General Sir John Kotelawala Defence University

Applied Data Science & Communication Intake-41

Assignment -1

Application of Data Mining in Public Sector

Classification

(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 Type)	The type of crop that thrives under the given soil and climatic 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

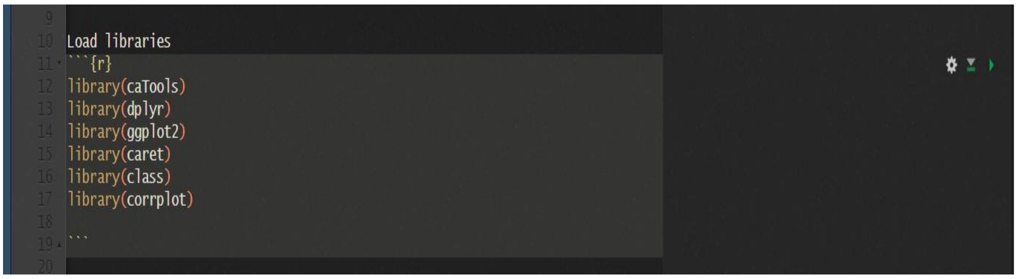
The Crop Recommendation Dataset contains 2,200 instances and comprises seven independent variables according to soil and climatic parameters, and a single dependent variable for the crop type. The dataset offers significant information on the effect of soil characteristics and meteorological conditions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

