Homework 3

Group BUAN635.501-1

04/02/2020

CLASS: "BUAN 6356"
GROUP MEMBERS: "Sai Raghavendra Sridhar(sxs180281), Shreya Tippannawar(sst190000), Smruti Viswanath Iyer(sxi180001), Piyush Dangwal(pxd142430),Shanshan Luo(sxl130330)"

a. Load the packages:

```
if(!require("pacman")) install.packages("pacman")
```

```
## Loading required package: pacman
```

```
pacman::p_load(caret, data.table, MASS, ggplot2, dplyr, gains, AUC)
search()
```

```
##  [1] ".GlobalEnv"         "package:AUC"        "package:gains"
##  [4] "package:dplyr"      "package:MASS"       "package:data.table"
##  [7] "package:caret"      "package:ggplot2"    "package:lattice"
## [10] "package:pacman"     "package:stats"      "package:graphics"
## [13] "package:grDevices"  "package:utils"      "package:datasets"
## [16] "package:methods"    "Autoloads"          "package:base"
```

b. Read in the data from "spambase data":

```
options(digits = 3)
options(scipen=999)
# Load data
spambase <- fread("spambase.data")

# Load Column Names for the data.table
cnames = read.table("spambase.names", comment.char="|", header=F)[1]
cnames = gsub(":.*", "", as.matrix(cnames))
cnames = c(cnames[c(2:nrow(cnames))],"spam")
colnames(spambase) = cnames
spambase$spam <- ifelse(spambase$spam==0,"Regular","Spam")
spambase$spam <- as.factor(spambase$spam)


head(spambase)
```

```
##    word_freq_make word_freq_address word_freq_all word_freq_3d word_freq_our
## 1:           0.00              0.64          0.64            0          0.32
## 2:           0.21              0.28          0.50            0          0.14
## 3:           0.06              0.00          0.71            0          1.23
## 4:           0.00              0.00          0.00            0          0.63
## 5:           0.00              0.00          0.00            0          0.63
## 6:           0.00              0.00          0.00            0          1.85
##    word_freq_over word_freq_remove word_freq_internet word_freq_order
## 1:           0.00             0.00               0.00           0.00
## 2:           0.28             0.21               0.07           0.00
## 3:           0.19             0.19               0.12           0.64
## 4:           0.00             0.31               0.63           0.31
## 5:           0.00             0.31               0.63           0.31
## 6:           0.00             0.00               1.85           0.00
##    word_freq_mail word_freq_receive word_freq_will word_freq_people
## 1:           0.00              0.00           0.64            0.00
## 2:           0.94              0.21           0.79            0.65
## 3:           0.25              0.38           0.45            0.12
## 4:           0.63              0.31           0.31            0.31
## 5:           0.63              0.31           0.31            0.31
## 6:           0.00              0.00           0.00            0.00
##    word_freq_report word_freq_addresses word_freq_free word_freq_business
## 1:             0.00                0.00           0.32               0.00
## 2:             0.21                0.14           0.14               0.07
## 3:             0.00                1.75           0.06               0.06
## 4:             0.00                0.00           0.31               0.00
## 5:             0.00                0.00           0.31               0.00
## 6:             0.00                0.00           0.00               0.00
##    word_freq_email word_freq_you word_freq_credit word_freq_your word_freq_font
## 1:            1.29          1.93             0.00           0.96              0
## 2:            0.28          3.47             0.00           1.59              0
## 3:            1.03          1.36             0.32           0.51              0
## 4:            0.00          3.18             0.00           0.31              0
## 5:            0.00          3.18             0.00           0.31              0
## 6:            0.00          0.00             0.00           0.00              0
##    word_freq_000 word_freq_money word_freq_hp word_freq_hpl word_freq_george
## 1:          0.00            0.00            0             0                0
## 2:          0.43            0.43            0             0                0
## 3:          1.16            0.06            0             0                0
## 4:          0.00            0.00            0             0                0
## 5:          0.00            0.00            0             0                0
## 6:          0.00            0.00            0             0                0
##    word_freq_650 word_freq_lab word_freq_labs word_freq_telnet word_freq_857
## 1:             0             0              0                0             0
## 2:             0             0              0                0             0
## 3:             0             0              0                0             0
## 4:             0             0              0                0             0
## 5:             0             0              0                0             0
## 6:             0             0              0                0             0
##    word_freq_data word_freq_415 word_freq_85 word_freq_technology
## 1:              0             0           0                    0
## 2:              0             0           0                    0
## 3:              0             0           0                    0
## 4:              0             0           0                    0
```

```
## 5:              0            0           0                   0
## 6:              0            0           0                   0
##    word_freq_1999 word_freq_parts word_freq_pm word_freq_direct word_freq_cs
## 1:           0.00               0           0             0.00            0
## 2:           0.07               0           0             0.00            0
## 3:           0.00               0           0             0.06            0
## 4:           0.00               0           0             0.00            0
## 5:           0.00               0           0             0.00            0
## 6:           0.00               0           0             0.00            0
##    word_freq_meeting word_freq_original word_freq_project word_freq_re
## 1:                 0              0.00                 0        0.00
## 2:                 0              0.00                 0        0.00
## 3:                 0              0.12                 0        0.06
## 4:                 0              0.00                 0        0.00
## 5:                 0              0.00                 