Comparison of different Classification Models

Abhilash Singh (191003)
Ankit Gupta (191016)

November 2020

1 Acknowledgement

We would like to express our heartfelt gratitude to Dr. Shankar Prawesh for helping us in every difficulty which we face during this project. It has been a great learning experience and has also provided us with a practical insight of the theoretical knowledge gathered during the course Advanced Statistical Methods For Business Analytics.
We also take this opportunity to thank the authors and publishers of the various books and journals we have consulted. Without those this work would not have been completed.

We would also like to thank our parents and seniors for their extensive support throughout the session. Their constant encouragement has enabled us to complete the project within the stipulated time-period.

Contents

1 Acknowledgement **2**

2 Introduction **4**

3 Objective **4**

4 Performance Metric **5**
4.1 Misclassification Error rate 5
4.2 ROC curve 5
4.3 Confusion Matrix 5

5 Data description **6**

6 Exploratory Data Analysis **6**

7 Classification Model:- **6**
7.1 Logistic Regression:- 6
7.2 Linear Discriminant Analysis (LDA) 7
7.3 Quadratic Discriminant Analysis (QDA) 7
7.4 Random Forest 8
7.5 KNN 8
7.6 Support Vector Machine 9

8 Comparison between applied classification model **10**

9 Why LDA Performing Well:- **12**

10 Interpretation of ROC curve **14**

11 Conclusion **14**

12 References:- **14**

13 Contribution **14**

2 Introduction

Nowadays data scientist are using different machine learning techniques to discover patterns in big data.Either they use Supervised algorithms or unsupervised algorithms while dealing with regression problem or classification problem.
In this project work we used different machine learning techniques to deal with classification problem. In classification, response variable may have two class levels (binary model) or more than two class levels (multinomial model).In this case we have to build a binary classification model on training data using different classification techniques like **Logistic Regression, Linear Discriminant Analysis,SVM,KNN etc.** and using these model we have to predict class levels.
Key Words : − Misclassification error, Logistic Regression,LDA,QDA, KNN,Random Forest,SVM,ROC curve.

3 Objective

· Develop different classification model on the training set of observation.
· Predict class levels of test observation using these model and compute misclassification error rate.
· Compare accuracy of different model on test data.
· Generate ROC curve for the best performing classification model.

4 Performance Metric

In this project work two type of performance metric has been used for comparing accuracy of different model.

4.1 Misclassification Error rate

If y is the actual class level of a observation and $\hat{y}$ is predicted class level of observation using suitable model than mean of the no of observation that misclassified is known as misclassification error(ME) that is defined as

$$\textbf{Misclassification Error} = \frac{1}{n}\sum_{i=1}^{n}(y_i \neq \hat{y}_i)$$

where n is total no of observation. (misclassification error$\times$100) is defined as **misclassification erroe ra**

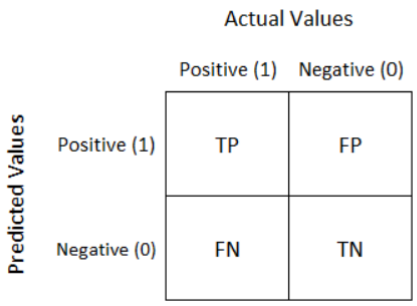
Accuracy=(1-misclassification error)$\times$100

4.2 ROC curve

This is a commonly used graph that summarizes the performance of a classifier over all possible thresholds. It is generated by plotting the True Positive Rate (y-axis) against the False Positive Rate (x-axis) as you vary the threshold for assigning observations to a given class.
False Positive Rate: When it's actually no, how often does it predict yes? FP/actual no
True Negative Rate: When it's actually no, how often does it predict no? TN/actual no

4.3 Confusion Matrix

A confusion matrix is a table that is often used to describe the performance of a classification model (or "classifier") on a set of data for which the true values are known. Confusion matrix for binary classifier is depicted below :- Oftenly Mislassification and



Accuracy can be determined with the help of Performance matrix, Which is given as..
1. Accuracy = (TP+TN)/Total
where, Total=TP+TN+FN+FP 2. Misclassification Error = (FP+FN)/Total = 1 - Accuracy

5 Data description

There are two data sets, one is train data set on which we have to train a model and another data set is test data set for which we have to predict the class level.
There are 21 variables in data set in which y is response variable and rest x1,x2...x20 are predictors.There are 200 observation in training data set and 1000 observation in test data set.

