# lab2_block1

Shwetha

11/18/2020

## Assignment 2

### 1.

Partitioning the data into train , test and validation.

```
bank = read.csv("C:/Users/vcshw/Machine Learning and Stats/Sem1/Machine learning/lab/bank-full.csv", hea

bank = bank[,-12]
n = nrow(bank)
set.seed(12345)
id1=sample(1:n, floor(n*0.4))
train=bank[id1,]
d2 = bank[-id1,]
n2 = nrow(d2)
id2=sample(1:n2, floor(n2*0.5))
test=d2[id2,]
validate=d2[-id2,]
```

### 2.

Fitting decision tree to training data

```
library(tree)
dt_default = tree(y~.,data = train)
dt_size = tree(y~.,data = train, control = tree.control(nrow(train),minsize = 7000))
dt_dev = tree(y~.,data = train, control = tree.control(nrow(train),mindev = 0.0005))
```

Training data missclassification rate:

Default fit : 0.1048441

Min node size is 7000 fit : 0.1048441

Min deviance is .0005 fit : 0.0936187

Validation data missclassification rate:

Default fit : 0.1116927

Min node size is 7000 fit : 0.1116927

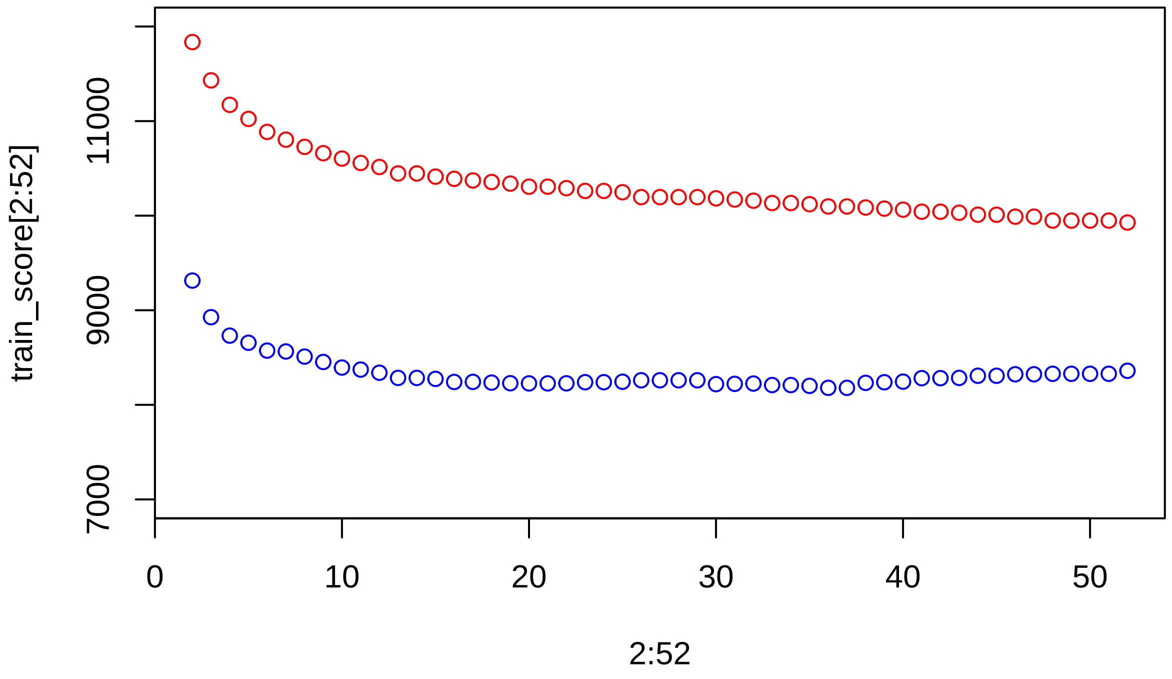
Min deviance is .0005 fit : 0.112946

Choosing the best of three fits : Though the missclassification error was least for mindev=0.0005 fit ,it has large tree of 150 terminal nodes, this leads to a slightly overfit tree. As a result of this , on fitting the validation data , the missclassification error rate slightly increases more in this when compared to other two trees. Model a and b , both have same missclassification error , however since the b has very small tree, we want to avoid the chances of underfit , hence we choose "a" as the best fit. But if we are allowed to find the optimal number of leaves to avoid overfitting, then c would be the best model.

