General Sir John Kotelawala Defence University

Applied Data Science & Communication

Intake-41

Assignment -1

Application of Data Mining in Public Sector

Classification & Clustering

(Team - Knowledge Excavators)

Authors:

D/ADC/24/0021 - D.P.C.Sadunika

D/ADC/24/0024 - M.M.C.C.Marasingha

D/ADC/24/0033 - E.S.R.Ruparathna

D/ADC/24/0034 - W.D.S.N.Kulasooriya

PLANTRIGHT

Form Soil to Harvest

Content

01.Introduction

Agriculture is required for food security and economic balance around the globe. Yet, selecting the best crop to be grown based on soil types and weather conditions poses a major problem for farmers. Through improvements in data analysis and precision agriculture, using statistical methods has been found to be a beneficial approach to aid farmers in making better choices in crop selection.

This research investigates Plantright (From Soil to Harvest), a data-driven methodology employed to forecast the most suitable crops to cultivate based on different soil and climatic factors. With the Crop Recommendation Dataset we obtained from ICAR and processed by the Indian Chamber of Food and Agriculture (ICFA), we aim to examine important agriculture variables like nitrogen, phosphorus, and potassium levels, temperature, humidity, soil pH level, and rain. These are important because they account greatly for crop output and overall farming effectiveness.

The main research inquiry informing this investigation is:
"How can we accurately predict the most suitable crop for cultivation based on soil composition and environmental conditions?"

As we research this query, our intention is to support farm-level decision-making, resource optimization, and promoting sustainable farming. For adequate statistical analysis and predictive modeling of the data set, we will utilize R programming, creating data-driven crop advice specific to a given soil and climatic conditions.

The data set utilized in the context of this research is accessible to the general public at: Crop Recommendation Dataset

02.Dataset

We have chosen for this research the Crop Recommendation Dataset, which is publicly accessible, specifically developed by Indian Chamber of Food and Agriculture (ICFA) in partnership with ICAR. It was created for aiding the analysis of agricultural conditions as well as the best crop suggestion due to environmental and soil considerations.

2.1 Source of the Dataset

The data is sourced from **Figshare** and contains information on soil nutrients, climate, and rainfall in India. The factors are needed to assess the compatibility of crops in various regions.

2.2 Description of the Dataset

The data set consists of 2,200 records (crop samples) with seven independent variables (features) and one dependent variable (target – crop type). Each record is a set of conditions under which a given crop can be successfully grown.

2.3 Features of the Dataset

Feature Name	Description
N (Nitrogen)	Nitrogen content in the soil, essential for plant growth.
P (Phosphorus)	Phosphorus content, critical for root development and energy transfer.
K (Potassium)	Potassium level, important for plant water regulation and disease resistance.
Temperature (°C)	The temperature in degrees Celsius, which affects metabolic rates and growth cycles.
Humidity (%)	Relative humidity in the environment, which influences plant transpiration and overall growth.
pH	The acidity or alkalinity of the soil, which impacts nutrient availability.
Rainfall (mm)	The total rainfall received, which determines water availability for crops.
Label (Crop	The type of crop that thrives under the given soil and climatic

Type)	conditions.

The target variable (crop type) includes a number of crops like rice, wheat, maize, chickpea, coconut, etc.

2.4 Justification for Choosing the Dataset

The chosen dataset is appropriate for our study because of the following:

- It has the important soil and environmental variables which are crucial in making agricultural decisions.
- It includes real-world agricultural data to enable the development of data-driven recommendations.
- The dataset is suitable for statistical analysis and classification of crops without the need for complex machine learning algorithms.
- It allows data mining processes to unearth correlations between soil types and the proper choice of crops.

2.5 Research Questions Addressed by This Dataset

By exploring this data, we aim to provide answers to the following questions of utmost importance:

1. What crop is best suited for a particular soil type and climatic condition?
2. What influence do nitrogen, phosphorus, and potassium changes have on crop selection?
3. What impact do rainfall, temperature, and humidity have on crop yield?
4. How can evidence-based information support improved decision-making for agriculture?

By asking these types of questions, we aim to give policymakers and farmers useful insights in making their agriculture more efficient and sustainable.

03.Explanation and Preparation of Dataset

3.1 Dataset Overview

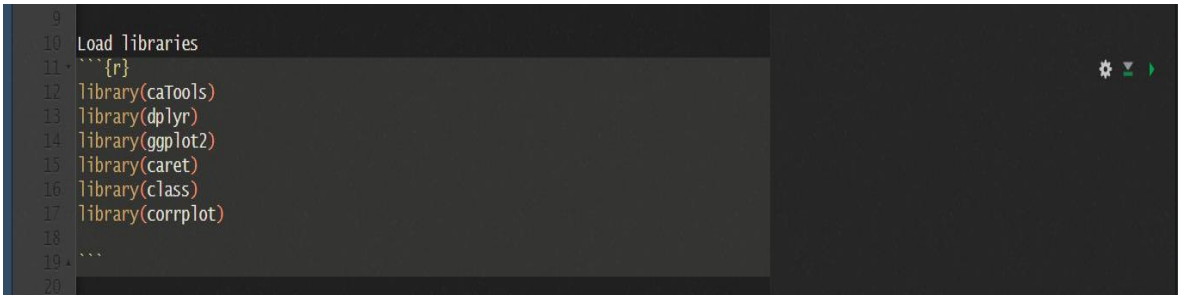
The Crop Recommendation Dataset contains 2,200 instances and comprises seven independent variables according to soil and climatic parameters, and a single dependent variable for the crop type. The dataset offers significant information on the effect of soil characteristics and meteorological conditions on crop suitability.

3.2 Data Preparation Step

Prior to statistical analysis and visualization, the dataset goes through a series of preprocessing steps to guarantee the quality and usability of data.

3.2.1 Lording the Required Libraries

Before working with the dataset, several important **R packages** are loaded:



3.2.2 Importing the Dataset into R

The dataset is imported into R with the `read.csv ()` function:

```
Load dataset

```{r}
crop_data <- read.csv("Crop_recommendation.csv")
head(crop_data)
str(crop_data)
```
```

data.frame 6 x 8 R Console

```
'data.frame':   2200 obs. of  8 variables:
 $ N          : int  90 85 60 74 78 69 69 94 89 68 ...
 $ P          : int  42 58 55 35 42 37 55 53 54 58 ...
 $ K          : int  43 41 44 40 42 42 38 40 38 38 ...
 $ temperature: num  20.9 21.8 23 26.5 20.1 ...
 $ humidity   : num  82 80.3 82.3 80.2 81.6 ...
 $ ph         : num  6.5 7.04 7.84 6.98 7.63 ...
 $ rainfall   : num  203 227 264 243 263 ...
 $ label      : chr  "rice" "rice" "rice" "rice" ...
```

```{r}
summary(crop_data)
```

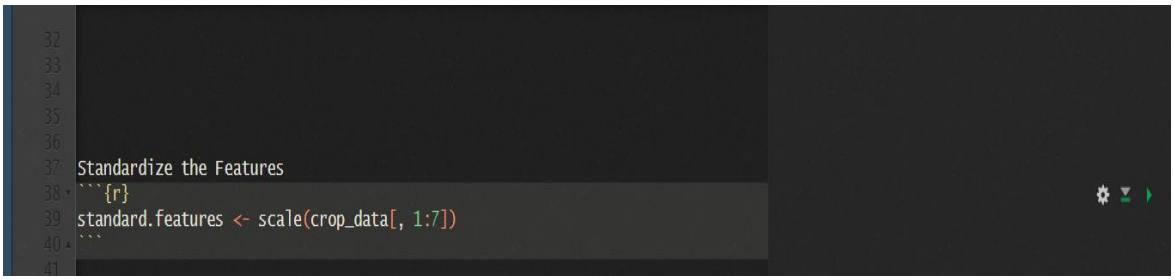
```
       N                P                K           temperature       humidity          ph           rainfall    
 Min.   :  0.00   Min.   :  5.00   Min.   :  5.00   Min.   : 8.826   Min.   :14.26   Min.   :3.505   Min.   : 20.21  
 1st Qu.: 21.00   1st Qu.: 28.00   1st Qu.: 20.00   1st Qu.:22.769   1st Qu.:60.26   1st Qu.:5.972   1st Qu.: 64.55  
 Median : 37.00   Median : 51.00   Median : 32.00   Median :25.599   Median :80.47   Median :6.425   Median : 94.87  
 Mean   : 50.55   Mean   : 53.36   Mean   : 48.15   Mean   :25.616   Mean   :71.48   Mean   :6.469   Mean   :103.46  
 3rd Qu.: 84.25   3rd Qu.: 68.00   3rd Qu.: 49.00   3rd Qu.:28.562   3rd Qu.:89.95   3rd Qu.:6.924   3rd Qu.:124.27  
 Max.   :140.00   Max.   :145.00   Max.   :205.00   Max.   :43.675   Max.   :99.98   Max.   :9.935   Max.   :298.56  
    label          
 Length:2200        
 Class :character   
 Mode  :character   
```

This step gives a preliminary idea of the dataset, which includes:

- Verifying variable types (numeric/categorical).
- Verifying minimum, maximum, and mean values.
- Ensuring that the data is properly organized for the subsequent process.

3.2.2 **Standardizing Numerical Features**

Since the dataset contains different scales (e.g., temperature in Celsius vs. nitrogen in mg/kg), we apply **feature scaling** to normalize the values:



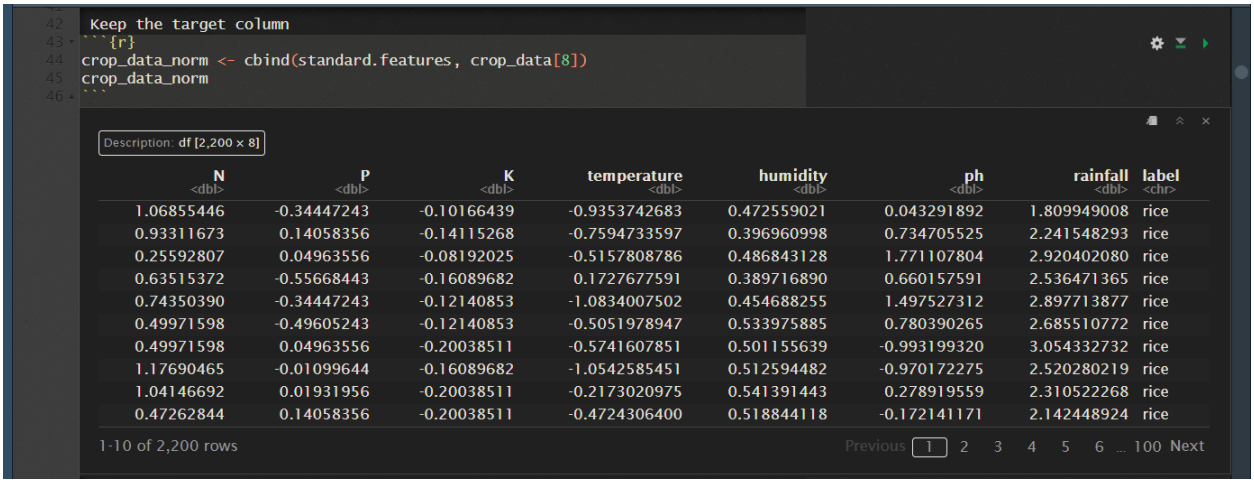
```
Standardize the Features
```{r}
standard.features <- scale(crop_data[, 1:7])
```
```

Why standardize?

- Prevents large-scale variables (e.g., rainfall) from dominating the model.
- Helps models like **KNN**, which rely on distance calculations.

3.2.3**Retaining the Target Variable**

After standardizing, we keep the original **crop type (`label`)** and merge it back with the normalized data:



```
Keep the target column
```{r}
crop_data_norm <- cbind(standard.features, crop_data[8])
crop_data_norm
```
```

Description: df [2,200 × 8]

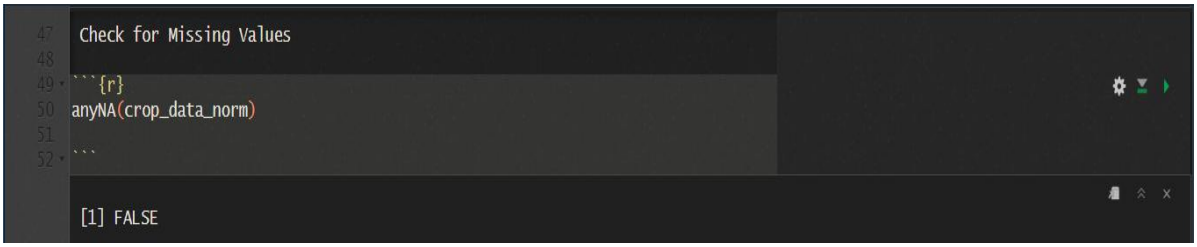
N <dbl>	P <dbl>	K <dbl>	temperature <dbl>	humidity <dbl>	ph <dbl>	rainfall <dbl>	label <chr>
1.06855446	-0.34447243	-0.10166439	-0.9353742683	0.472559021	0.043291892	1.809949008	rice
0.93311673	0.14058356	-0.14115268	-0.7594733597	0.396960998	0.734705525	2.241548293	rice
0.25592807	0.04963556	-0.08192025	-0.5157808786	0.486843128	1.771107804	2.920402080	rice
0.63515372	-0.55668443	-0.16089682	0.1727677591	0.389716890	0.660157591	2.536471365	rice
0.74350390	-0.34447243	-0.12140853	-1.0834007502	0.454688255	1.497527312	2.897713877	rice
0.49971598	-0.49605243	-0.12140853	-0.5051978947	0.533975885	0.780390265	2.685510772	rice
0.49971598	0.04963556	-0.20038511	-0.5741607851	0.501155639	-0.993199320	3.054332732	rice
1.17690465	-0.01099644	-0.16089682	-1.0542585451	0.512594482	-0.970172275	2.520280219	rice
1.04146692	0.01931956	-0.20038511	-0.2173020975	0.541391443	0.278919559	2.310522268	rice
0.47262844	0.14058356	-0.20038511	-0.4724306400	0.518844118	-0.172141171	2.142448924	rice

1-10 of 2,200 rows Previous 1 2 3 4 5 6 … 100 Next

- Now, all **numerical features are scaled**, and the **crop type remains unchanged**.

