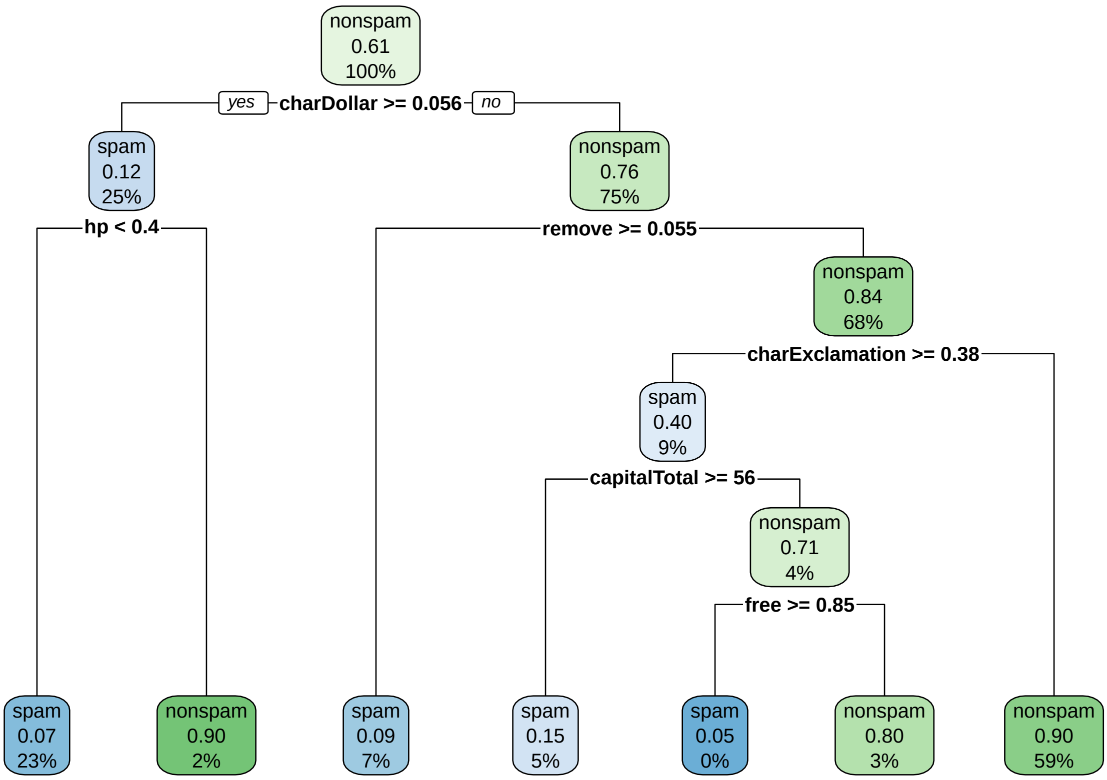
Solution 1:

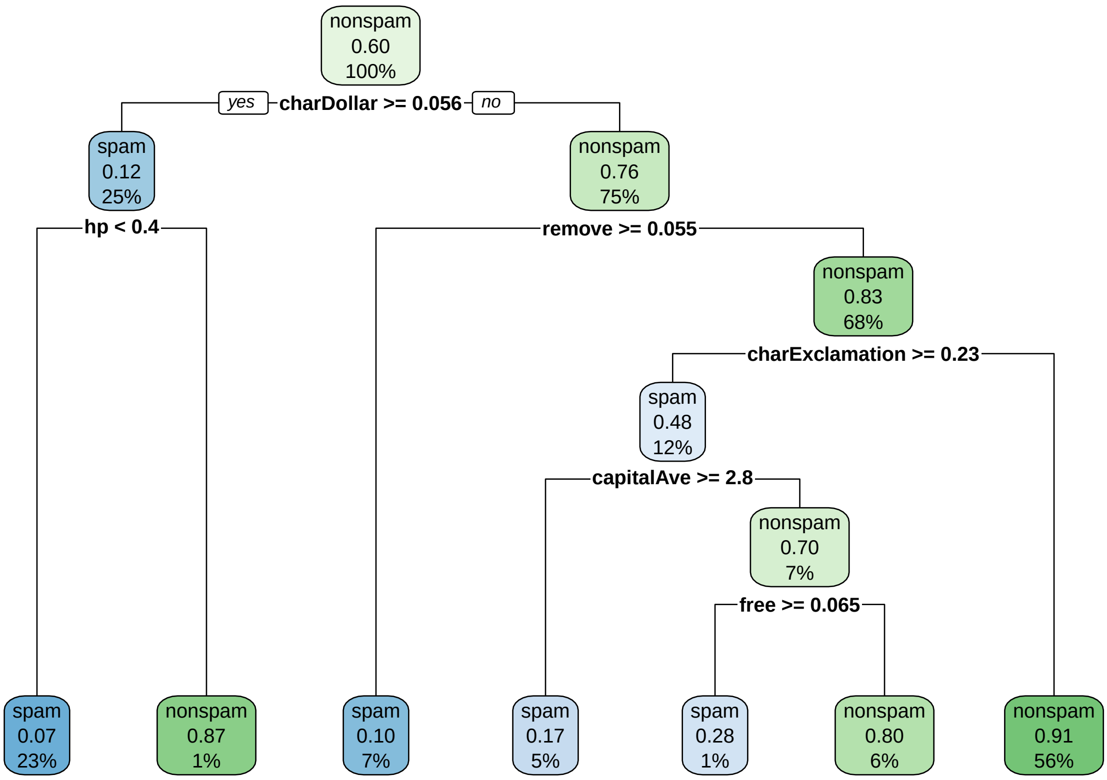
a) The spam data is a binary classification task where the aim is to classify an email as spam or no-spam.

```
library(mlr3)
library(mlr3learners)
library(mlr3filters)

tsk("spam")

## <TaskClassif:spam> (4601 x 58)
## * Target: type
## * Properties: twoclass
## * Features (57):
##   - dbl (57): address, addresses, all, business, capitalAve,
##     capitalLong, capitalTotal, charDollar, charExclamation, charHash,
##     charRoundbracket, charSemicolon, charSquarebracket, conference,
##     credit, cs, data, direct, edu, email, font, free, george, hp, hpl,
##     internet, lab, labs, mail, make, meeting, money, num000, num1999,
##     num3d, num415, num650, num85, num857, order, original, our, over,
##     parts, people, pm, project, re, receive, remove, report, table,
##     technology, telnet, will, you, your
```

