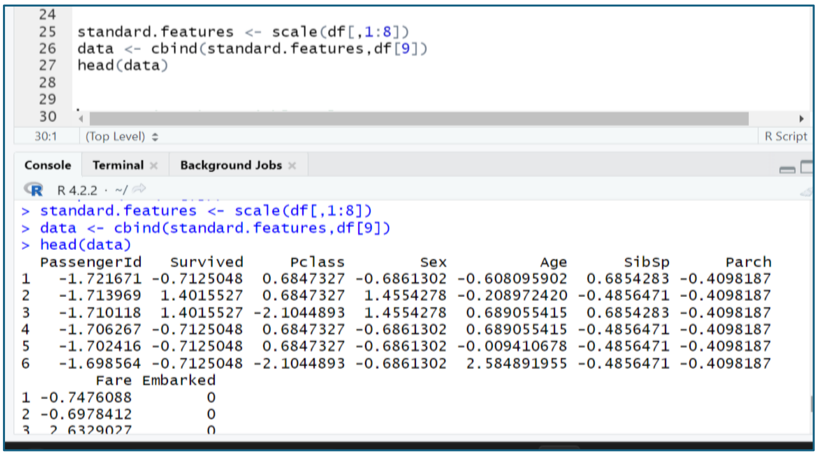


Figure 7

Step 07

Finally, we can view the summary statistics of the standardized features by using the summary () function.

This will give you the minimum, 1st quartile, median, mean, 3rd quartile, and maximum values for each standardized feature.

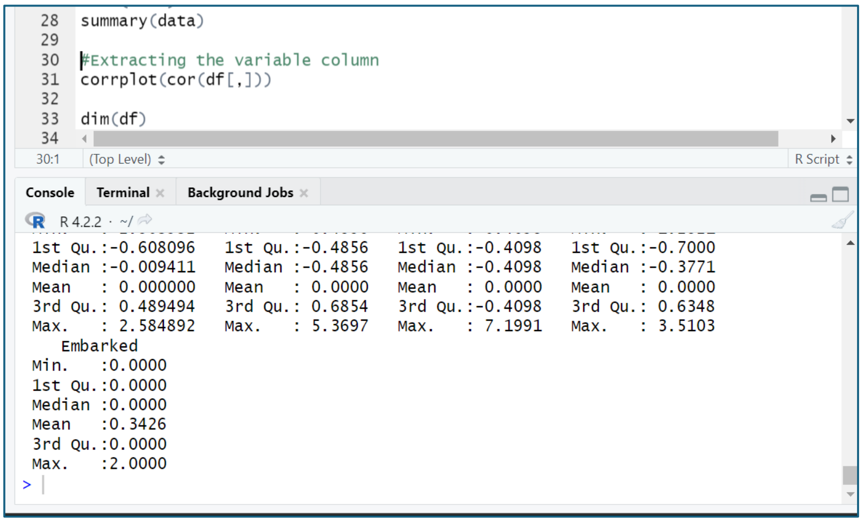


Figure 8: Summary of the data set

3.3 Data Visualization

Step 08

check if the dataset has any correlations between the data columns.

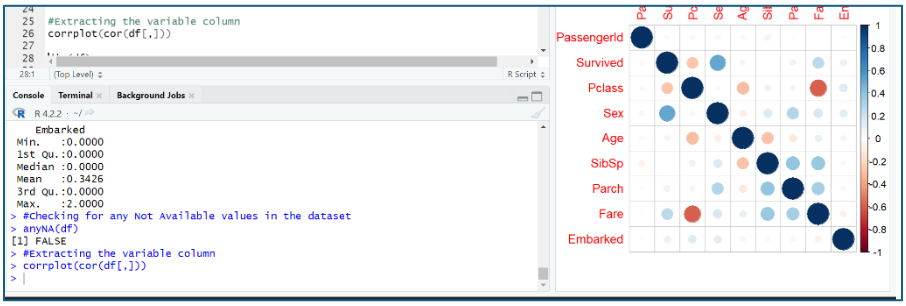


Figure 9: Below plot explains the relation between different features.

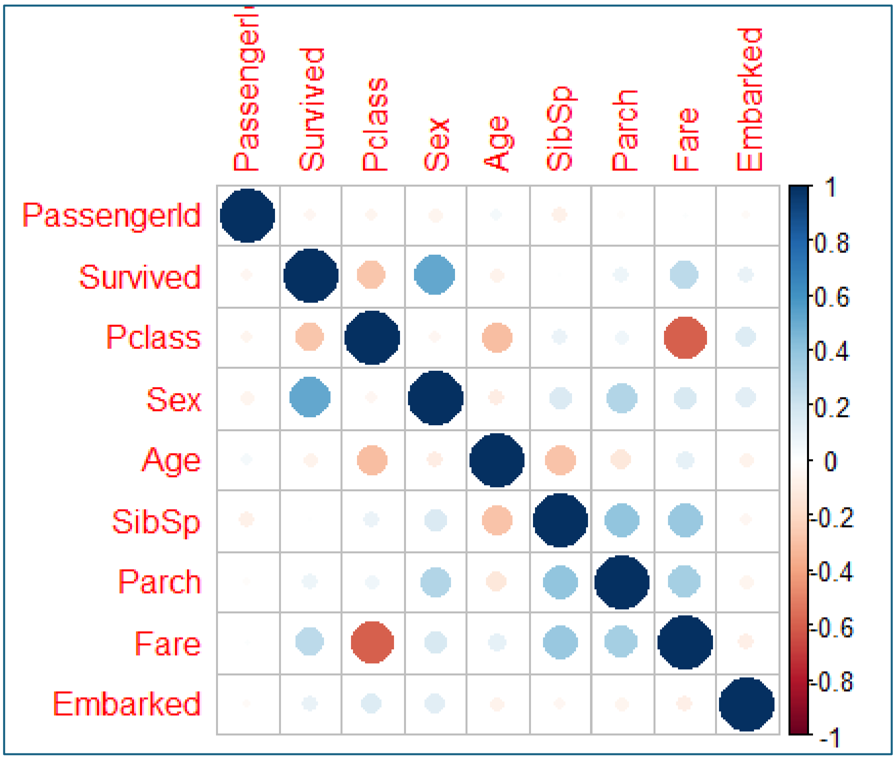


Figure 10: Plot

In this code, we use the cor() function to calculate the correlations between the specified variables. The cor() function returns a correlation matrix, which is a square matrix with the same dimensions as the number of variables. The diagonal elements of the correlation matrix are always 1, because a variable is perfectly correlated with itself. The off-diagonal elements represent the correlation coefficients between the corresponding pair of variables.

3.4 Test and Train Data Split

Step 07

After standardization, the data splicing into the training and testing dataset

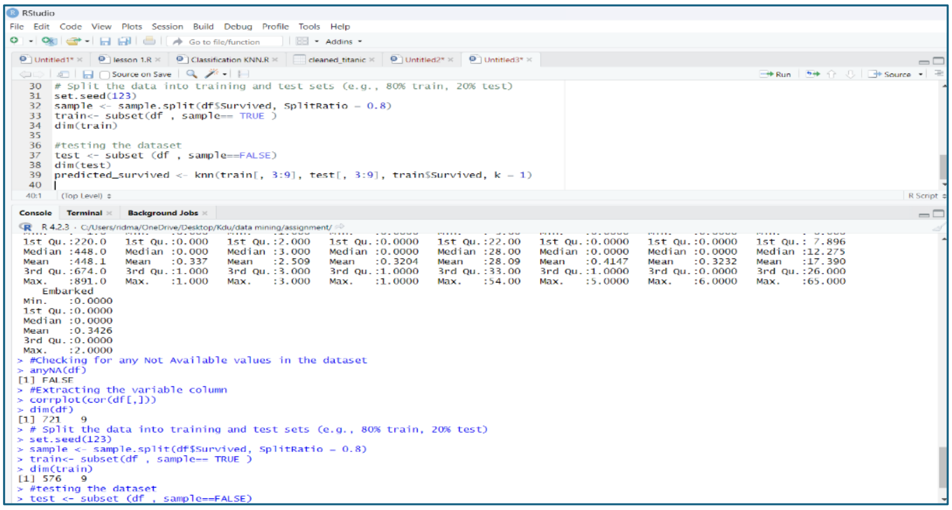


Figure 11: sample.split() function is used to divide the data into two sets, train set and the test set

We use the set.seed() function to set the seed for reproducibility. This ensures that the random splitting of the data will always produce the same results.Finally, we use the train vector to subset the training and testing data from the original data frame. Note that in this example, we're using a 70% training set and a 30% testing set.

The resulting correlation matrix will show the correlation coefficients between each pair of variables, ranging from -1 (perfect negative correlation) to 1 (perfect positive correlation).

In general, variables with a correlation coefficient close to 1 or -1 are highly correlated, while variables with a correlation coefficient close to 0 are not correlated. In the context of the Titanic dataset, you might expect some correlation between the age and fare variables, as older passengers may have been more likely to afford higher fares. However, there should not be a strong correlation between the survived variable and any of the other variables, as survival on the Titanic was largely determined by factors outside of the passengers' control.

3.5 KNN Model

In this report, we aim to predict the "Type" variable in the Titanic dataset using the k-nearest neighbors (KNN) algorithm with k=1

As the confusion matrix says, our model has obtained 6.69% accuracy.

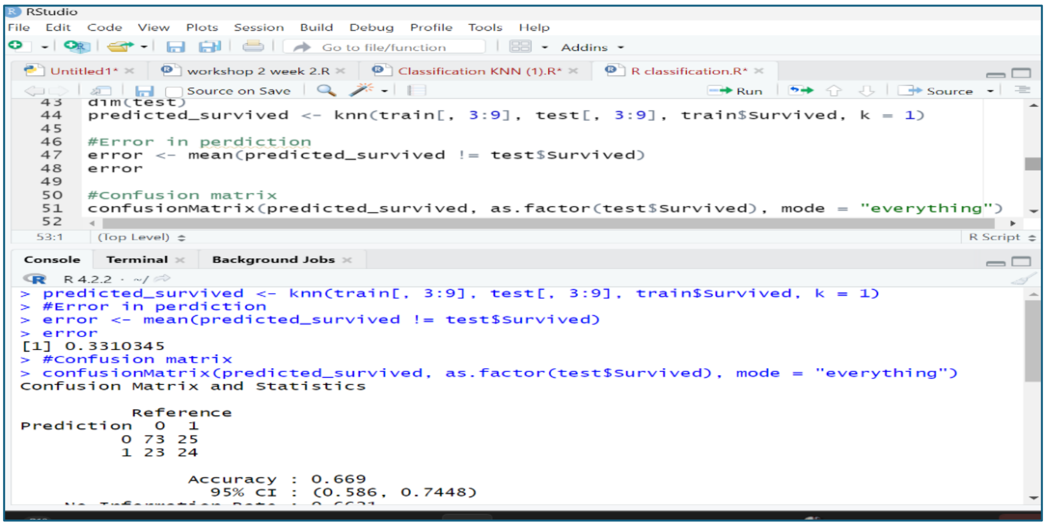


Figure 12: predict our target variable

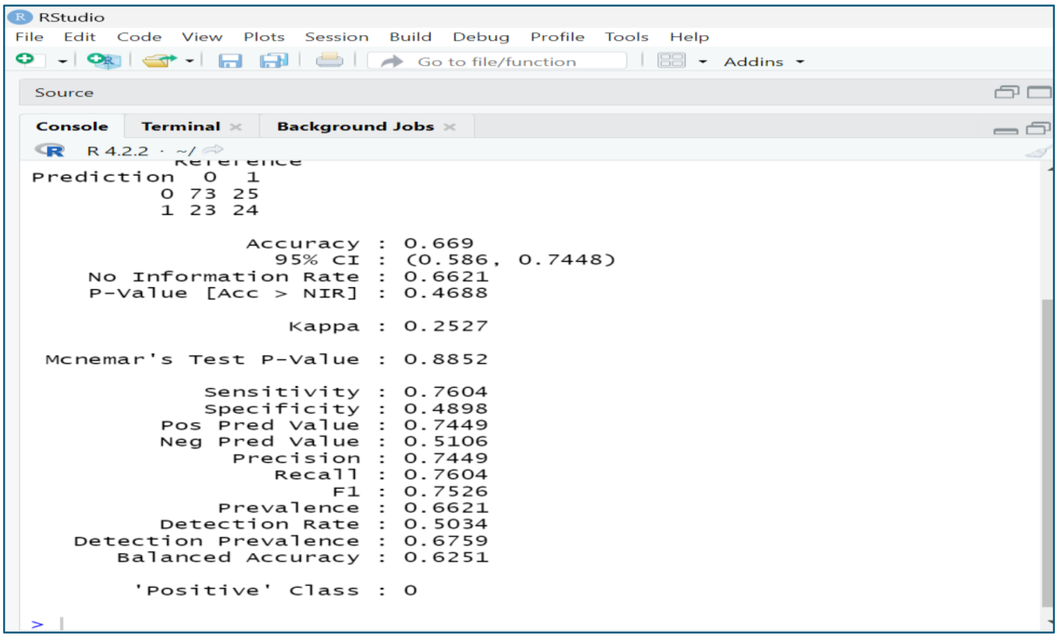


Figure 13

In this updated code, we first load the class package. Then, we convert the training data and test data into matrices using the as.matrix() function.We then use the knn() function to predict the target variable Type of the test dataset with k=1.We calculate the error by comparing the predicted values with the actual values in the test_data data frame.

