Decision Tree

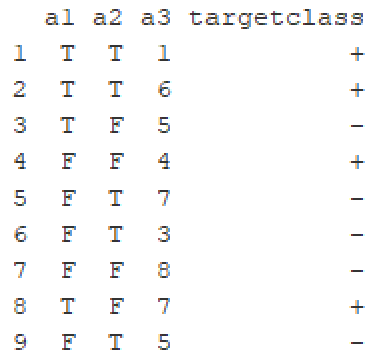
Vazgen Tadevosyan

April 28, 2019

You are required to submit both Markdown and HTML files. Data set (Spam E-mail Data) relevant for this assignment can be found in the R package "DAAG".

Problem 1 (7 pt.)

Consider the training examples shown in the table for a binary classification problem:



```
  a1 a2 a3 targetclass
1  T  T  1           +
2  T  T  6           +
3  T  F  5           -
4  F  F  4           +
5  F  T  7           -
6  F  T  3           -
7  F  F  8           -
8  T  F  7           +
9  F  T  5           -
```

a. What is the best split (between a1 and a2) according to the classification error rate? Show calculations in R.
b. What is the best split (between a1 and a2) according to the Gini index? Show calculations in R. Which attribute would the decision tree algorithm choose? Why?
c. Why does not entropy start with 0?
d. Why DT is called a greedy algorithm?

---

a.

```
class_err_a1<-1-max(7/9,2/9)
class_err_a2<-1-max(5/9,4/9)
class_err_a1<class_err_a2
```

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

The best split is a1 as classification rate error is less than a2.

b.

```
a1<-1-((7/9)^2+(2/9)^2)
a2<-1-((5/9)^2+(4/9)^2)
a1<a2
```

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

the best split is a1 again because gini index was lower than a2.

c. $entropy(A)=-_{k = 1}^{m} p_k*log_2(p_k) $ So when we $P_k = 0$ we will have $log_2(p_k) = -inf$ So it does not start from 0.
d. it is making the locally optimal choice at each stage the algorithm doesn't consider the larger problem as a whole. Once a decision has been made, it is never reconsidered.

Problem 2 (6 pt.)

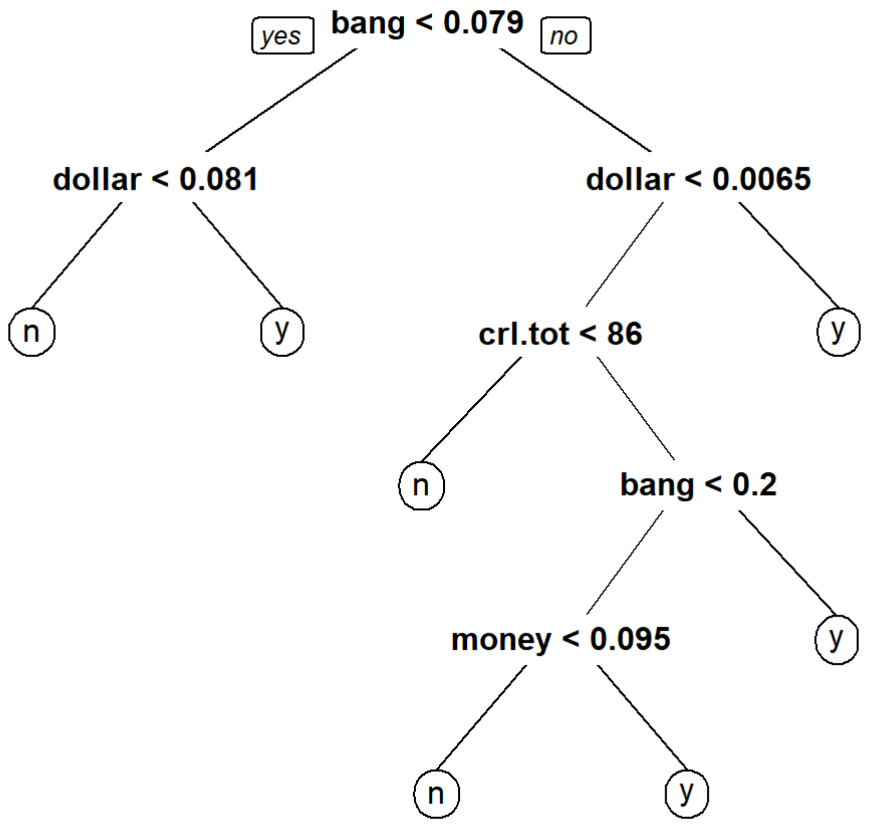
a. Suppose the sysadmin wants to understand and predict the process of recognizing the emails as spam for new e-mails which make up 10% of your initial data. Use the full decision tree algorithm to solve this task. Show your tree and interpret the results.
b. How many observations have your smallest node? Set the minimum number of observations in any terminal node 25% of the number of your initial data. Show the tree. Why do we need the arguments minbucket and minsplit?
c. Which are the advantages and disadvantages (at least 2 of each) of the DT algorithm?

