Introduction to Data Analytics Project

Classification using Decision tree Algorithm

Course instructor:
Dr.Sreeja SR

Group ID: 12

Team Members

- Basava Chari Boppudi
- Kalluri Balarajaiah
- Sanjay Gnana Kukutla
- Palani Mani Srisatya Karthik
- K. Venkat Subhash

Problem Statement

Topic 12: Classification using Decision tree algorithm

Reference: Breast cancer data

a) Do proper data pre-processing

b) Build a classifier model based on ID3/C4.5 algorithm. You should divide the data set randomly in 2:1 ratio using any random sampling method and then learn the model using the training data set.

c) Verify the classifier's performance on the test set. Report the performance measure in terms of Confusion matrix, Predictive accuracy, F1-score, Precision and Recall.

d) Use *k*-fold cross validation with different values of *k*. Obtain an ROC curve with different values of *k*.

Understanding the problem.

About data set.

- Given data set is "BreastCancer"
- The attributes in the data set are like CL.thickenss, Cell.size, Cell.shape, Marg.adhesion, Epith.c.size, Bare.nuclei, Bl.cromatin, Normal.nucleoli, Mitoses, Class.
- All attributes are of numeric data types.
- Class attribute is binary having values 0, 1 which means the patient having cancer or not .
- We uses read.csv("file_path") to get the data from csv file.

Concepts:-

ID3 Algorithm (Iterative Dichotomizer 3) :-

- In ID3 we use entropy for measuring how informative the node is for splitting further.
- It is mandatory that if we are splitting any attribute the property that average entropy of the resulting training subsets will be less than or equal to that of the previous training set.
- We use Information gain to determine the goodness of a split.
- We choose largest value of Information gain for further splitting attribute.
- Information gain never be negative.
- It partitions into a number of smaller training sets based on the distinct values of attribute under split.

Continued..

K-fold cross validation:-

- 1)Randomly shuffle the dataset.
- 2)Create k groups from the dataset.
- 3)For every distinct group:

    a)Select one group should be used as a test data set.

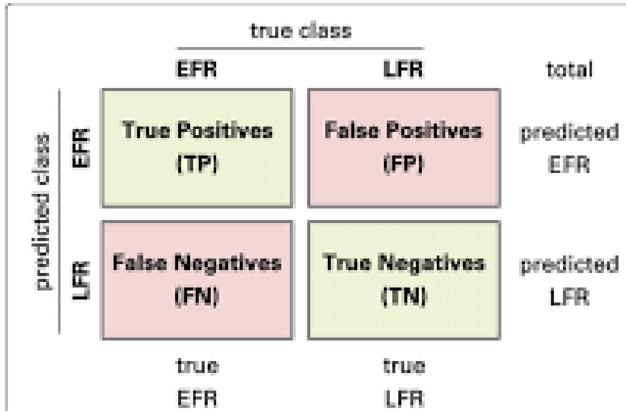
    b)As a training data set, use the remaining groupings.

    c)Fit a model to the training data, then check it against the test data.

    d)Retain the evaluation score and discard the model.

- Continue this for k folds to summarize a model.

Performance Metrics



$$PR = \frac{TP}{TP+FP}$$

$$RE = \frac{TP}{TP+FN}$$

$$CA = \frac{TP+TN}{TP+TN+FP+FN}$$

$$F_1 = \frac{2TP}{2TP+FP+FN}$$

PR- precision
RE- recall
CA- accuracy
F1- f measure

Implementation of project

- Step 1 : Data Preprocessing
- Step 2 : Split the data set into train and test data
- Step 3 : Using ID3 algorithm train the classifier model with this train data.
- Step 4 : verify the performance of model with the test data. And calculate the measures like accuracy, F1 score, Precision, Recall.
- Step 5 : Using k-fold cross validation draw the ROC curve.

