Linear Discriminant Analysis

Mowniesh Asokan(mowas455)

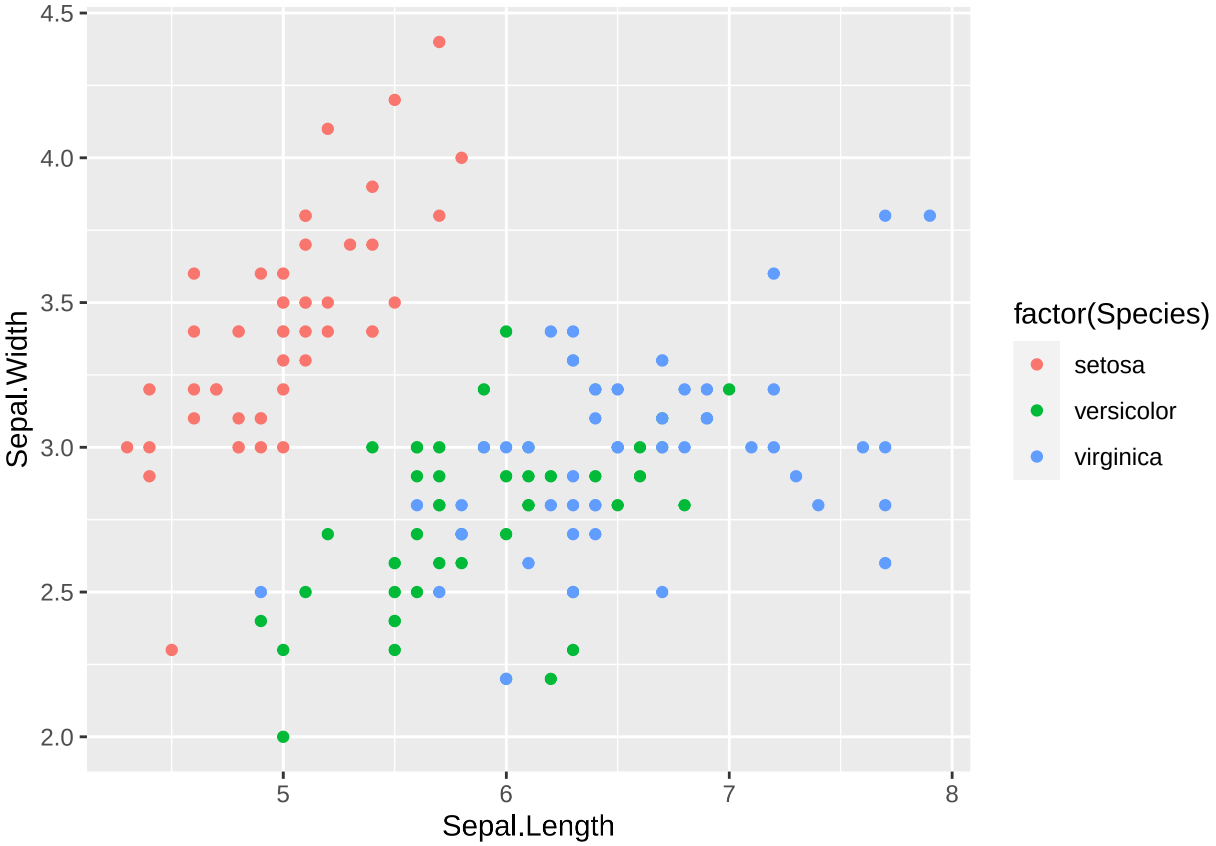
12/6/2020

Assignment 1. LDA and logistic regression

1.

Actually LDA tries to reduce dimensions of the feature set while retaining the information that discriminates output classes. LDA tries to find a decision boundary around each cluster of a class.The goal of a LDA is often to project a feature space (a data-set n-dimensional samples) into a smaller subspace k.

- It is linearly non-separable
- Iris Setosa is linearly separable from the other two classes.so,that we can draw a line or hyper-plane to classify a each groups.



2.

