Prediction Challenge 03

Jack Lin

Model 1: All attributes considered, no minsplit/minbucket

R code:

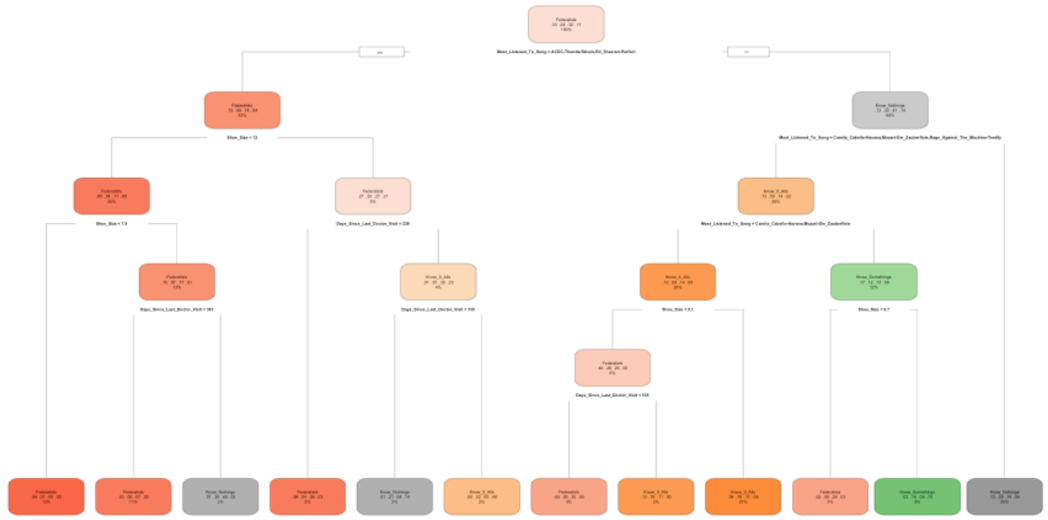
```
> library(rpart)
> library(rpart.plot)
> Model1 <- rpart(Party~., data=Facebook_Training)
> rpart.plot(Model1)
> predictedModel1 <- predict(Model1, newdata=Facebook_Training, type="class")
> model1.accuracy <- mean(predictedModel1 == Facebook_Training$Party)
> model1.accuracy
[1] 0.78325
> cross_validate(Facebook_Training,Model1,10,0.8)
[[1]]
   accuracy_subset accuracy_all
1          0.79750      0.79750
2          0.78500      0.78500
3          0.79875      0.79875
4          0.80625      0.80625
5          0.79500      0.79500
6          0.80250      0.80250
7          0.80500      0.80500
8          0.79625      0.79625
9          0.80875      0.80875
10         0.82250      0.82250

[[2]]
[[2]]$average_accuracy_subset
[1] 0.80175

[[2]]$average_accuracy_all
[1] 0.80175

[[2]]$variance_accuracy_subset
[1] 9.9375e-05

[[2]]$variance_accuracy_all
[1] 9.9375e-05
```



Model 2: minsplit=150

R code:

