Midterm Project

Ekta Chaudhary

26/03/2020

```
library(tidyverse)
library(readxl)
library(caret)
library(ModelMetrics)
library(glmnet)
library(gam)
library(mgcv)
library(splines)
library(pdp)
library(earth)
library(dplyr)
library(naniar)
library(bnstruct)
library(corrplot)
library(logisticPCA)
library(MASS)
library(e1071)
library(mlbench)
library(pROC)
library(AppliedPredictiveModeling)
library(ggplot2)
```

Reading the Dataset. Adding the column names to the Breast_Cancer dataset.

```
Breast_Cancer =
  read_csv(file = './data/Breast_Cancer.csv' ,col_names = c('id_number','Clump_thickness','Uniformity_o
```

Replacing the missing observations that are denoted by a ? with na and then using KNN imputation to impute the missing values.

```
Breast_Cancer = Breast_Cancer %>% replace_with_na_all(condition = ~.x == '?')
```

```
Breast_Cancer <- knn.impute(as.matrix(Breast_Cancer), k = 10, cat.var = 2:ncol(Breast_Cancer) - 2,
  to.impute = 1:nrow(Breast_Cancer), using = 1:nrow(Breast_Cancer))
```

Creating a data frame called Breast_Cancer. The str method will allows us to know the data type of each variable. As we can see the data type of all the columns is numeric. The column Class_cancer takes two values: 2 for benign and 4 for malignant.

```
Breast_Cancer <- data.frame(Breast_Cancer)
dim(Breast_Cancer)
```

```
## [1] 699  11
```

```r
str(Breast_Cancer)
```

```
## 'data.frame':    699 obs. of  11 variables:
##  $ id_number                  : num  1000025 1002945 1015425 1016277 1017023 ...
##  $ Clump_thickness            : num  5 5 3 6 4 8 1 2 2 4 ...
##  $ Uniformity_of_Cell_Size : num  1 4 1 8 1 10 1 1 1 2 ...
##  $ Uniformity_of_Cell_Shape: num  1 4 1 8 1 10 1 2 1 1 ...
##  $ Marginal_Adhesion          : num  1 5 1 1 3 8 1 1 1 1 ...
##  $ Epithelial_Cell_Size     : num  2 7 2 3 2 7 2 2 2 2 ...
##  $ Bare_Nuclei                : num  1 10 2 4 1 10 10 1 1 1 ...
##  $ Bland_Chromatin            : num  3 3 3 3 3 9 3 3 1 2 ...
##  $ Normal_Nucleoli            : num  1 2 1 7 1 7 1 1 1 1 ...
##  $ Mitoses                    : num  1 1 1 1 1 1 1 1 5 1 ...
##  $ Class_cancer               : num  2 2 2 2 2 4 2 2 2 2 ...
```

Using ifelse on the column Class_cancer –> If Class_cancer = 4 then it's malignant, else its benign. Converted the Class_cancer into factors.