Finally, we create a confusion matrix using the confusion Matrix() function from the caret package.

Step 10

we initialize the predicted_type and error_rate vectors to store the predicted values and error rates, respectively.

We loop through the values of k from 1 to 10. For each value of k, we use the

knn() function to predict the target variable Survived of the test dataset.

We calculate the error rate for each value of k by comparing the predicted values with the actual values in the test_data data frame.

Finally, we create a data frame to store the k values and the corresponding error rates.

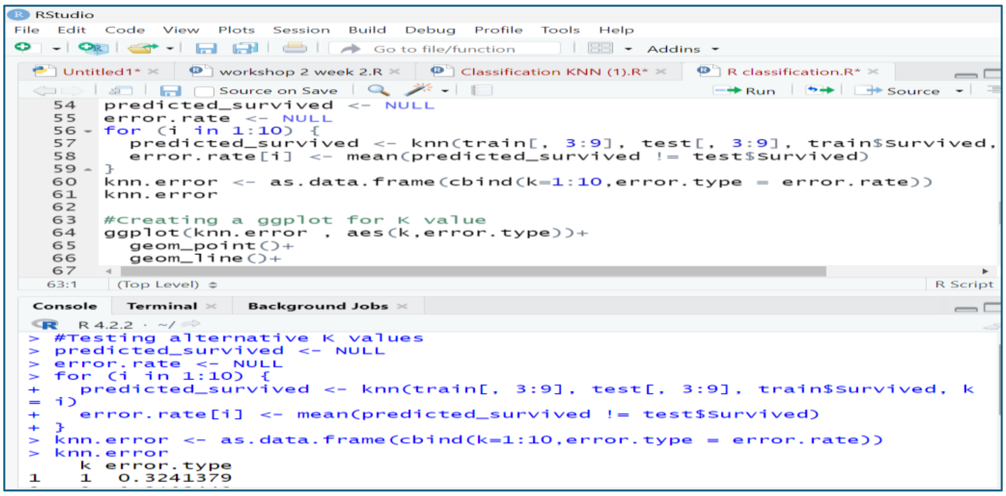


Figure 14

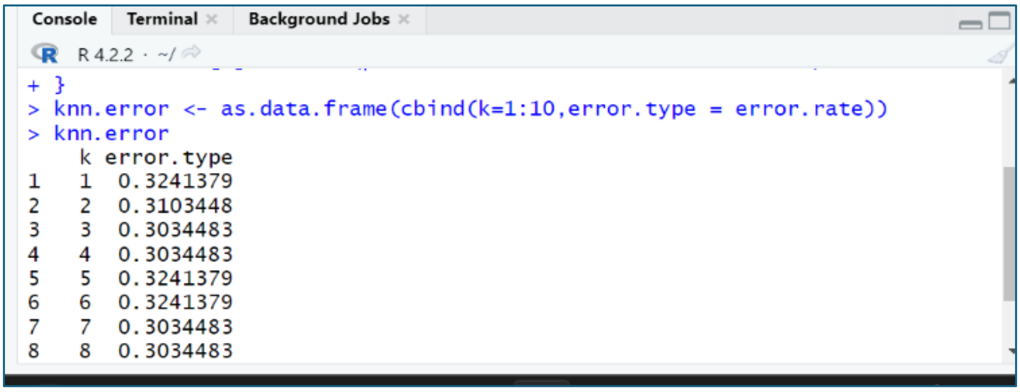


Figure 15

3.6 Choosing K Value by Visualization

Next, what is the best value that we can determine to apply for K. Before training the model we must determine and apply the minimum K value. Let's plot the chart using the 'ggplot2' library.

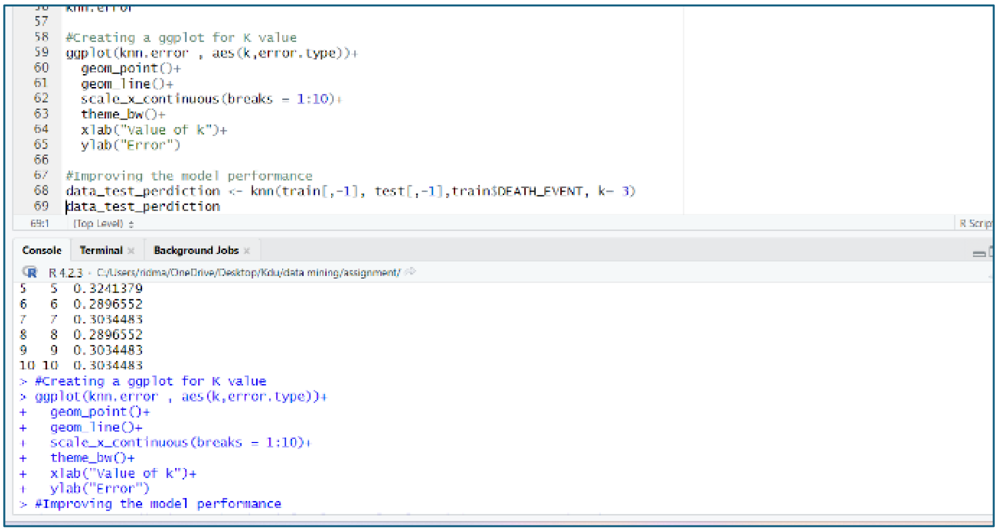
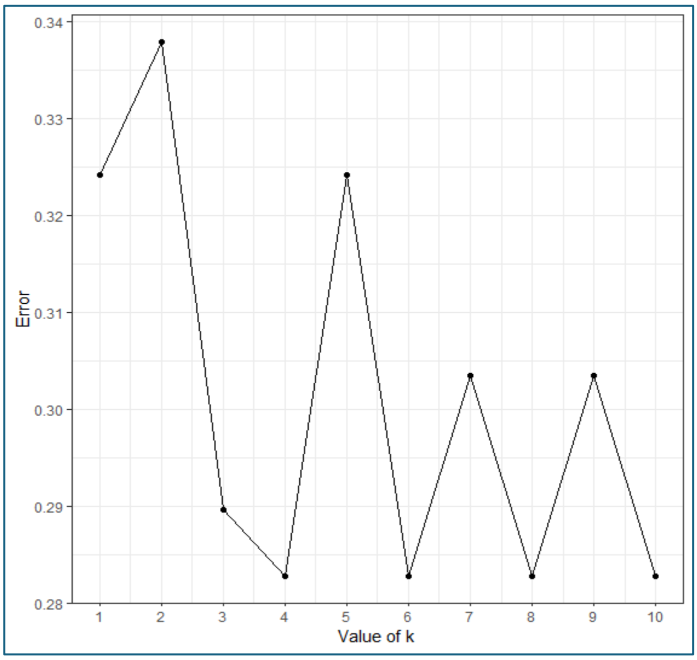


Figure 12



Based on the plot, we can observe that the error rate is lowest when k=3. This suggests that a k-nearest neighbors model with k=3 might be a good choice for predicting the Survived variable in the Titanic data set.

Result

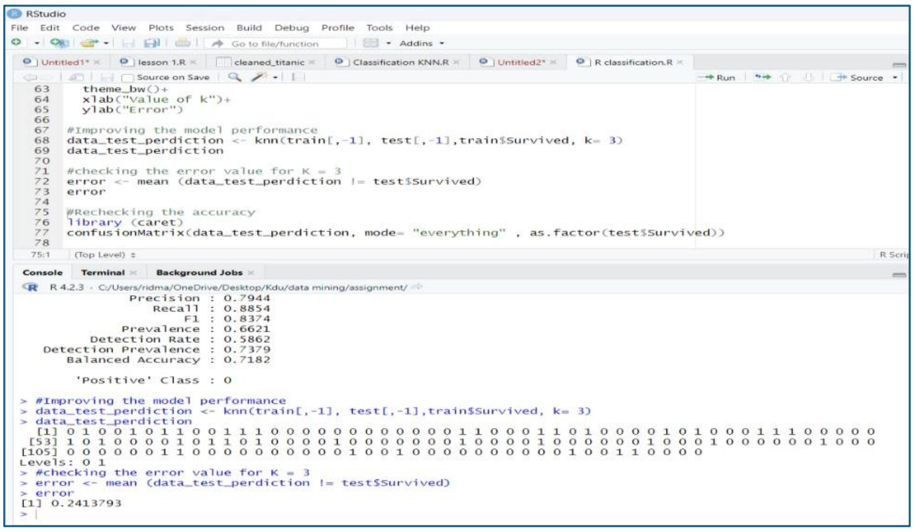
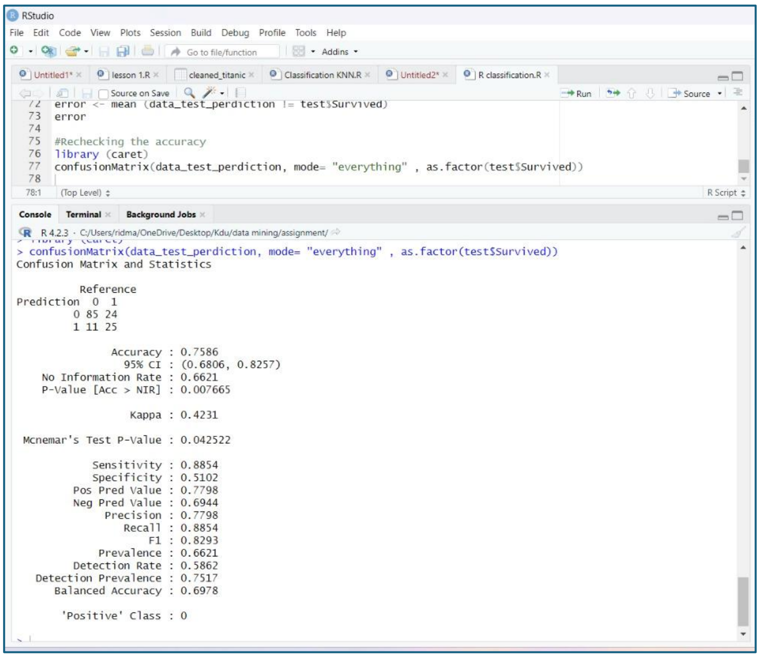


Figure 13



As the confusion matrix says, our model has obtained 75.86% accuracy.

5. Result analysis and discussion

The result of our model can be evaluated by using a confusion matrix. The confusion matrix shows the incidence of the model as follows.

- True Positive (TP): The number of instances correctly classified as survivors. value is the number of instances where the actual class is 1 and the predicted class is also 1. In this case, **TP = 27.**

- True Negative (TN): The number of instances correctly classified as non-survivors. value is the number of instances where the actual class is 0 and the predicted class is also 0. In this case**, TN = 85.**

- False Positive (FP): The number of instances incorrectly classified as survivors when they were non-survivors. value is the number of instances where the actual class is 0 but the predicted class is 1. In this case, **FP = 22**

- False Negative (FN): The number of instances incorrectly classified as non-survivors when they were survivor. value is the number of instances where the actual class is 1 but the predicted class is 0. In this case, **FN = 11.**

6. Conclusion

our analysis of the Titanic dataset utilizing the K-Nearest Neighbors (KNN) algorithm provides valuable insights into the predictive capabilities of machine learning in the context of historical events. Through meticulous data preprocessing, feature selection, and model training, we were able to successfully develop a KNN classifier capable of predicting survival outcomes with a commendable accuracy rate of 75.86%.