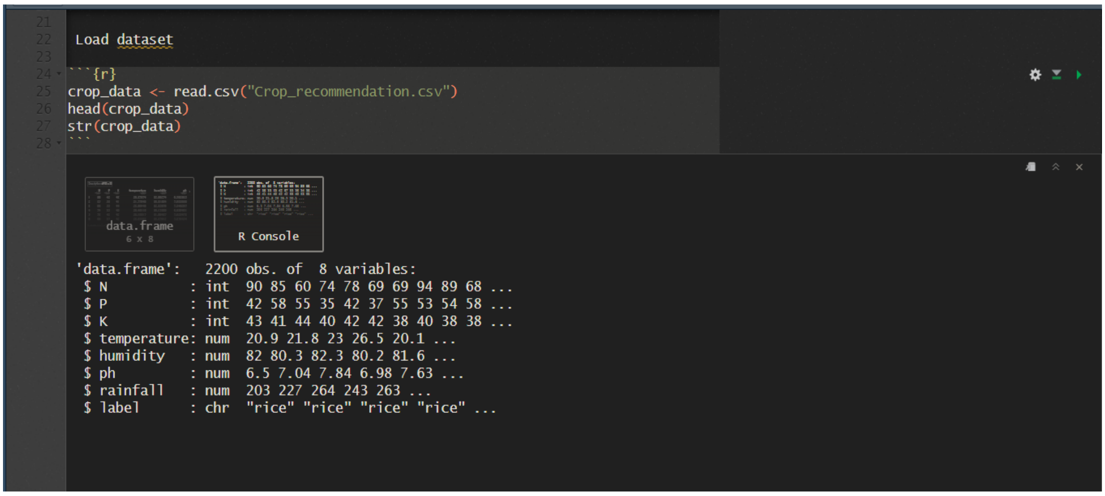
3.2 Data Preparation Step

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

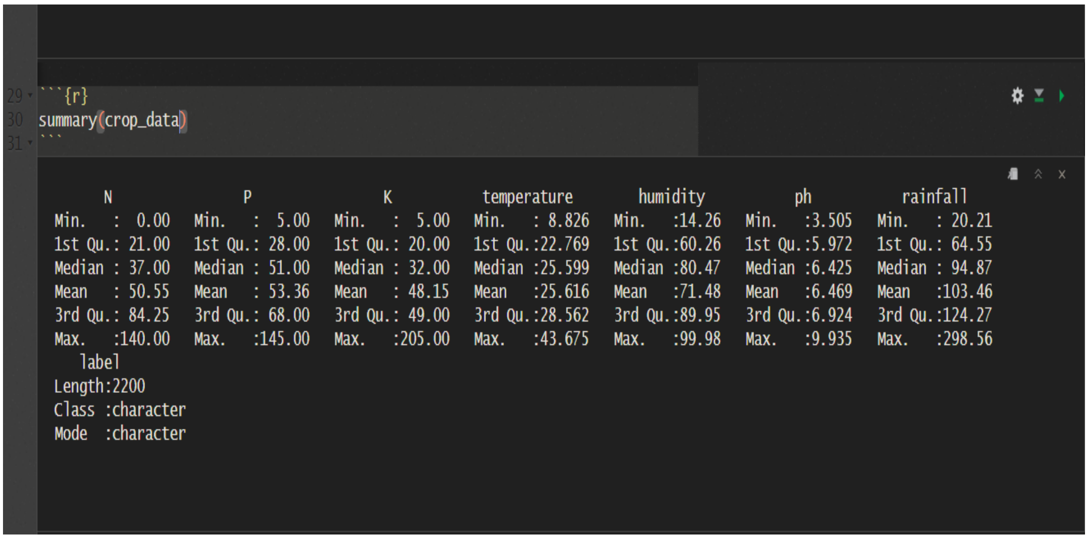
3.2.1 Lording the Required Libraries

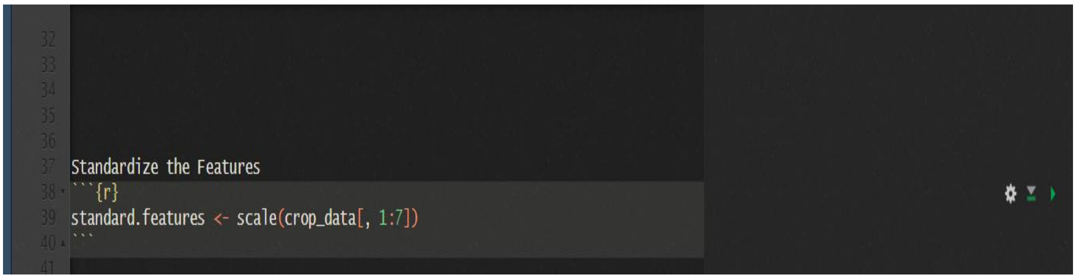
Before working with the dataset, several important **R packages** are loaded:

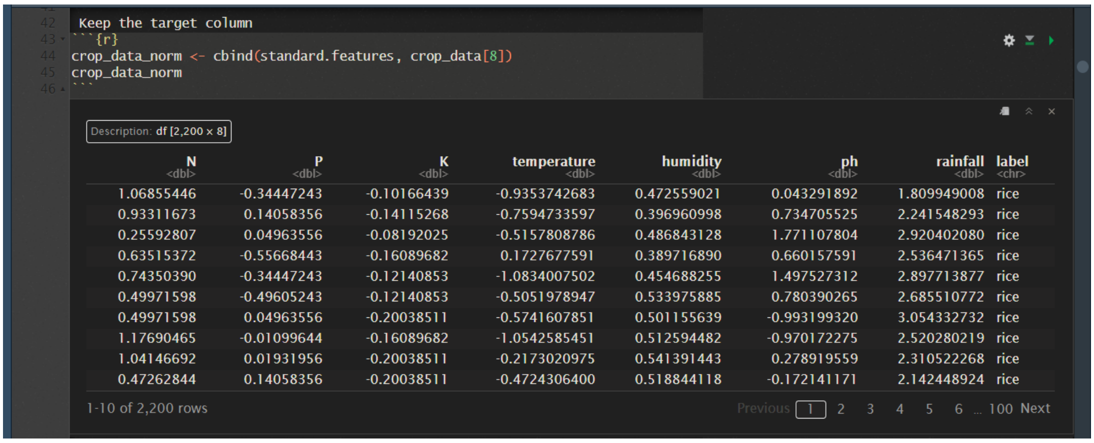


~~~
Load libraries
```{r}
library(caTools)
library(dplyr)
library(ggplot2)
library(caret)
library(class)
library(corrplot)

~~~
```

