

Untitled

Yue Lai

5/15/2020

import data

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

```
## Warning in FUN(X[[i]], ...): strings not representable in native encoding will  
## be translated to UTF-8
```

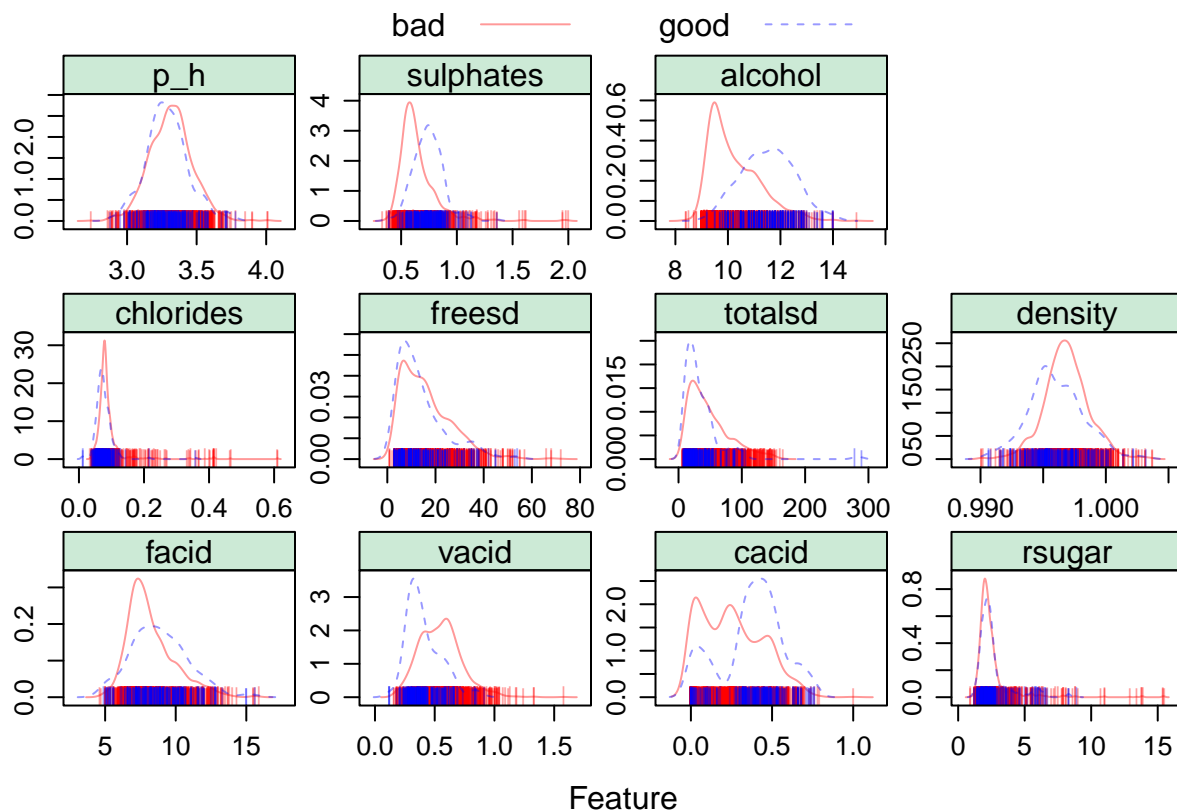
```
white = read_excel(path = "./data/wine.xlsx", sheet = "white") %>%  
  janitor::clean_names()
```

```
wine = data.frame(rbind(red, white))
```

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

The data contains 6497 observations and 11 variables. The outcome is the binary variable category. We start from some simple visualization of the data.

```
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(666)
model.glm = train(x = wine[rowtrain, 1:11],
                  y = wine$quality[rowtrain],
                  method = "glm",
                  metric = "ROC",
                  trControl = ctrl)
```

```
## Warning: The `i` argument of `[`() can't be a matrix as of tibble 3.0.0.
## Convert to a vector.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.
```


