PSTAT131_FinalProject

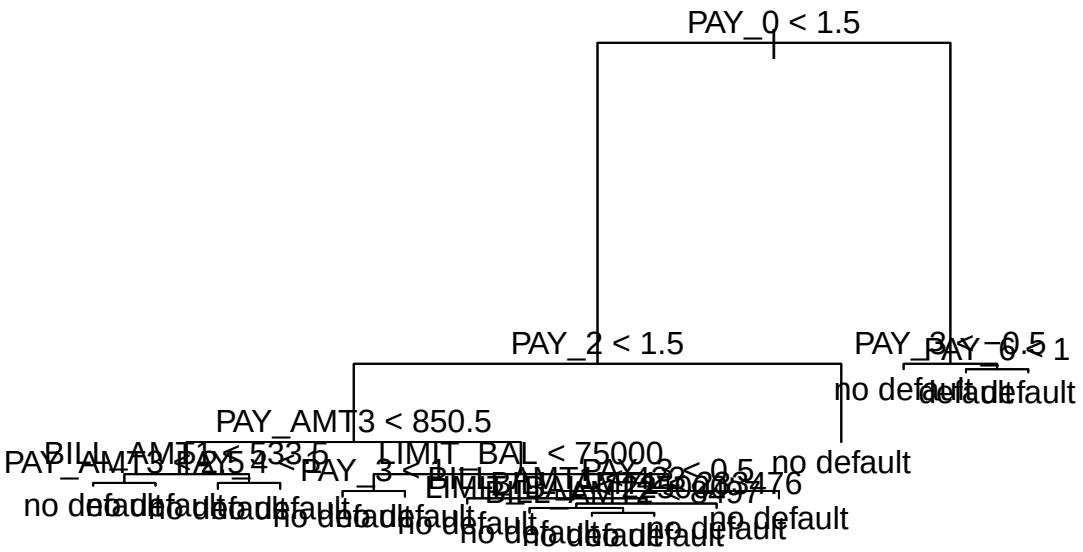
Daniel Ko, Josh Lee, Cindy Liang, Jesus Castellanos

May 31, 2016

```
##creating the dataset
defaultdata1 <- read.xls("~/Desktop/credit_default.xls", header = TRUE)
defaultdata1 <- defaultdata1[,-1]
defaultdata1 <- defaultdata1[-30001,]

# have to reformat final column for classification
defaultdata1$default.payment.next.month<-factor(defaultdata1$default.payment.next.month, levels = c(0,1)
```

CLASSIFICATION TREE

```
# creating training and test data sets
set.seed(1)
sub = sample(nrow(defaultdata1), size = 10000)
traindef = defaultdata1[-sub,]
testdef = defaultdata1[sub,]

#creating prune tree
deftree = tree(default.payment.next.month~., data = traindef, control = tree.control(nrow(defaultdata1)
prune.deftree <- prune.misclass(deftree, best = 16)
#plot & summary of pruned tree
plot(prune.deftree)
text(prune.deftree, pretty = 0)
```




```
# error rate for prned tree ~ 17.9%
```

```
#ROC Curve for Pruned Tree
t1.pred <- predict(prune.deftree, testdef)
t1.pred <- data.table(t1.pred)
treepred.data <- prediction(t1.pred[,1,with=FALSE], testdef$default.payment.next.month)
perf.treepred <- performance(treepred.data, measure = "tpr", x.measure = "fpr")
```

BAGGED TREE

```
# creating the bagged tree
bag.def = randomForest(default.payment.next.month~., data = traindef, mtry = 23, importance = TRUE, ntr
# error rate for bagged tree ~ 18.95%
```

```
#Roc Curve for the bagged tree
t2.pred = predict(bag.def, testdef, type = "prob")
t2.pred = data.table(t2.pred)

bag.roc <- prediction(t2.pred[,1,with=FALSE], testdef$default.payment.next.month)
bag.perf <- performance(bag.roc, measure = "tpr", x.measure = "fpr")
```