6 Exploratory Data Analysis

Dimension of the train data is (200 × 21).The response variable is categorical having two class level 0 and 1 and all other predictors are numerical.There are no missing observation found in the data.
Average of each predictors is approximately same which can also be seen by the standard deviation of mean of each predictors. Similarly standard deviation is also approximately same for each predictor.
Each predictor are approximately normal which can be seen from the histogram and density plot of predictor.

7 Classification Model:-

7.1 Logistic Regression:-

Logistic regression is a model used for classification of different class levels where class level may be binary or multinomial. In perticular,the logistic function,which gives the probability of p(X) = p(Y=1|X) (here Y is class level which takes 0 or 1) and if we have to predict binary class using singal predictor, the logistic function is given by

$$\text{p(X)}=\frac{e^{\beta_0+\beta_1}}{1+e^{\beta_0+\beta_1}} \ \ldots \ (1)$$

Generally, in problem of binary response prediction when multiple predictors are given, logistic function is given by

$$p(X) = \frac{e^{beta^T * X}}{(1 + e^{beta^T * X})}$$

$$\ldots(2)$$

we can also write the logistic regression model as:-

$$log(\frac{p(X)}{1 - p(X)}) = beta^T * X$$

After calculating the value of p(X) we take a threshold say c and if p(X) > c we classify response Y corresponding to given predictor X in calss '1' otherwise in class '0'. In many of the case we take c = 0.5.
In the given problem we take threshold c=0.5 and fit a logistic regression model and compute p(X) using 2 (as we have 20 predictors) for the train and test set.
Applying this model, **train error=0.095 and test error =.173**. The performance matrix is given as

So, Accuracy of train set= 90.5 percent
Accuracy of test set=82.7percent

```
> train_error
train error
      0.095
> test_error
test error
     0.173
> log_confusion_mat
           Y_test
test_pred   0   1
        0 407  89
        1  84 420
```

7.2 Linear Discriminant Analysis (LDA)

Assumption:-we assume that X =(X1,X2,...,Xp) is drawn from a multivariate Gaussian (or multivariate normal) distribution, with a class-specific multivariate Gaussian mean vector and a common covariance matrix.the LDA classifier assumes that the observations in the kth class are drawn from a multivariate Gaussian distribution N(μ_k,σ), where μ_k is a class-specific mean vector, and is a covariance matrix that is common to all K classes. Dicriminant function is written as :-

$$\delta_k(x) = x^T \sum^{-1} \mu_k - \mu_k^T \sum^{-1} \mu_k + log(\pi_k)$$

and we assign a observation to that class for which this function is maximum. Applying this model, **train error=0.1 and test error =.158**. The performance matrix is given as

```
> lda_train_error
[1] 0.1
> lda_test_error
[1] 0.158
> lda_confusion_mat
                    Y_test
lda_test_pred_class   0   1
                  0 419  86
                  1  72 423
```

So, Accuracy of train set= 90.00 percent
Accuracy of test set=84.20 percent

7.3 Quadratic Discriminant Analysis (QDA)

.QDA is quite similar to LDA just a liitle relaxation in assumption of covariance matrix is here that in QDA feature vecctor have class specific covariance matrix . here the dicrminant function are quadratic function of X and is written as :-

$$\delta_k(x) = \frac{-1}{2} * (x - \mu_k)^T \sum_k^{-1} (x - \mu_k) + log(\pi_k)$$