[illegible]

glmn

```
set.seed(666)
glmngrid = expand.grid(.alpha = seq(0,1,length = 6),
```


[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

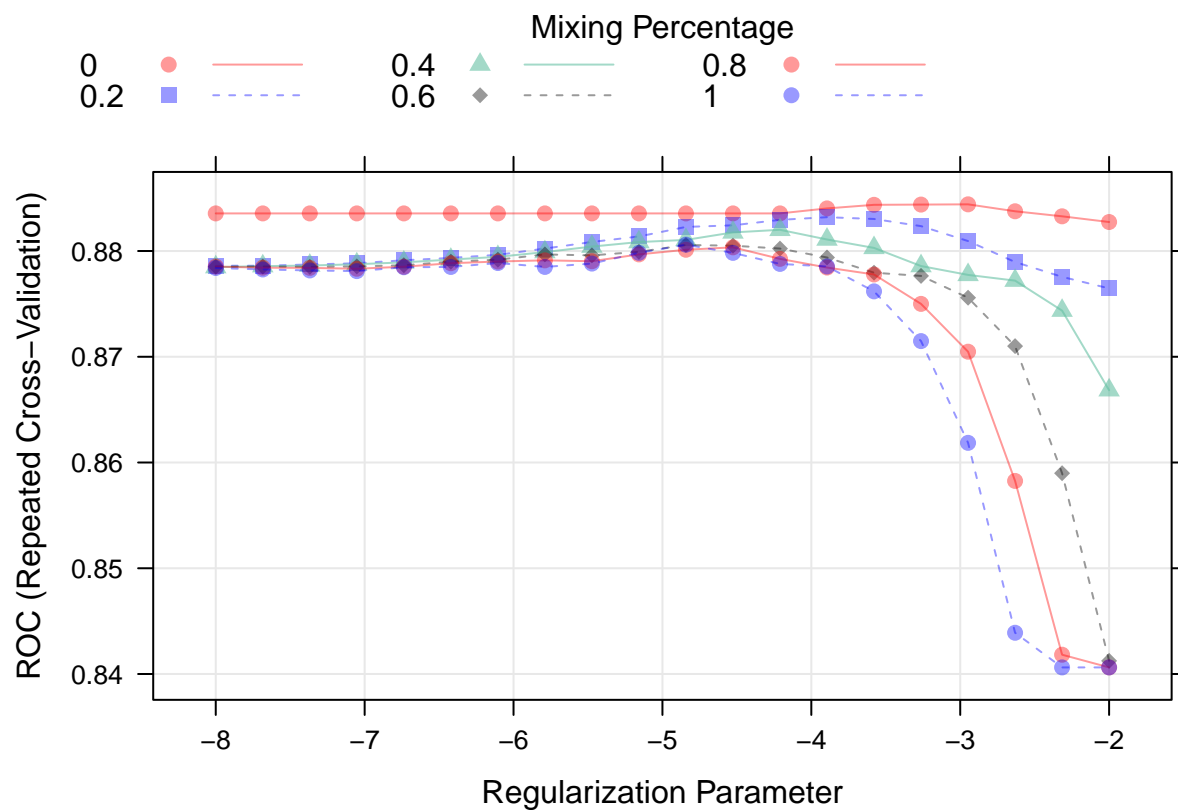
[illegible]

[illegible]

[illegible]

```
## Warning: Setting row names on a tibble is deprecated.
## Warning: Setting row names on a tibble is deprecated.
## Warning: Setting row names on a tibble is deprecated.
## Warning: Setting row names on a tibble is deprecated.
## Warning: Setting row names on a tibble is deprecated.
## Warning: Setting row names on a tibble is deprecated.
## Warning: Setting row names on a tibble is deprecated.
## Warning: Setting row names on a tibble is deprecated.
## Warning: Setting row names on a tibble is deprecated.
```

```
plot(model.glmn, xTrans = function(x) log(x))
```



```
model.glmn$bestTune
```

```
##      alpha      lambda
## 17      0 0.05247762
```


