AMERICAN INTERNATIONAL UNIVERSITY-BANGLADESH

408/1, Kuratoli, Khilkhet, Dhaka 1229, Bangladesh

Assignment Title:	FINAL TERM PROJECT REPORT		
Assignment No:	01	Date of Submission:	15/08/2023
Course Title:	INTORODUCTION TO DATA SCIENCE		
Course Code:	01153	Section:	C
Semester:	Summer 2022 - 23	Course Teacher:	DR. ABDUS SALAM

Declaration and Statement of Authorship:

1. I/we hold a copy of this Assignment/Case-Study, which can be produced if the original is lost/damaged.
2. This Assignment/Case-Study is my/our original work and no part of it has been copied from any other student's work or from any other source except where due acknowledgement is made.
3. No part of this Assignment/Case-Study has been written for me/us by any other person except where such collaborationhas been authorized by the concerned teacher and is clearly acknowledged in the assignment.
4. I/we have not previously submitted or currently submitting this work for any other course/unit.
5. This work may be reproduced, communicated, compared and archived for the purpose of detecting plagiarism.
6. I/we give permission for a copy of my/our marked work to be retained by the Faculty for review and comparison, including review by external examiners.
7. I/we understand thatPlagiarism is the presentation of the work, idea or creation of another person as though it is your own. It is a formofcheatingandisaveryseriousacademicoffencethatmayleadtoexpulsionfromtheUniversity. Plagiarized material can be drawn from, and presented in, written, graphic and visual form, including electronic data, and oral presentations. Plagiarism occurs when the origin of them arterial used is not appropriately cited.
8. I/we also understand that enabling plagiarism is the act of assisting or allowing another person to plagiarize or to copy my/our work.

* *Student(s) must complete all details except the faculty use part.*
** Please submit all assignments to your course teacher or the office of the concerned teacher.

Group Name/No.:

No	Name	ID	Program	Signature
1	MD QUANET UL AHKAM ROKON	20-43582-1	BSc [CSE]	Quanetul
2				
3				
4				
5				
6				
7				
8				
9				
10				

Faculty use only		
FACULTYCOMMENTS	Marks Obtained	
	Total Marks	

Title: K-Nearest Neighbors (KNN) Classification Report

Introduction:

The introduction serves as a brief overview of the project, setting the context for the analysis and explaining the purpose of the report. It provides essential background information about the dataset and the problem aim to address with K-Nearest Neighbors (KNN) classification.

In this report, we delve into the application of K-Nearest Neighbors (KNN) classification using a glass composition dataset (https://www.kaggle.com/datasets/uciml/glass). The dataset encompasses various attributes, including refractive index, elemental content (e.g., sodium, magnesium), and type of glass. Our objective is to leverage KNN to classify glass types accurately based on their composition attributes. We will navigate through data preprocessing, correlation analysis, model training, and evaluation using a combination of training-test split and cross-validation. Furthermore, we will explore visualizations to depict class distributions and correlations. By the end of this report, we will have gained insights into the effectiveness of the KNN algorithm for glass type classification and its performance in terms of recall, precision, and accuracy.

Data Preparation:

The following steps were performed for data preparation:

1. Reading and Renaming Columns:

The dataset was read from the provided path and column names were renamed.

```
> library(tidyverse)
> base_glass<-read_csv("F:/IDS FINAL/glass.csv")
Rows: 499 Columns: 10
── Column specification ──────────────────────────────────────────
Delimiter: ","
dbl (10): RI, Na, Mg, Al, Si, K, Ca, Ba, Fe, Type

> head(base_glass)
# A tibble: 6 × 10
  `refractive index`  sodium magnesium aluminium silicon potassium calcium barium   iron type
               <dbl>   <dbl>     <dbl>     <dbl>   <dbl>     <dbl>   <dbl>  <dbl>  <dbl> <fct>
1             1.61    -0.358      2.06    -1.57   -2.07     -0.415 -0.0182  -1.28 -0.831 1
2            -0.0171   0.0144     1.52    -0.996  -0.531     0.470 -0.989   -1.28 -0.831 1
3            -0.703   -0.522      1.49    -0.599  -0.110     0.280 -1.04    -1.28 -0.831 1
4             0.00686 -0.998      1.57    -1.15   -0.726     0.659 -0.577   -1.28 -0.831 1
5            -0.108   -0.909      1.53    -1.26    0.0363    0.617 -0.736   -1.28 -0.831 1
6            -0.808   -1.62       1.52    -0.422  -0.142     0.807 -0.736   -1.28  3.24  1
> 
```

