

Assignment 2

Maggie Wang, Sogand Golshahian, Elias Krapf

2023-10-10

Setup

```
# Load required libraries
library(ggplot2)
library(ggbiplot)
library(ROCR)
library(corrplot)
library(ISLR)
library(caret)
library(randomForest)

# Read data
ovarian.data <- na.omit(read.delim("ovarian.data", sep=",", header = FALSE))
features <- c("perimeter", "area", "smoothness", "symmetry", "concavity",
             paste("protein", seq(1, 25), sep=""))
names(ovarian.data) <- c("cell_id", "diagnosis", features)
# paste0(features, "_mean"), paste0(features, "_se"), paste0(features, "_worst"))

dim(ovarian.data)
head(ovarian.data)
```

Q1. Dimensionality Reduction

Q1.1

```
ovarian.pca <- prcomp(ovarian.data[,c(3:32)], center = TRUE, scale. = TRUE)
summary(ovarian.pca)
```



```
## Importance of components:
##              PC1      PC2      PC3      PC4      PC5      PC6      PC7
## Standard deviation    3.5820 2.2873 1.62395 1.37410 1.24910 1.0844 0.8306
## Proportion of Variance 0.4277 0.1744 0.08791 0.06294 0.05201 0.0392 0.0230
## Cumulative Proportion 0.4277 0.6021 0.68997 0.75291 0.80492 0.8441 0.8671
##              PC8      PC9     PC10     PC11     PC12     PC13     PC14
## Standard deviation    0.74686 0.67762 0.61684 0.60200 0.5771 0.5139 0.5021
## Proportion of Variance 0.01859 0.01531 0.01268 0.01208 0.0111 0.0088 0.0084
## Cumulative Proportion 0.88571 0.90101 0.91369 0.92578 0.9369 0.9457 0.9541
##              PC15     PC16     PC17     PC18     PC19     PC20     PC21
## Standard deviation    0.45896 0.3989 0.3834 0.36254 0.32797 0.30949 0.3001
```

```
## Proportion of Variance 0.00702 0.0053 0.0049 0.00438 0.00359 0.00319 0.0030
## Cumulative Proportion 0.96110 0.9664 0.9713 0.97569 0.97928 0.98247 0.9855
##          PC22    PC23    PC24    PC25    PC26    PC27    PC28
## Standard deviation    0.27191 0.26081 0.24722 0.2326 0.22154 0.20068 0.18042
## Proportion of Variance 0.00246 0.00227 0.00204 0.0018 0.00164 0.00134 0.00108
## Cumulative Proportion 0.98794 0.99020 0.99224 0.9940 0.99568 0.99702 0.99811
##          PC29    PC30
## Standard deviation    0.17164 0.16532
## Proportion of Variance 0.00098 0.00091
## Cumulative Proportion 0.99909 1.00000
```

```
str(ovarian.pca)
```

```
## List of 5
## $ sdev      : num [1:30] 3.58 2.29 1.62 1.37 1.25 ...
## $ rotation: num [1:30, 1:30] -0.22 -0.11 -0.229 -0.222 -0.137 ...
##   .. attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:30] "perimeter" "area" "smoothness" "symmetry" ...
##   .. ..$ : chr [1:30] "PC1" "PC2" "PC3" "PC4" ...
## $ center   : Named num [1:30] 14.1809 19.3922 92.1982 663.7854 0.0965 ...
##   .. attr(*, "names")= chr [1:30] "perimeter" "area" "smoothness" "symmetry" ...
## $ scale    : Named num [1:30] 3.5715 4.2746 24.1993 354.8356 0.0142 ...
##   .. attr(*, "names")= chr [1:30] "perimeter" "area" "smoothness" "symmetry" ...
## $ x        : num [1:625, 1:30] -4.476 0.448 1.916 1.874 -2.802 ...
##   .. attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:625] "1" "2" "3" "4" ...
##   .. ..$ : chr [1:30] "PC1" "PC2" "PC3" "PC4" ...
## - attr(*, "class")= chr "prcomp"
```

