Assignment 2 STAT7040 Statistical Learning

Classification Problem

Nowadays, prostate tumours are histologically and clinically acknowledged as one of the most heterogeneous cancers. Hence, scientist was interested to study this phenomenon by studying gene expression differences between tumour and normal prostate samples using an analysis called microarray expression. The goal of the analysis is to classify patient samples into tumour or normal patient categories based on their lots of genes expression.

As the aim of this research is to classify sample patients into either tumour prostate or normal patients, the response variable for this case should be sample patients which consist of two categories in which tumour prostate and normal patient. Furthermore, as classification analysis is based on the gene expressions, the predictor variables used for this study are the 12600 *ith* observed genes. Subsequently, to achieve the goal of microarray expression analysis fitting a classification model to the observed dataset should be the most appropriate approach. Furthermore, as we deal with a classification model, we need both training and test dataset in a purpose of the model assessment later. Here, the training dataset contains 102 patient samples, 52 of which are prostate tumour samples and 50 of which are normal prostate. As for the test datasets, it contains 34 sample patients which consists of 25 tumour patients and 9 normal patient samples. To make the analysis interpretable, prostate tumour patients are labelled as "0" and the normal patients are coded as "1" in both the datasets.

Considering the description on the prostate cancer classification project, fitting a classification model which produce linear boundary seems to be the appropriate one to achieve the goal. The KNN model is not considered here because here we deal with large predictor variables to which we will encounter a complex difficulty in determining the neighbourhood to calculate the probability. Hence, either the logistic regression model or the linear discriminant analysis which might be the best fit to approach the interested dataset. However, since there is not any prior information on the distribution of the predictor variables, fitting the logistic linear model seems preferable compared to fit the LDA. For prostate cancer data, the two-categories setting with $p = j$ number of predictors, we let the $p_1(\mathbf{x})$ and $p_0(\mathbf{x}) = 1 - p_1(\mathbf{x})$ be the probabilities that the set of observation $\mathbf{x}$ belong to the categories normal and prostate tumour patients respectively.

Afterwards, we need to compute the unknown parameters for the logistic regression model uses the Maximum Likelihood Estimation approach which is defined in the following methods;

Firstly, we construct the following equation explaining the logistic regression model

$$\mathrm{p}(\mathrm{X} = \mathbf{x}) = \frac{e^{\beta_0+\beta_1 X}}{1+ e^{\beta_0+\beta_1 X}} \qquad (1)$$

Assuming we have several data points, we need to find the likelihood function of the joint distribution for the pair (x_i, y_i) to derive the estimators for the fitted parameters model. The likelihood function expresses the probability of the observed data as a function of the unknown parameters. If the response variable is coded as 0 or 1, the expression for *p(x)* is given in equation (1), and that the probability of *p(x=0) = 1-p(x)*. By assuming that the observations are independent, the likelihood function to derive the MLE estimation is expressed in the following equation

$$l(\boldsymbol{\beta}\ ) = \prod_{i:y_i=1} p(x_i) \prod_{i''= y_i'=0} (1 - p(x_i))$$

Since the principle of the maximum likelihood is to find the $\boldsymbol{\beta}$ which maximizes the likelihood function, it is easier mathematically to work with the log of the likelihood function. The expression of the log-likelihood function is defined as

$$L(\boldsymbol{\beta}) = ln\left[l(\boldsymbol{\beta}) = \sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} \{x_i ln\big(p(x_i)\big) + (1 - x_i) ln(1 - p(x_i)\}\right]$$

To derive the $\boldsymbol{\beta}$ estimators which maximizes the likelihood function, we do a partial derivation of the $L(\boldsymbol{\beta})$ with respect to each of the β_j and set the results equal to zero. Solving each the equation, we will obtain the estimator β_j.

