Untitled

Yue Lai

5/15/2020

import data

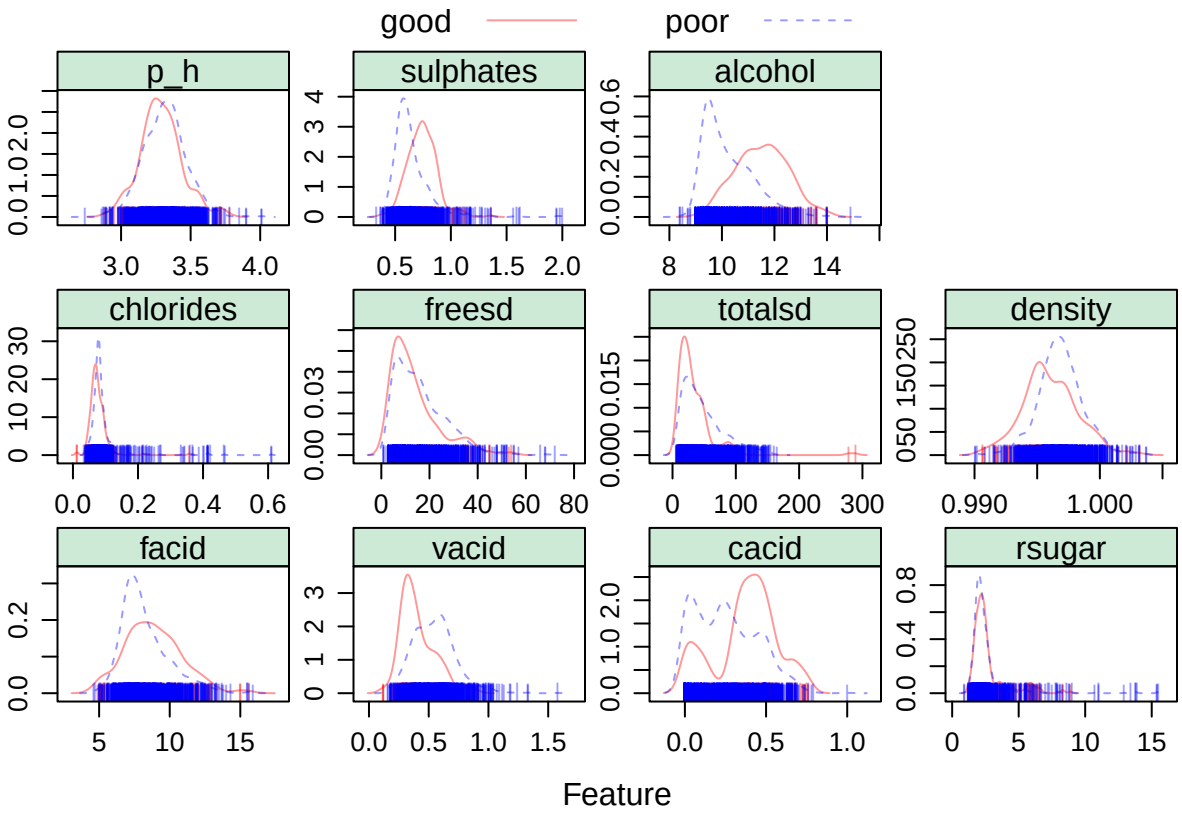
```
red = read_excel(path = "./data/wine.xlsx", sheet = "red") %>%
  janitor::clean_names()

wine = data.frame(red)

wine = wine %>%
  mutate(quality = as.factor(ifelse(quality > 6.5, "good", "poor")))
```

```
theme1 = transparentTheme(trans = .4)
theme1$strip.background$col = rgb(.0, .6, .2, .2)
trellis.par.set(theme1)

featurePlot(x = wine[,1:11],
            y = wine$quality,
            scales = list(x = list(relation = "free"),
                          y = list(relation = "free")),
            plot = "density", pch = "|",
            auto.key = list(columns = 2))
```



Divide the data into two part (training and test)

```
rowtrain = createDataPartition(y = wine$quality,
                               p = 2/3,
                               list = FALSE)

ctrl = trainControl(method = "repeatedcv",
                    repeats = 5,
                    summaryFunction = twoClassSummary,
                    classProbs = TRUE)
```