Our study underscores the significance of leveraging advanced analytics techniques to extract meaningful patterns from complex datasets, such as the Titanic dataset. By understanding the relationships between passenger attributes and survival probabilities, we gain valuable insights into the dynamics of historical events like the Titanic disaster.

Moving forward, this analysis sets the stage for further exploration and refinement of machine learning models in predictive analytics. Future research could delve deeper into feature engineering, model optimization, and ensemble methods to enhance predictive accuracy and robustness.our findings not only contribute to the broader field of data science and machine learning but also shed light on the potential applications of predictive analytics in historical analysis and risk assessment scenarios. As we continue to harness the power of data-driven approaches, we empower decision-makers with actionable insights to mitigate risks and shape a more resilient future.

7. References

- **https://data.world/nrippner/titanic-disaster-dataset**

TASK 02

Clustering to the data set of forest fires, in the northeast region of Portugal

Table of contents

1. Introduction

The Forest Fires dataset available at the UCI Machine Learning Repository is widely used in the field of environmental science and machine learning. This dataset contains information regarding forest fires in the northeast region of Portugal. The dataset provides various attributes related to the environmental conditions and factors that might influence the occurrence and behavior of forest fires. Some of the key attributes include meteorological data (such as temperature, relative humidity, wind speed, and direction), as well as spatial data (such as the coordinates and area). Additionally, it includes information on the day of the week and month when the fire occurred.

The primary objective of this dataset is to facilitate research and analysis aimed at understanding the patterns and drivers of forest fires. Researchers and practitioners often utilize this dataset to develop predictive models that can help in forecasting and mitigating the risk of forest fires, as well as to gain insights into the relationships between environmental variables and fire occurrence.

1.1 Data Set

This forest fire data set is taken from the UCI Machine Learning Repository (https://archive.ics.uci.edu/dataset/162/forest+fires). This data set is a popular dataset for clustering. It contains information about 512 forest fires in Portugal. There are 9 variables in this dataset such as temperature, humidity, wind speed, and the area of the burned area.

A1 X

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	X	Y	month	day	FFMC	DMC	DC	ISI	temp	RH	wind	rain	area	
2	7	5	mar	fri	86.2	26.2	94.3	5.1	8.2	51	6.7	0	0	
3	7	4	oct	tue	90.6	35.4	669.1	6.7	18	33	0.9	0	0	
4	7	4	oct	sat	90.6	43.7	686.9	6.7	14.6	33	1.3	0	0	
5	8	6	mar	fri	91.7	33.3	77.5	9	8.3	97	4	0.2	0	
6	8	6	mar	sun	89.3	51.3	102.2	9.6	11.4	99	1.8	0	0	
7	8	6	aug	sun	92.3	85.3	488	14.7	22.2	29	5.4	0	0	
8	8	6	aug	mon	92.3	88.9	495.6	8.5	24.1	27	3.1	0	0	
9	8	6	aug	mon	91.5	145.4	608.2	10.7	8	86	2.2	0	0	
10	8	6	sep	tue	91	129.5	692.6	7	13.1	63	5.4	0	0	
11	7	5	sep	sat	92.5	88	698.6	7.1	22.8	40	4	0	0	
12	7	5	sep	sat	92.5	88	698.6	7.1	17.8	51	7.2	0	0	
13	7	5	sep	sat	92.8	73.2	713	22.6	19.3	38	4	0	0	
14	6	5	aug	fri	63.5	70.8	665.3	0.8	17	72	6.7	0	0	
15	6	5	sep	mon	90.9	126.5	686.5	7	21.3	42	2.2	0	0	
16	6	5	sep	wed	92.9	133.3	699.6	9.2	26.4	21	4.5	0	0	
17	6	5	sep	fri	93.3	141.2	713.9	13.9	22.9	44	5.4	0	0	
18	5	5	mar	sat	91.7	35.8	80.8	7.8	15.1	27	5.4	0	0	
19	8	5	oct	mon	84.9	32.8	664.2	3	16.7	47	4.9	0	0	
20	6	4	mar	wed	89.2	27.9	70.8	6.3	15.9	35	4	0	0	
21	6	4	apr	sat	86.3	27.4	97.1	5.1	9.3	44	4.5	0	0	
22	6	4	sep	tue	91	129.5	692.6	7	18.3	40	2.7	0	0	
23	5	4	sep	mon	91.8	78.5	724.3	9.2	19.1	38	2.7	0	0	
24	7	4	jun	sun	94.3	96.3	200	56.1	21	44	4.5	0	0	

forestfires +

2. Explanation and Preparation of Dataset

2.1 Data Processing

There were null values in our data set.so we cleaned the data set and finally the data Set was prepared as follows.

	month	FFMC	DMC	DC	ISI	temp	RH	wind	area
2	aug	95.5	99.9	513.3	13.2	23.3	31	4.5	1
3	aug	90.1	108	529.8	12.5	21.2	51	8.9	1
4	jul	90	51.3	296.3	8.7	16.6	53	5.4	1
5	aug	95.5	99.9	513.3	13.2	23.8	32	5.4	1
6	aug	95.2	131.7	578.8	10.4	27.4	22	4	1
7	mar	90.1	39.7	86.6	6.2	13.2	40	5.4	1
8	sep	84.4	73.4	671.9	3.2	24.2	28	3.6	1
9	aug	94.8	108.3	647.1	17	17.4	43	6.7	1
10	sep	93.7	80.9	685.2	17.9	23.7	25	4.5	1
11	jun	92.5	56.4	433.3	7.1	23.2	39	5.4	1
12	jul	90.1	68.6	355.2	7.2	24.8	29	2.2	1
13	jul	90.1	51.2	424.1	6.2	24.6	43	1.8	1
14	sep	94.3	85.1	692.3	15.9	20.1	47	4.9	1
15	sep	93.4	145.4	721.4	8.1	29.6	27	2.7	1
16	aug	94.8	108.3	647.1	17	16.4	47	1.3	2
17	sep	93.4	145.4	721.4	8.1	28.6	27	2.2	2
18	aug	92.1	111.2	654.1	9.6	18.4	45	3.6	2
19	aug	92.1	111.2	654.1	9.6	20.5	35	4	2
20	sep	92.4	117.9	668	12.2	19	34	5.8	2
21	mar	90.1	39.7	86.6	6.2	16.1	29	3.1	2
22	aug	95.2	131.7	578.8	10.4	20.3	41	4	2
23	mar	90.6	50.1	100.4	7.8	15.2	31	8.5	2
24	sep	92.5	121.1	674.4	8.6	17.8	56	1.8	2
25	sep	89.7	90	704.4	4.8	17.8	67	2.2	2

Figure 2

2.2 Data explanation

- ✓ month - month of the year: 'Jan' to 'Dec'
- ✓ FFMC - FFMC index from the FWI system: 18.7 to 96.20
- ✓ DMC - DMC index from the FWI system: 1.1 to 291.3
- ✓ DC - DC index from the FWI system: 7.9 to 860.6
- ✓ ISI - ISI index from the FWI system: 0.0 to 56.10
- ✓ temp - temperature in Celsius degrees: 2.2 to 33.30.
- ✓ RH - relative humidity in %: 15.0 to 100
- ✓ wind - wind speed in km/h: 0.40 to 9.40.
- ✓ rain - outside rain in mm/m2: 0.0 to 6.4
- ✓ area - the burned area of the forest (in ha): 0.00 to 1090.84

3.Data Mining

3.1 Clustering

Clustering refers to a set of techniques used to group similar data points together based on certain features or characteristics. Clustering is an unsupervised learning technique, meaning that it doesn't require labeled data; instead, it seeks to identify inherent patterns or structures within the dataset itself.

3.2 K- means clustering.

K-means clustering is one of the most widely used unsupervised machine learning algorithms for clustering data points into groups, or clusters, based on their similarity. The algorithm aims to partition a dataset into K clusters, where each data point belongs to the cluster with the nearest mean, or centroid.

K-means clustering is sensitive to the initial positions of the centroids, which can affect the final clustering result. Therefore, the algorithm is often run multiple times with different initializations, and the clustering result with the lowest overall within-cluster variation is selected.

4. Implementation in R

4.1 R Packages.

- **Cluster package** - The cluster package in R is a comprehensive package for clustering, classification, and visualization of high-dimensional data. It provides a wide range of clustering algorithms, including hierarchical clustering, k-means clustering, and DBSCAN. The package also includes functions for visualizing clusters and evaluating the performance of clustering algorithms.

- **Factoextra package-** This package provides additional functions for factor analysis, visualization, and evaluation. The package includes functions for performing advanced factor analysis techniques, such as oblique rotation and parallel analysis, as well as for visualizing and evaluating the results of factor analysis.

4.2 Implementation

Step 1

Set as working directory

```
> #setwd
> mypath="C:/Users/ASUS/Desktop/fire/forestfire.xlsx"
> setwd("C:/Users/ASUS/Desktop/fire")
> getwd()
[1] "C:/Users/ASUS/Desktop/fire"
> |
```

Step 2

Read the data set as follows

```
> #Read the data file
> library(readxl)
> forest_fire <- read_excel("forestfire.xlsx")
> forest_fire
# A tibble: 100 × 9
   month  FFMC   DMC    DC   ISI  temp    RH  wind  area
   <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
 1 aug    95.5  99.9 513.   13.2  23.3    31   4.5  0.55
 2 aug    90.1 108   530.   12.5  21.2    51   8.9  0.61
 3 jul    90    51.3 296.    8.7  16.6    53   5.4  0.71
 4 aug    95.5  99.9 513.   13.2  23.8    32   5.4  0.77
 5 aug    95.2 132.  579.   10.4  27.4    22   4    0.9
 6 mar    90.1  39.7  86.6   6.2  13.2    40   5.4  0.95
 7 sep    84.4  73.4 672.    3.2  24.2    28   3.6  0.96
 8 aug    94.8 108.  647.   17    17.4    43   6.7  1.07
 9 sep    93.7  80.9 685.   17.9  23.7    25   4.5  1.12
10 jun    92.5  56.4 433.    7.1  23.2    39   5.4  1.19
# ℹ 90 more rows
# ℹ Use `print(n = ...)` to see more rows
> |
```

Step 3

Explore the dataset.

```
> #Inspect the dataset in R
> names(forest_fire)
[1] "month" "FFMC"  "DMC"   "DC"    "ISI"   "temp"  "RH"    "wind"  "area"
> |
```

Using the **head()**function, we can get 1st six rows in data set

```
> head(forest_fire)
# A tibble: 6 x 9
  month  FFMC   DMC    DC   ISI  temp    RH  wind  area
  <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1 aug    95.5  99.9 513.   13.2  23.3    31   4.5  0.55
2 aug    90.1 108   530.   12.5  21.2    51   8.9  0.61
3 jul    90    51.3 296.    8.7  16.6    53   5.4  0.71
4 aug    95.5  99.9 513.   13.2  23.8    32   5.4  0.77
5 aug    95.2 132.  579.   10.4  27.4    22   4    0.9
6 mar    90.1  39.7  86.6   6.2  13.2    40   5.4  0.95
> |
```

From the code summary () inspect the summary of the dataset.

```
> summary(forest_fire)
    month                FFMC            DMC               DC             ISI
 Length:100         Min.   :18.70   Min.   :  1.10   Min.   : 25.6   Min.   : 0.00
 Class :character   1st Qu.:90.10   1st Qu.: 48.50   1st Qu.:306.9   1st Qu.: 6.30
 Mode  :character   Median :91.40   Median : 96.90   Median :619.4   Median : 8.50
                    Mean   :90.31   Mean   : 88.78   Mean   :494.4   Mean   : 9.03
                    3rd Qu.:93.00   3rd Qu.:129.50   3rd Qu.:685.5   3rd Qu.:11.10
                    Max.   :96.00   Max.   :150.30   Max.   :735.7   Max.   :20.30
      temp             RH              wind            area
 Min.   : 4.60   Min.   : 19.00   Min.   :0.900   Min.   :   0.000
 1st Qu.:15.18   1st Qu.: 31.00   1st Qu.:2.700   1st Qu.:   2.107
 Median :18.50   Median : 39.00   Median :4.000   Median :   7.820
 Mean   :18.08   Mean   : 41.71   Mean   :4.149   Mean   :  34.060
 3rd Qu.:22.30   3rd Qu.: 47.50   3rd Qu.:5.400   3rd Qu.:  28.660
 Max.   :29.60   Max.   :100.00   Max.   :9.400   Max.   :1090.840
> |
```