RANDOM FOREST

```
set.seed(1)
rf.def = randomForest(default.payment.next.month~., data = traindef, mtry = 5, importance = TRUE, ntree
rf.def
```

```
##
## Call:
##  randomForest(formula = default.payment.next.month ~ ., data = traindef,      mtry = 5, importance =
##                Type of random forest: classification
##                      Number of trees: 100
## No. of variables tried at each split: 5
##
##         OOB estimate of  error rate: 18.5%
## Confusion matrix:
##            no default default class.error
## no default      14673     927  0.05942308
## default          2772    1628  0.63000000
```

```
# error rate for random forest ~ 18.5%
```

```
# Roc Curve for Random Forest
t3.pred = predict(rf.def, testdef, type = "prob")
t3.pred = data.table(t3.pred)

rf.roc <- prediction(t3.pred[,1,with=FALSE], testdef$default.payment.next.month)
rf.perf <- performance(rf.roc, measure = "tpr", x.measure = "fpr")
```

KNN CROSS VALIDATION

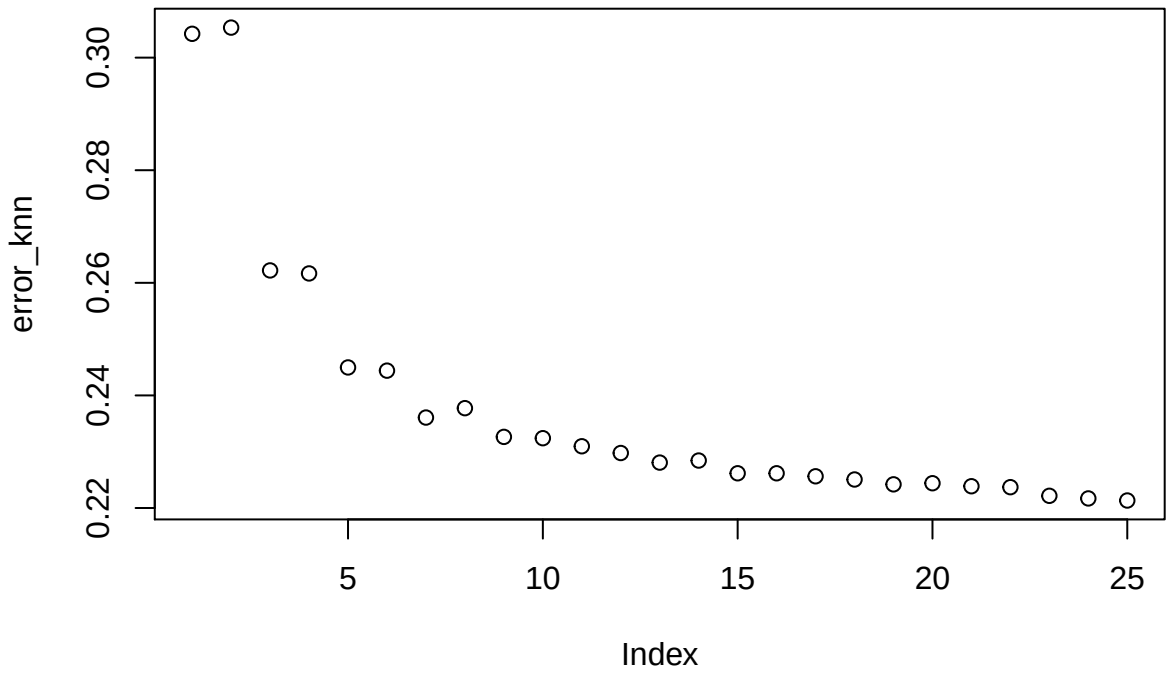
```
##knn cross validation
error_knn <- NULL
for (i in 1:25) {
  pred_class  <- knn.cv(train = defaultdata1[,-24], cl=defaultdata1[,24],k = i)
  error_knn[i] <-mean(pred_class != defaultdata1[,24])
}

min(error_knn)
```

```
## [1] 0.2213333
```