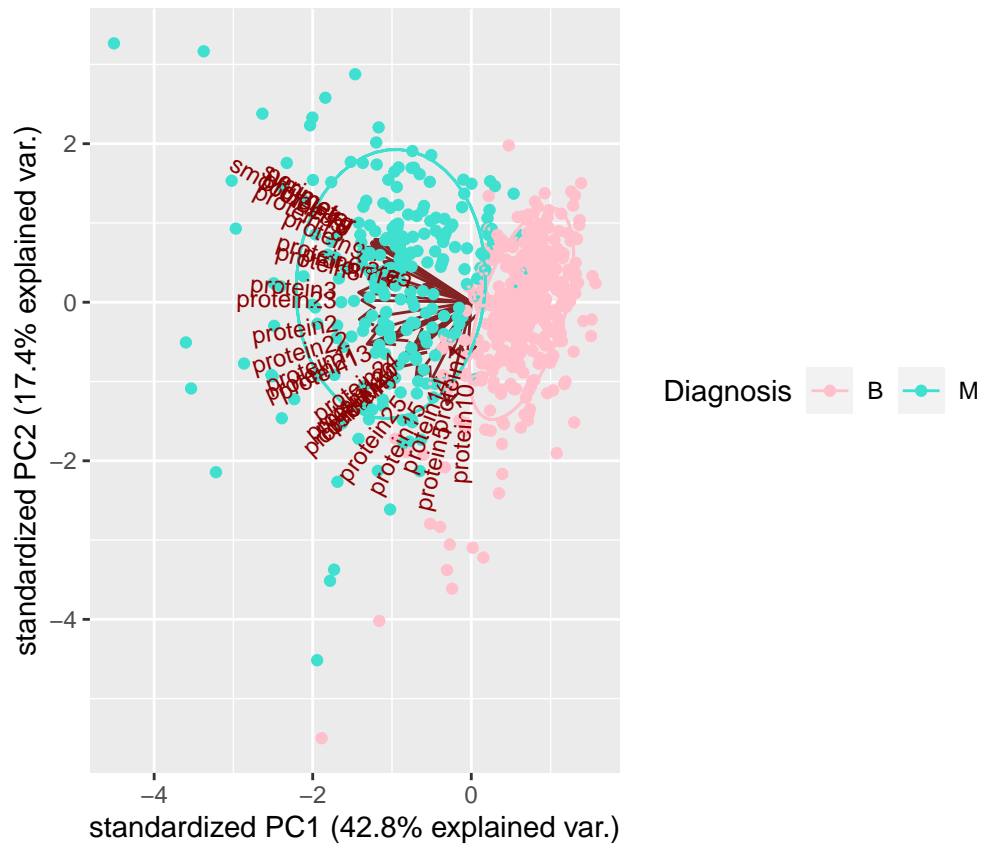
About 42.77% of the variation in the data is associated with PC1.

Q1.2 To represent 90% of the variance in the data by dimensionality reduction, you would need about 9 PCs.