In our case , setting the deviance very small ie 0.0005, made the tree grow more deeper as a large tree with 150 terminal nodes , this did reduce the missclassification on the training data, but ended up overfitting for validation data.

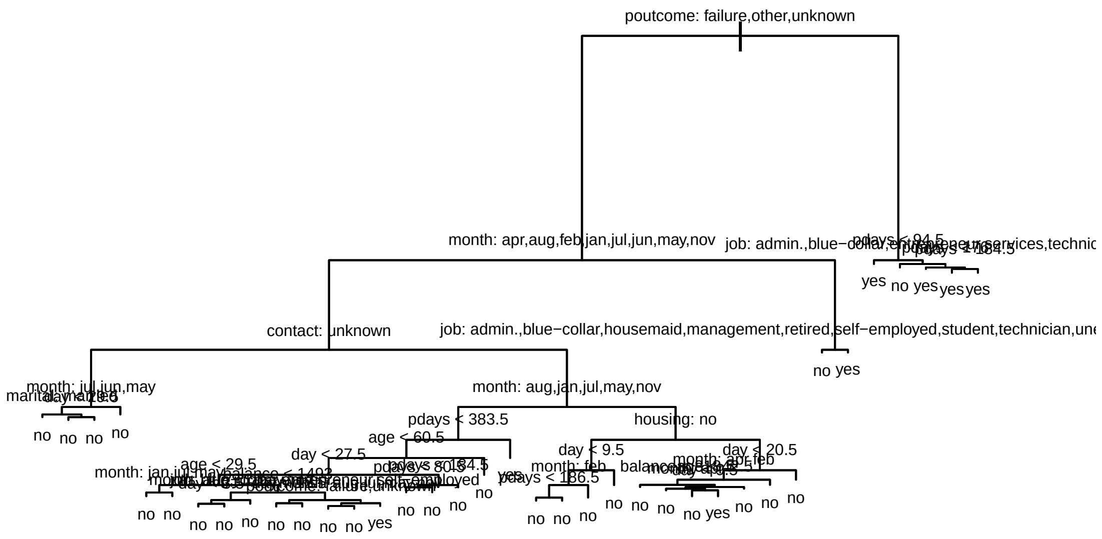
## 3.

Selecting optimal tree by training and validation



Optimal amount of leaves = 37

```
optimal_tree = prune.tree(dt_dev, best = 37)
plot(optimal_tree)
text(optimal_tree, pretty = 0, cex = 0.5)
```



poutcome: failure,other,unknown
month: apr,aug,feb,jan,jul,jun,may,nov
pdays < 94.5
contact: unknown
job: admin.,blue−collar,housemaid,management,retired,self−employed,student,technician,une
month: jul,jun,may
month: aug,jan,jul,may,nov
pdays < 383.5
housing: no
age < 60.5
day < 27.5
day < 9.5
day < 20.5
age < 29.5
pdays < 186.5
month: feb
month: apr,feb
yes
no

Variable : poutcome is the most important for decision making in this tree.

Tree sturcture : variable included = poutcome, month, contact, marital, day, pdays, age, balance, job and housing Number of leaves = 37 First variable considered in the tree is "poutcome", the partition is made by taking the condition poutcome = failure, other or unknown, if this is true , the data goes to left side to the next condition on variable month. If the poutcome was not satisfied , ie say poutcome was "success" then the data flows to right part , where next node condition is on pdays < 94.5. The tree continues until the number of leaves are 37.

Confusion matrix and missclassification rate for test data.

confusion matrix :

| | no | yes |
|---|---|---|
| no | 11868 | 122 |
| yes | 1347 | 226 |

Missclassification rate : 0.1083094

we can see that the misclassification rate for the test data has reduced for the optimal tree. Hence this is a better fit compared to the previously tried tree fits.