$\sum_k$ denote the class specific covariance matrix and μ_k denote the class specific mean
π_k denote the probality that response belong to the k^{th} class
We assign a observation to that class for which this function is maximum.
Applying this model, **train error=0.07 and test error =.258**. The performance matrix is given as

```
> qda_train_error
[1] 0.07
> qda_test_error
[1] 0.258
> qda_confusion_mat
                    Y_test
test_pred_class   0   1
              0 359 126
              1 132 383
```

So, Accuracy of train set= 93.00 percent
Accuracy of test set=74.20 percent

7.4 Random Forest

Random forest is the one of the most widely used nonlinear supervised machine learning algorithm which is used in both classification and regression. Random forest algorithm constructs the multitude decision trees at the time of training and gives the output of class as the mode of the classes in case of classification and average of the prediction in case of regression of the individual trees. Random forest approach is always preferred over the decision tree approach because decision trees provide poor accuracy as compared to the random forest algorithm. We call this algorithm as random because they choose predictor randomly at the time of training and they are called forest because they take output of multiple trees to make decision. Random Forest gives better performance when the data set is large.

Algorithm used in random forest:

- Pick at random k data points from Training data set.
- Build the Decision Tree associated to these k data points.
- Choose the number of Ntree of trees which we want to build and repeat step 1 and step 2
- For a new data point. Make each one of our Ntree trees predict the category to which the data points belongs, and assign the new data point to the category that wins the majority vote.

After applying Random forest model on our data set we get **0%** misclassification rate on **training data set** and **21.5%** misclassification rate on **test data set.**

```
> train_error_forest
[1] 0
> test_error_forest
[1] 0.218
> forest_confusion_mat
      pred_forest_test
Y_test   0   1
     0 399  92
     1 126 383
```

So, Accuracy of train set=100.00 percent
Accuracy of test set=78.20 percent

7.5 KNN

K-Nearest Neighbors is the one of the most basic supervised machine learning algorithm that can be used to solve both type of the problems i.e. classification and regression. knn is a non-parametric algorithm means that it does not make any underlying assumption about the distribution of the data. This algorithm becomes significantly slower as the number of predictors or independent variables increase.

Algorithm used in KNN:

- Choose the number k of neighbors.
- Take the K neighbors of the new data point, according to the Euclidean distance.
- Among these k neighbors ,count the number of data points in each category.
- Assign the new data point to the category where you counted the most neighbors.

```
> train_error_knn_optimum
[1] 0.23
> test_error_knn_optimum
[1] 0.225
> knn_confusion_mat
      pred_knn_test
Y_test   0   1
     0 365 126
     1  99 410
```

After applying K Nearest Neighbors model on our data set we get **23%** misclassification rate on **training data set** and 22.5% misclassification rate on **test data set**.
So, Accuracy of train set= 70.00 percent
Accuracy of test set=77.50 percent

7.6 Support Vector Machine

Support vector machine is a **supervised machine learning** algorithm that is used for **classification** and **regression** analysis of data. Generally we use this model for classification of our data set. Most of the time, support vector machine constructs a **hyperplane or a set of hyper plane** in high or infinite-dimensional space, which is used for classification, regression or other task like outlier's detection. We set the **best hyperplane** to that hyperplane which has the largest distance to the nearest training data point of any class. In general, **larger margins** give the lower generalization error of the classifier. The distance between the points and the dividing line is **known as margin**. According to the SVM, we have to find the points that lie closest to both the classes and is called support vector .**Basic principle behind the working of support vector machine is to create a hyperplane that separates the data into classes. However not all data are linearly separable which makes it hard to separate into different classes linearly. When we are not able to separate data linearly then we need to use suitable transformation to show that we can separate the data linearly, and that is why we need to find a kernel function**. Kernel Function is a method used to take data as input and transform it into the required form of processing data. **Kernel function generally transforms the training data set so that a non-linear decision surface is able to transform to a linear equation in a higher number of dimension spaces.**
There are some standard kernel functions:

- **Standard kernel Function Equation:**

$$k(\bar{x}) = \begin{cases} 1 & ||\bar{x}|| \leq 1 \\ 0 & otherwise \end{cases}$$

- **Linear kernel:**
  Equation for prediction for a new input using the dot product between the input x and each support vector xi is calculated as

```
> svm_train_error
[1] 0.1
> svm_test_error
[1] 0.174
> svm_confusion_matrix
              Y_test
svm_pred_test   0   1
            0 417 100
            1  74 409
```

So, Accuracy of train set=90.00 percent
Accuracy of test set=82.6 percent

K(x,xi)=B(0)+sum(ai.(x,xi))
this is equation that involves calculating the inner product of a new input vector(x) with all support vector in training data. Where B(0) and ai are estimated parameter from training data set.

- **Polynomial kernel:**
  Polynomial Kernel function is most commonly used kernel function in SVM. Mathematical function of polynomial kernel is defined as

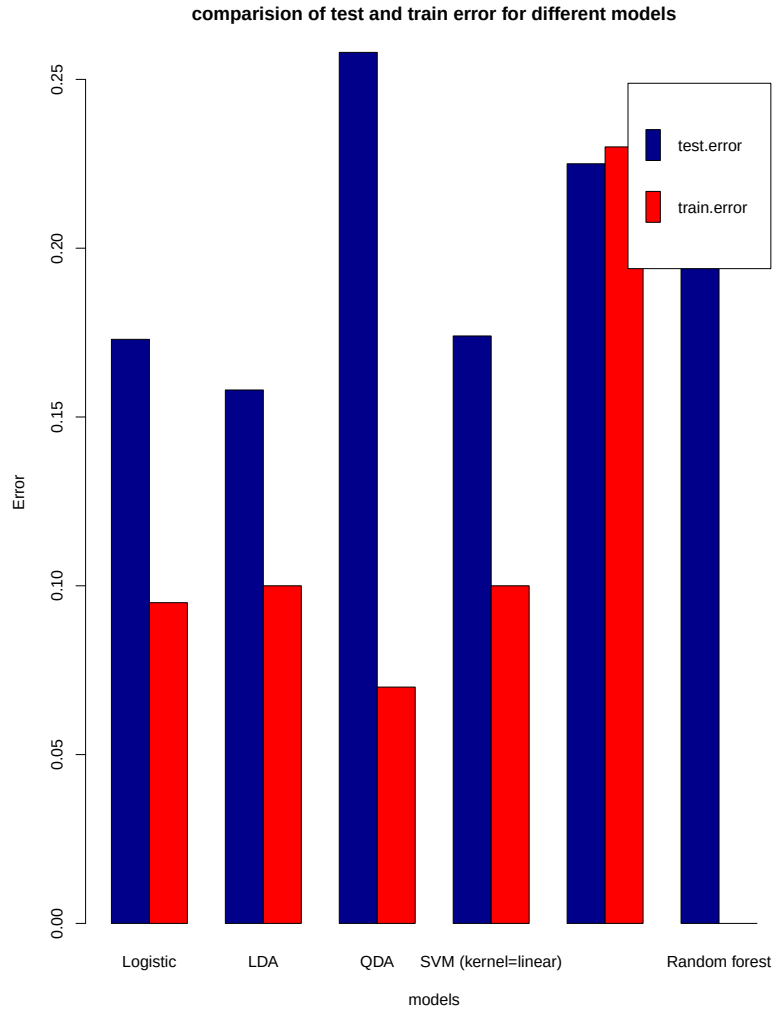
$$K(x, xi) = 1 + sum(x * xi)^d$$

  Where d is the dimension of the polynomial

- **Radial kernel:** Most of time we used the radial kernel. Radial kernel is defined as:

