```
# Week 4 - "Judge, Jury, and Classifier" Lecture


# VIDEO 4

# Read in the data
stevens = read.csv("stevens.csv")
str(stevens)

# Split the data
library(caTools)
set.seed(3000)
split = sample.split(stevens$Reverse, SplitRatio = 0.7)
Train = subset(stevens, split==TRUE)
Test = subset(stevens, split==FALSE)

# Install rpart library
install.packages("rpart")
library(rpart)
install.packages("rpart.plot")
library(rpart.plot)

# CART model
StevensTree = rpart(Reverse ~ Circuit + Issue + Petitioner +
 Respondent + LowerCourt + Unconst, method="class", data = Train,
 control=rpart.control(minbucket=25))
prp(StevensTree)

# Make predictions
PredictCART = predict(StevensTree, newdata = Test, type = "class")
table(Test$Reverse, PredictCART)
(41+71)/(41+36+22+71)

# ROC curve
library(ROCR)

PredictROC = predict(StevensTree, newdata = Test)
PredictROC

pred = prediction(PredictROC[,2], Test$Reverse)
perf = performance(pred, "tpr", "fpr")
plot(perf)



# VIDEO 5 - Random Forests

# Install randomForest package
```

```r
install.packages("randomForest")
library(randomForest)

# Build random forest model
StevensForest = randomForest(Reverse ~ Circuit + Issue + Petitioner +
 Respondent + LowerCourt + Unconst, data = Train, ntree=200,
 nodesize=25 )

# Convert outcome to factor
Train$Reverse = as.factor(Train$Reverse)
Test$Reverse = as.factor(Test$Reverse)

# Try again
StevensForest = randomForest(Reverse ~ Circuit + Issue + Petitioner +
 Respondent + LowerCourt + Unconst, data = Train, ntree=200,
 nodesize=25 )

# Make predictions
PredictForest = predict(StevensForest, newdata = Test)
table(Test$Reverse, PredictForest)
(40+74)/(40+37+19+74)



# VIDEO 6

# Install cross-validation packages
install.packages("caret")
library(caret)
install.packages("e1071")
library(e1071)

# Define cross-validation experiment
fitControl = trainControl( method = "cv", number = 10 )
cartGrid = expand.grid( .cp = (1:50)*0.01)

# Perform the cross validation
train(Reverse ~ Circuit + Issue + Petitioner + Respondent +
 LowerCourt + Unconst, data = Train, method = "rpart", trControl =
 fitControl, tuneGrid = cartGrid )

# Create a new CART model
StevensTreeCV = rpart(Reverse ~ Circuit + Issue + Petitioner +
 Respondent + LowerCourt + Unconst, method="class", data = Train,
 control=rpart.control(cp = 0.18))

# Make predictions
PredictCV = predict(StevensTreeCV, newdata = Test, type = "class")
```

```
table(Test$Reverse, PredictCV)
(59+64)/(59+18+29+64)
```