Alternative Turnout Models - Fit to Monroe County

```
library(randomForest)
```

```
## randomForest 4.6-12

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

```
library(ROCR)
```

```
## Warning: package 'ROCR' was built under R version 3.3.3

## Loading required package: gplots

##
## Attaching package: 'gplots'

## The following object is masked from 'package:stats':
##
##     lowess
```

```
library(ggplot2)
```

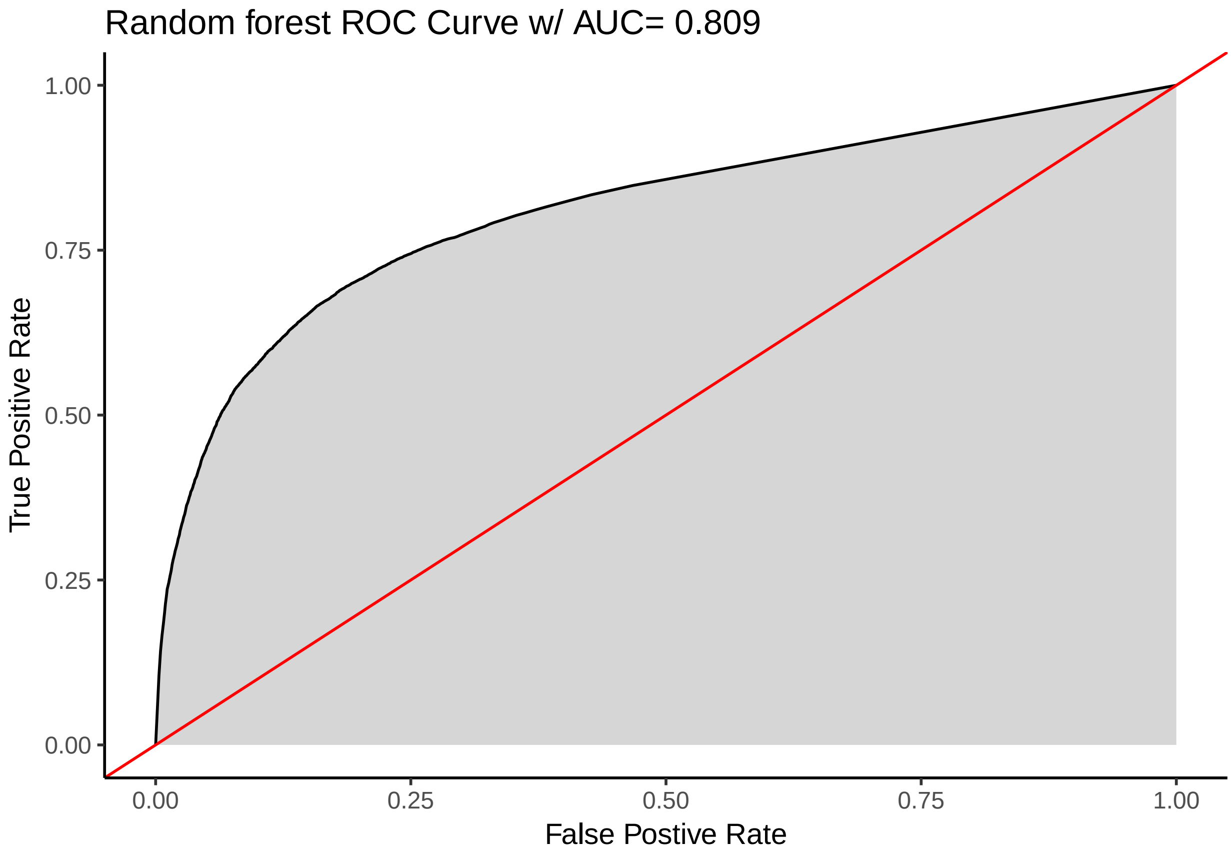
```
##
## Attaching package: 'ggplot2'