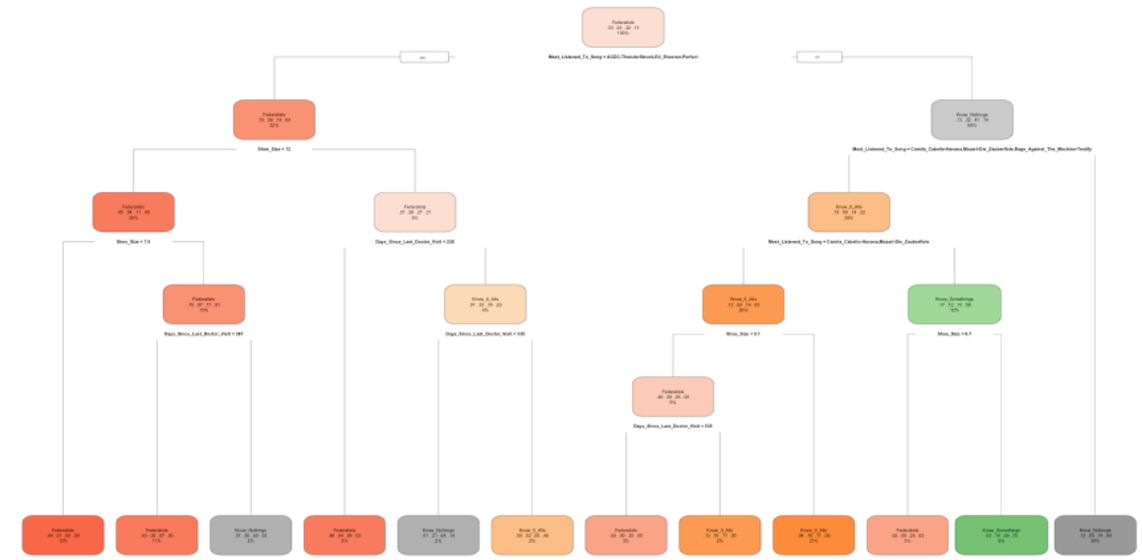
```
> Model2 <- rpart(Party~., data=Facebook_Training,control=rpart.control(minsplit=150)
> rpart.plot(Model2)
> predictedModel2 <- predict(Model2, newdata=Facebook_Training, type="class")
> model2.accuracy <- mean(predictedModel2 == Facebook_Training$Party)
> model2.accuracy
[1] 0.78325
> cross_validate(Facebook_Training,Model2,10,0.8)
[[1]]
   accuracy_subset accuracy_all
1          0.77125      0.78625
2          0.78000      0.80125
3          0.78250      0.79625
4          0.77500      0.79625
5          0.76250      0.76500
6          0.76500      0.79500
7          0.76750      0.77750
8          0.79625      0.81000
9          0.76750      0.78750
10         0.78375      0.79750

[[2]]
[[2]]$average_accuracy_subset
[1] 0.775125

[[2]]$average_accuracy_all
[1] 0.79125

[[2]]$variance_accuracy_subset
[1] 0.0001102257

[[2]]$variance_accuracy_all
[1] 0.0001635417
```



Model 3: minbucket=50

R code:

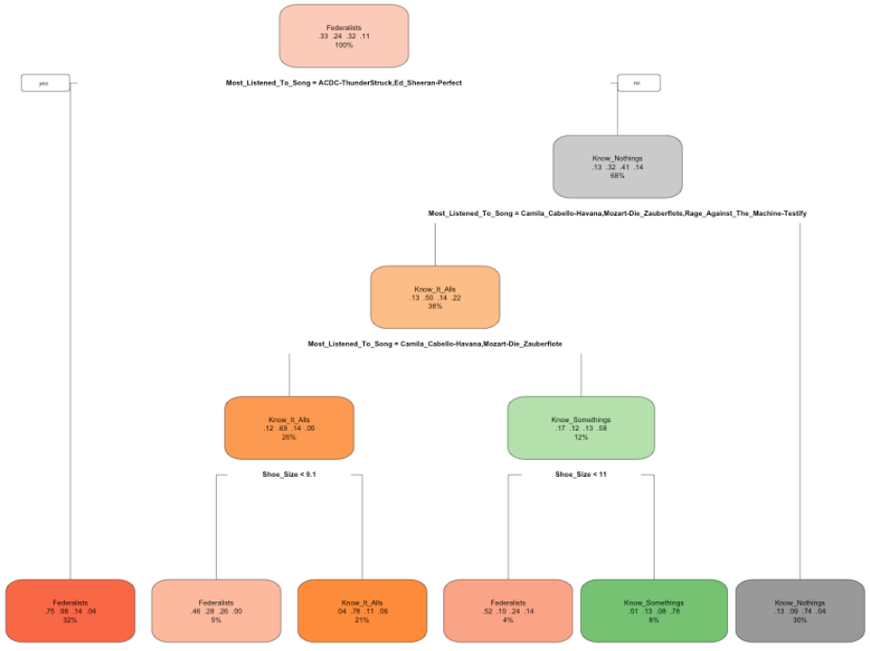
```
> Model3 <- rpart(Party~., data=Facebook_Training,control=rpart.control(minbucket=50)
> rpart.plot(Model3)
> predictedModel3<- predict(Model3, newdata=Facebook_Training, type="class")
> model3.accuracy <- mean(predictedModel3 == Facebook_Training$Party)
> model3.accuracy
[1] 0.78325
> cross_validate(Facebook_Training,Model3,10,0.8)
[[1]]
   accuracy_subset accuracy_all
1          0.75625      0.80500
2          0.76500      0.80000
3          0.75250      0.76500
4          0.79375      0.79875
5          0.77000      0.79000
6          0.79625      0.80250
7          0.73000      0.73000
8          0.76000      0.77875
9          0.76625      0.79750
10         0.77625      0.80000

[[2]]
[[2]]$average_accuracy_subset
[1] 0.766625

[[2]]$average_accuracy_all
[1] 0.78675

[[2]]$variance_accuracy_subset
[1] 0.000379184

[[2]]$variance_accuracy_all
[1] 0.0005511111
```