2. Converting Target Variable to Factor:

The 'type' variable was converted to a factor.

```
> names(base_glass)<-c("refractive index","sodium","magnesium","aluminium","silicon","potassium","calcium","barium","iron","type")
> base_glass$`type`<-as_factor(base_glass$`type`)
> base_glass$`type`
  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [51] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
[101] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3 3
[151] 3 3 3 3 3 3 3 3 3 3 3 3 3 5 5 5 5 5 5 5 5 5 5 5 5 5 6 6 6 6 6 6 6 6 6 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7
[201] 7 7 7 7 7 7 7 7 7 7 7 7 7 7 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6
[251] 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6
[301] 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 7 7 7 7 7 7 7 7 7 7 7 7 7
[351] 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7
[401] 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
[451] 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
Levels: 1 2 3 5 6 7
```

3. Handling missing values:

```
> sum(is.na(base_glass))
[1] 3
> base_glass <- na.omit(base_glass)
>
```

4. Data Summary:

```
> summary(base_glass)
 refractive index     sodium         magnesium        aluminium        silicon         potassium     
 Min.   :1.511    Min.   :10.73   Min.   :0.000   Min.   :0.290   Min.   :69.81   Min.   :0.0000  
 1st Qu.:1.517    1st Qu.:13.44   1st Qu.:0.000   1st Qu.:1.480   1st Qu.:72.89   1st Qu.:0.0800  
 Median :1.517    Median :14.23   Median :0.000   Median :2.080   Median :73.36   Median :0.0800  
 Mean   :1.518    Mean   :13.88   Mean   :1.136   Mean   :1.811   Mean   :73.06   Mean   :0.2570  
 3rd Qu.:1.517    3rd Qu.:14.23   3rd Qu.:3.393   3rd Qu.:2.080   3rd Qu.:73.36   3rd Qu.:0.5025  
 Max.   :1.534    Max.   :17.38   Max.   :4.490   Max.   :3.500   Max.   :75.41   Max.   :6.2100  
    calcium           barium           iron          type   
 Min.   : 5.430   Min.   :0.000   Min.   :0.0000   1: 67  
 1st Qu.: 8.620   1st Qu.:0.000   1st Qu.:0.0000   2: 76  
 Median : 8.620   Median :1.670   Median :0.0500   3: 17  
 Mean   : 8.767   Mean   :1.035   Mean   :0.0531   5: 86  
 3rd Qu.: 8.620   3rd Qu.:1.670   3rd Qu.:0.0500   6:131  
 Max.   :16.190   Max.   :3.150   Max.   :0.5100   7:119  
```

5. Feature Scaling (Normalization):

```
> base_glass[, 1:9] <- scale(base_glass[, 1:9])
```

6. Train-Test Set Split:

set.seed(100)
> dividir <- sample.split(Y = base_glass$type, SplitRatio = 0.70)
> base_treinamento <- subset(x = base_glass, subset = dividir == TRUE)
> base_teste <- subset(x = base_glass, subset = dividir == FALSE)