0        0.00
## 6:                 0              0.00                 0        0.00
##    word_freq_edu word_freq_table word_freq_conference char_freq_; char_freq_(
## 1:          0.00               0                    0       0.00       0.000
## 2:          0.00               0                    0       0.00       0.132
## 3:          0.06               0                    0       0.01       0.143
## 4:          0.00               0                    0       0.00       0.137
## 5:          0.00               0                    0       0.00       0.135
## 6:          0.00               0                    0       0.00       0.223
##    char_freq_[ char_freq_! char_freq_$ char_freq_# capital_run_length_average
## 1:           0       0.778       0.000       0.000                      3.76
## 2:           0       0.372       0.180       0.048                      5.11
## 3:           0       0.276       0.184       0.010                      9.82
## 4:           0       0.137       0.000       0.000                      3.54
## 5:           0       0.135       0.000       0.000                      3.54
## 6:           0       0.000       0.000       0.000                      3.00
##    capital_run_length_longest capital_run_length_total spam
## 1:                         61                     278 Spam
## 2:                        101                    1028 Spam
## 3:                        485                    2259 Spam
## 4:                         40                     191 Spam
## 5:                         40                     191 Spam
## 6:                         15                      54 Spam
```

```
dim(spambase)
```

```
## [1] 4601   58
```

```
str(spambase)
```

```
## Classes 'data.table' and 'data.frame':   4601 obs. of  58 variables:
##  $ word_freq_make           : num  0 0.21 0.06 0 0 0 0 0 0.15 0.06 ...
##  $ word_freq_address        : num  0.64 0.28 0 0 0 0 0 0 0 0.12 ...
##  $ word_freq_all            : num  0.64 0.5 0.71 0 0 0 0 0 0.46 0.77 ...
##  $ word_freq_3d             : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ word_freq_our            : num  0.32 0.14 1.23 0.63 0.63 1.85 1.92 1.88 0.61 0.19 ...
##  $ word_freq_over           : num  0 0.28 0.19 0 0 0 0 0 0 0.32 ...
##  $ word_freq_remove         : num  0 0.21 0.19 0.31 0.31 0 0 0 0.3 0.38 ...
##  $ word_freq_internet       : num  0 0.07 0.12 0.63 0.63 1.85 0 1.88 0 0 ...
```

```
##  $ word_freq_order          : num  0 0 0.64 0.31 0.31 0 0 0 0.92 0.06 ...
##  $ word_freq_mail           : num  0 0.94 0.25 0.63 0.63 0 0.64 0 0.76 0 ...
##  $ word_freq_receive        : num  0 0.21 0.38 0.31 0.31 0 0.96 0 0.76 0 ...
##  $ word_freq_will           : num  0.64 0.79 0.45 0.31 0.31 0 1.28 0 0.92 0.64 ...
##  $ word_freq_people         : num  0 0.65 0.12 0.31 0.31 0 0 0 0 0.25 ...
##  $ word_freq_report         : num  0 0.21 0 0 0 0 0 0 0 0 ...
##  $ word_freq_addresses      : num  0 0.14 1.75 0 0 0 0 0 0 0.12 ...
##  $ word_freq_free           : num  0.32 0.14 0.06 0.31 0.31 0 0.96 0 0 0 ...
##  $ word_freq_business       : num  0 0.07 0.06 0 0 0 0 0 0 0 ...
##  $ word_freq_email          : num  1.29 0.28 1.03 0 0 0 0.32 0 0.15 0.12 ...
##  $ word_freq_you            : num  1.93 3.47 1.36 3.18 3.18 0 3.85 0 1.23 1.67 ...
##  $ word_freq_credit         : num  0 0 0.32 0 0 0 0 0 3.53 0.06 ...
##  $ word_freq_your           : num  0.96 1.59 0.51 0.31 0.31 0 0.64 0 2 0.71 ...
##  $ word_freq_font           : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ word_freq_000            : num  0 0.43 1.16 0 0 0 0 0 0 0.19 ...
##  $ word_freq_money          : num  0 0.43 0.06 0 0 0 0 0 0.15 0 ...
##  $ word_freq_hp             : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ word_freq_hpl            : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ word_freq_george         : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ word_freq_650            : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ word_freq_lab            : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ word_freq_labs           : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ word_freq_telnet         : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ word_freq_857            : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ word_freq_data           : num  0 0 0 0 0 0 0 0 0.15 0 ...
##  $ word_freq_415            : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ word_freq_85             : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ word_freq_technology     : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ word_freq_1999           : num  0 0.07 0 0 0 0 0 0 0 0 ...
##  $ word_freq_parts          : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ word_freq_pm             : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ word_freq_direct         : num  0 0 0.06 0 0 0 0 0 0 0 ...
##  $ word_freq_cs             : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ word_freq_meeting        : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ word_freq_original       : num  0 0 0.12 0 0 0 0 0 0.3 0 ...
##  $ word_freq_project        : num  0 0 0 0 0 0 0 0 0 0.06 ...
##  $ word_freq_re             : num  0 0 0.06 0 0 0 0 0 0 0 ...
##  $ word_freq_edu            : num  0 0 0.06 0 0 0 0 0 0 0 ...
##  $ word_freq_table          : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ word_freq_conference     : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ char_freq_;              : num  0 0 0.01 0 0 0 0 0 0 0.04 ...
##  $ char_freq_(              : num  0 0.132 0.143 0.137 0.135 0.223 0.054 0.206 0.271 0.03 ...
##  $ char_freq_[              : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ char_freq_!              : num  0.778 0.372 0.276 0.137 0.135 0 0.164 0 0.181 0.244 ...
##  $ char_freq_$              : num  0 0.18 0.184 0 0 0 0.054 0 0.203 0.081 ...
##  $ char_freq_#              : num  0 0.048 0.01 0 0 0 0 0 0.022 0 ...
##  $ capital_run_length_average: num  3.76 5.11 9.82 3.54 3.54 ...
##  $ capital_run_length_longest: int  61 101 485 40 40 15 4 11 445 43 ...
##  $ capital_run_length_total  : int  278 1028 2259 191 191 54 112 49 1257 749 ...
##  $ spam                     : Factor w/ 2 levels "Regular","Spam": 2 2 2 2 2 2 2 2 2 2 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

####*Question 1 Examine how each predictor differs between the spam and non-spam e-mails by comparing the spam-class average and non-spam-class average. Identify 10 predictors for which the difference

between the spam-class average and non-spam class average is highest.

