ISDS Assignment 2

December 2022

1 Introduction

Breast cancer has already been one of the biggest threats to women's health and it is the most common malignant tumour in women which is the result of abnormal cell growth in the breast tissue. A tumour does not mean cancer, as there are benign and malignant tumours. The data set used for this analysis was from the Breast Cancer data set in R package "mlbench", created by Dr. Wolberg from Wisconsin. The data in this data set used the FNAC method to characterize samples collected from the breast tissue of 699 women in Wisconsin. The data set recorded nine cytological features of breast tissue cells with a scale of one to ten for each tissue sample. The aim of this analysis was to observe which features were most helpful in predicting malignant or benign tumours, and to view patterns of general trends that might contribute to our model selection. The objective was to classify whether the breast cancer was benign or malignant. To achieve this goal, several classification methods like logistic regression, linear discriminant analysis and quadratic discriminant analysis would be used to fit a function that enables classifying the new input data.

2 Data Processing

2.1 Data Details

For the analysis objective, we assumed that the patients can be seen as a random sample. The data set is part of the `mlbench` package. The package should be installed before processing, then the data can be loaded into R.

```
# install.packages("mlbench") -- executed only once, so it is commented
library(mlbench)  # load package mlbench
data(BreastCancer)  # load data BreastCancer
dim(BreastCancer)  # Inspect the size
```

```
699  11
```

From the result above, we can observe there were 699 rows and 11 columns in this data set.

```
head(BreastCancer)  # Print the first few rows
```

Please see Appendix

The first few rows showed the data structure of the data set, the columns contains Id, Cl.thickness, Cell.size, Cell.shape, Marg.adhesion, Epith.c.size, Bare.nuclei, Bl.cromatin, Normal.nucleoli, Mitoses and Class. The Class variable would be the response variable.

```
summary(BreastCancer)
```

Please see Appendix

2.2 Cleaning the data

The data set summary showed that the data set contains 16 `NA`, which means missing observations. We will examine the proportion of missing values in the overall data when considering what to do with these observations with missing values. There are several ways to process the missing values. If the size of missing

values is small, they will have little influence on the overall data, and it is appropriate to remove the data with missing values. According to summary of `BreastCancer` data set, there are 16 missing values out from 699 rows, which is a small size. Therefore, the data with missing values will be removed directly

```
BC_nna <- na.omit(BreastCancer)  # Remove na value, create a new data set
sum(is.na(BC_nna)) # Check if the na value still exist
```

```
0
```

```
dim(BC_nna)
```

```
683  11
```

`na.omit` function was applied to remove the missing values and the results showed that there is 0 `NA` and 683 rows in the data set, indicating the missing data were removed correctly.

As we know, the data in `BreastCancer` were encoded as factors, while the analysis required treating them as quantitative variables. The data should be converted to quantitative variables. At the same time, the type of data should be converted to `dataframe` and some columns should be removed as they were not involved in the calculation. For the goals, the the type of data were converted to `dataframe` and a method in `data.frame` was applied to remove the first column `Id`. Also, `benign` is represented internally as a `1` and `malignant` as a `2`. They should be better adopted to standard 0/1 numerical labels. To achieve this, a `-1` was applied.

```
BC_nna = data.frame(BC_nna[,-c(1,11)], Class = as.integer(BC_nna$Class)-1)
BC_nna[1:9] <- lapply(BC_nna[1:9], FUN = function(y){as.numeric(y)})
head(BC_nna)
```

Please see Appendix

3 Exploratory Data Analysis

Now the data set was clean and the response variable were clear, we can perform some exploratory data analysis.

```
table(BC_nna$Class)
```

```
 0   1
444 239
```

As the result, there were more benign cells than the malignant ones. That was in line with our common sense. The relationships between variables were important for observing the features of cells, so the package `corrplot` was applied to show the relationships graphically

```
library(corrplot)
bcp <- cor(BC_nna[,1:9])
corrplot.mixed(bcp)
```

Figure 1 indicated that each pair of variables had relatively strong positive correlation, which reminded us that the multicollinearity should be considered. While the figure showed the the relationship between each of the variables, the relationship between these variables and the response variable, `Class`, were not reflected. Producing a pairs plot of the predictors and colouring the points according to whether the cell was benign or malignant was a good idea.

Figure 1: Relationships between variables

Please see Appendix

```
pairs(BC_nna[,1:9], col = BC_nna[,10]+1)
```

Figure 2: Scatter plot for the Breast Cancer data

Please see Appendix

On the basis of this scatter plot above, there was only one obvious positive linear relationship between `Cell.size` and `Cell.shape`, which indicated the cell size was getting bigger as the cell shape increased. Apart from this pattern, there were relatively clear distribution of the benign cells and malignant cells in each relationship plot. Based on the distribution, it might be easy to classify the type of cells. There were no other obvious patterns because of the limited data size and the scale of one to ten measurement.