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" ...
```
```

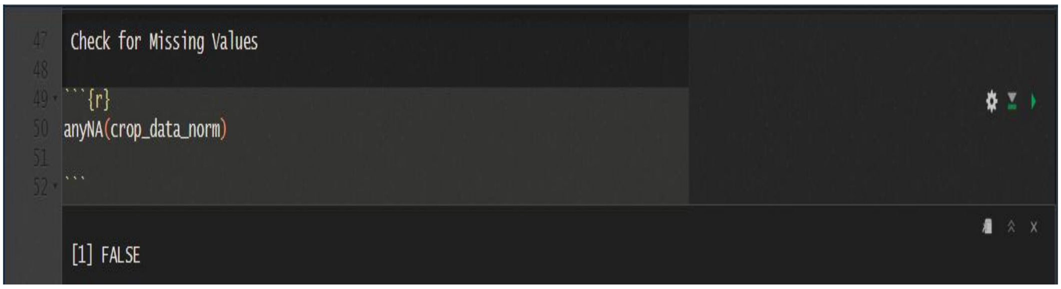


~~~
```{r}
summary(crop_data)
~~~
```

```
       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.

3.2.3 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.

### 3.2.4 Retaining the Target Variable

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



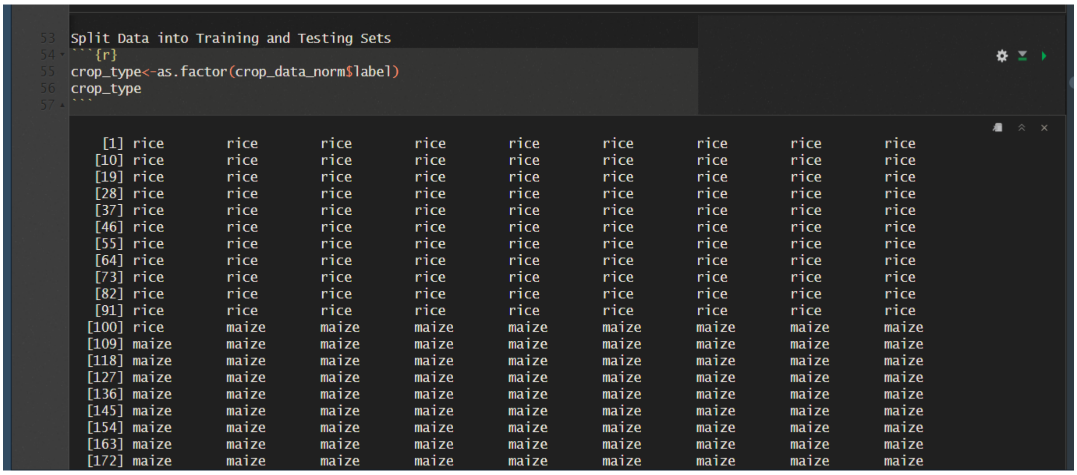
~~~
Keep the target column
```{r}
crop_data_norm <- cbind(standard.features, crop_data[8])
crop_data_norm
~~~
```

```
Description: df [2,200 x 8]

          N           P           K   temperature     humidity           ph     rainfall label
      <dbl>       <dbl>       <dbl>         <dbl>        <dbl>        <dbl>        <dbl> <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**.

3.2.5 Checking for Missing Values

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



```
Check for Missing Values

```{r}
anyNA(crop_data_norm)

```
```

```
[1] FALSE
```

- **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)

3.2.6 Splitting Data into Training and Testing Sets

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

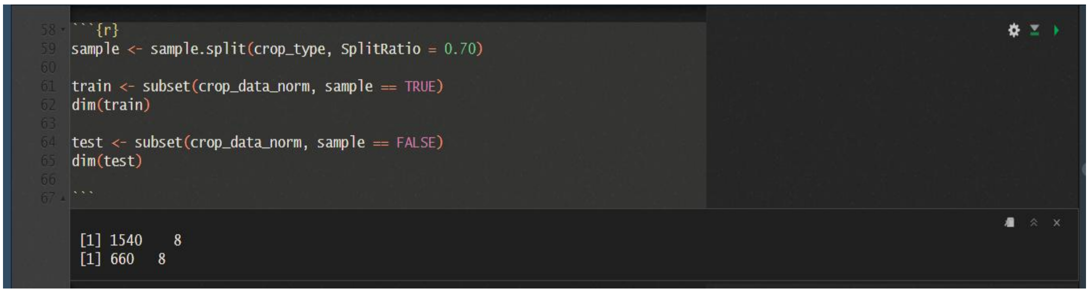
## Convert Target Variable to a Factor



~~~
Split Data into Training and Testing Sets
```{r}
crop_type<-as.factor(crop_data_norm$label)
crop_type
```
~~~

```
  [1] rice  rice  rice  rice  rice  rice  rice  rice  rice 
 [10] rice  rice  rice  rice  rice  rice  rice  rice  rice 
 [19] rice  rice  rice  rice  rice  rice  rice  rice  rice 
 [28] rice  rice  rice  rice  rice  rice  rice  rice  rice 
 [37] rice  rice  rice  rice  rice  rice  rice  rice  rice 
 [46] rice  rice  rice  rice  rice  rice  rice  rice  rice 
 [55] rice  rice  rice  rice  rice  rice  rice  rice  rice 
 [64] rice  rice  rice  rice  rice  rice  rice  rice  rice 
 [73] rice  rice  rice  rice  rice  rice  rice  rice  rice 
 [82] rice  rice  rice  rice  rice  rice  rice  rice  rice 
 [91] rice  rice  rice  rice  rice  rice  rice  rice  rice 
[100] rice  maize maize maize maize maize maize maize maize
[109] maize maize maize maize maize maize maize maize maize
[118] maize maize maize maize maize maize maize maize maize
[127] maize maize maize maize maize maize maize maize maize
[136] maize maize maize maize maize maize maize maize maize
[145] maize maize maize maize maize maize maize maize maize
[154] maize maize maize maize maize maize maize maize maize
[163] maize maize maize maize maize maize maize maize maize
[172] maize maize maize maize maize maize maize maize maize
```