```
spamlda <- lda(spam~.,data=spambase)

ldameans <-spamlda$means

for (val in 1:57)
{
ldameans[2,val] <- abs(ldameans[2,val]-ldameans[1,val])
}
ldameansdiff <- ldameans[2,]
#ldameansdiff
names(ldameansdiff) <- cnames[1:57]
predictors<- tail(sort(ldameansdiff),10)
predictors
```

```
##                char_freq_!              word_freq_hpl
##                      0.404                      0.423
##             word_freq_free               word_freq_hp
##                      0.445                      0.878
##             word_freq_your              word_freq_you
##                      0.942                      0.994
##           word_freq_george capital_run_length_average
##                      1.264                      7.142
## capital_run_length_longest   capital_run_length_total
##                     86.179                    309.148
```

```
predictornames <- names(predictors)
predictornames
```

```
##  [1] "char_freq_!"                "word_freq_hpl"
##  [3] "word_freq_free"             "word_freq_hp"
##  [5] "word_freq_your"             "word_freq_you"
##  [7] "word_freq_george"           "capital_run_length_average"
##  [9] "capital_run_length_longest" "capital_run_length_total"
```

*Interpretation 1 - From the above output the top 10 predictors for which the difference between the spam and non-spam class is highest is in the following order

#1) capital_run_length_total #2) capital_run_length_longest
#3) capital_run_length_average #4) word_freq_george
#5) word_freq_you
#6) word_freq_your
#7) word_freq_hp
#8) word_freq_free
#9) word_freq_hpl
#10)char_freq_!

####*Question 2 :Perform a linear discriminant analysis using the training dataset. Include only 10 predictors identified in the question above in the model

```r
predictornames[11] <- "spam"
spambase1 <-spambase[,predictornames , with = FALSE]
# Split the data into training and validation/test set
set.seed(42)
training.index <- createDataPartition(spambase1$spam, p = 0.8, list = FALSE)
#training.index
spam.train <- spambase1[training.index, ]
spam.valid <- spambase1[-training.index, ]
# Normalize the data
# Estimate preprocessing parameters
norm.values <- preProcess(spam.train, method = c("center", "scale"))
# Transform the training and testing data using the estimated parameters
spam.train.norm <- predict(norm.values, spam.train)
spam.valid.norm <- predict(norm.values, spam.valid)

# Perform LDA
spam.lda <- lda(spam~., data = spam.train.norm)
spam.lda
```

```
## Call:
## lda(spam ~ ., data = spam.train.norm)
##
## Prior probabilities of groups:
## Regular    Spam
##   0.606   0.394
##
## Group means:
##         `char_freq_!` word_freq_hpl word_freq_free word_freq_hp word_freq_your
## Regular        -0.201         0.185         -0.203        0.206         -0.318
## Spam            0.309        -0.284          0.312       -0.317          0.489
##         word_freq_you word_freq_george capital_run_length_average
## Regular        -0.231            0.146                    -0.0853
## Spam            0.355           -0.224                     0.1311
##         capital_run_length_longest capital_run_length_total
## Regular                     -0.169                   -0.200
## Spam                         0.259                    0.308
##
## Coefficients of linear discriminants:
##                                LD1
## `char_freq_!`               0.3268
## word_freq_hpl              -0.1506
## word_freq_free              0.3875
## word_freq_hp               -0.2354
## word_freq_your              0.5715
## word_freq_you               0.2466
## word_freq_george           -0.2104
## capital_run_length_average  0.0527
## capital_run_length_longest  0.1297
## capital_run_length_total    0.3725
```

*Interpretation 2: LDA has been performed using normalized training dataset. Only 10 predictors are included as identified in the question above

####*Question 3 : What are the prior probabilities?

```
spam.lda$prior
```

```
## Regular    Spam
##   0.606   0.394
```

*Interpretation 3: The prior probabilties that we observe are 0.606 for Regular(Non-Spam) and 0.394 for spam

####*Question 4 : What are the coefficients of linear discriminants? Explain

```
spam.lda$scaling
```

```
##                                LD1
## `char_freq_!`               0.3268
## word_freq_hpl              -0.1506
## word_freq_free              0.3875
## word_freq_hp               -0.2354
## word_freq_your              0.5715
## word_freq_you               0.2466
## word_freq_george           -0.2104
## capital_run_length_average  0.0527
## capital_run_length_longest  0.1297
## capital_run_length_total    0.3725
```

Interpretation 4: Coefficients of linear discriminants are LD1 values. Here it represents weight of each variable among the total representation. Also we have only one LD1 for this dataset becuase coefficient of the linear discrimination is always one less than the number of classes. Here since the number of classes is 2 (spam and non- spam), So 2-1 = 1

####*Question 5 : Generate linear discriminants using your analysis. How are they used in classifying spams and non-spams?