R functions only to implement Linear Discriminant Analysis between the three species based on variables Sepal Length and Sepal Width: Setosa -1 Versicolor-2 Virginca -3

2a.Mean, Covariance matrices (use cov()) and Prior probabilities per class

```
## [1] "Prior probabilities of groups:"

## [1] 0.3333333 0.3333333 0.3333333

## [1] "Group means:"

##     [,1]  [,2]  [,3]
## x1 5.006 5.936 6.588
## x2 3.428 2.770 2.974

## [1] "covariance matrix of group 1"

##            x1         x2
## x1 0.12424898 0.09921633
## x2 0.09921633 0.14368980

## [1] "covariance matrix of group 2"

##            x1         x2
## x1 0.26643265 0.08518367
## x2 0.08518367 0.09846939

## [1] "covariance matrix of group 3"

##            x1         x2
## x1 0.40434286 0.09376327
## x2 0.09376327 0.10400408
```

2b.Pooled Covariance Matrix

```
## [1] "Pooled covariance matrix of groups"

##            x1         x2
## x1 0.26500816 0.09272109
## x2 0.09272109 0.11538776
```

2c.Probabilistic Model for LDA

$$x|y = C_i, \mu_i, \Sigma \sim N(\mu_i, \Sigma)$$
$$y|\pi \sim Multinomial(\pi_1, ...\pi k)$$

2d.Discriminant Function

```
# Discriminant function

 disc_fn<-function(v,p_cv,m_g,p_g)
 {
   v<-as.matrix(v)
   p_cv<-solve(p_cv)
   d1<-((v%*%p_cv)%*%(m_g))
   d2<-(0.5*t(m_g))%*%(p_cv)%*%(m_g)
   t_d<-d1-(as.numeric(d2))+log(p_g)
   return(t_d)
 }
```

2e.Decision Boundary

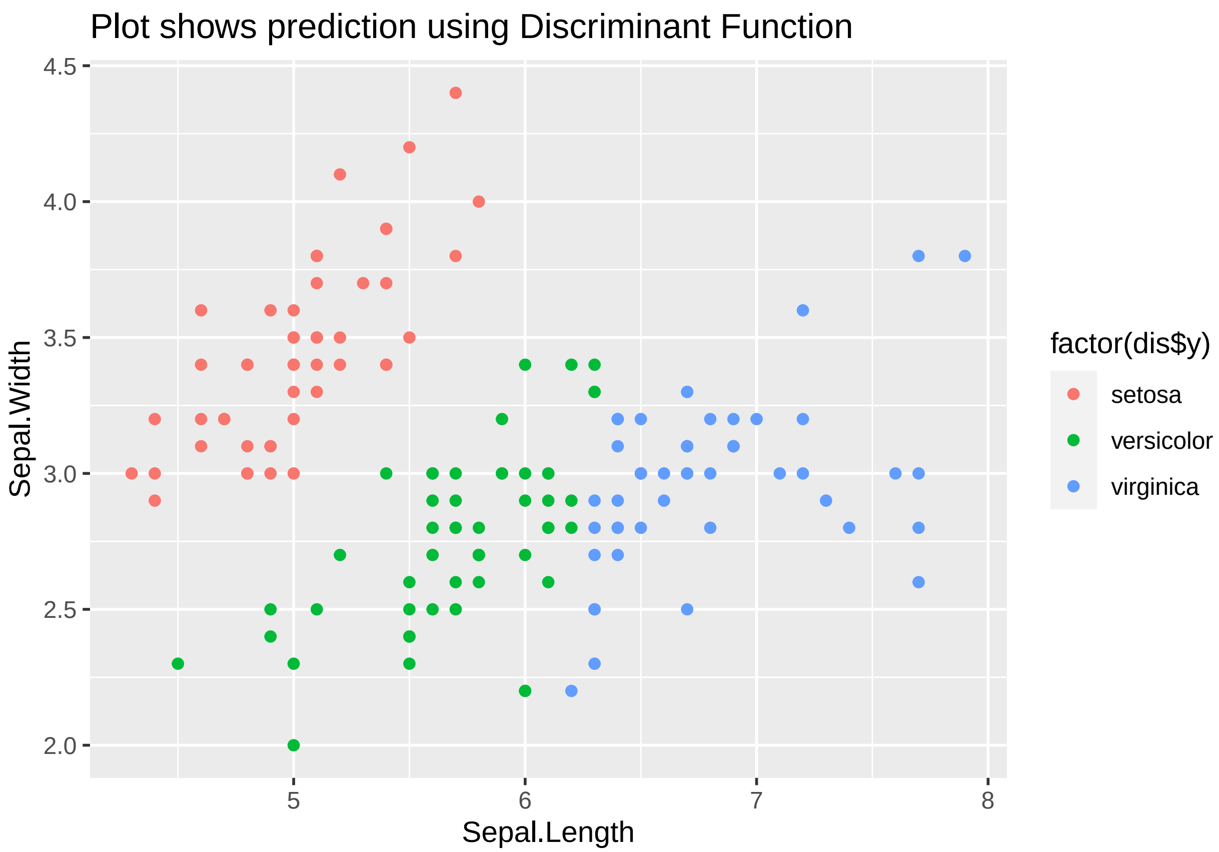
```
## Coefficients
```

```
##       Setosa Versicolor Virginica
## x1 47.343881   18.36513  12.21604
## x2 -8.833515   12.24331  17.58185
```