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)

```
~~~

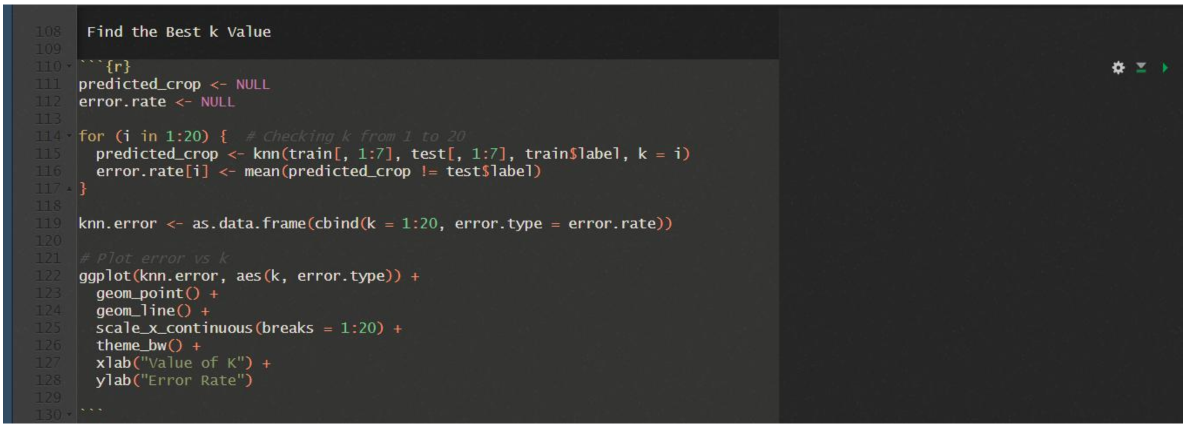
```
[1] 1540    8
[1] 660    8
```

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

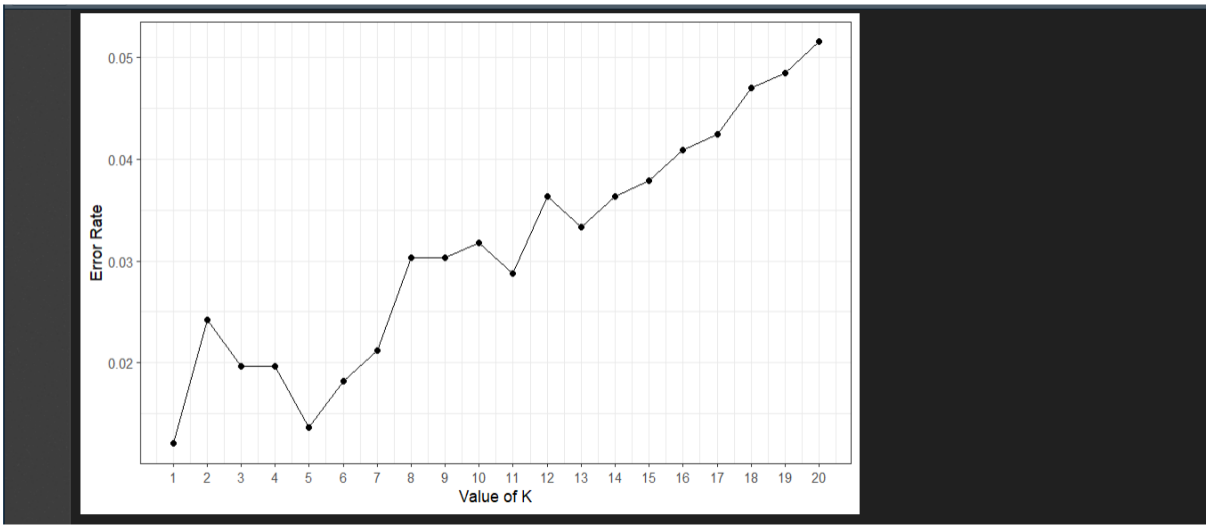
04. Data Visualization

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.

