Breast Cancer Classification

Dimitrios Poulimenos - 200291237

Semester 1 - 2023/24

Introduction

In this project, I will analyse the BreastCancer data set which concerns characteristics of breast tissue samples collected from 699 women in Wisconsin using fine needle aspiration cytology (FNAC). The aim of this analysis is to create multiple classifiers based on the nine cytological characteristics in order to predict the response variable which is the "Class" column in the BreastCancer data set. Through my analysis, I am going to discover useful insights into the dataset and based on my findings I will create the classifiers and discuss their results respectively.

Data Cleaning

Before cleaning take place I will check the data shape and the structure of the data.

Data shape:

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

Data Structure:

```
## ’data.frame':    699 obs. of  11 variables:
##  $ Id             : chr  "1000025" "1002945" "1015425" "1016277" ...
##  $ Cl.thickness   : Ord.factor w/ 10 levels "1"<"2"<"3"<"4"<..: 5 5 3 6 4 8 1 2 2 4 ...
##  $ Cell.size      : Ord.factor w/ 10 levels "1"<"2"<"3"<"4"<..: 1 4 1 8 1 10 1 1 1 2 ...
##  $ Cell.shape     : Ord.factor w/ 10 levels "1"<"2"<"3"<"4"<..: 1 4 1 8 1 10 1 2 1 1 ...
##  $ Marg.adhesion  : Ord.factor w/ 10 levels "1"<"2"<"3"<"4"<..: 1 5 1 1 3 8 1 1 1 1 ...
##  $ Epith.c.size   : Ord.factor w/ 10 levels "1"<"2"<"3"<"4"<..: 2 7 2 3 2 7 2 2 2 2 ...
##  $ Bare.nuclei    : Factor w/ 10 levels "1","2","3","4",..: 1 10 2 4 1 10 10 1 1 1 ...
##  $ Bl.cromatin    : Factor w/ 10 levels "1","2","3","4",..: 3 3 3 3 3 9 3 3 1 2 ...
##  $ Normal.nucleoli: Factor w/ 10 levels "1","2","3","4",..: 1 2 1 7 1 7 1 1 1 1 ...
##  $ Mitoses        : Factor w/ 9 levels "1","2","3","4",..: 1 1 1 1 1 1 1 1 5 1 ...
##  $ Class          : Factor w/ 2 levels "benign","malignant": 1 1 1 1 1 2 1 1 1 1 ...
```

First things first I will check for any missing values in the data and I am going to remove the rows with the missing values (NA)

```
## The number of rows with missing values is: 16
```

As we can see there are 16 rows with NA values. After removing the rows with the missing values lets check how our data look like.

```
## After omitting missing values, the new number of rows with missing values is: 0
```

Data Preprocessing

Now that there are no missing values in the dataset and I am ready to continue my analysis and I will proceed with the conversion of the 9 columns that I will use as predictors for my classifiers.

```
## 'data.frame':    683 obs. of  10 variables:
##  $ Cl.thickness   : num  5 5 3 6 4 8 1 2 2 4 ...
##  $ Cell.size      : num  1 4 1 8 1 10 1 1 1 2 ...
##  $ Cell.shape     : num  1 4 1 8 1 10 1 2 1 1 ...
##  $ Marg.adhesion  : num  1 5 1 1 3 8 1 1 1 1 ...
##  $ Epith.c.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 ...
##  $ Bl.cromatin    : 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          : Factor w/ 2 levels "benign","malignant": 1 1 1 1 1 2 1 1 1 1 ...
##  - attr(*, "na.action")= 'omit' Named int [1:16] 24 41 140 146 159 165 236 250 276 293 ...
##   ..- attr(*, "names")= chr [1:16] "24" "41" "140" "146" ...
```

As we can see from the data there are two classes in the "Class" column. The first one is the **benign** and the second one is the **malignant** class. Using a linear model such as logistic regression it would be better to implement three things in my data. The first thing will be the standardization of the data because it can make it easier to compare the importance of different features. The second thing is to convert the benign class to 0 and malignant class to 1 for classification purposes. And the third and the last thing is that in the new data frame the "Id" column will not be included because it represents the unique identification of each patient therefore it will have no use for our model.

Prior to the analysis a new data frame will be created including the standardised predictor variables and the response variable which will be renamed as "y" and converted to numerical as well.