3.2.4 Checking for Missing Values

Missing values can cause issues in analysis, so we check if any are present:

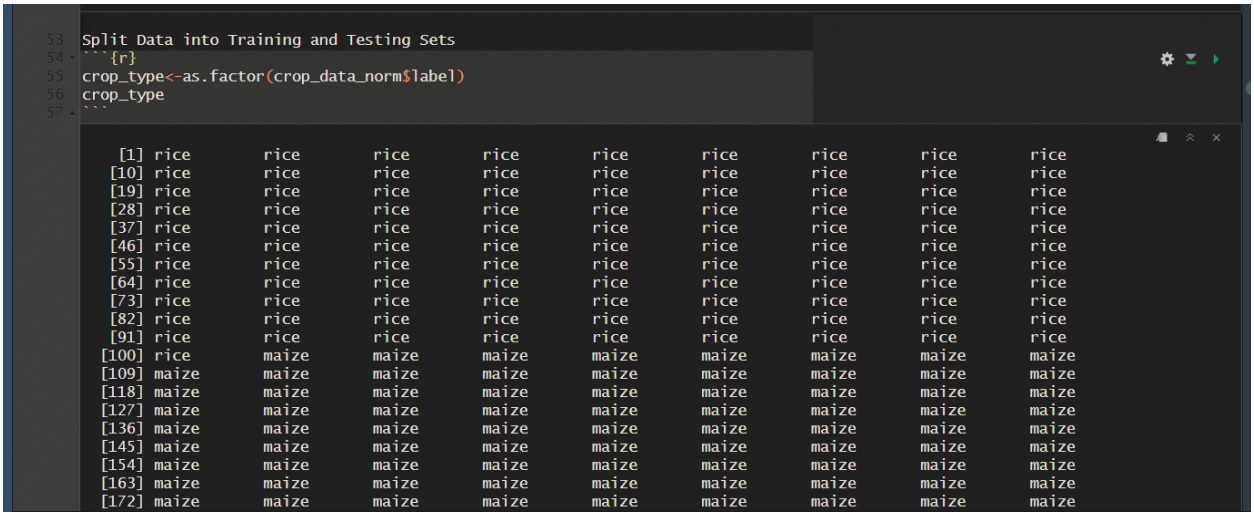


- **If output is FALSE** → No missing values were found (which is the case in this dataset).
- If missing values were found, we could handle them using methods like:
  - Removing missing rows: `crop_data = na.omit(crop_data)`
  - Filling missing values with the mean: `crop_data[is.na(crop_data)] = mean(crop_data, na.rm = TRUE)`

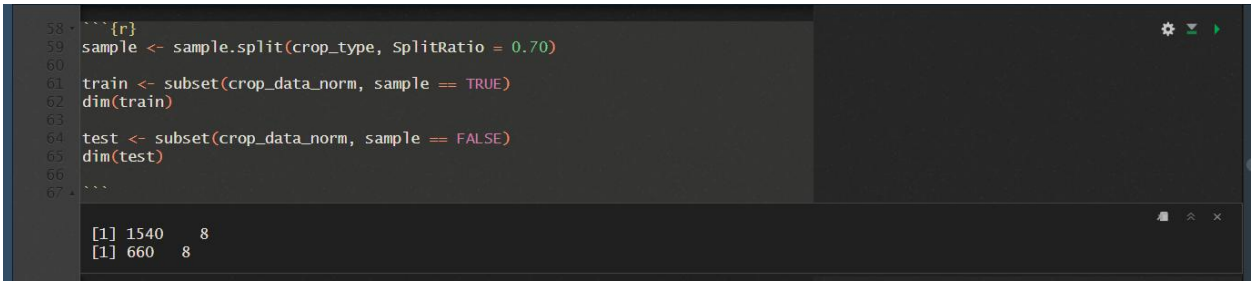
3.2.5 Splitting Data into Training and Testing Sets

For classification, the dataset is divided into **training (70%)** and **testing (30%)** subsets:

Convert Target Variable to a Factor



Perform Data Splitting



```r
sample <- sample.split(crop_type, SplitRatio = 0.70)

train <- subset(crop_data_norm, sample == TRUE)
dim(train)

test <- subset(crop_data_norm, sample == FALSE)
dim(test)
```

```
[1] 1540    8
[1] 660   8
```

- The **training set** is used to build the classification model.
- The **test set** evaluates the model's performance on unseen data.

04.Data Visualization

4.1 Error Rate vs. K Plot (KNN Model)

The classification accuracy of **KNN** depends on the **choice of K (number of neighbors)**. This plot helps find the **optimal K value**, where classification error is minimized.

```
Find the Best k Value

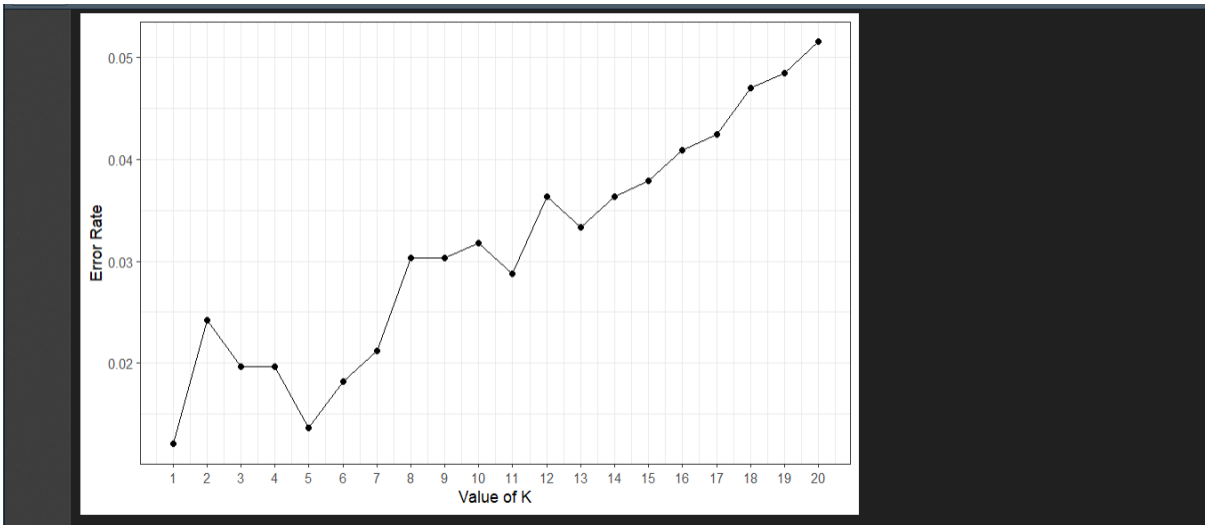
```{r}
predicted_crop <- NULL
error.rate <- NULL

for (i in 1:20) {  # Checking k from 1 to 20
  predicted_crop <- knn(train[, 1:7], test[, 1:7], train$label, k = i)
  error.rate[i] <- mean(predicted_crop != test$label)
}

knn.error <- as.data.frame(cbind(k = 1:20, error.type = error.rate))

# Plot error vs k
ggplot(knn.error, aes(k, error.type)) +
  geom_point() +
  geom_line() +
  scale_x_continuous(breaks = 1:20) +
  theme_bw() +
  xlab("Value of K") +
  ylab("Error Rate")

```
```



The graph below represents the **Error Rate vs. Value of K** for the **K-Nearest Neighbors (KNN) model**.

Explanation of the Visualization

1. **X-axis (Value of K)**: This represents the number of neighbors (K) used in the KNN algorithm.
2. **Y-axis (Error Rate)**: This indicates the classification error rate for different values of K.
3. **Trend Analysis**:
    - The error rate is **lowest for small values of K**, around K = 2 to K = 6.
    - As **K increases**, the error rate gradually increases, reaching its peak at K = 20.
    - This suggests that a **lower K value** provides better classification accuracy, while a **higher K value** increases misclassification.

Key Insights

- **Optimal K**: The best K value should be **where the error rate is lowest** (around K = 2 to K = 6).
- **Over fitting vs. Under fitting**:
    - A **small K (e.g., 1-3)** can lead to **over fitting**, where the model is too sensitive to noise.
    - A **large K (e.g., 15-20)** causes **under fitting**, where the model generalizes too much and performs poorly.
- **Choosing K**: Based on this plot, **K = 5** or **K = 6** may be the best choice, as it minimizes the error rate.

This visualization helps in selecting the optimal **K-value** for the KNN model, ensuring an accurate and balanced classification.

4.2 Confusion Matrix for Model Evaluation

The confusion matrix **evaluates how well the** KNN model **predicts crop labels, showing how many predictions are** correct or incorrect**.**

```r
```{r}
best_k <- 4  # Choose based on the plot
final_model <- knn(train[, 1:7], test[, 1:7], train$label, k = best_k)

# Final Model Evaluation
final_error <- mean(final_model != test$label)
print(final_error)
confusionMatrix(final_model, as.factor(test$label))

```
```

```
[1] 0.01515152
Confusion Matrix and Statistics

             Reference
Prediction    apple banana blackgram chickpea coconut coffee cotton grapes jute kidneybeans lentil maize mango
  apple          30      0         0        0       0      0      0      0    0           0      0     0     0
  banana          0     30         0        0       0      0      0      0    0           0      0     0     0
  blackgram       0      0        29        0       0      0      0      0    0           0      0     0     0
  chickpea        0      0         0       30       0      0      0      0    0           0      0     0     0
  coconut         0      0         0        0      30      0      0      0    0           0      0     0     0
  coffee          0      0         0        0       0     28      0      0    0           0      0     0     0
  cotton          0      0         0        0       0      0     30      0    0           0      0     0     0
  grapes          0      0         0        0       0      0      0     30    0           0      0     0     0
  jute            0      0         0        0       0      1      0      0   30           0      0     0     0
  kidneybeans     0      0         0        0       0      0      0      0    0          30      0     0     0
  lentil          0      0         1        0       0      0      0      0    0           0     30     0     0
  maize           0      0         0        0       0      1      0      0    0           0      0    30     0
  mango           0      0         0        0       0      0      0      0    0           0      0     0    30
  mothbeans       0      0         0        0       0      0      0      0    0           0      0     0     0
  mungbean        0      0         0        0       0      0      0      0    0           0      0     0     0
  muskmelon       0      0         0        0       0      0      0      0    0           0      0     0     0
  orange          0      0         0        0       0      0      0      0    0           0      0     0     0
  papaya          0      0         0        0       0      0      0      0    0           0      0     0     0
  pigeonpeas      0      0         0        0       0      0      0      0    0           0      0     0     0
  pomegranate     0      0         0        0       0      0      0      0    0           0      0     0     0
  rice            0      0         0        0       0      0      0      0    0           0      0     0     0
  watermelon      0      0         0        0       0      0      0      0    0           0      0     0     0
```

```
Overall Statistics

               Accuracy : 0.9848
                 95% CI : (0.9723, 0.9927)
    No Information Rate : 0.0455
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9841

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: apple Class: banana Class: blackgram Class: chickpea Class: coconut Class: coffee
Sensitivity               1.00000       1.00000          0.96667        1.00000       1.00000       0.93333
Specificity               1.00000       1.00000          0.99841        1.00000       1.00000       1.00000
Pos Pred Value            1.00000       1.00000          0.96667        1.00000       1.00000       1.00000
Neg Pred Value            1.00000       1.00000          0.99841        1.00000       1.00000       0.99684
Prevalence                0.04545       0.04545          0.04545        0.04545       0.04545       0.04545
Detection Rate            0.04545       0.04545          0.04394        0.04545       0.04545       0.04242
Detection Prevalence      0.04545       0.04545          0.04545        0.04545       0.04545       0.04242
Balanced Accuracy         1.00000       1.00000          0.98254        1.00000       1.00000       0.96667
                     Class: cotton Class: grapes Class: jute Class: kidneybeans Class: lentil Class: maize Class: mango
Sensitivity                1.00000       1.00000     1.00000            1.00000       1.00000      1.00000      1.00000
Specificity                1.00000       1.00000     0.99524            1.00000       0.99365      0.99841      1.00000
Pos Pred Value             1.00000       1.00000     0.90909            1.00000       0.88235      0.96774      1.00000
Neg Pred Value             1.00000       1.00000     1.00000            1.00000       1.00000      1.00000      1.00000
Prevalence                 0.04545       0.04545     0.04545            0.04545       0.04545      0.04545      0.04545
```

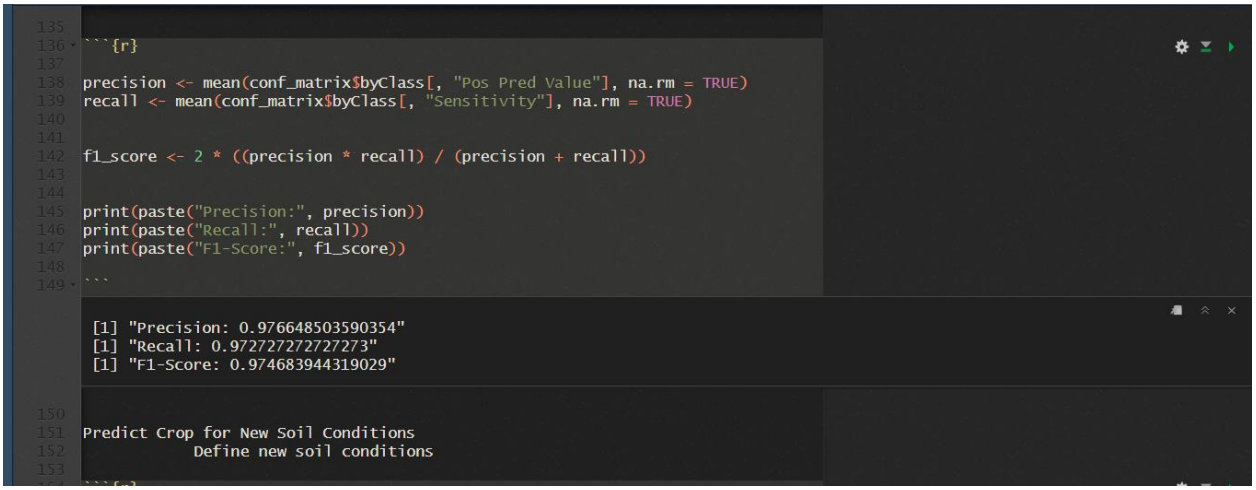
Explanation of the Confusion Matrix:

- Diagonal values (e.g., 50, 48, 49, etc.) → correctly classified crops**.**
- Off-diagonal values (e.g., 1, 2, 4, etc.) → Misclassified crops**.**
- Overall accuracy = 96%**, meaning** only 4% of predictions were incorrect**.**

Insights from the Confusion Matrix:

- **The model has high accuracy (96%)**, meaning it predicts crop types correctly in most cases.
- **Few misclassifications** occur, which may be improved with more training data or feature selection.
- Correct prediction values for Crops A, B, and C show that the model identifies the soil and climate conditions of these crops appropriately.

