Poorvi_Raut_HW05_DTree-Part2.R

Owner

2023-03-18

```
#knowledge Discovery and Data Mining (CS 513) Homework 5: Decision Tree USing CART Algorithm
#Course : CS 513-A
# First Name : Poorvi
#Last Name : Raut
# ID : 20009560
# Purpose : HW_05_DTree

#clearing object environment
rm(list = ls())
#get working directory
getwd()
```

```
## [1] "C:/Users/Owner/Desktop/Spring 2023/CS 513 KDD"
```

```
#Import package rpart for CART Decision Tree Algorithm , caret package to calculate confusion ma
trix metrics
library(class)
library(rpart)
library(caret)
```

```
## Loading required package: ggplot2
```

```
## Loading required package: lattice
```

```
#Load the “breast-cancer-wisconsin.data.csv” from canvas into R and perform the CART algorithm
dataSet<-read.csv("/Users/Owner/Desktop/Spring 2023/CS 513 KDD/breast-cancer-wisconsin.csv",na.s
tring = "?" )
#View Breast Cancer Dataset
View(dataSet)
#head(df, n=5)
#Summarizing each column
summary(dataSet)
```

```
##      Sample               F1               F2               F3
##  Min.   :   61634   Min.   : 1.000   Min.   : 1.000   Min.   : 1.000
##  1st Qu.:  870688   1st Qu.: 2.000   1st Qu.: 1.000   1st Qu.: 1.000
##  Median : 1171710   Median : 4.000   Median : 1.000   Median : 1.000
##  Mean   : 1071704   Mean   : 4.418   Mean   : 3.134   Mean   : 3.207
##  3rd Qu.: 1238298   3rd Qu.: 6.000   3rd Qu.: 5.000   3rd Qu.: 5.000
##  Max.   :13454352   Max.   :10.000   Max.   :10.000   Max.   :10.000
##
##        F4               F5               F6               F7
##  Min.   : 1.000   Min.   : 1.000   Min.   : 1.000   Min.   : 1.000
##  1st Qu.: 1.000   1st Qu.: 2.000   1st Qu.: 1.000   1st Qu.: 2.000
##  Median : 1.000   Median : 2.000   Median : 1.000   Median : 3.000
##  Mean   : 2.807   Mean   : 3.216   Mean   : 3.545   Mean   : 3.438
##  3rd Qu.: 4.000   3rd Qu.: 4.000   3rd Qu.: 6.000   3rd Qu.: 5.000
##  Max.   :10.000   Max.   :10.000   Max.   :10.000   Max.   :10.000
##                                    NA's   :16
##        F8               F9            Class
##  Min.   : 1.000   Min.   : 1.000   Min.   :2.00
##  1st Qu.: 1.000   1st Qu.: 1.000   1st Qu.:2.00
##  Median : 1.000   Median : 1.000   Median :2.00
##  Mean   : 2.867   Mean   : 1.589   Mean   :2.69
##  3rd Qu.: 4.000   3rd Qu.: 1.000   3rd Qu.:4.00
##  Max.   :10.000   Max.   :10.000   Max.   :4.00
##
```

```
#Converting the type of column F6 from character to numeric
n<-as.numeric(as.character(dataSet$F6))
summary(n,na.rm=TRUE)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   1.000   1.000   1.000   3.545   6.000  10.000      16
```

```
#Checking the number of rows
nrow(dataSet)
```

```
## [1] 699
```

```
#Remove the rows with missing values
dataSet<-na.omit(dataSet)
nrow(dataSet)
```

```
## [1] 683
```

```
#Converting the class column to factor class
dataSet$Class<-factor(dataSet$Class,levels = c("2","4"),labels = c("benign","malignant"))
is.factor(dataSet$Class)
```

```
## [1] TRUE
```

```
dataSet1<-dataSet[2:11]
View(dataSet1)

#partitioning 70% of size
sample_size<-floor(0.70*nrow(dataSet1))
#Set the seed to make your partition reproducible
set.seed(123)
traindata<-sample(seq_len(nrow(dataSet1)),size = sample_size)
# 70% of data in training set
train<-dataSet1[traindata,]

# 30% of data in testing set
test<-dataSet1[traindata,]
#Implementing CART algorithm
cart_algo<-rpart(Class ~.,data=train,method = "class")

#Predicting target class
predict_alg<-predict(cart_algo,test,type = "class")
print(length(predict_alg))
```

```
## [1] 478
```

```
#print(length(test$Class))

#creating confusion matrix
conf_matrix<-table(predict_alg,test$Class)
print(conf_matrix)
```

```
##
## predict_alg benign malignant
##   benign       298        12
##   malignant      7       161
```

```
confusionMatrix(predict_alg,test$Class)
```

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction  benign malignant
##   benign       298        12
##   malignant      7       161
##
##                Accuracy : 0.9603
##                  95% CI : (0.9386, 0.9759)
##     No Information Rate : 0.6381
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9134
##
##  Mcnemar's Test P-Value : 0.3588
##
##             Sensitivity : 0.9770
##             Specificity : 0.9306
##          Pos Pred Value : 0.9613
##          Neg Pred Value : 0.9583
##              Prevalence : 0.6381
##          Detection Rate : 0.6234
##    Detection Prevalence : 0.6485
##       Balanced Accuracy : 0.9538
##
##        'Positive' Class : benign
##
```

```
#Calculating Accuracy of the algorithm
accuracy<-function(x){sum(diag(x)/sum(rowSums(x)))*100}
accuracy(conf_matrix)
```

```
## [1] 96.0251
```

```
#Error rate
e<- 100- accuracy(conf_matrix)
print(e)
```

```
## [1] 3.974895
```