MATHEMATICS FOR MACHINE LEARNING

PROJECT 2

DIMENSIONALITY REDUCTION IN CLASSIFICATION

Students:
Sanjay Chandran MM | IST1104332
Deepak Laxman | IST1104341

Bologna Masters degree in Biotechnology,
Instituto Superior Tecnico

1 Introduction

Feature selection is an important preprocessing process in machine learning. Too much of redundant features in the dataset is a disaster for any machine learning model. Feature selection algorithm majorly focuses on maximizing relevant information and minimizing redundant information. In this project we have used 3 dimensionality reduction methods - feature selection based on mutual information with correlation coefficient (CCMI), Principle component analysis (PCA) and Linear Discriminant Analysis (LDA). Features obtained from these 3 methods are evaluated for classification by 4 measure of performances including Accuracy, Macro_Recall, Macro_Precision and Macro_F1 measure.In this project we have used Dry bean dataset. It has 13,611 samples, 16 features and 7 classes - SEKER, BARBUNYA, BOMBAY, CALI, HOROZ, SIRA and DERMASON.

2 Preliminary data analysis

It is important to do preliminary data analysis before proceeding into feature selection methods. To understand the redundancy among the features, we did correlation plot of the dataset. Higher correlation between any 2 features indicates higher redundancy between them. Including these features in the dataset would slow down the training process of the classifier, reduces its classification ability and it makes the model hard to interpret.

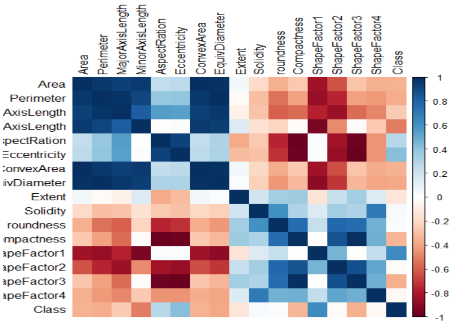


Figure 2.1: Correlation plot

From the correlation plot we can interpret that, Area, Perimeter,Major Axis Length, Minor Axis Length are highly positively correlated with Convex Area and Equiv diameter as expected. Similarly, Area, Perimeter, MajorAxis Length and Minor Axis Length are highly negatively correlated with ShapeFactor 1 and ShapeFactor 2. Aspect Ration and Eccentricity are highly negatively correlated with Shape factor3 and compactness. Therefore, we can clearly see high redundancy among the features and it is a potential problem of our dataset.

3 Data split

The data set has been randomly divided with the seed value of 123, and 70% of the data is taken as training data set and remaining 30% is taken as test data set. Training data set is used in further steps like training KNN classifier and running forward feature selection using information theory.

4 Feature selection methods

4.1 Forward Feature selection using correlation and mutual information (CCMI)

4.1.1 Concept

In this method we use correlation coefficient along with mutual information to calculate the relationship between features in our dataset. It is known that correlation coefficient measures the linear relationship between the features, therefore the absolute value of correlation coefficient is taken as weight for the redundancy term. It is paramount to understand the concept of mutual information and conditional mutual information:

Mutual information is used to measure the amount of information that a random variable contains in another random variable. Considering two random variables X and Y, their marginal probability density functions are p(x) and p(y), and their joint probability density function is p(x,y). Mutual information is represented by I(X;Y).

$$I(X;Y) = \sum\sum p(x,y) log \frac{p(x,y)}{p(x)p(y)} \tag{4.1}$$

Conditional Mutual Information of random variables X and Y given random variable Z is defined as

$$I(X;Y|Z) = \sum\sum\sum p(x,y,z) log \frac{p(z)p(x,y,z)}{p(x,z)p(y,z)} \tag{4.2}$$

Our goal is to select features that are relevant to class and have low redundancy among themselves. Relevancy of features with classes by knowing few other features is measured by conditional mutual information between the class label C and feature X_m, where the conditioning variable is feature X_s (from the selected features) i.e I(X_m;C|X_s).Redundancy is measured by calculating mutual information between the feature X_m and the selected feature X_s i.e, I(X_m; X_s). Redundancy term is weighted by correlation term in CCMI method.