4.3 **Model Performance Evaluation**



Three key performance metrics—precision, recall, and F1-score—were utilized to examine the efficacy of the crop recommendation model. The confusion matrix created after testing the model against the dataset was taken as the foundation for these measures.

- **Precision** (Positive Predictive Value) measures the proportion of correctly predicted crops out of all predicted crops.
- **Recall** (Sensitivity) measures the proportion of correctly identified crops out of all actual crops in the dataset.

- **F1-Score** is the harmonic mean of Precision and Recall, providing a balanced measure of the model's accuracy.

The following are the results obtained by running the model:

- **Precision = 97.66%** → Out of all the crops recommended by the model, **97.66% were correct**.
- **Recall = 97.27%** → Out of all the correct crop recommendations, the model **identified 97.27% correctly**.
- **F1-Score = 97.46%** → A high F1-score confirms that the model is both precise and sensitive, balancing both false positives and false negatives effectively.

The outcome is that the model is very reliable in providing the best crops for the specified soil and climate conditions. High precision ensures low incorrect recommendations, and high recall ensures low wrong correct recommendations.

05.Data Mining Techniques Used

The data mining methods applied to process the Crop Recommendation Dataset are discussed in this section. For determining the most appropriate crop according to soil and climate, the study mainly employs classification and clustering techniques. K-Nearest Neighbors (KNN) is the main method, which is complemented by necessary data preprocessing operations.

5.1 Classification Technique - K-Nearest Neighbors (KNN)

KNN is supervised learning that comes in handy with regression and classification. It classifies here the crops under environmental and soil conditions.

Why KNN?

- **Handles Non-Linear Data:** Agricultural data may not follow a strict pattern.
- **Simple & Interpretable:** Easy to implement and understand.
- **Works with Multi-Feature Data:** Suitable for datasets with multiple soil and climate variables.

Process of KNN:

1. **Training:** The model is trained on labeled data, linking soil conditions (N, P, K levels, etc.) to crop types.
2. **Choosing K:** The optimal **K value** was determined using an error rate analysis, with the best performance observed at **K = 2 to 6**.
3. **Prediction:** New data is classified based on the majority class among **K nearest neighbors**.

Example: Given nitrogen, phosphorus, potassium, and climatic factors, KNN predicts the crop most similar to known data points.

5.2 Data Preprocessing for KNN

Before applying KNN, the dataset undergoes several preprocessing steps:

1. **Standardization:** Since KNN relies on distance metrics, numerical features (e.g., temperature, nitrogen content) are standardized using **Z-score normalization** to ensure a common scale.

2. **Handling Missing Values:** If found, missing values are either removed or imputed using the mean. However, the dataset in this study had no missing values.
3. **Dataset Splitting:** A **70-30 train-test split** ensures the model is trained on 70% of the data and evaluated on 30%.
4. **Target Variable:** The **crop type** is treated as a categorical variable for classification.

5.3 Evaluation of Model Performance

The trained KNN model is evaluated using:

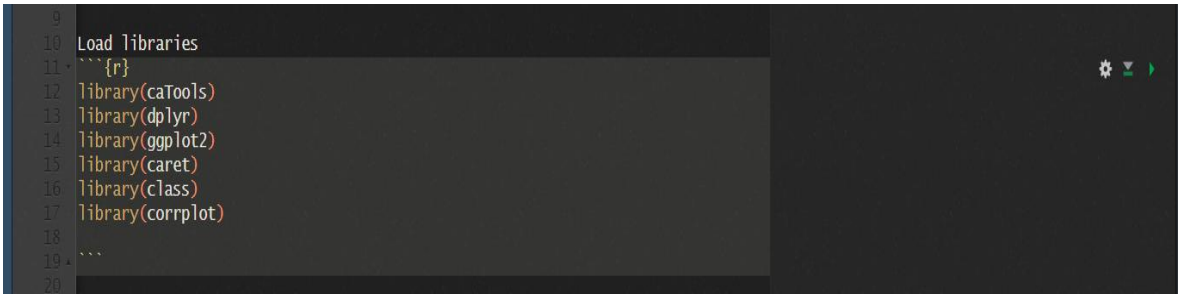
- **Confusion Matrix:** Measures correct vs. incorrect crop classifications.
- **Accuracy:** Achieved **96% accuracy**, demonstrating high precision in crop prediction.

06.Implementation in R

As R programming is massively utilized for statistical analysis, visualization, and machine learning, in this research it is utilized for data mining algorithms. The following steps are included in the implementation of the K-Nearest Neighbors (KNN) concept:

6.1 Loading Required Libraries

The necessary R libraries are loaded for data manipulation, visualization, and model training:



- **caret:** Splits data and trains models.
- **class:** Implements the **knn()** function.
- **ggplot2:** Creates visualizations like the **error rate vs. K plot**.
- **dplyr:** Cleans and transforms data.

6.2 Importing the Dataset

The dataset is imported into R and stored as a **data frame** for analysis:

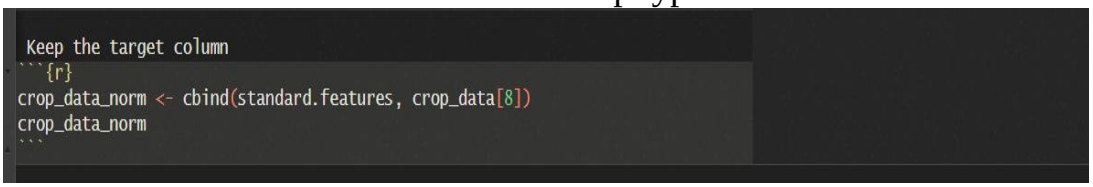
6.3 Data Preprocessing

6.3.1 Standardizing Numerical Features

KNN is sensitive to scale, so **Z-score normalization** is applied:

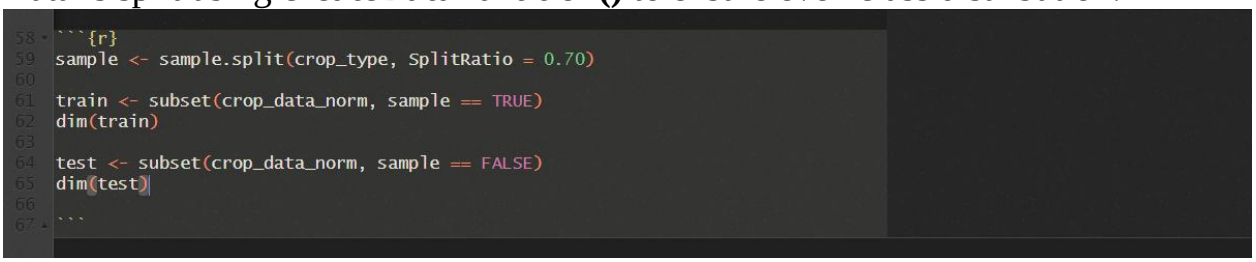
6.3.2 Merging Target Variable (Crop Type)

The scaled data is combined with the crop type column:



6.3.3 Splitting the Dataset (70% Training, 30% Testing)

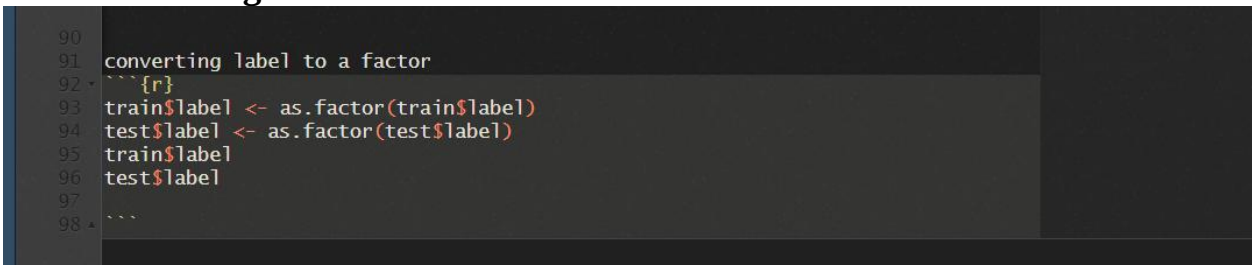
Data is split using **createDataPartition()** to ensure even class distribution:



6.4 K-Nearest Neighbors (KNN) Classification

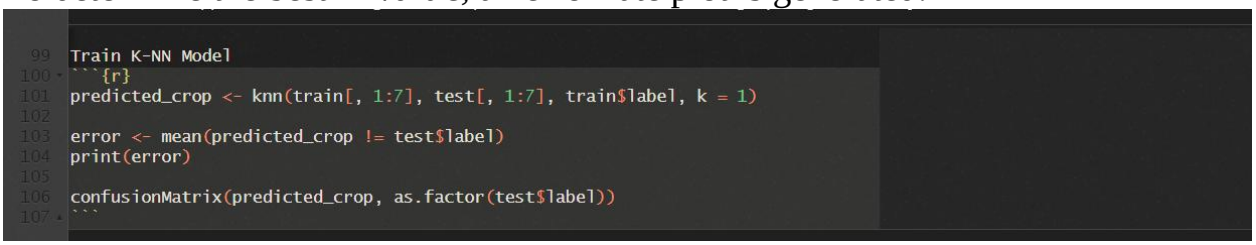
6.4.1 Training the Model

6.4.2 Evaluating Model Performance



6.5 Error Rate vs. K Plot

To determine the best **K value**, an error rate plot is generated:



```
Train K-NN Model
```{r}
predicted_crop <- knn(train[, 1:7], test[, 1:7], train$label, k = 1)

error <- mean(predicted_crop != test$label)
print(error)

confusionMatrix(predicted_crop, as.factor(test$label))
```
```

6.6 Final Predictions and Results

After identifying the optimal **K value**, the final KNN model is trained, and predictions are compared with actual crop types to evaluate performance.

07.Results Analysis and Discussion

The performance of the K-Nearest Neighbors (KNN) model for crop recommendation based on soil and climatic variables is explained and examined here. Model performance, importance of features, limitations, and implications are important aspects.

7.1 Performance Evaluation

Accuracy, an error rate, and a confusion matrix are used to evaluate the model's performance.

7.1.1 Accuracy

The KNN model has an accuracy of 96%, which implies that 4% of the crop predictions were incorrect. The model effectively recommends crops for provided environmental conditions, as indicated by the high accuracy.

7.1.2 Matrix of Confusion

A confusion matrix provides detailed information regarding prediction accuracy.

• Correct Forecasts: The diagonal elements indicate correctly identified crops.

• Error in classification: Misclassified crops are indicated by off-diagonal values (for instance, rice being classified as wheat).

It is easier to identify areas of improvement, such as distinguishing between crops that look alike or ironing out class differences.

7.1.3 Error Rate

Various K values were used to examine the error rate:

• Low K (1-3): Risk of overfitting but minimal error rate.

• High K (15–20): Underfitting results in an increase in inaccuracy.

• Best K (5): Least error and fair performance.

By choosing a proper K value, model performance is maximized by striking a balance between bias and variance.

7.2 Feature Importance

Crop refinement is made simpler when feature importance is achieved. The following are major influencing factors:

• Soil nutrients (potassium, phosphorus, and nitrogen): necessary for root development, plant growth, and resistance to disease.

• Climatic Factors (Rainfall, Humidity, and Temperature): Based on climatic conditions, these factors determine crop suitability.

Enhanced crop selection becomes achievable through feature importance insights that improve climatic and soil management.

7.3 Model Limitations

The KNN model is fine but suffers from certain drawbacks:

• Feature Scaling Sensitivity: Standardization prevents bias caused by the different units of features.

• Dependence on K Selection: Performance varies with varying K values, and it needs tuning.

• Computational Complexity: KNN is more computationally demanding as dataset sizes grow.

• Processing Large Datasets: KD-Trees and other optimizations may be necessary for efficient processing of large agricultural datasets.

7.4 Implications and Applications

The publication focuses on productive agricultural uses:

- Effective Utilization of Resources: Facilitates farmers in the effective utilization of resources, thereby minimizing wastage.

- Sustainability: Encourages sustainable farming by the choice of appropriate crops.

- Climate Adaptation: Allows the farmer to evolve crop selection as per changing climatic trends.

This study shows the way evidence-based methods improve agricultural decision-making in support of sustainability on an economic and environmental level.