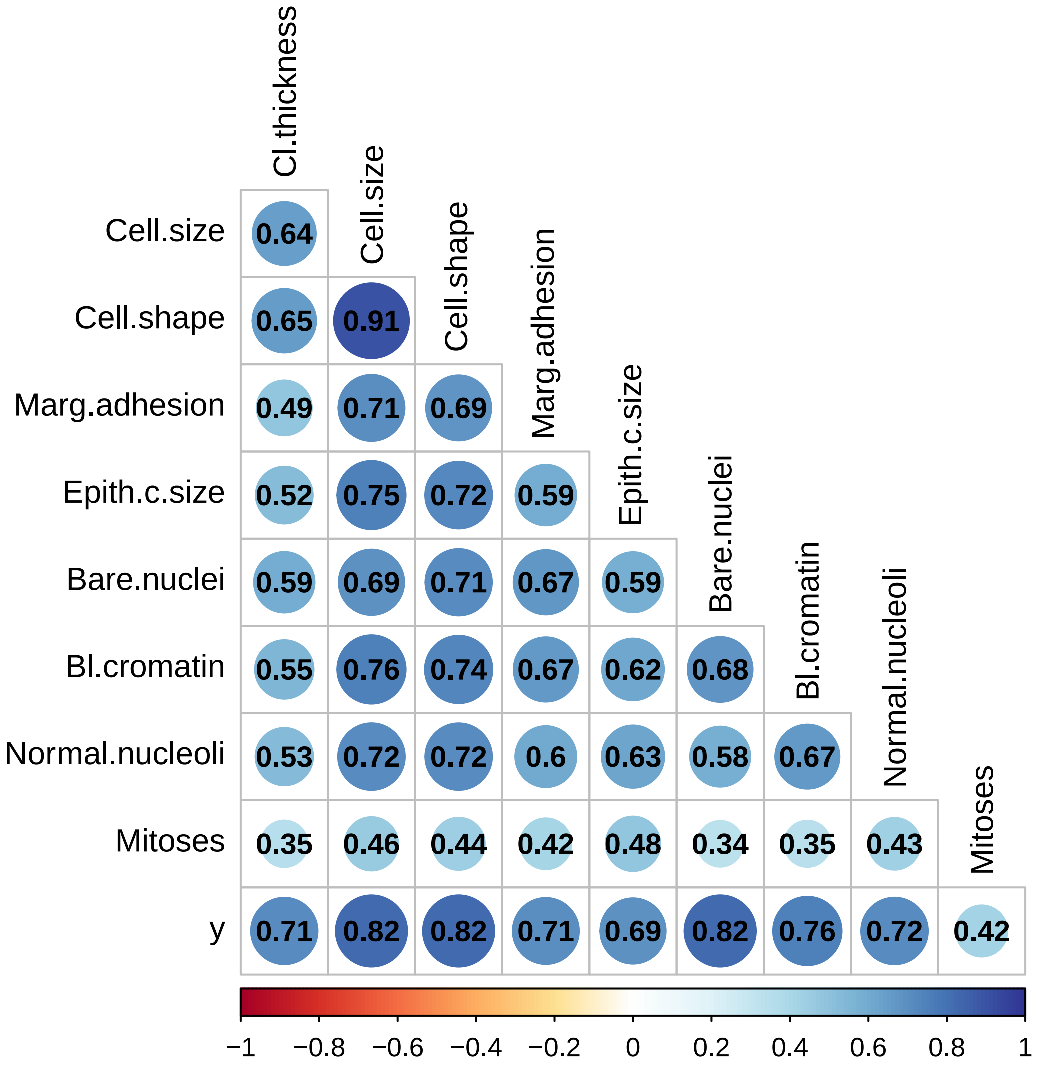
```r
# Convert the response variable to numeric (0 for 'benign', 1 for 'malignant')
y <- as.numeric(New_BreastCancer[, 10]) - 1

# Extract predictor variables
X1_original <- New_BreastCancer[, -10]

# Standardize predictor variables
X1 <- scale(X1_original)

# Combine standardized predictors and response variables in a new data frame
Breast_Cancer_Final <- data.frame(X1, y)
```

Exploratory Data Analysis (EDA)

Here I will conduct the exploratory data analysis to get some useful numerical and visual insights. In the table below we can see the distribution of each class in our data.

```
## 
##   0   1 
## 444 239
```

From the table above, it's evident that the 'benign' class is the predominant class in the data with 65.01%, significantly outnumbering the 'malignant' class with 34.99%. Now I am going to investigate the relationship of each predictor variable with the the class and also the correlation between them.