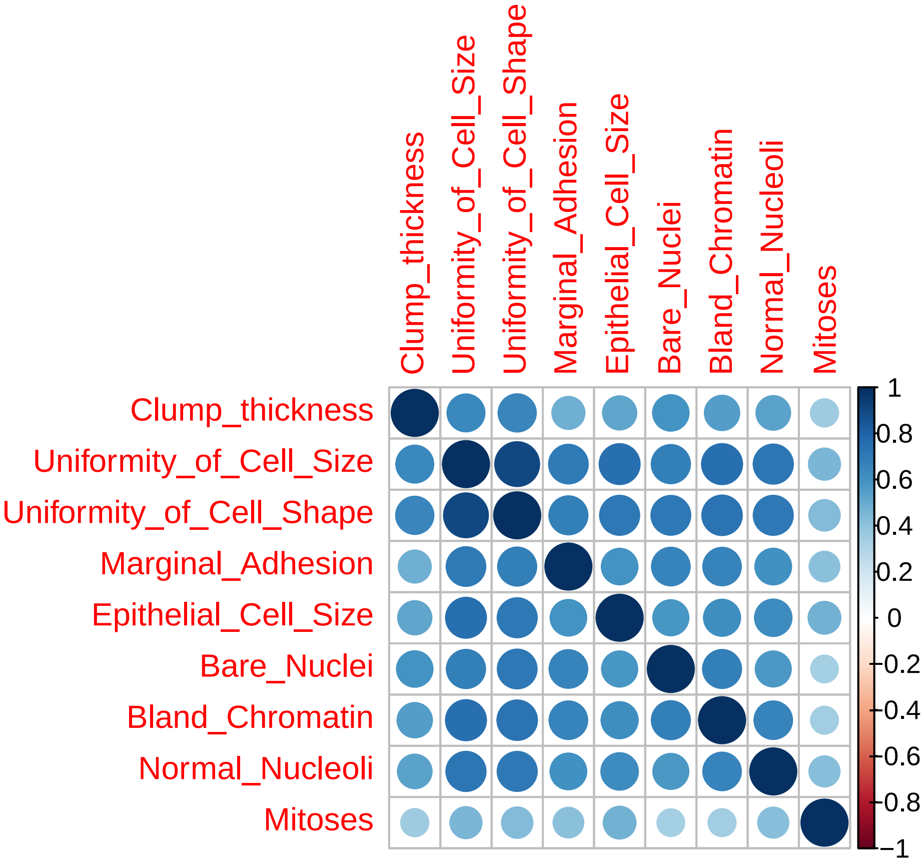
```r
Breast_Cancer$Class_cancer = as.factor(ifelse(Breast_Cancer$Class_cancer == 4, 'mal','benign'))

Breast_Cancer = Breast_Cancer[,2:11]
```

```r
str(Breast_Cancer)
```

```
## 'data.frame':    699 obs. of  10 variables:
##  $ Clump_thickness            : num  5 5 3 6 4 8 1 2 2 4 ...
##  $ Uniformity_of_Cell_Size : num  1 4 1 8 1 10 1 1 1 2 ...
##  $ Uniformity_of_Cell_Shape: num  1 4 1 8 1 10 1 2 1 1 ...
##  $ Marginal_Adhesion          : num  1 5 1 1 3 8 1 1 1 1 ...
##  $ Epithelial_Cell_Size     : num  2 7 2 3 2 7 2 2 2 2 ...
##  $ Bare_Nuclei                : num  1 10 2 4 1 10 10 1 1 1 ...
##  $ Bland_Chromatin            : num  3 3 3 3 3 9 3 3 1 2 ...
##  $ Normal_Nucleoli            : num  1 2 1 7 1 7 1 1 1 1 ...
##  $ Mitoses                    : num  1 1 1 1 1 1 1 1 5 1 ...
##  $ Class_cancer               : Factor w/ 2 levels "benign","mal": 1 1 1 1 1 2 1 1 1 1 ...
```

Creating a correlation plot to check the correlation between the variables.

```r
x = model.matrix(Class_cancer~., Breast_Cancer) [,-1]
corrplot(cor(x))
```



Here we are diving the data into training and test data.

```
set.seed(1)
rowTrain <- createDataPartition(y = Breast_Cancer$Class_cancer,
                                p = 2/3,
                                list = FALSE)
```