LDA

```
set.seed(666)
model.lda = train(x = wine[rowtrain, 1:11],
                  y = wine$quality[rowtrain],
                  method = "lda",
                  metric = "ROC",
                  trControl = ctrl)
```

```
## Warning: Setting row names on a tibble is deprecated.
```

```
## Warning: Setting row names on a tibble is deprecated.
```

```
## Warning: Setting row names on a tibble is deprecated.
```

```
## Warning: Setting row names on a tibble is deprecated.
```

```
## Warning: Setting row names on a tibble is deprecated.
```

```
## Warning: Setting row names on a tibble is deprecated.
```

```
## Warning: Setting row names on a tibble is deprecated.
```

```
## Warning: Setting row names on a tibble is deprecated.
```

```
## Warning: Setting row names on a tibble is deprecated.
```

```
## Warning: Setting row names on a tibble is deprecated.
```

```
## Warning: Setting row names on a tibble is deprecated.
```

```
## Warning: Setting row names on a tibble is deprecated.
```

```
## Warning: Setting row names on a tibble is deprecated.
```

```
## Warning: Setting row names on a tibble is deprecated.
```

```
## Warning: Setting row names on a tibble is deprecated.
```

```
## Warning: Setting row names on a tibble is deprecated.
```

```
## Warning: Setting row names on a tibble is deprecated.
```

```
## Warning: Setting row names on a tibble is deprecated.
```

```
## Warning: Setting row names on a tibble is deprecated.
```

```
## Warning: Setting row names on a tibble is deprecated.
```

```
## Warning: Setting row names on a tibble is deprecated.
```

```
## Warning: Setting row names on a tibble is deprecated.
```



```
## Warning: Setting row names on a tibble is deprecated.
```

```
## Warning: Setting row names on a tibble is deprecated.
```

QDA

```
set.seed(666)
model.qda = train(x = wine[rowtrain, 1:11],
                  y = wine$quality[rowtrain],
                  method = "qda",
                  metric = "ROC",
                  trControl = ctrl)
```

```
## Warning: Setting row names on a tibble is deprecated.
```

```
## Warning: Setting row names on a tibble is deprecated.
```

```
## Warning: Setting row names on a tibble is deprecated.
```

```
## Warning: Setting row names on a tibble is deprecated.
```

```
## Warning: Setting row names on a tibble is deprecated.
```

```
## Warning: Setting row names on a tibble is deprecated.
```

```
## Warning: Setting row names on a tibble is deprecated.
```

```
## Warning: Setting row names on a tibble is deprecated.
```

```
## Warning: Setting row names on a tibble is deprecated.
```

```
## Warning: Setting row names on a tibble is deprecated.
```

```
## Warning: Setting row names on a tibble is deprecated.
```

```
## Warning: Setting row names on a tibble is deprecated.
```

```
## Warning: Setting row names on a tibble is deprecated.
```

```
## Warning: Setting row names on a tibble is deprecated.
```

```
## Warning: Setting row names on a tibble is deprecated.
```

```
## Warning: Setting row names on a tibble is deprecated.
```

```
## Warning: Setting row names on a tibble is deprecated.
```

```
## Warning: Setting row names on a tibble is deprecated.
```

```
## Warning: Setting row names on a tibble is deprecated.
```

[illegible]

```
## Warning: Setting row names on a tibble is deprecated.
## Warning: Setting row names on a tibble is deprecated.
## Warning: Setting row names on a tibble is deprecated.
## Warning: Setting row names on a tibble is deprecated.
## Warning: Setting row names on a tibble is deprecated.
```

Naive Bayes

KNN

```
set.seed(666)
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)
```

```
## Warning in train.default(x = wine[rowtrain, 1:11], y = wine$quality[rowtrain], :
## The metric "Accuracy" was not in the result set. ROC will be used instead.
```

```
## Warning: Setting row names on a tibble is deprecated.
## Warning: Setting row names on a tibble is deprecated.
## Warning: Setting row names on a tibble is deprecated.
## Warning: Setting row names on a tibble is deprecated.
## Warning: Setting row names on a tibble is deprecated.
## Warning: Setting row names on a tibble is deprecated.
## Warning: Setting row names on a tibble is deprecated.
## Warning: Setting row names on a tibble is deprecated.
## Warning: Setting row names on a tibble is deprecated.
## Warning: Setting row names on a tibble is deprecated.
## Warning: Setting row names on a tibble is deprecated.
```