Step 4-

Check the dimensions of the data set.

```
> nrow(forest_fire)
[1] 100
> ncol(forest_fire)
[1] 9
> dim(forest_fire)
[1] 100   9
> |
```

Step- 5

Install “cluster” and “factoextra ” packages

```
> #Install and activate “cluster” package.
> install.packages("cluster")
Error in install.packages : Updating loaded packages
> install.packages("cluster")
WARNING: Rtools is required to build R packages but is not currently installed. Please download and instal
l the appropriate version of Rtools before proceeding:

https://cran.rstudio.com/bin/windows/Rtools/
Warning in install.packages :
  package ‘cluster’ is in use and will not be installed
> library(cluster)
```

```
> install.packages("factoextra") # install “factoextra” package
Error in install.packages : Updating loaded packages
> install.packages("factoextra")
WARNING: Rtools is required to build R packages but is not currently installed. Please download and instal
l the appropriate version of Rtools before proceeding:

https://cran.rstudio.com/bin/windows/Rtools/
Warning in install.packages :
  package ‘factoextra’ is in use and will not be installed
> library(factoextra)
> |
```

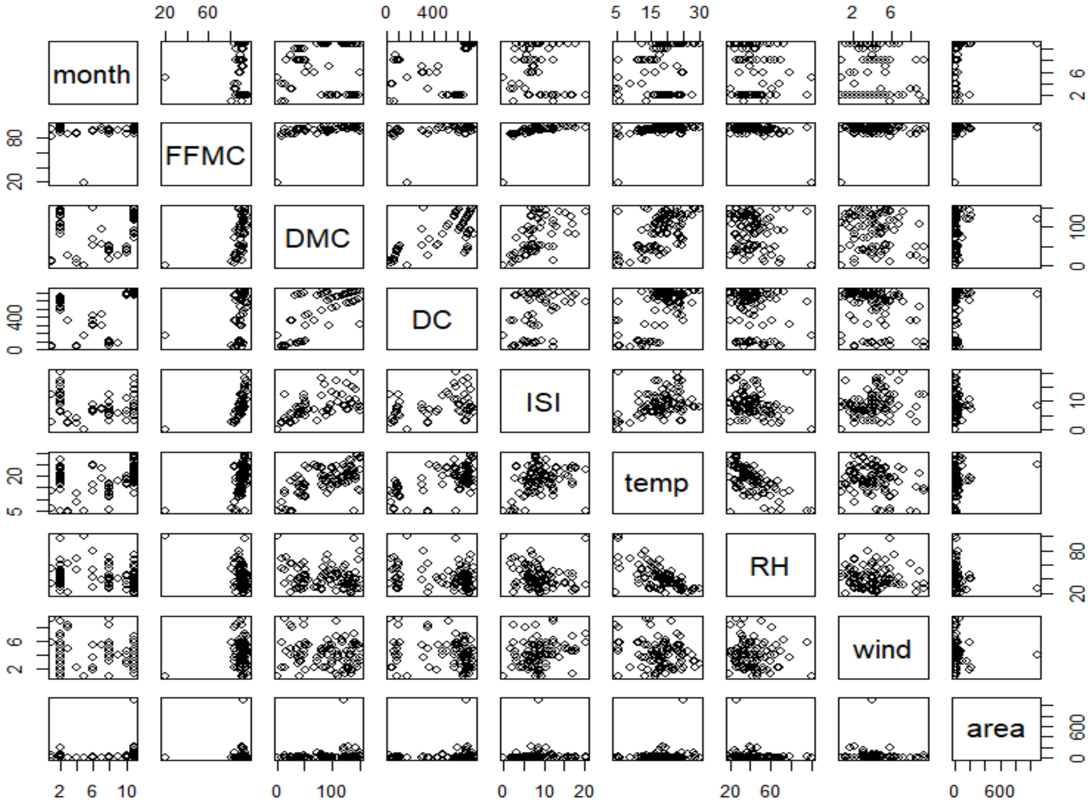
Step 6

Create a scatterplot matrix to understand the variables.

This is the r code for the scatterplot matrix. Pairs () If the dataset includes non-numeric arguments, you need to remove those columns before running the code.

```
> #create scatterplot matrix to compare the variables
> forest_fire$month<-as.factor(forest_fire$month)
> pairs(forest_fire)
>
```

After running this code can see a scatterplot like below. Numeric variables are compared by using this plot.



Step 7

Use the following line of code to plot and understand the relationship between the temperature and wind.

```
> plot(temp ~ wind, data =forest_fire)
> with(forest_fire,text(temp~wind, labels= month,pos=4,cex=.6))
>
```

Result-

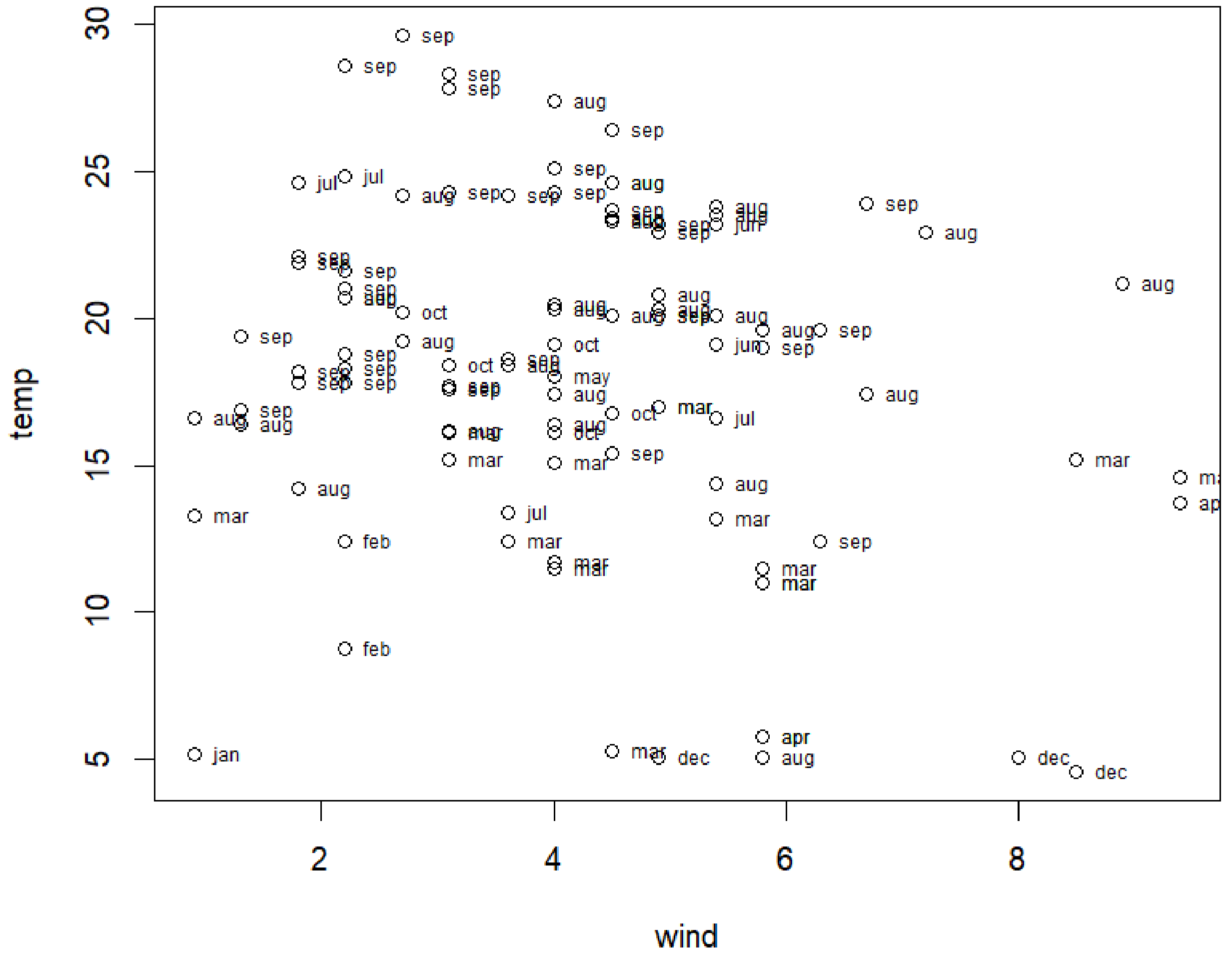


Figure 4.2-plot

Step 8

Scale the data

```
> #Scale the data
> forest_fire_new=forest_fire[,2:9]
> forest_scaled=scale(forest_fire_new)
> forest_scaled
             FFMC         DMC          DC          ISI        temp          RH        wind
 [1,]  0.65964920  0.25620965  0.07648712  0.990938248  0.90336434 -0.69545825  0.18571639
 [2,] -0.02603043  0.44280363  0.14317338  0.824593698  0.53980261  0.60324996  2.51378505
 [3,] -0.03872820 -0.86335420 -0.80053825 -0.078419574 -0.25657071  0.73312079  0.66191225
 [4,]  0.65964920  0.25620965  0.07648712  0.990938248  0.98992665 -0.63052284  0.66191225
 [5,]  0.62155589  0.98876378  0.34121137  0.325560048  1.61317533 -1.27987694 -0.07883687
 [6,] -0.02603043 -1.13057521 -1.64806000 -0.672507252 -0.84519446 -0.11103955  0.66191225
 [7,] -0.74980337 -0.35425212  0.71748354 -1.385412467  1.05917651 -0.89026448 -0.29047948
 [8,]  0.57076480  0.44971452  0.61725207  1.893951519 -0.11807100  0.08376668  1.34975072
 [9,]  0.43108932 -0.18147992  0.77123671  2.107823084  0.97261419 -1.08507071  0.18571639
[10,]  0.27871607 -0.74586911 -0.24684021 -0.458635688  0.88605187 -0.17597496  0.66191225
[11,] -0.02603043 -0.46482633 -0.56248851 -0.434872181  1.16305129 -0.82532907 -1.03122860
[12,] -0.02603043 -0.86565783 -0.28402285 -0.672507252  1.12842636  0.08376668 -1.24287120
[13,]  0.50727595 -0.08472749  0.79993201  1.632552941  0.34936551  0.34350832  0.39735899
[14,]  0.39299601  1.30436100  0.91754233 -0.221000616  1.99404953 -0.95519989 -0.76667534
[15,]  0.57076480  0.44971452  0.61725207  1.893951519 -0.29119563  0.34350832 -1.50742446
[16,]  0.39299601  1.30436100  0.91754233 -0.221000616  1.82092489 -0.95519989 -1.03122860
[17,]  0.22792499  0.51651977  0.64554321  0.135451991  0.05505363  0.21363750 -0.29047948
[18,]  0.22792499  0.51651977  0.64554321  0.135451991  0.41861536 -0.43571661 -0.07883687
[19,]  0.26601830  0.67086294  0.70172134  0.753303176  0.15892841 -0.50065202  0.87355485
[20,] -0.02603043 -1.13057521 -1.64806000 -0.672507252 -0.34313302 -0.82532907 -0.55503273
```

Step 9

Calculate the distance matrix using "Euclidean" method

After scaling the data, we can run the following code to get the distance matrix. Below distance matrix below shows the numbers rounded to 3 decimals.

```
> print(distance)
            1          2          3          4          5          6          7          8
2   29.836120
3  237.244776 254.975756
4    1.540114  28.999807 237.157314
5   77.997133  66.209702 313.532252  78.117869
6  457.388187 475.914077 223.231872 457.378856 531.703022
7  171.322748 157.588413 400.125446 171.351140 117.543849 622.112690
8  142.980179 124.983451 377.249299 142.900765  80.808725 599.093019  52.096796
9  183.627280 169.796953 414.951472 183.668500 125.423181 636.842607  24.779655  53.800909
10  97.232458 117.011136 146.369606  97.141628 174.823067 368.321644 254.171609 233.727472
11 171.199410 191.530742  70.553138 171.226981 246.644046 287.061240 336.036661 313.156973
            9         10         11         12         13         14         15         16
```