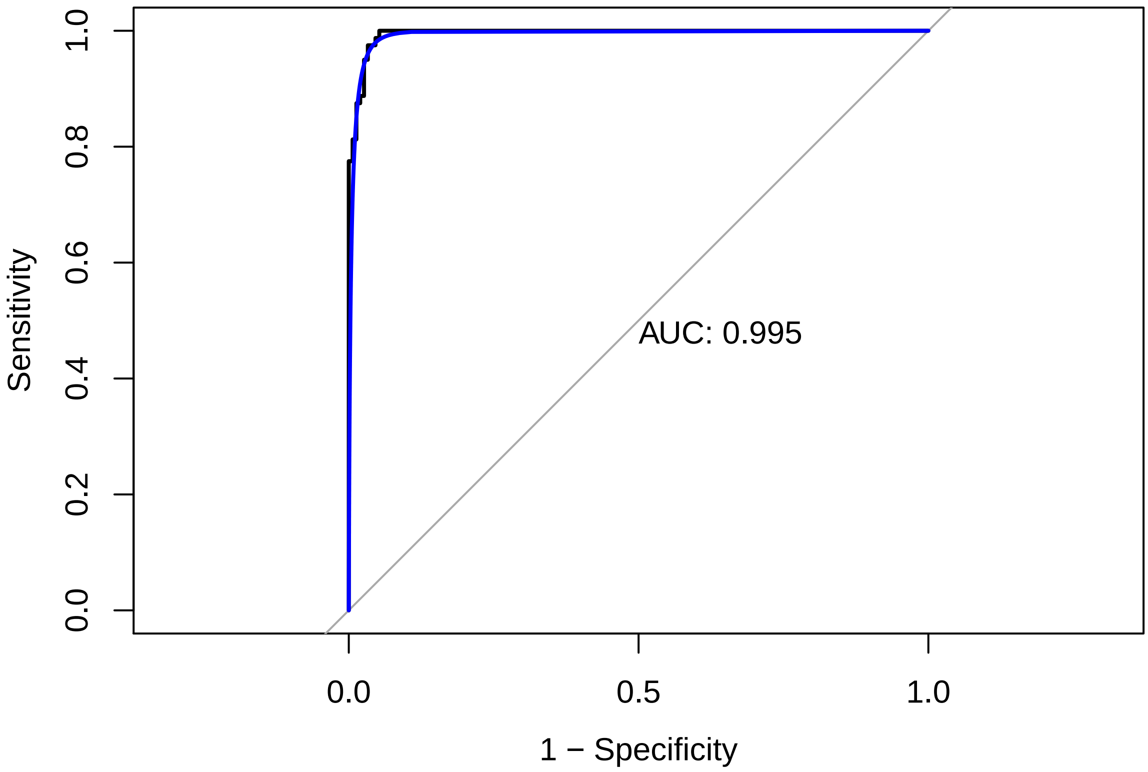
Logistic Regression using GLM:

```
glm.fit <- glm(Class_cancer~.,
               data = Breast_Cancer,
               subset = rowTrain,
               family = binomial)

contrasts(Breast_Cancer$Class_cancer)
```

```
##        mal
## benign   0
## mal      1
```

Evaluating the performance on the test data and then plotting the test ROC curve.

```
test.pred.prob  <- predict(glm.fit, newdata = Breast_Cancer[-rowTrain,1:9],
                           type = "response")
test.pred <- rep("benign", length(test.pred.prob))
test.pred[test.pred.prob > 0.5] <- "mal"

roc.glm <- roc(Breast_Cancer$Class_cancer[-rowTrain], test.pred.prob)
plot(roc.glm, legacy.axes = TRUE, print.auc = TRUE)
plot(smooth(roc.glm), col = 4, add = TRUE)
```



We can also fit a logistic regression using caret. This is to get the best performance using cross-valiation.

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

```
set.seed(1)
model.glm <- train(x = Breast_Cancer[rowTrain,1:9],
                  y = Breast_Cancer$Class_cancer[rowTrain],
                   method = "glm",
                   metric = "ROC",
                   trControl = ctrl)
```

```
coef(glm.fit)
```