After obtaining the estimated coefficients, the predicted probability for each observation can be computed by plugging in the value of the of **x** variables for the respective observation. Furthermore, to classify the observation, we need to set the boundary to separate the two classes. The boundary is set based on the Bernoulli concept as here we have the dichotomous possibility for each of the n-trials for the response variable. The observation will belong to the category normal patients if its probability larger than 0.5, and an observed patient will be assigned to tumour patients if its probability smaller than 0.5. Hence the designated class for each observation depends on the predicted probability of each of the observation,

In the case of sample prostate cancer classification, we firstly began the classification analysis by fitting the training dataset into the logistic regression model using R computation. The computation is run with the following R-script

```
names(prostatetests)
names(prostatetrains)
head(prostatetrains$V12601)
head(prostatetests$V12601)

#  Training Dataset
Types <- rep("Normal", 102)
Types <- as.factor(ifelse(prostatetrains$V12601==0,"Prostate",Types))
contrasts(Types)
levels(Types)
Tumours <- factor(Types, levels=c("Prostate", "Normal"))
levels(Tumours)
contrasts(Tumours)
prostatetrains$Tumours <- Tumours
summary(Tumours)
head(prostatetrains)
traindata <- prostatetrains[,-c(1,12602)]
X.train <- as.matrix(traindata[,-12601])
Y.train <- traindata[,12601]
head(traindata)

# Test dataset
Typest <- rep("Normal", 34)
Typest <- as.factor(ifelse(prostatetests$V12601==0,"Prostate",Typest))
Tumourst <- factor(Typest, levels=c("Prostate", "Normal"))
prostatetests$Tumourst <- Tumourst
summary(Tumourst)
testdata <- prostatetests[,-c(1,12602)]
```

```
X.test <- as.matrix(testdata[,-12601])
Y.test <- testdata[,12601]

# Logistic Regression
glm.fits <- glm(Tumours~.,data=traindata,family=binomial)
glm.probs <- predict(glm.fits,type="response")
glm.pred <- rep("Prostate", 102)
glm.pred[glm.probs>.5]="Normal"
contrasts(Tumours)
table(glm.pred, Tumours)
```

The execution of the above algorithm results the following classification;

Table 1. Classification Table for the Training Dataset

```
                Tumours
glm.pred   Prostate Normal
  Normal          0     50
  Prostate       52      0
```

In a classification problem, the measures of fit include classification table and model validation via an outside data set or by splitting a data set. In the prostate cancer data, it is obvious that the logistic regression model yielded a perfect classification for the training data set. based on Table 1, all the sample patients who belong to the prostate tumour patients are also predicted as prostate patients. Similarly, all the 50 normal patients also belong to the normal patient category based on the logistic regression prediction. Hence, the test error rate performed by the logistic linear regression model using the training data set is zero. Furthermore, as here we deal with medical dataset, we can assess the classification output in term of their sensitivity and specificity. The sensitivity using the training dataset is a high of 100% which means that all the tumour patients will be correctly identified when they are truly the tumour patients. Similarly, as the specificity is also 100%, all the normal patients will belong to normal patient class when they are truly not having tumour prostate. Based on the classification results for the training dataset, the fitted model does an accurate specification

However, as in most cases the fitted model work well for the training dataset, we cannot only assess the goodness of fit of certain model only based on the training dataset's result. Additionally, we also need to do a model validation via the test dataset. By running the R-code to compute the predicted value for the test dataset using the fitted model, we obtained the following outputs

```
# Predicted Probabilities Using Test dataset

glm.probstest <-predict(glm.fits,testdata,type="response")
glm.predtest <- rep("Prostate",34)
glm.predtest[glm.probstest>.5]="Normal"
table(glm.predtest,Tumourst)
```

Table 2. Classification Table for the Test Dataset

```
                 Y.test
    glm.predtest Prostate Normal
       Normal          24      9
       Prostate         1      0
```

Based on the classification table assessment given in Table 2, only 1 of the 25 prostate patients was correctly classified. Although the prediction for the normal patients is perfectly accurate, the test error rate here is strikingly high as 97.05%. Furthermore, the fitted model has significantly small sensitivity, a low 4.167% while the specificity here is high as 100%. In general, although the specificity here is high, the fitted linear logistic regression does a poor job as it has a significantly low sensitivity. Considering the validation result based on the test dataset, the overfitting problem might arise here. The poor performance in classifying the patient samples for the data outside the training dataset might occur because of the complexity of the fitted model. In this case, the fitted model included large number of predictor variables (12600) while the available observation is considered small (102). In conclusion, based on the classification table and the validation model using the data outside the training dataset, an overfitting problem might appear here.