Lesson learned from Models 1–3:

1. Manipulating minsplit/minbucket does not improve accuracy.

2. The three factors common in Models 1–3 are:
(a) Most listened to song
(b) Days since last doctor visit
(c) Shoe size

What if we use just two of them to construct a decision tree? In Models 4–6, only two among (a)–(c) will be picked.

Model 4: Most listened to song + Doctor visit

R code:

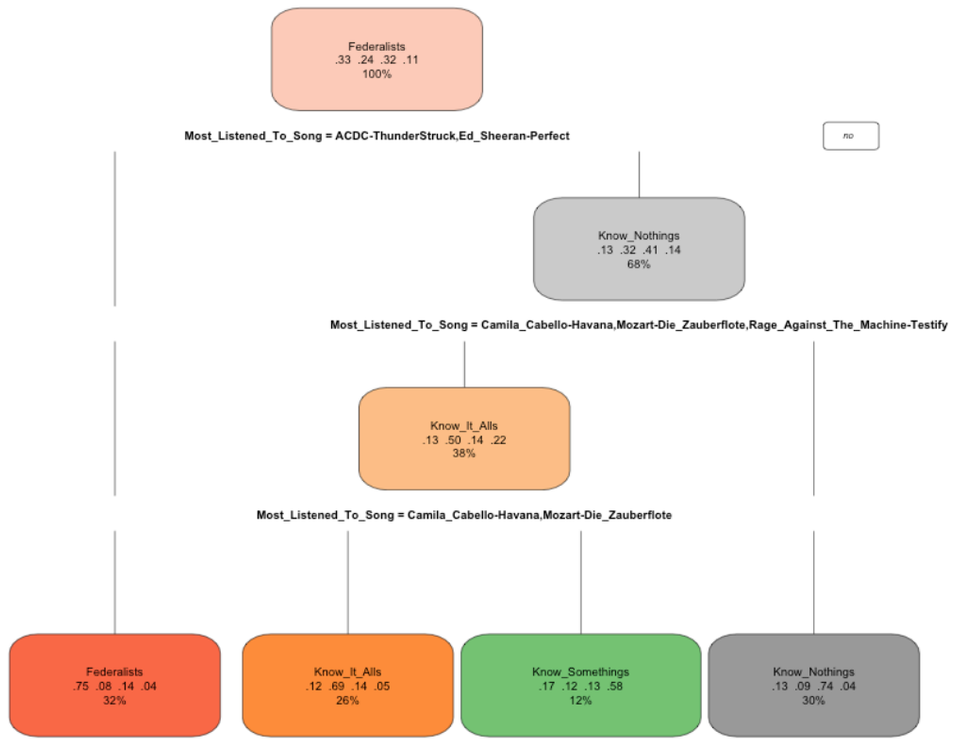
```
> Model4 <- rpart(Party~Most_Listened_To_Song+Days_Since_Last_Doctor_Visit, data=Facebook_Training)
> rpart.plot(Model4)
> predictModel4 <- predict(Model4, newdata=Facebook_Training, type="class")
> model4.accuracy <- mean(predictModel4==Facebook_Training$Party)
> model4.accuracy
[1] 0.70975
> cross_validate(Facebook_Training,Model4,10,0.8)
[[1]]
   accuracy_subset accuracy_all
1          0.71125      0.78250
2          0.71625      0.79375
3          0.69250      0.77375
4          0.70250      0.76500
5          0.70375      0.77125
6          0.71500      0.80000
7          0.71500      0.78625
8          0.69000      0.78875
9          0.69875      0.78750
10         0.71625      0.76625

[[2]]
[[2]]$average_accuracy_subset
[1] 0.706125

[[2]]$average_accuracy_all
[1] 0.7815

[[2]]$variance_accuracy_subset
[1] 0.0001008507

[[2]]$variance_accuracy_all
[1] 0.0001415972
```