4 Modelling

The first step in the analysis is to build assumptions. If we label all these nine variables as x_1,...,x_9 and then we perform nine hypothesis tests:

$$H_0 : \beta_j = 0 \text{ versus } H_1 : \beta_j \neq 0$$

When modelling using different approaches, the index of β would be different. Three approaches would be used to model in this analysis. The first approach would be a full model using logistic regression, which contained all these nine predictors. In the full model, the effects of all the nine variables could presented. After getting the full model, the result would help find the useful predictors. These predictors contributed to select a best data subset. When it comes to best subset selection, both AIC and BIC approaches are good methods to determine which set of variables were the best to fit the data. Also, $k-$fold cross-validation would be applied in this analysis to select a good model and good predictors, and assess the model predictive performance. Finally, The Bayes classifier for linear discriminant analysis (LDA) or quadratic discriminant analysis (QDA) were used to build classifier. LDA is a method used as a linear classifier or dimensionality reduction before classification. While QDA is also a discriminant-based approach as LDA with a different function.

4.1 Logistic Regression

As the assumption above, the patients can be seen as a random sample which means we can assume the variables as normal distribution. At the same time, the `Class` contained `benign` and `malignant` which means $K = 2$, so that Y is binary.In the data set, there were nine predictors so that we could write an function:

$$\mu = \beta_0 + \beta_1 x_1 + ... + \beta_9 x_9$$

Given that Y is binary, we can modify the function with a link function:

$$\mu = \frac{e^(\beta_0 + \beta_1 x_1 + ... + \beta_9 x_9)}{1 + e^(\beta_0 + \beta_1 x_1 + ... + \beta_9 x_9}$$

And then we perform nine hypothesis tests:

$$H_0 : \beta_j = 0 \text{ versus } H_1 : \beta_j \neq 0$$

4.2 The Full Model

The analysis would be started by recording the size of rows and predictor variables.

```
(n = nrow(BC_nna))
(p = ncol(BC_nna) - 1)
```

Then we built a logistic regression model for `Class` with nine predictors, and then summarized the fit of the model:

```
logreg_fit = glm(Class ~ ., data = BC_nna, family = "binomial")
summary(logreg_fit)
```

```
    Call:
glm(formula = Class ~ ., family = "binomial", data = BC_nna)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.4855  -0.1152  -0.0619   0.0222   2.4702

Coefficients:
                 Estimate Std. Error z value Pr(>|z|)
(Intercept)     -10.110096   1.173774  -8.613  < 2e-16 ***
Cl.thickness      0.535256   0.141938   3.771 0.000163 ***
Cell.size        -0.005943   0.209158  -0.028 0.977332
Cell.shape        0.322136   0.230644   1.397 0.162510
Marg.adhesion     0.330694   0.123462   2.679 0.007395 **
Epith.c.size      0.096797   0.156568   0.618 0.536415
Bare.nuclei       0.383015   0.093865   4.080 4.49e-05 ***
Bl.cromatin       0.447401   0.171392   2.610 0.009044 **
Normal.nucleoli   0.213074   0.112894   1.887 0.059109 .
Mitoses           0.538551   0.325615   1.654 0.098138 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 884.35  on 682  degrees of freedom
Residual deviance: 102.90  on 673  degrees of freedom
AIC: 122.9

Number of Fisher Scoring iterations: 8
```

When observing the $p-$value column, `Cl.thickness`, `Marg.adhesion`, `are.nuclei`, and `Bl.cromatin` had a coefficient which was significantly different from zero at the 5% level. For the hypothesis tests, we would reject the null hypothesis at the 5% level for $j = 1, 4, 6, 7$. This indicated that all the other variables in the model contributed little to forecasting the cell is benign or malignant. We can calculate the fitted values and compare to with actual values. Also, we should calculate the training error for this model to assess model performance.

```
phat = predict(logreg_fit, BC_nna, type = "response")
yhat = as.numeric(ifelse(phat > 0.5, 1, 0))
(training_error = 1- mean(yhat == BC_nna$Class))
```

```
0.03074671
```

The training error is around 3%, which seems like a tiny error. Given that the training error was calculated by the same data to train and test, the performance of this model was overestimated. More than half of the variables contributed little to fit the model, a further analysis was needed.

4.3 Best Subset Selection

The full model suggested only four predictor variables had a coefficient, significantly different from zero at the 5% level. That indicated the full model contains too many unnecessary variables. There may not be many parameters that really matter in a model, and considering all of them in action can trigger over fitting. And with fewer parameters the explanatory power of the model becomes stronger. Considering this, subset selection methods should be applied to help fit a model with fewer predictors and better performance. Increasing the number of free parameters improves the performance of the fit, and AIC encourages goodness of fit but tries to avoid over fitting. While under incomplete information, BIC estimates the partially unknown condition with subjective probabilities, then corrects the occurrence probabilities using Bayesian formulas, and finally makes an optimal decision using the expected value and the corrected probabilities.

```
bss_fit_AIC = bestglm(BC_nna, family = binomial, IC = "AIC" )
bss_fit_BIC = bestglm(BC_nna, family = binomial, IC = "BIC" )
bss_fit_AIC$Subsets
bss_fit_BIC$Subsets
(best_AIC = bss_fit_AIC$ModelReport$Bestk)
(best_BIC = bss_fit_BIC$ModelReport$Bestk)
```

