Report on decision tree implementation

Submitted to

Kishan Kumar Ganguli

Lecturer

IIT, DU

Submitted by

Tulshi Chandra Das

BSSE0811

Introduction:

In this report I will describe about creating decision tree in c++ and R program and I will show the output.

Objective:

- Learning to create decision tree in raw code(c++) and R program.
- Analyzing the output.
- Clear concept on decision tree.

Methodology:

I have done development, running and testing manually. The raw code is in c++ program. I have used g++ compiler to compile the file.

For R program I have used party library.

VSCode is used as a development tool.

Result and Discussion

Raw decision tree code

Training Data:

File name: data.txt

14 5

outlook temperature humidity windy play

sunny hot high FALSE no

sunny hot high TRUE no

overcast hot high FALSE yes

rainy mild high FALSE yes

rainy cool normal FALSE yes

rainy cool normal TRUE no

overcast cool normal TRUE yes

sunny mild high FALSE no

sunny cool normal FALSE yes

rainy mild normal FALSE yes

sunny mild normal TRUE yes

overcast mild high TRUE yes

overcast hot normal FALSE yes

rainy mild high TRUE no

Decision Tree and test result:

```
PS E:\workspace\RSpace\dTree_assignment> .\dTree1.exe
outlook=sunny-->humidity=high-->-->no
outlook=sunny-->humidity=normal-->-->no
outlook=overcast-->humidity=high-->-->yes
outlook=overcast-->humidity=normal-->-->yes
outlook=rainy-->humidity=high-->-->yes
outlook=rainy-->humidity=normal-->-->yes
testing data:
rainy  hot  high  FALSE
play?
result:yes
PS E:\workspace\RSpace\dTree_assignment>
```

Test data: rainy(outlook), hot(temperature), high(humidity) FALSE (windy)

Result: yes (will play)

Decision tree code in R

Data: data generated using binomial and normal distribution function in R. Total 6000 row.

.

.

.

5985 0 B 62.29494

5986 1 B 62.18922

5987 0 B 62.10769

5988 0 B 61.99682

5989 1 B 61.92244

5990 0 B 61.61489

5991 0 B 61.55009

5992 0 B 61.44251

5993 0 B 61.42977

5994 0 B 61.35821

5995 0 B 61.31325

5996 0 B 61.30184

5997 0 B 61.21212

5998 1 B 60.51398

5999 0 B 60.38281

6000 0 B 59.08804

Decision Tree:

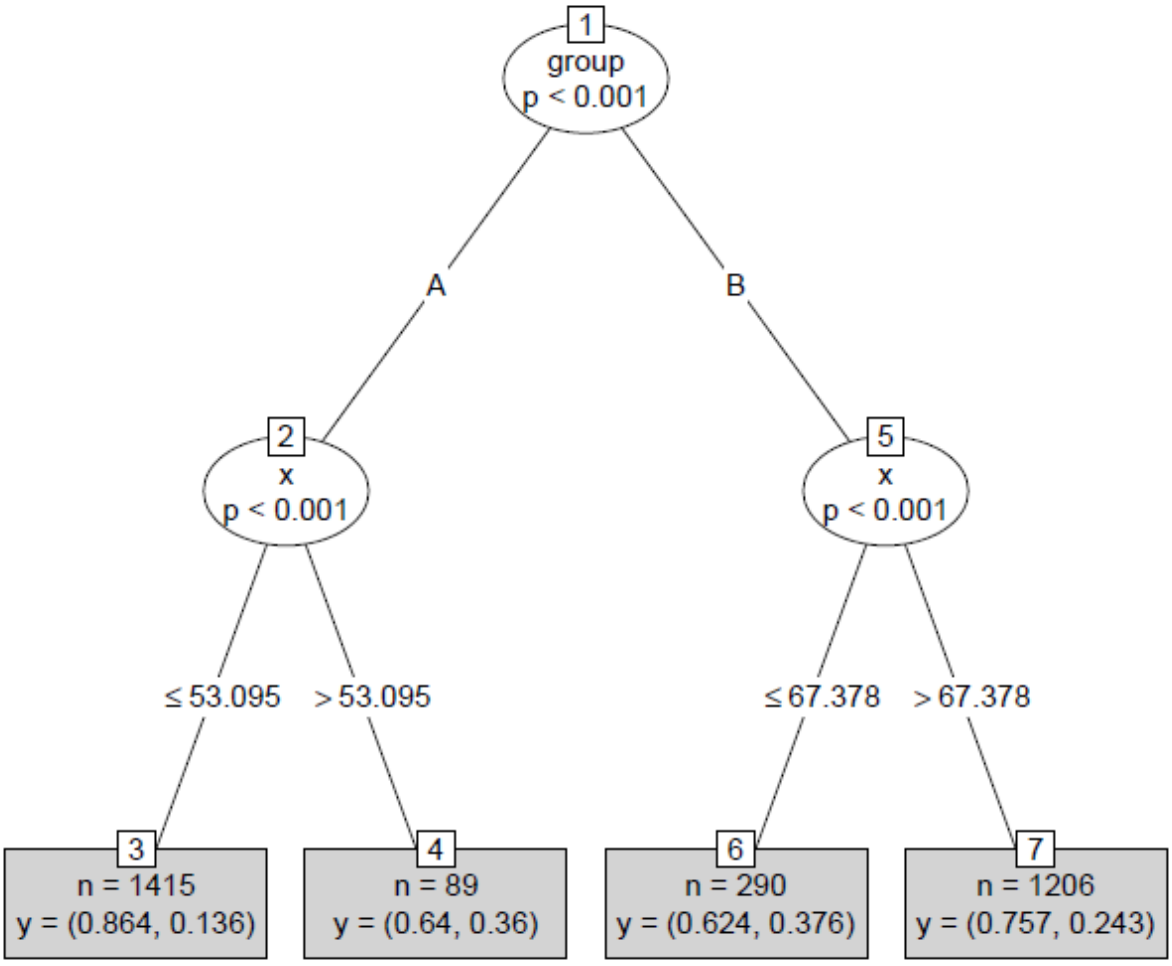

Figure 1:Decision Tree

Cross Validated Result (50%):

	0	1
0	2403	597
1	0	0

Library Implementation Description:

Preparing Data:

```
dt = data.frame(y = c(rbinom(n=2000,size=1,prob=0.1),
                      rbinom(n=2000,size=1,prob=0.2),
                      rbinom(n=2000,size=1,prob=0.3)),
                group = c(rep("A",3000), rep("B",3000)),
                x = c(sort(rnorm(3000,50,2)), sort(rnorm(3000,70,3), decreasing =
 T)))
```

the code above generate data of 6000 with 4 column data values.

```
dt$y = as.factor(dt$y)
```

this convert numeric data to categorical data.

```
rn = sample(1:nrow(dt), 3000)

dt_train = dt[rn,]
dt_test = dt[-rn,]
```

Separate data by 50% for training and 50% for cross validation.

```
plot(model, type="simple")
```

this draw the decision tree plot.

```
dt_test$predClass = predict(model, newdata=dt_test, type="response")     # obtain
the class (0/1)
dt_test$predProb = sapply(predict(model, newdata=dt_test,type="prob"),'[[',2)  #
obtain probability of class 1 (second element from the lists)
dt_test$predNode = predict(model, newdata=dt_test, type="node")   # obtain the pr
edicted node (in case you need it)
```

this perform test on model.

Conclusion

For both raw code and R implementation work on different dataset and build different type of two decision tree. The raw code gives mostly the possible result. For R implementation we see all output values are 0. It is because we don't give any threshold value. If we provide threshold value it will work more better.