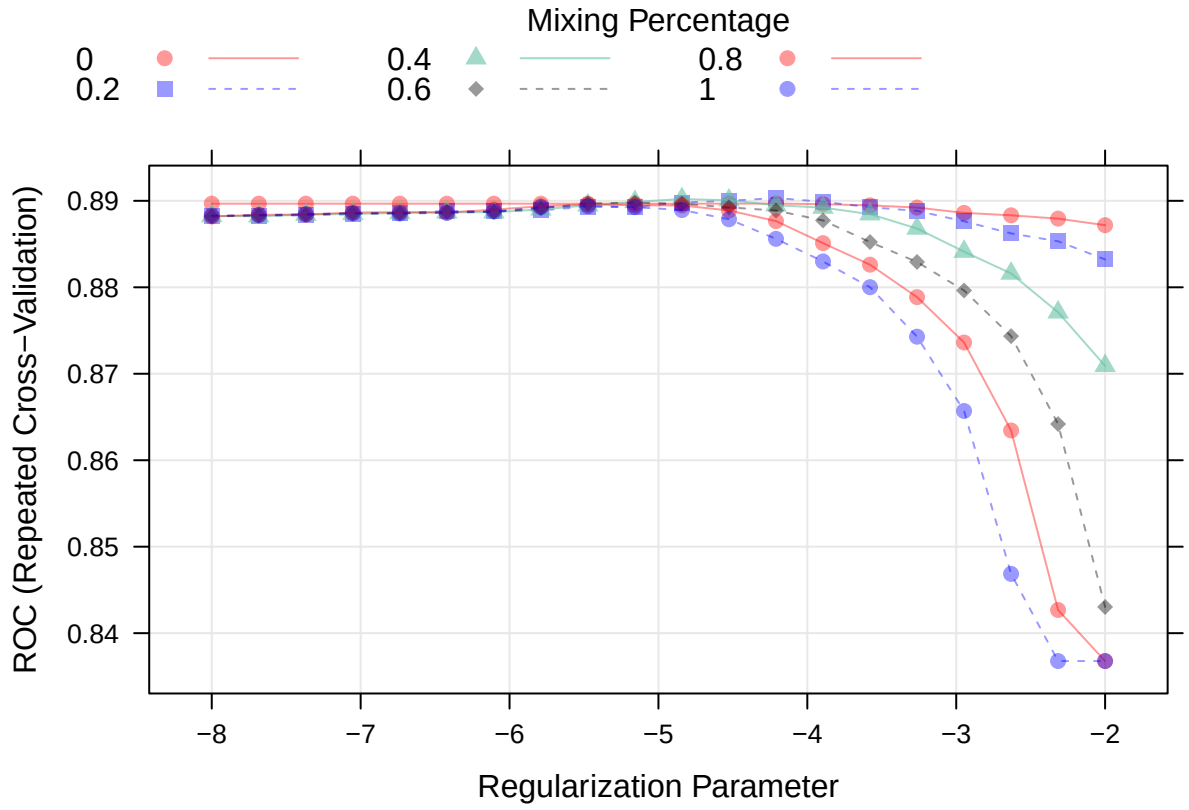
glm

```
set.seed(1)
model.glm = train(x = wine[rowtrain, 1:11],
                  y = wine$quality[rowtrain],
                  method = "glm",
                  metric = "ROC",
                  trControl = ctrl)
```

glmn

```
set.seed(1)
glmngrid = expand.grid(.alpha = seq(0,1,length = 6),
                       .lambda = exp(seq(-8,-2, length = 20)))

model.glmn = train(x = wine[rowtrain, 1:11],
                   y = wine$quality[rowtrain],
                   method = "glmnet",
                   metric = "ROC",
                   tuneGrid = glmngrid,
                   trControl = ctrl)
plot(model.glmn, xTrans = function(x) log(x))
```




```
model.glmn$bestTune
```

```
##    alpha     lambda
## 33   0.2 0.01483856
```

LDA

```
set.seed(1)
model.lda = train(x = wine[rowtrain, 1:11],
                  y = wine$quality[rowtrain],
                  method = "lda",
                  metric = "ROC",
                  trControl = ctrl)
```