Please see Appendix

```
(best_AIC = bss_fit_AIC$ModelReport$Bestk)
(best_BIC = bss_fit_BIC$ModelReport$Bestk)
```

```
7
5
```

The models were the same in each case but the last column, as different information criterion were applied. At the same time, the method in `bss_fit_AIC` and `bss_fit_BIC` can identify the best subset number. The AIC suggested 7 variables as best subset predictors, while the BIC suggested 5 variables.

4.4 *k*-fold Cross Validation

As we mentioned above, k-fold cross validation is a good way to estimate the test error for a logistic-regression-based classifier. In this analysis, 10-fold cross-validation would be considered.

```
(test_error = general_cv(BC_nna[,1:p], BC_nna[ ,p+1], fold_index,
   logistic_reg_fold_error))
```

```
0.03221083
```

As the training error tended to give an optimistic estimate of the performance of the model, the test error was larger than the training error which made sense. Apart from that, the k-fold cross-validation would work in best subset selection.

```
cv_errors = logistic_reg_bss_cv(BC_nna[,1:p], BC_nna[,p+1], fold_index)
(best_cv = which.min(cv_errors) - 1)
```

```
6
```

The cross validation indicated the model should have 6 predictors, while the AIC and BIC indicated 7 and 5 variables. A plot would be a good way to show the number of predictors as different criteria was used.

```
points(best_cv, cv_errors[best_cv+1], col="red", pch=16)
```

Figure 3: Best subset selection for the Breast Cancer Data

Please see Appendix

As different criteria suggested 5, 6, and 7 predictors for the model, 6-predictor would be a good compromise. The variables from the best-fitting 1-predictor model would be extracted with the `bestglm` function.

```
pstar = 6
bss_fit_AIC$Subsets[pstar+1,]
```

```
   Intercept Cl.thickness Cell.size Cell.shape Marg.adhesion Epith.c.size
6   TRUE         TRUE     FALSE       TRUE          TRUE        FALSE
  Bare.nuclei Bl.cromatin Normal.nucleoli Mitoses logLikelihood      AIC
6   TRUE        TRUE         TRUE   FALSE     -53.57186 119.1437
```

4.4.1 Logistic Regression with best subset selection;

```
BC_nna_bss = BC_nna[,c(indices, p+1)]
logregl_fit = glm(Class ~., data = BC_nna_bss, family = "binomial")
summary(logregl_fit)
```

```
    Call:
glm(formula = Class ~ ., family = "binomial", data = BC_nna_bss)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.5201  -0.1186  -0.0570   0.0250   2.4055

Coefficients:
                Estimate Std. Error z value Pr(>|z|)
(Intercept)     -9.76708    1.08506  -9.001  < 2e-16 ***
Cl.thickness     0.62253    0.13712   4.540 5.62e-06 ***
Cell.shape       0.34951    0.16503   2.118  0.03419 *
Marg.adhesion    0.33753    0.11561   2.920  0.00350 **
Bare.nuclei      0.37855    0.09381   4.035 5.45e-05 ***
Bl.cromatin      0.47134    0.16612   2.837  0.00455 **
Normal.nucleoli  0.24317    0.10855   2.240  0.02509 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 884.35  on 682  degrees of freedom
Residual deviance: 107.14  on 676  degrees of freedom
AIC: 121.14

Number of Fisher Scoring iterations: 8
```

In the summary, the coefficients in the 6-predictor model were different from zero significantly. The coefficients ranged from around 0.24317 to 0.62253, which indicated that as the numbers of these cytological features increased, the probability to be malignant would be higher. As the calculation above, the test error should be estimated as follows:

```
(test_error = general_cv(BC_nna_bss[,1:pstar], BC_nna_bss[, pstar+1],
    fold_index, logistic_reg_fold_error))
```

```
    0.03221083
```

The test error for both full model and 6-predictor model were almost equal. Relying only on the test error, the performance of the two models was nearly identical.