Upon examining the correlation plot, it is evident that several predictor variables exhibit strong correlations. Notably, the highest correlation is observed between Cell.size and Cell.shape, reaching 0.91. This suggests a high degree of redundancy between these two predictors, prompting consideration for potential exclusion from our models. Furthermore, there are several instances of substantial correlations exceeding 0.7 among other predictor variables. This redundancy implies that we may not require all nine predictors for our models, and a thoughtful selection process could enhance model efficiency and interpretability.

Additionally, investigating the correlation between the response variable (y) and predictors reveals intriguing patterns. The lowest correlation is found between Mitoses and y, registering at 0.42. In contrast, the remaining eight predictors exhibit correlations surpassing 0.69 with the response variable. Specifically, Bare.nuclei, Cell.size, and Cell.shape emerge as highly correlated with the response variable, each boasting a correlation coefficient of 0.82.

Next I am going to examine the mean of each column based on the response in order to extract some useful information.

Benign Class:

```
##    Cl.thickness        Cell.size       Cell.shape   Marg.adhesion    Epith.c.size
##          -0.524           -0.602           -0.603          -0.518          -0.507
##     Bare.nuclei     Bl.cromatin Normal.nucleoli         Mitoses               y
##          -0.603           -0.556           -0.527          -0.310           0.000
```

Malignant Class:

```
##    Cl.thickness        Cell.size       Cell.shape   Marg.adhesion    Epith.c.size
##           0.974            1.118            1.119           0.962           0.941
##     Bare.nuclei     Bl.cromatin Normal.nucleoli         Mitoses               y
##           1.121            1.033            0.979           0.577           1.000
```

As can be seen from the mean of the columns for each class, the values of the "malignant" class are higher than the "benign" ones. On average, tumors in the Malignant class tend to have higher values for these features compared to tumors in the Benign class.

Classification

Having outlined the distinguishing features of each tumor class, "benign" and "malignant," the next phase involves creating five classification models to effectively differentiate between these classes. Subsequently, a thorough performance comparison will be conducted based on test errors. To ensure a fair evaluation, a consistent 10-fold validation approach will be applied across all models. The selection criterion will prioritize the model demonstrating the lowest mean squared error (MSE) on the test data, with the same set of 10 folds used for each model assessment.

Logistic Regression using Subset Selection

Here I am, gearing up to implement subset selection, a method optimal for dimensionality reduction, aimed at identifying the most effective subset of predictor variables for the actual model. Given the scenario where the number of predictor variables is less than the number of observations (p < n), I have opted for an "exhaustive" subset selection approach over a "stepwise" one. Three distinct criteria will guide the selection process: the Bayesian Information Criterion (BIC), the Akaike Information Criterion (AIC), and the mean squared error. These criteria will serve as benchmarks for determining the most suitable subset of predictor variables for model refinement.

```r
# Perform best subset selection using BIC for logistic regression
best_subset_BIC_model <- bestglm(Breast_Cancer_Final, family = binomial,
                                  method = "exhaustive", nvmax = p)
```

```
## The model with the lowest BIC is at index: 5
```