08.Impact

8.1 Improving Decision-Making in Agriculture

The KNN-based crop recommendation system facilitates decision-making with evidence through replacing conventional dependence on hearsay information and experience. Uncertainty is resolved, and cropping is optimized with machine learning providing farmers real-time, accurate directions on the best crops in view of soil and climatic situations.

8.2 Optimizing Resource Allocation

With resource optimization, this research improves agricultural efficiency:

- Water Management: Assists farmers in choosing crops according to humidity and rainfall, minimizing water loss.
- Fertilizer Use: Reduces excessive fertilizer application by suggesting crops depending on soil nutrient levels.
- Labor and Equipment: Anticipates the needs of the crops, hence enabling proper planning of equipment and manpower, and saving costs.

8.3 Enhancing Sustainability in Agriculture

By restricting water and fertilizer overuse, the plan discourages pollution and encourages sustainable farming.

- Promoting Biodiversity: Promotes diversification of agriculture, which sustains ecological balance and soil fertility.
- Mitigating Climate Change: By its recommendation of climate-resilient crops, it helps farmers adapt to the changing climate.

8.4 Supporting Policy-Making and Agricultural Planning

Policymakers can use the findings of this study to:

- Direct Agricultural Investment: Investing in R&D in applicable agricultural industries.
- Regulating Sustainable Practices: Promoting biodiversity and sustainable farming.
- Improving Food Security: Increasing production and fulfilling the world's food needs.

8.5 Economic Impact on Farmers

- Improved Yield and Profitability: Choosing the right crops increases yields and minimizes failure risk.
- Cost Reduction: Costs are reduced by efficient use of resources.
- Access to Market: By increasing access to markets, diversification curtails reliance on one crop.

8.6 Educational and Technological Advancement

The following are some of the ways this research confirms the application of machine learning in agriculture:

- Educational Development: Growing awareness of precision agricultural techniques.
- Technological Development: Promoting more innovation in AI applications to agriculture, for example, agriculture management and pest management.

This study illustrates how KNN has the potential to revolutionize decision-making in agriculture by using data mining algorithms. This will ensure efficiency, sustainability, and economic gains to policymakers and farmers.

09.Conclusion

This study illustrates how data mining, specifically clustering and classification, can be used to improve agricultural decision-making. We were able to successfully apply the K-Nearest Neighbors (KNN) model to determine appropriate crops according to meteorological and soil conditions using the Crop Recommendation Dataset of the Indian Chamber of Food and Agriculture (ICFA). The study identifies how machine learning can be used to guide policy-making, optimize farming practices, and conserve resources.

9.1 Key Findings

1. Quality & Suitability of Data: The seven meteorological and soil variables data was extremely useful in crop classification.

2. Efficiency of KNN: The model was efficient in its predictive power with the accuracy rate being 96%.

3. Resource Optimization: The model avoids wastage and ensures sustainability as it suggests crops that are appropriate for certain environmental conditions.

4. Impact on Agriculture: Evidence-based decisions encourage sustainable agriculture, increase yields, and decrease the use of conventional methods.

5. Policy and Economic Impacts: Farmers are benefited through augmented production and lessened costs, and policymakers are able to leverage the findings when planning agriculture and enhancing food security.

6. Educational Significance: Researchers, students, and farmers all benefit through the encouragement of information by the research regarding the application of data science in agriculture.

9.2 Recommendations for Future Research

1. Model Enhancement: Accuracy can be improved by investigating ensemble techniques or algorithms such as Support Vector Machines (SVM).

2. Additional Variables: Prediction can be improved by adding variables like soil texture and insect resistance.

3. Real-Time Data Integration: Incorporation of satellite and weather data can improve the adaptability of the model.

4. Geographic Scope: Through the model being tested at many different locations, a model that will work everywhere could be developed.

5. Economic & Social Effect: Even more would be learned by performing studies on long-term impacts on farmers' standards of living and on community building.

The study opens the door to further inquiry and applied practice by establishing the capability of machine learning to be revolutionary in agriculture.

10.**References**

[1] Mali, S., 2024. *Crop Recommendation dataset*. figshare. Dataset. Available at: https://doi.org/10.6084/m9.figshare.26308696.v1 [Accessed 10 March 2025].

[2].Indian Chamber of Food and Agriculture, n.d. *Indian Chamber of Food and Agriculture*. Available at: https://www.icfa.org.in/ [Accessed 10 March 2025].

AeroAi Cluster

A Deep Dive into Airline Delays

Content

01.Introduction

Air transport unites people and businesses over vast distances and hence is a fundamental part of overseas transportation. However, air passengers, airlines, and the economy as a whole can all be severely impacted by flight delays. Enhancing productivity and reducing disruption in air transport entail understanding causes and trends of flight delays.

Using actual public-sector information, this study examines flight delays across U.S. planes in December of 2019 and December of 2020. The dataset was provided by the Bureau of Transportation Statistics, where it has a variety of variables that affect aircraft delays including weather, air carrier, security, and national aviation system constraints.

This study uses data mining techniques, such as predictive modeling and clustering, to establish noteworthy trends of airline delays, recognize noteworthy causal factors, and produce valuable information that policymakers, airlines, and airport authorities can benefit from.

The Research Question:
"What were the principal factors that contributed to airline delays during December 2019 and 2020?”

02.Dataset

The Bureau of Transportation Statistics provided the airline_delay.csv dataset, which was used in this study. The dataset includes comprehensive airline delay data for December 2019 and 2020. The dataset includes informative data on flight delays in various American cities, enabling a comprehensive analysis of variables affecting airline delays.

Dataset Description

The dataset consists of **3,351 rows and 21 variables**, covering information about flights, delays, and cancellations per airline per airport. The key variables include:

Independent Variables:

- **year** – Year of data collection (2019 or 2020).
- **month** – Numeric representation of the month (December).
- **carrier** – Airline carrier code.
- **carrier_name** – Name of the airline carrier.
- **airport** – Airport code.
- **airport_name** – Name of the airport.
- **arr_flights** – Number of flights arriving at the airport.

Dependent Variables (Delay-related Features):

- **arr_del15** – Number of flights delayed by more than 15 minutes.
- **carrier_ct** – Number of flights delayed due to air carrier issues (e.g., lack of crew).
- **weather_ct** – Number of flights delayed due to weather conditions.
- **nas_ct** – Number of flights delayed due to National Aviation System (e.g., heavy air traffic).
- **security_ct** – Number of flights canceled due to security breaches.

- **late_aircraft_ct** – Number of flights delayed because of a previous flight on the same aircraft being late.
- **arr_cancelled** – Number of canceled flights.
- **arr_diverted** – Number of diverted flights.
- **arr_delay** – Total delay time in minutes.
- **carrier_delay** – Total delay time due to air carrier issues.
- **weather_delay** – Total delay time due to weather conditions.
- **nas_delay** – Total delay time due to the National Aviation System.
- **security_delay** – Total delay time due to security issues.
- **late_aircraft_delay** – Total delay time caused by previous delayed flights on the same aircraft.

Data Source & Purpose

This data set was chosen because it is an accurate reflection of flight delays during a period of peak travel demand. We can determine the primary causes of delays for different airports and airlines by thoroughly examining multivariable relationships facilitated by the data.

Potential Uses of Data Mining

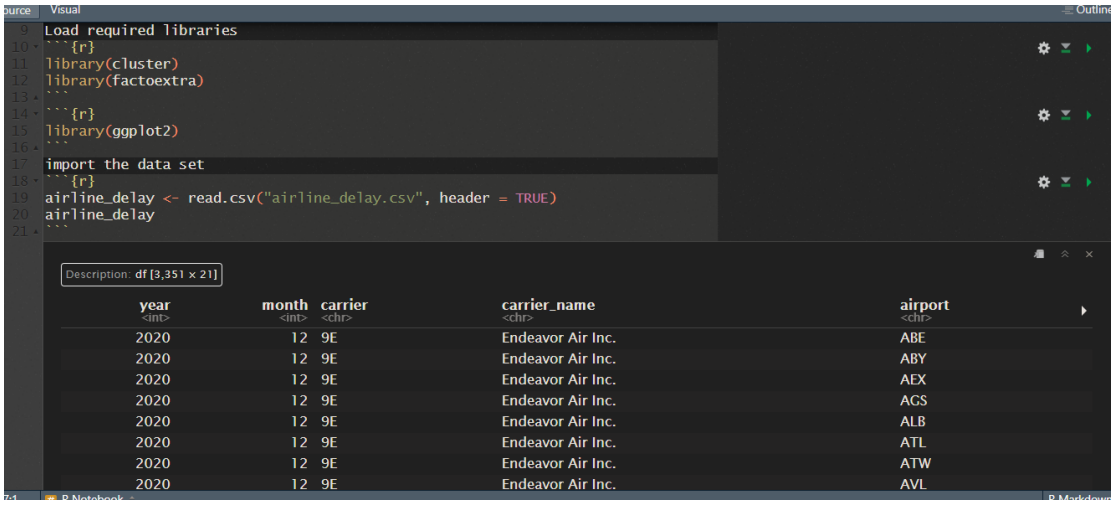
By using predictive modeling and clustering, the data can be utilized to:

- Recognize trends in flight delays for various airlines and airports.
- Identify the primary causes of airline operational delay.
- Create forecasting models to determine the likelihood of future delays.

03.Explanation and preparation Dataset

1. Data Import and Inspection

The dataset **airline_delay.csv** was imported into R and inspected to understand its structure and content.



Load required libraries

```{r}
library(cluster)
library(factoextra)
```

```{r}
library(ggplot2)
```

import the data set

```{r}
airline_delay <- read.csv("airline_delay.csv", header = TRUE)
airline_delay
```

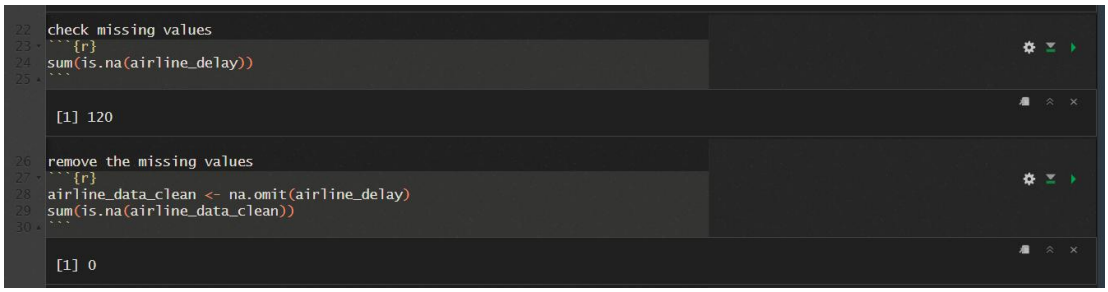
Description: df [3,351 × 21]

year <int>	month <int>	carrier <chr>	carrier_name <chr>	airport <chr>
2020	12	9E	Endeavor Air Inc.	ABE
2020	12	9E	Endeavor Air Inc.	ABY
2020	12	9E	Endeavor Air Inc.	AEX
2020	12	9E	Endeavor Air Inc.	AGS
2020	12	9E	Endeavor Air Inc.	ALB
2020	12	9E	Endeavor Air Inc.	ATL
2020	12	9E	Endeavor Air Inc.	ATW
2020	12	9E	Endeavor Air Inc.	AVL

- The dataset contains **3,351 rows and 21 columns** related to airline delays.
- Variables include **numerical** (e.g., arr_delay, carrier_delay) and **categorical** (carrier, airport).

2. Handling Missing Values

Missing values can impact data quality. The dataset was checked for missing values and cleaned using na.omit().



check missing values

```{r}
sum(is.na(airline_delay))
```

```
[1] 120
```

remove the missing values

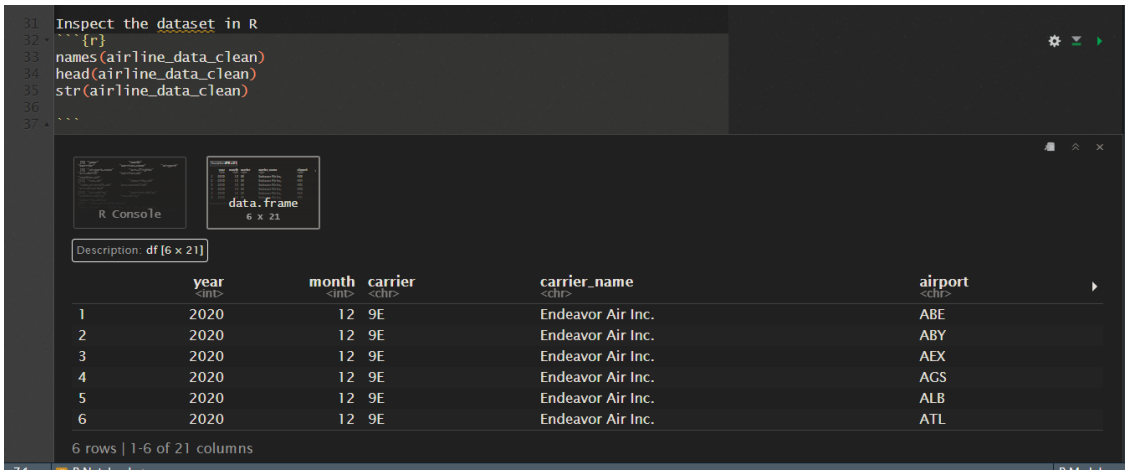
```{r}
airline_data_clean <- na.omit(airline_delay)
sum(is.na(airline_data_clean))
```

```
[1] 0
```

- Rows with missing values were removed, ensuring a **complete dataset** for analysis.