Statistically, the overfitting problem identified earlier might appear because of the ratio between the parameters to the number of n samples is significantly large $(\frac{12600}{102})$. The large ratio indicates that the fitted model is complex for which large number of parameters need to be estimated. However, the available observation is insufficient compared to the large degree of freedom, making impossible to obtain the accurate and stable estimation. Under such condition, the bias for the model is small but the irreducible error is large. Therefore, the fitted model only does a good classification for the training dataset and gives poor classification results for the test dataset. Therefore, previous findings lead to a preliminary hypothesis that an overfitting problem might appear in the cancer prostate dataset. As the problem of overfitting comes from the big ratio between the total number of parameters and the number of n observation samples, a curse of dimensionality obviously appears in this study.

The previous discoveries suggest that the curse of dimensionality occurs here because of the involvement of large variable predictors in the fitted model. Hence, we proposed a modified model which can deal with immense number of variables. The proposed modified model called regularized logistic regression which involve the lasso or ridge regression approach. As for this study, instead of choosing the regularized logistic regression with ridge approach, the regularization with lasso method is considered more appropriate to deal with the identified overfitting problem. The reason is because the ridge approach cannot shrink the coefficients until zero, while the lasso approach is able to do the shrinkage for the coefficient variables until zero. The regularized logistic regression with lasso approach gives a penalty for the predictor variables which cause a problem or they are unnecessary to be included in model fitting. Therefore, before fitting the regularized logistic regression, we firstly obtain the penalty λ using the cross-validation method. Afterwards, we fit the regularized logistic regression mode with the chosen penalty λ. The new proposed model is expected to remove overfitting model which means we achieve stable model which gives accurate predicted probability for both of training and test dataset.

Implementing the idea of the regularized logistic regression explained above, we compute the classification method for the tumour sample patients in R using the following algorithm;

```
# Reguralized Logistic Regression by Lasso

install.packages("glmnet")
library(glmnet)
X <- model.matrix(Tumours~., traindata)[,-c(1,12602)]
Y <- Y.train
```

```
set.seed(123)
cv.lasso <- cv.glmnet(X.train, Y.train, alpha = 1, family = "binomial")
glm.regularized <- glmnet(X.train, Y.train, alpha = 1, family = "binomial",
                    standardize= TRUE,lambda = cv.lasso$lambda.min)

# Identifying how many zeros coefficients
sum(coef(cv.lasso, cv.lasso$lambda.min)==0)

# train error rate
glmregularized.probs <- predict(glm.regularized,newx=X.train,
                          type="response")
glmregularized.predtrain <- rep("Prostate", 102)
glmregularized.predtrain[glmregularized.probs>.5]="Normal"
table(glmregularized.predtrain, Y.train)


#test error rate
glmregularized.probstest <- predict(glm.regularized,newx=X.test,
                              type="response")
glmregularized.predtest <- rep("Prostate", 34)
glmregularized.predtest[glmregularized.probstest>.5]="Normal"
table(glmregularized.predtest, Y.test)
```

After doing the computation in R, we obtained the following classification results for the training dataset;

Table 3. The Classification Table for the Training Dataset by Fitting the Regularized Logistic Regression.

```
                        Y.train
glmregularized.predtrain Prostate Normal
                Normal          0     50
                Prostate       52      0
```

Before measuring the goodness of fit of the fitted regularized model, we firstly analyse how the regularized logistic deal with large size predictor variables. As explained before, the regularized model used here is based on lasso approach. Hence, we firstly find the penalty coefficient by using the cross-validation method. In this case, the chosen coefficient penalty λ by cross-validation approach is 0.004534. here, the regularized logistic regression model does a variable selection. Among all the 12600 predictor variables, only 30 of them has non-zero coefficients. As most of the variable coefficients being shrunken to zero by the regularized logistic regression, the overfitting problem identified before fitting the modified model might be disappeared. To further confirm that the overfitting problem already discarded, we do the classification assessment in both the training and test dataset using the classification table and validation model using test dataset measurements.