Q1.3

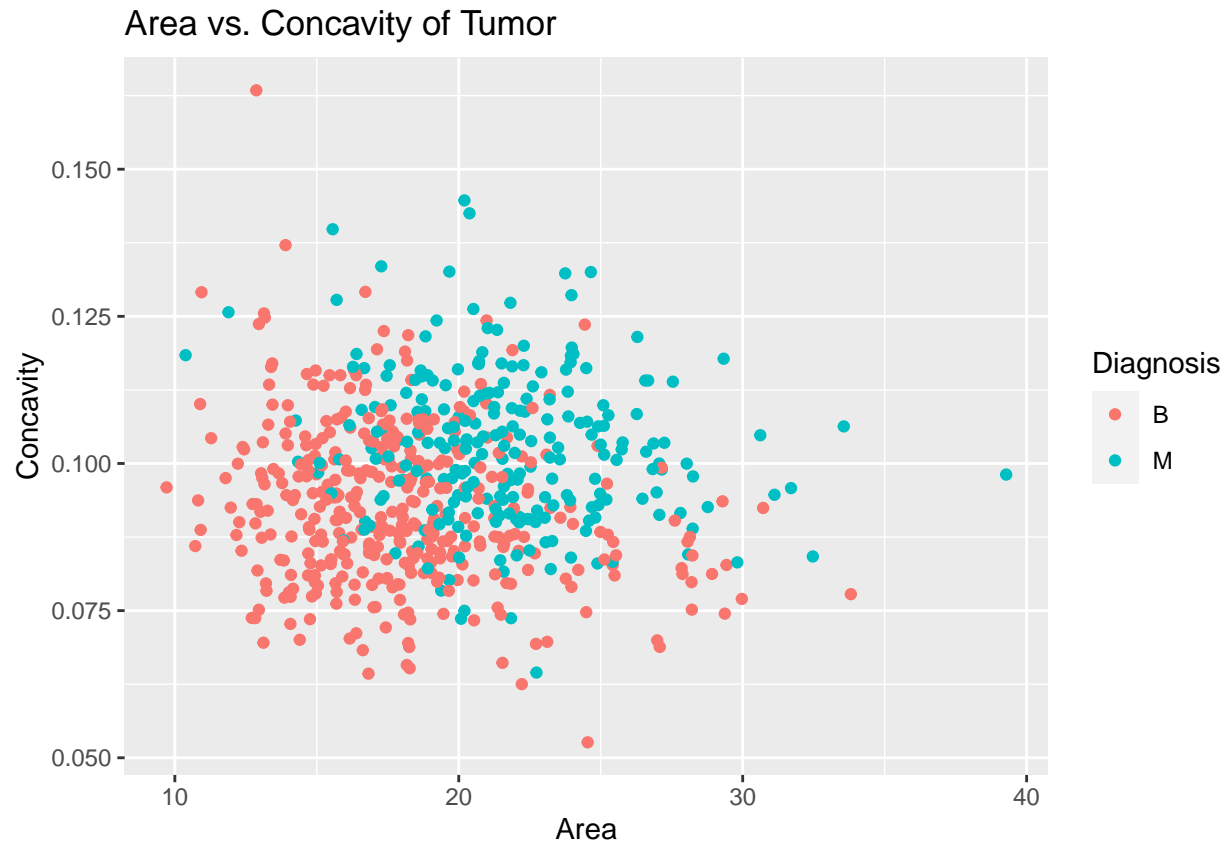
```
diagnosis <- ovarian.data[,2]

ggbiplot(ovarian.pca, choices=c(1,2), ellipse=TRUE, groups=diagnosis) +
  scale_color_manual(name="Diagnosis", values=c("pink", "turquoise")) +
  scale_shape_manual(name="Variety", values=c(2)) +
  geom_point(aes(colour=diagnosis), size = 0.01) +
  theme(legend.direction = "horizontal", legend.position = "right")
```



Q1.4

```
q1.4_plot <- ggplot(ovarian.data, aes(x = area, y = concavity)) +
  geom_point(aes(color = diagnosis)) +
  labs(title = "Area vs. Concavity of Tumor",
       x = "Area",
       y = "Concavity",
       color = "Diagnosis")
q1.4_plot
```



Q1.5 The first plot using the first two important PCs has two distinct groups, while in the second one they are a lot more mixed. This is because the first two PCs have the highest proportion of the variation in the dataset, so they will have the most difference between them.

Q1.6

Q2. Clustering

Q2.1

```
# Scaling the data
ovarian.scaled <- scale(ovarian.data[,c(3:32)])

# Set seed to get reproducible results

# Performing kmeans
km.out <- kmeans(ovarian.scaled, centers = 2, iter.max = 1, nstart = 20)
km.out$cluster <- ifelse(km.out$cluster == 1, "M", "B")
table(ovarian.data$diagnosis, km.out$cluster)
```

```
##
##      B   M
## B  14 371
## M 205  35
```

```
mean(ovarian.data$diagnosis == km.out$cluster)
```

```
## [1] 0.0784
```

There is a good amount of concordance between the identified clusters and the true labels of the cell. The model had an accuracy of 92.16%.