d.

```
data("spam7")
set.seed(1)
colnames(train)
```

```
## NULL
```

```
train_index<-createDataPartition(spam7$yesno,p = 0.9,list = F)
train<-spam7[train_index,]
test<-spam7[-train_index,]
full<-rpart(yesno~.,data = train)
prp(full)
```



DT algorithm used DOllar, crt.tot,money, bang attributes to split. There is 7 leaf nodes and 5 child nodes.

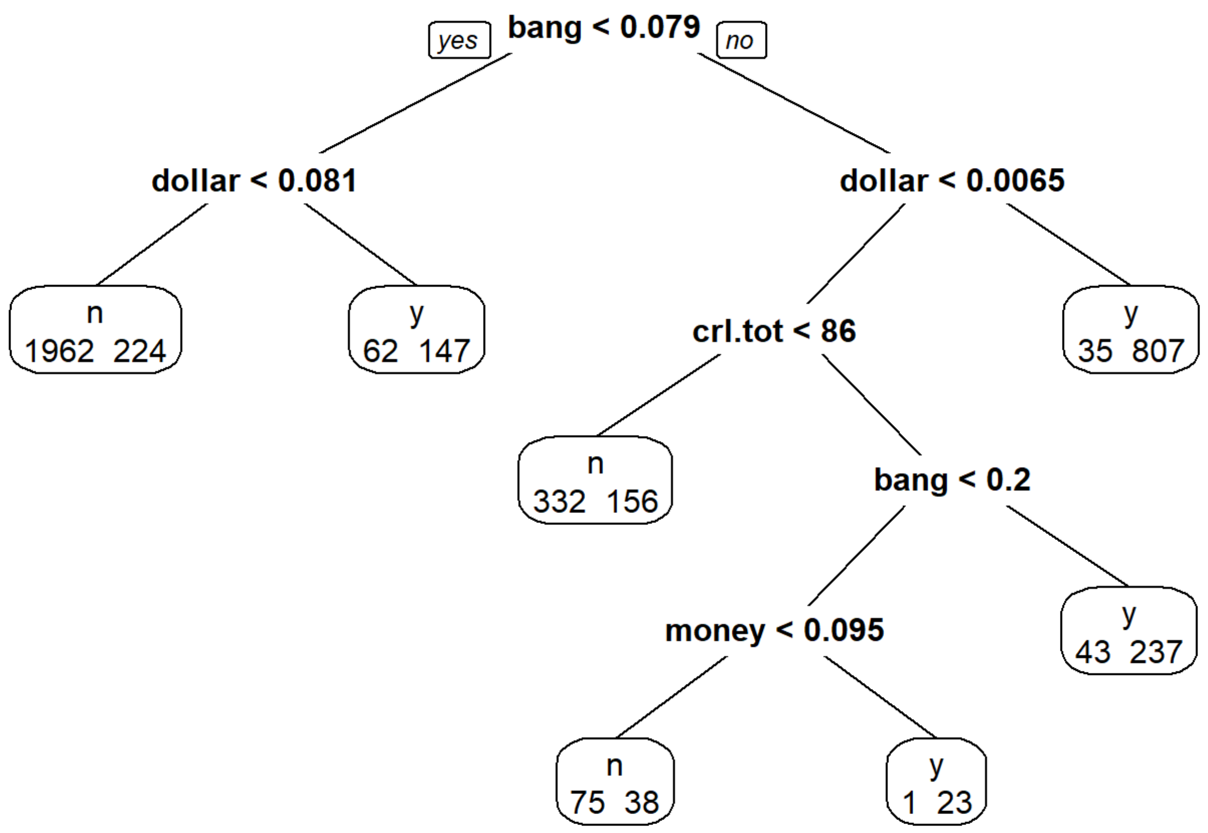
```
asRules(full)
```

```
##
##  Rule number: 7 [yesno=y cover=842 (20%) prob=0.96]
##    bang>=0.0785
##    dollar>=0.0065
##
##  Rule number: 53 [yesno=y cover=24 (1%) prob=0.96]
##    bang>=0.0785
##    dollar< 0.0065
##    crl.tot>=85.5
##    bang< 0.1955
##    money>=0.095
##
##  Rule number: 27 [yesno=y cover=280 (7%) prob=0.85]
##    bang>=0.0785
##    dollar< 0.0065
##    crl.tot>=85.5
##    bang>=0.1955
##
##  Rule number: 5 [yesno=y cover=209 (5%) prob=0.70]
##    bang< 0.0785
##    dollar>=0.081
##
##  Rule number: 52 [yesno=n cover=113 (3%) prob=0.34]
##    bang>=0.0785
##    dollar< 0.0065
##    crl.tot>=85.5
##    bang< 0.1955
##    money< 0.095
##
##  Rule number: 12 [yesno=n cover=488 (12%) prob=0.32]
##    bang>=0.0785
##    dollar< 0.0065
##    crl.tot< 85.5
##
##  Rule number: 4 [yesno=n cover=2186 (53%) prob=0.10]
##    bang< 0.0785
##    dollar< 0.081
```