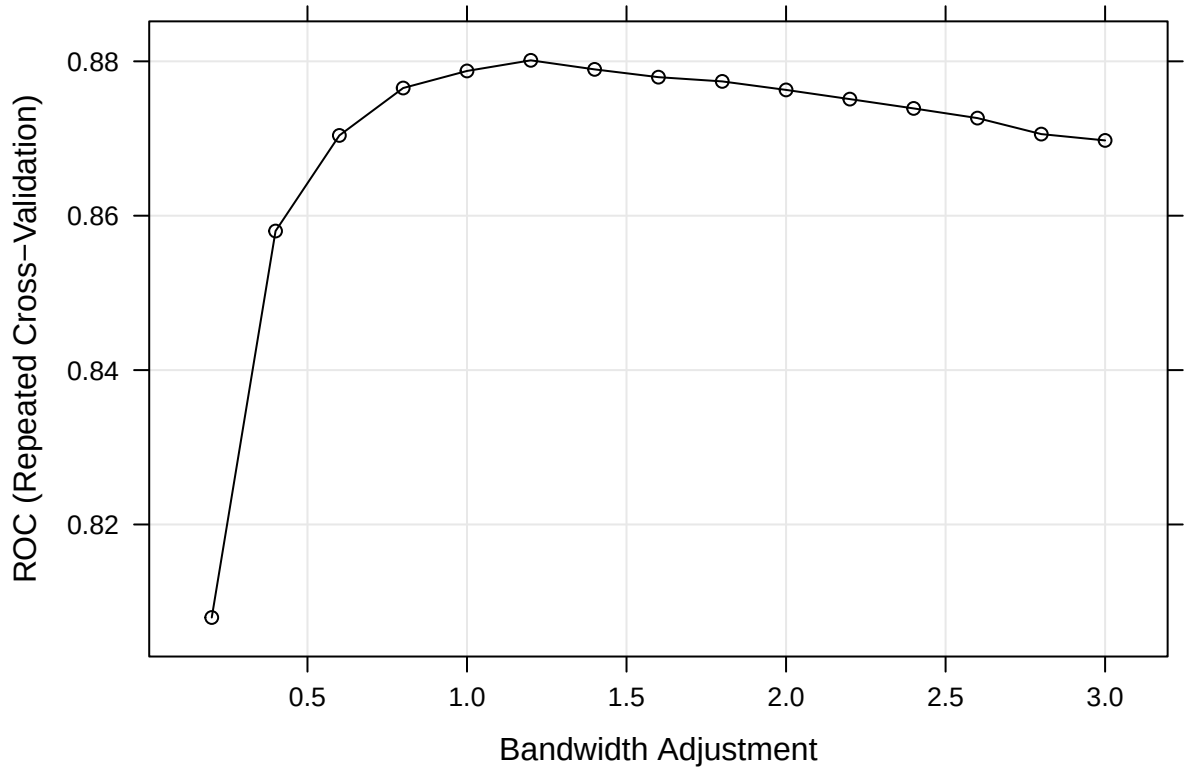
QDA

```
set.seed(1)
model.qda = train(x = wine[rowtrain, 1:11],
                  y = wine$quality[rowtrain],
                  method = "qda",
                  metric = "ROC",
                  trControl = ctrl)
```