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

```
monroe_data <- as.data.frame(read.csv("c:/Dropbox (UFL)//incubator/monroe_data.csv"))
#monroe_data <- as.data.frame(read.csv("/Users/isaac/Dropbox (UFL)//incubator/monroe_data.csv"))
test_election = monroe_data['X2014.11']
train_election = monroe_data['X2010.11']

red_train_data <- as.data.frame(read.csv("c:/Dropbox (UFL)//incubator/monroe_red_train_dat.csv"))[,-1]
#red_train_data <- as.data.frame(read.csv("/Users/isaac/Dropbox (UFL)//incubator/monroe_red_train_dat.c
head(red_train_data)
```

```
##   X2006.01 X2006.03 X2006.04 X2006.09 X2006.11 X2007.03 X2007.04 X2007.05
## 1        0        0        0        0        0        0        0        0
## 2        0        0        0        0        1        0        0        0
## 3        0        0        0        0        0        0        0        0
## 4        0        0        0        0        1        0        0        0
## 5        0        0        0        0        0        0        0        0
## 6        0        0        0        0        0        0        0        0
##   X2007.10 X2007.11 X2008.01 X2008.03 X2008.04 X2008.08 X2008.11 X2009.03
## 1        0        0        0        0        0        0        0        0
## 2        0        0        0        0        0        0        1        0
## 3        0        0        1        0        0        0        1        0
## 4        0        0        0        0        0        0        1        0
## 5        0        0        0        0        0        0        0        0
## 6        0        0        0        0        0        0        0        0
##   X2009.07 X2009.10 X2009.11 X2010.03 X2010.08 X2011.10 X2011.11 X2012.01
## 1        0        0        0        0        0        0        0        0
## 2        0        0        0        0        0        0        0        0
## 3        0        0        0        0        1        0        1        1
## 4        0        0        0        0        0        0        0        0
## 5        0        0        0        0        0        0        0        0
## 6        0        0        0        0        0        0        0        0
##   X2012.08 X2012.11 X2013.03 X2013.10 X2013.11 X2014.03 X2014.08
## 1        0        1        0        0        0        0        0
## 2        0        1        0        0        0        0        0
## 3        1        1        0        0        0        0        1
## 4        0        1        0        0        0        0        0
## 5        0        1        0        0        0        0        0
## 6        0        0        0        0        0        0        0
##   ResidenceZipcode Gender Race PartyAffiliation Precinct VoterStatus
## 1                0      0    0                0        0           0
## 2                1      1    0                1        1           0
## 3                2      1    0                0        2           0
## 4                1      0    0                0        1           0
## 5                3      1    0                0        3           0
## 6                4      0    0                0        4           0
##   Years_reg
## 1 0.7619048
## 2 0.8412698
## 3 0.5396825
## 4 0.5238095
## 5 0.4761905
## 6 0.4761905
```

Random Forest

```r
mtry.grid <- data.frame(mtry = c(2, 4, 8, 16, 20))

RF_tune <- randomForest(x = red_train_data, y= as.factor(train_election[[1]]), mtry = mtry.grid$mtry[3]

preds <- RF_tune$predicted
test_election <- as.factor(test_election[[1]])

misclassRF <- mean(preds != test_election)
probs <- predict(RF_tune, newdata = red_train_data, type="prob")[,2]


pred <- prediction(probs, test_election)
perf <- performance(pred, measure = "tpr", x.measure = "fpr")
# I know, the following code is bizarre. Just go with it.
auc <- performance(pred, measure = "auc")
auc <- auc@y.values[[1]]
roc.data <- data.frame(fpr=unlist(perf@x.values),
           tpr=unlist(perf@y.values),
           model="RF")

ggplot(roc.data, aes(x=fpr, ymin=0, ymax=tpr)) +
  geom_ribbon(alpha=0.2) +
  geom_line(aes(y=tpr)) +
  xlab("False Postive Rate") +
  ylab("True Positive Rate") +
  geom_abline(slope=1, colour = "red") +
  ggtitle(paste("Random forest ROC Curve w/ AUC=", round(auc,3)))+
  theme_classic()
```