Now, the goodness of fit for the regularized regression model here is firstly measured by calculating the train error rate based on the classification table. The outputs for the training dataset suggest that all the samples are correctly classified by the fitted model which means the train error rate is 0%. In term of the sensitivity, the modified fitted model yielded a high of 100% and so is the specificity. As the sensitivity is high, the regularized logistic regression here performed a precise prediction. However, we also need doing the model validation on the outside training dataset. To demonstrate the idea, we fitted the fitted model into the test dataset and obtained the classification result as written in Table 4

Table 4. The Classification Table for the Test Dataset

```
                      Y.test
glmregularized.predtest Prostate Normal
              Normal           1      8
              Prostate        24      1
```

The validation result for the fitted model using the test data set shows a considered precise classification. For each of the class, only one of the patient sample is misclassified. In average, the test error rate here is a low 5.88%. Furthermore, in term of sensitivity and specificity, the sensitivity for the test dataset is a high of 96% while its specificity is of 88.88%. Connected with the assessment model result by using the classification table and validation model using the test dataset, it can be concluded that the regularized regression logistic able to discard the overfitting problem. Additionally, the stable classification results shown in both the training and test dataset indicates that the regularized logistic regression tackle the explosive variance identified in the prior model although the reducible error is larger here because of the decrease in the degree of freedom. However, that is not a big deal as long as the irreducible error is not significantly large.

Intuitively, we can relate the curse of dimensionality found here in term of the trade-off between bias and variance. It is obvious that when we overfit a model to the training data set, its irreducible error decrease while the irreducible error from the test dataset might increase. Connected to the findings of the classification results for the prostate tumour dataset, it is obvious that the problem of interested suffers a curse of dimensionality. Before the first proposed model being modified, the classification output suggests that the fitted model is overfitting as the test error rate is strikingly high. The discovery indicates that although the bias is small, indicated by the low error rate in the training dataset, the variance is explosive. However, the curse of dimensionality can be handled in sacrifices of larger bias. As we all know, there are several methods to overcome the problem by sacrificing bias to decrease the variance of estimators such as Lasso that we use in this study. By using the lasso approach, the fitted model is less flexible because the variable selection to which the variance is lesser while the bias is larger.

Assignment 3
STAT7040 Statistical Learning
Dimension Reduction

Prostate Cancer Data

Nowadays, prostate tumours are histologically and clinically acknowledged as one of the most heterogeneous cancers. Hence, the scientist was interested to study this phenomenon by studying the gene expression between tumour and normal prostate samples using microarray expression analysis. The goal of the microarray expression analysis is to classify patient samples into tumour or normal group based on their lots of genes expression. In this case, a number of 12600 genes expression variable were used as the predictor variables to classify the patient samples. Since here we deal with classification problem, we need a classification model for which both train and test dataset were needed for the fitted model validation later. For this study, the training dataset contains 102 patient samples, 52 of which are prostate tumour samples and 50 of which are normal prostate. As for the test datasets, it contains 34 sample patients which consists of 25 tumour patients and 9 normal patient samples. To make the analysis interpretable, prostate tumour patients are labelled as “0” and the normal patients are coded as ”1” in both the datasets.

As explained in the above description, this study involves a high dimension data. If we directly fit a model into the dataset, the model will likely suffer from a curse of dimensionality and/or overfitting. Furthermore, interpreting the fitted model will be hard as the proposed model involves large number of variable predictors. Hence, constructing a dimension data reduction by Principal Component Analysis (PCA) might be necessary to overcome the problem. The motivation of the PCA is to reduce the dimensionality of a large dataset by transforming the variable of a given dataset into a matrix of fewer variables, which are those that retain most of the variation of the original variables.

In this case, we firstly worked the PCA into the training dataset to obtain the loading vector components. Here, although the gene expressions are all at the same unit of measurement, some of the variances are significantly greater than the other. Hence, the work of the PCA here was done by firstly standardizing all the 12600 gene expression variables. Afterwards, we compute the PCA in R and obtained 102 principal component loading vectors. Subsequently, we project the 12600 gene expression variable into the principal components scores through the loading vectors and produced the following first two principal component scores,

```
            PC1          PC2
[1,] -36.9504957   -2.9690750
[2,] -55.1955710   -8.4149288
[3,]  16.8473197    8.8259130
```

```
   [4,]   9.3631295    4.8469192
   [5,] -48.2689104   -3.7804650
   [6,]  -3.7757014    7.6000267
   [7,]  -2.5102984    5.3964873
   [8,] -61.9556604   -7.1961318
   [9,]  14.7544530    4.9173240
  [10,]  16.9358339    9.3729304
    .
    .
    .
    .
    .
    .
    .
 [100,] -56.3998977   -5.9175211
 [101,] -62.3587313   -9.2834671
 [102,] -60.4490033   -8.2907966
```