The CCMI score for feature X_m is calculated by:

$$J_{CCMI} = min I(X_m; C|X_s) - min|\rho X_m X_s|.I(X_m; X_s) \tag{4.3}$$

$$\rho X_m X_s = \frac{Cov(X_m, X_s)}{\sqrt{D(X_m)}\sqrt{D(X_s)}} \tag{4.4}$$

Here, $\rho X_m X_s$ represents the correlation coefficient between the candidate feature X_m and the selected feature X_S. Covariance between X_m and X_s is represented by Cov(X_m, X_s). D(X_m) and D(X_S) are the variances of X_m and X_s respectively.

4.1.2 Results

We used this concept (as mentioned above) to select 10 best feature out of all 16 features. To calculate mutual information between random variables, a built-in function called *mutinformation* from infotheo R package is used. Similarly to calculate conditional mutual information, a built-in function called condinformation is used. Based on CCMI scores of features, we have selected top 10 features.

The selected features - **Perimeter, shape factor4, compactness, Extent, minoraxis length, solidity, area, convex area, equivdiameter, shape factor1.**

4.2 Principal Component Analysis (PCA)

4.2.1 Concept

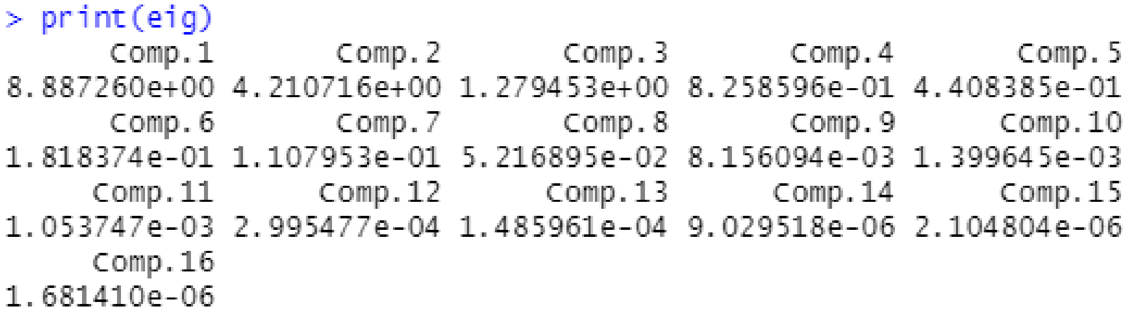
This method involves a dimensionality reduction technique which tries to reduce the number of independent variables by capturing the maximum variance of the given data. We try to find out the straight line that best spreads the data out when it is projected along it. It is constructed from the linear combination of the coordinates with a weighted value determined by the eigen values of the correlation matrix. This way we transform a set of X correlated variable over Y samples to a set of P uncorrelated principals components over the samples. The first principal component captures the maximum variance in a dataset and also determines the direction of higher variability. The ability to capture variability keeps decreasing with successive principal components.

We import a library called **psych** to plot the correlation matrix of dependant data with the function **pairs.panel**. Now the function **princomp** with attributes *scores and correlation* specified is used to find the principal components of given variables.

In Algebraic Notation:
Given a sample of n observations on a vector of p variables, $X = (x_1, x_2, ..., x_p)$
define the first principal component of the sample by the linear transformation
$z_1 = a_1^T = \sum a_{i1} X_i$, where the vector $a_1 = (a_{11}, a21, ...a_{p1})$ is chosen such that $var[z_1]$ is maximum.

In general for our data set, after applying the principal component analysis function, we received

16 principal components corresponding to the 16 variables excluding the class variable. All these 16 components have a separate attribute for their standard deviations. Considering the square of this standard deviation, we get the eigen values of the components, which if is greater than value 1 can be considered a significant principal component. All the other components with an eigen value less than 1 can be neglected. For our data, we have got the eigen values for the components as follows:



```
> print(eig)
      Comp.1       Comp.2       Comp.3       Comp.4       Comp.5
8.887260e+00 4.210716e+00 1.279453e+00 8.258596e-01 4.408385e-01
      Comp.6       Comp.7       Comp.8       Comp.9      Comp.10
1.818374e-01 1.107953e-01 5.216895e-02 8.156094e-03 1.399645e-03
     Comp.11      Comp.12      Comp.13      Comp.14      Comp.15
1.053747e-03 2.995477e-04 1.485961e-04 9.029518e-06 2.104804e-06
     Comp.16
1.681410e-06
```

Figure 4.1: Eigen values data

So we take in account only the first three components and discard the rest. Now we find the scores of these 3 components and create a data frame just with the scores data for our analysis. Thus we have managed to reduce the dimension of the data from 13611x16 to 13611x3.