Naive Bayes

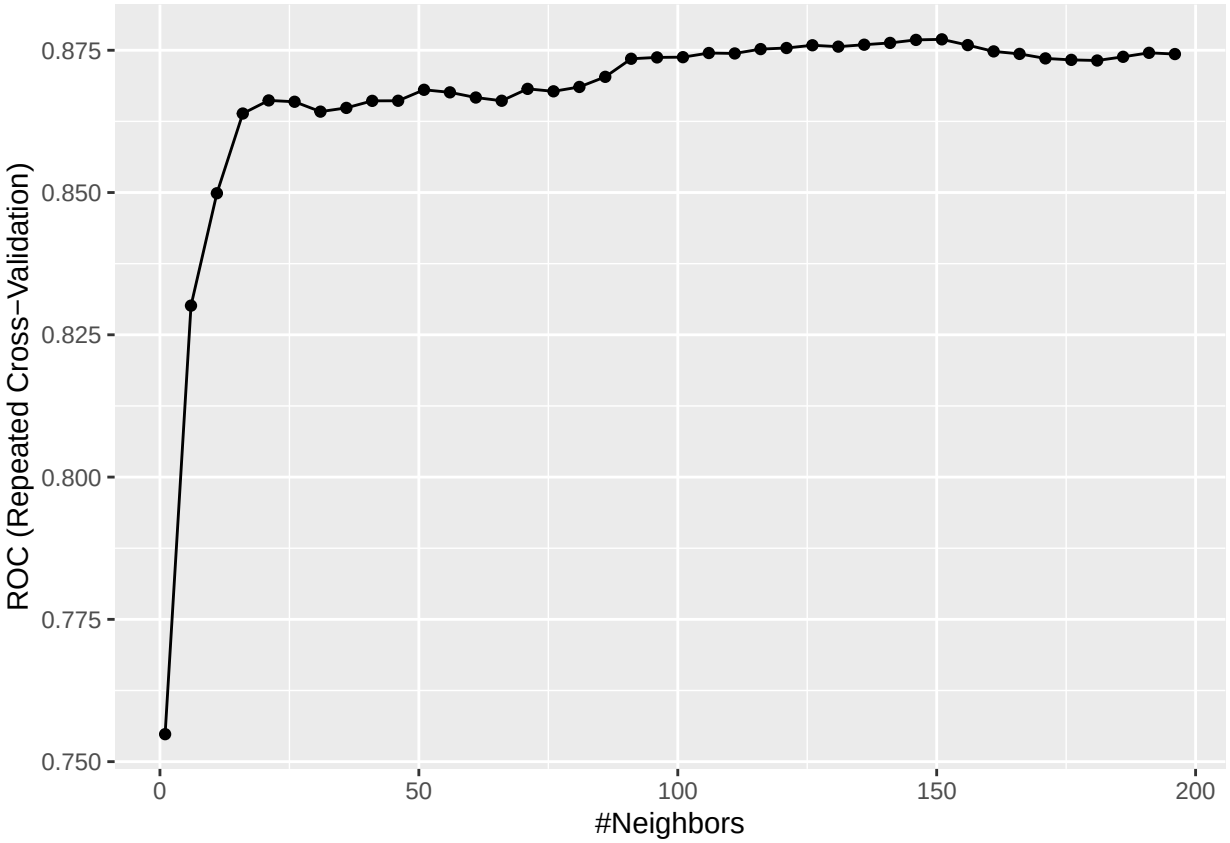
```
nbgrid = expand.grid(usekernel = TRUE,
                     fL = 1,
                     adjust = seq(.2, 3, by = .2))
set.seed(1)
model.nb = train(x = wine[rowtrain, 1:11],
                 y = wine$quality[rowtrain],
                 method = "nb",
                 tuneGrid = nbgrid,
                 metric = "ROC",
                 trControl = ctrl)

plot(model.nb)
```



KNN

```
set.seed(1)
model.knn = train(x = wine[rowtrain, 1:11],
                  y = wine$quality[rowtrain],
                  method = "knn",
                  preProcess = c("center", "scale"),
                  tuneGrid = data.frame(k = seq(1, 200, by = 5)),
                  trControl = ctrl)
ggplot(model.knn)
```