To visualize whether the principal component success in reducing the data dimension without losing much information of the original variables, we need to plot the first two principle component scores for the gene expression dataset, and the plot is shown in the following figure

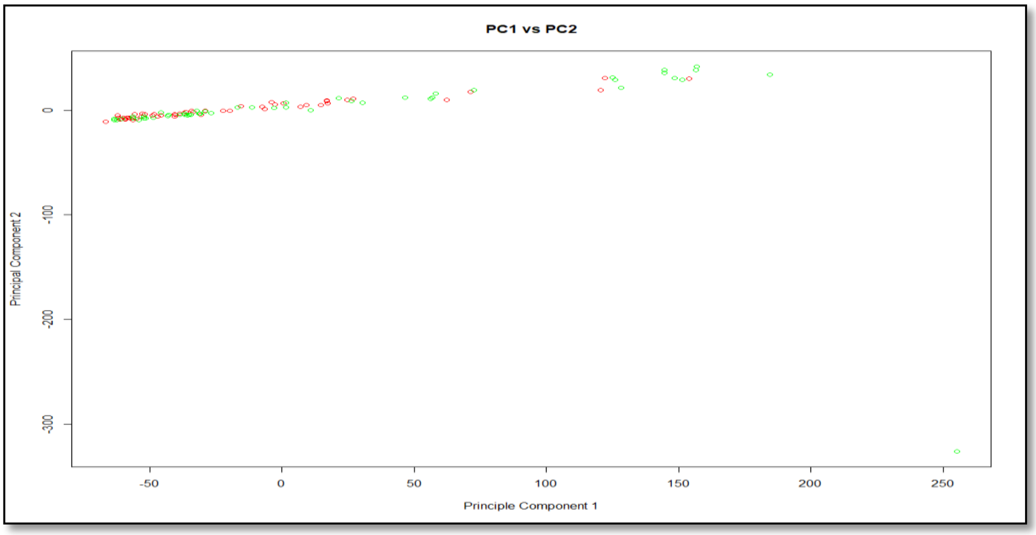


Figure 1. Scatterplot for the first and second principal component scores

Based on Figure 1, it is obvious that there is no apparent separation between the tumour and normal patients. This is a clear indication that the PCA might be not a good approach to reduce the data dimension for the gene expression data. Furthermore, the predictor variable constructed by PCA might be not a great predictors to develop a classification model such as logistic regression based on PCA features.

In PCA term, we can measure the performance of each of the principal components in explaining the variation of the original dataset by calculating the proportion of variance explained (PVE). Commonly, the PVE values is plotted and then we can observe the performance through observing the plot. In general, the PVE plot is constructed in two forms, which are a PVE plot for individual principal component and a scree plot which depicts the cumulative PVE explained by the 102 constructed principal components. Both of the PVE plots are shown in the following figures,