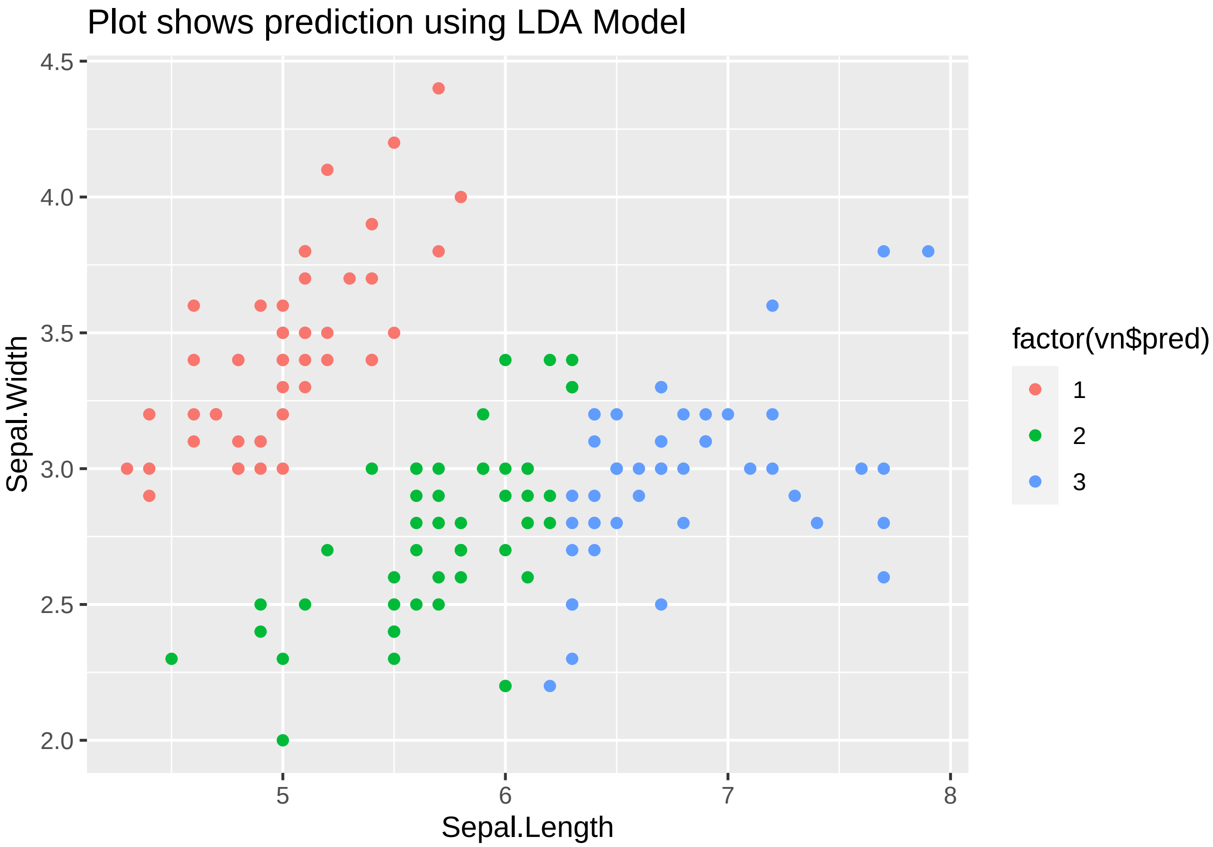
3.