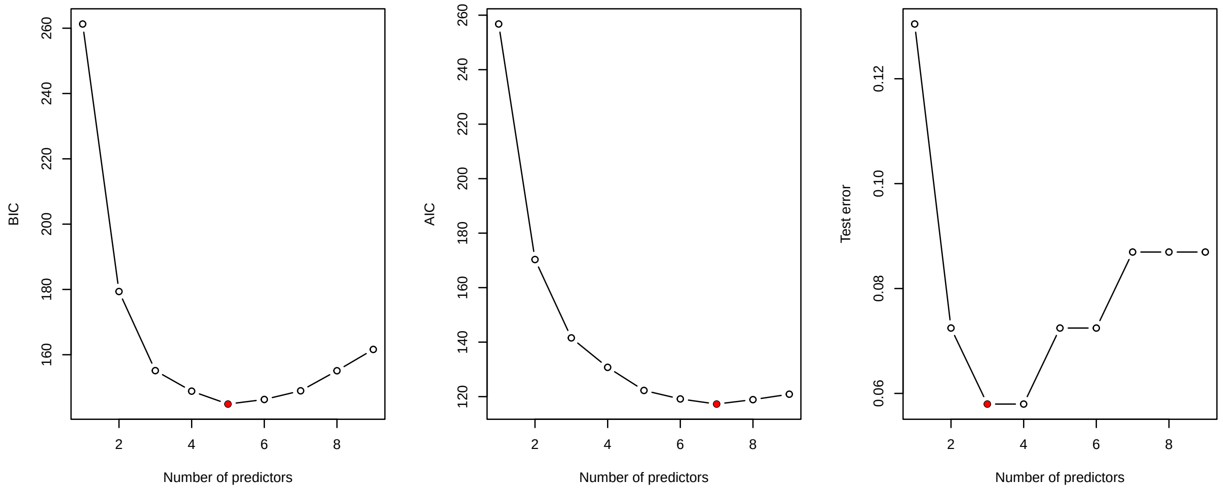
```r
# Perform best subset selection using AIC for logistic regression
best_subset_AIC_model = bestglm(Breast_Cancer_Final, family = binomial,
                                 method="exhaustive", nvmax=p, IC = "AIC")
```

```
## The model with the lowest AIC is at index: 7
```

```r
# Applying the best subset selection model to assess MSE via cross-validation
bss_mse <- reg_bss_cv(X1, y, as.matrix(best_subset_AIC_model$Subsets[2:10,2:10]), fold_index)
```

```
## The model with the lowest error is at cross-validation index: 3
```

Compare Bic, AIC and Test Error



As we can see from the above plots the best model for BIC has five predictor variables and the best model for AIC has 7. However based on the test error according to 10-fold validation, the best model has only three predictor variables. Based on the test error we can observe that the model with four predictor variables has the lowe test error as the three predictor variable model and from the BIC and AIC we can tell that the models with four predictor variables are slightly different. Therefore, the model that I will choose will be the model with the four predictor variables.

Below are the **coefficients** of the selected model:

```
##                Estimate Std. Error   z value     Pr(>|z|)
## (Intercept)  -1.142954  0.2674806 -4.273036 1.928297e-05
## Cl.thickness  1.593822  0.3521535  4.525929 6.013080e-06
## Cell.shape    1.729167  0.4568975  3.784584 1.539661e-04
## Bare.nuclei   1.560146  0.3259944  4.785807 1.703018e-06
## Bl.cromatin   1.451359  0.3706230  3.915997 9.003119e-05
```

The Cl.thickness and Cell.shape have the higher coefficients with 1.593 and 1.729 respectively.

And now we can print the **test error** value of the selected model:

```
## Lowest error value for Subset Selection: 0.05797101
```

And finally the **confusion matrix** of the selected model:

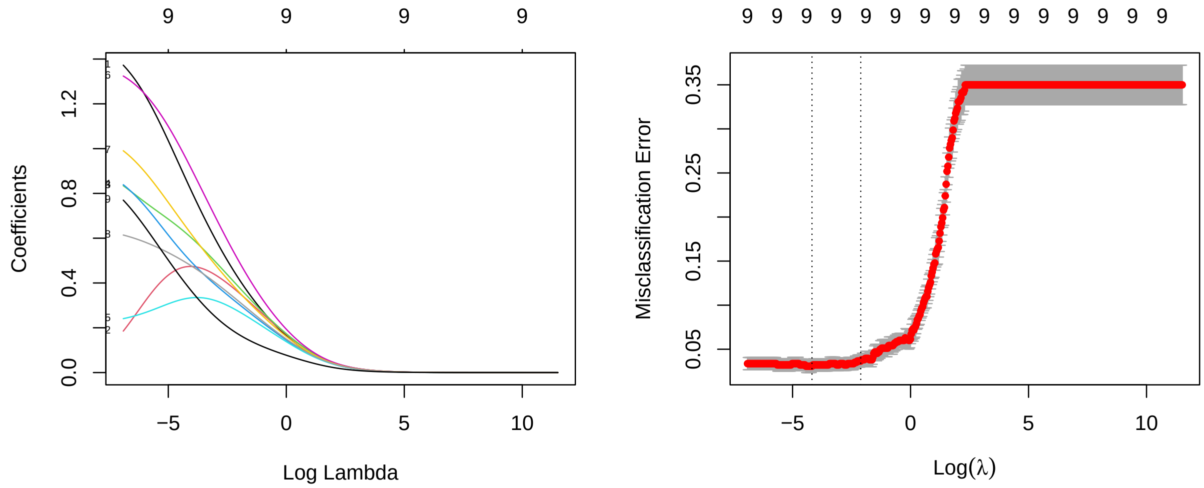
```
##       Predicted
## Actual   0   1
##      0 434  10
##      1  11 228
```

Regularized Form of Logistic Regression (Ridge and LASSO)

The advantage of regularization methods is that they used in order to minimize the coefficients of the variables towards to 0 or even equal to 0. For this analysis both of Ridge and LASSO regularization methods will be used.