4.3 Linear Discriminant Analysis (LDA)

4.3.1 Concept

Linear Discriminant Analysis (LDA) helps us in finding the linear combinations of features that separates multiple classes. LDA primarily focuses in separating the classes along a particular axis. The maximum separability among classes is achieved by maximizing the between class variance and minimizing the within class variance. Therefore the equation for LDA can be given by:

$$LDA = \frac{Betweenclassvariance}{Withinclassvariance} \tag{4.5}$$

$$A = S_w^{-1} S_b \tag{4.6}$$

Here, S_w contains the data about within class variance and S_b contains the data about between class variance.
S_w is given by:

$$S_w = \sum (x_i - \overline{x_c}'''')(x_i - \overline{x_c})^T \tag{4.7}$$

S_b is given by:

$$S_b = \sum(\overline{x_c} - \overline{x})(\overline{x_c} - \overline{x})^T \tag{4.8}$$

$\overline{x_c}$ is the average of all components within a single class and $\overline{x}$ is the overall average including all classes.

Eigenvalues and Eigenvectors are found out for matrix A (Eqn 3.6), this is similar to PCA whereas the matrix considered in LDA is different. Eigenvalues are now sorted in descending order and first ten eigenvalues are only considered. Eigenvectors corresponding to these 10 eigenvalues are taken into consideration for further calculations. Therefore the dimensionality of the dataset is now reduced to **13,611x10** from 13,611x16.

5 KNN Classifier

5.1 Concept

For the purpose of this project, we have used the **K nearest neighbour Classifier** to predict the class of each data points. This supervised learning algorithm considers the euclidean distance between k nearest data points to predict the class of a new data point. In simple terms, lets say there are two classes A and B for a given data set. Now a new data point enters the set and we need to predict which class this point belongs to. To perform this, the algorithm considers k data points that lies near to the new data point. By calculating the euclidean distance between the new point and the other points individually, the model finds the first k nearest neighbours to make use of. Later amongst the k neighbours, depending on the maximum set that has a particular class, it decides that the new data point also belongs to the same class.

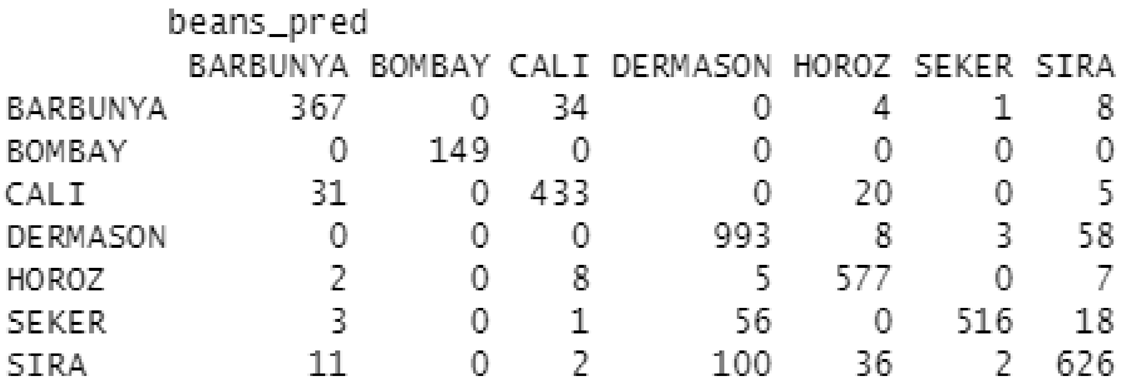
5.2 Implementation in Code

To start with the implementation of this classifier in R, we first install the **caret** and **class** packages.We have three different input data sets which are derived from the three dimensionality reduction methods. These input data set is divided into training and testing with a probability of 70% and 30% respectively. This is achieved by the commands:

```
smp_size <- floor(0.70 * nrow(data_name))
train_index <- sample(seq_len(nrow(data_name)), size = smp_size)
train_data <- data_name[train_index, ]
test_data <- data_name[-train_index, ]
```