Step 1

Installing necessary packages and loading libraries. After, removing the nan values from the data set

```
#preprocess data
data <- na.omit(data)
str(data)
data$Class<- as.factor(data$Class)
str(data)
```

Finding the relation between the attributes and removing the most related attributes i.e: correlation coefficent >= 0.95

```
corr_mat = cor(data[1:9],method = 'pearson')
View(corr_mat)
#plot(corr_mat)

corr_mat[!upper.tri(corr_mat)] <- 0

ggcorrplot(corr_mat)

data <- data[, !apply(corr_mat,2,function(x) any(abs(x) > 0.95 ,na.rm = TRUE)]

View(data)
```

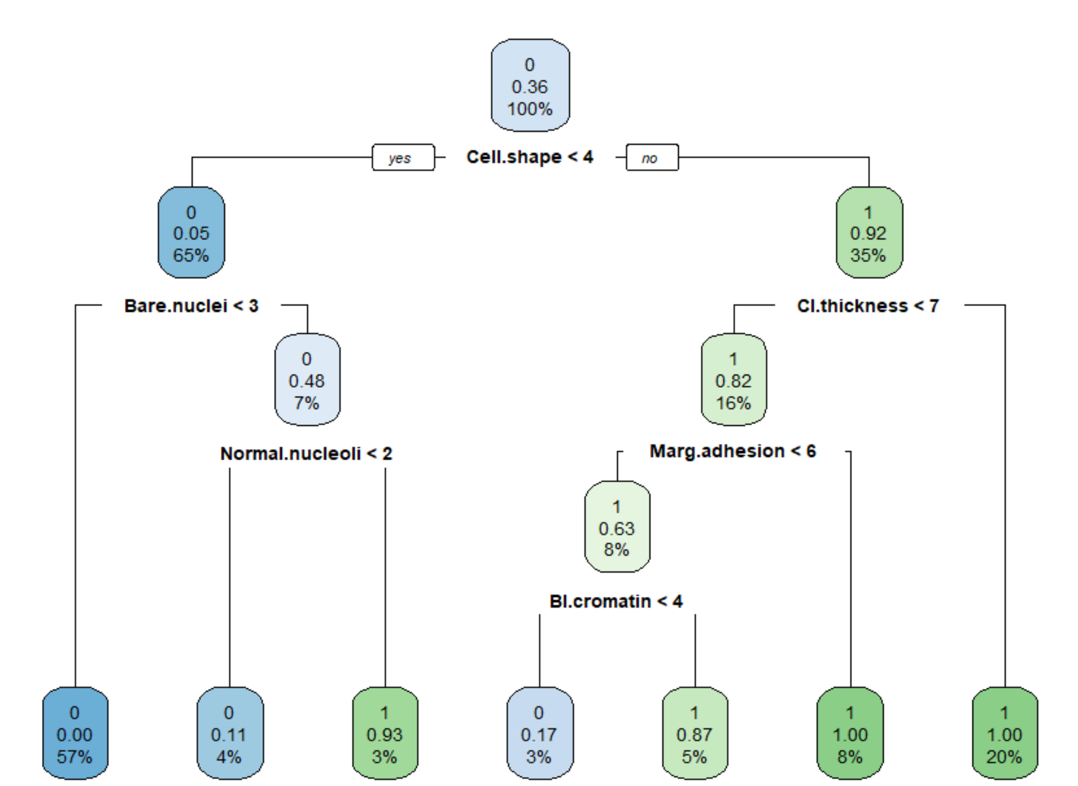
Step 2

```
37  sample_split <- floor(.67*nrow(data))
38  sample_split
39
40  set.seed(1)
41  k = seq_len(nrow(data))
42
43  training <- sample(k,size=sample_split)
44  training
45
46  cancer_train <- data[training,]
47  cancer_train
48
49  cancer_test <- data[-training,]
50  cancer_test
51
```

- Splitting the data set into training and test with the ratio of 2:1
- In line 37, "0.67" define 2/3 of total dataset as trainset.
- Cancer_train <---- Training data
- Cancer_train <---- Test data

Step 3 : Creating the model

```
59 #building model using id3 algorithm
60 tree_model <- rpart(Class~.,data=cancer_train,method="class",parms=(list(split='information')))
61 tree_model
62 #analyzing results and plotting tree
63 printcp(tree_model)
64 plotcp(tree_model)
65 summary(tree_model)
66 rpart.plot(tree_model)
67
```



- In line 60, split = 'information' state that the split is done based on information gain of the attributes.
- "rpart" is the function from "rpart" library which creates the model for decision tree.
- In line 66, we draw the entire decision tree.