So when dollar number of occurrences of the bang >=0.0785 and dollar>=0.0065 it classified as spam with probability 0.96, and so on.

b.

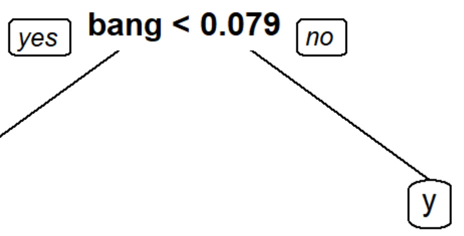
```
prp(full,extra = 1)
```



In the smallest node we have 24

observations.

```
model<-rpart(yesno~.,data = train,minbucket=nrow(train)*0.25)
prp(model)
```



We need to use minbucket and

minsplit to handle problem with overfitting. Disadvantages There is a high probability of overfitting in Decision Tree. Calculations can become complex when there are many class labels. It uses Greedy algorithm

Advantages. Decision Trees are easy to explain. It results in a set of rules. It follows the same approach as humans generally follow while making decisions. We can represent information graphically. —————————————-

Problem 3 (7 pt.)

a. Make predictions based on both models.
b. Compare the models using the function confusionMatrix() and their results, ROC curve, and AUC. Which does perform better? Why?
c. What is the difference between regression and classification trees (in terms of the loss function, prediction, and type of dependent variable)?

```r
full_pred<-predict(full,test,type = "prob")
model_pred<-predict(model,test,type = "prob")
pred_f<-factor(ifelse(full_pred[,2]>0.5,"y","n"))
pred_m<-factor(ifelse(model_pred[,2]>0.5,"y","n"))
confusionMatrix(pred_f,test$yesno,positive = "y")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   n   y
##          n 260  52
##          y  18 129
##
##                Accuracy : 0.8475
##                  95% CI : (0.8113, 0.8791)
##     No Information Rate : 0.6057
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6699
##  Mcnemar's Test P-Value : 8.005e-05
##
##             Sensitivity : 0.7127
##             Specificity : 0.9353
##          Pos Pred Value : 0.8776
##          Neg Pred Value : 0.8333
##              Prevalence : 0.3943
##          Detection Rate : 0.2810
##    Detection Prevalence : 0.3203
##       Balanced Accuracy : 0.8240
##
##        'Positive' Class : y
##
```

```r
confusionMatrix(pred_m,test$yesno,positive = "y")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   n   y
##          n 218  44
##          y  60 137
##
##                Accuracy : 0.7734
##                  95% CI : (0.7323, 0.8109)
##     No Information Rate : 0.6057
##     P-Value [Acc > NIR] : 1.718e-14
##
##                   Kappa : 0.5329
##  Mcnemar's Test P-Value : 0.1413
##
##             Sensitivity : 0.7569
##             Specificity : 0.7842
##          Pos Pred Value : 0.6954
##          Neg Pred Value : 0.8321
##              Prevalence : 0.3943
##          Detection Rate : 0.2985
##    Detection Prevalence : 0.4292
##       Balanced Accuracy : 0.7705
##
##        'Positive' Class : y
##
```

