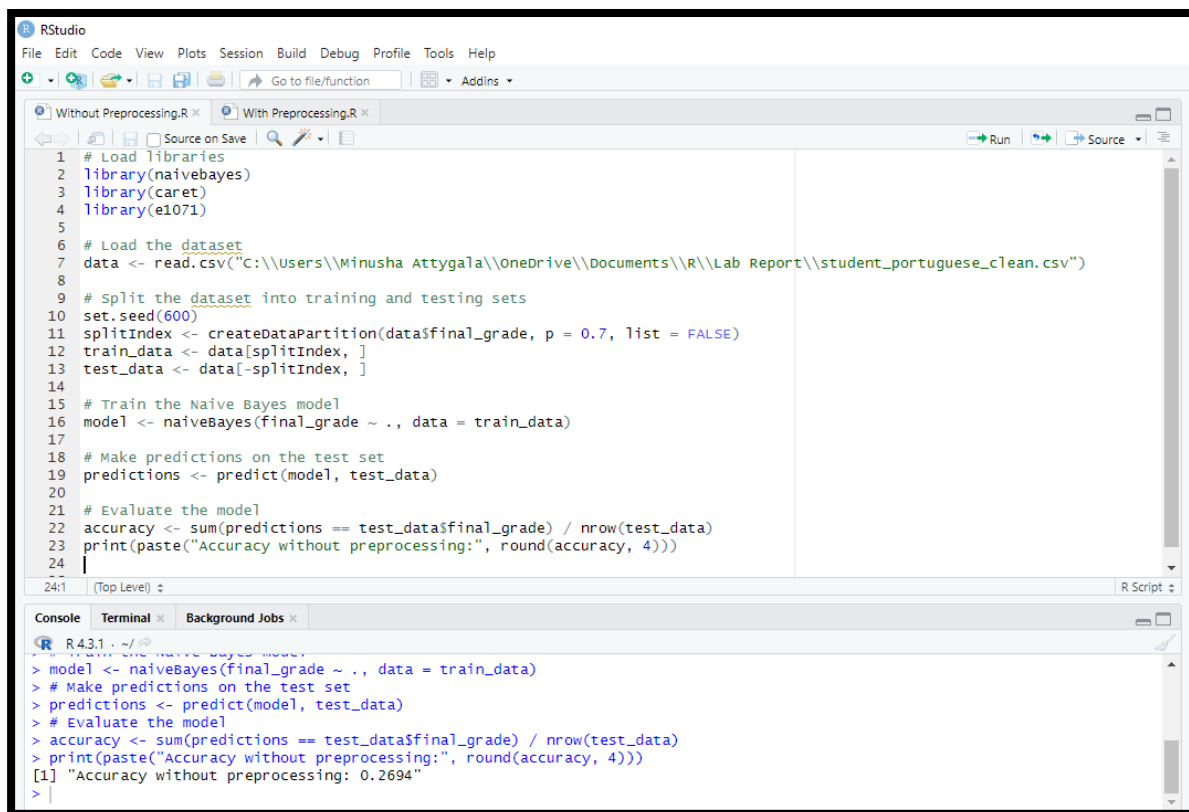


Minusha Attygala

27305

## Programming in R – Lab Report

- Without Preprocessing



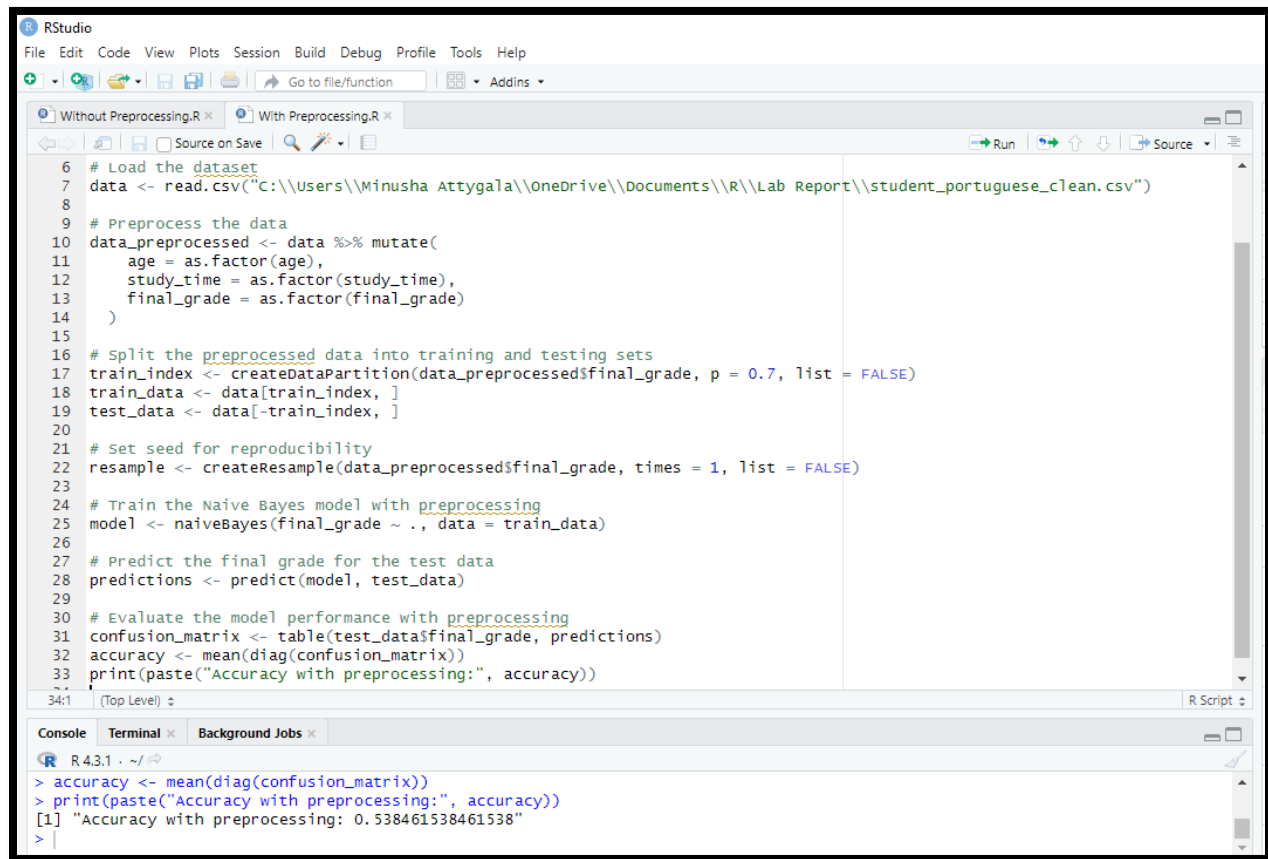
```
1 # Load libraries
2 library(naivebayes)
3 library(caret)
4 library(e1071)
5
6 # Load the dataset
7 data <- read.csv("C:\\Users\\Minusha Attygala\\OneDrive\\Documents\\R\\Lab Report\\student_portuguese_clean.csv")
8
9 # Split the dataset into training and testing sets
10 set.seed(600)
11 splitIndex <- createDataPartition(data$final_grade, p = 0.7, list = FALSE)
12 train_data <- data[splitIndex, ]
13 test_data <- data[-splitIndex, ]
14
15 # Train the Naive Bayes model
16 model <- naiveBayes(final_grade ~ ., data = train_data)
17
18 # Make predictions on the test set
19 predictions <- predict(model, test_data)
20
21 # Evaluate the model
22 accuracy <- sum(predictions == test_data$final_grade) / nrow(test_data)
23 print(paste("Accuracy without preprocessing:", round(accuracy, 4)))
24
```

```
R 4.3.1 ~/> # Train the Naive Bayes model
> model <- naiveBayes(final_grade ~ ., data = train_data)
> # Make predictions on the test set
> predictions <- predict(model, test_data)
> # Evaluate the model
> accuracy <- sum(predictions == test_data$final_grade) / nrow(test_data)
> print(paste("Accuracy without preprocessing:", round(accuracy, 4)))
[1] "Accuracy without preprocessing: 0.2694"
>
```

### **Accuracy without preprocessing: 0.2694**

This accuracy value represents the proportion of correctly classified instances by the Naive Bayes model when no preprocessing is applied to the data. Here, approximately 26.94% of the instances are correctly classified.

- With Preprocessing



The screenshot shows the RStudio interface with a script editor and a console. The script editor contains R code for loading a dataset, preprocessing it, splitting it into training and testing sets, training a Naive Bayes model, and evaluating its performance. The console shows the output of the evaluation, which is an accuracy of 0.538461538461538.