Step 4 : Predicting

```r
#checking accuracy
predict.cls <- tree_model %>%
  predict(cancer_test,type="class")

#prediction accuracy
mean(predict.cls==cancer_test$Class)
head(predict.cls)
```

```r
#calculating f1 score
F1_Score(y_pred=predict.cls,y_true=cancer_test$Class,positive="1")
#confusion matrix
res<-confusionMatrix(cancer_test$Class,predict.cls,positive = "1")

#precision
precision <- res$byClass['Pos Pred Value']
print(precision)
#recall
recall <- res$byClass['Sensitivity']
print(recall)
```

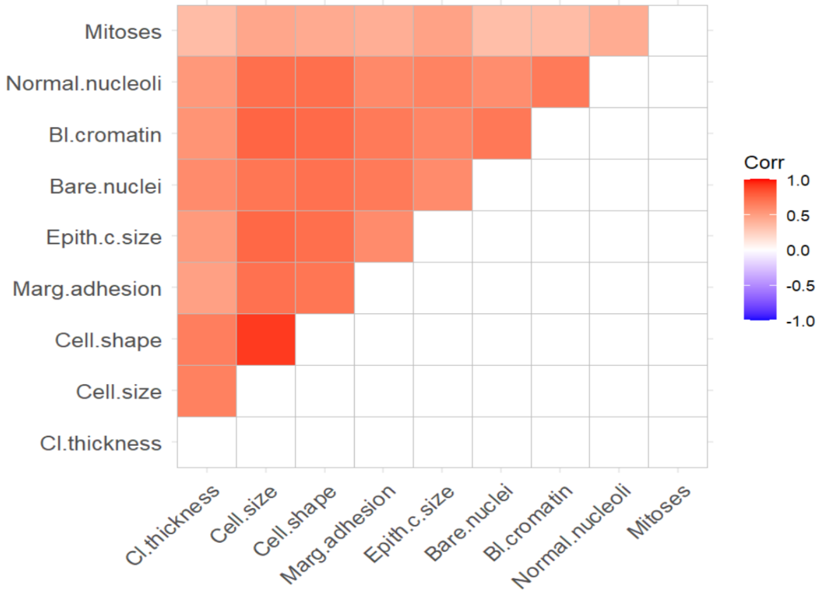
- Calculating the performance measure metrics
- Accuracy, F1 score, precision, Recall

Step 5 : k fold Validation

```r
folds <- createFolds(data$Class,k=10)

crossvalidation = lapply(folds,function(x){
   training_fold = data[-x,]
   test_fold = data[x,]
   tree_model_kfold <- rpart(Class~.,data=training_fold,method='class',parms = list(split='information'))
   test_fold_data <- select(test_fold,-10)
   test_fold_out <- test_fold["Class"]

   y_pred <- tree_model_kfold %>%
     predict(test_fold_data,type="class")
   confusionmatrix = table(test_fold[,10],y_pred)
   acc = (confusionmatrix[1,1]+confusionmatrix[2,2])/(confusionmatrix[1,2]+confusionmatrix[2,1]+confusionmatrix[1,1]+confusionmatrix[2,2
   accuracy <-c(accuracy,acc)
   y = confusionmatrix[1,1]/(confusionmatrix[1,1]+confusionmatrix[1,2])
   TPR <- c(TPR,y)
   x = confusionmatrix[2,1]/(confusionmatrix[2,2]+confusionmatrix[2,1])
   FPR <- c(FPR,x)
   result <- cbind(accuracy,TPR,FPR)
   return (result)
   })
b = crossvalidation
b
#average accuracy
```

- Code for k fold Cross Validation with k as 10.