Q2.2

```
accuracies <- numeric(10)
```

```
# Repeat kmeans 10 times
```

```
for(i in 1:10){  
  km.out <- kmeans(ovarian.scaled, centers = 2, iter.max = 10, nstart = 20)  
  km.out$cluster <- ifelse(km.out$cluster == 1, "M", "B")  
  accuracies[i] <- mean(ovarian.data$diagnosis == km.out$cluster)  
}
```

```
mean(accuracies)
```

```
## [1] 0.58432
```

The values change from run to run because the results of the kmeans algorithm is dependent on the initialization of the centers, which is different each time.

Q2.3

```
# Transform pca results to dataframe
```

```
pca.data <- as.data.frame(ovarian.pca$x[,1:5])
```

```
# Perform kmeans analysis
```

```
km.out <- kmeans(pca.data, centers = 2, nstart = 20)  
km.out$cluster <- ifelse(km.out$cluster == 1, "M", "B")  
table(ovarian.data$diagnosis, km.out$cluster)
```

```
##  
##      B      M  
## B 369   16  
## M   35  205
```

```
mean(ovarian.data$diagnosis == km.out$cluster)
```

```
## [1] 0.9184
```

```
# Perform kmeans analysis 10 times
```

```
accuracies.pca <- numeric(10)
```

```
for(i in 1:10){  
  km.out <- kmeans(pca.data, centers = 2, nstart = 20)  
  km.out$cluster <- ifelse(km.out$cluster == 1, "M", "B")  
  accuracies.pca[i] <- mean(ovarian.data$diagnosis == km.out$cluster)  
}
```

```
mean(accuracies.pca)
```

```
## [1] 0.5
```

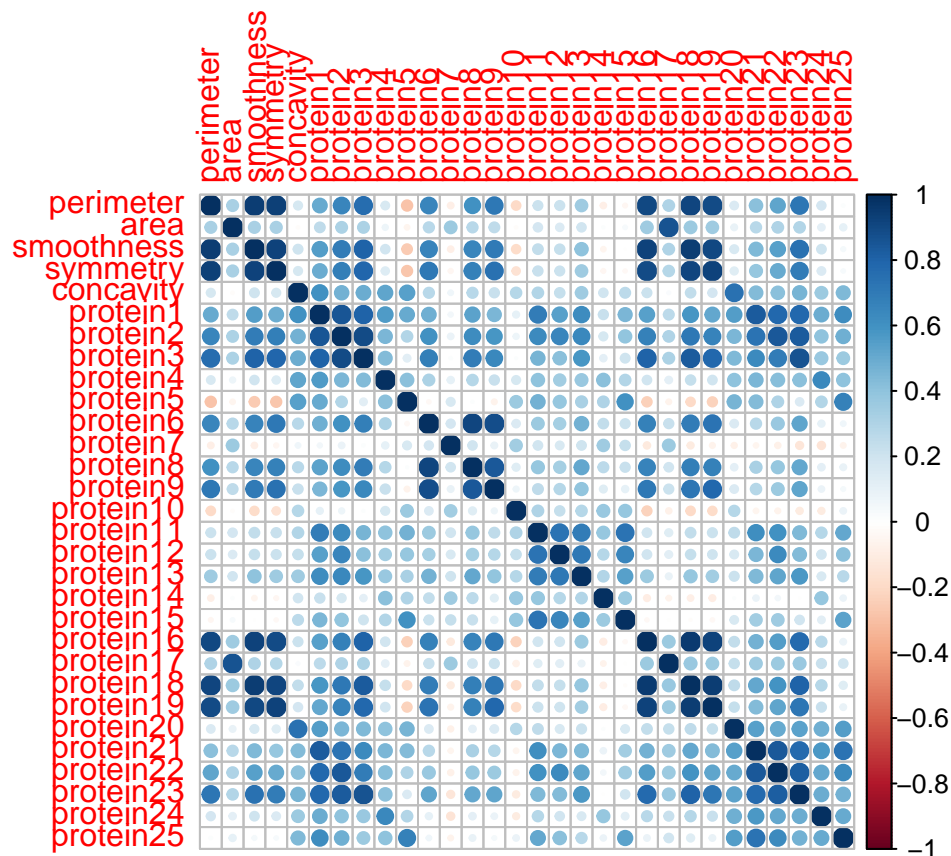
Q2.4 The results from 2.3 were very slightly worse than 2.2. The highest average from 2.2 was 0.9216, while the highest from 2.3 was 0.9184. This is because the entire data set is used in 2.2 and most of the variance in the data is covered.