Whether the error obtained using discriminant function is not same as error obtained by using LDA model, but the miss-classification rate of the two model are to be same.

- Both the model and function works perfectly for the class1 (setosa) and class3(virginica) , but it confuse with the class2(versicolor)

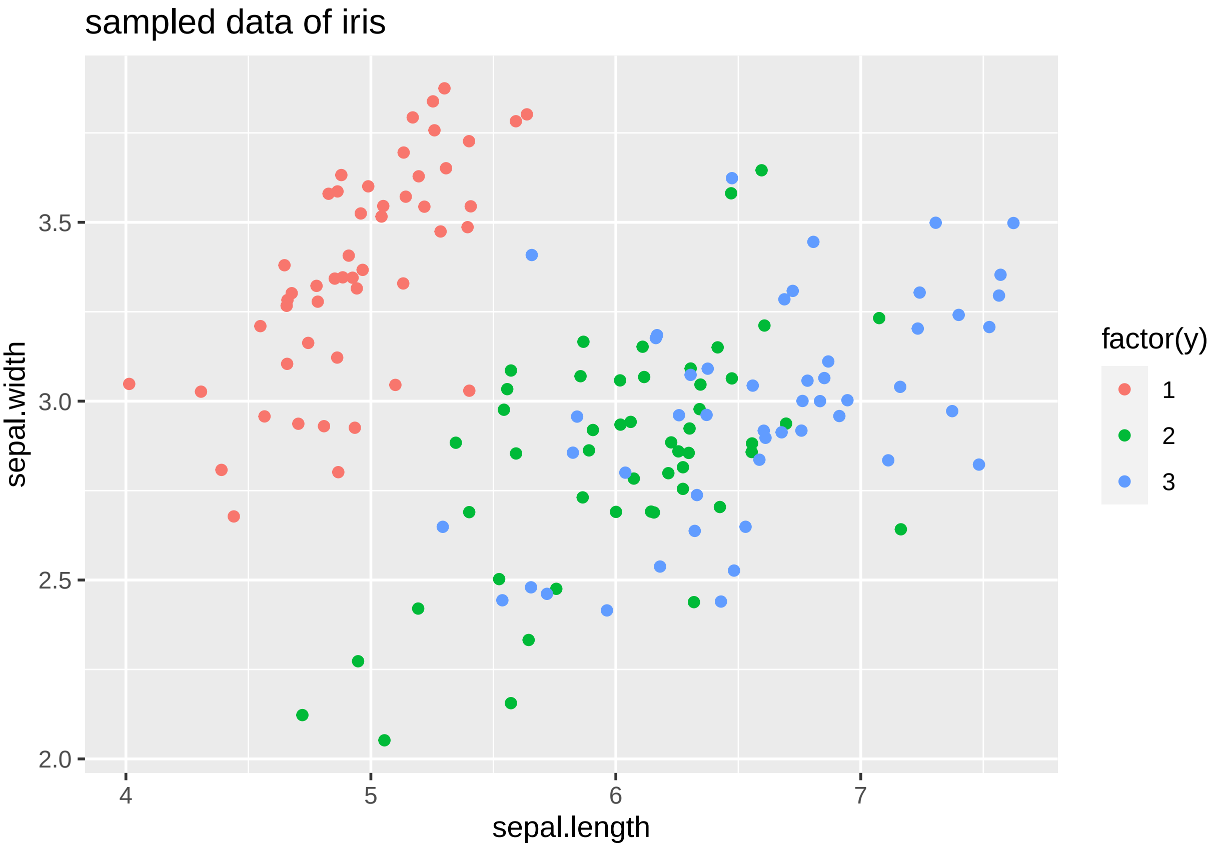


```
## Confusion Matrix of prediction using Discriminant function

##              y
##                1  2  3
##   setosa      49  0  0
##   versicolor   1 36 15
##   virginica    0 14 35

## Misclassification Rate using Discrimiant function

## [1] 0.2

## Confusion Matrix using LDA model

##
##       1  2  3
##   1 49  0  0
##   2  1 36 15
##   3  0 14 35

## Misclassification Rate using LDA model

## [1] 0.2
```