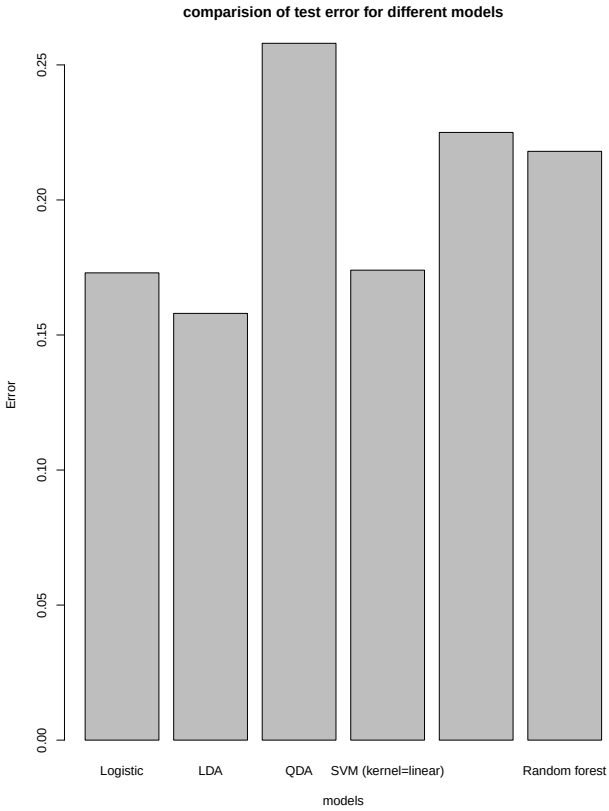
$$k(x, xi) = exp(-gamma * sum(x - xi)^2)$$

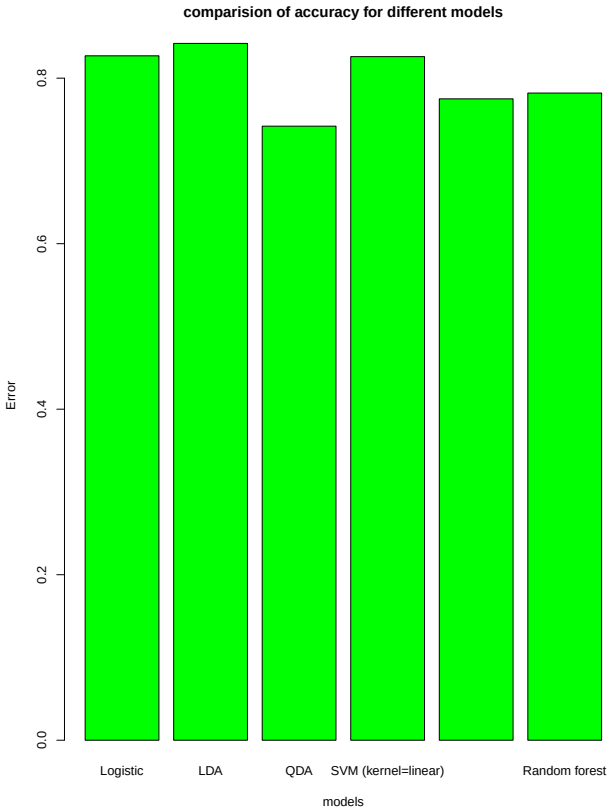
8 Comparison between applied classification model



This plot is comparing the training and test error for different model and from this it is clear that for all case model is working well in traininig data in comparison to test

```
> error_matrix    # contain misclassification error of different model on test set and train set
            Logistic   LDA   QDA SVM (kernel=linear) KNN (k=83) Random forest
train error    0.095 0.100 0.070               0.100      0.230         0.000
test error     0.173 0.158 0.258               0.174      0.225         0.218
> accuray_mat     # contain Accuracy of different model on test set and train set
                  Logistic  LDA  QDA SVM (kernel=linear) KNN (k=83) Random forest
training accuracy     90.5 90.0 93.0                90.0       77.0         100.0
test accuracy         82.7 84.2 74.2                82.6       77.5          78.2
```





data which is as usual and expected.