Model 5: Most listened to song + Shoe size

R code:

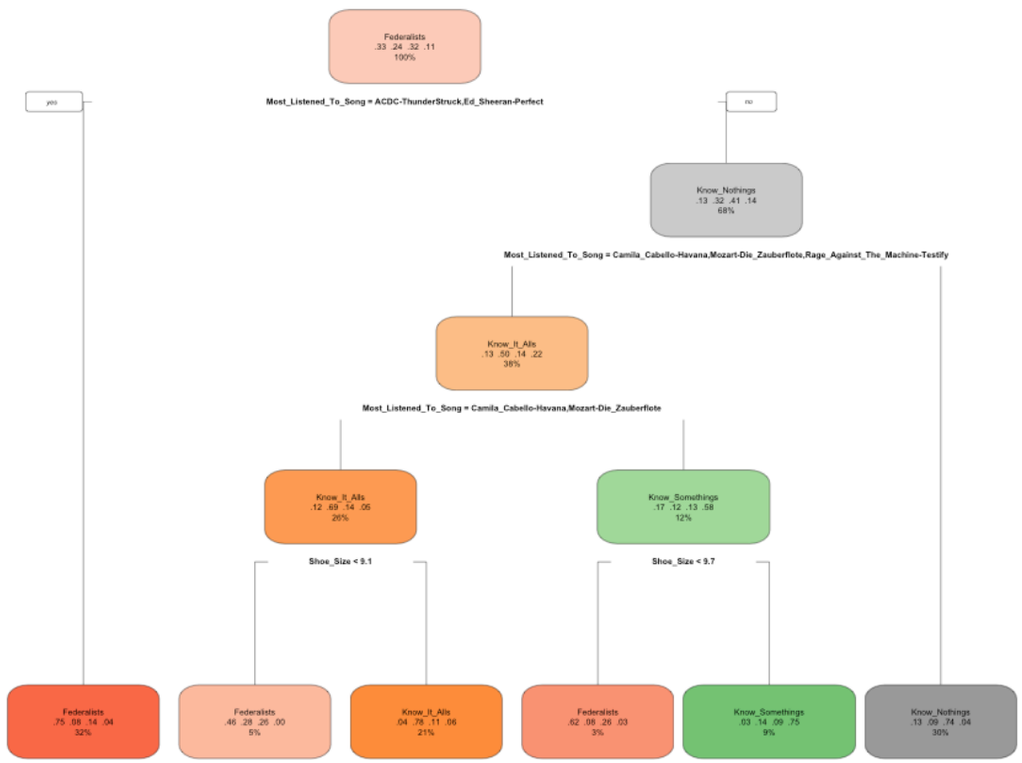
```
> Model5 <- rpart(Party~Most_Listened_To_Song+Shoe_Size, data=Facebook_Training)
> rpart.plot(Model5)
> predictModel5 <- predict(Model5, newdata=Facebook_Training, type="class")
> model5.accuracy <- mean(predictModel5==Facebook_Training$Party)
> model5.accuracy
[1] 0.73575
> cross_validate(Facebook_Training,Model5,10,0.8)
[[1]]
   accuracy_subset accuracy_all
1          0.71875      0.79000
2          0.73750      0.78000
3          0.69625      0.77500
4          0.74250      0.79125
5          0.72500      0.79375
6          0.74500      0.79750
7          0.72500      0.76875
8          0.75125      0.82125
9          0.70500      0.76375
10          0.74000      0.78625

[[2]]
[[2]]$average_accuracy_subset
[1] 0.728625

[[2]]$average_accuracy_all
[1] 0.78675

[[2]]$variance_accuracy_subset
[1] 0.0003234201

[[2]]$variance_accuracy_all
[1] 0.0002691667
```