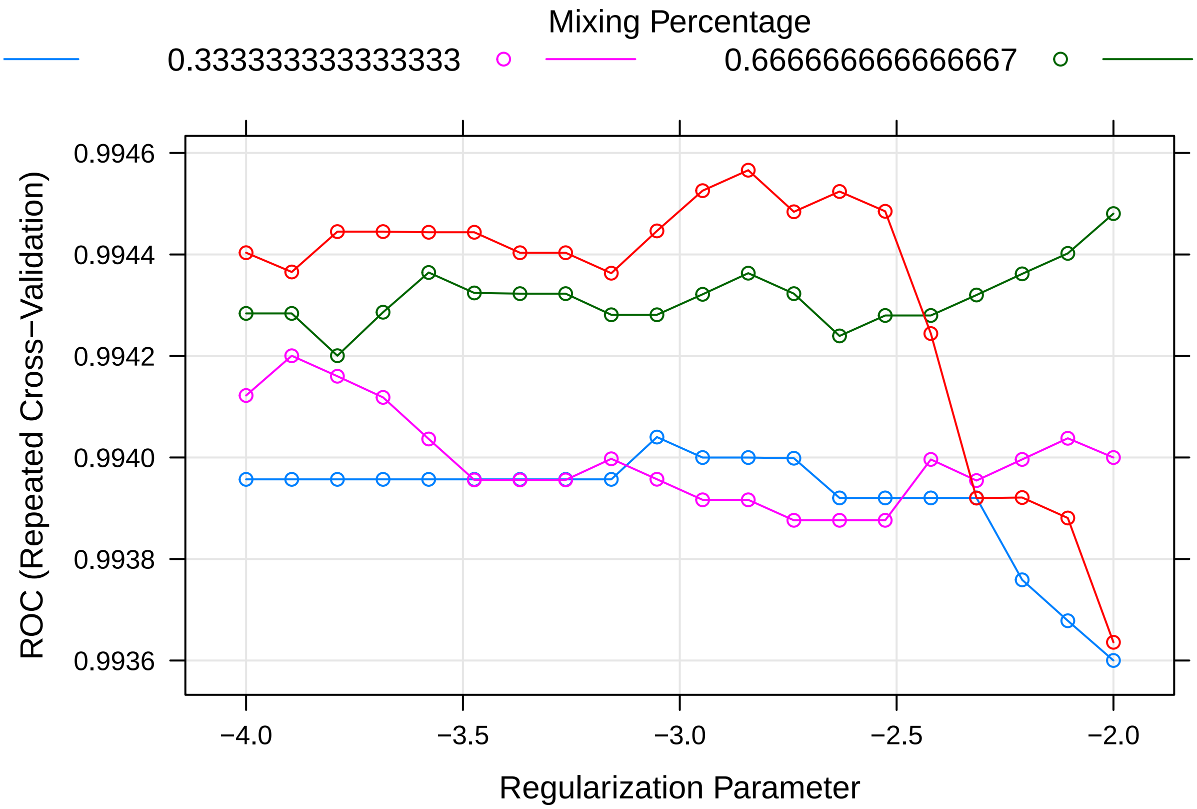
```
##             (Intercept)         Clump_thickness  Uniformity_of_Cell_Size
##              -9.0107637               0.4221055                0.1457212
## Uniformity_of_Cell_Shape       Marginal_Adhesion     Epithelial_Cell_Size
##               0.6370116               0.1479078               -0.2071035
##             Bare_Nuclei          Bland_Chromatin          Normal_Nucleoli
##                0.5172205                0.2023960                0.1863717
##                  Mitoses
##                0.6735681
```

```
summary(glm.fit)$coefficients
```

```
##                             Estimate Std. Error    z value     Pr(>|z|)
## (Intercept)               -9.0107637  1.2121934 -7.4334372 1.058115e-13
## Clump_thickness            0.4221055  0.1655169  2.5502258 1.076532e-02
## Uniformity_of_Cell_Size    0.1457212  0.2364887  0.6161865 5.377715e-01
## Uniformity_of_Cell_Shape   0.6370116  0.2746305  2.3195228 2.036671e-02
## Marginal_Adhesion          0.1479078  0.1437487  1.0289337 3.035108e-01
## Epithelial_Cell_Size      -0.2071035  0.2176655 -0.9514763 3.413627e-01
## Bare_Nuclei                0.5172205  0.1221590  4.2339955 2.295755e-05
## Bland_Chromatin            0.2023960  0.2142662  0.9446009 3.448627e-01
## Normal_Nucleoli            0.1863717  0.1252358  1.4881661 1.367071e-01
## Mitoses                    0.6735681  0.3364145  2.0021970 4.526355e-02
```