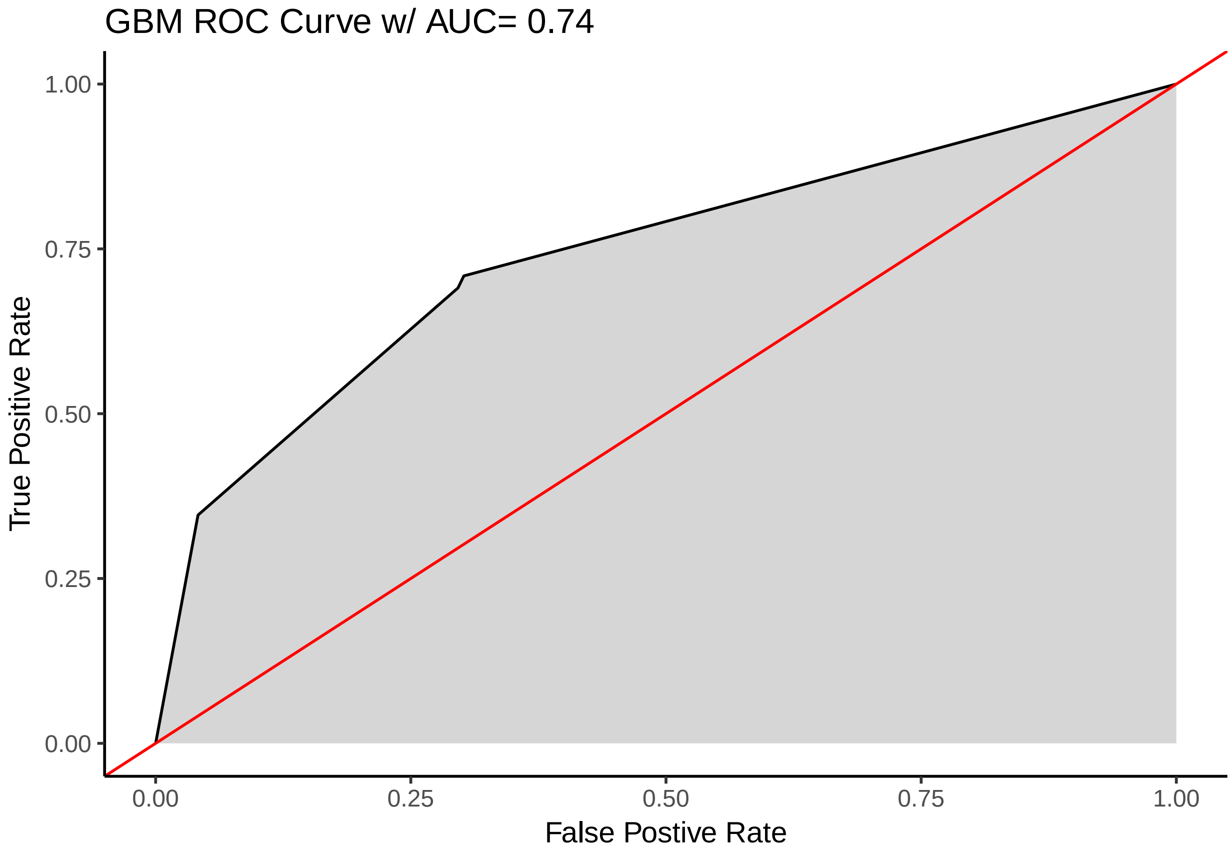
GBM

```
library(gbm)
```

```
## Warning: package 'gbm' was built under R version 3.3.3

## Loading required package: survival

## Loading required package: lattice

## Loading required package: splines

## Loading required package: parallel

## Loaded gbm 2.1.3
```

```
#train_election <- as.factor(train_election[[1]])

gbm_tune <- gbm.fit(x=red_train_data,  y=train_election[[1]], verbose = F, n.trees = 500)

preds <- round(predict(gbm_tune,n.trees = 500, type="response"),0)
test_election <- monroe_data['X2014.11'][[1]]
```

```r
misclassRF <- mean(preds != test_election)
probs <- predict(gbm_tune, newdata = red_train_data, n.trees = 200, type="response")


pred <- prediction(probs, test_election)
perf <- performance(pred, measure = "tpr", x.measure = "fpr")
# I know, the following code is bizarre. Just go with it.
auc <- performance(pred, measure = "auc")
auc <- auc@y.values[[1]]
roc.data <- data.frame(fpr=unlist(perf@x.values),
           tpr=unlist(perf@y.values),
           model="RF")

ggplot(roc.data, aes(x=fpr, ymin=0, ymax=tpr)) +
  geom_ribbon(alpha=0.2) +
  geom_line(aes(y=tpr)) +
  xlab("False Postive Rate") +
  ylab("True Positive Rate") +
  geom_abline(slope=1, colour = "red") +
  ggtitle(paste("GBM ROC Curve w/ AUC=", round(auc,3)))+
  theme_classic()
```