From the above table and plots it is clear that Logistic Regression, LDA and SVM with linear kernel are performing approximately same but Linear Discriminant Analysis (LDA) is the best model as it has lowest misclassification error on test data or highest accuracy on test data.
So, best model among all is LDA

9 Why LDA Performing Well:-

First of all we will state the assumptions of LDA and then we will check these assumption of LDA. As we have more than one (20 in our case) predictors so we discussed assumption for p>1.

Assumption
1− When we have multiple predictors (p>1) where p is no of predictors, then while applying LDA we assume X=$(X_1, X_2, ..., X_p)$ is drawn from Multivariate Gaussian distribution.
2− The LDA classifier assume that the observation in the kth class are drawn from Multivariate Gaussian distribution N(μ_k,$\sum$), where μ_k is class specific mean vector for the k^{th} class and $\sum$ is variance - covariance matrix which is common to all K classes.
· **Check for first assumption**
Now First we check (1) normality assumption of X=$(X_1, X_2, ..., X_p)$.

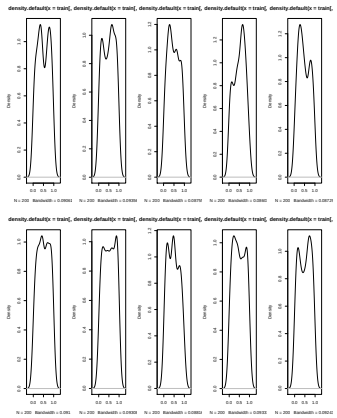


Figure 1: density plot of predictors

From the density plot of each predictor we can say that predictors are approximately normally distributed. Also we apply shapiro test to all predictors to check the normality. As shapiro test provides very high p value which clearly indicate that all predictors are normally distributed.

From the scatter diagram of different predictors, it can be seen that predictors are approximately uncorrelated and we also know that predictors are normal so we can say that predictors are approximately independent.
So all predictors $x_{i]}$ are normal and independent so X=$(X_1, X_2, ..., X_p)$ is follow Multivariate Gaussian distribution.

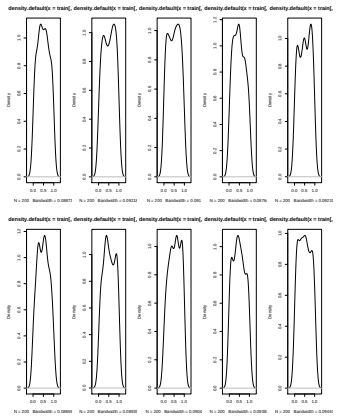


Figure 2: density plot of predictors

```
> t(p_val)        # contain p val of shapiro test of all predictors
      [,1]      [,2]      [,3]      [,4]      [,5]      [,6]      [,7]      [,8]      [,9]      [,10]     [,11]
[1,] 0.9449006 0.9437783 0.9554964 0.9549113 0.9593876 0.9535088 0.9421249 0.9560258 0.9451453 0.9414429 0.9599657
     [,12]     [,13]     [,14]     [,15]     [,16]     [,17]     [,18]     [,19]     [,20]
[1,] 0.9496055 0.9508644 0.9596614 0.9421084 0.9657632 0.9531279 0.9533404 0.9501686 0.9472654
```