```
# Load the dataset
data <- read.csv("C:\\Users\\Minusha Attygala\\OneDrive\\Documents\\R\\Lab Report\\student_portuguese_clean.csv")

# Preprocess the data
data_preprocessed <- data %>% mutate(
  age = as.factor(age),
  study_time = as.factor(study_time),
  final_grade = as.factor(final_grade)
)

# Split the preprocessed data into training and testing sets
train_index <- createDataPartition(data_preprocessed$final_grade, p = 0.7, list = FALSE)
train_data <- data[train_index, ]
test_data <- data[-train_index, ]

# Set seed for reproducibility
resample <- createResample(data_preprocessed$final_grade, times = 1, list = FALSE)

# Train the Naive Bayes model with preprocessing
model <- naiveBayes(final_grade ~ ., data = train_data)

# Predict the final grade for the test data
predictions <- predict(model, test_data)

# Evaluate the model performance with preprocessing
confusion_matrix <- table(test_data$final_grade, predictions)
accuracy <- mean(diag(confusion_matrix))
print(paste("Accuracy with preprocessing:", accuracy))
```

```
> accuracy <- mean(diag(confusion_matrix))
> print(paste("Accuracy with preprocessing:", accuracy))
[1] "Accuracy with preprocessing: 0.538461538461538"
```

### Accuracy with preprocessing: 0.538461538461538

This accuracy value, ranging between 0 and 1, indicates the proportion of correctly classified instances by the Naive Bayes model when preprocessing is applied to the data. In this case, about 53.85% of the instances are correctly classified.

### *Comparing the two accuracies:*

The accuracy with preprocessing is higher than the accuracy without preprocessing. This suggests that the preprocessing steps applied to the data have a positive impact on the model's performance, as it leads to a higher proportion of correct predictions.