Regularized logistic regression can be fitted using glmnet'. We use thetrain‘ function to select the optimal tuning parameters.

```
glmnGrid <- expand.grid(.alpha = seq(0, 1, length = 4),
                        .lambda = exp(seq(-4, -2, length = 20)))


set.seed(1)
model.glmn <- train(x = Breast_Cancer[rowTrain,1:9],
                    y = Breast_Cancer$Class_cancer[rowTrain],
                    method = "glmnet",
                    tuneGrid = glmnGrid,
                    metric = "ROC",
                    trControl = ctrl)

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



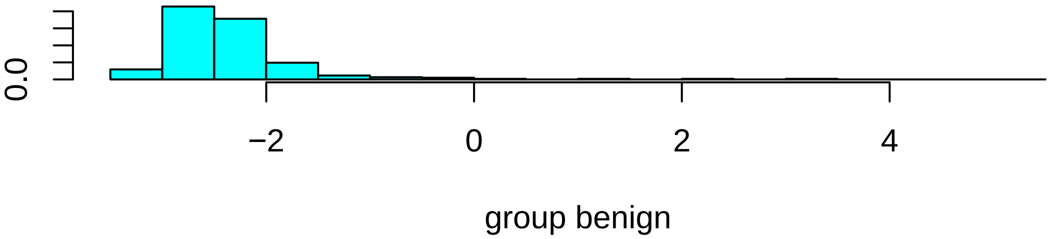
```
model.glmn$bestTune
```

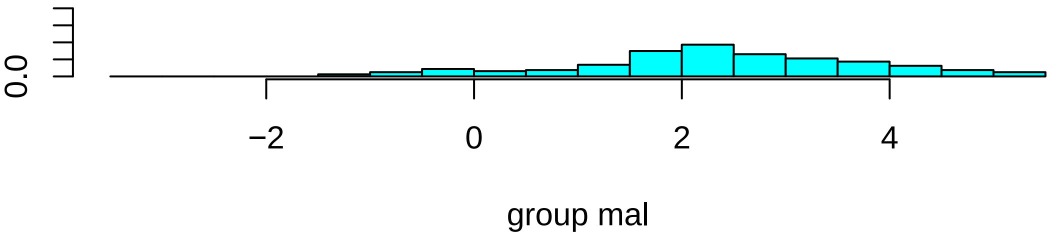
```
##    alpha     lambda
## 72     1 0.05830279
```

```
test.pred.prob  <- predict(model.glmn, x = Breast_Cancer[-rowTrain,1:9],
                      method = "glmnet")
```

Here we use the function `lda` in library `MASS` to conduct LDA.

```
library(MASS)

lda.fit <- lda(Class_cancer~., data = Breast_Cancer,
               subset = rowTrain)
plot(lda.fit)
```



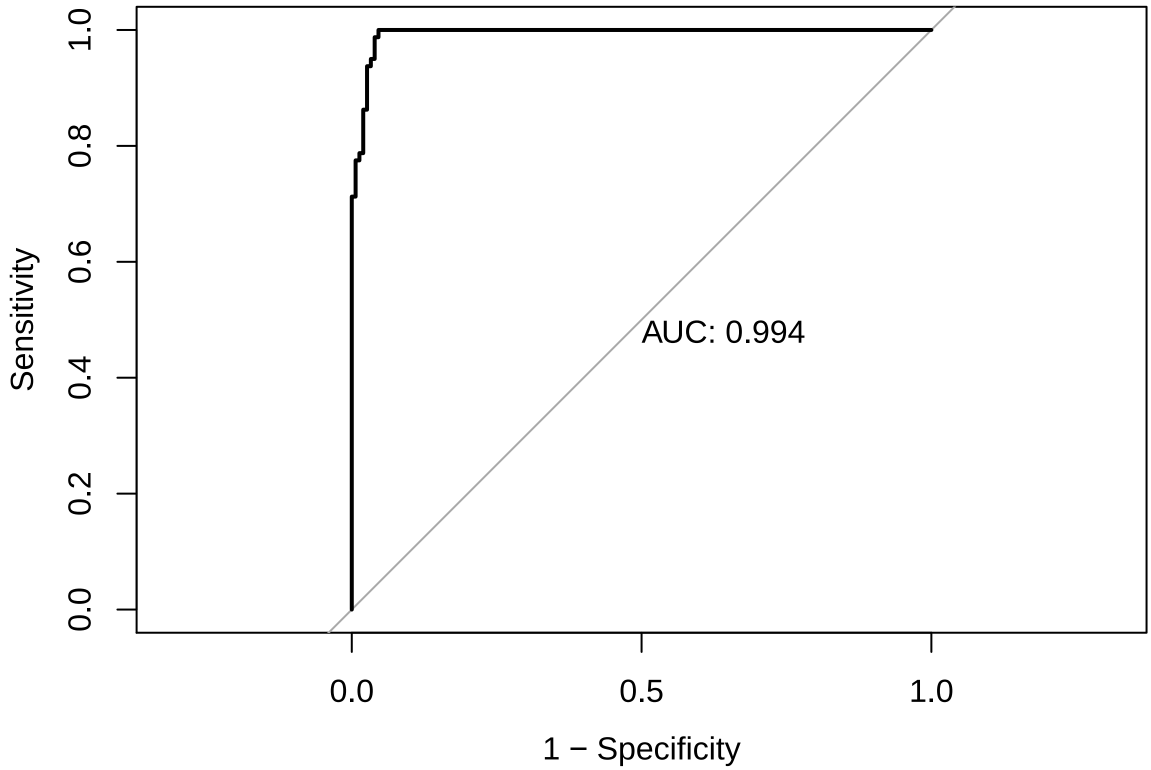


Here we are evaluating the test set performance using ROC.