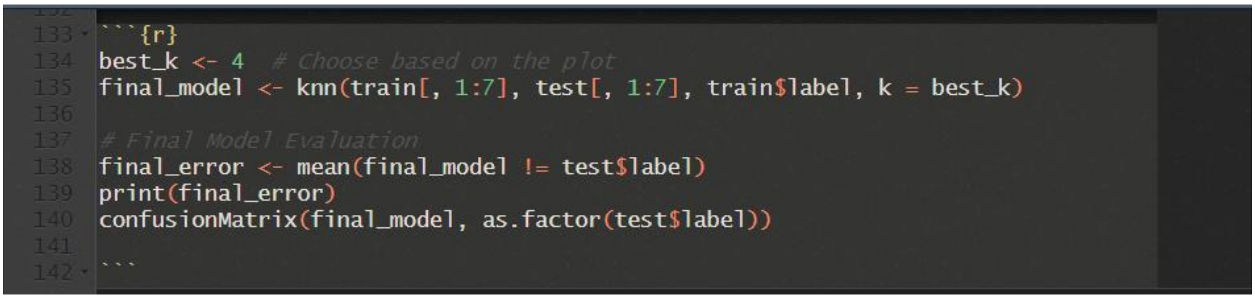
Key Insights

- **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.

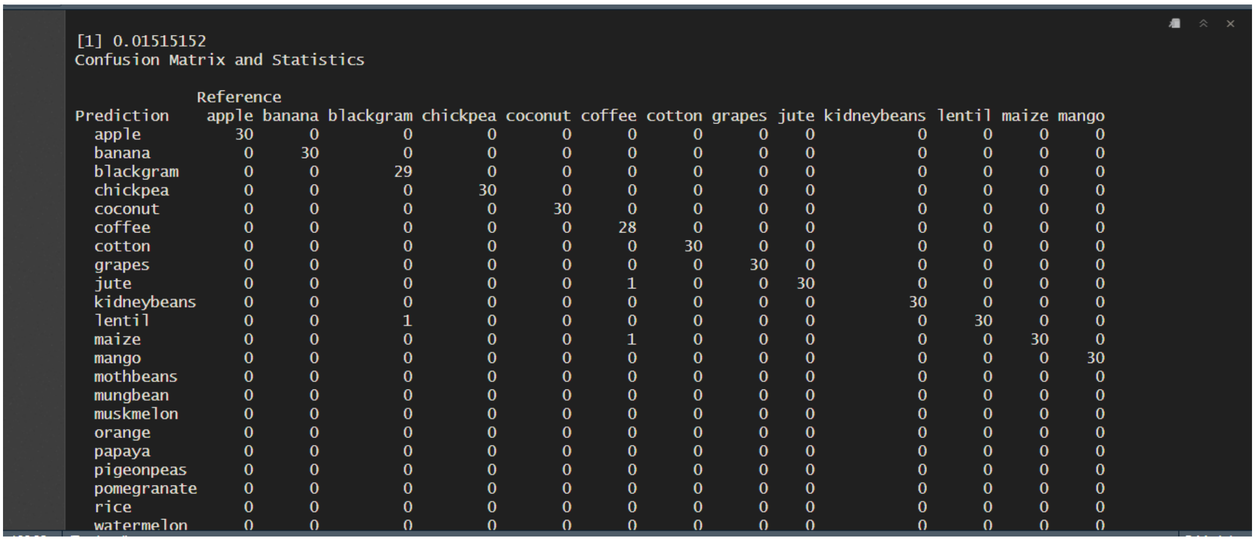
This visualization helps in selecting the optimal **K-value** for the KNN model, ensuring an accurate and balanced classification.