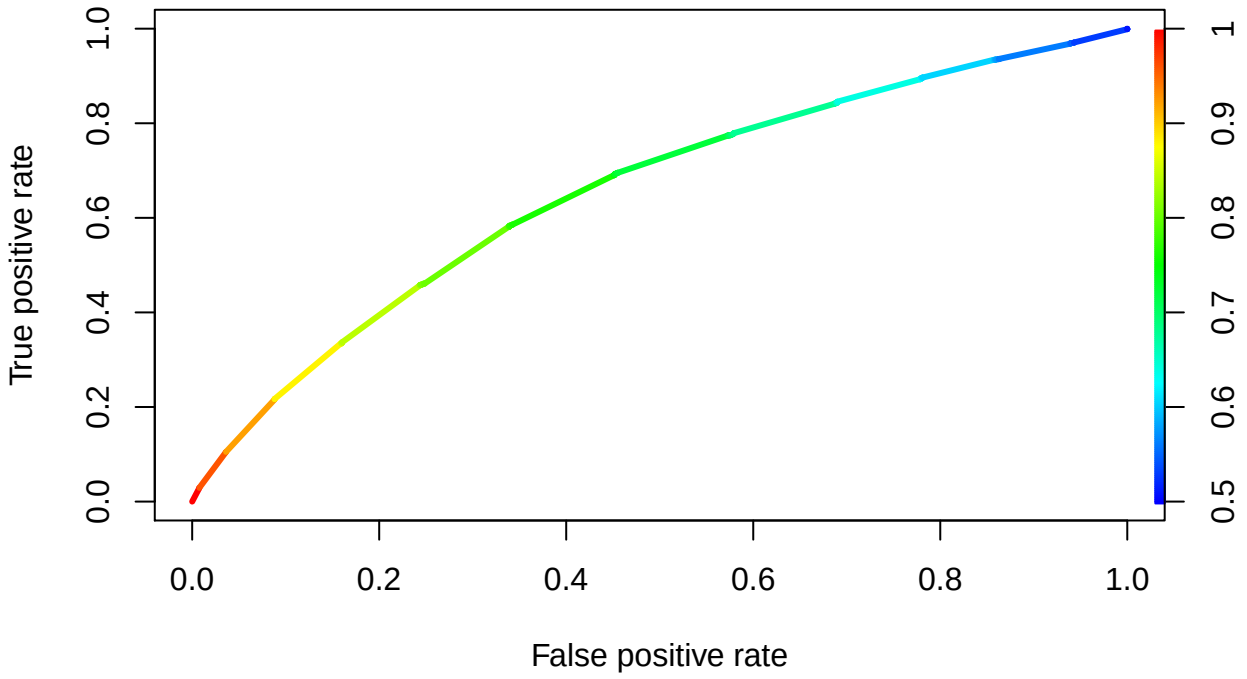
```
which.min(error_knn)
```

```
## [1] 25
```

```
plot(error_knn)
```

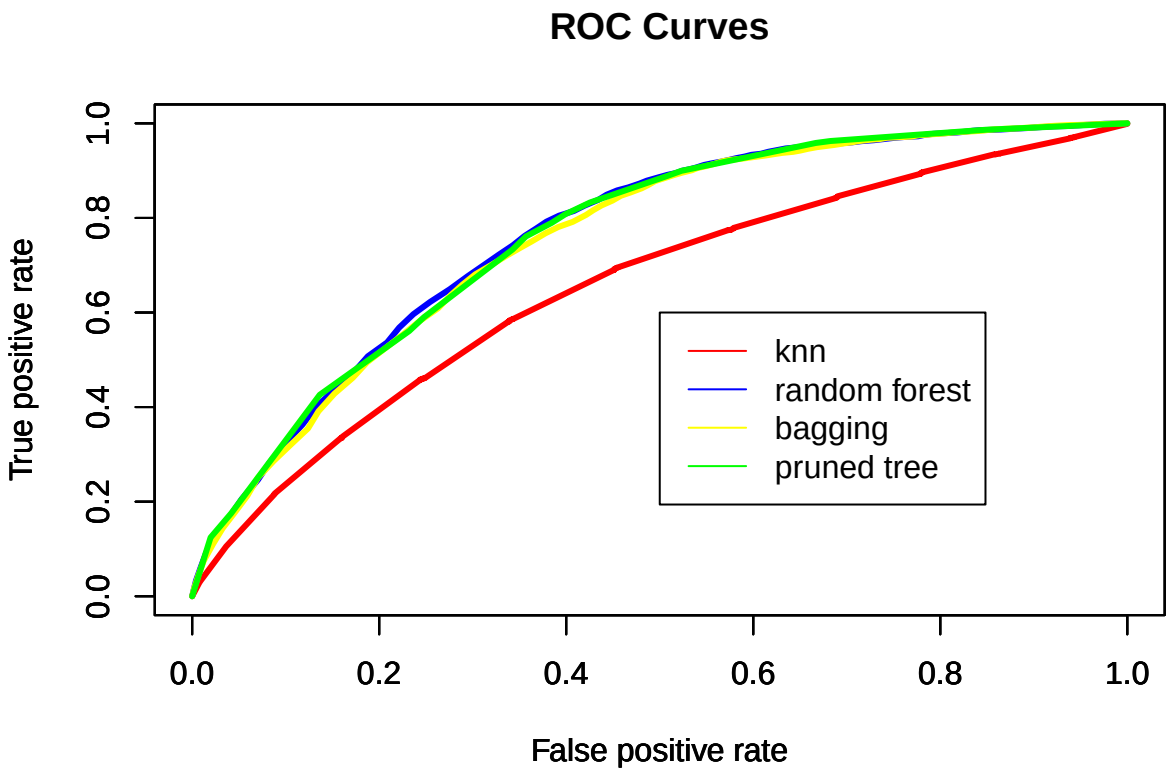



```
# ROC Curve for KNN
knn.pred=knn(traindef[,-24],testdef[,-24],traindef[,24],k=25,prob=TRUE)
knn.p=attributes(knn.pred)$prob #probabilites of spam

knn.roc <- prediction(knn.p, testdef[,24])
knn.perf <- performance(knn.roc, measure= "tpr", x.measure = "fpr")
plot(knn.perf, colorize = T, lwd = 3)
```



