Analyzing the Wisconsin Breast Cancer Database

MOHAMMAD RAZIV HASAN

2025-06-12

```r
# Loading the mlbench package
library(mlbench)
# Load the data
data(BreastCancer)
# Loading the dplyr package
library(dplyr)
# Loading the bestglm package
library(bestglm)
# Loading the glmnet library
library(glmnet)
# Loading the MASS package
library(MASS)
# Loading the nclSLR package
library(nclSLR)

# Data Preparation

# Technically, the nine cytological characteristics are ordinal variables on a
# 1 - 10 scale. In the BreastCancer data, they are encoded as factors. For the
# purposes of this project, we will treat them as quantitative variables. We
# will therefore convert the factor variables to quantitative ones. This will
# allow us to easily fit them into machine learning models.

# We will now convert all columns into numeric data.
BreastCancer$Id = as.numeric(BreastCancer$Id)
BreastCancer$Cl.thickness = as.numeric(BreastCancer$Cl.thickness)
BreastCancer$Cell.size = as.numeric(BreastCancer$Cell.size)
BreastCancer$Cell.shape = as.numeric(BreastCancer$Cell.shape)
BreastCancer$Marg.adhesion = as.numeric(BreastCancer$Marg.adhesion)
BreastCancer$Epith.c.size = as.numeric(BreastCancer$Epith.c.size)
BreastCancer$Bare.nuclei = as.numeric(BreastCancer$Bare.nuclei)
BreastCancer$Bl.cromatin = as.numeric(BreastCancer$Bl.cromatin)
BreastCancer$Normal.nucleoli = as.numeric(BreastCancer$Normal.nucleoli)
BreastCancer$Mitoses = as.numeric(BreastCancer$Mitoses)
BreastCancer$Class = as.numeric(BreastCancer$Class)

# This data set contains some missing observations on predictors, encoded as
# NA. For the purposes of this project, we will remove all of the rows where
# there are missing values before carrying out any further analysis.

BreastCancer = na.omit(BreastCancer)
```

```r
# We will now look for duplicate entries in data set by checking whether all
# the values in the Id column are unique or not. We are checking by the Id
# column as each woman has a unique Id whereas two or more women may
# have the same readings for the other variables.

print(nrow(BreastCancer) == nrow(distinct(BreastCancer, Id, .keep_all=TRUE)))
```

```r
# This shows us that there are duplicate entries in the data set. So we will
# now remove the duplicate entries.

BreastCancer = distinct(BreastCancer, Id, .keep_all=TRUE)

# For the sake of simplicity, we will turn the Class column of the data set
# into values of 0 and 1. 0 will represent "benign" and 1 will represent
# "malignant".

BreastCancer$Class = BreastCancer$Class - 1

# From here on out, we will not need the Id column anymore as our aim is to
# find out how well one can identify the nature of abnormal-appearing breast
# tissue using the nine cytological characteristics in this data set.

# So now we will create a new DataFrame without the Id column.

BreastCancer_noId = BreastCancer[,-1]

# For unsupervised learning methods, our goal will be to find out how many
# clusters we can group our data into. So we will be further removing the
# Class column.

BreastCancer_unsupervised = BreastCancer_noId[,1:9]
```

Abstract

In this project, we will analyze the BreastCancer data set which concerns characteristics of breast tissue samples collected from 699 women in Wisconsin using fine needle aspiration cytology (FNAC). Nine easily-assessed cytological characteristics were measured for each tissue sample on a one to ten scale. Smaller numbers indicate cells that looked healthier in terms of that characteristic. Further histological examination established whether each of the samples was benign or malignant.

Our goal is to find out how well one can identify the nature of abnormal-appearing breast tissue using the nine cytological characteristics in this data set, that is, whether the tissue is benign or malignant.

We began by cleaning our data and getting it ready for our analysis. More details about this can be found in the corresponding html file. For exploratory data analysis we looked at the correlation matrix of the data and the distribution of the classes. Furthermore, we used unsupervised learning methods to determine whether the appropriate number of clusters matched the number of classes. We then proceeded to use supervised learning models to find the best possible subset of predictor variables and trained some classification models using this subset. We calculated test errors and concluded that one can identify the nature of abnormal-appearing breast tissue using the nine cytological characteristics rather well. As with any analysis, there were limitations and possible improvements.