Ridge Regression

In order to perform ridge regression we need to select a grid of values for the tuning parameter lambda and to fit the ridge regression model for every value of it. Primarily, we are going to select the best value of lambda based on the cross-validation with the same folds that we used for subset selection. Subsequently, log lambda against misclassification error will be plotted and the lowest error will be selected. We are going to use the plots discussed in order to see how the coefficients of the predictors behave.



The left plot clearly illustrates that the coefficients of the predictor variables progressively shrink towards zero as lambda approaches 5. Conversely, the second plot indicates that the minimum lambda value falls between the two dotted vertical lines. Consequently, identifying the precise minimum lambda value for cross-validation is essential.

```
## The optimal value for the tuning parameter (lambda) in Ridge regression is: 0.01535935
```

Below are the **coefficients** of Ridge Regression model:

```
## 10 x 1 sparse Matrix of class "dgCMatrix"
##                     s1
## (Intercept)     -0.974
## Cl.thickness     0.846
## Cell.size        0.473
## Cell.shape       0.616
## Marg.adhesion    0.511
## Epith.c.size     0.332
## Bare.nuclei      0.941
## Bl.cromatin      0.640
## Normal.nucleoli  0.486
## Mitoses          0.385
```

Now check the coefficients of a normal glm fit:

```
# Fit a logistic regression model using all predictor variables
glm_fit <- glm(y ~ ., data = Breast_Cancer_Final, family = binomial)
```

```
##     (Intercept)    Cl.thickness       Cell.size      Cell.shape   Marg.adhesion 
##          -1.094           1.509          -0.019           0.964           0.947 
##     Epith.c.size     Bare.nuclei     Bl.cromatin Normal.nucleoli         Mitoses 
##           0.215           1.396           1.095           0.650           0.927
```

The observed contraction of coefficients towards zero indicates the successful functioning of our model. Nevertheless, it's noteworthy that certain coefficients, such as those associated with "Epith.c.size" and "Mitoses," have experienced an increase instead.

And now we can print the **test error** value of Ridge Regression:

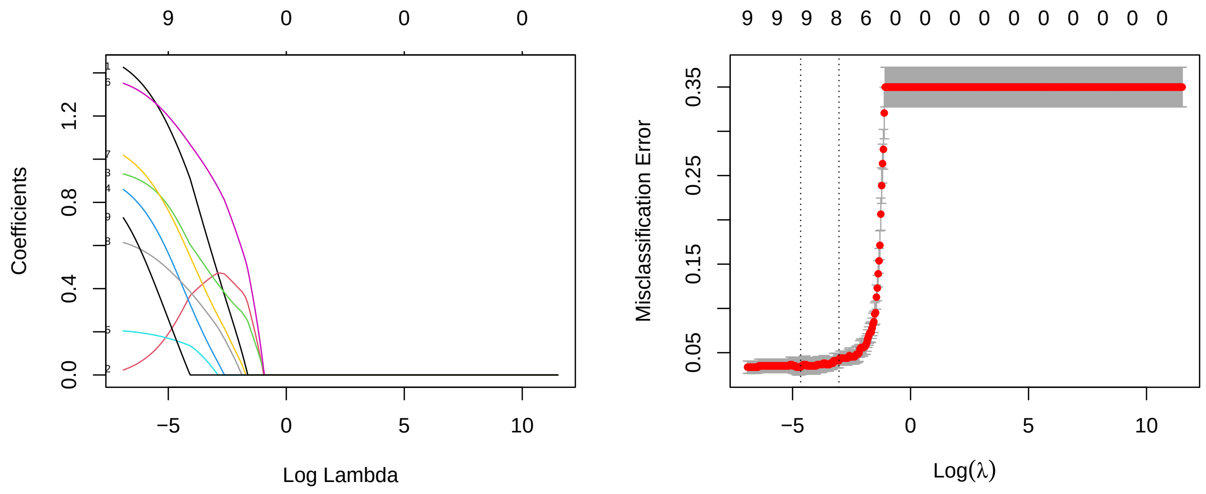
```
## Lowest error value for Ridge Regression: 0.03074671
```