Experimental Results :-



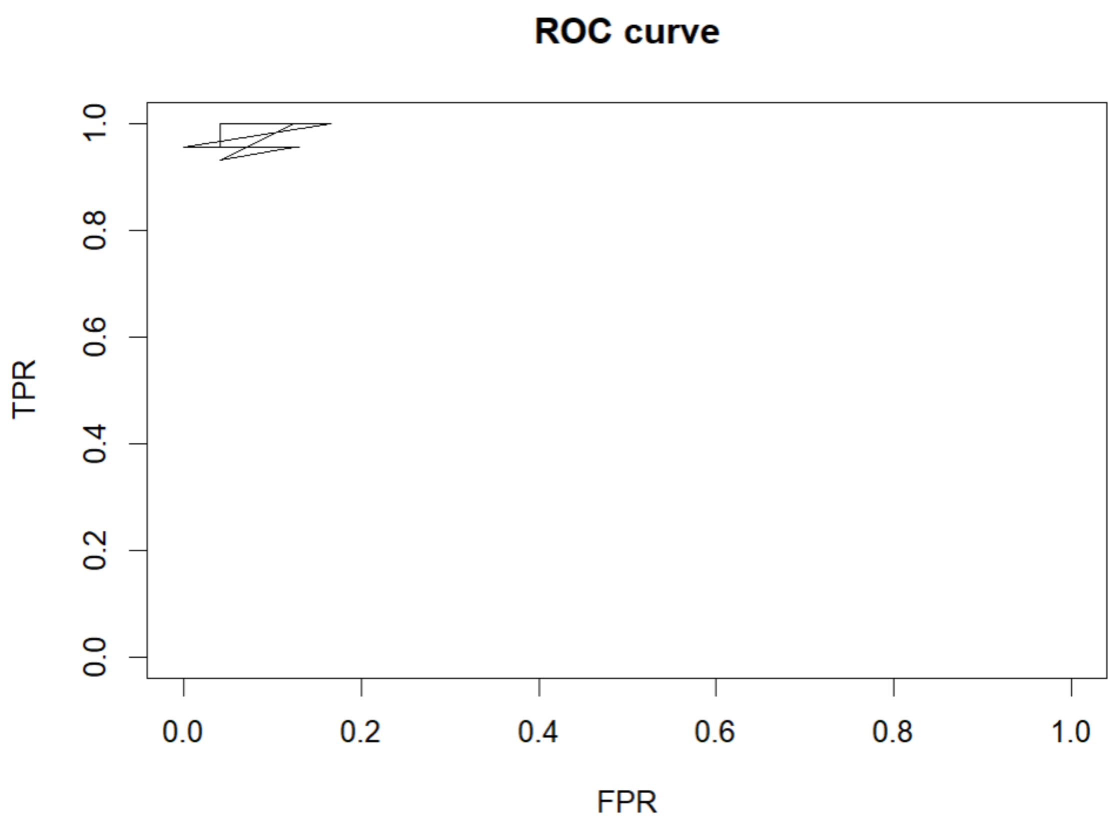
- Correlation plot using Pearson's method.

- Here, we get confusion matrix and performance values such as sensitivity(TPR), specificity(TNR) and accuracy etc...