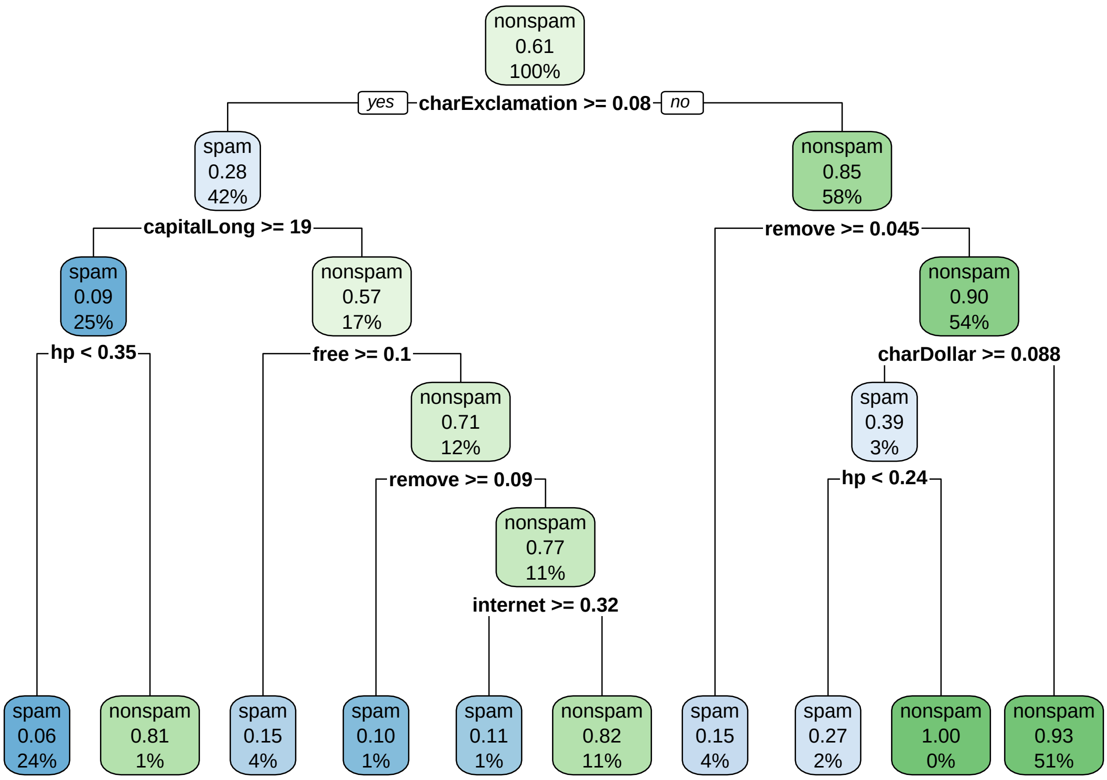
b)
```
library(rpart.plot)

## Loading required package:  rpart

task_spam <- tsk("spam")

learner <- lrn("classif.rpart")
learner$train(task_spam)

rpart.plot(learner$model, roundint=FALSE)
```




```r
set.seed(42)

subset1 <- sample.int(task_spam$nrow, size = 0.8 * task_spam$nrow)
subset2 <- sample.int(task_spam$nrow, size = 0.8 * task_spam$nrow)

learner$train(task_spam, row_ids = subset1)
rpart.plot(learner$model, roundint=FALSE)
```




```r
learner$train(task_spam, row_ids = subset2)
rpart.plot(learner$model, roundint=FALSE)
```



Observation: Trees with different sample find different split points and variables, leading to different trees!

c)
```
learner <- lrn("classif.ranger", "oob.error" = TRUE)
learner$train(tsk("spam"))

model <- learner$model

model$prediction.error

## [1] 0.04499022
```

d) Variable importance in general measures the contributions of features to a model. One way of computing the variable importance of the j-th variable is based on permutations of the OOB observations of the j-th variable, which measures the mean deacrease of the predictive accuracy induced by this permutation. To determine the n variables with the biggest influence on the prediction quality, one can choose the n variables with the highest variable importance based on permutations of the OOB, e.g. for $n = 5$:

```
learner <- lrn("classif.ranger", importance = "permutation", "oob.error" = TRUE)
filter <- flt("importance", learner = learner)
filter$calculate(tsk("spam"))
head(as.data.table(filter), 5)

##            feature      score
## 1:     capitalLong 0.04570583
## 2:              hp 0.04053346
## 3: charExclamation 0.04018373
## 4:          remove 0.03736120
## 5:      capitalAve 0.03457560
```

Solution 2:
See R code **randomForest_l_2.R**