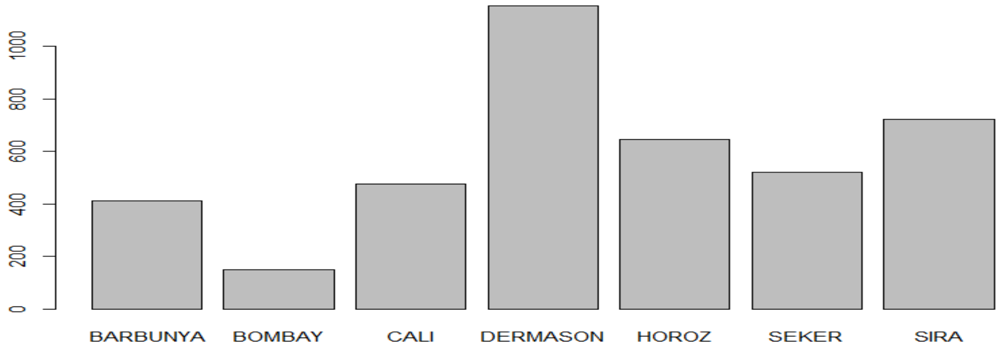
We normalize the data before using, to avoid the problem of over dependency on one variable. The **data_norm** function normalizes the data in our code. **beans_training** and **beans_testing** are two tables which contains the normalized data and the column name **Class** removed so that these completely contain numeric values. We use the function **knn()** to train the knn model with training data set and predict the rest with the testing data. We generate a Confusion Matrix for the predicted class and original test data class in the variable **confusion_matrix**. Finally with the help of this confusion matrix, we calculate the accuracy of the KNN classifier, Macro precision, Macro recall and the F score.

6 Observations

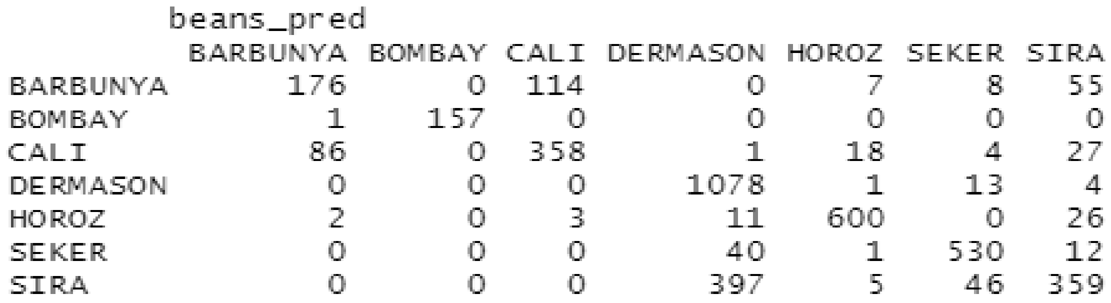


```
           beans_pred
            BARBUNYA BOMBAY CALI DERMASON HOROZ SEKER SIRA
  BARBUNYA       367      0   34        0     4     1    8
  BOMBAY           0    149    0        0     0     0    0
  CALI            31      0  433        0    20     0    5
  DERMASON         0      0    0      993     8     3   58
  HOROZ            2      0    8        5   577     0    7
  SEKER            3      0    1       56     0   516   18
  SIRA            11      0    2      100    36     2  626
```