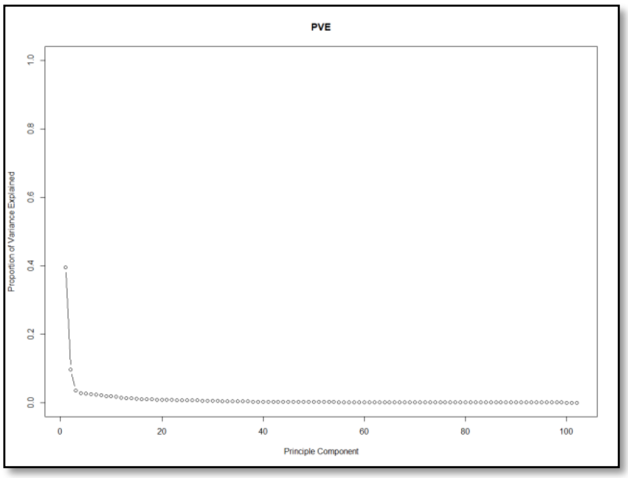


Figure 2(a). PVE for each principal component

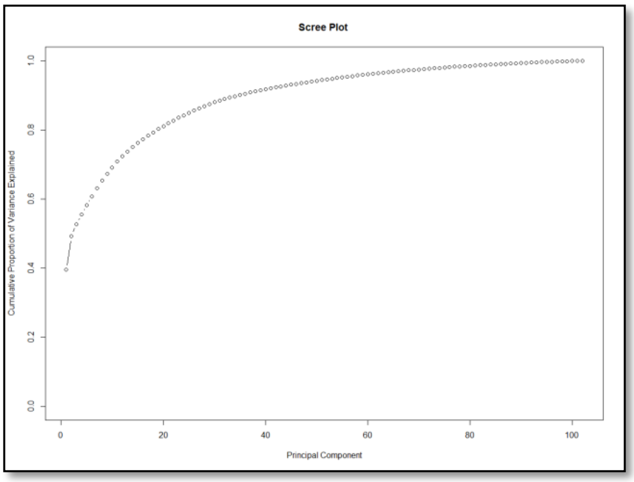


Figure 2(b). Cumulative PVE

Based on the Figure 2(a), after the 19th principle component, each of the principle component only explained a small amount of variance of the original dataset so that the line starts to look constant. For the gene expression data, the first principal component explains 39.46 percent of the variance in the original data, while the second principal component explains at 9.75 percent. Here, the first two principle components already explain the variance of the original data at almost 50 percent, means that the other 100 PCA variables explain roughly a half information of the original data altogether. Therefore, as the PVE by the first two component still at around 50 percent, we can still include the next PCA variables so that we do not lost much information of the original dataset for the further analysis purposes.

In PCA, we usually are not interested in all principal component and generally there is not any clear rule to determine how many PCA variables we would like to use for further analysis. However, researchers commonly use scree plot to decide the number of principle components they will include for modelling purposes. This is done by eyeballing the Scree plot and look at which point the cumulative PVE start to constantly behave. By observing Figure 2, the roughly 80 percent variation of the original data is achieved by the first nineteenth components. Based on PVE, it is reasonable to use the nineteenth first components for the classification modelling.

However, we can also find the number of PCA to use for model fitting by observing the eigenvalue. For the gene expression data, the eigenvalue for each of the principal component is their respective standard deviation as we standardized the dataset before constructing PCA. Regarding the PCA for the gene expression data, its first component until the 101th eigenvalues are larger than 1. By which, all the 101 component contains important information of the original dataset. However, if all the 101 first principal components will be used for the fitting classification model, the curse of dimensionality might still appears. Hence, we try to find the estimated number of principle components through the cross-validation method. Through cross-validation method, we obtained the following cross-validation plot

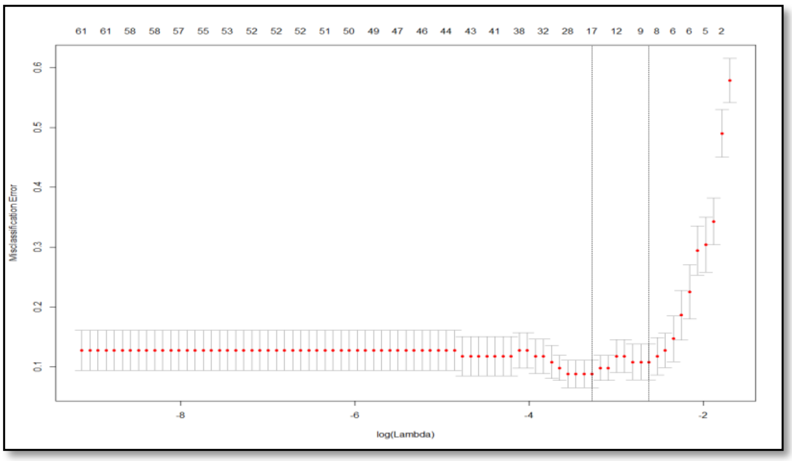
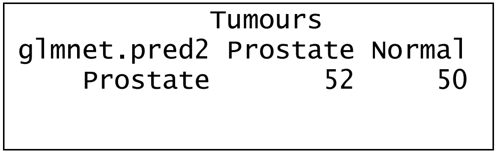


Figure 3. Cross-validation

Based on Figure 3, the smallest misclassification error is obtained when the number of used component variables is around 17. Hence, combining the chosen principal components by PVE and cross-validation, it is preferable to use the 17 components. The decision was made by considering the fact that the difference in the cumulative PVE between the two number PCA components is considered small and also avoiding the likelihood of overfitting.