```
pred.valid <- predict(spam.lda, spam.valid.norm)
head(pred.valid$x,10)
```

```
##         LD1
## 1  -0.84633
```

```
## 2   3.68654
## 3   0.83065
## 4  -0.56349
## 5   0.00457
## 6   1.40882
## 7   0.77136
## 8   0.71558
## 9   1.01091
## 10  0.15680
```

#*Interpretation 5: The LD1 that are the linear discriminants are generated above. A record is classified as spam and non spam based on the posterior probability and LD1 values.The LD1 corresponds to the amount of weights each of the entry suffices.The default cut off value is 0.5 and by respective calculations of LD1 values of respective rows amount to, the record is classified to its posterior probability and hence to its class.The LD1 values are obtained by statistical distance which is distance observed by records wirh centroid of various elements.When LD1 value is less than 0 posterior probability of recording falling into non-spam is more and LD1 is greater than 0 posterior probability of record falling into spam is more.

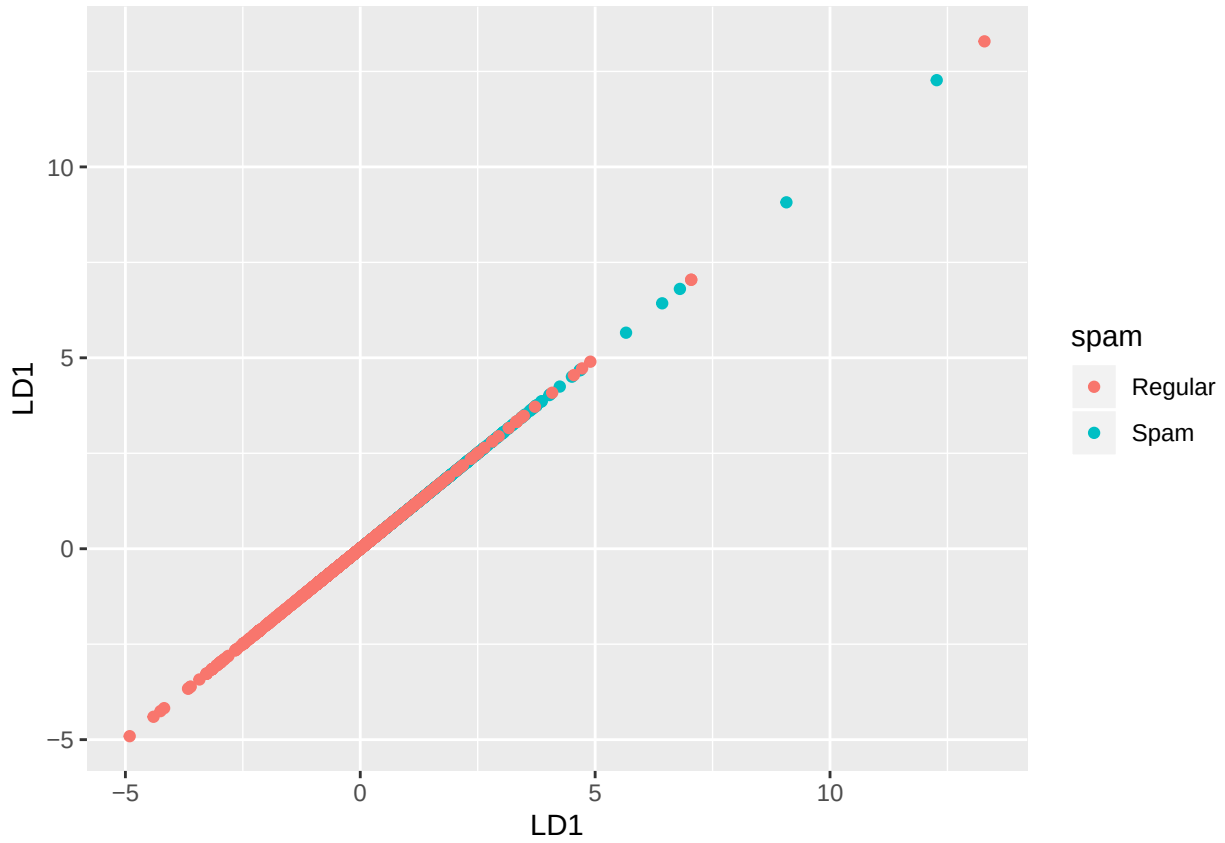
####*Question 6 :How many linear discriminants are in the model? Why?

Interpretation 6 : The number of linear discriminant in this model is 1.The linear discriminants would always be one less than the number of classes. Since the number of classes are two here i.e. - Spam and Non Spam , the number of linear discriminants is 2-1 = 1

####*Question 7 : Generate LDA plot using the training and validation data. What information is presented in these plots? How are they different?

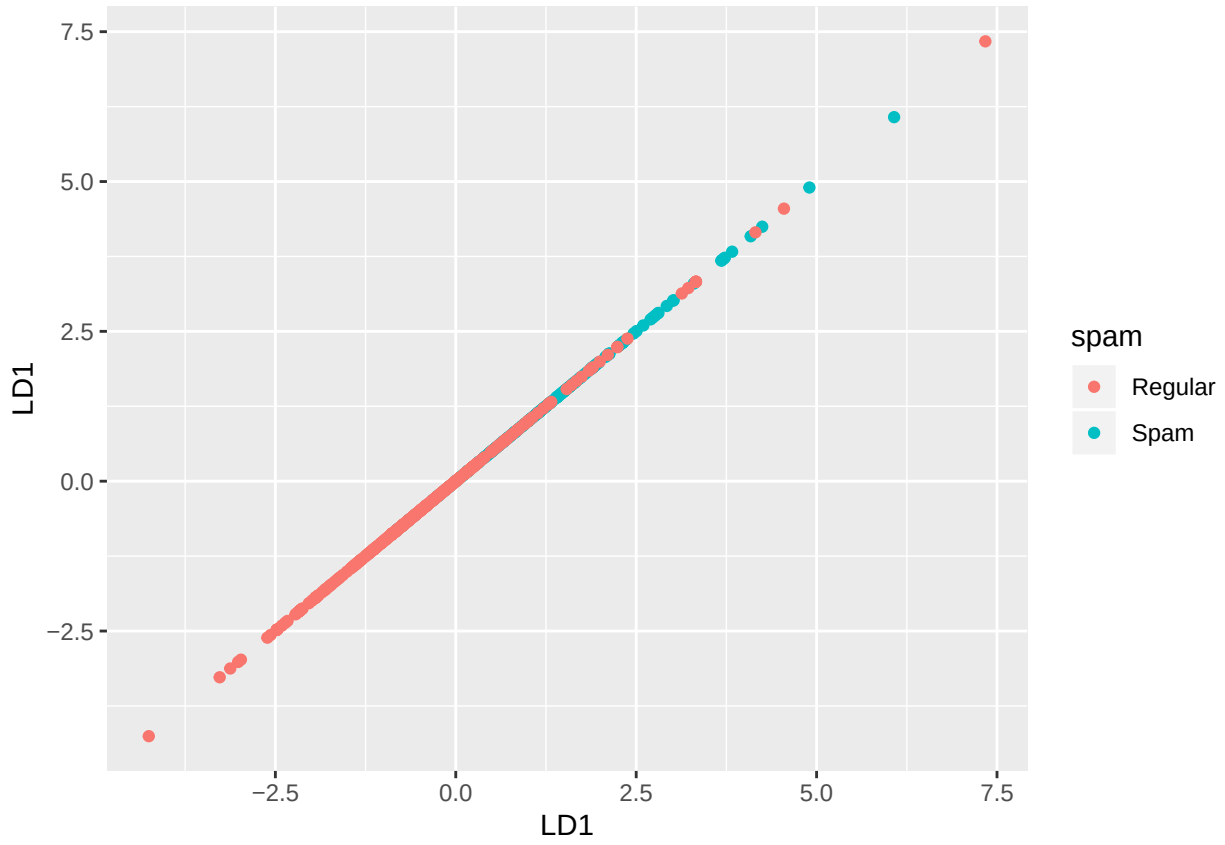
```
#Training data