Figure 3: p values of predictors of shapiro test

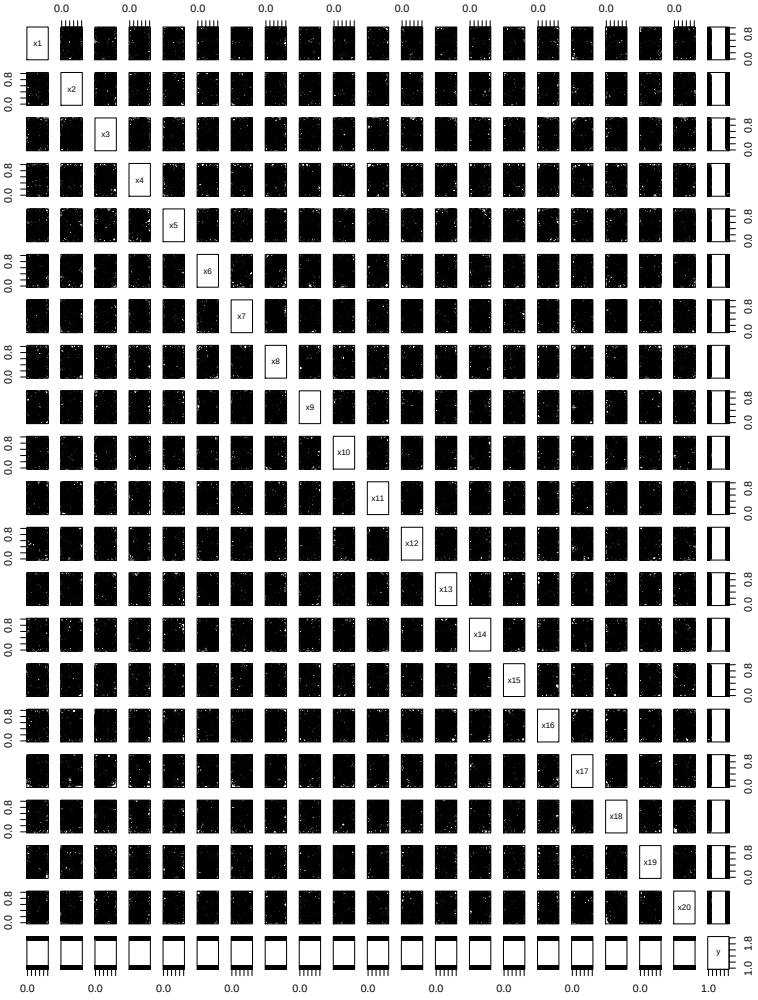


Figure 4: scatter plot between different predictors

· **Check for second assumption**

Now we Separate the train data according to class level 0 and 1 that is all the observation corresponding class level 0 forms one data subset and rest observation corresponding to class level 1 forms another class level.Now We class speccific mean vector for both data subset and variance-covariance matrix $\sum_k$.Now we can easily observed that these variance-covariance matrix $\sum_k$ are approximately same for both class k=0,1.(hare we subtract the both variance-covariance matrix $\sum_k$ and found that each element of resultant matrix is very clase to zero which ensure that both variance-covariance matrix $\sum_k$ are closely same).

As our data satisfied approximately both assumption of LDA that's why LDA is performing relatively well over tha data.

10 Interpretation of ROC curve

11 Conclusion

: $\dot{-}$ We fit the Logistic, LDA,QDA,KNN,SVM and Random Forest classification model on the training model, predict response variable for test data and compare test misclassification error for different model.

: $\dot{-}$ LDA has minimum misclassification error on test data that is .158 or highest accuracy that is 84.20 percent among all models.

: $\dot{-}$ LDA has 0.1 misclassification error on training data set which is less that misclassification error on test data.

; $\dot{-}$

12 References:-

(1) An Introduction to Statistical Learning with Appli- cations in R by Gareth James, Daniela Witten, Trevor Hastie and Robert Tibshirani.
(2) The Elements of Statistical Learning:Data Mining, Inference, and Prediction.
(3) www.analyticsvidhya.com
(4) stats.stackexchange.com
(5) datasciencebeginners.com
(6) geeksforgeeks

13 Contribution

ContributioninTechnicalwork

Abhilash (191003)- Worked on data preparation, KNN,Random Forest,SVM and model comparison.

Ankit (191016)- Worked on Logistic Regression,LDA,QDA,Interpretation of ROC curve and "Why LDA working well".

Contribution in Project Report.

Both equally participated in making the project report.