4.5 Linear Discriminant Analysis

LDA classifiers assume that the conditional distributions for the predictor variables are multivariate normal with a group-specific mean vector and a common covariance matrix. In this analysis, the data were assumed to be normal distribution with a common covariance matrix. LDA classifier could be used to analysis the data set.

```
(lda_fit = lda(Class ~., data = BC_nna_bss))
```

```
    Call:
lda(Class ~ ., data = BC_nna_bss)

Prior probabilities of groups:
        0         1
0.6500732 0.3499268

Group means:
  Cl.thickness Cell.shape Marg.adhesion Bare.nuclei Bl.cromatin Normal.nucleoli
0     2.963964   1.414414      1.346847    1.346847    2.083333        1.261261
1     7.188285   6.560669      5.585774    7.627615    5.974895        5.857741

Coefficients of linear discriminants:
                     LD1
Cl.thickness    0.19055042
Cell.shape      0.19122091
Marg.adhesion   0.06655343
Bare.nuclei     0.25782925
Bl.cromatin     0.13282062
Normal.nucleoli 0.12236079
```

Based on the result above, the group means for 0(benign) group ranged from around 1.261261 to 2.963964, while the group means for 1(malignant) group were much greater, ranging from around 5.585774 to 7.627615. The coefficients of linear discriminants for each predictor ranged from around 0.06655343 to 0.25782925, which was positive. It seemed to suggest that as the size of each predictor increased, it was more possible to be malignant.

```
    (confusion = table(Observed = BC_nna_bss$Class, Predicted = yhat))
```

```
          Predicted
Observed   0   1
       0 436   8
       1  19 220
```

```
    (test_error = general_cv(BC_nna_bss[,1:pstar], BC_nna_bss[,pstar+1],
       fold_index, lda_fold_error))
```

```
    0.04099561
```

Though the test error was very small, it was greater than the test error in logistic regression models with the full data set and best subset selection. From this point of view, LDA model performed somewhat worse than the logistic regression model.

4.6 Quadratic discriminant analysis

The difference between QDA and LDA is not that great, except that QDA allows different groups to have different covariance matrices. As mentioned above, the data were assumed to be normal distribution with a common covariance matrix. Therefore, QDA could be applied to build the classifier.

```
    (qda_fit = qda(Class ~ ., data=BC_nna_bss))
```

```
    Call:
qda(Class ~ ., data = BC_nna_bss)

Prior probabilities of groups:
        0         1
0.6500732 0.3499268
```

```
Group means:
  Cl.thickness Cell.shape Marg.adhesion Bare.nuclei Bl.cromatin Normal.nucleoli
0     2.963964   1.414414      1.346847    1.346847    2.083333        1.261261
1     7.188285   6.560669      5.585774    7.627615    5.974895        5.857741
```

As the result showed, the group means of 0(benign) group ranging from 1.261261 to 2.963964 were much smaller than 1(malignant) group ranging from 5.585774 to 7.627615. That also suggested that the numbers of malignant group tended to be larger than the benign group.

```
(confusion = table(Observed=BC_nna_bss$Class, Predicted=yhat))
```

```
          Predicted
Observed   0   1
       0 420  24
       1   7 232
```

```
(test_error = general_cv(BC_nna_bss[,1:pstar], BC_nna_bss[,pstar+1],
   fold_index, qda_fold_error))
```

```
0.04685212
```

In comparison, test error of QDA was larger than either of the other two models, and in particular much larger than the logistic regression model.

5 Conclusion

The test error for these four models were around 0.03221083, 0.03221083, 0.04099561 and 0.04685212. Test error was a simple way to assess the performance of models. If only the test error was considered, the logistic regression model performs best. It is well known that the model comparison is complex as many parameters can be use to assess the performance. For example, some parameters associated to the confusion matrix like accuracy, precision recall and F-score are helpful in assessing the performance.

In this analysis, both LDA and QDA models got confusion matrix. The confusion matrix suggested that LDA model got more correct response numbers than QDA model. This may suggest LDA model was better than QDA model. Besides, When there is a high degree of differentiation between categories, the parameter estimates of the logistic regression model are not robust enough, whereas this is not a problem with linear discriminant analysis. If the sample size is relatively small and the predictor variables in each category of response classification are approximately normally distributed, then linear discriminant analysis is more stable than logistic regression. Considering the pair plots above showed, degree of differentiation between categories seemed to be high. In addition to the small size of the sample, the LDA model seemed to be a better model.

When comparing the logistic regression model with full data set and best subset selection, the test error were almost equal. If we applied an anova function in R to test the performance between two models, we found the performance did not have a significant difference. But the model with best subset selection contains less predictors which means the explanatory power of the model becomes stronger. Therefore, the logistic regression model with best subset selection would be the better one.

Since logistic regression is very popular in classification questions, especially K = 2, and the test error in this analysis was the smallest, the logistic regression with best subset selection would be the best for this question.

Furthermore, the analysis suggested that six predictors (Cl.thickness, Cell.shape, Marg.adhesion, Bare.nuclei, Bl.cromatin, Normal.nucleoli) were the better variables to determine the cell to be benign or malignant. At the same time, the result suggested that these six predictors had a positive correlation with the class of cell which means if the size of these six variables got bigger, the probability to be malignant increased as well.