pred.train <- predict(spam.lda, spam.train.norm)
lda.plot.train <- cbind(spam.train.norm, pred.train$x)
ggplot(lda.plot.train, aes(LD1, LD1)) +
geom_point(aes(color = spam))
```

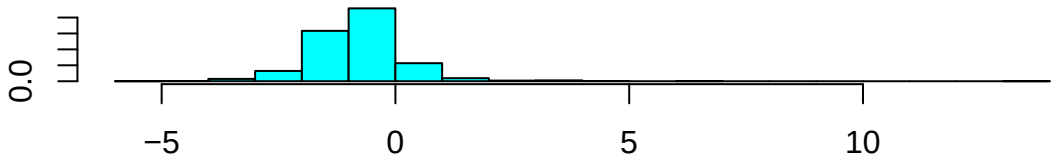


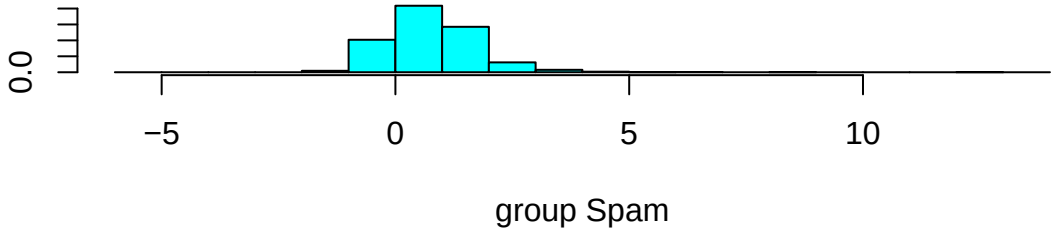

