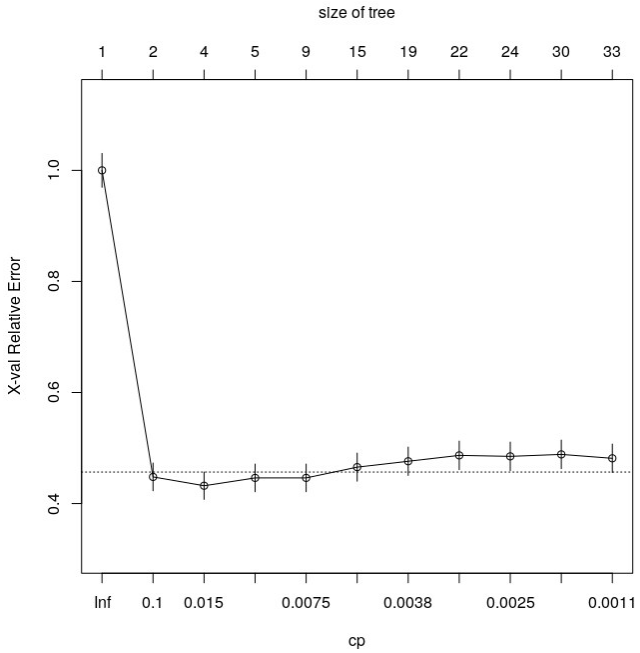
DECISION TREES AND RANDOM FORESTS

Illustrations to example code

You can run the example code using

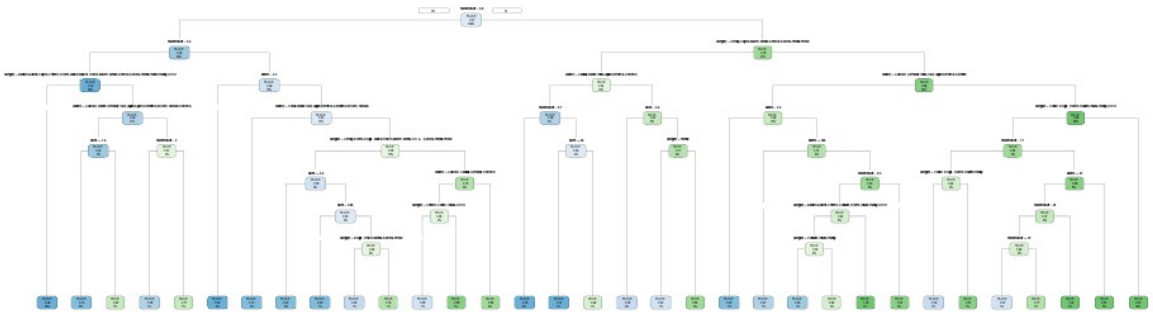
```
> source("forbes.dt.rf.R")
```

```
> printcp(model.DT)

Classification tree:
rpart(formula = profits ~ category + sales + assets + marketvalue + country,
      data = forbes.train, cp = 0.001)

Variables actually used in tree construction:
[1] assets      category    country     marketvalue      sales

Root node error: 567/1200 = 0.4725

n= 1200

           CP nsplit rel error  xerror     xstd
1  0.5520282      0   1.00000 1.00000 0.030501
2  0.0194004      1   0.44797 0.44797 0.024957
3  0.0123457      3   0.40917 0.43210 0.024627
4  0.0105820      4   0.39683 0.44621 0.024921
5  0.0052910      8   0.34392 0.44621 0.024921
6  0.0041152     14   0.30864 0.46561 0.025308
7  0.0035273     18   0.29101 0.47619 0.025512
8  0.0026455     21   0.28042 0.48677 0.025711
9  0.0023516     23   0.27513 0.48501 0.025678
10 0.0011758     29   0.26102 0.48854 0.025743
11 0.0010000     32   0.25750 0.48148 0.025612

> plotcp(model.DT)
```