Q3. Classification

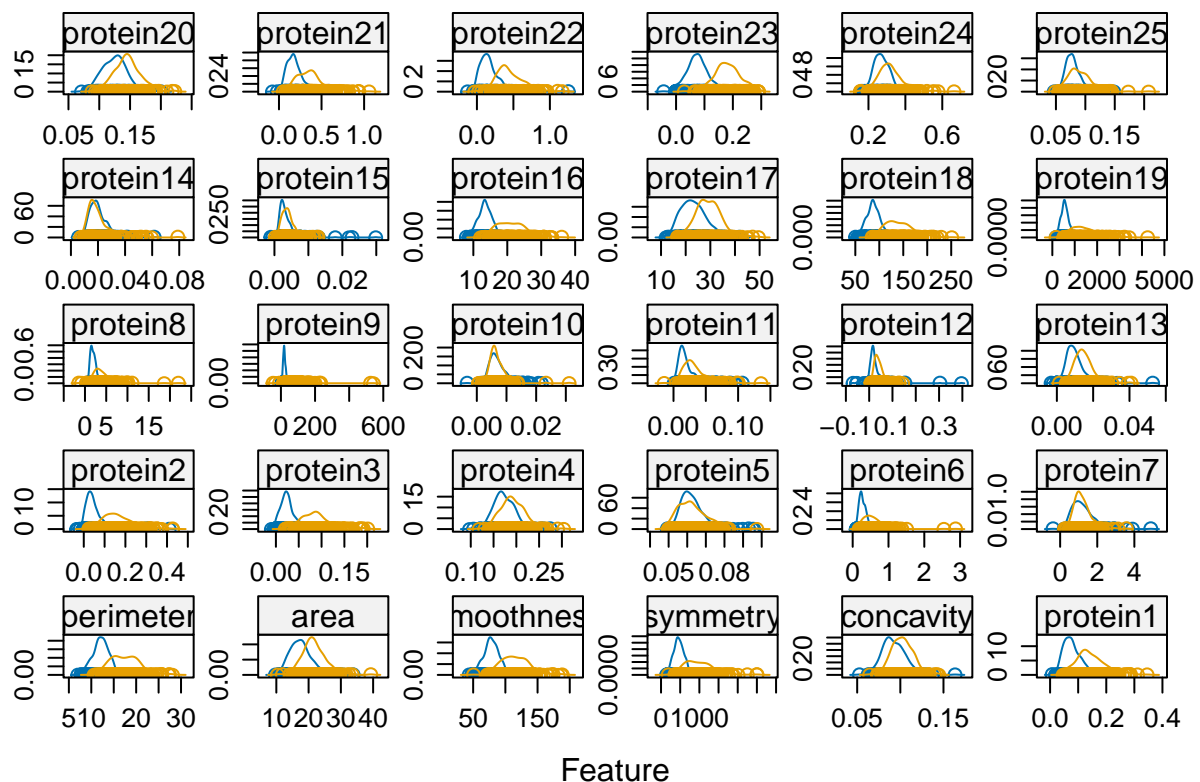
```
# Divide dataset into training and testing sets
ovarian.data.train <- ovarian.data[sample(nrow(ovarian.data))[1:(nrow(ovarian.data)/2)],]
ovarian.data.test  <- ovarian.data[sample(nrow(ovarian.data))[(nrow(ovarian.data)/2):(nrow(ovarian.data))],]
```