Exploratory Data Analysis: Data Summary

Due to the ordinal nature of the data and the large number of variables, a pairs plot of the data turns out to be too cluttered and difficult to interpret. So instead, we will look at the correlation matrix of the data.

```r
cor(BreastCancer_noId)
```

```
##                 Cl.thickness Cell.size Cell.shape Marg.adhesion Epith.c.size
## Cl.thickness       1.0000000 0.6422504  0.6548687     0.4864706    0.5169340
## Cell.size          0.6422504 1.0000000  0.9041848     0.7131159    0.7509574
## Cell.shape         0.6548687 0.9041848  1.0000000     0.6889754    0.7149982
## Marg.adhesion      0.4864706 0.7131159  0.6889754     1.0000000    0.5957037
## Epith.c.size       0.5169340 0.7509574  0.7149982     0.5957037    1.0000000
## Bare.nuclei        0.5918317 0.6818800  0.7060106     0.6724923    0.5774609
## Bl.cromatin        0.5562122 0.7607827  0.7343201     0.6672022    0.6193729
## Normal.nucleoli    0.5331138 0.7246847  0.7209817     0.5936105    0.6293423
## Mitoses            0.3503942 0.4640081  0.4430838     0.4113362    0.4818774
## Class              0.7235999 0.8208236  0.8191137     0.7101063    0.6875647
##                 Bare.nuclei Bl.cromatin Normal.nucleoli   Mitoses     Class
## Cl.thickness      0.5918317   0.5562122       0.5331138 0.3503942 0.7235999
## Cell.size         0.6818800   0.7607827       0.7246847 0.4640081 0.8208236
## Cell.shape        0.7060106   0.7343201       0.7209817 0.4430838 0.8191137
## Marg.adhesion     0.6724923   0.6672022       0.5936105 0.4113362 0.7101063
## Epith.c.size      0.5774609   0.6193729       0.6293423 0.4818774 0.6875647
## Bare.nuclei       1.0000000   0.6785264       0.5764841 0.3366509 0.8212379
## Bl.cromatin       0.6785264   1.0000000       0.6630368 0.3386258 0.7568049
## Normal.nucleoli   0.5764841   0.6630368       1.0000000 0.4228833 0.7169245
## Mitoses           0.3366509   0.3386258       0.4228833 1.0000000 0.4278489
## Class             0.8212379   0.7568049       0.7169245 0.4278489 1.0000000
```

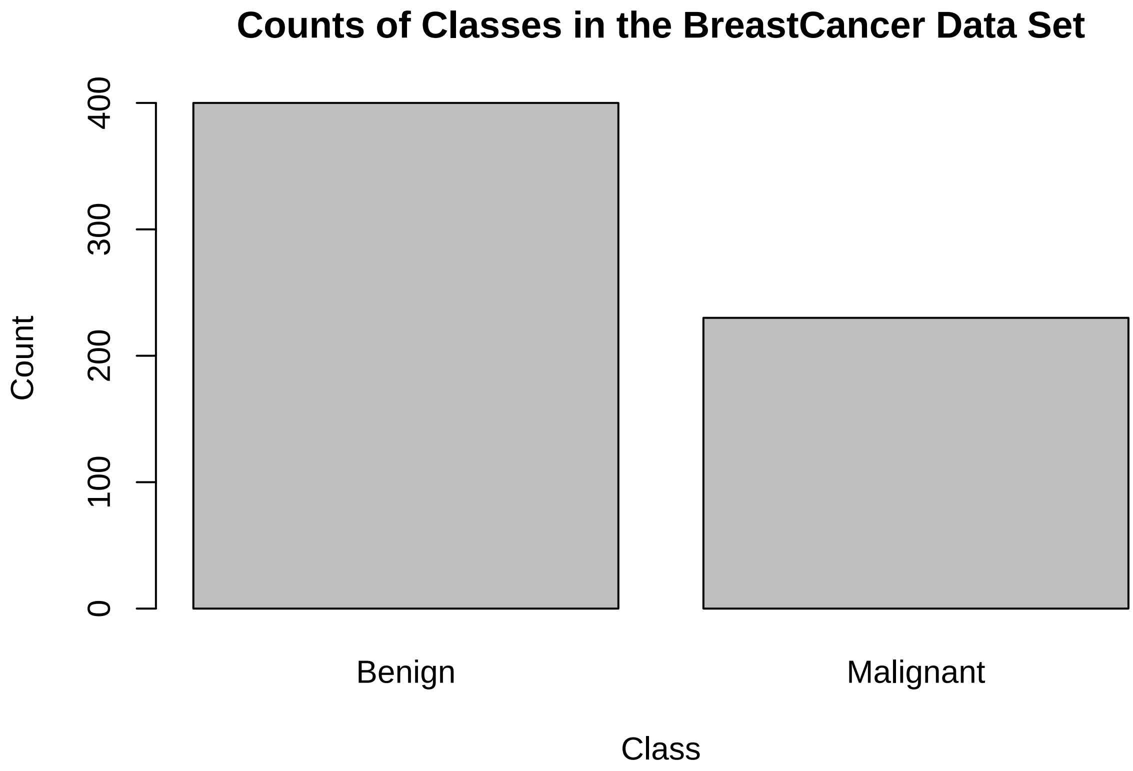
Examination of the correlation matrix reveals:

- Uniformity of Cell Size is strongly correlated with Uniformity of Cell Shape.
- Weak and moderate correlations exists among the other predictor variables.
- Strong correlations exist between Class and Uniformity of Cell Size; Class and Uniformity of Cell Shape; and Class and Bare Nuclei. A weak correlation exists between Class and Mitoses. The rest of the predictor variables have moderate correlations between the Class and themselves.

Next, we will be looking at the distribution of the classes. Since there are only two classes, we will be using a bar plot for this purpose.

```r
# Getting counts of the benign and malignant classes
count_malignant = sum(BreastCancer_noId$Class)
count_benign = nrow(BreastCancer_noId) - count_malignant

barplot(
  c(count_benign, count_malignant),
  names.arg = c("Benign", "Malignant"),
  main = "Counts of Classes in the BreastCancer Data Set",
  xlab = "Class",
  ylab = "Count"
)
```



The barplot shows us that there are almost twice as many entries of benign tissue compared to entries of malignant tissue. This might lead to the building of models that are better at identifying one class than another.

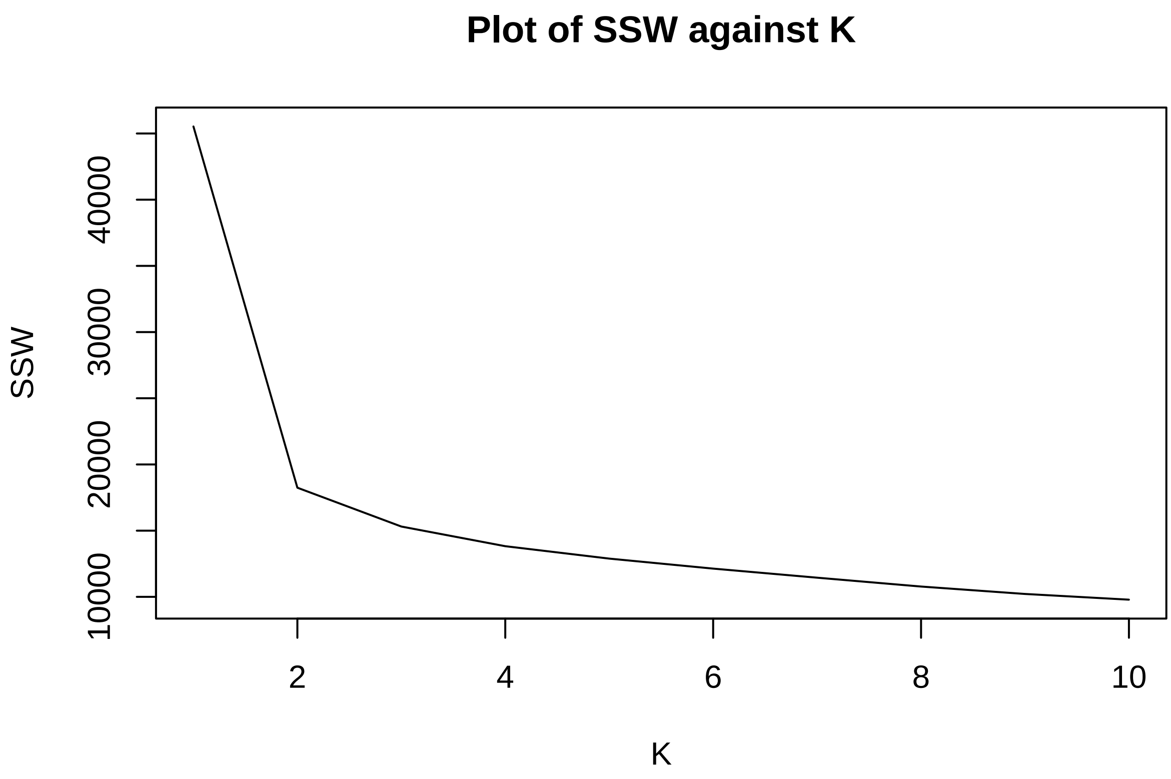
Exploratory Data Analysis: Unsupervised Learning