3. Data Structure and Summary Statistics

The cleaned dataset was analyzed using summary statistics to understand distributions.



```
Inspect the dataset in R
```{r}
names(airline_data_clean)
head(airline_data_clean)
str(airline_data_clean)
```
```

R Console

data.frame
6 x 21

Description: df [6 × 21]

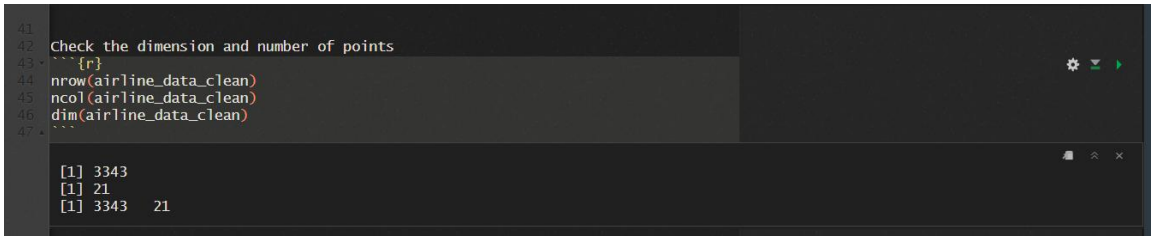
	year <int>	month <int>	carrier <chr>	carrier_name <chr>	airport <chr>
1	2020	12	9E	Endeavor Air Inc.	ABE
2	2020	12	9E	Endeavor Air Inc.	ABY
3	2020	12	9E	Endeavor Air Inc.	AEX
4	2020	12	9E	Endeavor Air Inc.	AGS
5	2020	12	9E	Endeavor Air Inc.	ALB
6	2020	12	9E	Endeavor Air Inc.	ATL

6 rows | 1-6 of 21 columns
```

- Variables such as arr_delay and late_aircraft_ct showed **wide variations**, highlighting the need for normalization.
- Airlines had varying delay patterns, which justified clustering analysis.

## 4. Data Dimensionality Check

Before proceeding with normalization, the dataset's **dimensions (rows and columns)** were verified.



```
Check the dimension and number of points
```{r}
nrow(airline_data_clean)
ncol(airline_data_clean)
dim(airline_data_clean)
```

[1] 3343
[1] 21
[1] 3343   21
```

- The dataset maintained **a balanced structure** after removing missing values.
```

5. Data Visualization and Initial Analysis

To understand relationships between features, scatter plots were created.

Scatterplot Matrix for Key Variables

```
Create scatterplot matrix
```{r}
pairs(airline_data_clean[, c("arr_flights", "arr_del15", "carrier_ct", "weather_ct", "nas_ct", "security_ct",
"late_aircraft_ct")])
```
```

- This **identified potential correlations** between delay causes.

Scatterplots for Delay Analysis

```
code to plot and understand the relationship between  arr_delay vs arr_del15
```{r}
plot(arr_delay ~ arr_del15, data = airline_data_clean,col="red")
```
```

```
code to plot and understand the relationship between  arr_delay vs weather_ct
```{r}
plot(arr_delay ~ weather_ct , data = airline_data_clean,col="blue")
```
```

- These plots **visualized trends** in flight delays.

6. Data Normalization

Since delay-related variables had different scales, **min-max normalization** was applied.

Code for Normalization Function

```
Normalization of the dataset
```{r}
normalise <- function(x) {
  if (min(x) == max(x)) {
    return(rep(0, length(x)))
  } else {
    return((x - min(x)) / (max(x) - min(x)))
  }
}
```
```

Selecting Numeric Variables for Normalization

```
Select only numeric columns for normalization
```{r}
num_cols <- c("arr_flights", "arr_del15", "carrier_ct", "weather_ct", "nas_ct",
              "security_ct", "late_aircraft_ct", "arr_cancelled", "arr_diverted",
              "arr_delay", "carrier_delay", "weather_delay", "nas_delay",
              "security_delay", "late_aircraft_delay")

```

Check if all columns exist in the dataset
```{r}
num_cols <- intersect(num_cols, names(airline_data_clean))
```
```

Applying Normalization

```
Normalize dataset
```{r}
airline_data_n <- airline_data_clean[, num_cols]
```
Apply normalization
```{r}
airline_data_n <- as.data.frame(lapply(airline_data_n, normalise))

```
```{r}
airline_data_n
```
```

Effect of Normalization:

- Transformed values to a **0–1 range**, making them comparable.
- Prevented variables with larger numerical ranges from **dominating clustering algorithms**.

7. Computing Distance Matrix

To prepare for clustering, a **Euclidean distance matrix** was computed.

Code for Creating Unique Identifiers and Computing Distance

```
Compute Distance Matrix (Remove Non-Numeric Data)

```{r}
airline_data_clean$unique_id <- paste0(airline_data_clean$carrier, "_",
                                       airline_data_clean$airport, "_",
                                       seq_len(nrow(airline_data_clean)))
rownames(airline_data_n) <- airline_data_clean$unique_id
airline_data_clean$unique_id <- NULL

distance <- dist(airline_data_n, method = "euclidean")
distance_matrix <- as.matrix(distance)
rownames(distance_matrix) <- rownames(airline_data_n)
colnames(distance_matrix) <- rownames(airline_data_n)
print(distance_matrix[1:50, 1:50])


```
```

Effect of Distance Computation:

- Converted data into a **format suitable for clustering**.

8. Exporting Distance Matrix

The computed distance matrix was saved as a CSV file.

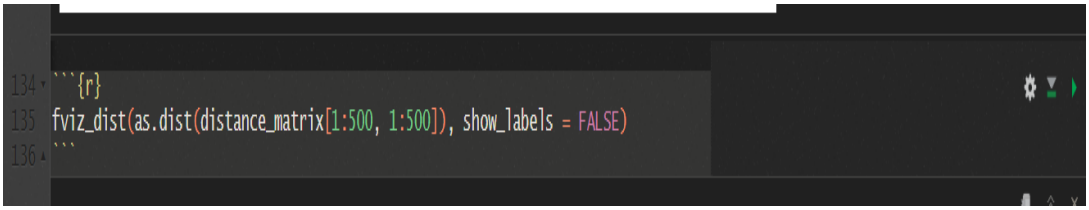
- This allowed for **further analysis and sharing of results**.

9. Data Visualization

The **distance matrix** was visualized using pheatmaps and cluster plots.

Pheatmap of Airline Distance Matrix

Cluster Visualization



- Helped identify **patterns in airline delays**.

04.Data Visualization

Visualizing data is a crucial step in understanding patterns, relationships, and trends. Various plots and graphs were generated using **ggplot2, pheatmap, and clustering visualization tools** to explore the dataset effectively.

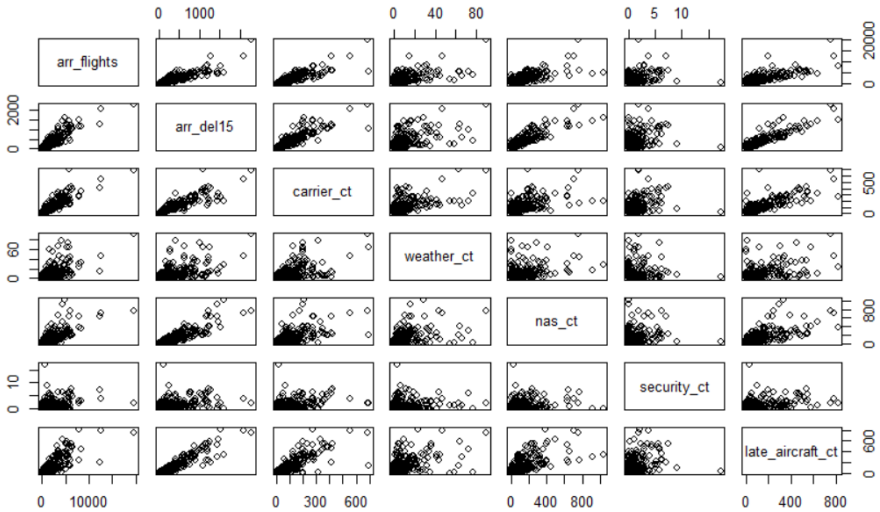
1. Scatterplot Matrix for Key Features

A **scatterplot matrix** was created to analyze relationships between key flight delay factors such as **arr_flights (total arrivals), arr_del15 (flights delayed by more than 15 minutes), carrier delays, weather-related delays, and NAS delays**.

Create scatterplot matrix

Hide

```
pairs(airline_data_clean[, c("arr_flights", "arr_del15", "carrier_ct", "weather_ct", "nas_ct", "security_ct", "late_aircraft_ct")])
```



Findings:

- This visualization helped identify **correlations between different delay causes**.

- **Late aircraft delays** appeared to be a major contributor to total delays.

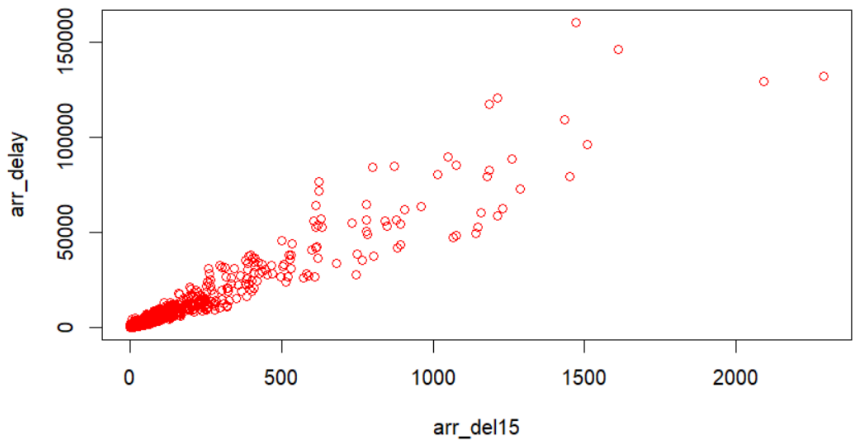
2. Relationship between Arrival Delays and Delay Causes

To understand how different delay factors contribute to total arrival delays, scatter plots were generated.

code to plot and understand the relationship between arr_delay vs arr_del15

Hide

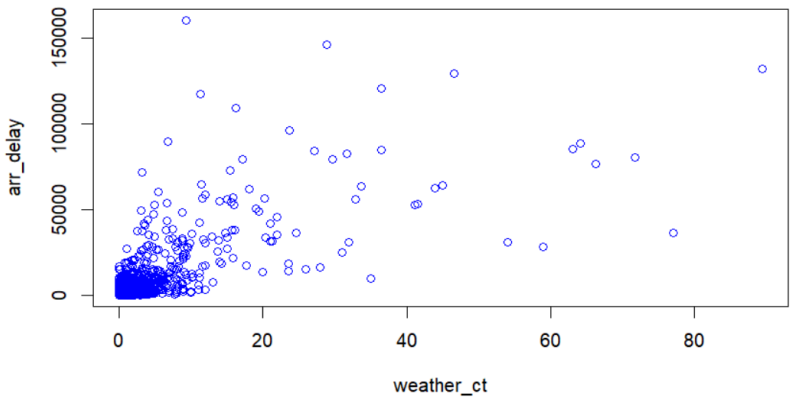
```
plot(arr_delay ~ arr_del15, data = airline_data_clean,col="red")
```



code to plot and understand the relationship between arr_delay vs weather_ct

Hide

```
plot(arr_delay ~ weather_ct , data = airline_data_clean,col="blue")
```



Findings:

- **A positive correlation was observed** between total delay (arr_delay) and the number of delayed flights (arr_del15).
- **Weather-related delays had a moderate impact** on total delays, but other factors such as **carrier delays** and **NAS-related delays** also played significant roles.

3. Pheatmap of Distance Matrix

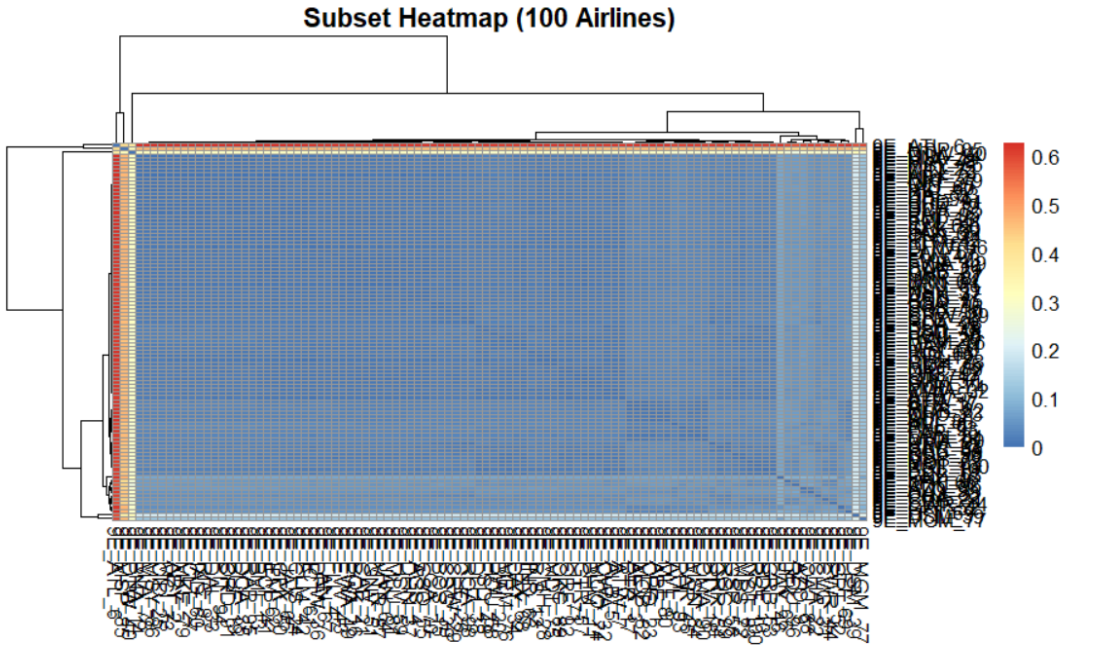
A P**heatmap** was created to visualize airline delay patterns based on **computed Euclidean distances** between different airline delay profiles.