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

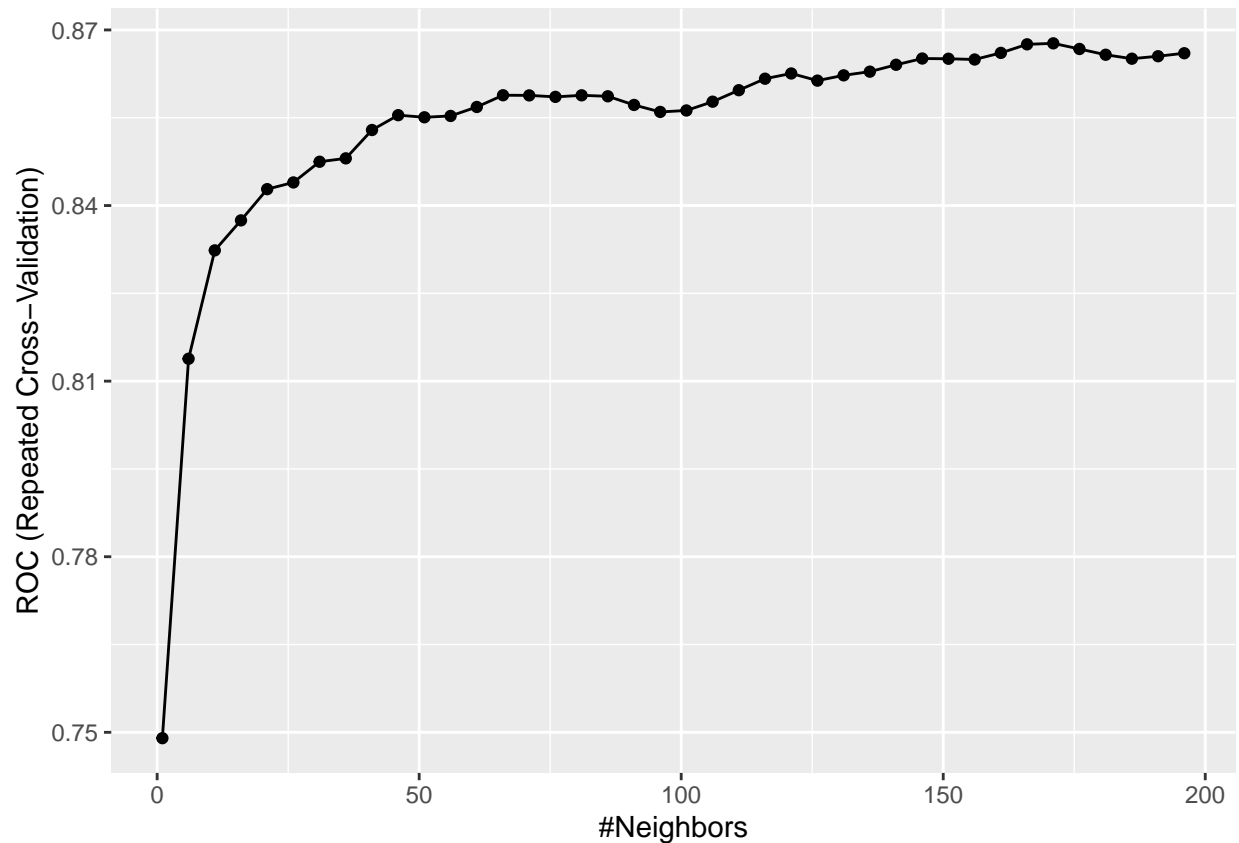
[illegible]

[illegible]

[illegible]

```
## Warning: Setting row names on a tibble is deprecated.  
## Warning: Setting row names on a tibble is deprecated.  
## Warning: Setting row names on a tibble is deprecated.  
## Warning: Setting row names on a tibble is deprecated.  
## Warning: Setting row names on a tibble is deprecated.  
## Warning: Setting row names on a tibble is deprecated.  
## Warning: Setting row names on a tibble is deprecated.  
## Warning: Setting row names on a tibble is deprecated.  
## Warning: Setting row names on a tibble is deprecated.  
## Warning: Setting row names on a tibble is deprecated.  
## Warning: Setting row names on a tibble is deprecated.  
## Warning: Setting row names on a tibble is deprecated.  
## Warning: Setting row names on a tibble is deprecated.  
## Warning: Setting row names on a tibble is deprecated.  
## Warning: Setting row names on a tibble is deprecated.  
## Warning: Setting row names on a tibble is deprecated.  
## Warning: Setting row names on a tibble is deprecated.  
## Warning: Setting row names on a tibble is deprecated.  
## Warning: Setting row names on a tibble is deprecated.
```

```
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, KNN
## Number of resamples: 50
##
## ROC
```