To understand the data better, we will try unsupervised machine learning.

We will begin with K-means Clustering. This will allow us to find the appropriate number of clusters in the data. Since we already know that there are two classes, we are expecting to find that the number of appropriate clusters to be two. In order to investigate an appropriate choice of K, we will consider the values K = 1, 2, . . . , 10 and compute the minimum within-cluster-sum-of-squares, SSW, for each. Then we will plot SSW against K.

```r
# To make our analysis reproducible, we will set a seed.
set.seed(1234)
# Setting the maximum value of K:
K_max = 10
# Setting up a vector to store the values of SSW for each value of K
SSW = numeric(K_max)
# Setting up a list to store the k_means fits
k_means_fits = list()
# Training the model
for(K in 1:K_max) {
  k_means_fits[[K]] = kmeans(BreastCancer_unsupervised, K, iter.max = 50, nstart = 20)
  SSW[K] = k_means_fits[[K]]$tot.withinss
}

plot(1:K_max, SSW, type = "l", xlab = "K", ylab = "SSW", main = "Plot of SSW against K")
```

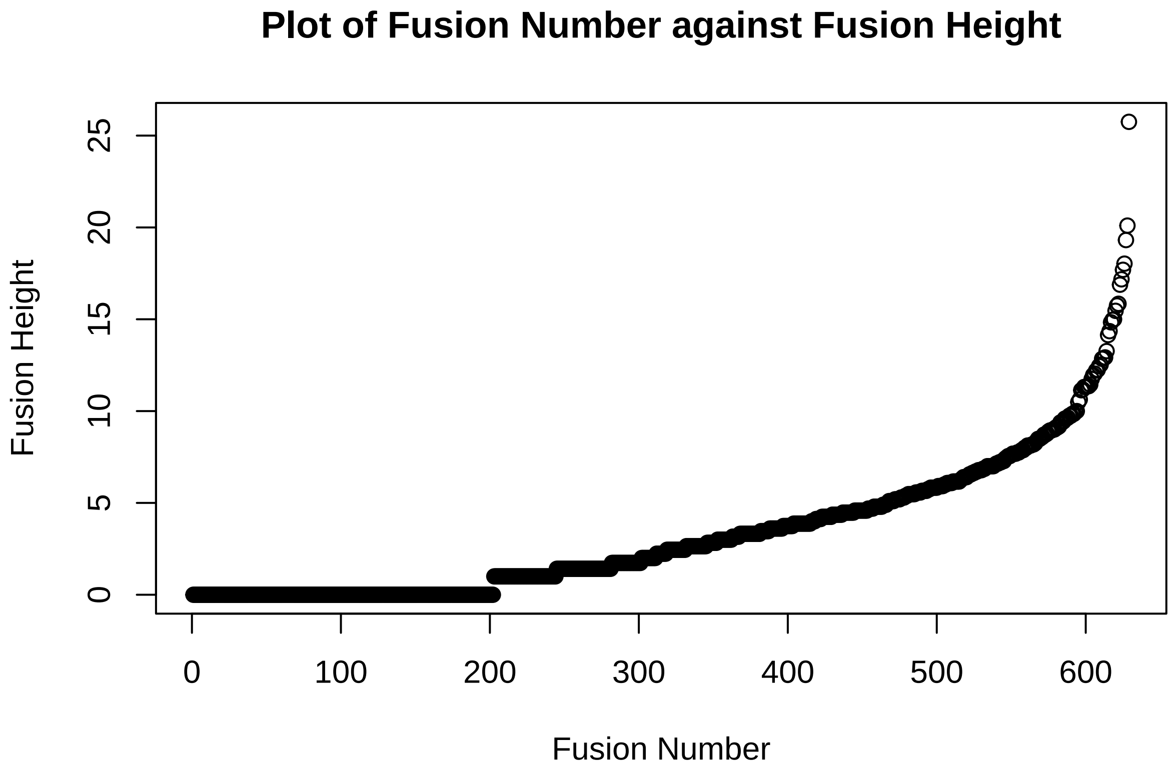


There appears to be an elbow at K = 2, which suggests that it might be appropriate to uses two clusters. This lines up with our expectations.