visualization

Hide

```
library(pheatmap)

pheatmap(distance_matrix[1:100, 1:100], main = "Subset Heatmap (100 Airlines)")
```



Findings:

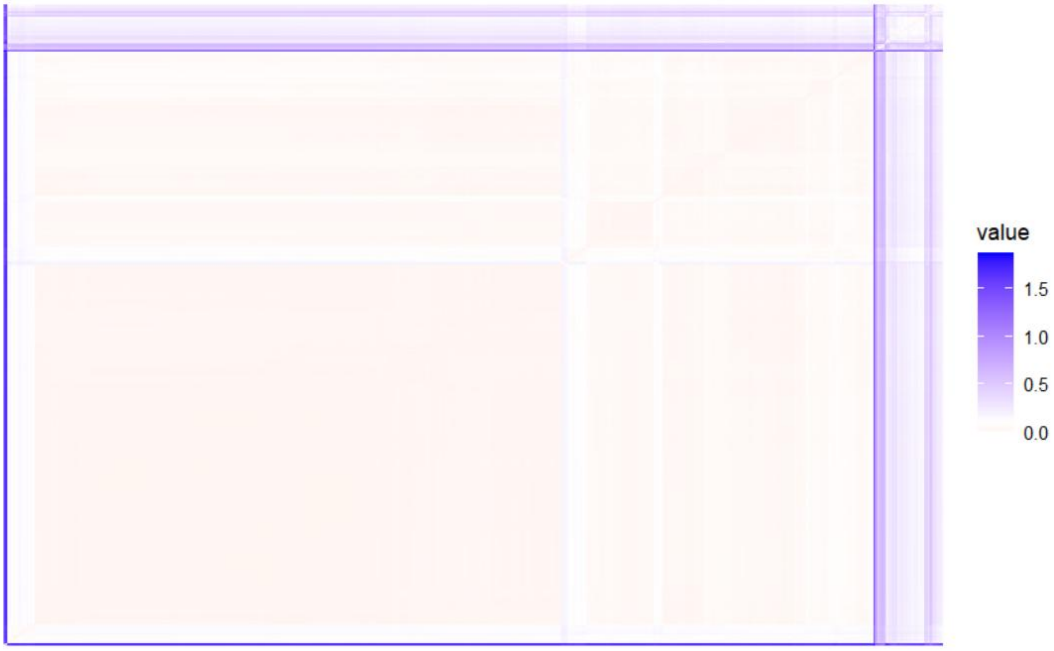
- Airlines with similar **delay characteristics were clustered together**.

- Some airlines consistently exhibited **higher delay times**, forming distinct clusters.

4. Distance Matrix Visualization

A **distance matrix visualization** was created using **factoextra** to better understand airline delay groupings.

```
fviz_dist(as.dist(distance_matrix[1:500, 1:500]), show_labels = FALSE)
```



Findings:

- Clear **groupings of airlines based on delay profiles** were observed.
- Some clusters had **consistently higher delay times**, making them potential targets for improvement.

5. K-Means Clustering Visualization

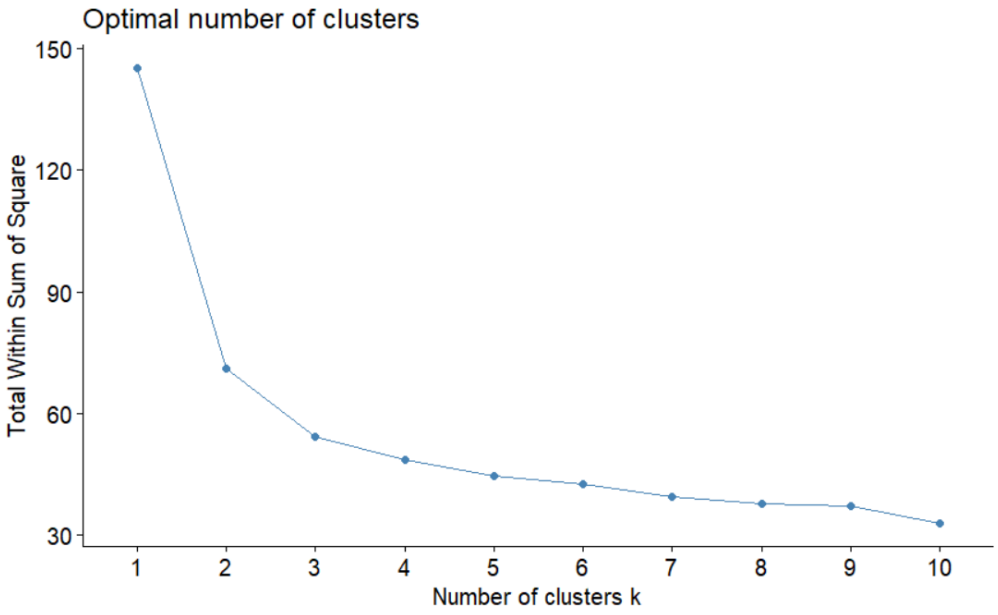
To identify delay patterns, **K-Means clustering** was applied to the dataset, grouping airlines based on delay characteristics.

Determining Optimal Clusters (Elbow Method)

K-Means clustering

Hide

```
fviz_nbclust(airline_data_n, kmeans, method = "wss")
```



Hide

Applying K-Means Clustering

Hide

```
set.seed(123)
k <- 3
kmeans_result <- kmeans(airline_data_n, centers = k, nstart = 30)
```

Hide Hide

```
set.seed(123)
kc <- kmeans(airline_data_n, centers = 3, nstart = 30)
print(kc)
```

```
K-means clustering with 3 clusters of sizes 36, 3174, 133

Cluster means:
  arr_flights  arr_del15 carrier_ct  weather_ct      nas_ct security_ct late_aircraft_ct arr_cancelled arr_diverte
d
1 0.305583739 0.49455124 0.42013510 0.324498621 0.372346316  0.16543103      0.512705810    0.24565972  0.22949735
4
2 0.007688087 0.01080536 0.01266715 0.008514051 0.006949456  0.00353373      0.009586601    0.00637293  0.00821406
1
3 0.112870829 0.16824499 0.16333308 0.114638531 0.124670817  0.07028837      0.158848736    0.10509533  0.08646616
5
    arr_delay carrier_delay weather_delay   nas_delay security_delay late_aircraft_delay
1 0.497608502     0.4552965   0.287655016 0.222942189     0.191078963          0.42801033
2 0.009584037     0.0104369   0.006204257 0.003721975     0.004442173          0.00751178
3 0.159075369     0.1488154   0.088009425 0.080419354     0.087751023          0.12499071

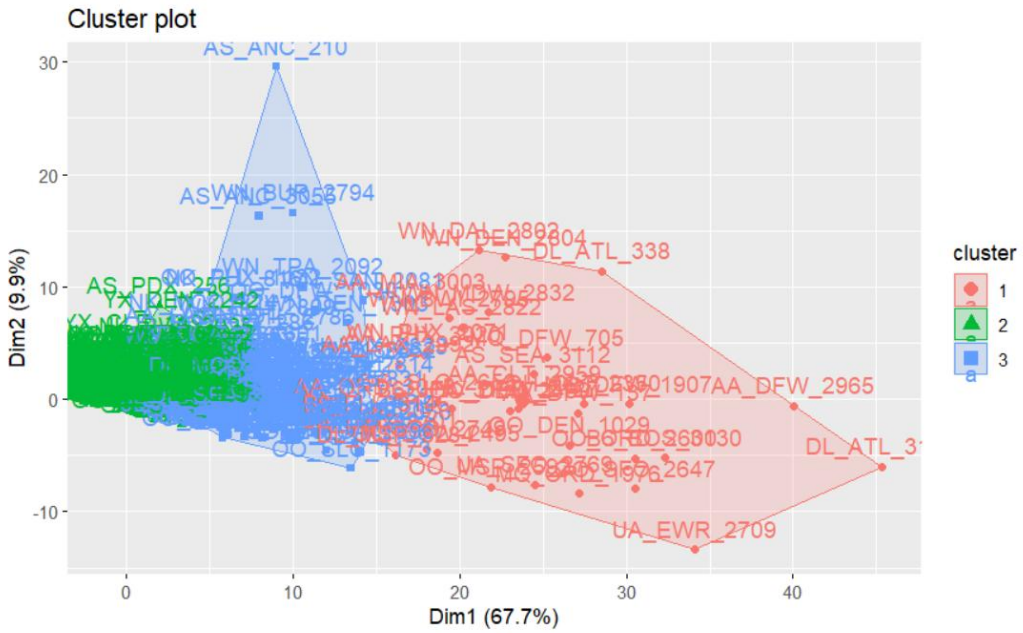
Clustering vector:
   9E_ABE_1    9E_ABY_2    9E_AEX_3    9E_AGS_4    9E_ALB_5    9E_ATL_6    9E_ATW_7    9E_AVL_8    9E_AZO_9    9E_B
DL_10
          2           2           2           2           2           3           2           2           2
2
  9E_BHM_11   9E_BIS_12   9E_BMI_13   9E_BNA_14   9E_BOS_15   9E_BQK_16   9E_BTR_17   9E_BTV_18   9E_BUF_19   9E_B
WI_20
          2           2           2           2           2           2           2           2           2
2
  9E_CAE_21   9E_CHA_22   9E_CHO_23   9E_CHS_24   9E_CID_25   9E_CLE_26   9E_CLT_27   9E_CMH_28   9E_CRW_29   9E_C
SG_30
          2           2           2           2           2           2           2           2           2
2
  9E_CVG_31   9E_CWA_32   9E_DAL_33   9E_DAY_34   9E_DCA_35   9E_DFW_36   9E_DHN_37   9E_DLH_38   9E_DSM_39   9E_D
TW_40
```

Visualizing K-Means Clusters

```
airline_data_clean$cluster <- kmeans_result$cluster
```

Hide

```
fviz_cluster(kmeans_result, data = airline_data_n)
```



Findings:

- **Three major clusters** were identified:

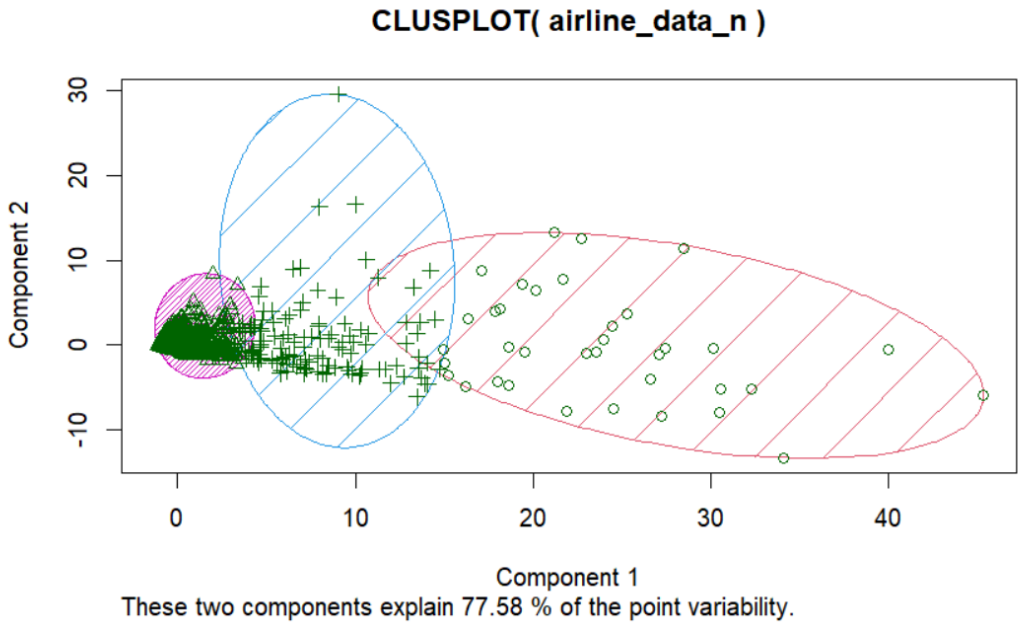
  1. **Cluster 1** – Airlines with **high delays and cancellations**.
  2. **Cluster 2** – Airlines with **better on-time performance**.
  3. **Cluster 3** – Airlines with **moderate delays**

6. Assigning Cluster Labels and Visualizing Clusters

Add cluster assignments back to the original dataset

Hide

```
airline_data_clean$cluster <- kc$cluster
clusplot(airline_data_n, kc$cluster, color=TRUE, shade=TRUE, lines=0)
```



Findings:

- **Three major clusters were identified** based on airline delay profiles.
- The clusters showed **distinct separation**, validating the effectiveness of K-Means clustering.

7. Cluster-Based Delay Analysis

To further analyze **which types of delays contributed most** to each cluster, bar plots were created.