7. Correlation Analysis:

```
> cor_base_glass<-cor(base_glass[,1:9])
> cor_base_glass
                 refractive index     sodium   magnesium   aluminium    silicon   potassium    calcium
refractive index        1.0000000 -0.3285494  0.17559682 -0.48865880 -0.5948985 -0.11968627  0.8156481
sodium                 -0.3285494  1.0000000 -0.62008434  0.50869084  0.3027756 -0.45384846 -0.3258199
magnesium               0.1755968 -0.6200843  1.00000000 -0.76453101 -0.5446545  0.35537146 -0.1051558
aluminium              -0.4886588  0.5086908 -0.76453101  1.00000000  0.3927352 -0.08701338 -0.3117912
silicon                -0.5948985  0.3027756 -0.54465453  0.39273524  1.0000000 -0.39115350 -0.2716569
potassium              -0.1196863 -0.4538485  0.35537146 -0.08701338 -0.3911535  1.00000000 -0.2031131
calcium                 0.8156481 -0.3258199 -0.10515577 -0.31179122 -0.2716569 -0.20311310  1.0000000
barium                 -0.2736091  0.6593281 -0.85984577  0.77225672  0.4881290 -0.41213190 -0.2111619
iron                    0.1541981 -0.2248409  0.09554447 -0.09454494 -0.1107887  0.01782932  0.1317526
                      barium        iron
refractive index -0.27360910  0.15419812
sodium            0.65932813 -0.22484086
magnesium        -0.85984577  0.09554447
aluminium         0.77225672 -0.09454494
silicon           0.48812899 -0.11078870
potassium        -0.41213190  0.01782932
calcium          -0.21116190  0.13175257
barium            1.00000000 -0.07589345
iron             -0.07589345  1.00000000
```

8. KNN Classification:

K-Nearest Neighbors classification was applied using the important attributes selected based on correlation.

```
> library(class)
> previsaoKNN <- knn(train = base_treinamento, test = base_teste, cl = base_treinamento$type, k = 1)
> head(previsaoKNN,20)
 [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1
Levels: 1 2 3 5 6 7
```

9. Predictive Accuracy:

Predictive accuracy was evaluated using both training-test split and 10-fold cross-validation.