summary

```
res = resamples(list(GLM = model.glm,
                     GLMNET = model.glmn,
                     LDA = model.lda,
                     DQA = model.qda,
                     NB = model.nb,
                     KNN = model.knn))
summary(res)
```

```
##
## Call:
## summary.resamples(object = res)
##
## Models: GLM, GLMNET, LDA, DQA, NB, KNN
## Number of resamples: 50
##
## ROC
##              Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## GLM    0.8172043 0.8604196 0.8916149 0.8881226 0.9118377 0.9818841    0
## GLMNET 0.8097826 0.8667660 0.8920549 0.8903277 0.9141281 0.9811594    0
## LDA    0.8144410 0.8581134 0.8873706 0.8863589 0.9111542 0.9862319    0
## DQA    0.7305901 0.8323370 0.8563665 0.8523540 0.8850932 0.9536232    0
```

```
## NB     0.7717391 0.8590454 0.8870342 0.8801251 0.9069746 0.9637681    0
## KNN    0.7491039 0.8483890 0.8864001 0.8769163 0.9042831 0.9844203    0
##
## Sens
##             Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## GLM    0.1428571 0.2857143 0.3571429 0.3902857 0.4666667 0.7333333    0
## GLMNET 0.1333333 0.2666667 0.3333333 0.3380000 0.4285714 0.6000000    0
## LDA    0.2000000 0.3571429 0.4666667 0.4708571 0.5714286 0.8000000    0
## DQA    0.2857143 0.5785714 0.6666667 0.6506667 0.7333333 0.8666667    0
## NB     0.2857143 0.5714286 0.6547619 0.6561905 0.7333333 0.9333333    0
## KNN    0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000    0
##
## Spec
##             Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## GLM    0.9130435 0.9456522 0.9567555 0.9583474 0.9677419 1.0000000    0
## GLMNET 0.9347826 0.9566386 0.9730014 0.9711571 0.9784362 1.0000000    0
## LDA    0.9021739 0.9241176 0.9456522 0.9444670 0.9565217 0.9891304    0
## DQA    0.8043478 0.8478261 0.8648901 0.8674778 0.8921809 0.9239130    0
## NB     0.8260870 0.8586957 0.8804348 0.8802641 0.8997487 0.9456522    0
## KNN    1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000    0
```

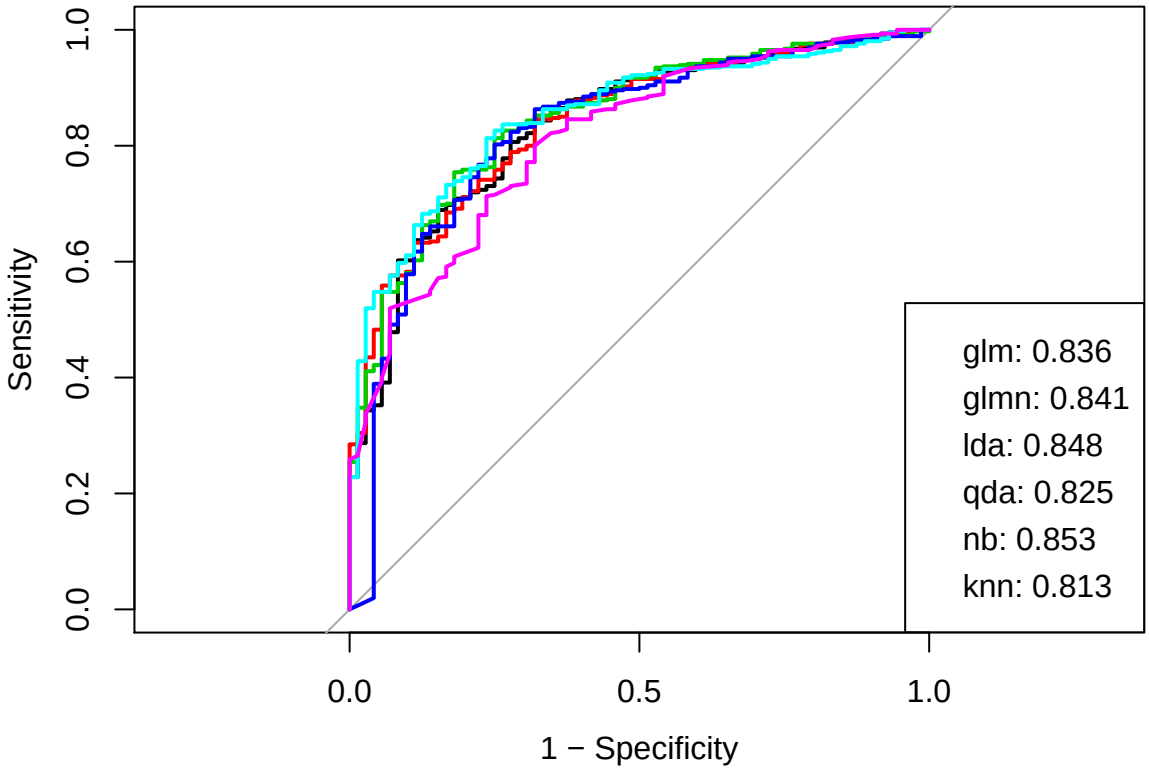
test set performance

```
glm.pred = predict(model.glm, newdata = wine[-rowtrain,], type = "prob")[,2]
glmn.pred = predict(model.glmn, newdata = wine[-rowtrain,], type = "prob")[,2]
lda.pred = predict(model.lda, newdata = wine[-rowtrain,], type = "prob")[,2]
qda.pred = predict(model.qda, newdata = wine[-rowtrain,], type = "prob")[,2]
nb.pred = predict(model.nb, newdata = wine[-rowtrain,], type = "prob")[,2]
knn.pred = predict(model.knn, newdata = wine[-rowtrain,], type = "prob")[,2]

roc.glm = roc(wine$quality[-rowtrain], glm.pred)
roc.glmn = roc(wine$quality[-rowtrain], glmn.pred)
roc.lda = roc(wine$quality[-rowtrain], lda.pred)
roc.qda = roc(wine$quality[-rowtrain], qda.pred)
roc.nb = roc(wine$quality[-rowtrain], nb.pred)
roc.knn = roc(wine$quality[-rowtrain], knn.pred)

auc = c(roc.glm$auc[1], roc.glmn$auc[1], roc.lda$auc[1], roc.qda$auc[1], roc.nb$auc[1], roc.knn$auc[1])

plot(roc.glm, legacy.axes = TRUE)
plot(roc.glmn, col = 2, add = TRUE)
plot(roc.lda, col = 3, add = TRUE)
plot(roc.qda, col = 4, add = TRUE)
plot(roc.nb, col = 5, add = TRUE)
plot(roc.knn, col = 6, add = TRUE)
modelNames = c("glm", "glmn", "lda", "qda", "nb", "knn")
legend("bottomright", legend = paste0(modelNames, ": ", round(auc, 3)))
```




```
glm=1-sum(predict(model.glm, newdata = wine[-rowtrain,]) ==wine[-rowtrain,12])/nrow(wine[-rowtrain,])
knn=1-sum(predict(model.knn, newdata = wine[-rowtrain,])==wine[-rowtrain,12])/nrow(wine[-rowtrain,])
glmn=1-sum(predict(model.glmn, newdata = wine[-rowtrain,])==wine[-rowtrain,12])/nrow(wine[-rowtrain,])
lda =1-sum(predict(model.lda, newdata = wine[-rowtrain,])==wine[-rowtrain,12])/nrow(wine[-rowtrain,])
qda =1-sum(predict(model.qda, newdata = wine[-rowtrain,])==wine[-rowtrain,12])/nrow(wine[-rowtrain,])
nb = 1-sum(predict(model.nb, newdata = wine[-rowtrain,])==wine[-rowtrain,12])/nrow(wine[-rowtrain,])

# test data misclassification rate
T1=data.frame(model=c("GLM","GLMN","LDA","QDA","NB","KNN"),Error_rate=c(glm,glmn,lda,qda,nb,knn)) %>% kn
T1
```