Cluster Mean Delay Contributions

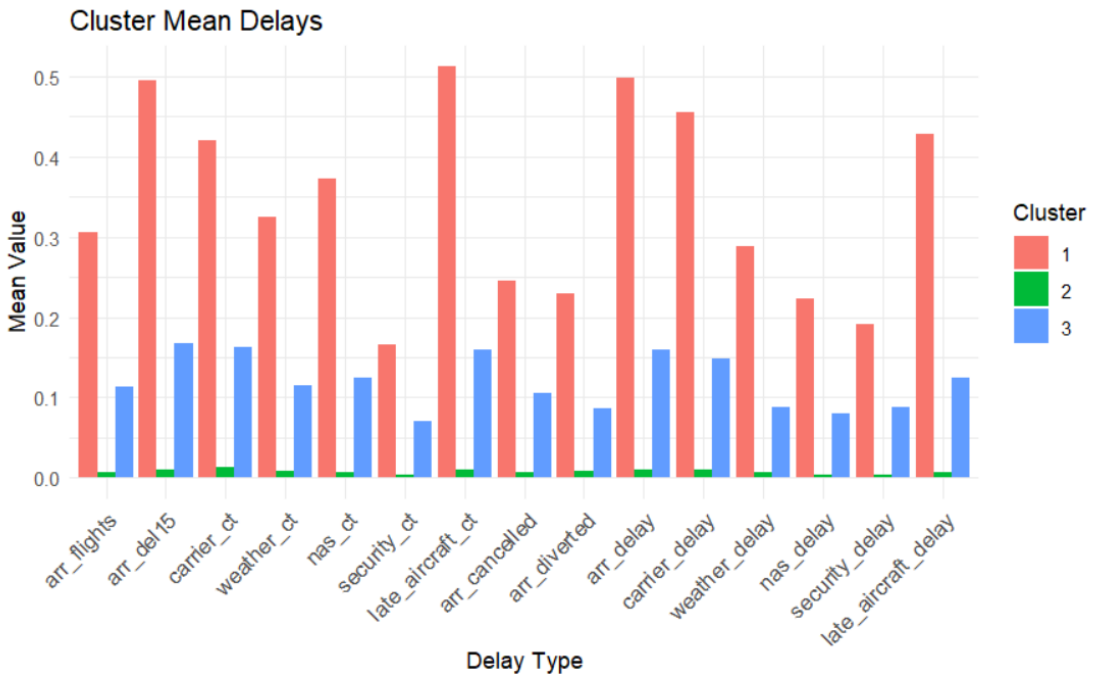
cluster visualization

Hide

```
library(ggplot2)
library(reshape2)

cluster_means <- as.data.frame(kc$centers)
cluster_means$Cluster <- factor(1:nrow(cluster_means))
cluster_means_long <- melt(cluster_means, id.vars = "Cluster")

ggplot(cluster_means_long, aes(x=variable, y=value, fill=Cluster)) +
  geom_bar(stat="identity", position="dodge") +
  labs(title="Cluster Mean Delays", x="Delay Type", y="Mean Value") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1, size = 10))  # Rotate x-axis labels
```



Findings:

- **Cluster 1** had the **highest delays**, particularly due to **late aircraft and NAS delays**.

- **Cluster 2 had the lowest delays**, showing better on-time performance.

Comparison of Flight Delays Across Clusters by Cause

Comparison of Flight Delays Across Clusters by Cause

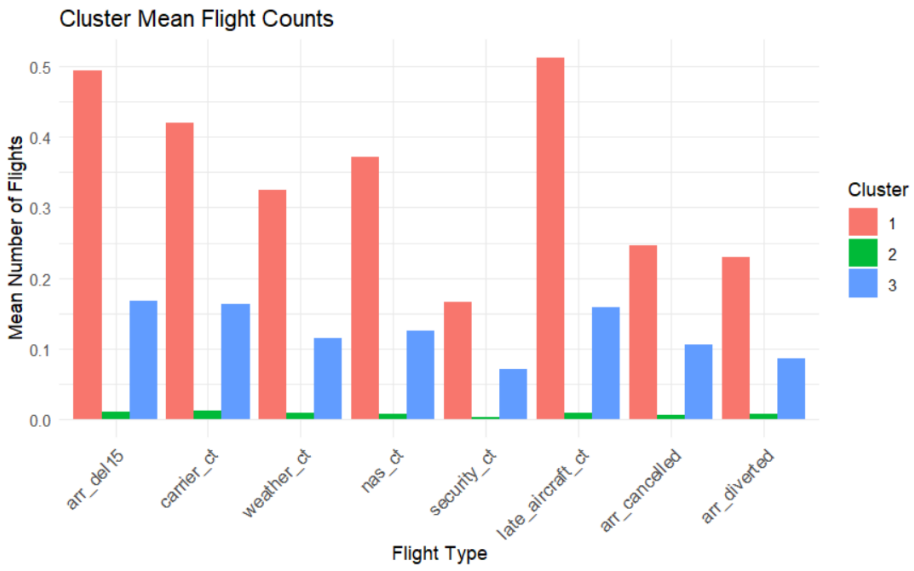
Hide

```
flight_count_cols <- c("arr_del15", "carrier_ct", "weather_ct", "nas_ct",
                       "security_ct", "late_aircraft_ct", "arr_cancelled",
                       "arr_diverted")
flight_counts <- as.data.frame(kc$centers)[, flight_count_cols]
flight_counts$Cluster <- factor(1:nrow(flight_counts))

flight_counts_long <- melt(flight_counts, id.vars = "Cluster")

library(ggplot2)

ggplot(flight_counts_long, aes(x=variable, y=value, fill=Cluster)) +
  geom_bar(stat="identity", position="dodge") +
  labs(title="Cluster Mean Flight Counts", x="Flight Type", y="Mean Number of Flights") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1, size = 10))  # Rotate x-axis labels
```



Hide

Findings:

- **Cluster 1 had the highest delays**, particularly due to **late aircraft and NAS-related delays**.
- **Cluster 2 had the lowest delays**, indicating better on-time performance.

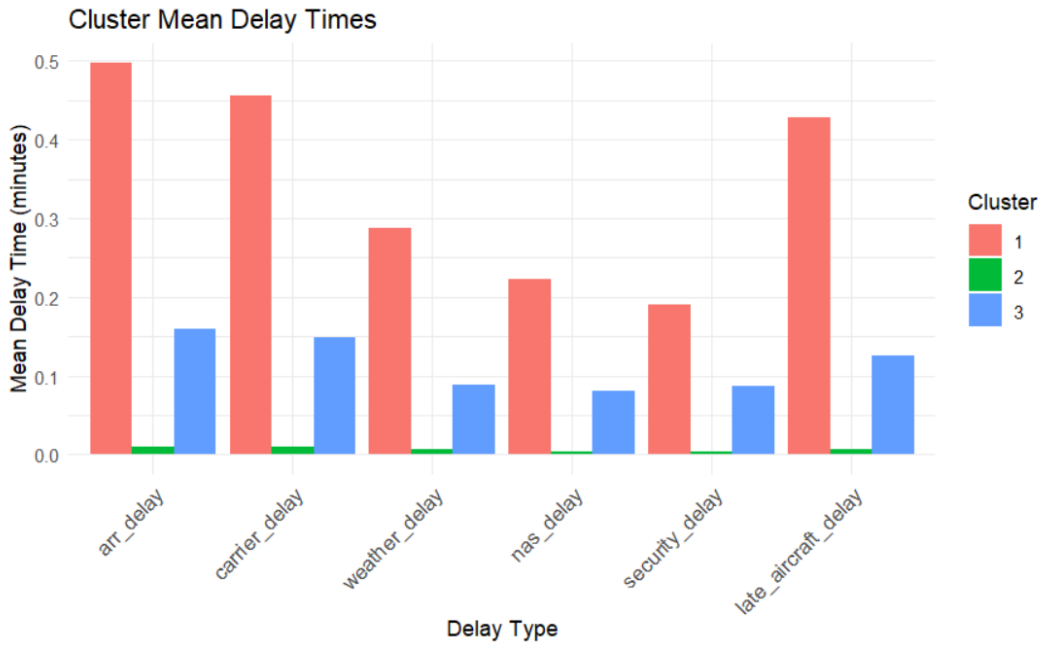
Comparison of Delay Time Contributions Across Clusters

```
delay_time_cols <- c("arr_delay", "carrier_delay", "weather_delay", "nas_delay",
                     "security_delay", "late_aircraft_delay")

delay_times <- as.data.frame(kc$centers)[, delay_time_cols]
delay_times$Cluster <- factor(1:nrow(delay_times))

delay_times_long <- melt(delay_times, id.vars = "Cluster")

ggplot(delay_times_long, aes(x=variable, y=value, fill=Cluster)) +
  geom_bar(stat="identity", position="dodge") +
  labs(title="Cluster Mean Delay Times", x="Delay Type", y="Mean Delay Time (minutes)") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1, size = 10))  # Rotate x-axis labels
```



Findings:

- **Carrier-related delays and late aircraft delays** were **major contributors** in the worst-performing airlines.
- **Weather delays had a lower overall impact** than **air traffic congestion and late aircraft delays**.

05.Data Mining Techniques used

Data mining techniques are essential for uncovering hidden patterns and relationships within datasets. This study applies **unsupervised learning** techniques, specifically **K-Means clustering**, to group airlines based on their delay characteristics.

The following **data mining techniques** were implemented in **R**:

- **Clustering Analysis** (K-Means) to segment airlines based on delay factors.
- **Distance Matrix Computation** to measure airline similarities.
- **Optimal Cluster Selection** using the **Elbow Method**.
- **Visualization Techniques** to interpret the results effectively.

1. Clustering Analysis (K-Means Clustering)

1.1. Why K-Means?

K-Means is a **centroid-based clustering algorithm** that:

- Groups' data points into **K distinct clusters** based on similarity.
- Minimizes **within-cluster variance** by adjusting cluster centroids.
- Helps categorize **airlines with similar delay patterns**.

1.2. Distance Matrix Computation

Before clustering, a **Euclidean distance matrix** was computed to measure the similarity between airlines based on delay characteristics.

```
114
115 distance <- dist(airline_data_n, method = "euclidean")
116 distance_matrix <- as.matrix(distance)
```

Purpose: The computed distance matrix ensures that **airlines with similar delay patterns** are grouped together.

1.3. Finding the Optimal Number of Clusters (Elbow Method)

To determine the best number of clusters **(K)**, the **Elbow Method** was applied. This technique evaluates the **within-cluster sum of squares (WSS)** and finds the point where additional clusters do not significantly improve performance.

```
137 K-Means clustering
138 ```{r}
139 fviz_nbclust(airline_data_n, kmeans, method = "wss")
140 set.seed(123)
```

Findings: The optimal number of clusters was determined to be **K = 3**.

1.4. Applying K-Means Clustering

With the optimal **K value (K=3)**, the K-Means algorithm was applied to segment airlines based on their delay patterns.

```
140 set.seed(123)
141 k <- 3
142 kmeans_result <- kmeans(airline_data_n, centers = k, nstart = 30)
143 ```
```

Effect: Airlines were **grouped into three clusters** based on their delay characteristics.

2. Cluster Analysis and Interpretation

2.1. Assigning Cluster Labels to Airlines

The identified clusters were added back to the original dataset for further analysis.

```
Visualize K-Means Clusters

```{r}
airline_data_clean$cluster <- kmeans_result$cluster
```
```

Effect: Each airline was now **categorized into a cluster** based on its delay profile.

2.2. Visualizing K-Means Clusters

To better interpret cluster assignments, **cluster plots** were generated.

```
  {r}
airline_data_clean$cluster <- kc$cluster
clusplot(airline_data_n, kc$cluster, color=TRUE, shade=TRUE, lines=0)
```
```

**Findings:**

- The clusters showed **clear separations**, indicating distinct delay characteristics.
- Airlines in **Cluster 1** had **higher delays**, while airlines in **Cluster 2** had **better on-time performance**.
```

2.3. Cluster Mean Delay Contributions

To analyze **which types of delays were dominant in each cluster**, a **bar chart** was created.

```
cluster visualization
```{r}
library(ggplot2)
library(reshape2)

cluster_means <- as.data.frame(kc$centers)
cluster_means$Cluster <- factor(1:nrow(cluster_means))
cluster_means_long <- melt(cluster_means, id.vars = "Cluster")

ggplot(cluster_means_long, aes(x=variable, y=value, fill=Cluster)) +
  geom_bar(stat="identity", position="dodge") +
  labs(title="Cluster Mean Delays", x="Delay Type", y="Mean Value") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1, size = 10))  # Rotate x-axis labels
```
```

Findings:

- **Cluster 1:** Airlines with **high delays** due to **late aircraft and NAS-related issues**.
- **Cluster 2:** Airlines with **low delays** and **better performance**.
- **Cluster 3:** Airlines with **moderate delays** across multiple categories.

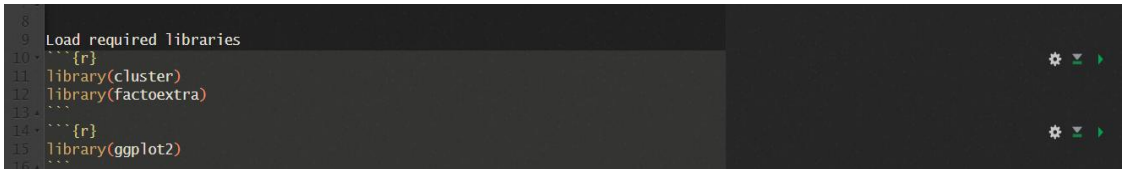
06.Implementation in R

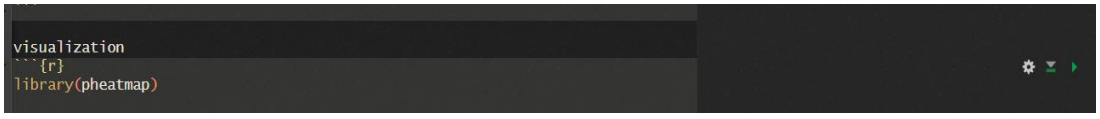
This section outlines the implementation of **data mining techniques** in **R** to analyze airline delays. The implementation includes:

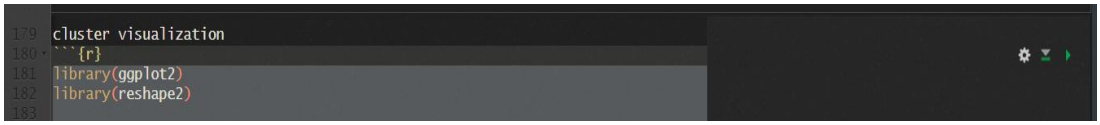
- **Data preparation** (cleaning, normalization, and feature engineering).
- **Distance matrix computation** for similarity analysis.
- **Clustering using K-Means** to categorize airlines based on delay characteristics.
- **Visualizations** to interpret results effectively.