(i)Confusion Matrix for FFS method

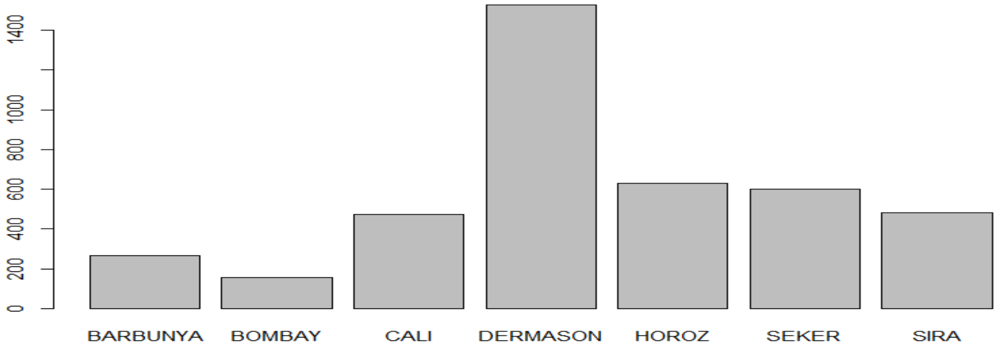


(ii)Bar Plot of Predicted data in FFS method

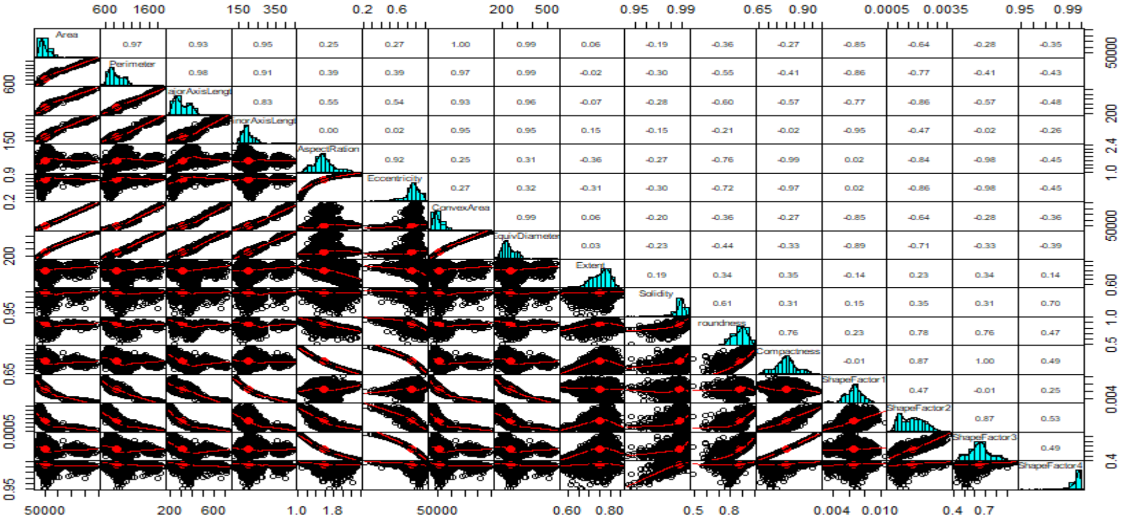


```
           beans_pred
            BARBUNYA BOMBAY CALI DERMASON HOROZ SEKER SIRA
  BARBUNYA       176      0  114        0     7     8   55
  BOMBAY           1    157    0        0     0     0    0
  CALI            86      0  358        1    18     4   27
  DERMASON         0      0    0     1078     1    13    4
  HOROZ            2      0    3       11   600     0   26
  SEKER            0      0    0       40     1   530   12
  SIRA             0      0    0      397     5    46  359
```