Next, we will use hierarchical clustering to once again find the appropriate number of clusters. We will perform the clustering based on Euclidean distance and considering complete linkage. We will then plot Fusion Height against Fusion Number.

```
# Computing the Euclidean distance matrix
d = dist(BreastCancer_unsupervised)
# Performing hierarchical clustering
hc = hclust(d, method = "complete")

# Plotting Fusion Height against Fusion Number
plot(hc$height,
     xlab = "Fusion Number",
     ylab = "Fusion Height",
     main = "Plot of Fusion Number against Fusion Height")
```



Examining the plot reveals that a very large change in Fusion Height, compared to earlier changes, occurs between the Fusion Heights of about 21 and about 26. So we will choose to cut at a height of 23.5.

```
print(unique(cutree(hc, h = 23.5)))
```

```
## [1] 1 2
```

Cutting at a height of 23.5 then reveals that it might be appropriate to use two clusters. Once again, this lines up with our expectations.

Therefore, both K-means and Hierarchical Clustering suggests that it might be appropriate to use two clusters, which matches our previous knowledge of there being two classes, benign and malignant.

Results: Supervised Learning

This is a data set that is clearly suited for using classifier methods. But before further analysis, we will scale the data.

```
# Picking out and scaling the predictor variables
X1orig = BreastCancer_noId[, 1:9]
X1 = scale(X1orig)

# Picking out the response variable
y = BreastCancer_noId[, 10]
```

```r
# Combining the scaled data and the response variable into a new DataFrame
scaled_BC = data.frame(X1, y)
```

Now we will split the data into testing and training sets, with the train to test ratio being 70:30.

```r
# Setting a seed to make the analysis reproducible
set.seed(1234)

# Creating a vector of Booleans to split the data by
sample = sample(c(TRUE, FALSE), nrow(scaled_BC), replace = TRUE, prob = c(0.7, 0.3))

# Splitting the data into the training and testing sets
train = scaled_BC[sample,]
test = scaled_BC[!sample,]
```

We will now carry out subset selection in logistic regression. We will apply best subset selection using the AIC and the BIC by two applications of the bestglm function. In both cases, small values of the information criterion indicate a better model.

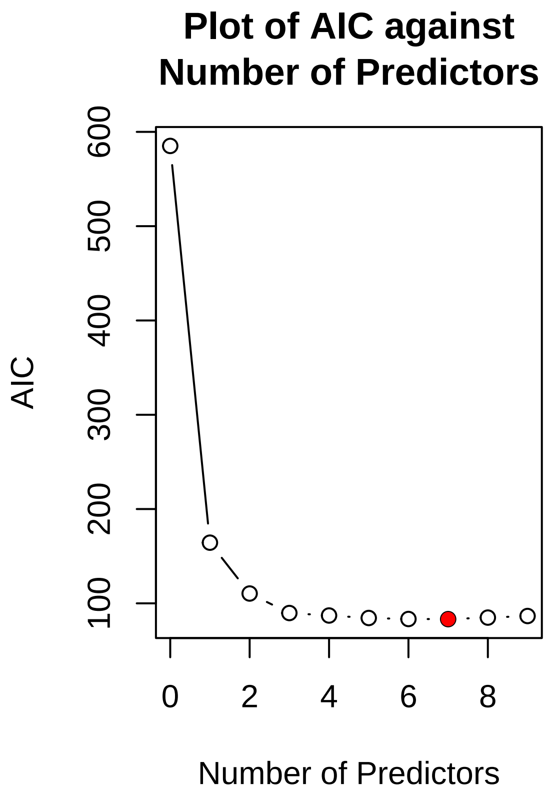
```r
# Applying best subset selection
bss_fit_AIC = bestglm(train, family = binomial, IC = "AIC")
bss_fit_BIC = bestglm(train, family = binomial, IC = "BIC")
```