Q3.1

```
# Plot correlation between pairs of variables
correlations <- cor(ovarian.data[,3:32])
corrplot(correlations, method="circle")
```



```
# Plot density distribution of each variable, separated by diagnosis
x <- ovarian.data[,3:32]
y <- as.factor(ovarian.data[,2])
scales <- list(x=list(relation="free"), y=list(relation="free"))
featurePlot(x=x, y=y, plot="density", scales=scales)
```



```
# Change diagnosis column to factors
ovarian.data.train$diagnosis <- as.factor(ovarian.data.train$diagnosis)

# Logistic regression training model
training.model <- glm(diagnosis ~. -cell_id, data = ovarian.data.train, family = binomial)

# Predicting on testing model
probabilities <- predict(training.model, ovarian.data.test, type = "response")
predicted.diagnosis <- ifelse(probabilities > 0.5, "M", "B")
prediction <- as.factor(predicted.diagnosis)
actual <- as.factor(ovarian.data.test$diagnosis)

# Confusion matrix
table(prediction, actual)

##           actual
## prediction   B   M
##           B 195   5
##           M   2 111

# To calculate accuracy, precision, recall
accuracy <- mean(prediction == actual)
precision <- posPredValue(prediction, actual, positive='M', negative = 'B')
recall <- sensitivity(prediction, actual, positive="M")
accuracy
```

```
## [1] 0.9776358
```

```
precision
```

```
## [1] 0.9823009
```

```
recall
```

```
## [1] 0.9568966
```

Q3.2

```
# Logistic regression training model using top 5 PCs
pca.training.model <- glm(diagnosis ~ perimeter + area + smoothness + symmetry
                          + concavity, data = ovarian.data.train, family = binomial)

# Predicting on testing set
pca.probabilities <- predict(pca.training.model, ovarian.data.test, type = "response")
pca.predicted.diagnosis <- ifelse(pca.probabilities > 0.5, "M", "B")
pca.prediction <- as.factor(pca.predicted.diagnosis)

# Confusion matrix
table(pca.prediction, actual)
```

```
##               actual
## pca.prediction  B   M
##               B 186  14
##               M  11 102
```

```
# To calculate accuracy, precision, recall
pca.accuracy <- mean(pca.prediction == actual)
pca.precision <- posPredValue(pca.prediction, actual, positive='M', negative = 'B')
pca.recall <- sensitivity(pca.prediction, actual, positive="M")
pca.accuracy
```

```
## [1] 0.9201278
```

```
pca.precision
```

```
## [1] 0.9026549
```

```
pca.recall
```

```
## [1] 0.8793103
```