Step 10

Visualize the distance matrix.

```
> fviz_dist(distance)
>
```

After running the above code can get the results as follow,

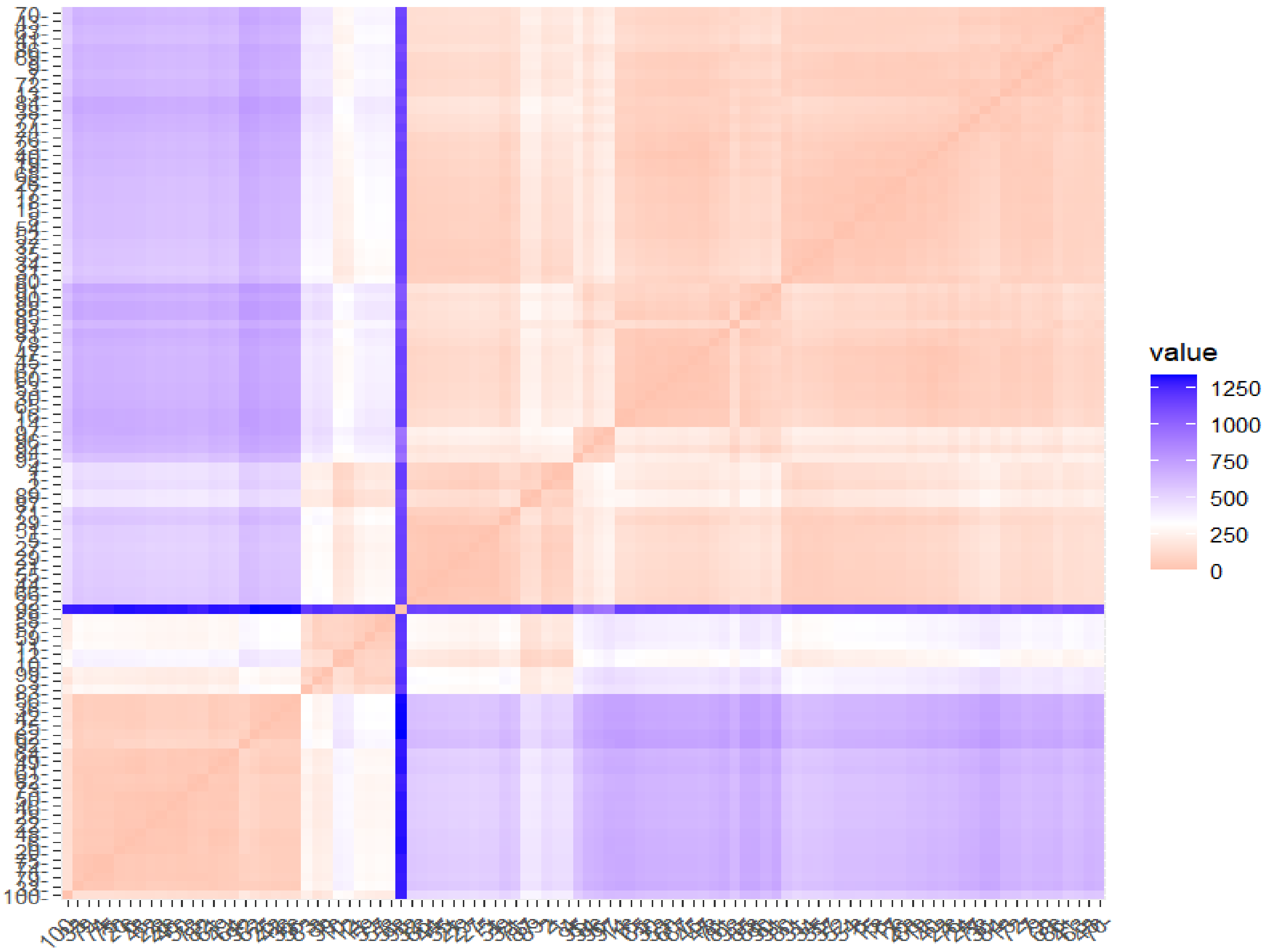


Figure 4.3

Step 11

Determining the optimal number of clusters

- The elbow method- The elbow method is a technique used in cluster analysis to determine the optimal number of clusters.

```
> fviz_nbclust(forest_scaled,kmeans,method = "wss")
>
```

Result-

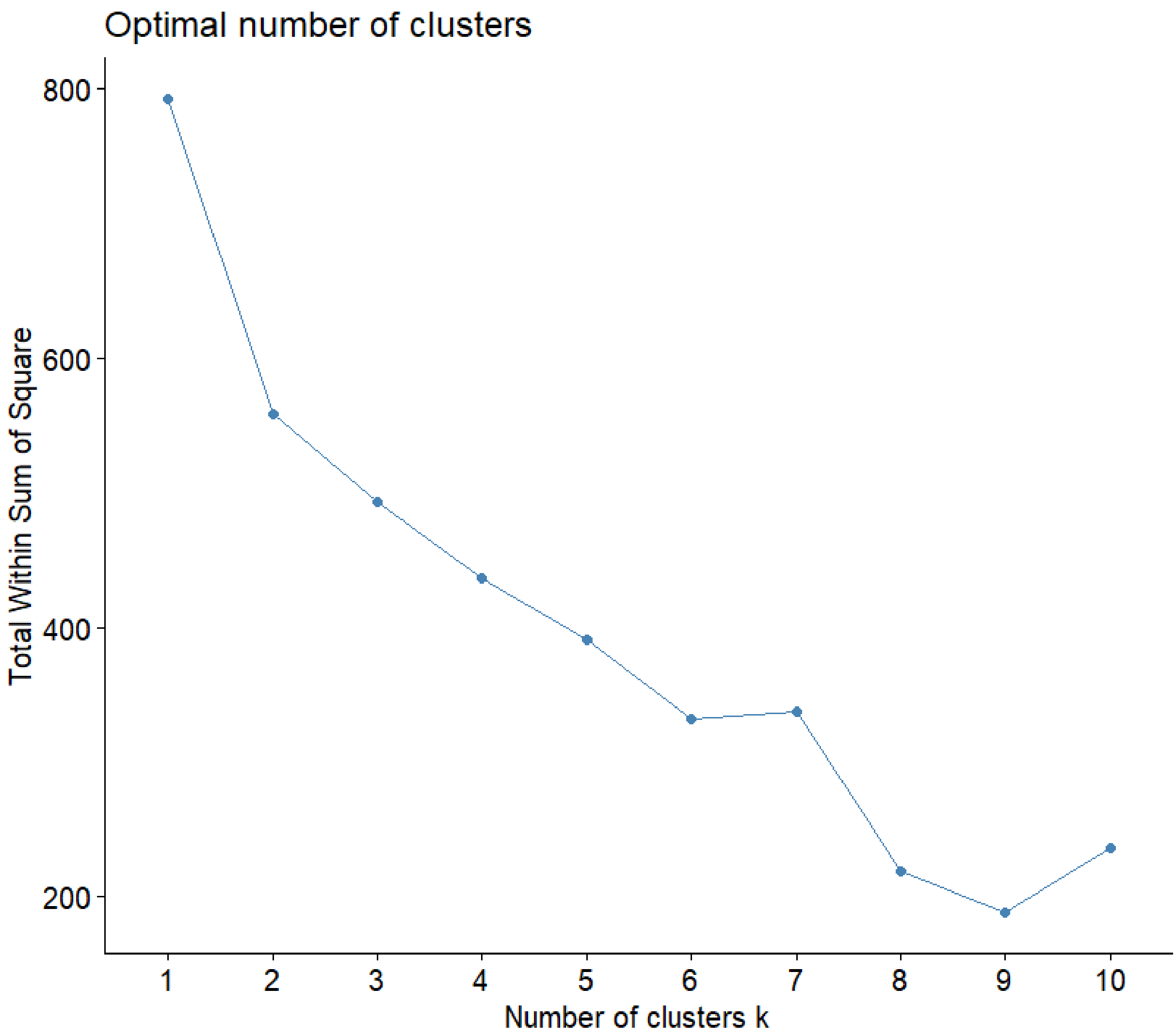


Figure4.4

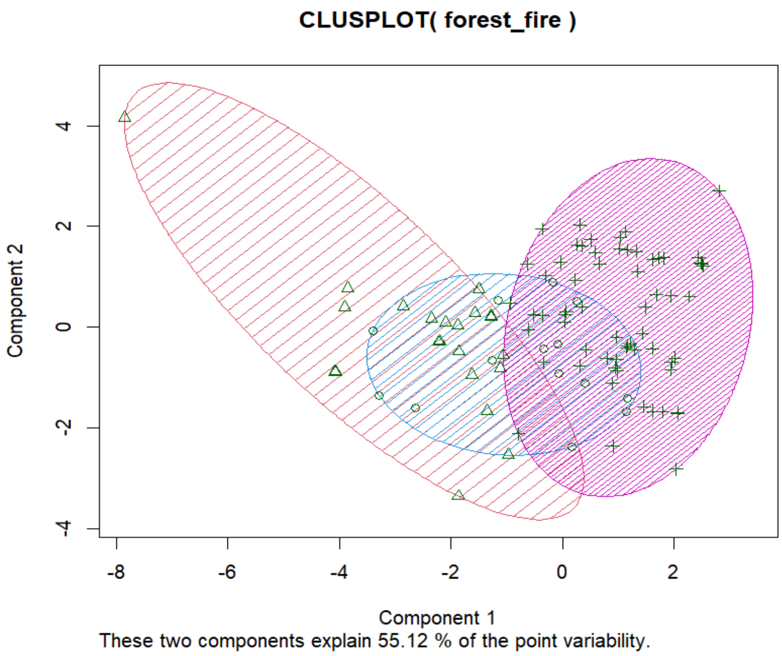
Step 11

K-mean clustering

```
> kc<-kmeans(forest_fire[,-1],3) #k=3
> kc
K-means clustering with 3 clusters of sizes 14, 23, 63

Cluster means:
      FFMC       DMC        DC       ISI     temp       RH     wind     area
1 90.06429  73.00714 406.51429  8.292857 17.23571 44.00000 5.178571 15.08857
2 85.70870  33.41304  83.02609  6.408696 12.20870 46.21739 4.695652 15.38652
3 92.03651 112.49524 664.07460 10.150794 20.41429 39.55556 3.720635 45.09317

Clustering vector:
  [1] 1 1 1 1 3 2 3 3 3 1 1 1 3 3 3 3 3 3 3 2 3 2 3 3 2 3 3 2 3 3 3 3 2 3 3 2 3 3 3 3 3 2 3 3 3 2 3
 [48] 2 2 2 3 3 3 3 3 2 1 1 1 3 2 2 3 2 3 3 3 3 3 3 3 3 2 2 2 3 3 3 2 3 3 2 1 3 3 3 1 3 1 3 3 2 3 3
 [95] 3 3 3 3 1 2

Within cluster sum of squares by cluster:
[1]  123210.2   44168.2 1459540.0
 (between_SS / total_SS =  78.5 %)

Available components:

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

- The result can be plotted using the clusplot () function.