A Appendix

A.1 head(BreastCancer)

```
head(BreastCancer)  # Print the first few rows
```

```
          Id Cl.thickness Cell.size Cell.shape Marg.adhesion Epith.c.size Bare.nuclei
1 1000025            5         1          1             1            2           1
2 1002945            5         4          4             5            7          10
3 1015425            3         1          1             1            2           2
4 1016277            6         8          8             1            3           4
5 1017023            4         1          1             3            2           1
6 1017122            8        10         10             8            7          10
  Bl.cromatin Normal.nucleoli Mitoses     Class
1           3               1       1    benign
2           3               2       1    benign
3           3               1       1    benign
4           3               7       1    benign
5           3               1       1    benign
6           9               7       1 malignant
```

A.2 summary(BreastCancer)

```
summary(BreastCancer)
```

```
    Id             Cl.thickness   Cell.size     Cell.shape  Marg.adhesion  Epith.c.size
 Length:699         1      :145   1      :384   1      :353   1      :407   2      :386
 Class :character   5      :130   10     : 67   2      : 59   2      : 58   3      : 72
 Mode  :character   3      :108   3      : 52   10     : 58   3      : 58   4      : 48
                    4      : 80   2      : 45   3      : 56   10     : 55   1      : 47
                    10     : 69   4      : 40   4      : 44   4      : 33   6      : 41
                    2      : 50   5      : 30   5      : 34   8      : 25   5      : 39
                    (Other):117   (Other): 81   (Other): 95   (Other): 63   (Other): 66
 Bare.nuclei   Bl.cromatin  Normal.nucleoli    Mitoses          Class
 1      :402   2      :166   1      :443     1      :579   benign   :458
 10     :132   3      :165   10     : 61     2      : 35   malignant:241
 2      : 30   1      :152   3      : 44     3      : 33
 5      : 30   7      : 73   2      : 36     10     : 14
 3      : 28   4      : 40   8      : 24     4      : 12
 (Other): 61   5      : 34   6      : 22     7      :  9
 NA's   : 16   (Other): 69   (Other): 69     (Other): 17
```

A.3 head(BC_nna)

```
BC_nna = data.frame(BC_nna[,-c(1,11)], Class = as.integer(BC_nna$Class)-1)
BC_nna[1:9] <- lapply(BC_nna[1:9], FUN = function(y){as.numeric(y)})
head(BC_nna)
```

```
   Cl.thickness Cell.size Cell.shape Marg.adhesion Epith.c.size Bare.nuclei Bl.cromatin
1             5         1          1             1            2           1           3
2             5         4          4             5            7          10           3
3             3         1          1             1            2           2           3
4             6         8          8             1            3           4           3
5             4         1          1             3            2           1           3
6             8        10         10             8            7          10           9
```

```
  Normal.nucleoli Mitoses Class
1               1       1     0
2               2       1     0
3               1       1     0
4               7       1     0
5               1       1     0
6               7       1     1
```