```
#Validation data

lda.plot.valid <- cbind(spam.valid.norm, predict(spam.lda, spam.valid.norm)$x)
ggplot(lda.plot.valid, aes(LD1, LD1)) +
geom_point(aes(color = spam))
```




```
plot(spam.lda)
```






```
pred.train <- predict(spam.lda, spam.train.norm)
```

#*Interpretation 7 : Here we can see most of the non-spam values are below zero and spam values are above zero in the histogram. There is very little amount of overlap between spam and non spam. From the scatter plot we can see its a straight line with positive slope. As LD1 increases, the posterior probability for record to be classified as spam increases. This can be shown in the graph through red where its regular(non-spam) and blue when its spam.

####*Question 8 : Generate the relevant confusion matrix. What are the sensitivity and specificity?

```
pred.valid <- predict(spam.lda, spam.valid.norm)
#Table creation for predicted vs actual
acc <- table(pred.valid$class, spam.valid.norm$spam)
confusionMatrix(acc)
```

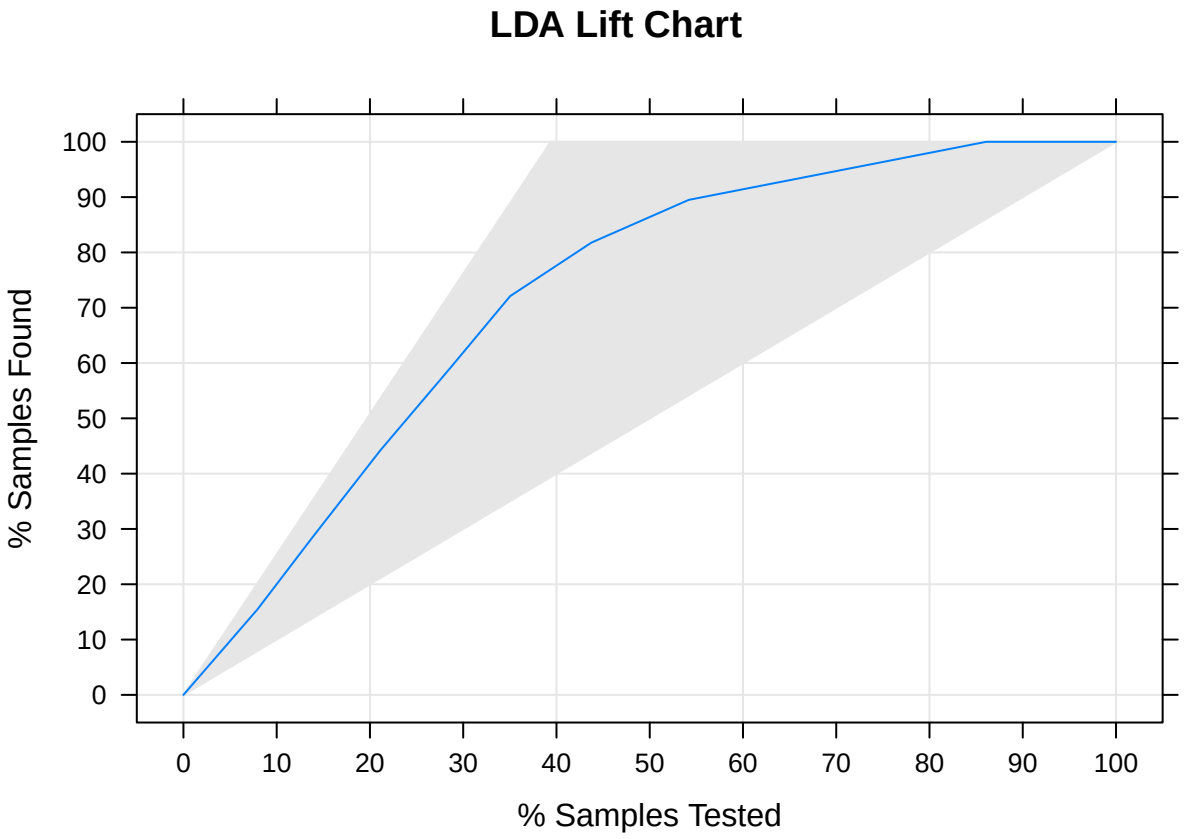
```
## Confusion Matrix and Statistics
##
##
##           Regular Spam
##   Regular     502  118
##   Spam         55  244
##
##                Accuracy : 0.812
##                  95% CI : (0.785, 0.837)
##     No Information Rate : 0.606
```