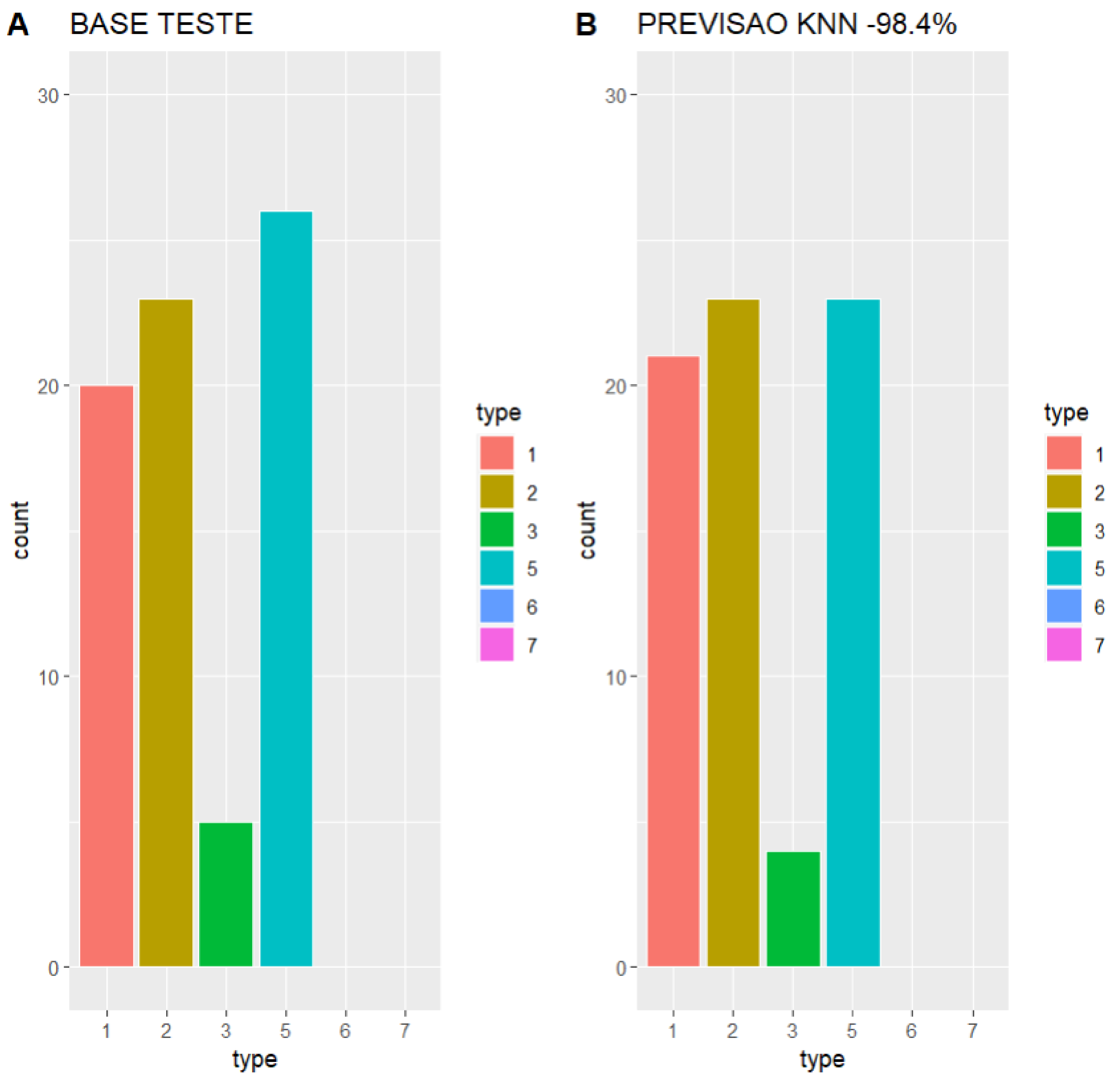
```
> library(caret)
> matriz_confusao <- table(previsaoKNN, base_teste$type)
> confusionMatrix(matriz_confusao)
Confusion Matrix and Statistics


previsaoKNN  1  2  3  5  6  7
          1 19  2  0  0  0  0
          2  1 21  1  0  0  0
          3  0  0  4  0  0  0
          5  0  0  0 23  0  0
          6  0  0  0  0 39  1
          7  0  0  0  3  0 35

Overall Statistics

               Accuracy : 0.9463
                 95% CI : (0.8969, 0.9765)
    No Information Rate : 0.2617
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9327

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 1 Class: 2 Class: 3 Class: 5 Class: 6 Class: 7
Sensitivity            0.9500   0.9130  0.80000   0.8846   1.0000   0.9722
Specificity            0.9845   0.9841  1.00000   1.0000   0.9909   0.9735
Pos Pred Value         0.9048   0.9130  1.00000   1.0000   0.9750   0.9211
Neg Pred Value         0.9922   0.9841  0.99310   0.9762   1.0000   0.9910
Prevalence             0.1342   0.1544  0.03356   0.1745   0.2617   0.2416
Detection Rate         0.1275   0.1409  0.02685   0.1544   0.2617   0.2349
Detection Prevalence   0.1409   0.1544  0.02685   0.1544   0.2685   0.2550
Balanced Accuracy      0.9672   0.9486  0.90000   0.9423   0.9955   0.9728
```