PLOTTING ROC CURVES TOGETHER

```
plot(knn.perf, col="red",lwd=3)
par(new=TRUE)
plot(rf.perf, col="blue",lwd=3)
par(new=TRUE)
plot(bag.perf, col="yellow",lwd=3)
par(new=TRUE)
plot(perf.treepred, col="green",lwd=3, main = 'ROC Curves')

legend(.5,.6,legend = c('knn','random forest','bagging','pruned tree '), col = c('red', 'blue','yellow'
```



LOGISTIC REGRESSION MODEL

```
traindef2 = data.table(traindef)
testdef2 = data.table(testdef)
glm.fit=glm(default.payment.next.month~., data=traindef2, family=binomial)

glm.prob=predict(glm.fit,testdef2,type="response")
glm.pred=rep(0,10000)
glm.pred[glm.prob >.5]=1
glm.pred=factor(glm.pred,levels=c(0,1),labels=c("no default","default"))
glm.test.error=mean(glm.pred != testdef2$default.payment.next.month)
table(glm.pred,testdef2$default.payment.next.month)
```

```
##
## glm.pred     no default default
##   no default       7532    1685
##   default           232     551
```

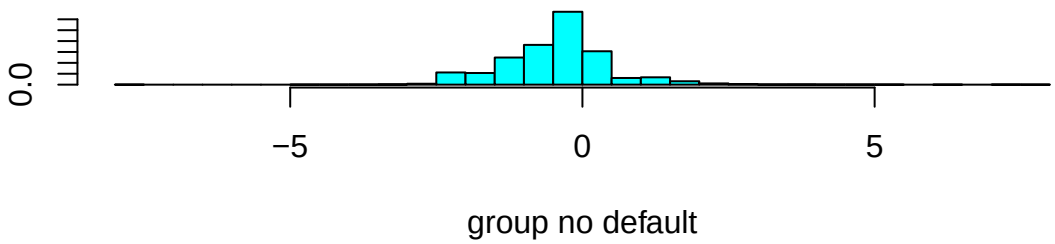
```
glm.test.error
```

```
## [1] 0.1917
```

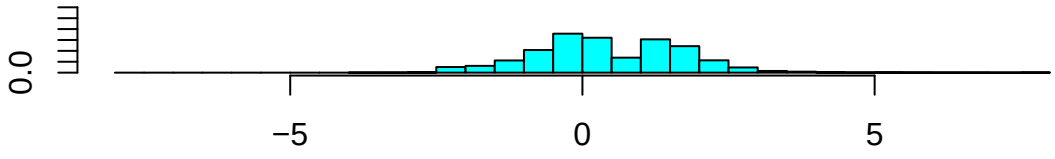
```
#test error rate is ~ 19.17%
```

LDA

```
lda.fit=lda(default.payment.next.month~.,data=traindef2)
plot(lda.fit)
```



group no default



group default

```
lda.pred = predict(lda.fit,testdef2)
lda.test.error = mean(lda.pred$class != testdef2$default.payment.next.month)
lda.test.error
```

```
## [1] 0.1913
```

```
table(lda.pred$class, testdef2$default.payment.next.month)
```

```
##
##              no default default
##   no default       7503    1652
##   default           261     584
```

```
# test error rate is ~ 19.13%
```