Apply Naive Bayes Algorithm from klaR Package

Nithya Sarabudla

02-03-2024

```
# load the klaR library
library(klaR)
```

```
## Loading required package: MASS
```

```
# Load the iris dataset
data(iris)
```

In this code chunk, we load the klaR library and loads the iris dataset.

```
# Display the number of rows in the iris dataset
nrow(iris)
```

```
## [1] 150
```

```
# Display summary statistics for the iris dataset
summary(iris)
```

```
##   Sepal.Length    Sepal.Width     Petal.Length    Petal.Width
##  Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100
##  1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300
##  Median :5.800   Median :3.000   Median :4.350   Median :1.300
##  Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
##  3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
##  Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
##        Species
##  setosa    :50
##  versicolor:50
##  virginica :50
##
##
##
```

```
# Display the first few rows of the iris dataset
head(iris)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
```

```
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

The iris dataset contains 150 observations of iris flowers, with measurements of four features: sepal length, sepal width, petal length, and petal width. These measurements are recorded in centimeters. Additionally, each observation is labeled with the species of iris it belongs to. The dataset comprises three species of iris: setosa, versicolor, and virginica, each with 50 observations.

```
# Create an index for selecting every fifth row for testing
testidx <- which(1:length(iris[, 1]) %% 5 == 0)
```

The code generates an index named testidx that selects every fifth row from the iris dataset. It does so by iterating through the indices of the dataset's rows and checking if the index is divisible by 5 without any remainder using the modulus operator (%%). If the condition is met, the corresponding index is included in the testidx vector.

```
# Separate the iris dataset into training and testing datasets based on the test index
iristrain <- iris[-testidx,] # Training dataset
iristest <- iris[testidx,] # Testing dataset
```

These lines of code split the iris dataset into training and testing datasets. The training dataset (iristrain) contains all rows except those selected for testing, while the testing dataset (iristest) contains only the rows selected for testing.

```
# apply Naive Bayes
nbmodel <- NaiveBayes(Species~., data=iristrain)
```

In this line, we train a Naive Bayes model (nbmodel) using the NaiveBayes() function from the klaR package. The formula Species ~ . specifies that we are predicting the Species variable based on all other variables in the dataset.

```
# check the accuracy
# Predict the classes of instances in the testing dataset using the trained model
prediction <- predict(nbmodel, iristest[,-5])
# Create a confusion matrix to check the accuracy of the predictions
table(prediction$class, iristest[,5])
```

```
##
##              setosa versicolor virginica
##   setosa         10          0         0
##   versicolor      0         10         2
##   virginica       0          0         8
```

This code predicts the classes of instances in the testing dataset (iristest) using the trained Naive Bayes model (nbmodel). The predicted classes are stored in the prediction object. we create a confusion matrix to assess the accuracy of the predictions. The table() function compares the predicted classes with the actual classes in the testing dataset, allowing us to evaluate how well the model performs.

The table presented is a confusion matrix, providing a detailed overview of a classification model's performance on a test dataset. Each row corresponds to the predicted class, while each column represents the

actual class. The values within the table indicate the number of instances where the model's predictions align with the true labels. For instance, in the first row, the model correctly identified 10 instances of the "setosa" species as "setosa". In the second row, it correctly classified 10 instances of "versicolor" as "versicolor" but incorrectly labeled 2 "versicolor" instances as "virginica". Similarly, in the third row, the model accurately predicted 8 instances of "virginica" as "virginica". The confusion matrix serves as a valuable tool for evaluating the model's performance, allowing for an assessment of its accuracy and the nature of any misclassifications.