```
lda.pred <- predict(lda.fit, newdata = Breast_Cancer[-rowTrain,])
head(lda.pred$posterior)
```

```
##       benign          mal
## 3 0.99998228 1.772454e-05
## 4 0.00875467 9.912453e-01
## 5 0.99998862 1.137890e-05
## 7 0.81869551 1.813045e-01
## 8 0.99999630 3.695724e-06
## 9 0.99999902 9.816974e-07
```

```
roc.lda <- roc(Breast_Cancer$Class_cancer[-rowTrain], lda.pred$posterior[,2],
               levels = c("benign", "mal"))

plot(roc.lda, legacy.axes = TRUE, print.auc = TRUE)
```

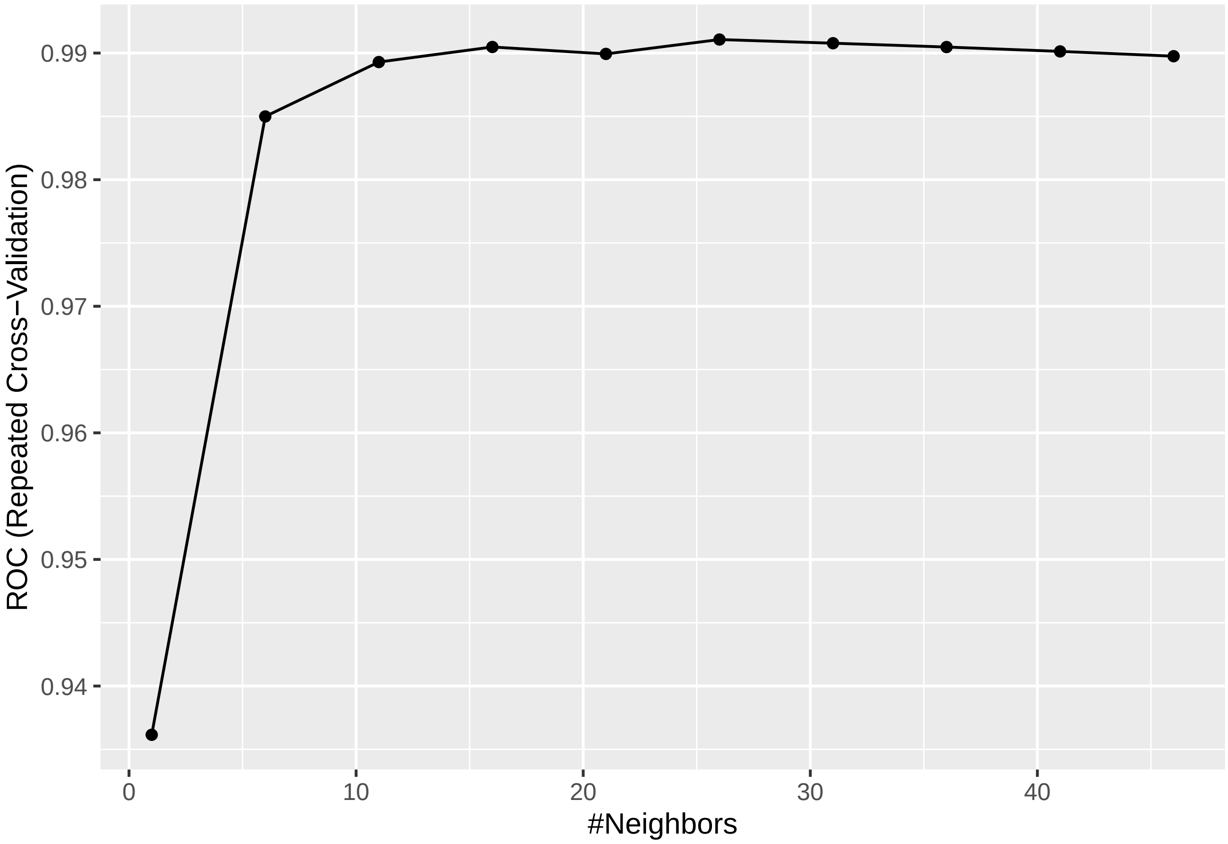


Using caret:

```
set.seed(1)
model.lda <- train(x = Breast_Cancer[rowTrain,1:9],
                   y = Breast_Cancer$Class_cancer[rowTrain],
                   method = "lda",
                   metric = "ROC",
                   trControl = ctrl)
```

Using KNN:

```
set.seed(1)
model.knn <- train(x = Breast_Cancer[rowTrain,1:9],
                   y = Breast_Cancer$Class_cancer[rowTrain],
                   method = "knn",
                   preProcess = c("center","scale"),
                   tuneGrid = data.frame(k = seq(1,50,by = 5)),
                   trControl = ctrl)

ggplot(model.knn)
```



# GLMNET and LDA have relatively good performance.

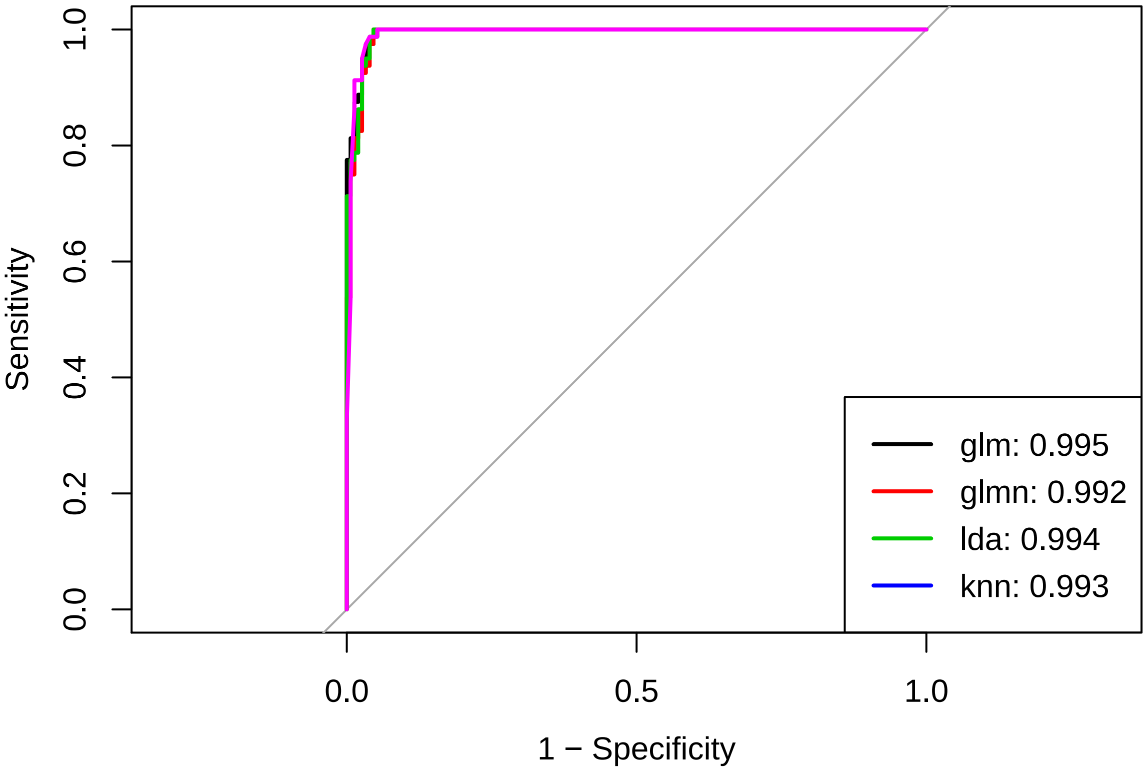
```
res <- resamples(list(GLM = model.glm, GLMNET = model.glmn,
                      LDA = model.lda, KNN = model.knn))
summary(res)
```