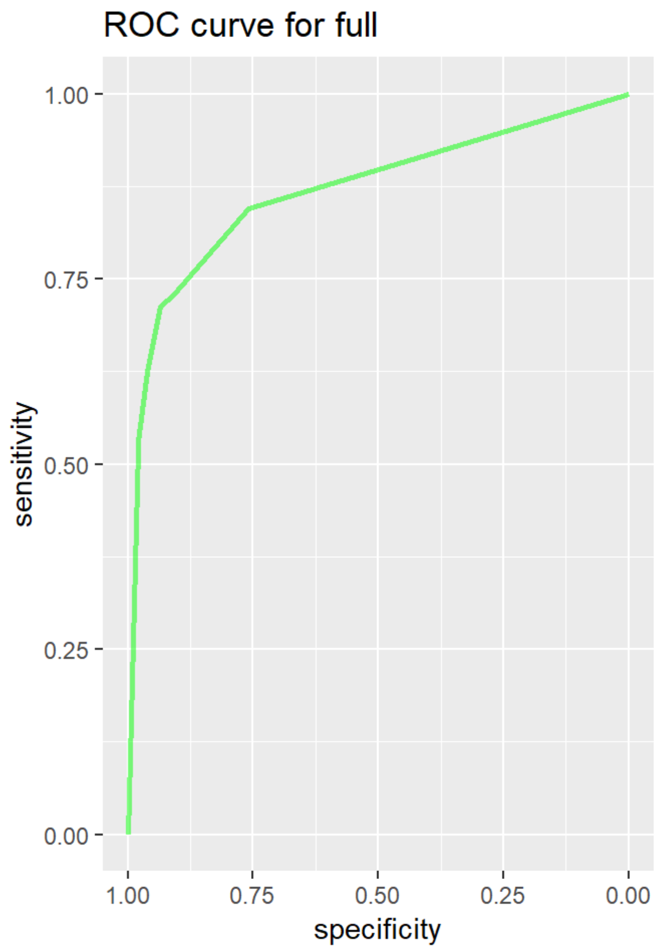
```
rrr<-roc(test$yesno,full_pred[,2])
g1<-ggroc(rrr,alpha = 0.5, colour = "green", linetype = 1, size = 1)+ggtitle("ROC curve for full")
rrr1<-roc(test$yesno,model_pred[,2])
g2<-ggroc(rrr1,alpha = 0.5, colour = "green", linetype = 1, size = 1)+ggtitle("ROC curve for Model")
auc(rrr)
```

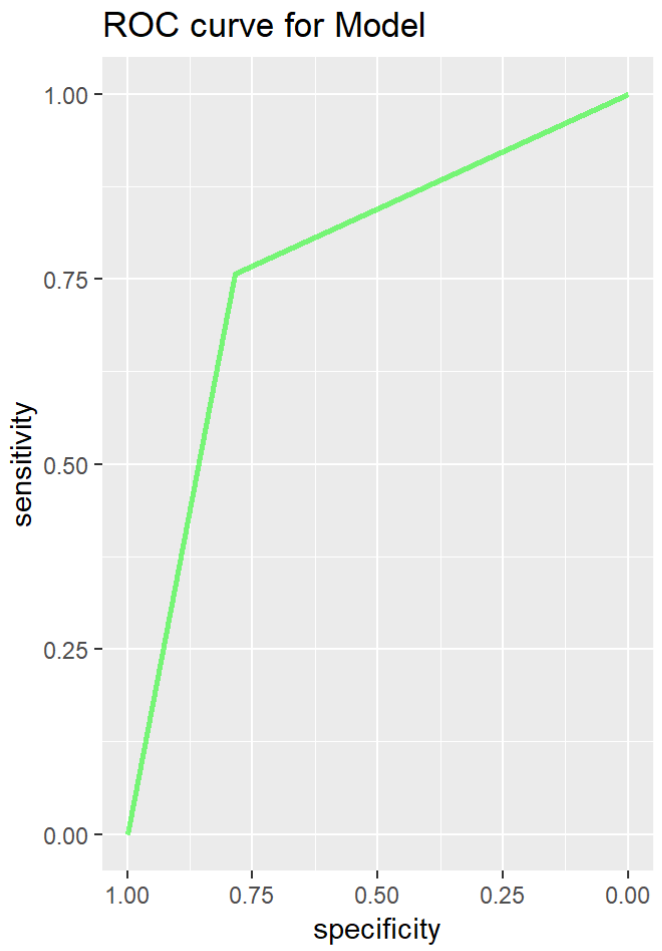
```
## Area under the curve: 0.87
```

```r
auc(rrr1)
```

```
## Area under the curve: 0.7705
```

```r
grid.arrange(g1, g2,nrow=1)
```





Overall Full model predict better as almost all metrics are better than metrics provided model.Only sensitivity is a little lower than the model’s score. AUC for full model(0.87) is larger than model’s AUC (0.77) c.dependent cariable is numeric for refression and categorical for decishion tree. Loss function for LR is minimizing squared residuals, for decision tree it is Misclassification cost,regression predicts a quantitative variable decision tree a qualitative variable.