4.Sample the Data

Sample the data of iris using the multivariate normal distribution .By specifying the mean and sigma in the old dataset. we can generate a new sample from same mean and variance of each groups. so,the scatter plot of the generated data and the orginal data is looks similar.



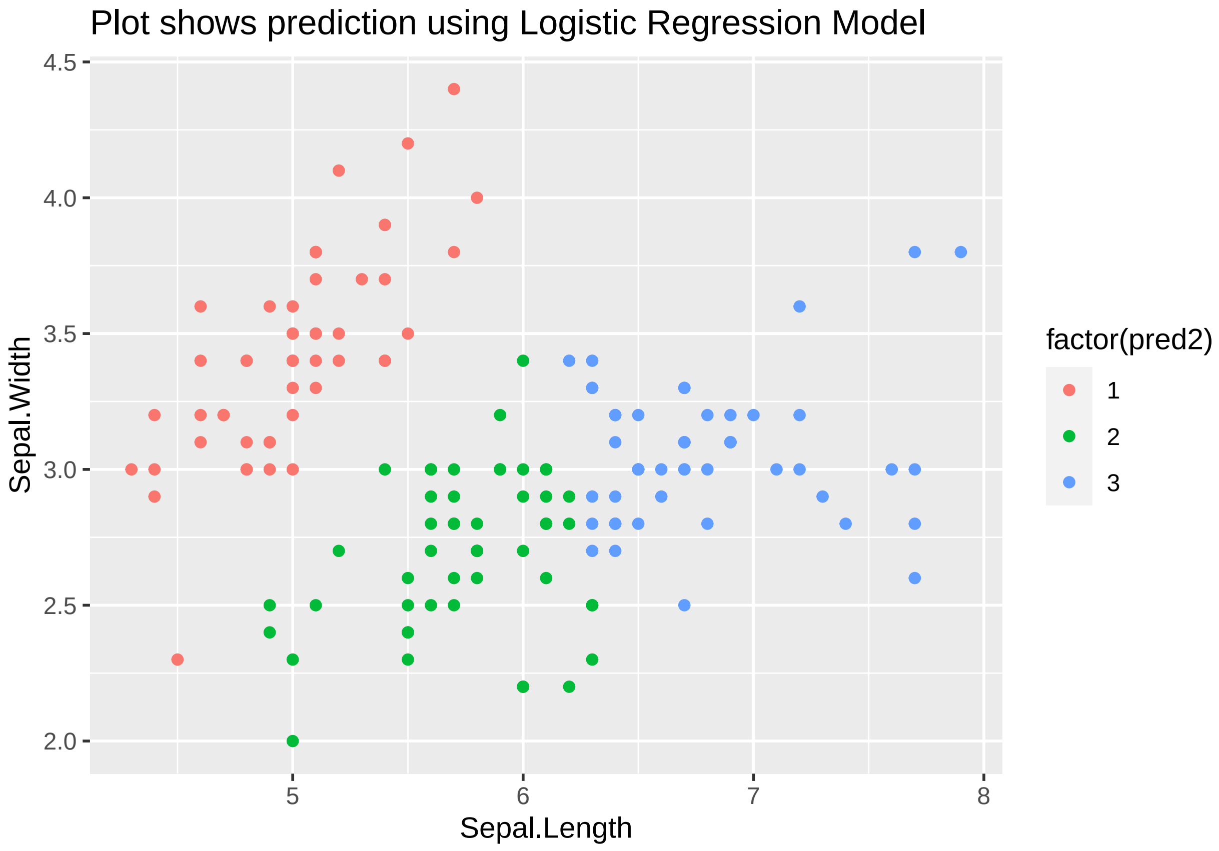
5.Logistic Regression model

Logistic regression measures the relationship between one or more independent variables (X) and the categorical dependent variable (Y) by estimating probabilities using a logistic(sigmoid) function[1].