Complex tree

```
> rpart.plot(model.DT, type=1, faclen=-3)
```

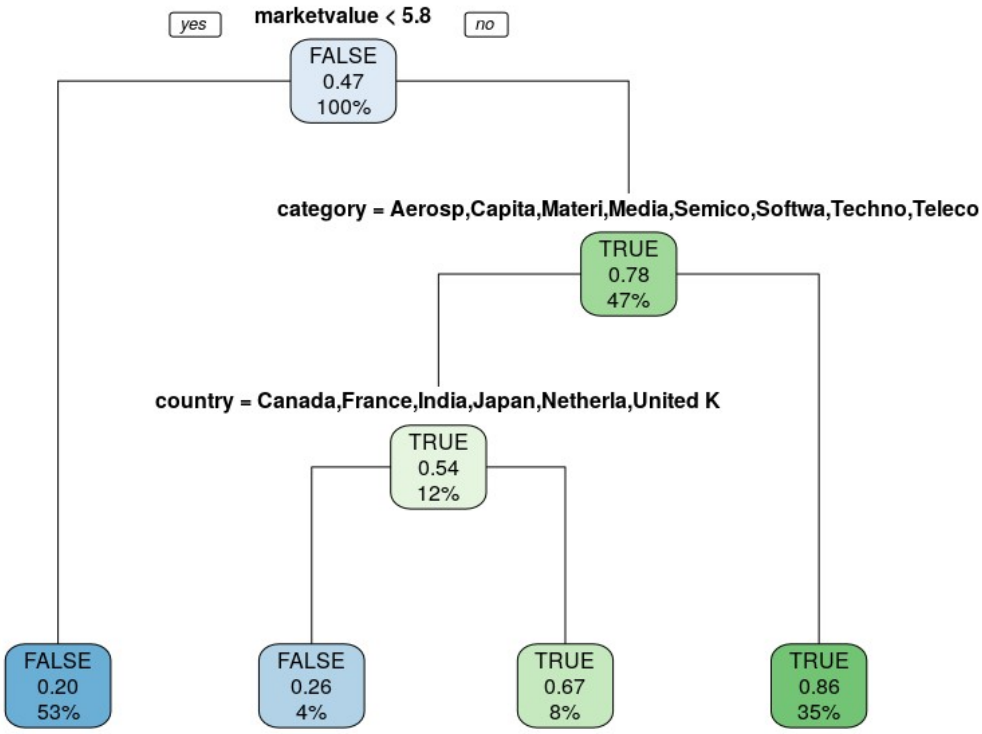


```
> prediction.train = predict(model.DT, forbes.train, type="class")
> error.rate.train = 1 - mean(prediction.train == forbes.train$profits)
> prediction.test = predict(model.DT, forbes.test, type="class")
> error.rate.test = 1 - mean(prediction.test == forbes.test$profits)

> cat("Training error rate =", round(error.rate.train*100,2), "%\n")
Training error rate = 12.17 %

> cat("Test error rate =", round(error.rate.test*100,2), "%\n\n")
Test error rate = 24.31 %
```

Pruned tree

```
> model.DT.pruned = prune(model.DT, cp=0.015)
> printcp(model.DT.pruned)

Variables actually used in tree construction:
[1] category    country     marketvalue

Root node error: 567/1200 = 0.4725

n= 1200

        CP nsplit rel error  xerror     xstd
1 0.55203      0   1.00000 1.00000 0.030501
2 0.01940      1   0.44797 0.44797 0.024957
3 0.01500      3   0.40917 0.43210 0.024627
```

```
> rpart.plot(model.DT.pruned, type=1, faclen=-3)
```



Training error rate = 19.33 %

Test error rate = 21.96 %

RANDOM FOREST

```
> print(model.RF)

Call:
 randomForest(formula = profits ~ category+sales+assets+marketvalue+country,
              data = forbes.train, ntree = 1000)

               Type of random forest: classification
                     Number of trees: 1000
No. of variables tried at each split: 2


        OOB estimate of  error rate: 18.42%
Confusion matrix:
      FALSE TRUE class.error
FALSE   501  132   0.2085308
TRUE     89  478   0.1569665


> table(forbes.train$profits)
      FALSE  TRUE
        633   567
```

Evaluation using test set

```
> prediction.test = predict(model.RF, forbes.test, type="class")
> error.rate.test = 1 - mean(prediction.test == forbes.test$profits)
> cat("Test error rate =", round(error.rate.test*100,2), "%\n\n")
```

Test error rate = 19.22 %

```
> CM.RF = table(prediction.test, forbes.test$profits)
> CM.RF

prediction.test FALSE TRUE
          FALSE   214   39         # negative predictions
          TRUE     59  198         # positive predictions
```