Q3.3

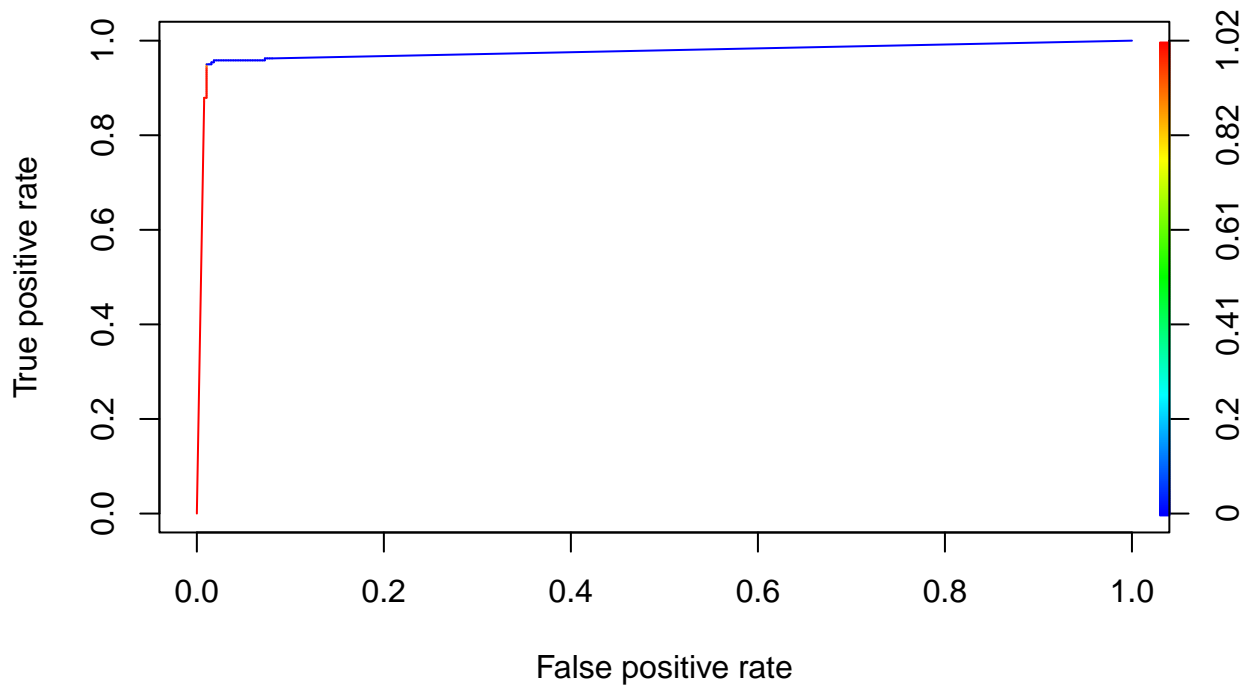
Q3.4

Q3.5


```

pred.probab <- predict(training.model, ovarian.data, type="response")
predict <- prediction(pred.probab, ovarian.data$diagnosis, label.ordering=c("B","M"))
perform <- performance(predict,"tpr","fpr")
plot(perform,colorize=TRUE)

```



Q3.6

```

set.seed(123)

# Split into training (70%) and testing (30%)
chunk <- sample(nrow(ovarian.data), 0.7 * nrow(ovarian.data))
rf.training <- ovarian.data[chunk, ]
rf.testing <- ovarian.data[-chunk, ]

# Random forest model
rf.training$diagnosis <- as.factor(rf.training$diagnosis)
ovarian.rf <- randomForest(diagnosis ~.-cell_id, rf.training)

# Predicting on train set
pred.train <- predict(ovarian.rf, rf.training, type = "class")

# Checking classification accuracy
table(pred.train, rf.training$diagnosis)

```

##

```
## pred.train   B    M
##             B 273   0
##             M   0 164
```

```
# Predicting on Validation set
pred.test <- predict(ovarian.rf, rf.testing, type = "class")

# Checking classification accuracy
mean(pred.test == rf.testing$diagnosis)
```

```
## [1] 0.962766
```

```
table(pred.test, rf.testing$diagnosis)
```

```
##
## pred.test   B    M
##           B 109   4
##           M   3  72
```

```
# Repeat with top 5 PCs
```

```
# Random forest model
pca.rf <- randomForest(diagnosis ~ perimeter + area + smoothness + symmetry
                        + concavity, rf.training)
```

```
# Predicting on Validation set
pca.pred.test <- predict(pca.rf, rf.testing, type = "class")

# Checking classification accuracy
mean(pca.pred.test == rf.testing$diagnosis)
```

```
## [1] 0.9308511
```

```
table(pca.pred.test, rf.testing$diagnosis)
```

```
##
## pca.pred.test   B    M
##               B 107   8
##               M   5  68
```

Contributions