A.4 Figure 1

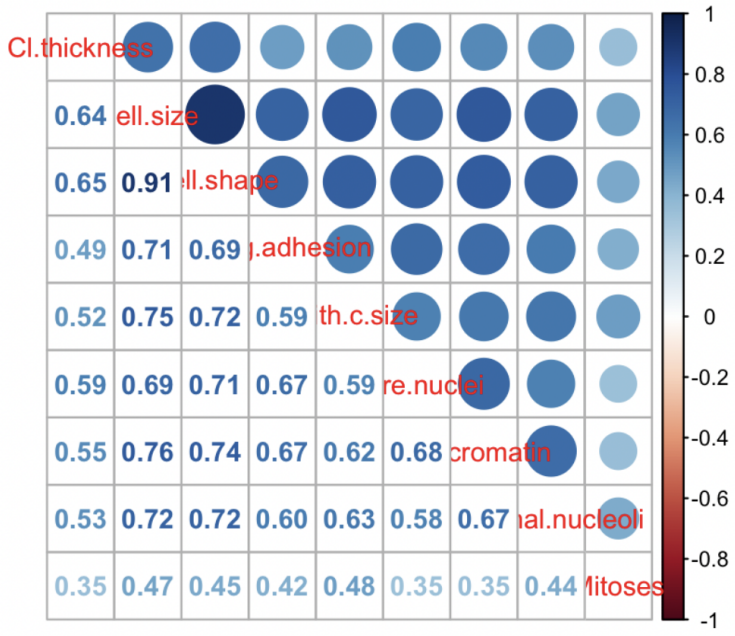

Figure 1: Relationships between variables

A.5 Figure 2

Figure 2: Scatter plot for the Breast Cancer data

A.6 bss_fit_AIC$Subsets

```
    bss_fit_AIC = bestglm(BC_nna, family = binomial, IC = "AIC" )
    bss_fit_BIC = bestglm(BC_nna, family = binomial, IC = "BIC" )
    bss_fit_AIC$Subsets
    bss_fit_BIC$Subsets
    (best_AIC = bss_fit_AIC$ModelReport$Bestk)
    (best_BIC = bss_fit_BIC$ModelReport$Bestk)
```

```
      Intercept Cl.thickness Cell.size Cell.shape Marg.adhesion Epith.c.size Bare.nuclei
0        TRUE        FALSE     FALSE      FALSE         FALSE        FALSE       FALSE
1        TRUE        FALSE      TRUE      FALSE         FALSE        FALSE       FALSE
2        TRUE        FALSE      TRUE      FALSE         FALSE        FALSE        TRUE
3        TRUE         TRUE      TRUE      FALSE         FALSE        FALSE        TRUE
4        TRUE         TRUE     FALSE       TRUE         FALSE        FALSE        TRUE
5        TRUE         TRUE     FALSE      FALSE          TRUE        FALSE        TRUE
6        TRUE         TRUE     FALSE       TRUE          TRUE        FALSE        TRUE
7*       TRUE         TRUE     FALSE       TRUE          TRUE        FALSE        TRUE
8        TRUE         TRUE     FALSE       TRUE          TRUE         TRUE        TRUE
9        TRUE         TRUE      TRUE       TRUE          TRUE         TRUE        TRUE
   Bl.cromatin Normal.nucleoli Mitoses logLikelihood      AIC
0    FALSE     FALSE   FALSE    -442.17509 884.3502
1    FALSE     FALSE   FALSE    -127.37980 256.7596
2    FALSE     FALSE   FALSE     -83.15598 170.3120
3    FALSE     FALSE   FALSE     -67.77778 141.5556
```

```
4     TRUE       FALSE    FALSE      -61.37155 130.7431
5     TRUE        TRUE    FALSE      -56.13177 122.2635
6     TRUE        TRUE    FALSE      -53.57186 119.1437
7*    TRUE        TRUE     TRUE      -51.63998 117.2800
8     TRUE        TRUE     TRUE      -51.45031 118.9006
9     TRUE        TRUE     TRUE      -51.44991 120.8998
```