- The logistic Regression model is works better than the lda model.Because the miss-classification Rate of this model small compare with LDA model.

```
## # weights:  12 (6 variable)
## initial  value 164.791843
## iter  10 value 62.715967
## iter  20 value 59.808291
## iter  30 value 55.445984
## iter  40 value 55.375704
## iter  50 value 55.346472
## iter  60 value 55.301707
## iter  70 value 55.253532
## iter  80 value 55.243230
## iter  90 value 55.230241
## iter 100 value 55.212479
## final  value 55.212479
## stopped after 100 iterations
```

```
##   [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [38] 1 1 1 1 1 1 1 1 1 1 1 1 1 3 3 3 2 3 2 3 2 3 2 2 2 2 2 2 3 2 2 2 2 2 2 2 2
##  [75] 3 3 3 3 2 2 2 2 2 2 2 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2 3 2 3 3 3 3 2 3 3 3 3
## [112] 3 3 2 2 3 3 3 3 2 3 2 3 3 3 3 2 2 3 3 3 3 3 3 2 3 3 3 2 3 3 3 2 3 3 3 2 3
## [149] 3 2
## Levels: 1 2 3
```



```
##             Reference
## Prediction   1  2  3
##           1 50  0  0
##           2  0 38 13
##           3  0 12 37
```

```
## Misclassification Rate using Logistic Regression model
```

```
## [1] 0.1666667
```