After deciding the number of principal components that will be used for the model fitting, we fitted the proposed classification model into the training dataset with the chosen PCA components as the predictor variables and we obtained the following results,

Table 1. Classification Table using PCA Variables



```
              Tumours
glmnet.pred2 Prostate Normal
    Prostate       52     50
```

Based on Table 1, there is no misclassification in both group. Hence, the train error rate performed by the logistic linear regression using the chosen PCA variables is significantly low at 0 percent. Furthermore, as prostate cancer data related to medical case, we can also assess the goodness of fit of the fitted model through the sensitivity and specificity term. In this case, the sensitivity using the training dataset and PCA variable is a high of 100 percent which means that all the tumour patients will be correctly classified if they are truly the tumour patients. Hence, based on the training error rate and the sensitivity term, the fitted model with the chosen PCA variables does an accurate classification.

However, as in most cases the fitted model perform well in the training dataset, we cannot only assess the model appropriateness only based on the training dataset. Therefore, the fitted model validation via the dataset outside the train dataset is needed. Here, the PCA component scores are constructed by borrowing the component loading vectors from the constructed PCA using the training dataset. Below are the first two principal component scores for the test dataset,

```
        PC1       PC2
  [1,] -253.1350 -151.7861
  [2,] -246.8972 -155.6205
  [3,] -251.0287 -150.5668
  [4,] -298.2377 -160.7715
  [5,] -341.6762 -166.9253
  [6,] -288.7791 -170.4375
  [7,] -150.3685 -161.0919
  [8,] -206.0297 -158.5992
  [9,] -236.1423 -155.9512
 [10,] -300.0698 -177.4619
   .
   .
   .
   .
   .
 [31,] -244.9060 -173.4395
 [32,] -227.6411 -166.1100
 [33,] -172.6386 -165.7031
 [34,] -255.9882 -153.6946
```

In the training dataset, the tenth first dataset were used to fit the logistic linear regression. Hence, in the test dataset, the tenth first dataset was also used for the fitted model validation via the test dataset. After running the computation in R, we obtained the following outputs;

Table 2. Classification Table using PCA variable for the test dataset

```
                      Y.test
glmnet.predtest Prostate Normal
       Prostate       25      9
```

Based on the classification table using the test dataset given in Table 2, while all the prostate sample are correctly classified by the logistic regression, all the normal patients are misclassified. Hence, the test error rate here is 26.47 percent which is quite high considering the observation of the test dataset is only 34. Additionally, all the predicted probability is one which indicate that the fitted model suffer from the overfitting.

The test error test by using the PCA variables from the test dataset shows at 26.47 percent which is significantly high compared to the regularized logistic regression (5.88 percent). Considering the test error rate yielded by the principal component logistic regression, this fitted model does not perform well in classifying the prostate and normal sample patient. Compared to the regularized logistic regression, the principle component analysis does not success in dealing with high dimensional data for the prostate cancer dataset. The relatively worse performance is because PCA is an algorithm that does not consider the response variable or prediction target into account since PCA is unsupervised statistical learning. In this case, PCA only treat the feature that has large variance as important features, but the features might have nothing to do with the prediction target. Furthermore, we know that the tenth first principal component used in the fitted model accounts 82 percent variation of the origin dataset. Hence, the lost information from the original dataset is about 18 percent which might be contain the important information for prostate cancer data classification.

For the prostate cancer classification, the regularized logistic regression has 30 predictor variables in the end, while the principle component classification used 17 predictor variables. Hence, in term of flexibility, the principle component classification model might outperform the regularized logistic regression. Having said so, the principle component classification should perform better in the test dataset compared to the regularized logistic regression. However, the results from the principal component classification show the opposite findings. The principal component classification does not perform well on the test dataset might be a consequence that PCR is not a feature of selection method and supervised statistical learning. In this case, even though PCR offer a much simpler way to fit logistic linear regression, it cannot guarantee that the components involved in the model fitting contains the important information for the response variable.

In this study, the regularized logistic regression offers simpler interpretation compared to the principal component classification model. Such condition occurs as the value of the predictor variables fitted into the regularized logistic regression come from the original dataset. In contrast, the interpretation for the principal component classification is hard and complex because the scores for its predictor variable are the function of the 12600 genes expression. Therefore, in term of interpretability, the regularized logistic regression exceeds the principal component classification.

Taking account all the findings on the performance for both regularized logistic regression and the principal component classification models, the regularized logistic regression might be the appropriate approach to deal with the high dimension data for the prostate cancer classification problem. The bad performance by the principal component analysis is expected because the plot of the first and second component scores does not show any separation between the prostate and normal patient.