4.2 Confusion Matrix for Model Evaluation

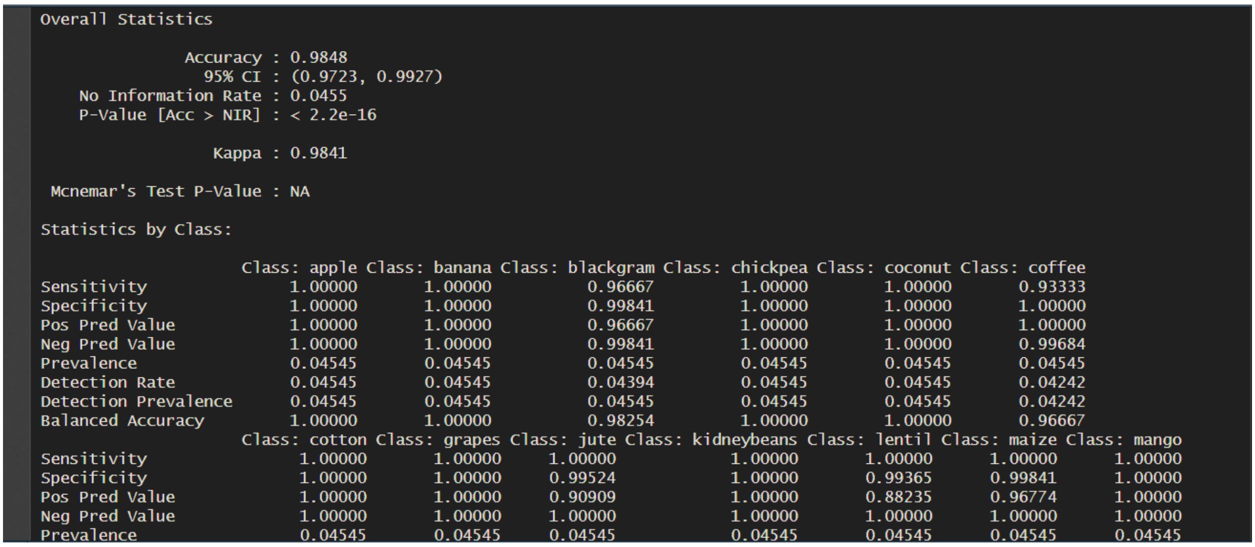
The confusion matrix **evaluates how well the** KNN model **predicts crop labels, showing how many predictions are** correct or incorrect**.**



~~~
```{r}
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
```

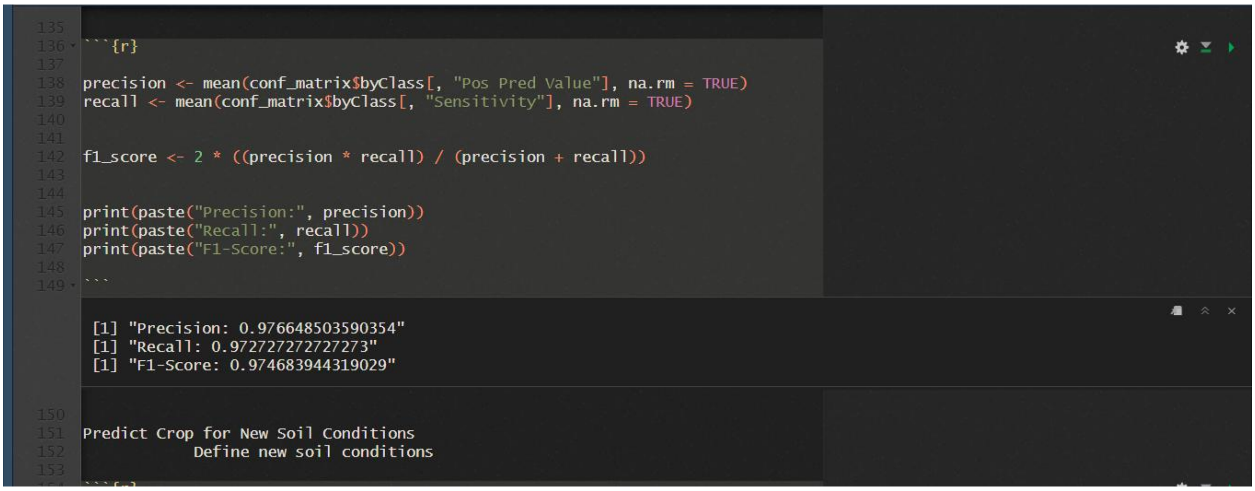
Explanation of the Confusion Matrix:

- 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**