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
## GLM	0.7775362	0.8446217	0.8740942	0.8779923	0.9168608	0.9829193	0
## GLMNET	0.8028986	0.8493221	0.8834627	0.8844142	0.9183931	0.9767081	0
## LDA	0.7884058	0.8485866	0.8815994	0.8839543	0.9170290	0.9836957	0
## DQA	0.7521739	0.8164855	0.8509834	0.8537102	0.8938923	0.9518634	0


```
## KNN      0.7734255 0.8393778 0.8716846 0.8677096 0.9028533 0.9491460      0
##
## Sens
##           Min.    1st Qu.    Median      Mean    3rd Qu.      Max. NA's
## GLM      0.9239130 0.9565217 0.9673913 0.9648527 0.9782609 1.0000000      0
## GLMNET   0.9569892 0.9891304 0.9946237 0.9913230 1.0000000 1.0000000      0
## LDA      0.9021739 0.9380259 0.9565217 0.9522768 0.9673913 0.9784946      0
## DQA      0.8172043 0.8807562 0.9189458 0.9121388 0.9347826 0.9784946      0
## KNN      1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000      0
##
## Spec
##           Min.    1st Qu.    Median      Mean    3rd Qu.      Max. NA's
## GLM      0.1428571 0.2666667 0.3571429 0.3725714 0.4666667 0.7857143      0
## GLMNET   0.0000000 0.1333333 0.2000000 0.1910476 0.2666667 0.5000000      0
## LDA      0.1428571 0.3571429 0.4285714 0.4362857 0.5333333 0.7857143      0
## DQA      0.1428571 0.3571429 0.4666667 0.4527619 0.5333333 0.7142857      0
## KNN      0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000      0
```

test set performance

```
glm.pred = predict(model.glm, newdata = wine[-rowtrain,], type = "prob")[,2]
glmnet.pred = predict(model.glmnet, 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)
```

```
## Setting levels: control = bad, case = good
```

```
## Setting direction: controls < cases
```

```
roc.glmnet = roc(wine$quality[-rowtrain], glmnet.pred)
```

```
## Setting levels: control = bad, case = good
```

```
## Setting direction: controls < cases
```

```
roc.lda = roc(wine$quality[-rowtrain], lda.pred)
```

```
## Setting levels: control = bad, case = good
```

```
## Setting direction: controls < cases
```

```
roc.qda = roc(wine$quality[-rowtrain], qda.pred)
```

```
## Setting levels: control = bad, case = good
```