And finally the **confusion matrix** of Ridge Regression:

```
##       Predicted
## Actual   0   1
##      0 434  10
##      1  10 229
```

LASSO Regression

The Lasso (Least Absolute Shrinkage and Selection Operator) is a regularization technique similar to ridge regression, but with a key difference. In addition to the sum of squared coefficients penalty term, the Lasso adds a penalty term proportional to the absolute values of the coefficients. We are going to use the same methodology as with the Ridge Regression.



The left plot reveals a distinctive pattern where certain predictor coefficients drop out earlier than others. Notably, "Mitoses" is the first to drop out, followed by "Epith.c.size," "Marg.adhesion," "Normal.nucleoli," “Bl.cromatin,” and “Cl.thickness.” The last trio of predictor coefficients, namely “Cell.shape,” “Cell.size,” and “Bare.nuclei”, appear to drop out simultaneously. As observed in the right plot, the minimum lambda value lies between the two dotted vertical lines.

```
## The optimal value for the tuning parameter (lambda) in LASSO regression is: 0.009505083
```

Below are the **coefficients** of LASSO Regression:

```
## 10 x 1 sparse Matrix of class "dgCMatrix"
##                     s1
## (Intercept)     -1.056
## Cl.thickness     1.071
## Cell.size        0.251
## Cell.shape       0.724
## Marg.adhesion    0.478
## Epith.c.size     0.159
## Bare.nuclei      1.154
## Bl.cromatin      0.688
## Normal.nucleoli  0.453
## Mitoses          0.161
```

And now we can print the **test error** value of LASSO Regression:

```
## Lowest error value for LASSO Regression: 0.03367496
```

Finally the **confusion matrix** of LASSO Regression:

```
##       Predicted
## Actual   0   1
##      0 435   9
##      1  11 228
```

Discriminant Analyis Method

Discriminant analysis serves as a powerful tool for classification, particularly in the context of this project, where the objective is to identify the optimal classifier for the variable “Class.” In linear discriminant analysis (LDA), the key assumption is that the classes have a common covariance matrix, meaning that the variance of each predictor variable is the same across all classes. On the other hand, quadratic discriminant analysis (QDA) relaxes the assumption of a common covariance matrix, allowing each class to have its own covariance matrix. This increased flexibility enables QDA to capture more intricate relationships between predictor variables within each class, accommodating scenarios where the variability within classes differs significantly.

Linear Discriminant Analysis

The objective here is to perform feature selection by exploring all possible subsets of predictor variables. The subset that exhibits the lowest test error through cross-validation will be chosen as the optimal set of features.

```r
# Create a vector that contain the names of all the predictor variables
colnames_vector <- colnames(Breast_Cancer_Final)[1:9]

# Now compute all the possible subsets of the variables
subset_combinations = unlist(lapply(1:length(colnames_vector),  combinat::combn,
                                    x = colnames_vector, simplify = FALSE),
                                    recursive = FALSE)
```

```
## The subset with the lowest cross-validation error for LDA is at index: 413
```

Now that we know the index of the subset with the lowest test error based on cross validation it is time to extract the predictor variables that this subset uses to achieve that.

```
## [[1]]
## [1] "Cl.thickness" "Cell.size"    "Epith.c.size" "Bare.nuclei"  "Bl.cromatin"
## [6] "Mitoses"
```

By fitting the lda model we can now see the **group means** of the model:

```r
# Perform LDA
lda_fit <- lda(y ~ Cl.thickness + Cell.size + Epith.c.size + Bare.nuclei + Bl.cromatin +
                 Mitoses, data = Breast_Cancer_Final)
```

```
## Call:
## lda(y ~ Cl.thickness + Cell.size + Epith.c.size + Bare.nuclei +
##     Bl.cromatin + Mitoses, data = Breast_Cancer_Final)
##
## Prior probabilities of groups:
##         0         1
## 0.6500732 0.3499268
##
## Group means:
##   Cl.thickness  Cell.size Epith.c.size Bare.nuclei Bl.cromatin    Mitoses
## 0   -0.5240440 -0.6017657   -0.5065718  -0.6031546   -0.555890 -0.3104483
## 1    0.9735377  1.1179245    0.9410791   1.1205047    1.032699  0.5767324
##
## Coefficients of linear discriminants:
##                     LD1
## Cl.thickness 0.53617253
## Cell.size    0.69643118
## Epith.c.size 0.18748211
## Bare.nuclei  1.01729128
## Bl.cromatin  0.38227754
## Mitoses      0.05956621
```