model	Error_rate
GLM	0.133
GLMN	0.128
LDA	0.133
QDA	0.173
NB	0.169
KNN	0.135

```
# train data misclassification rate
glm1=1-sum(predict(model.glm, newdata = wine[rowtrain,]) ==wine[rowtrain,12])/nrow(wine[rowtrain,])
knn1=1-sum(predict(model.knn, newdata = wine[rowtrain,])==wine[rowtrain,12])/nrow(wine[rowtrain,])
glmn1=1-sum(predict(model.glmn, newdata = wine[rowtrain,])==wine[rowtrain,12])/nrow(wine[rowtrain,])
lda1 =1-sum(predict(model.lda, newdata = wine[rowtrain,])==wine[rowtrain,12])/nrow(wine[rowtrain,])
```

```
qda1 =1-sum(predict(model.qda, newdata = wine[rowtrain,])==wine[rowtrain,12])/nrow(wine[rowtrain,])
nb1 =1-sum(predict(model.nb, newdata = wine[rowtrain,])==wine[rowtrain,12])/nrow(wine[rowtrain,])

T2=data.frame(model=c("GLM","GLMN","LDA","QDA","NB","KNN"),Error_rate=c(glm1,glmn1,lda1,qda1,nb1,knn1))
T2
```

model	Error_rate
GLM	0.110
GLMN	0.108
LDA	0.117
QDA	0.142
NB	0.127
KNN	0.136