Model 6: Doctor visit + Shoe size

R code:

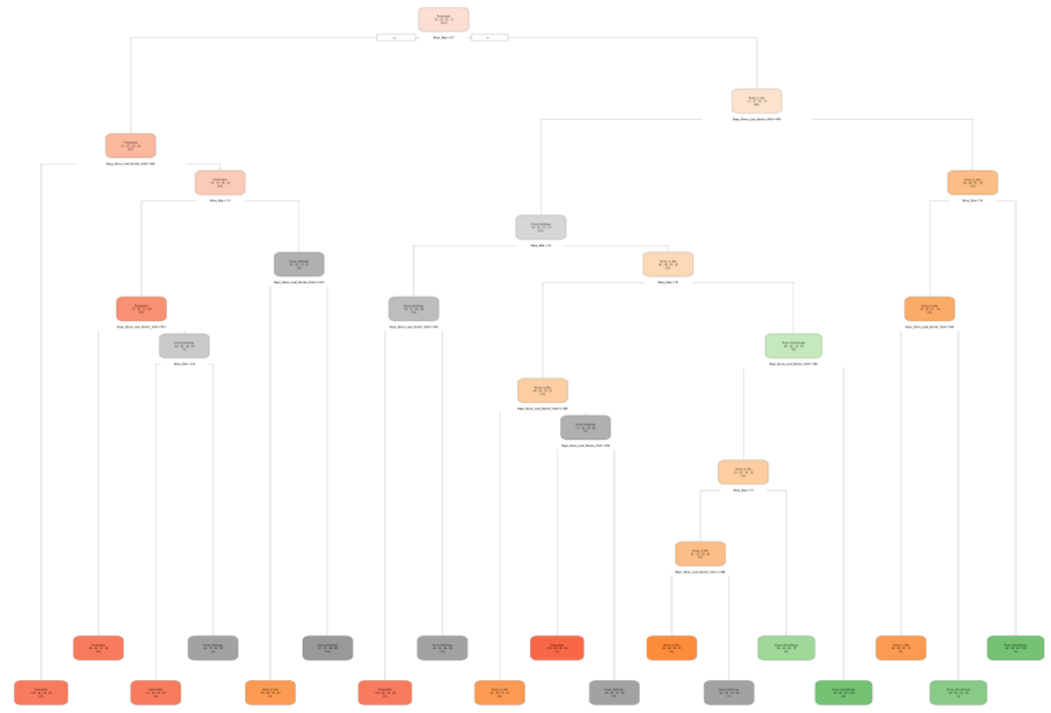
```
> Model6 <- rpart(Party~Days_Since_Last_Doctor_Visit+Shoe_Size, data=Facebook_Training, control=rpart.control(minsplit=100))
> predictModel6 <- predict(Model6, newdata=Facebook_Training, type="class")
> model6.accuracy <- mean(predictModel6==Facebook_Training$Party)
> model6.accuracy
[1] 0.96975
> cross_validate(Facebook_Training,Model6,10,0.8)
[[1]]
   accuracy_subset accuracy_all
1          0.94500      0.77500
2          0.95250      0.80750
3          0.94625      0.82625
4          0.95750      0.76250
5          0.94375      0.77000
6          0.96375      0.81500
7          0.95750      0.77500
8          0.95875      0.81750
9          0.95500      0.79125
10         0.94500      0.79500

[[2]]
[[2]]$average_accuracy_subset
[1] 0.9525

[[2]]$average_accuracy_all
[1] 0.7935

[[2]]$variance_accuracy_subset
[1] 5e-05

[[2]]$variance_accuracy_all
[1] 0.0005013194
```



Looks like we got a winner, and Model 6 will be implemented.

R code:

```
> FBtest_students <- read.csv("~/Documents/Rutgers/Data 101/Assignments/Prediction Challenge 3/FBtest_students.csv",
stringsAsFactors=FALSE)
>   View(FBtest_students)
> FBtest_students$Party <- predict(Model6,newdata=FBtest_students,type="class")
> Submission03 <- FBtest_students[,c(9,10)]
> write.csv(Submission03,'YuHonLinSubmission03.csv')
```

Results: 0.95250 on Kaggle public leaderboard.