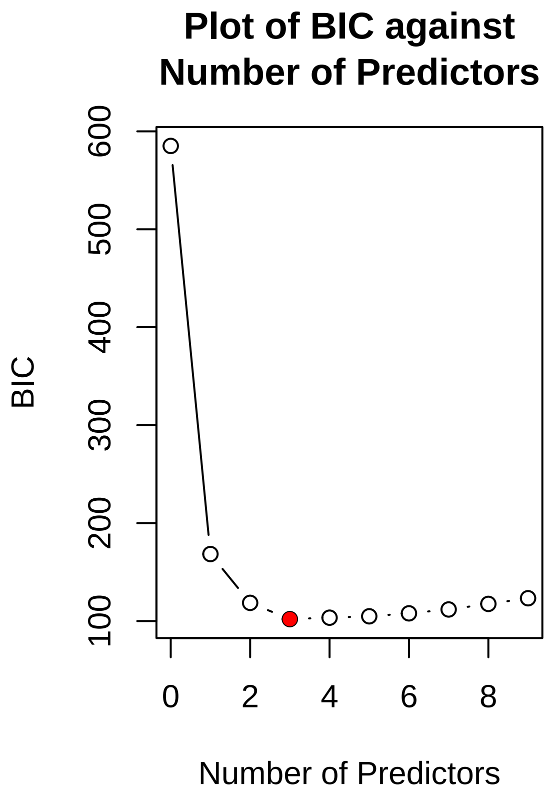
We will view plots to show how these criteria vary with the number of predictors.

```r
# Formatting the plots so that they appear side-by-side for better comparison
par(mfcol = c(1, 2))

# Plotting AIC against Number of Predictors
plot(
  0:9, bss_fit_AIC$Subsets$AIC,
  xlab = "Number of Predictors", ylab = "AIC",
  main = "Plot of AIC against\nNumber of Predictors",
  type = "b"
)
points(
  bss_fit_AIC$ModelReport$Bestk, bss_fit_AIC$Subsets$AIC[bss_fit_AIC$ModelReport$Bestk + 1],
  col = "red", pch =16
)

# Plotting BIC against Number of Predictors
plot(
  0:9, bss_fit_BIC$Subsets$BIC,
  xlab = "Number of Predictors", ylab = "BIC",
  main = "Plot of BIC against\nNumber of Predictors",
  type = "b"
)
points(
  bss_fit_BIC$ModelReport$Bestk, bss_fit_BIC$Subsets$BIC[bss_fit_BIC$ModelReport$Bestk + 1],
  col = "red", pch =16
)
```





Looking at the plot of AIC against Number of Predictors reveals that even though the AIC is lowest at seven predictors, the AIC does not decrease much from three predictors onward. On the other hand, the plot of BIC against Number of Predictors reveals that the BIC is lowest at three predictors. Therefore, we will choose a subset of three predictor variables.

We will now extract the variables from the best-fitting 3-predictor models and see if they are the same variables.

```
# Checking which variables are in the 3-predictor AIC model
bss_fit_AIC$Subsets[3+1,]
```

```
##    Intercept Cl.thickness Cell.size Cell.shape Marg.adhesion Epith.c.size
## 3       TRUE         TRUE      TRUE      FALSE         FALSE        FALSE
##    Bare.nuclei Bl.cromatin Normal.nucleoli Mitoses logLikelihood      AIC
## 3         TRUE       FALSE           FALSE   FALSE     -41.86833 89.73666
```

```
# Checking which variables are in the 3-predictor BIC model
bss_fit_BIC$Subsets[3+1,]
```

```
##    Intercept Cl.thickness Cell.size Cell.shape Marg.adhesion Epith.c.size
## 3*      TRUE         TRUE      TRUE      FALSE         FALSE        FALSE
##    Bare.nuclei Bl.cromatin Normal.nucleoli Mitoses logLikelihood      BIC
## 3*        TRUE       FALSE           FALSE   FALSE     -41.86833 101.9833
```

This shows us that both the models choose the same predictor variables for the 3-predictor models.