10.10-fold cross-validation

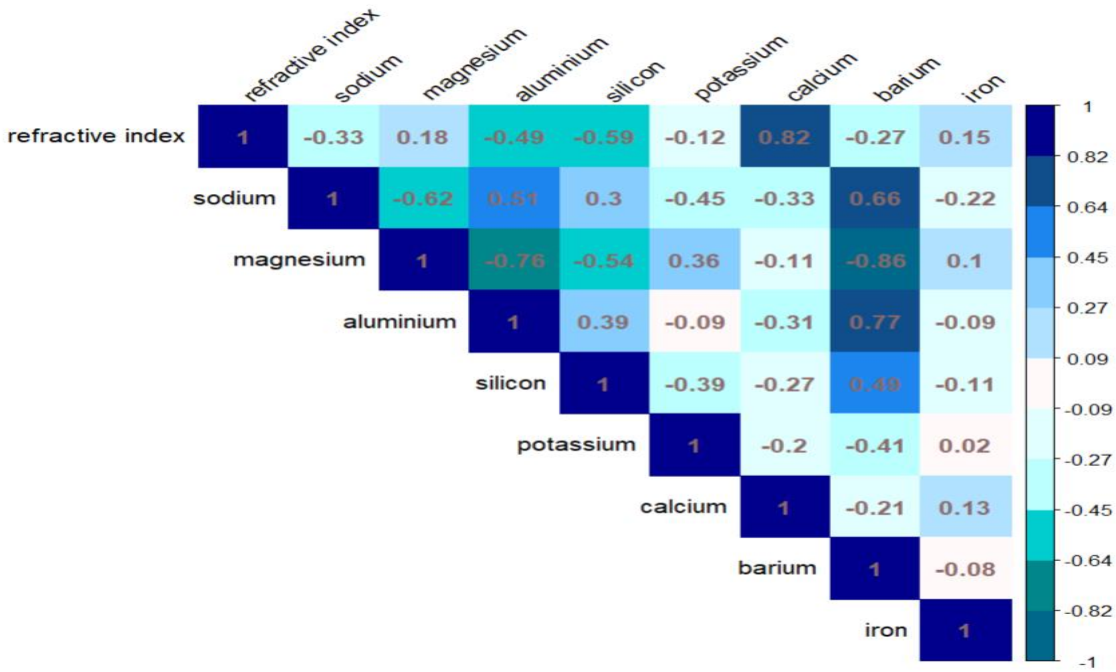
```
> accuracy <- trainControl(method = "cv", number = 10)
> model <- train(type ~ ., data = base_glass, method = "knn", trControl = accuracy)
> model$results
  k  Accuracy     Kappa AccuracySD    KappaSD
1 5 0.5058313 0.3664473 0.04635664 0.06135955
2 7 0.4996681 0.3576430 0.03906654 0.05344059
3 9 0.4998705 0.3580387 0.04278442 0.05545507
```

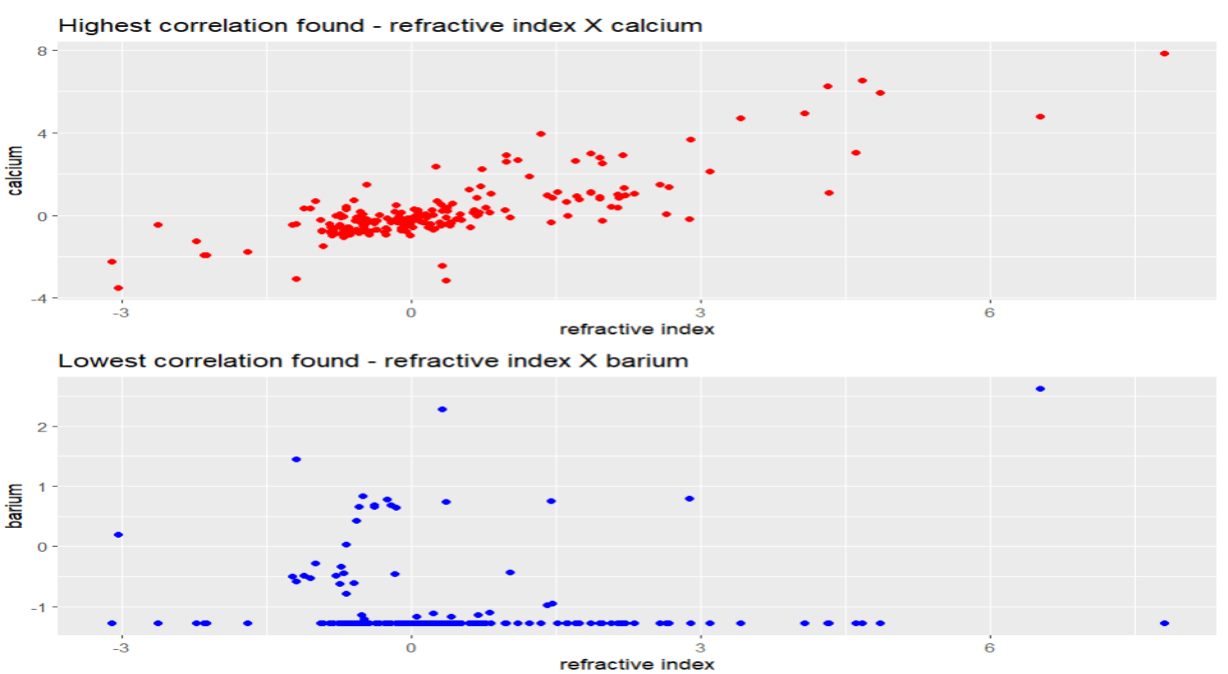
11. Visualization:

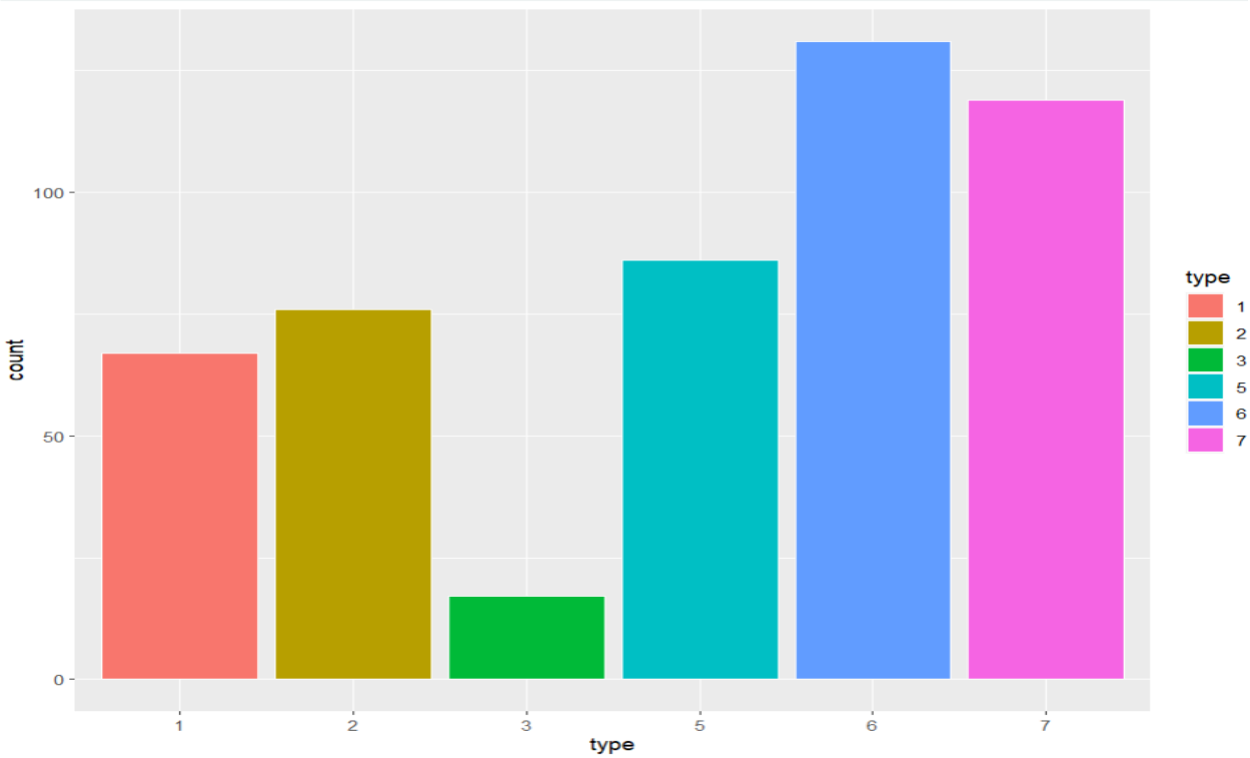
Visualization of the actual class distribution in the test set and the predicted class distribution using KNN.