```
> clusplot(forest_fire, kc$cluster, color=TRUE, shade=TRUE, lines=0)
>
```



Step 12

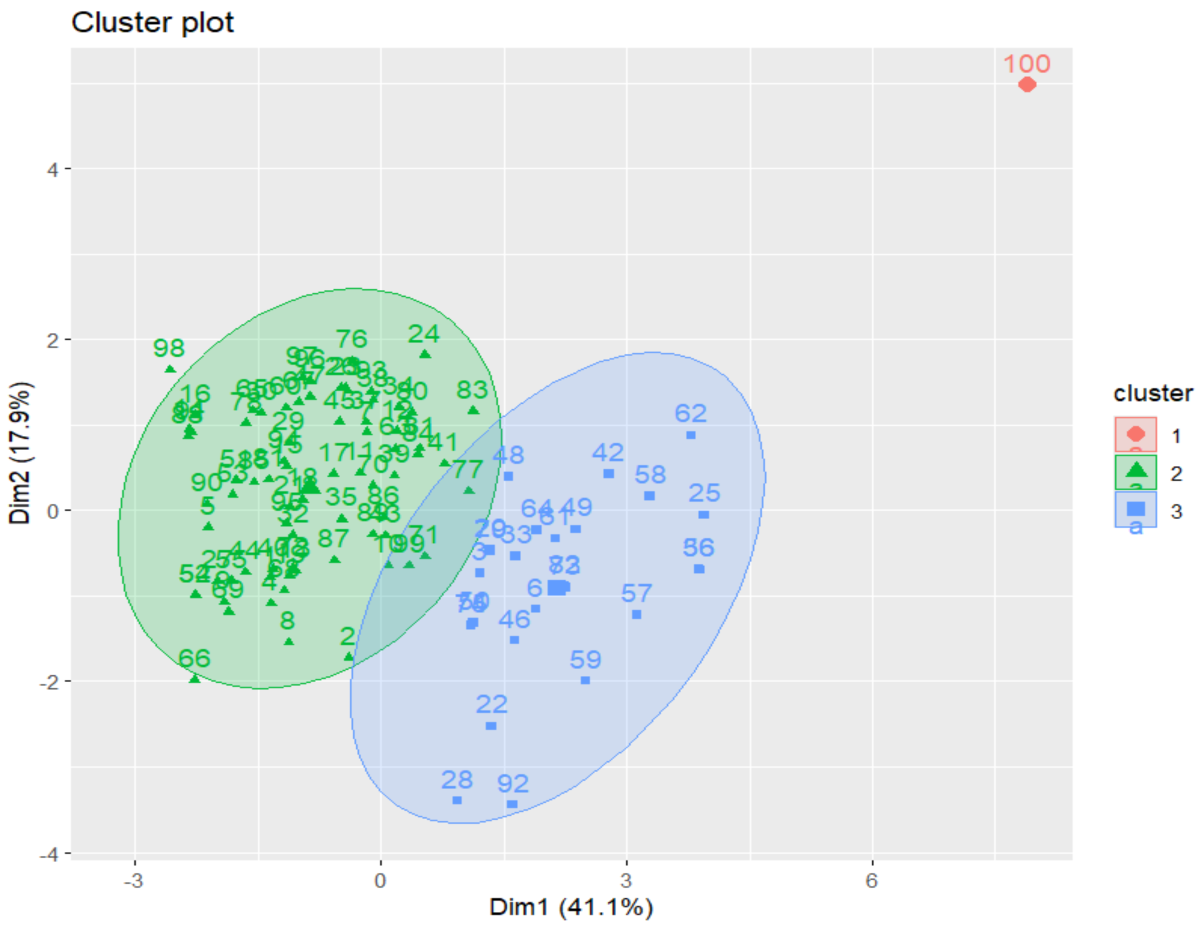
Perform k-means clustering on a forest_fire dataset with k=3

```
> set.seed(123)
> km.fit=kmeans(forest_scaled,3,nstart = 30)
> km.fit$cluster
  [1] 2 2 3 2 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2 3 2 3 2 2 3 2 2 3 2 2 2 2 3 2 2 3 2 2 2 2 2 3 2 2 2 3 2
 [48] 3 3 3 2 2 2 2 2 3 3 3 3 2 3 3 2 3 2 2 2 2 2 2 2 2 3 3 3 2 2 2 3 2 2 3 2 2 2 2 2 2 2 2 2 3 2 2
 [95] 2 2 2 2 2 1
> km.fit$size
[1]  1 73 26
> km.fit$centers
        FFMC       DMC         DC        ISI       temp         RH       wind        area
1 -9.0922389 -2.019776 -1.3053330 -2.1458447 -2.2301915  3.7850851 -1.7190671 -0.29817775
2  0.2138357  0.472593  0.5583999  0.2532929  0.4174296 -0.1181558 -0.1650885  0.06368228
3 -0.2506833 -1.249212 -1.5176099 -0.6286362 -1.0862372  0.1861648  0.5296356 -0.16733187
> |
```

Step 13

Visualize clusters using the **fviz_cluster ()** function in factoextra package

Result-



5. Findings and Conclusions

Two clusters are being plotted on a graph with two dimensions, Dim1 and Dim2, where Dim1 represents 41.1% of the variance and Dim2 represents 17.9% of the variance.

Here are some possible findings and conclusions based on the given cluster plot:

- There are two distinct clusters in the data, with one cluster having a higher value for Dim1 compared to the other cluster.
- The cluster with higher values for Dim1 also has a wider range of values for Dim2 compared to the other cluster.
- The cluster with lower values for Dim1 has a more concentrated range of values for Dim2, with most data points falling between -2 and 2 on the Dim2 axis.
- The data point with the highest value for Dim2 (90314218.09) appears to be an outlier and does not belong to either of the two clusters.
- The data points in the cluster with higher values for Dim1 are more evenly distributed along the Dim2 axis compared to the other cluster.
- The cluster with lower values for Dim1 has a higher density of data points in the lower range of values for Dim2.

6. References

https://archive.ics.uci.edu/dataset/162/forest+fires

Task 03

Power BI dashboard for Forest Fires Analyze

1. Introduction

In an era where climate change poses increasingly urgent challenges, understanding the dynamics of natural disasters such as forest fires is paramount. Forest fires not only endanger lives and ecosystems but also present significant economic and environmental impacts. To address these challenges, we present a comprehensive analysis leveraging data from the UCI Machine Learning Repository's forest fire dataset. This report aims to provide insights into forest fire occurrences through advanced visualization techniques and cluster analysis. By exploring temporal trends, spatial distribution, and predictive modeling, we seek to uncover underlying patterns and inform proactive strategies for forest fire management and mitigation. Through this analysis, we endeavor to contribute to the collective effort in safeguarding our forests and communities from the devastating effects of forest fires.

2. Exploration of Dataset

This is a very challenging regression task, in which the use of weather and meteorological data will be used to forecast forest fire burn areas in Portugal's northeast.

We used the same named “forest fire” data set that we used for the clustering. It includes 100 rows & 9 columns. For our dashboard visualization, we use 512 rows and 9 columns.

Attributes of the dataset are.

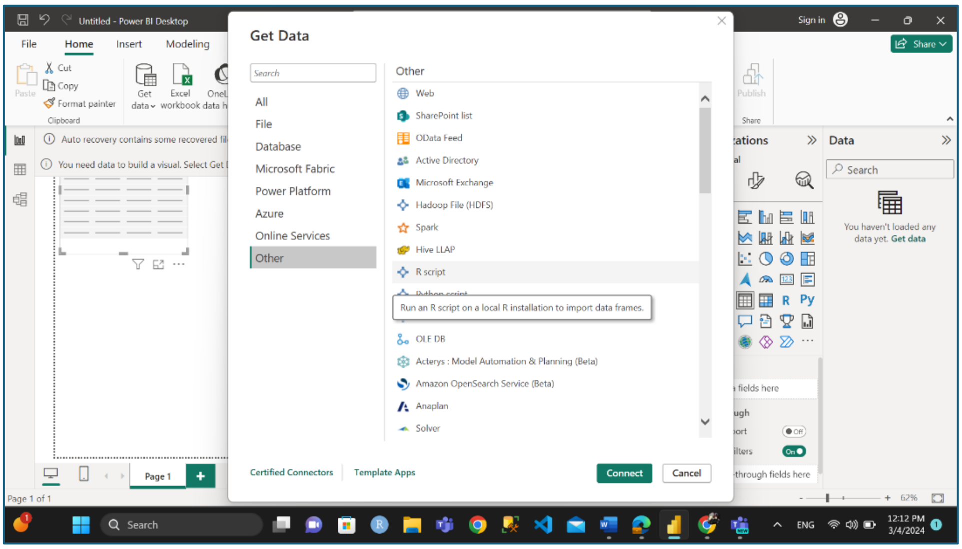
- month - month of the year: 'Jan' to 'Dec'
- FFMC - FFMC index from the FWI system: 18.7 to 96.20
- DMC - DMC index from the FWI system: 1.1 to 291.3
- DC - DC index from the FWI system: 7.9 to 860.6
- ISI - ISI index from the FWI system: 0.0 to 56.10
- temp - temperature in Celsius degrees: 2.2 to 33.30
- RH - relative humidity in %: 15.0 to 100
- wind - wind speed in km/h: 0.40 to 9.40
- rain - outside rain in mm/m2: 0.0 to 6.4

3. Dashboard Design & Implementation

A dashboard is a type of graphical user interface that often provides at glance view of key performance indicators relevant to specific objectives and business processes. In other words, "dashboard" is another name for a progress report or reporting object and it's considered to be an example of data visualization.

Before importing the data set, we used a different way to start the task,

After entering the R codes that we used in clustering with the working directory, power Bi automatically generated two normal and normalized data sets in the field section.

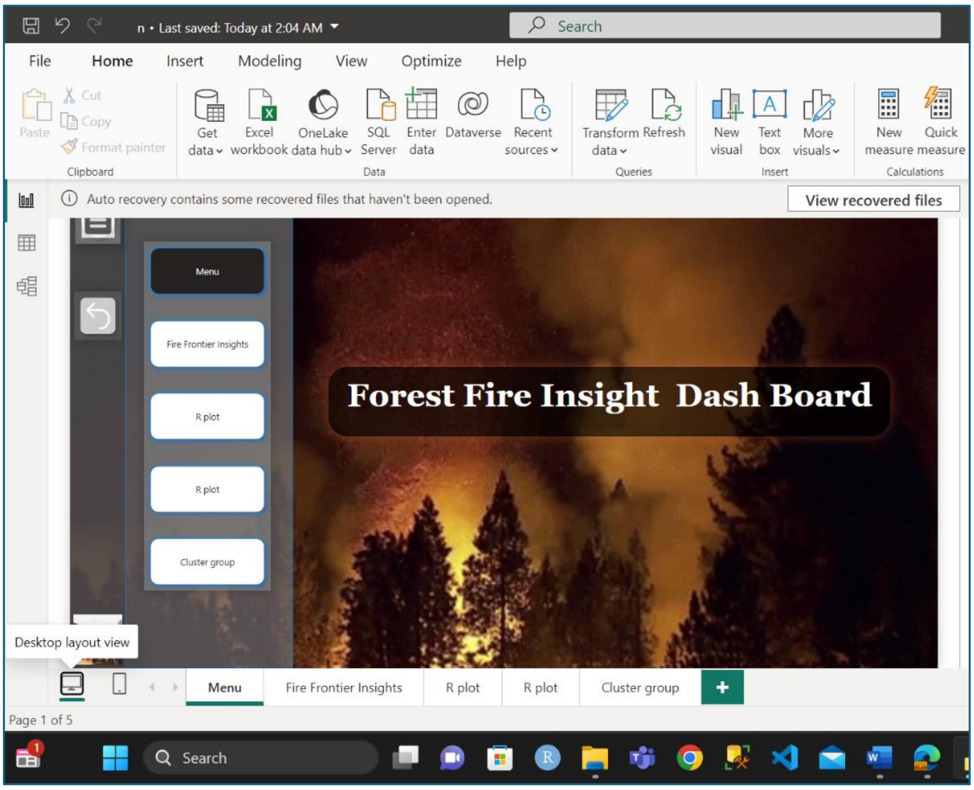


3.1 Dashboard implementation

We used the R script to import data into Power BI and loaded our main dataset and other data that we gathered to make better decisions (wind, temp, RH). And the raw dataset that we filtered our new data because some data columns are still useful for visualization purposes. After importing the data, we could see the field pane as follows.

Our Dashboard includes 5 pages.

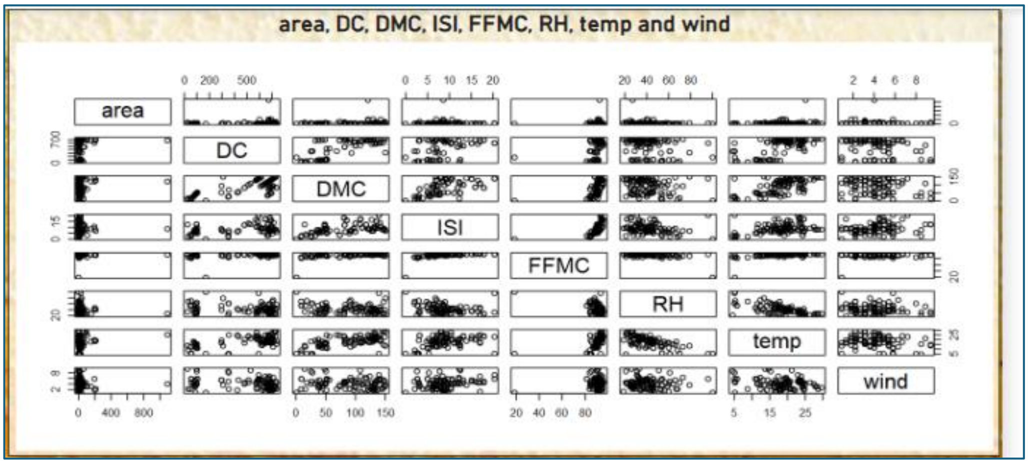
- Main menu=insight about our dashboard
- (2nd page) Forest fire visualization
- 3rd page and 4th page plot
- 5th page=Describe how we cluster our data set



3.2 Data pairs visualization – Scatterplot Matrix

We used a scatter plot matrix to display this chart named “Pairs of Attributes of forest fire”. A scatter plot matrix can reveal the relationships between numerous variables. the matrix can reveal the relationship between variables after charting all the two-way combinations of the variables to emphasize which associations are likely to be essential. Outliers in several scatter plots can also be detected using the matrix. In this data set scatter plot matrix is used to visualize and compare the numeric variables of the dataset. According to the task we used the following R code in the Power BI, R script editor to create this visualization.

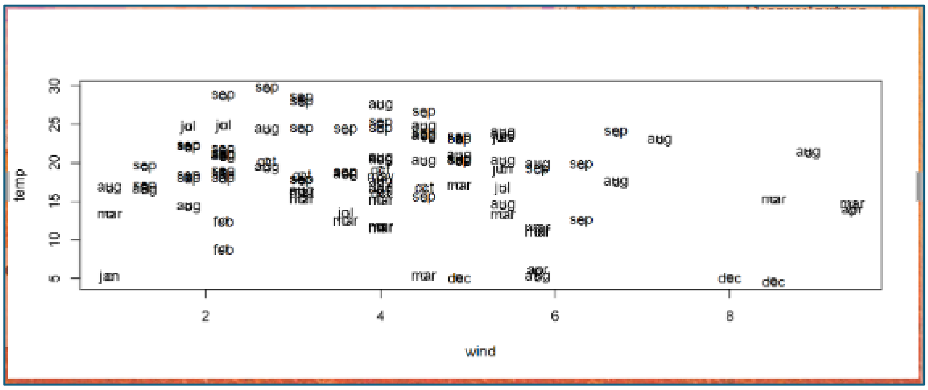
```
R script editor
⚠ Duplicate rows will be removed from the data.
# The following code to create a dataframe and remove duplicated rows is always executed and acts as a
preamble for your script:

# dataset <- data.frame(area, DC, DMC, ISI, FFMC, RH, temp, wind)
# dataset <- unique(dataset)

# Paste or type your script code here:
pairs(dataset)
```



3.3 Dot Plot-Visualization

We used this to visualize the temp and wind in the chart. A dot plot is comparable to a scatter plot. In the “forest fire” data set, the Total was used for the horizontal axis and the Year was used for the vertical axis to create this plot. According to the task we used the following R code in the Power BI R script editor to create this visualization.

```
R script editor
⚠ 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(area, DC, DMC, FFMC, ISI, month, RH, temp, wind)
# dataset <- unique(dataset)

# Paste or type your script code here:
plot(temp ~ wind, data =dataset)
with(dataset,text(temp ~ wind, labels=month))
```

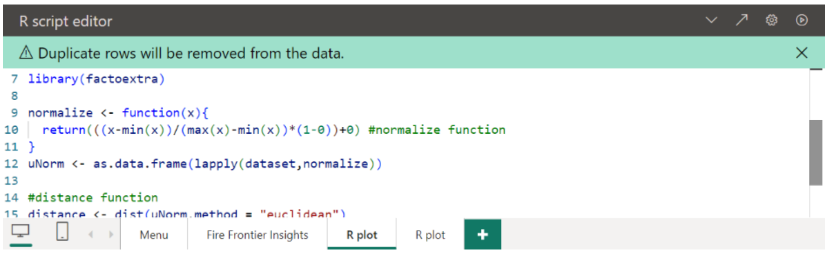


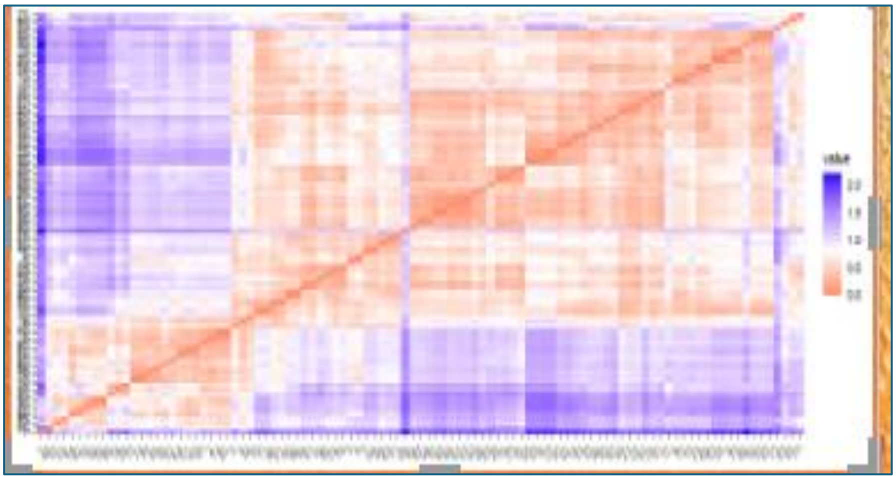
3.4 Dissimilarities of Forest fires

function is built in the ‘factoextra’ library and we used this chart to indicate the dissimilarities of each fighter according to their statistics.

Red color – Shows high similarity.

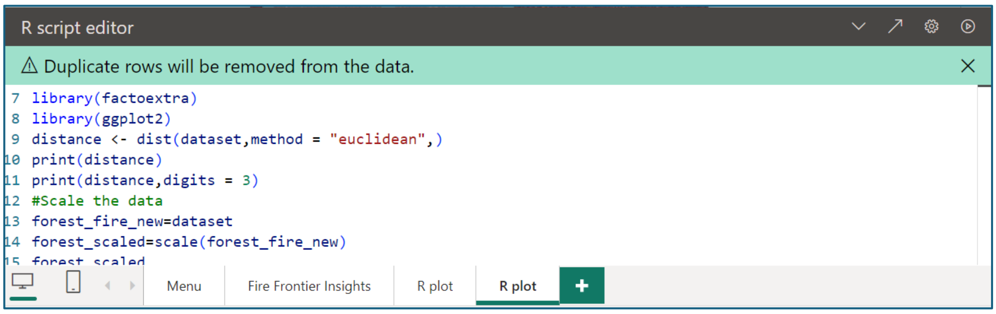
Blue color – Shows low similarity.

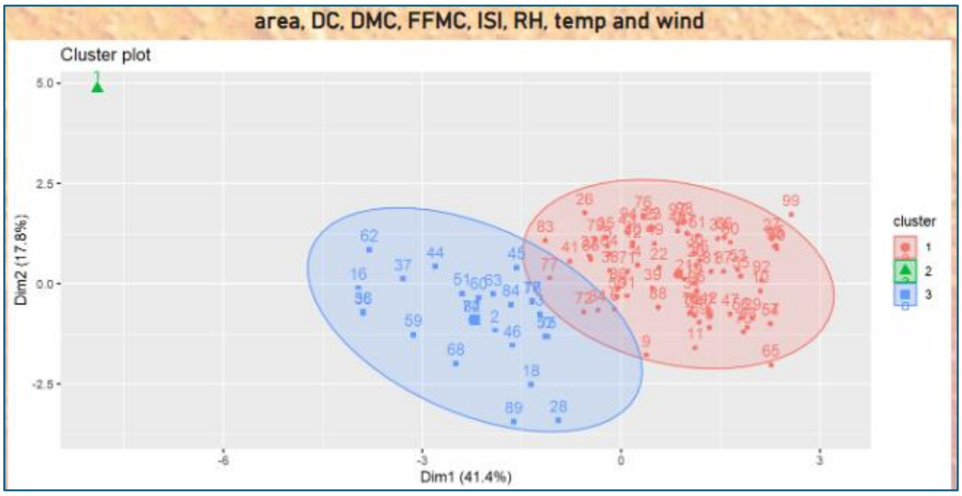




3.5. Cluster group of Forest fires

We used the above cluster plot to visualize the two clusters that we have distinguished before. The cluster plot shows similar patterns between the attributes and visualizes the groups of fires according to their latest fighting performances. Notice that, the cluster plot is not responsive to the slicer at the top of the page because the chart wants each data to give its graphical outcome.





3.6. Fire Frontier Insights

- Average of wind and RH by month

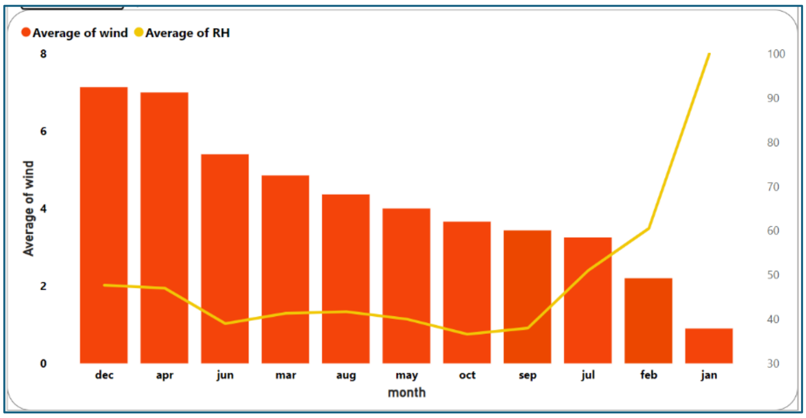
Two different information kinds are shown in the chart: a line chart and a bar chart. The average wind speed is displayed in the bar chart, while the average RH is represented by the line chart. Both are shown by month, with the y-axis showing the average values for wind speed and relative humidity and the x-axis showing the months of the year.

The annual variations in the average RH are depicted in the line graph. The line chart probably displays a wave-like pattern, with RH increasing in the winter and falling off in the summer. The reason why the relative humidity (RH) tends to be higher in the winter is because cold air cannot contain as much moisture as warm air.

Each month's average wind speed is displayed in a bar chart. The average wind speed for that month is indicated by the height of each bar. Throughout the year, there may be variations in the wind speed, with certain months seeing heavier winds than others. Any patterns or trends in the annual wind speed can be found on the chart.

You can examine the relationship between these two factors by viewing both the RH and wind speed on the same chart. In line with the prior explanation of the relationship between wind and relative humidity (RH), you might see, for instance, that months with high wind speeds also tend to have lower RH levels.

Overall, this chart can be useful for understanding how RH and wind speed change throughout the year and how they are related to each other. By analyzing this chart, you can gain insights into the local climate and make informed decisions based on the data.

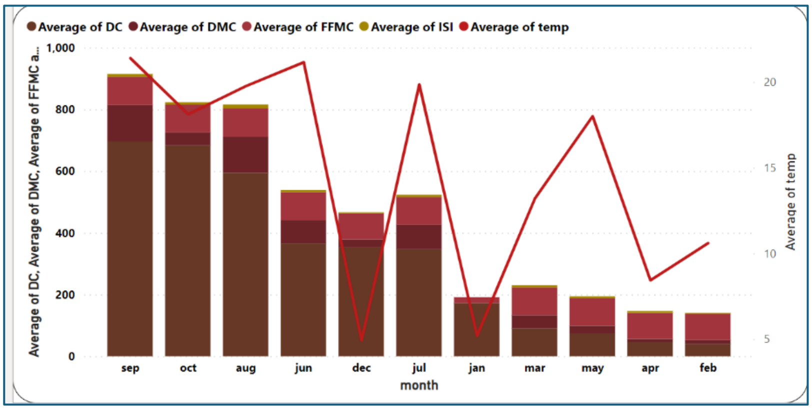


- Average of DMC.DC,FFMC, ISI temp by month

Two different information kinds are shown in the chart: a line chart and a bar chart. The average wind speed is displayed in the bar chart, while the average RH is represented by the line chart. Both are shown by month, with the y-axis showing the average values for wind speed and relative humidity and the x-axis showing the months of the year.
The annual variations in the average RH are depicted in the line graph. It is probable that the line chart displays a wave-like pattern, with RH increasing in the winter and falling off in the summer. The reason why the relative humidity (RH) tends to be higher in the winter is because cold air cannot contain as much moisture as warm air.

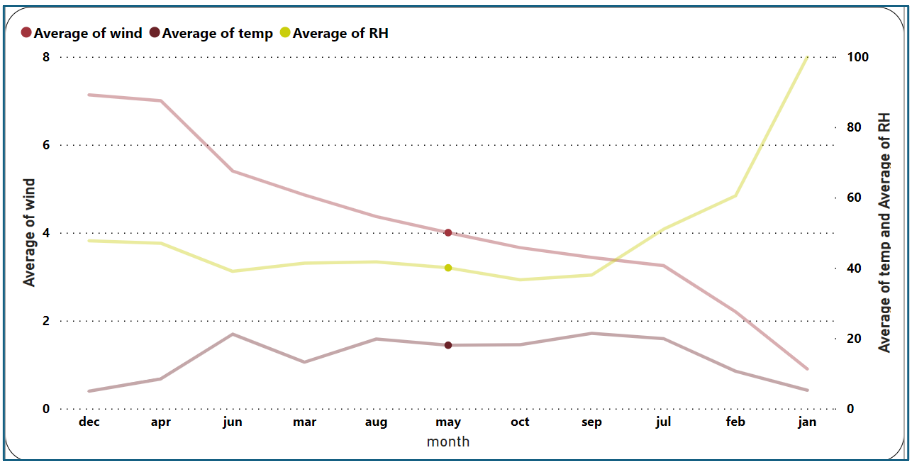