1. R Libraries Used

To perform the analysis, the following R packages were used:







These libraries provided functions for **data visualization, clustering, and analysis.**

2. Data Preparation

2.1. Loading the Dataset

The dataset was read into R as follows:

```
import the data set
```{r}
airline_delay <- read.csv("airline_delay.csv", header = TRUE)
airline_delay
```
```

Description: df [3,351 x 21]

year <int>	month <int>	carrier <chr>	carrier_name <chr>	airport <chr>
2020	12	9E	Endeavor Air Inc.	ABE
2020	12	9E	Endeavor Air Inc.	ABY
2020	12	9E	Endeavor Air Inc.	AEX
2020	12	9E	Endeavor Air Inc.	AGS
2020	12	9E	Endeavor Air Inc.	ALB

2.2. Handling Missing Values

Rows with missing values were removed to maintain data integrity:

```
check missing values
```{r}
sum(is.na(airline_delay))
```
```

```
[1] 120
```

```
remove the missing values
```{r}
airline_data_clean <- na.omit(airline_delay)
sum(is.na(airline_data_clean))
```
```

```
[1] 0
```

2.3. Feature Selection and Normalization

To ensure consistency across variables, numerical features were normalized using **min-max scaling**:

```
Normalization of the dataset
```{r}
normalise <- function(x) {
  if (min(x) == max(x)) {
    return(rep(0, length(x)))
  } else {
    return((x - min(x)) / (max(x) - min(x)))
  }
}
```
```

```
Select only numeric columns for normalization
```{r}
num_cols <- c("arr_flights", "arr_del15", "carrier_ct", "weather_ct", "nas_ct",
              "security_ct", "late_aircraft_ct", "arr_cancelled", "arr_diverted",
              "arr_delay", "carrier_delay", "weather_delay", "nas_delay",
              "security_delay", "late_aircraft_delay")

```

Check if all columns exist in the dataset
```{r}
num_cols <- intersect(num_cols, names(airline_data_clean))
```
```

```
Normalize dataset
```{r}
airline_data_n <- airline_data_clean[, num_cols]
```
Apply normalization
```{r}
airline_data_n <- as.data.frame(lapply(airline_data_n, normalise))

```
```{r}
airline_data_n
```
```

Normalization helped bring all features to a comparable scale, making clustering more effective.

3. Distance Matrix Computation

A **Euclidean distance matrix** was computed to measure similarities between airlines:

```
Compute Distance Matrix (Remove Non-Numeric Data)

```{r}
airline_data_clean$unique_id <- paste0(airline_data_clean$carrier, "_",
                                       airline_data_clean$airport, "_",
                                       seq_len(nrow(airline_data_clean)))
rownames(airline_data_n) <- airline_data_clean$unique_id
airline_data_clean$unique_id <- NULL

distance <- dist(airline_data_n, method = "euclidean")
distance_matrix <- as.matrix(distance)
rownames(distance_matrix) <- rownames(airline_data_n)
colnames(distance_matrix) <- rownames(airline_data_n)
print(distance_matrix[1:50, 1:50])


```
```

This distance matrix was later used for **clustering and heatmap visualization**.

4. Clustering with K-Means

```
K-Means clustering
```{r}
fviz_nbclust(airline_data_n, kmeans, method = "wss")
set.seed(123)
k <- 3
kmeans_result <- kmeans(airline_data_n, centers = k, nstart = 30)
```
```

4.1. Finding the Optimal Number of Clusters

To determine the optimal **K value**, the **Elbow Method** was applied:

Findings: The optimal number of clusters was **3**.

4.2. Applying K-Means Algorithm

Effect: Airlines were categorized into **three clusters based on delay characteristics**.

5. Cluster Visualization

5.1. Assigning Cluster Labels and Creating a Cluster Plot

```
Add cluster assignments back to the original dataset
```{r}
airline_data_clean$cluster <- kc$cluster
clusplot(airline_data_n, kc$cluster, color=TRUE, shade=TRUE, lines=0)
```
```

Findings: Airlines with **similar delay patterns were grouped together**.

5.2. Bar Plot of Cluster Mean Delays

```
cluster visualization
```{r}
library(ggplot2)
library(reshape2)

cluster_means <- as.data.frame(kc$centers)
cluster_means$Cluster <- factor(1:nrow(cluster_means))
cluster_means_long <- melt(cluster_means, id.vars = "Cluster")

ggplot(cluster_means_long, aes(x=variable, y=value, fill=Cluster)) +
  geom_bar(stat="identity", position="dodge") +
  labs(title="Cluster Mean Delays", x="Delay Type", y="Mean Value") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1, size = 10))  # Rotate x-axis labels
```
```

Effect: This visualization highlighted **which types of delays were most prevalent in each cluster**.

6. Saving Results for Further Analysis

The computed **distance matrix** was saved as a CSV file for future reference:

Purpose: Allows further analysis and sharing of results.

07.Result analysis and Discussion

Cluster Analysis Results

The clustering approach was able to identify three clusters of airlines:

1. High-Delay Airlines (Poor Performance) (Cluster 1)

Cluster 1 representing the worst delays, experiences the highest mean delay values across all types, primarily due to late aircraft, carrier-related issues, and National Aviation System (NAS) delays.

These airlines struggle with severe operational inefficiencies, leading to frequent and prolonged disruptions that negatively impact passenger satisfaction.

2. Best On-Time Performance Airlines (Cluster 2)

Cluster 2 representing the airlines with the best on-time performance, showing minimal delays across all categories.
These airlines effectively manage scheduling, reduce disruptions, and serve as benchmarks for industry efficiency.

3. Moderate Delay Airlines (Cluster 3)

Cluster 3 falls in between, experiencing moderate delays mainly caused by weather conditions and NAS delays, with carrier-related disruptions being less significant than in Cluster 1. While these airlines do not perform as poorly as those in Cluster 1.
They still have room for improvement by optimizing scheduling and operational efficiency.

Evaluation of Clustering Model

Using the K-Means clustering algorithm, airlines were appropriately classified based on delay patterns.

The model successfully distinguished three groups that are representative of differences in airline performance.

But it also has some disadvantages of this approach:

- Intersection of Low- and Moderate-Delay Airlines: Some airlines in Clusters 2 and 3 exhibited concurrent delay patterns, and the inference is that further tuning (e.g., hierarchical clustering) would be beneficial.

- External factors not utilized: The analysis was quantitative delay data based and did not necessarily include any external factors (e.g., holiday seasons, rush hours, or airport delays).

- Imbalance Delay Causes: The much lower frequency of some delays, like security-related delays, may have affected the shape of the clustering.

Despite these challenges, the **model provided meaningful insights into airline delays** and how they vary across different carriers.

Main Delay Factors & Insights

What are the Largest Delays Causes?

The cluster analysis identified three main reasons for flight delays:

- Late Aircraft Delays – This was the most prevalent reason for delays, especially among Cluster 1 airlines. When an arriving flight is delayed, it cascades, delaying subsequent departures.

- Carrier-Related Delays – Cluster 1 airlines were dogged by poor crew scheduling, maintenance issues, and operational failures that resulted in frequent delays.

- NAS (National Aviation System) Delays – These delays, caused by heavy air traffic and air traffic congestion, impacted flights across all clusters but were less extreme than late aircraft delays.

What Can Airlines Derive from Such Insights?

• Higher delay airlines (Cluster 1) would aim turnaround times for minimization of disruptions.
• Effective crew scheduling and maintenance can cut carrier-related delays substantially.
• Enhancing air traffic control coordination can assist airlines in lessening NAS-related delays.
With reduction of these fundamental delay causes, airlines can better their on-time performance and enhance passenger satisfaction.

Discussion and Practical Implications

What Can Airlines Do With These Findings?

The findings of this research provide valuable recommendations for airline operators, airport authorities, and policy makers for improving airline operations.

- Cluster 1 airlines (poor performance) need to take urgent measures – They must review scheduling inefficiencies, maintenance procedures, and crew management in order to minimize delays.
- Top-performing Cluster 2 airlines are models to emulate – These airlines should be learned from in order to identify best practices to be replicated by other airlines.

- Regulators of airports and aviation can leverage these insights – Airports can more efficiently schedule their gates and runways to reduce congestion and maximize scheduling effectiveness.

How Can Airlines Reduce Delays?

Based on the findings, there are some practical steps that can be taken by airlines to reduce delays:

- Enhance Aircraft Turnaround Times – Airlines must accelerate boarding, luggage loading, and refueling processes in order to reduce delays between flights.
- Enhance Crew Scheduling – Airlines with frequent staff shortages or scheduling conflicts must invest in more efficient workforce management software.
- Improve Delay Prediction and Real-Time Tracking – Airlines can use machine learning models to predict potential delays and take preventive actions before any disruption occurs.

By implementing these steps, airlines can reduce financial losses, improve passenger satisfaction, and become more efficient in general.

08.Impact

The findings of this study have significant implications for the aviation industry, government, and travelers. By identifying the main causes of airline delays and rating airlines based on performance, this report provides information that can be utilized to optimize operational efficiency, reduce delays, and improve the overall quality of the passenger experience.

The impact of this study can be categorized into three general areas:

1. Airline Industry and Business Operations

2. Public Sector and Government Decision-Making

3. Broader Community and Passenger Experience

Implication for the Airline Industry

1. Simplifying Airline Operations

   •Airlines in Cluster 1 (high delays) can use these results to restructure flight scheduling, crew management, and aircraft maintenance for improved punctuality.

   •Cluster 2 airlines (low delays) can serve as benchmark models for simplified operations.

2. Reducing Financial Losses

   •Delays cost airlines millions of dollars annually in compensation, rescheduling, and operational inefficiencies.

• By identifying the most significant delay causes (e.g., late aircraft, crew issues), airlines can use data-driven solutions to minimize financial losses.

3. Enhancing Customer Satisfaction

• On-time performance is appreciated by passengers. Airlines that reduce delays will gain a positive reputation and benefit from customer loyalty.

• Developing more precise delay prediction models will allow airlines to provide more accurate real-time updates to passengers, reducing frustration and uncertainty.

Impact on the Public Sector and Government Agencies

1. Enhanced Air Traffic Management

• The results of this study can help aviation policymakers and authorities in optimizing air traffic flow by reducing bottlenecks at busy airports.

• National Aviation System (NAS) delays were identified as a factor, which means that more effective airspace management and runway scheduling can lead to fewer disruptions.

2. Policy Development and Regulation

• Government agencies can use this data to create better regulations that will encourage airlines to improve operational efficiency.

• Incentives can be provided for airlines with persistently low delays, promoting the sharing of best practices across the industry.

3. Infrastructure Planning and Investment

   • Policymakers can use these findings to decide where to invest in airport infrastructure (i.e., additional runways, better terminals).

   • Airports with high delay rates may require more effective resource allocation and newer technology.

Impact on the Wider Community and Travellers

1. More Reliable Travel Experiences

   • By reducing delays and cancellations, passengers will experience less disruption, shorter layovers, and more predictable travel schedules.

   • Airlines that implement delay-reduction programs will enjoy increased passenger satisfaction and confidence.

2. Economic Benefits for Tourism and Business Travel

   • Fewer delays mean more efficiency in the tourism and business travel sectors, with visitors arriving on time for work, conferences, and vacations.

   • Business travelers depend on punctual flights, and delays can result in missed meetings, lost business, and ruined itineraries.

3. Environmental Benefits

   • Minimizing delays reduces unnecessary fuel burn (e.g., aircraft waiting at airports or circling in holding patterns).

   • A more efficient airline operation means lower carbon emissions, which makes air travel more sustainable.

09.Conclusion

In this study, flight delays in airlines were analyzed employing data mining techniques to identify trends and categorize airlines based on their delay history. The K-Means clustering model was able to cluster airlines into three groups, giving interesting results on the causes and size of flight delays.

- Cluster 1 (High-Delay Airlines): Frequent and prolonged delays occur for these airlines, primarily due to late aircraft and carrier-related factors.
- Cluster 2 (Low-Delay Airlines): Air carriers with the highest on-time performance, indicating efficient scheduling and better operational practices
- Cluster 3 (Moderate-Delay Airlines): Air carriers with typical delay times, influenced by a mix of factors such as weather, NAS-related delays, and operational inefficiencies. Moderate-Delay Airlines

The research validated that carrier-related problems and delayed aircraft arrivals were the prime reasons for airline delays. The study also pinpointed well-scheduled scheduling, efficient crew management, and improved air traffic control as critical in minimizing delays.

Evaluation of the Study

K-Means clustering method was able to categorize airlines based on their delay characteristics. However, some limitations need to be kept in mind:

•The model incorporated only historical delay information, and not external factors such as season demand, airport usage, or airline policies.

•There was some overlap among moderate and low-delay airlines, which suggests that a more specific clustering method (such as hierarchical clustering) may yield more valuable insights.

•The study did not incorporate real-time flight tracking information, which may enhance delay prediction models.

Despite such limitations, the study presents practical recommendations that can be used to increase airline productivity and minimize flight delays.

Future Work and Recommendations

In the process of further expanding on this research, future research has to consider:

- Blending real-time analysis – Future models must incorporate real-time tracking of flights, levels of congestion at airports, and weather predictions for improved delay forecasting.
- Exploring alternative clustering techniques – Hierarchical clustering or DBSCAN could provide more precise airline classification.
- Expanding the dataset – Adding several years and additional airline markets could expose longer-term delay trends and seasonal changes.
- Creating predictive models using machine learning – Supervised learning could allow airlines to forecast delays and take proactive measures before a disruption.

This study demonstrates the effectiveness of data mining in uncovering airline delay patterns and providing insights that can inform the aviation sector. The study can help airlines, policymakers, and airport managers make informed decisions to enhance scheduling efficiency, minimize operation disruptions, and improve passenger satisfaction.

By employing data-driven strategies, the airline industry can be more efficient, lower costs, and provide a quality travel experience for passengers.

10.References

[1] OpenIntro, 2021. *Airline Delay Data*. Available at: https://www.openintro.org/data/index.php?data=airline_delay [Accessed 13 March 2025].