## 4.

Loss Matrix :

|  | no | yes |
|---|---|---|
| no | 0 | 1 |
| yes | 5 | 0 |

Confusion matrix :

|  | no | yes |
|---|---|---|
| no | 10965 | 1025 |
| yes | 745 | 828 |

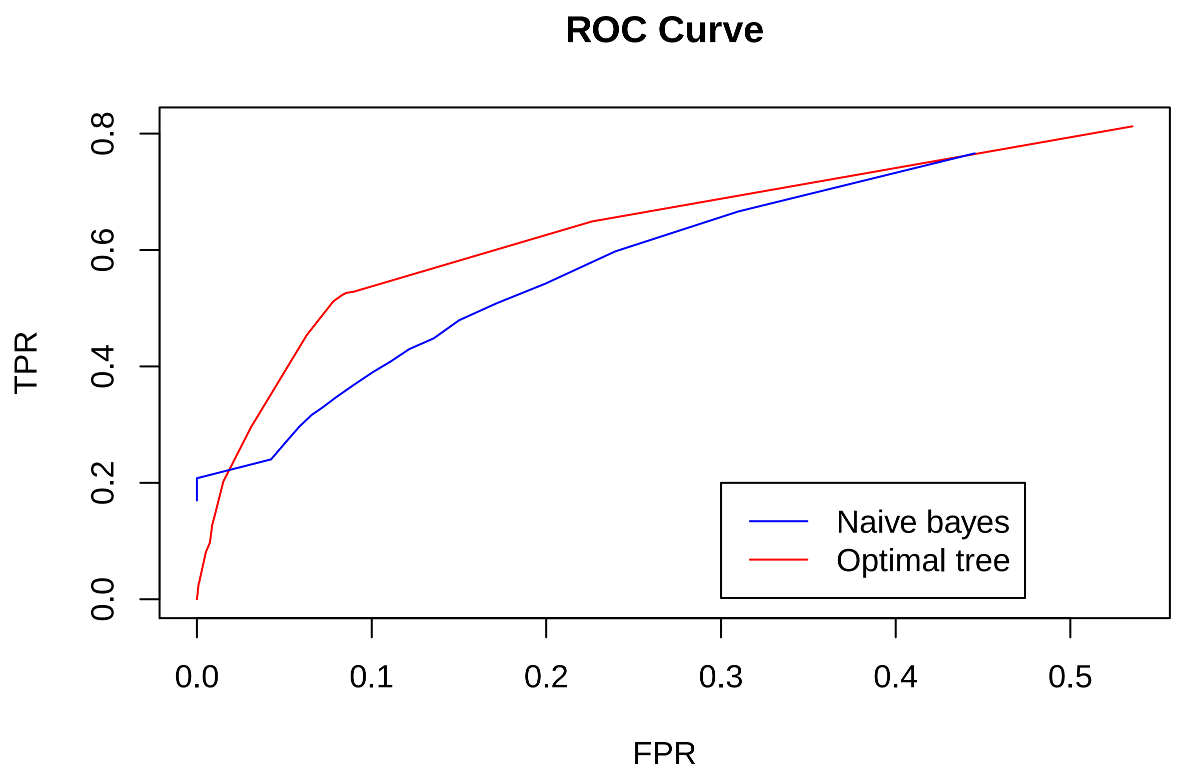
Missclassification rate : 0.1305021

Here in the loss function we can see that , penalty for predicting observed yes as no is 5 and no as yes is 1. So on applying loss function , as expected , the misclassification of an observed yes as no is reduced in the confusion matrix here. Previously observed yes predicted as no was 1347 , and now it is 745. However the missclassification error rate has increased.

We can try different loss matrix (assigning the loss function with suitable costs for the respective senario ) and choose the one which gives the lowest misclassification rate.

## 5.

Fitting naive bayes model, computing TPR and FPR for both models and plotting the ROC curve



Conclusion : Area under the curve is more for Optimal tree fit hence this is the best classifier.