A.7 Figure 3

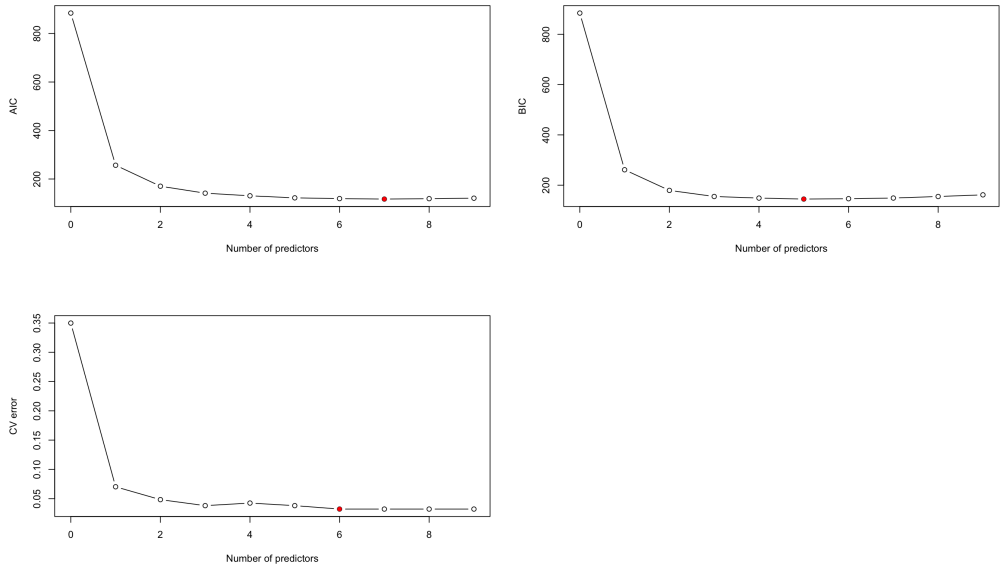

Figure 3: Best subset selection for the Breast Cancer Data

A.8 R Code

```
# install.packages("mlbench") -- executed only once, so it is commented
# load package mlbench
library(mlbench)
# load data BreastCancer
data(BreastCancer)

# Inspect the size
dim(BreastCancer)

# Print the first few rows
head(BreastCancer)

summary(BreastCancer)

# 2.2.1 Cleaning the Data
# This data set contains some missing observations
# install.packages("mice") -- executed only once, so it is commented
# library(mice)
# md.pattern(BreastCancer)
```

```
# According to the assignment notes, we can print 24th row of Breast Cancer
    data and note there is a NA in the Bare.nuclei column:
# is.na(BreastCancer[24,])
# Calculate the rows containing missing observations
# sum(is.na(BreastCancer))
# summary(BreastCancer) is enough for checking the missing observations


# Remove the na value and create a new data set
BC_nna <- na.omit(BreastCancer)
# Check if the na value still exist
sum(is.na(BC_nna))
dim(BC_nna)

# Check the type of data BC_nna
typeof(BC_nna)

# Convert list to dataframe
# Remove the first and last columns
# Create new column of 0s and 1s (labelling means 0:benign, 1:malignant)
BC_nna = data.frame(BC_nna[,-c(1,11)], Class = as.integer(BC_nna$Class)-1)
# Convert all observations to numeric for the logistic regression
    calculation
BC_nna[1:9] = lapply(BC_nna[1:9], FUN = function(y){as.numeric(y)})
# Print the first few rows
head(BC_nna)

# Compare the colnames before and after data cleaning to to make sure there
    are no problems in data cleaning
colnames(BC_nna)
colnames(BreastCancer)

# 2.2.2 Exploratory Data Analysis

table(BC_nna$Class)

# install.packages("reshape2")
# library(reshape2)
# library(ggplot2)
# bc_m = melt(BC_nna, id.vars = "Class")
# head(bc_m)
# ggplot(bc_m, aes(x=Class,y=value))+geom_boxplot()+facet_wrap(~variable,
    ncol = 3)

# See the relationships between variables
library(corrplot)
bcp <- cor(BC_nna[,1:9])
corrplot.mixed(bcp)


# Produce a pairs plot of the predictors according to the class was be
    benign or malignant
pairs(BC_nna[,1:9], col = BC_nna[,10]+1)

# 3 Logistic Regression
# 3.1 The Full Model
```

```
# Store n and p
(n = nrow(BC_nna))
(p = ncol(BC_nna) - 1)

# Fit a logistic regression model for Direction in terms of the predictors
logreg_fit = glm(Class ~ ., data = BC_nna, family = "binomial")
summary(logreg_fit)

# Compute the fitted values
phat = predict(logreg_fit, BC_nna, type = "response")
yhat = as.numeric(ifelse(phat > 0.5, 1, 0))
# Print first few elements:
head(yhat)

#  Compare with actual values:
head(BC_nna$Class)

# Compute the fitted values: did this above and they are stored in yhat
# Compute training error:
(training_error = 1- mean(yhat == BC_nna$Class))

# install.packages("bestglm")
# Load the bestglm package
library(bestglm)
head(BC_nna)

# Apply best subset selection
bss_fit_AIC = bestglm(BC_nna, family = binomial, IC = "AIC" )
bss_fit_BIC = bestglm(BC_nna, family = binomial, IC = "BIC" )

# Examine the results
bss_fit_AIC$Subsets
bss_fit_BIC$Subsets

# Identify best-fitting models
(best_AIC = bss_fit_AIC$ModelReport$Bestk)
(best_BIC = bss_fit_BIC$ModelReport$Bestk)

# Set the seed (say, at 10) to make the analysis reproducible
set.seed(10)
# Sample the fold-assignment index
nfolds = 10
fold_index = sample(nfolds, n, replace = TRUE)

# Print the first few fold-assignments
head(fold_index)

# Write a function to calculate the test error given a matrix of predictor
    variables X, a vector of response variables y and a vector of booleans
    test_data
logistic_reg_fold_error = function(X, y, test_data) {
  Xy = data.frame(X, y=y)
  if(ncol(Xy)>1) tmp_fit = glm(y ~ ., data=Xy[!test_data,], family="
      binomial")
  else tmp_fit = glm(y ~ 1, data=Xy[!test_data,,drop=FALSE], family="
      binomial")
```

```
  phat = predict(tmp_fit, Xy[test_data,,drop=FALSE], type="response")
  yhat = ifelse(phat > 0.5, 1, 0)
  yobs = y[test_data]
  test_error = 1 - mean(yobs == yhat)
  return(test_error)
}

# Write a function to calculate the test error given a particular split of
    the data into training and validation sets
general_cv = function(X, y, fold_ind, fold_error_function) {
  p = ncol(X)
  Xy = cbind(X, y=y)
  nfolds = max(fold_ind)
  if(!all.equal(sort(unique(fold_ind)), 1:nfolds)) stop("Invalid fold
      partition.")
  fold_errors = numeric(nfolds)
  for(fold in 1:nfolds) {
    fold_errors[fold] = fold_error_function(X, y, fold_ind==fold)
  }
  fold_sizes = numeric(nfolds)
  for(fold in 1:nfolds) fold_sizes[fold] = length(which(fold_ind==fold))