References

1.https://en.wikipedia.org/wiki/Logistic_regression

Appendix: All code for this report

```
knitr::opts_chunk$set(echo = TRUE)
library(ggplot2)
library(MASS)
library(mvtnorm)
data=iris
x0=c(data$Sepal.Length)
y0=c(data$Sepal.Width)
z=c(data$Species)
x = data.frame(x1=x0,x2=y0)
y=as.factor(z)

# scatter plot for original data with target
ggplot(data = data, aes(x = Sepal.Length,y = Sepal.Width)) +geom_point(aes(color = factor(Species)))
# Grouping the Targets
  group1_index = which( y == 1 )
  group2_index = which( y == 2 )
  group3_index = which( y == 3 )

#priors:
  prior_group1 = length(group1_index) / length(y)
  prior_group2 = length(group2_index) / length(y)
  prior_group3 = length(group3_index) / length(y)
  print("Prior probabilities of groups:")
  print(c(prior_group1, prior_group2,prior_group3))

#means:
  mean_group1 = as.matrix(colMeans(x[group1_index, ]))
  mean_group2 = as.matrix(colMeans(x[group2_index, ]))
  mean_group3 = as.matrix(colMeans(x[group3_index, ]))

  print("Group means:")
  print(cbind(mean_group1, mean_group2,mean_group3))

# Covariance Matrix
  cv1<-cov(x[group1_index, ])
  cv2<-cov(x[group2_index, ])
  cv3<-cov(x[group3_index, ])
  print("covariance matrix of group 1")
  print(cv1)

  print("covariance matrix of group 2")
  print(cv2)

  print("covariance matrix of group 3")
  print(cv3)
# Pooled Co-variance Matrix
  pooled_cv<-as.matrix((length(group1_index)*cv1)+(length(group2_index)*cv2)+
                         (length(group3_index)*cv3))/length(y)
  print("Pooled covariance matrix of groups")
  print(pooled_cv)
```

```
# Discriminant function

 disc_fn<-function(v,p_cv,m_g,p_g)
 {
   v<-as.matrix(v)
   p_cv<-solve(p_cv)
   d1<-((v%*%p_cv)%*%(m_g))
   d2<-(0.5*t(m_g))%*%(p_cv)%*%(m_g)
   t_d<-d1-(as.numeric(d2))+log(p_g)
   return(t_d)
 }

# Decision Boundary Function
decision_boundary<-function(x,y,z)
{
  y<-solve(y)
  m<-(-0.5*t(x))%*%(y)%*%(x)
  w_oi<-m+log(z)
  w_i<-y%*%x
  #cat("w_Oi",w_oi,"\n","w_i",w_i,"\n")
}

dc1<- decision_boundary(mean_group1,cv1,prior_group1)   #decision_bndy value of group 1
dc2<- decision_boundary(mean_group2,cv2,prior_group2)   #decision_bndy value of group 2
dc3<- decision_boundary(mean_group3,cv3,prior_group3)   #decision_bndy value of group 3

dc_b<-data.frame(dc1,dc2,dc3)
colnames(dc_b)<-c("Setosa","Versicolor","Virginica")
cat("Coefficients")
dc_b
#Bind the Discriminant values in the data.frame

 d_val1<- disc_fn(x,pooled_cv,mean_group1,prior_group1)
 d_val2<- disc_fn(x,pooled_cv,mean_group2,prior_group2)
 d_val3<- disc_fn(x,pooled_cv,mean_group3,prior_group3)

 disc_val<-cbind(d_val1,d_val2,d_val3)
 dis<-as.data.frame(disc_val)

colnames(dis)<-c("setosa","versicolor","virginica")

# Found the target value of this data by max value occurs on the row

  m<-colnames(dis)[max.col(dis, ties.method = "first")]

  dis$y<-m     # add those values as y in the same dataframe



 ggplot(data = data, aes(x = Sepal.Length,y = Sepal.Width)) +
  geom_point(aes(color = factor(dis$y))) + ggtitle("Plot shows prediction using Discriminant Function")

#confusion matrix
```

```
cat("Confusion Matrix of prediction using Discriminant function")
 table(as.factor(dis$y), y)

#miss-classification rate
missclassrate=function(y,y_i)
{
  n=length(y)
  v<-1-(sum(diag(table(y,y_i)))/n)
  return(v)
}
ms=missclassrate(as.factor(dis$y),y)

cat("Misclassification Rate using Discrimiant function","\n")
ms
#LDA Model
  fit_lda <- lda(y~., data = x)
  # ca("coefficients")
  # coef(fit_lda)
  pred_lda <- predict(fit_lda, x)
  vn<-data.frame(original = y, pred = pred_lda$class)
  cat("Confusion Matrix using LDA model")
  table(vn$pred,vn$original)
  ldms=missclassrate(as.factor(vn$pred),y)
  cat("Misclassification Rate using LDA model","\n")
  ldms

  ggplot(data = data, aes(x = Sepal.Length,y = Sepal.Width)) +
   geom_point(aes(color = factor(vn$pred)))+ ggtitle("Plot shows prediction using LDA Model")
# bind the all groups in a single dataframe
  bvn1 <- rmvnorm(50, mean = mean_group1, sigma = cv1 )
  bvn2 <- rmvnorm(50, mean = mean_group2, sigma = cv2 )
  bvn3 <- rmvnorm(50, mean = mean_group3, sigma = cv3 )

  bvn<-rbind(bvn1,bvn2,bvn3)
  bvn<-as.data.frame(bvn)
  bvn$y<-y

  sample_data<-bvn[sample(nrow(bvn), 150), ]

  colnames(bvn)<-c("sepal.length","sepal.width","species")


  ggplot(data =bvn, aes(x = sepal.length,y = sepal.width)) +
    geom_point(aes(color = factor(y))) + ggtitle("sampled data of iris")



### 5.Logistic Regression Model

 library(nnet)
 irisModel<-multinom(y~x1+x2 ,data =  x)
  m<-summary(irisModel)
```

```
pred2<-predict(irisModel,x)
pred2

ggplot(data = data, aes(x = Sepal.Length,y = Sepal.Width)) +
  geom_point(aes(color = factor(pred2)))+ ggtitle("Plot shows prediction using Logistic Regression Mo

table("Prediction"=as.factor(pred2), "Reference"=y)   # confusiom Matrix

lms=missclassrate(as.factor(pred2),y)
cat("Misclassification Rate using Logistic Regression model","\n")
lms
```