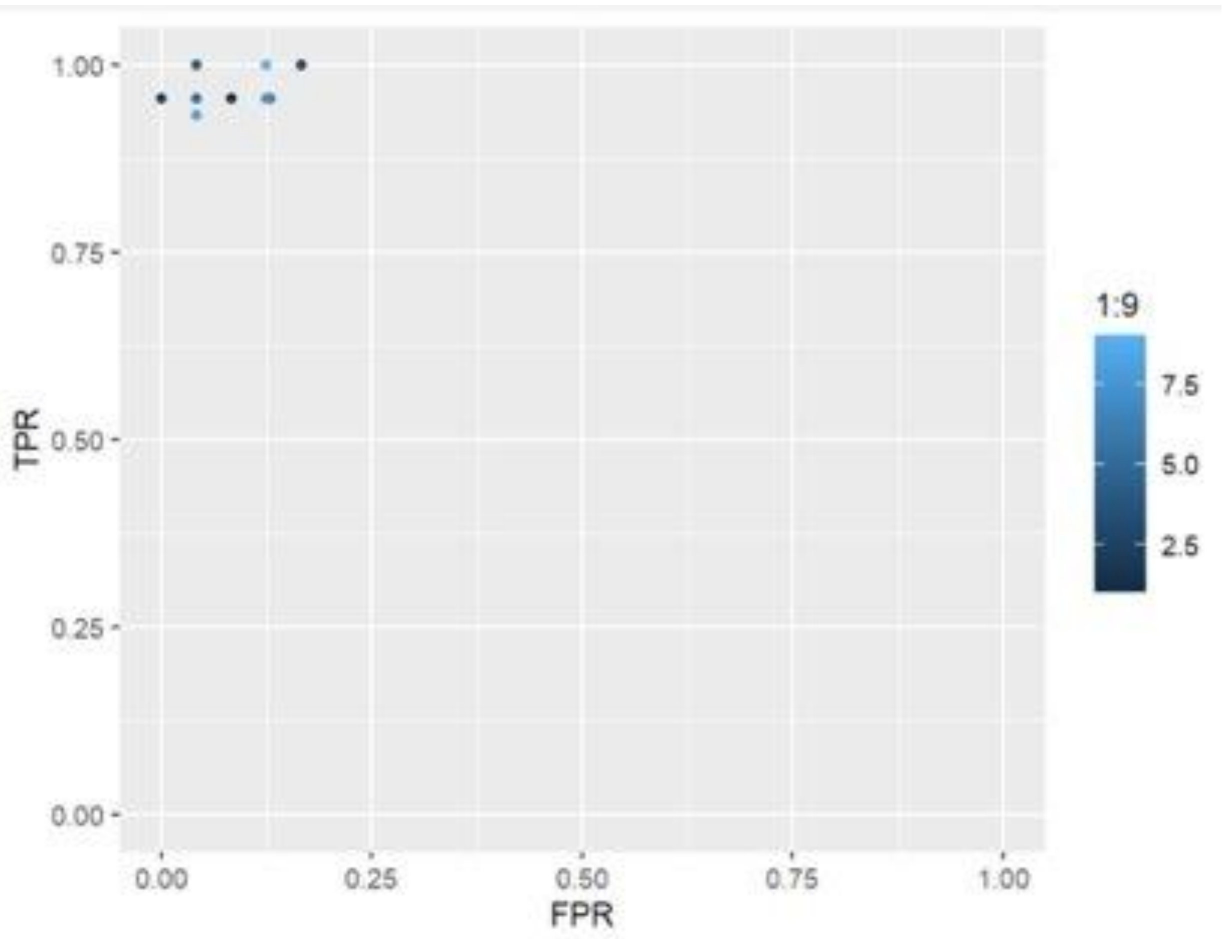
```
> res
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 148   4
         1  11  63

               Accuracy : 0.9336
                 95% CI : (0.8929, 0.9624)
    No Information Rate : 0.7035
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.8456

 Mcnemar's Test P-Value : 0.1213

            Sensitivity : 0.9403
            Specificity : 0.9308
         Pos Pred Value : 0.8514
         Neg Pred Value : 0.9737
             Prevalence : 0.2965
         Detection Rate : 0.2788
   Detection Prevalence : 0.3274
      Balanced Accuracy : 0.9356

       'Positive' Class : 1
```

Continued..

- In decision tree the information gain among all attributes is higher for "Cell.shape", hence it is a root node.
- From given data, we have 9 attributes among them only 6 are used for building decision tree using ID3 algorithm.
- We Predict the class based on the leaf node of the tree.



- Plot of ROC curve for k = 10.
- The plot is based on TPR and FPR values of k Folds.
- "ROC curve shows trade off between TPR and FPR"



- TPR vs FPR plot for k fold validation

Experimental results

- For the given dataset the decision tree is made by main attribute which having high Information Gain is "Cell-Shape".
- Before k fold validation the accuracy was 0.9336.
- We get better accuracy for k values 10.
- After 10 fold validation the accuracy rises to 0.9501673.

Thank you