Each month's average wind speed is displayed in a bar chart. The average wind speed for that month is indicated by the height of each bar. Throughout the year, there may be variations in the wind speed, with certain months seeing heavier winds than others. Any patterns or trends in the annual wind speed can be found on the chart.
You can examine the relationship between these two factors by viewing both the RH and wind speed on the same chart. In line with the prior explanation of the relationship between wind and relative humidity (RH), you might see, for instance, that months with high wind speeds also tend to have lower RH levels.

In general, this chart can help you understand how RH and wind speed vary with the seasons and how they connect to one another. You are able to learn more about the local climate and make data-driven judgments by examining this chart.



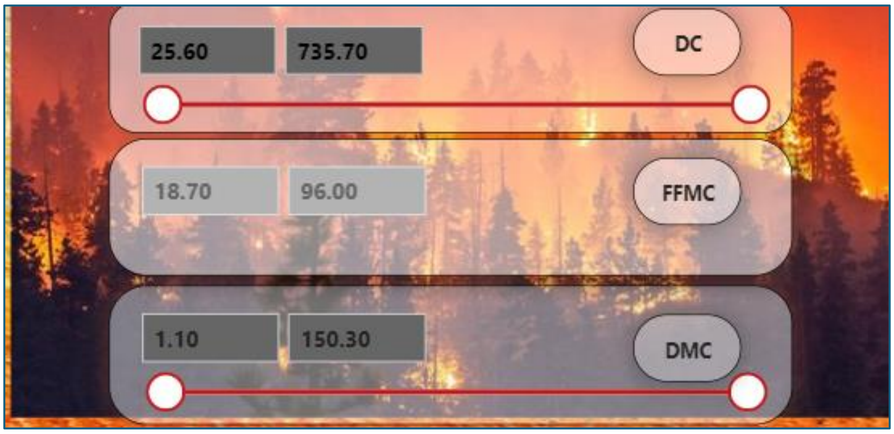
- ## Average Weather Parameters by Month

The average temperature, RH, and wind speed are shown by the three lines on the chart. The wind speed line illustrates the seasonal variations in wind speed, with higher wind speeds in certain months and lower wind speeds in others. The temperature line illustrates how the temperature fluctuates throughout the year, reaching higher summertime highs and lower wintertime lows. The relative humidity (RH) line illustrates how the RH varies throughout the year, being greater in the winter and lower in the summer.
There are several variables that can affect the complex link between wind speed, temperature, and relative humidity (RH), including time of day, location, and weather patterns. But generally speaking, certain trends do seem to surface. For instance, stronger winds can cause the rate of evaporation to rise, lowering the amount of moisture in the air and lowering RH. Similar to how warmer air can store more moisture than colder air, higher temperatures can also result in lower relative humidity.
All in all, this chart can help you comprehend the seasonal variations in wind speed, temperature, and relative humidity as well as their relationships. You are able to learn more about the local climate and make data-driven judgments by examining this chart.

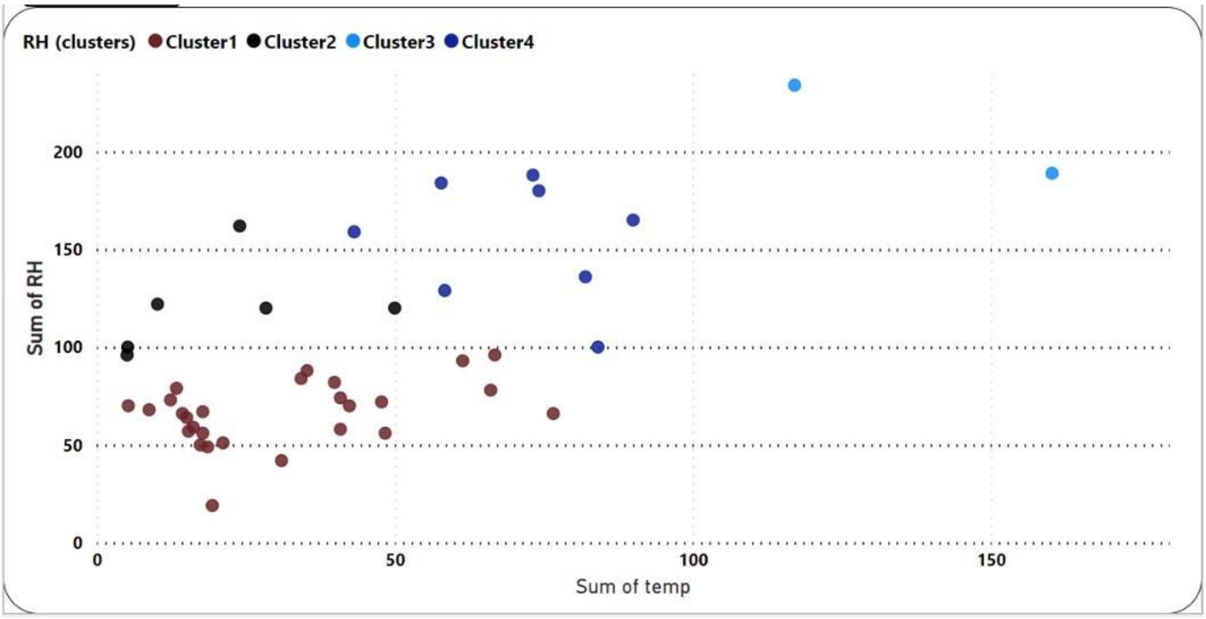


- Analysis of Fire Weather Indices

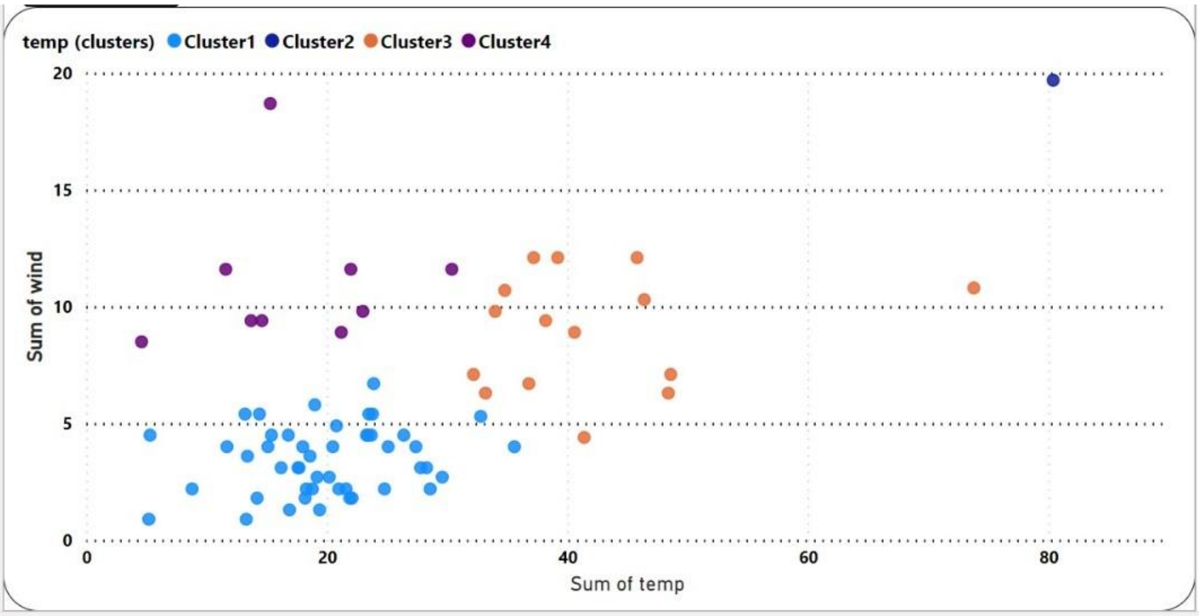
This interactive dashboard provides a comprehensive analysis of key fire weather indices: Drought Code (DC), Duff Moisture Code (DMC), and Fine Fuel Moisture Code (FFMC). Users can utilize the slicers to select and compare specific indices, gaining insights into their variations over time. The dashboard allows users to explore how these indices fluctuate across different months, facilitating a deeper understanding of the evolving fire weather conditions. Such insights are invaluable for assessing wildfire risk and implementing appropriate mitigation measures. By providing interactive tools for analysis, this dashboard empowers users to make informed decisions in wildfire management and prevention efforts.



5. Four main Clusters

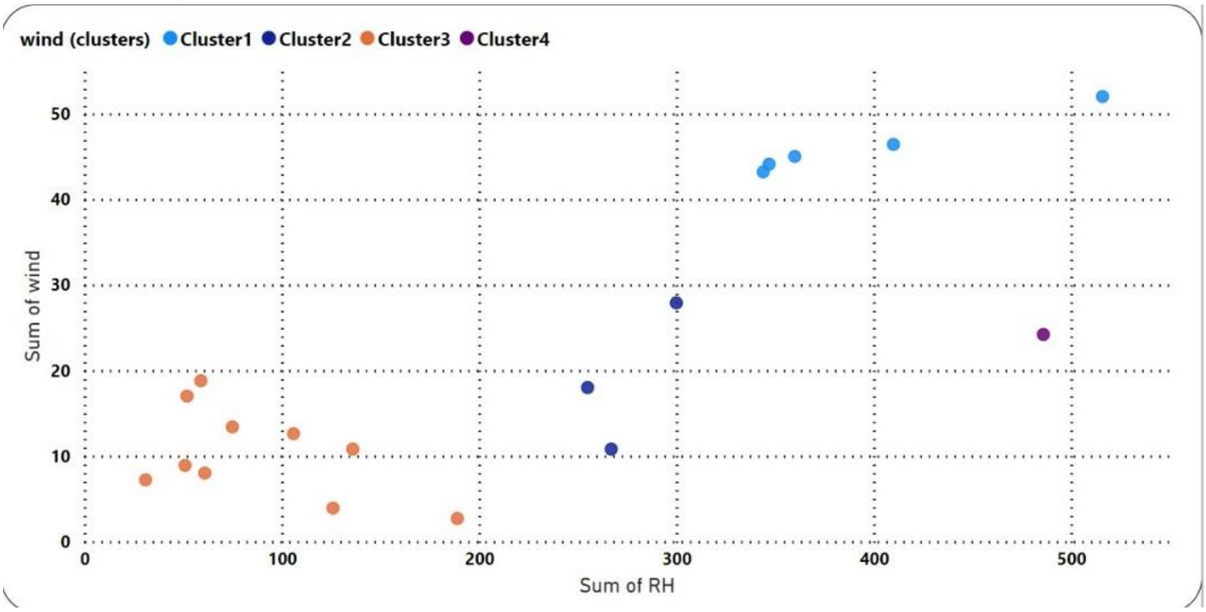


In summary, temperature and relative humidity are intricately linked in the atmosphere, with temperature influencing the air's moisture-holding capacity, and relative humidity indicating the proportion of water vapor present relative to that capacity. Understanding the relationship between these two variables is essential for assessing weather conditions, predicting atmospheric phenomena, and evaluating human comfort levels.



Overall, temperature and wind are closely linked in the Earth's atmosphere, with temperature variations playing a significant role in driving wind patterns and influencing wind speed and direction.

Temperature also indirectly affects wind speed and direction. Warmer air tends to hold more moisture and have a higher water vapor content, which can lead to the formation of clouds and precipitation. The presence of clouds and precipitation can affect wind speed and direction by altering atmospheric stability and pressure gradients.



wind speed and relative humidity are interconnected through their roles in moisture transport, evaporation, mixing, and atmospheric dynamics. Understanding the relationship between these variables is essential for interpreting weather patterns, assessing atmospheric conditions, and predicting environmental phenomena.

.