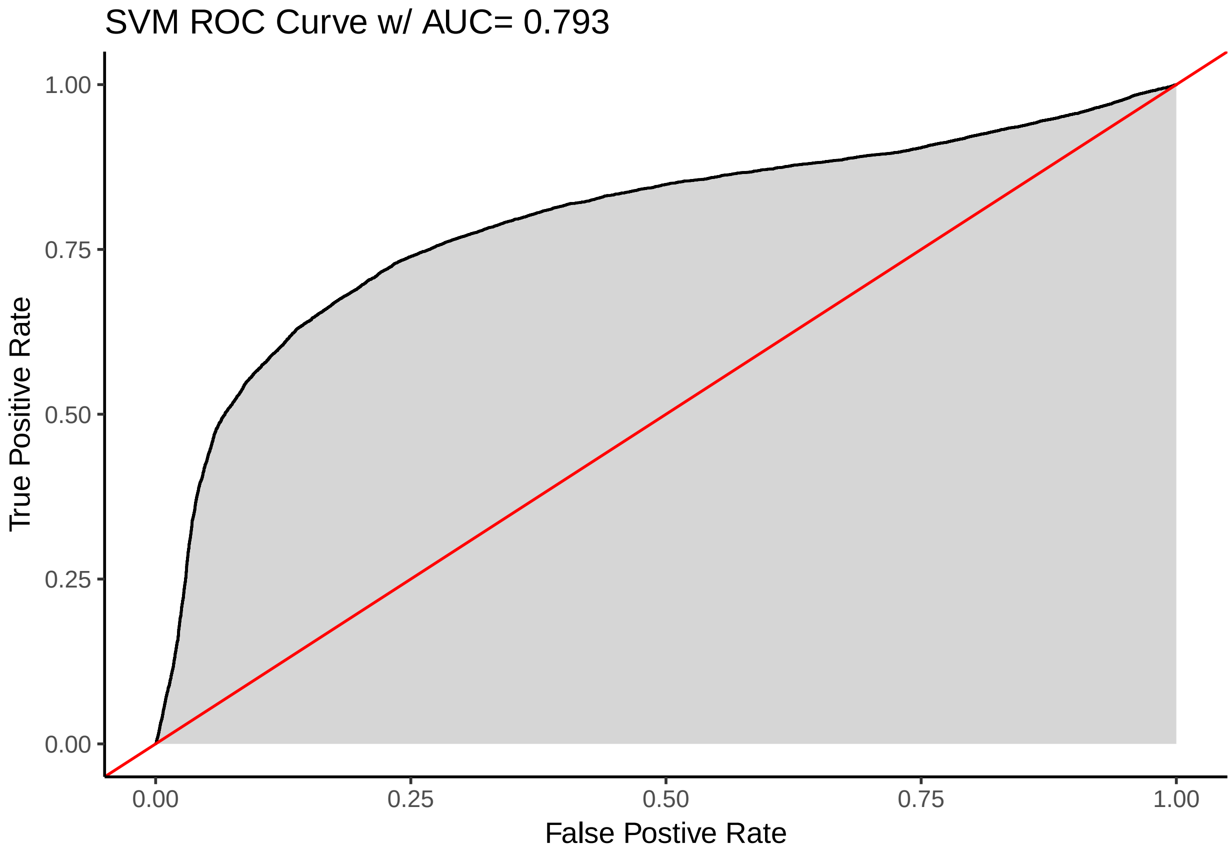
SVM

```
library(kernlab)
```

```
##
## Attaching package: 'kernlab'
```

```
## The following object is masked from 'package:ggplot2':
##
##     alpha
```

```
#train_election <- as.factor(train_election[[1]])

svm_tune <- ksvm(x=as.matrix(red_train_data),  y=as.numeric(train_election[[1]]), kernel="rbfdot")

probs <- predict(svm_tune, newdata = red_train_data, type="response")
preds <- round(probs,0)

test_election <- monroe_data['X2014.11'][[1]]

misclassRF <- mean(preds != test_election)


pred <- prediction(probs, test_election)
perf <- performance(pred, measure = "tpr", x.measure = "fpr")
# I know, the following code is bizarre. Just go with it.
auc <- performance(pred, measure = "auc")
auc <- auc@y.values[[1]]
roc.data <- data.frame(fpr=unlist(perf@x.values),
           tpr=unlist(perf@y.values),
           model="RF")

ggplot(roc.data, aes(x=fpr, ymin=0, ymax=tpr)) +
  geom_ribbon(alpha=0.2) +
  geom_line(aes(y=tpr)) +
  xlab("False Postive Rate") +
  ylab("True Positive Rate") +
  geom_abline(slope=1, colour = "red") +
  ggtitle(paste("SVM ROC Curve w/ AUC=", round(auc,3)))+
  theme_classic()
```


SVM ROC Curve w/ AUC= 0.793
True Positive Rate
1.00
0.75
0.50
0.25
0.00
0.00
0.25
0.50
0.75
1.00
False Postive Rate