As it can be seen from the group means of the above summary the “benign” class have negative means compared to the “malignant” ones which has positive means. This indicates that the tumors labeled as malignant are more likely to have higher values in the variables “Cl.thickness”, “Cell.size”, “Epith.c.size”, “Bare.nuclei”, “Bl.cromatin”, and “Mitoses”.

Next we print the **test error** value of the LDA model:

```
## The cross-validation error for the best LDA subset is: 0.03660322
```

Finally below is the **confusion matrix** of LDA:

```
##       Predicted
## Actual   0   1
##      0 437   7
##      1  18 221
```

Quadratic Discriminant Analysis

```
## The subset with the lowest cross-validation error for LDA is at index: 166
```

Just like we did for LDA now that we know the index of the subset with the lowest test error based on cross validation it is time to extract the predictor variables that this subset uses to achieve that.

```
## [[1]]
## [1] "Cl.thickness"  "Marg.adhesion" "Epith.c.size"  "Bare.nuclei"
```

By fitting the QDA model we can now see the **group means** of the model:

```
# Perform QDA
qda_fit = qda(y~ Cl.thickness+ Marg.adhesion + Epith.c.size  + Bare.nuclei,
               data = Breast_Cancer_Final)
```

```
## Call:
## qda(y ~ Cl.thickness + Marg.adhesion + Epith.c.size + Bare.nuclei,
##     data = Breast_Cancer_Final)
##
## Prior probabilities of groups:
##         0         1
## 0.6500732 0.3499268
##
## Group means:
##   Cl.thickness Marg.adhesion Epith.c.size Bare.nuclei
## 0   -0.5240440    -0.5178153   -0.5065718  -0.6031546
## 1    0.9735377     0.9619665    0.9410791   1.1205047
```

Once again just like in the LDA model the mean of the selected variables for "benign" class are all negative and for "malignant" class are positive. Again, this indicates that the cancer tumors labeled as benign are more likely to have smaller values in the variables "Cl.thickness", "Marg.adhesion", "Epith.c.size", and "Bare.nuclei".

Next we print the **test error** value of the QDA model:

```
## The cross-validation error for the best QDA subset is: 0.03513909
```

Finally below is the **confusion matrix** of QDA:

```
##       Predicted
## Actual   0   1
##      0 427  17
##      1   8 231
```

Models Comparisson

Having all the above models is now time to compare all of them in order to check which one is the best model based on the lowest test error provided by implementing cross validation.

In the table below are all the test errors from all the implemented models:

```
##             BSS-4  Ridge  LASSO    LDA    QDA
## Test Error 0.0580 0.0307 0.0337 0.0366 0.0351
```

The cross-validation results indicate notable differences in test errors among the models. The highest observed test error originates from the best subset selection, registering at 0.0580, whereas the lowest is associated with the LASSO model, showcasing a notably lower value of 0.0307. It's noteworthy that both Ridge and LASSO exhibit comparable test errors, demonstrating close performance metrics. Similarly, the LDA and QDA models align closely, revealing test errors of 0.0366 and 0.0351, respectively. These findings underscore the nuanced distinctions in predictive accuracy across the implemented models.

Comclusion

After a comprehensive analysis of the various metrics and comparisons among the models, my definitive recommendation is to adopt the Ridge regression model. This decision is grounded in its standout performance, boasting the lowest test error value derived from cross-validation. Notably, the Ridge model incorporates all nine cytological characteristics, suggesting that considering each of these variables is integral to achieving the model's minimal test error.

This preference for the Ridge regression model underscores the significance of every characteristic in the classification task. In contrast, alternative models utilizing fewer variables exhibit higher test errors, reinforcing the crucial role played by each cytological feature in discerning between "benign" and "malignant" tumor classes. The Ridge regression model emerges as the optimal choice, epitomizing the importance of a comprehensive approach in achieving superior predictive accuracy.