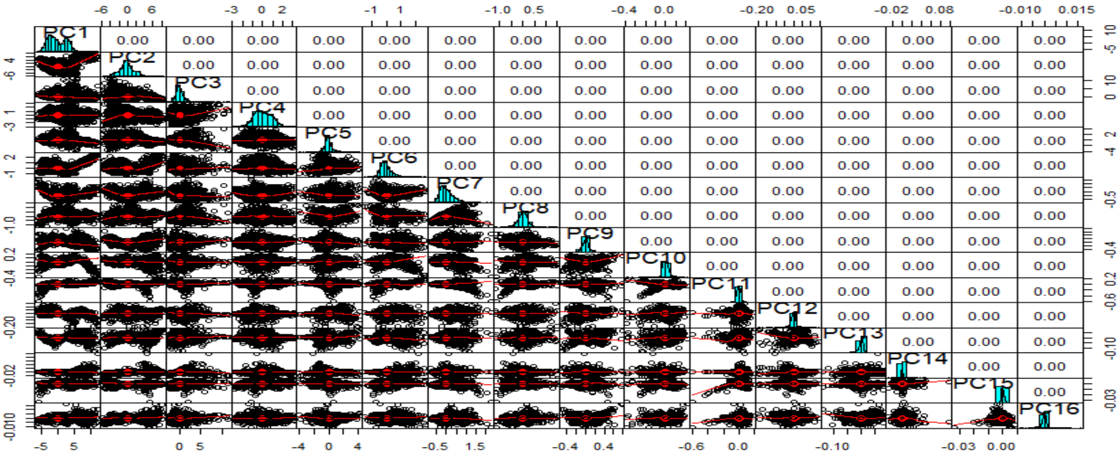
(iii)Confusion Matrix for PCA method



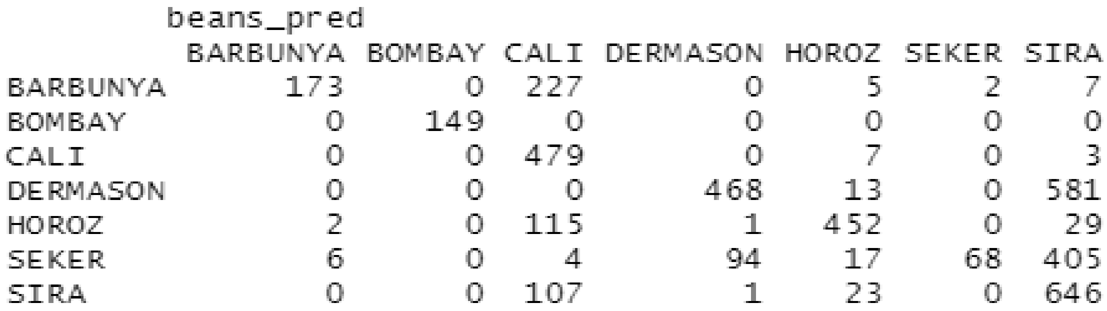
(iv)Bar Plot of Predicted data in PCA method



(v)Correlation matrix before PCA

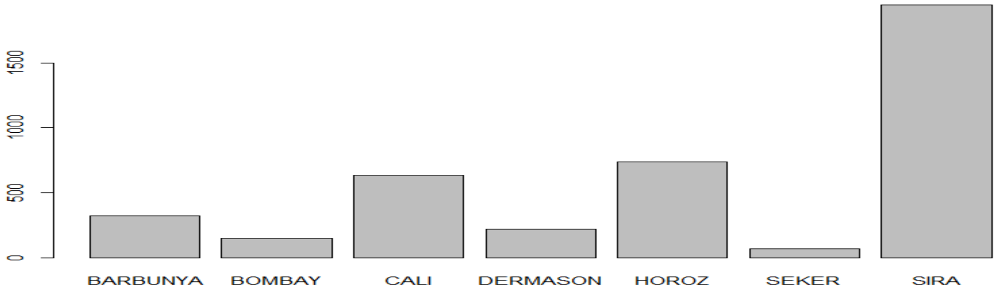


(vi)Correlation matrix after PCA



```
          beans_pred
           BARBUNYA BOMBAY CALI DERMASON HOROZ SEKER SIRA
  BARBUNYA      173      0  227        0     5     2    7
  BOMBAY          0    149    0        0     0     0    0
  CALI            0      0  479        0     7     0    3
  DERMASON        0      0    0      468    13     0  581
  HOROZ           2      0  115        1   452     0   29
  SEKER           6      0    4       94    17    68  405
  SIRA            0      0  107        1    23     0  646
```