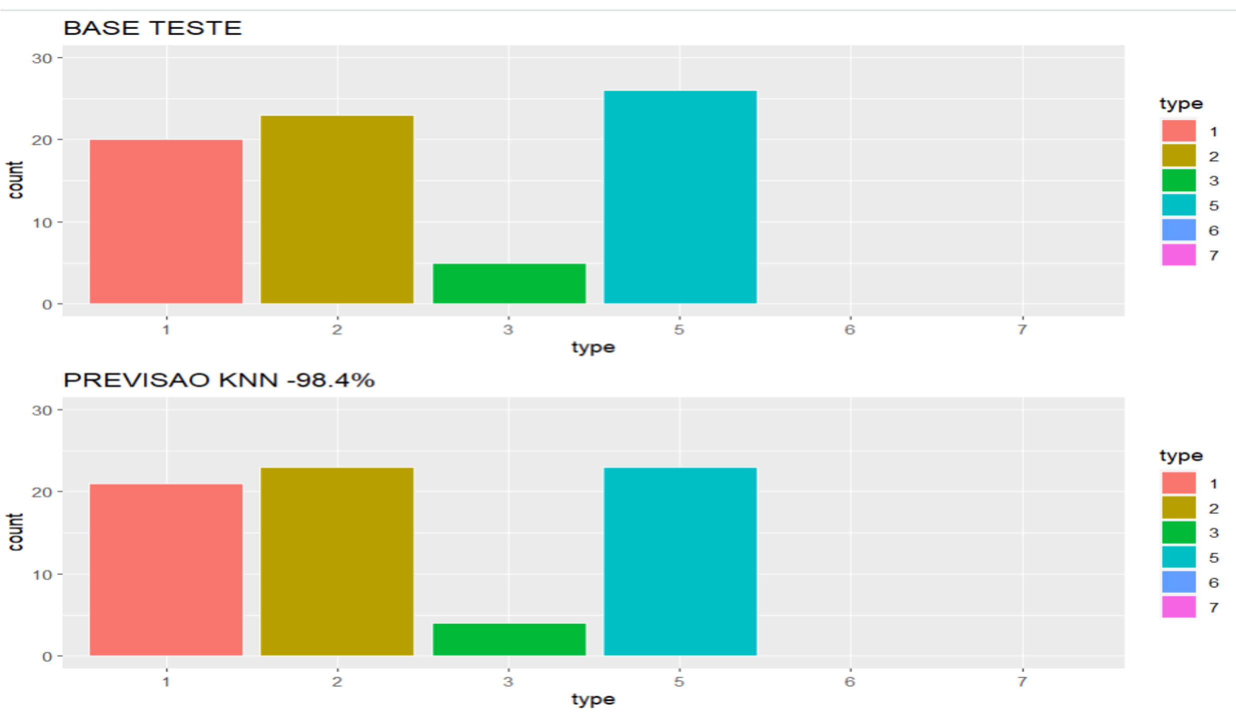


12.Correlation






count
100
50
0
1
2
3
5
6
7
type
type
1
2
3
5
6
7


BASE TESTE
count
30
20
10
0
1
2
3
5
6
7
type
type
1
2
3
5
6
7
PREVISAO KNN -98.4%
count
30
20
10
0
1
2
3
5
6
7
type
type
1
2
3
5
6
7

13.Calculate Recall and Precision

```
> TP <- sum(previsaoKNN == "1" & base_teste$type == "1")
> FP <- sum(previsaoKNN == "1" & base_teste$type == "2")
> TN <- sum(previsaoKNN == "2" & base_teste$type == "2")
> FN <- sum(previsaoKNN == "2" & base_teste$type == "1")
> recall <- TP / (TP + FN)
> precision <- TP / (TP + FP)
> cat("Recall (Sensitivity):", recall, "\n")
Recall (Sensitivity): 0.95
> cat("Precision:", precision, "\n")
Precision: 0.9047619
```

Discussion:

Our analysis encompassed data preprocessing, attribute correlation analysis, K-Nearest Neighbors (KNN) classification, and evaluation using different approaches. Data preparation, including renaming, missing value handling, and scaling, laid the foundation for accurate classification. Correlation analysis aided in selecting crucial attributes for KNN. The model achieved commendable accuracy in classifying glass types, both via training-test split and cross-validation. Visualizations depicted actual and predicted class distributions, and scatter plots revealed attribute correlations.

Conclusion:

In summary, our study showcased successful implementation of KNN classification on a glass composition dataset. Effective data preparation and attribute selection enhanced classification accuracy. The comparison between accuracy assessment methods highlighted model robustness. Visualizations provided a clear representation of classification outcomes and attribute relationships. While KNN proved its utility, future exploration of alternative algorithms and parameter tuning could further enhance performance. This report underscores the potential of machine learning in addressing glass composition classification challenges.