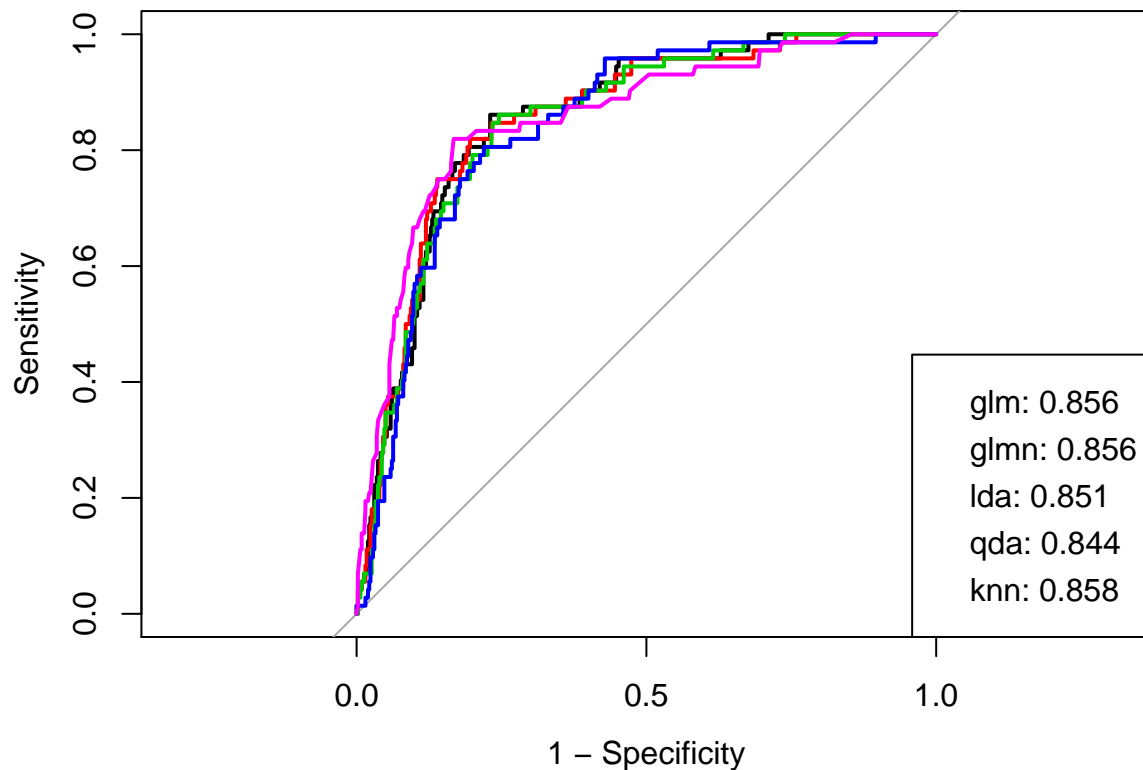
```
## Setting direction: controls < cases
```

```
#roc.nb = roc(wine$quality[-rowtrain], nb.pred)
roc.knn = roc(wine$quality[-rowtrain], knn.pred)
```

```
## Setting levels: control = bad, case = good
## Setting direction: controls < cases
```

```
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,]) %>% as.data.frame()==wine[-rowtrain,12])/nrow(wine)
knn=1-sum(predict(model.knn, newdata = wine[-rowtrain,]) %>% as.data.frame()==wine[-rowtrain,12])/nrow(wine)
glmn=1-sum(predict(model.glmn, newdata = wine[-rowtrain,]) %>% as.data.frame()==wine[-rowtrain,12])/nrow(wine)
```

```
lda =1-sum(predict(model.lda, newdata = wine[-rowtrain,]))%>% as.data.frame()==wine[-rowtrain,12])/nrow(wine)
qda =1-sum(predict(model.qda, newdata = wine[-rowtrain,]))%>% as.data.frame()==wine[-rowtrain,12])/nrow(wine)

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

model	Error_rate
GLM	0.1372180
GLMN	0.1372180
LDA	0.1372180
QDA	0.1503759
KNN	0.1353383

```
# train data misclassification rate
glm1=1-sum(predict(model.glm, newdata = wine[rowtrain,]))%>% as.data.frame()==wine[rowtrain,12])/nrow(wine)
knn1=1-sum(predict(model.knn, newdata = wine[rowtrain,]))%>% as.data.frame()==wine[rowtrain,12])/nrow(wine)
glm1=1-sum(predict(model.glm, newdata = wine[rowtrain,]))%>% as.data.frame()==wine[rowtrain,12])/nrow(wine)
glm1=1-sum(predict(model.glm, newdata = wine[rowtrain,]))%>% as.data.frame()==wine[rowtrain,12])/nrow(wine)
lda1 =1-sum(predict(model.lda, newdata = wine[rowtrain,]))%>% as.data.frame()==wine[rowtrain,12])/nrow(wine)
qda1 =1-sum(predict(model.qda, newdata = wine[rowtrain,]))%>% as.data.frame()==wine[rowtrain,12])/nrow(wine)

T2=data.frame(model=c("GLM", "GLMN", "LDA", "QDA", "KNN"),Error_rate=c(glm1,glm1,lda1,qda1,knn1)) %>% knitr::kable()
T2
```

model	Error_rate
GLM	0.1096532
GLMN	0.1134021
LDA	0.1143393
QDA	0.1190253
KNN	0.1358950