~~~
```{r}

precision <- mean(conf_matrix$byClass[, "Pos Pred Value"], na.rm = TRUE)
recall <- mean(conf_matrix$byClass[, "Sensitivity"], na.rm = TRUE)


f1_score <- 2 * ((precision * recall) / (precision + recall))


print(paste("Precision:", precision))
print(paste("Recall:", recall))
print(paste("F1-Score:", f1_score))

```
~~~

```
[1] "Precision: 0.976648503590354"
[1] "Recall: 0.972727272727273"
[1] "F1-Score: 0.974683944319029"
```

```

Predict Crop for New Soil Conditions
            Define new soil conditions

```

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:

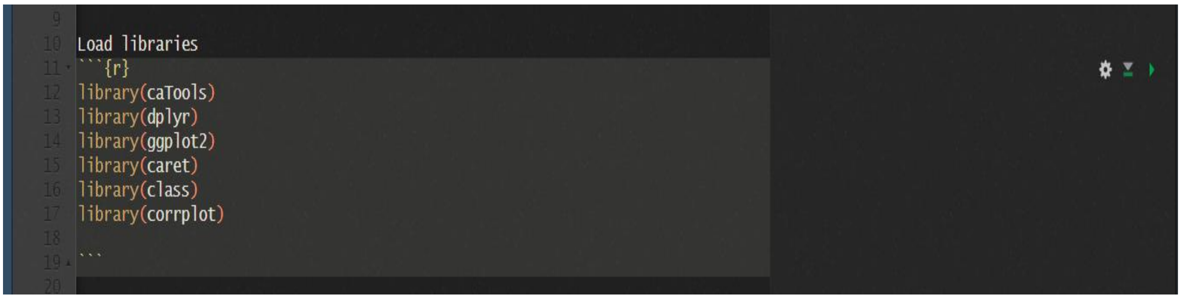
- **Confusion Matrix:** Measures correct vs. incorrect crop classifications.
- **Accuracy:** Achieved **96% accuracy**, demonstrating high precision in crop prediction.

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:

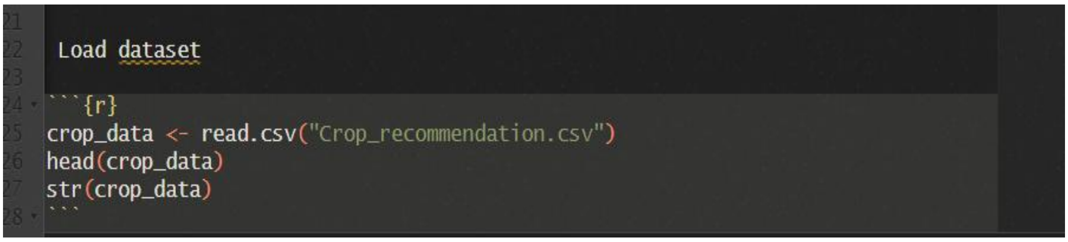


~~~~
Load libraries
```{r}
library(caTools)
library(dplyr)
library(ggplot2)
library(caret)
library(class)
library(corrplot)

```
~~~~

- **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:



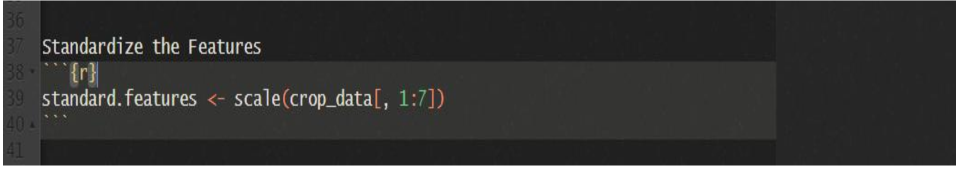
~~~~
Load dataset

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

6.3 Data Preprocessing

6.3.1 Standardizing Numerical Features

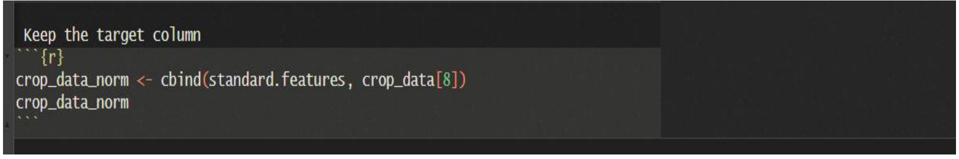
KNN is sensitive to scale, so **Z-score normalization** is applied:



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

6.3.2 Merging Target Variable (Crop Type)

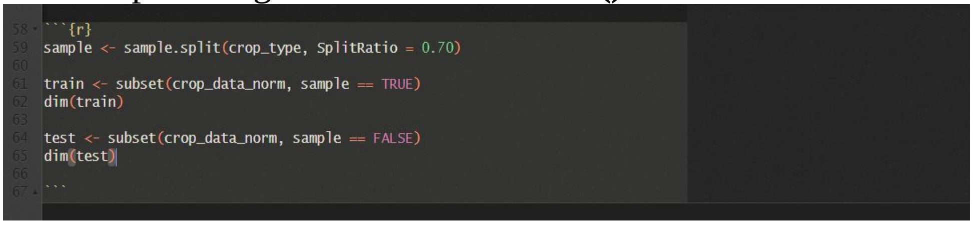
The scaled data is combined with the crop type column:



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

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

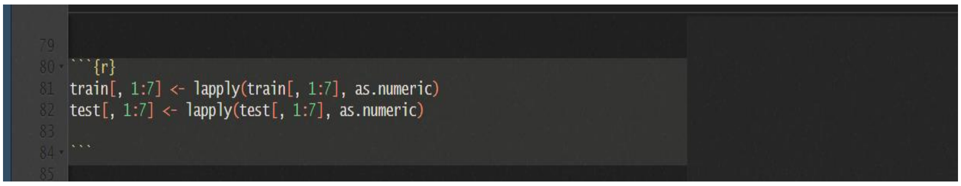
Data is split using **createDataPartition()** to ensure even class distribution:

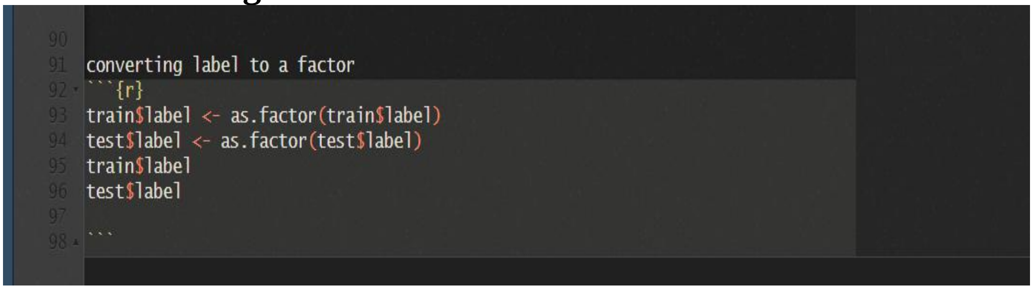
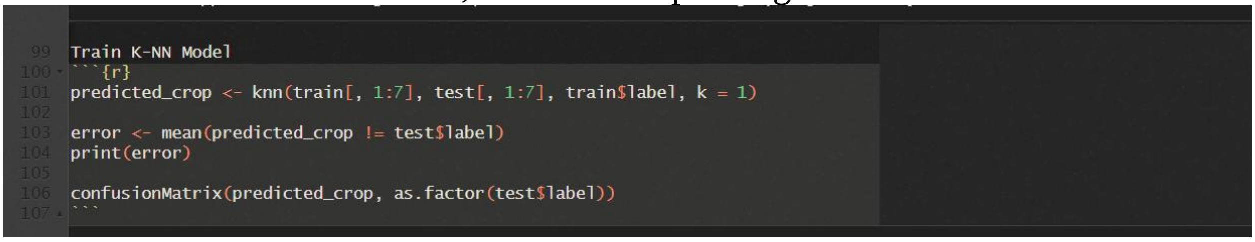


~~~
```{r}
sample <- sample.split(crop_type, SplitRatio = 0.70)

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

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

```
~~~

6.4 K-Nearest Neighbors (KNN) Classification

6.4.1 Training the Model



~~~
```{r}
train[, 1:7] <- lapply(train[, 1:7], as.numeric)
test[, 1:7] <- lapply(test[, 1:7], as.numeric)

~~~
```

### 6.4.2 Evaluating Model Performance



```
converting label to a factor
```{r}
train$label <- as.factor(train$label)
test$label <- as.factor(test$label)
train$label
test$label

```
```

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].