Bagging and Random Forest Models

Brooke Anderson

February 5, 2016

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tree)
library(randomForest)
```

```
## randomForest 4.6-10

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library(caret)
```

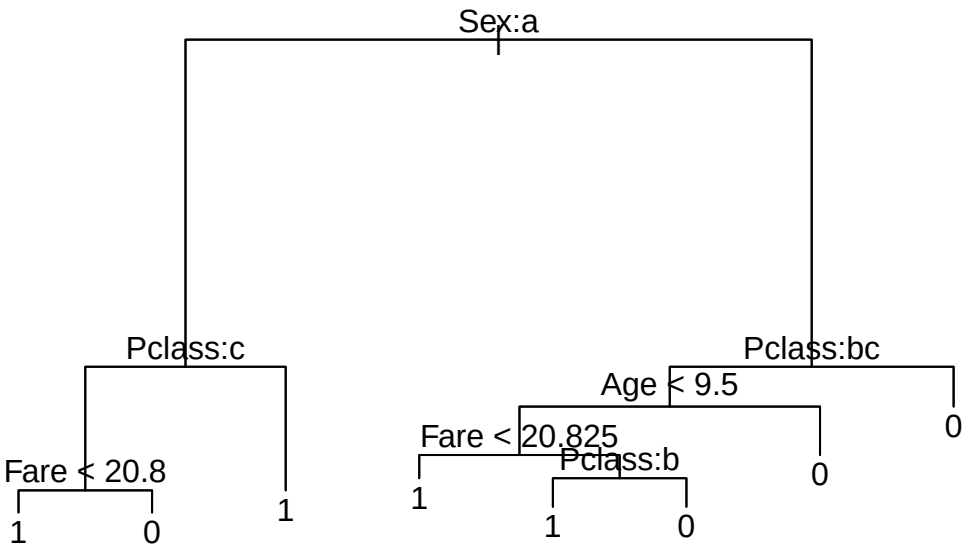
```
## Loading required package: lattice

## Loading required package: ggplot2

##
## Attaching package: 'ggplot2'

## The following object is masked from 'package:randomForest':
##
##     margin
```

Read in the data:

```
train <- read.csv("data/train.csv") %>%
  mutate(Survived = factor(Survived),
         Pclass = ordered(Pclass),
         Sex = factor(Sex)) %>%
  select(Survived, Pclass, Sex, Age, Fare, Embarked)
test <- read.csv("data/test.csv") %>%
  mutate(Pclass = ordered(Pclass),
         Sex = factor(Sex))  %>%
  select(Pclass, Sex, Age, Fare, Embarked)
test_ids <- read.csv("data/test.csv") %>%
            select(PassengerId)
```

Try fitting a simple tree model:

```
tree_1 <- tree(Survived ~ ., data = train)
plot(tree_1)
text(tree_1)
```



Try a random forest:

```
rf_mod_1 <- train(Survived ~ .,
                  data = train,
                  method = "rf",
                  metric = "Accuracy",
                  preProc = c("center", "scale"),
                  tuneLength = 20,
                  trControl = trainControl(method = "cv", number = 7))
```

```
## note: only 7 unique complexity parameters in default grid. Truncating the grid to 7 .
```

```
rf_mod_1
```

```
## Random Forest
##
```

```
## 891 samples
##   5 predictor
##   2 classes: '0', '1'
##
## Pre-processing: centered, scaled
## Resampling: Cross-Validated (7 fold)
##
## Summary of sample sizes: 611, 612, 613, 613, 612, 612, ...
##
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa      Accuracy SD  Kappa SD
##   2     0.8110088  0.5884716  0.05737870   0.13176991
##   3     0.8207589  0.6155643  0.05198858   0.11858890
##   4     0.8095130  0.5970654  0.03732161   0.08509169
##   5     0.8039791  0.5877493  0.03197251   0.07405594
##   6     0.7983629  0.5777964  0.02527018   0.06210477
##   7     0.7871169  0.5538257  0.03070313   0.07154961
##   8     0.7857712  0.5515526  0.03460931   0.07694610
##
## Accuracy was used to select the optimal model using  the largest value.
## The final value used for the model was mtry = 3.
```

```r
test_preds_1 <- predict(rf_mod_1, newdata = test)
test_preds <- rep(0, nrow(test))
test_preds[complete.cases(test)] <- as.numeric(test_preds_1) - 1
out <- cbind(test_ids, Survived = test_preds)
write.csv(out, file = "predictions/rf_cv.csv",
          row.names = FALSE)
```

When I tested the best random forest model (mtry picked using 10-fold cross-validation) on Kaggle, I got an accuracy of 0.77033.