```
##
## Call:
## summary.resamples(object = res)
##
## Models: GLM, GLMNET, LDA, KNN
## Number of resamples: 50
##
## ROC
##             Min.   1st Qu.    Median      Mean 3rd Qu. Max. NA's
## GLM    0.9556452 0.9883233 0.9979503 0.9935506       1    1    0
## GLMNET 0.9576613 0.9917339 0.9979839 0.9945660       1    1    0
## LDA    0.9556452 0.9905637 0.9979839 0.9941979       1    1    0
## KNN    0.9506048 0.9881048 0.9964767 0.9910701       1    1    0
##
## Sens
##             Min.   1st Qu.   Median      Mean 3rd Qu. Max. NA's
## GLM    0.9333333 0.9666667 0.983871 0.9777419       1    1    0
## GLMNET 0.9354839 0.9677419 1.000000 0.9875699       1    1    0
## LDA    0.9333333 0.9677419 1.000000 0.9855914       1    1    0
## KNN    0.9333333 0.9666667 0.983871 0.9803656       1    1    0
##
```

```
## Spec
##           Min.   1st Qu.    Median      Mean   3rd Qu. Max. NA's
## GLM    0.8125 0.8750000 0.9375000 0.9363235 1.0000000    1    0
## GLMNET 0.6875 0.8235294 0.8750000 0.8942647 0.9375000    1    0
## LDA    0.6875 0.8750000 0.8786765 0.9003676 0.9375000    1    0
## KNN    0.7500 0.8750000 0.9375000 0.9188971 0.9852941    1    0
```

Now looking at the test set performance.

```
lda.pred <- predict(model.lda, newdata = Breast_Cancer[-rowTrain,], type = "prob")[,2]
glm.pred <- predict(model.glm, newdata = Breast_Cancer[-rowTrain,], type = "prob")[,2]
glmn.pred <- predict(model.glmn, newdata = Breast_Cancer[-rowTrain,], type = "prob")[,2]
knn.pred <- predict(model.knn, newdata = Breast_Cancer[-rowTrain,], type = "prob")[,2]


roc.lda <- roc(Breast_Cancer$Class_cancer[-rowTrain], lda.pred)
roc.glm <- roc(Breast_Cancer$Class_cancer[-rowTrain], glm.pred)
roc.glmn <- roc(Breast_Cancer$Class_cancer[-rowTrain], glmn.pred)
roc.knn <- roc(Breast_Cancer$Class_cancer[-rowTrain], knn.pred)

auc <- c(roc.glm$auc[1], roc.glmn$auc[1], roc.lda$auc[1], roc.knn$auc[1])

plot(roc.glm, legacy.axes = TRUE)
plot(roc.glmn, col = 2, add = TRUE)
plot(roc.lda, col = 3, add = TRUE)
plot(roc.knn, col = 6, add = TRUE)
modelNames <- c("glm","glmn","lda","knn")
legend("bottomright", legend = paste0(modelNames, ": ", round(auc,3)),
       col = 1:6, lwd = 2)
```


Sensitivity
1 − Specificity
glm: 0.995
glmn: 0.992
lda: 0.994
knn: 0.993