We will now create reduced data sets for the train and test data containing only the selected predictor variables.

```r
reduced_train = train[, c(TRUE, TRUE, FALSE, FALSE, FALSE, TRUE, FALSE, FALSE, FALSE, TRUE)]
reduced_test = test[, c(TRUE, TRUE, FALSE, FALSE, FALSE, TRUE, FALSE, FALSE, FALSE, TRUE)]
```

We will now train a logistic regression model with LASSO penalty and see how it performs. To do this, we need to find the optimal value of the tuning parameter first. We will perform cross-validation to achieve this. Afterwards, we will train the model with the optimal value of the tuning parameter and then compute the test error, which in this case will be the proportion of misclassified predictions.

```r
# Creating grid of values for the tuning parameter
grid = 10^seq(-4, -1, length.out = 100)

# Cross-validation to find optimal value of tuning parameter
lasso_cv_fit = cv.glmnet(
  data.matrix(reduced_train[,1:3]), data.matrix(reduced_train[,4]), family = "binomial",
  alpha = 1, standardize = FALSE, lambda = grid, type.measure = "class"
)

# Identifying the optimal value of the tuning parameter
lambda_lasso_min = lasso_cv_fit$lambda.min

# Fitting logistic regression model with LASSO penalty to training data using optimal
# value of tuning parameter
lasso_train = glmnet(
  data.matrix(reduced_train[,1:3]), data.matrix(reduced_train[,4]), family = "binomial",
  alpha = 1, standardize = FALSE, lambda = lambda_lasso_min
)

# Computing the predicted values for the test set
phat_test = predict(lasso_train, data.matrix(reduced_test[,1:3]),
                    s = lambda_lasso_min, type = "response")
yhat_test = ifelse(phat_test > 0.5, 1, 0)
# Computing the test error
 test_error_lasso = 1 - mean(data.matrix(reduced_test[,4]) == yhat_test)
```

The test error for our logistic regression model with LASSO penalty is 0.0260417.

We will also look at the parameter estimates associated with the optimal value of the tuning parameter.

```r
coef(lasso_train)
```

```
## 4 x 1 sparse Matrix of class "dgCMatrix"
##                      s0
## (Intercept)  -0.6906963
## Cl.thickness  0.7019690
## Cell.size     1.2009488
## Bare.nuclei   1.1741382
```

The parameters show that as the predictor variables increase, so does the response variable. Moreover, the results match what we deduced in Exploratory Data Analysis: Data Summary - Uniformity of Cell Size and Bare Nuclei have the biggest effects on the response variable.

We will now move on to training a discriminant analysis model. For brevity, we will be considering only one model: we will choose quadratic discriminant analysis (QDA) over linear discriminant analysis (LDA) as the former has less strict assumptions regarding the group covariance matrix than the latter.

```r
# Fitting the QDA classifier with training data
qda_train = qda(y ~ ., data = reduced_train)
# Computing the predicted values for the test set
qda_test = predict(qda_train, reduced_test)
yhat_test = qda_test$class
# Computing the test error
test_error_qda = 1 - mean(reduced_test$y == yhat_test)
```

The test error for our QDA model is 0.0208333.

We will also look at the confusion matrix produced by our QDA model.

```r
(confusion_matrix = table(Observed = reduced_test$y, Predicted = yhat_test))
```

```
##          Predicted
## Observed   0   1
##        0 129   3
##        1   1  59
```

The classification matrix shows that out of 192 predictions, our QDA model got only 4 wrong and the rest correct.

Conclusions and Discussion

One point to note is that neither of our models include all the predictor variables. As we saw in Exploratory Data Analysis: Data Summary, strong and moderate correlations exist among our predictor variables. So, choosing the best subset of predictor variables when training our model was very important to improve model performance. To keep the comparison fair, we trained both our models with the same subset of predictor variables.

We will choose the best model by comparing the test errors for our models. The test errors we calculated is the proportion of misclassified predictions. So, the lower the test error, the better the model. This reasoning leads us to conclude that our QDA model is the better model, as it has a lower test error than our logistic regression model with LASSO penalty.

The test errors for both the models we built were very low, with the test error of our logistic regression model with LASSO penalty being 0.0260417 and the test error of our QDA model being 0.0208333. Therefore, we can conclude by saying that one can identify the nature of abnormal-appearing breast tissue using the nine cytological characteristics rather well.

However, there is always room for improvement and one of the limitations of this analysis was that there were almost twice as many entries of benign tissue compared to entries of malignant tissue. This might lead to the building of models that are better at identifying one kind of tissue than the other. However, due to the low number of samples, it is hard to determine the nature of the misclassification errors our QDA model tends to make, that is, the errors we viewed earlier using a confusion matrix. So, we cannot determine if our model is affected by this limitation. Carrying out the study again and obtaining a larger data set with more samples will surely enable us to better test our models.