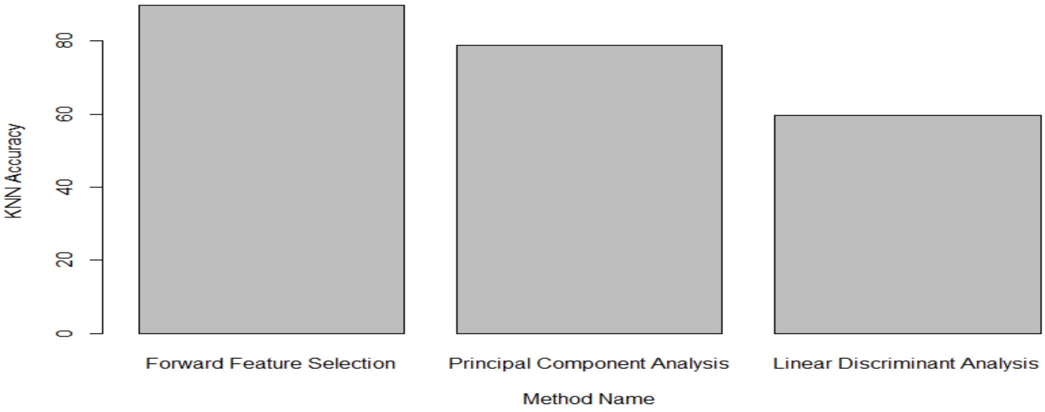
(vii) Confusion Matrix for LDA method

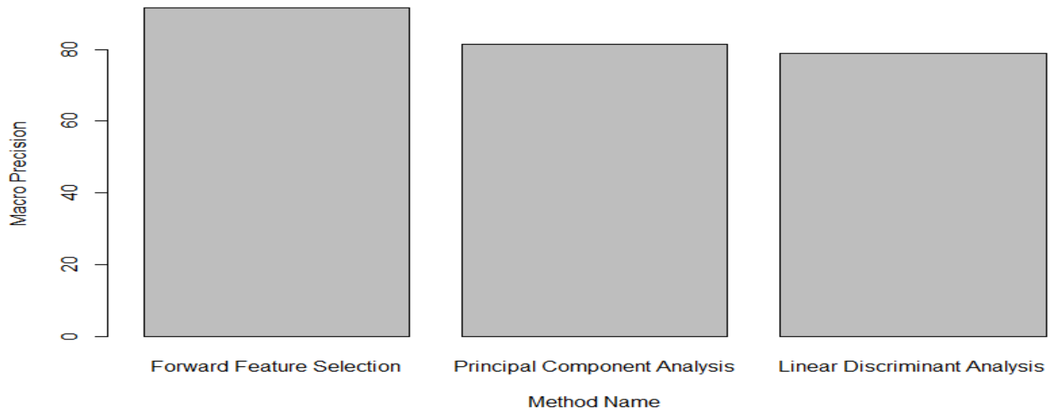


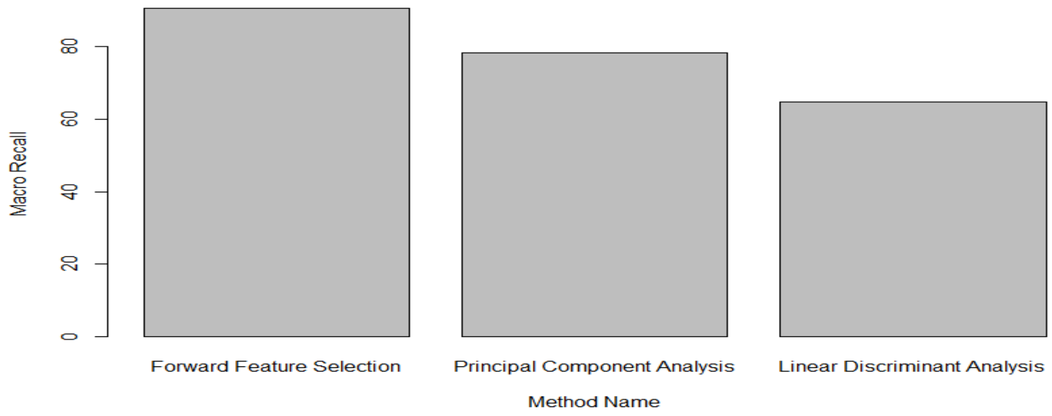
(viii) Bar Plot of Predicted data in LDA method

7 Discussion

In this report, we had a detailed analysis of different dimensionality reduction techniques such as the **Forward Feature Selection, Principal Component Analysis and Linear Discriminant Analysis**. All these methods gave us a a very simple data set which can be used to classify items easily using the KNN classifier. The following is the plots for the accuracy, macro precision and macro recall values of the methods:







We can interpret that the Forward Feature Selection method showed a relatively higher accuracy rate for the classification process. Nevertheless, it has a higher value of Macro precision and macro recall scores as well. All three bar plots also reveals that Principal Component Analysis performs better than Linear Discriminant Analysis. Forward Feature Selection method using Information theory performs better than all other methods as it uses the concept of mutual information that captures all kinds (linear as well as non-linear) of correlations between features. Also, it intelligently sorts the features based on redundancy among features and its relevance with class label. The consideration of relevancy with class label along with redundancy lacks in PCA and LDA methods. In LDA, we just consider the variance within the class and variance between class. The diagonal entries of the confusion matrix tells the number of data points that correctly matches with original test data and predicted data.