  test_error = weighted.mean(fold_errors, w=fold_sizes)
  return(test_error)
}

# calculate the test error of our full model
(test_error = general_cv(BC_nna[,1:p], BC_nna[ ,p+1], fold_index,
    logistic_reg_fold_error))


# Using k-fold Cross-Validation in Best Subset Selection
# Write a function to calculate the test error for best-fitting models
logistic_reg_bss_cv = function(X, y, fold_ind) {
  p = ncol(X)
  Xy = data.frame(X, y=y)
  X = as.matrix(X)
  nfolds = max(fold_ind)
  if(!all.equal(sort(unique(fold_ind)), 1:nfolds)) stop("Invalid fold
      partition.")
  fold_errors = matrix(NA, nfolds, p+1)
  for(fold in 1:nfolds) {
    tmp_fit = bestglm(Xy[fold_ind!=fold,], family=binomial, IC="AIC")
    best_models = as.matrix(tmp_fit$Subsets[,2:(1+p)])
    for(k in 1:(p+1)) {
      fold_errors[fold, k] = logistic_reg_fold_error(X[,best_models[k,]], y
          , fold_ind==fold)
    }
  }
  fold_sizes = numeric(nfolds)
  for(fold in 1:nfolds) fold_sizes[fold] = length(which(fold_ind==fold))
  test_errors = numeric(p+1)
  for(k in 1:(p+1)) {
    test_errors[k] = weighted.mean(fold_errors[,k], w=fold_sizes)
  }
  return(test_errors)
}
```

```
# Apply the cross-validation for best subset selection function
cv_errors = logistic_reg_bss_cv(BC_nna[,1:p], BC_nna[,p+1], fold_index)
# Identify the number of predictors in the model which minimises test error
(best_cv = which.min(cv_errors) - 1)

## Create multi-panel plotting device
par(mfrow=c(2, 2))

## Produce plots, highlighting optimal value of k
plot(0:p, bss_fit_AIC$Subsets$AIC, xlab="Number of predictors", ylab="AIC",
     type="b")
points(best_AIC, bss_fit_AIC$Subsets$AIC[best_AIC+1], col="red", pch=16)
plot(0:p, bss_fit_BIC$Subsets$BIC, xlab="Number of predictors", ylab="BIC",
     type="b")
points(best_BIC, bss_fit_BIC$Subsets$BIC[best_BIC+1], col="red", pch=16)
plot(0:p, cv_errors, xlab="Number of predictors", ylab="CV error", type="b
    ")
points(best_cv, cv_errors[best_cv+1], col="red", pch=16)

# AIC and BIC suggested the models with 7 and 5 predictors, respectively
# It seems like a good compromise to go with 6 predictors
pstar = 6

# Check which predictors are in the 6-predictor model
bss_fit_AIC$Subsets[pstar+1,]

# Construct a reduced data set containing only the 6 selected predictors
bss_fit_AIC$Subsets[pstar+1, 2:(p+1)]
(indices = which(bss_fit_AIC$Subsets[pstar+1, 2:(p+1)] == TRUE))

# Create best selection subset
BC_nna_bss = BC_nna[,c(indices, p+1)]
# Obtain regression coefficients for this model
logregl_fit = glm(Class ~., data = BC_nna_bss, family = "binomial")
summary(logregl_fit)

(test_error = general_cv(BC_nna_bss[,1:pstar], BC_nna_bss[, pstar+1],
    fold_index, logistic_reg_fold_error))

# 4. Linear Discriminant Analysis
# Load the MASS package
library(MASS)

# Apply LDA
(lda_fit = lda(Class ~., data = BC_nna_bss))

# Compute predicted values
lda_predict = predict(lda_fit, BC_nna_bss)
yhat = lda_predict$class

# Calculate confusion matrix
(confusion = table(Observed = BC_nna_bss$Class, Predicted = yhat))

# Calculate training error
1 - mean(BC_nna_bss$Class == yhat)
```

```
# Use 10-fold cross-validation to estimate the test error
# Write a function to calculate the test error for LDA
lda_fold_error = function(X, y, test_data) {
  Xy = data.frame(X, y=y)
  if(ncol(Xy)>1) tmp_fit = lda(y ~ ., data=Xy[!test_data,])
  tmp_predict = predict(tmp_fit, Xy[test_data,])
  yhat = tmp_predict$class
  yobs = y[test_data]
  test_error = 1 - mean(yobs == yhat)
  return(test_error)
}

# Calculate the test error
(test_error = general_cv(BC_nna_bss[,1:pstar], BC_nna_bss[,pstar+1],
    fold_index, lda_fold_error))

# Quadratic discriminant analysis
# # Apply QDA
(qda_fit = qda(Class ~ ., data=BC_nna_bss))

## Compute predicted values:
qda_predict = predict(qda_fit, BC_nna_bss)
yhat = qda_predict$class
## Calculate confusion matrix:
(confusion = table(Observed=BC_nna_bss$Class, Predicted=yhat))

# Calculate training error:
1 - mean(BC_nna_bss$Class == yhat)

# Write a function to caltulate the test error
qda_fold_error = function(X, y, test_data) {
  Xy = data.frame(X, y=y)
  if(ncol(Xy)>1) tmp_fit = qda(y ~ ., data=Xy[!test_data,])
  tmp_predict = predict(tmp_fit, Xy[test_data,])
  yhat = tmp_predict$class
  yobs = y[test_data]
  test_error = 1 - mean(yobs == yhat)
  return(test_error)
}

# Apply the general_cv function to calculate the test error
(test_error = general_cv(BC_nna_bss[,1:pstar], BC_nna_bss[,pstar+1],
    fold_index, qda_fold_error))


# comparison logistic regression with full set and best subset selection
anova(logregl_fit,logreg_fit, test="Chisq")
```