```
##     P-Value [Acc > NIR] : < 0.0000000000000002
##
##                   Kappa : 0.593
##
##  Mcnemar's Test P-Value : 0.00000243
##
##             Sensitivity : 0.901
##             Specificity : 0.674
##          Pos Pred Value : 0.810
##          Neg Pred Value : 0.816
##              Prevalence : 0.606
##          Detection Rate : 0.546
##    Detection Prevalence : 0.675
##       Balanced Accuracy : 0.788
##
##        'Positive' Class : Regular
##
```

#*Interpretation 8 : From above we get to know that accuracy is 0.812, while the specificity and senstivity are 0.674 and 0.901 respectively.

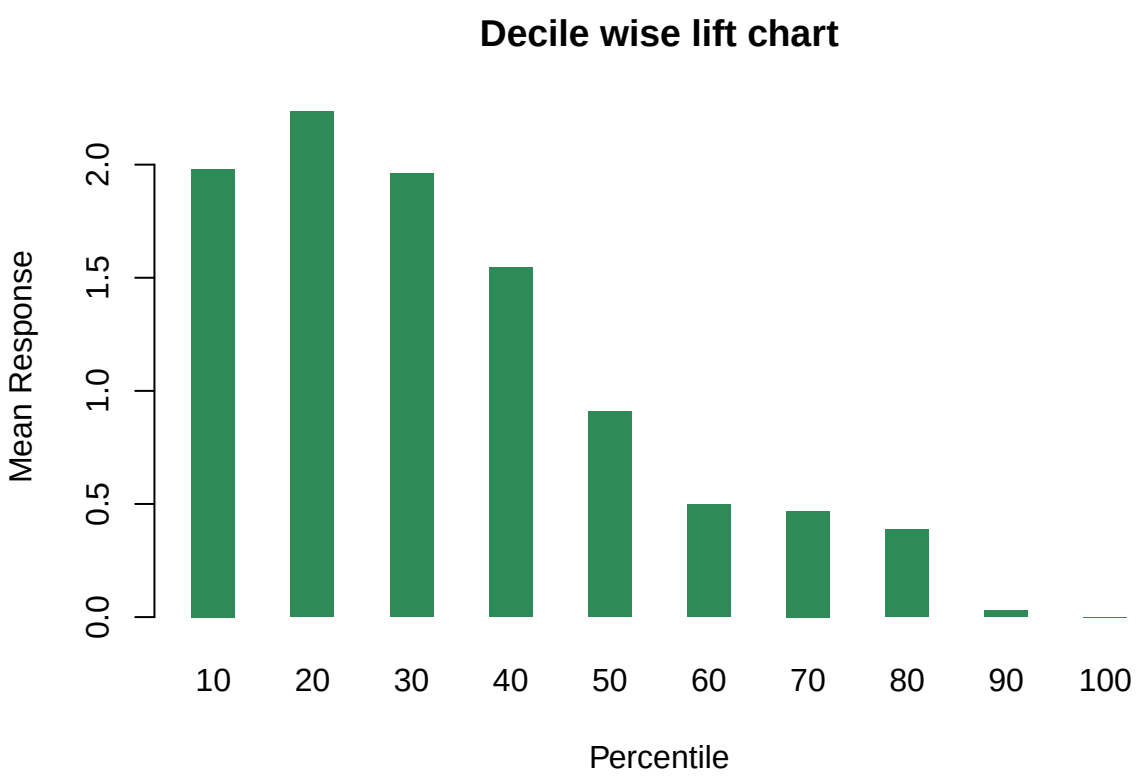
####*Question 9 :Generate lift and decile charts for the validation dataset and evaluate the effectiveness of the model in identifying spams.

```
pb <- pred.valid$posterior
pb <- as.data.frame(pb)
pred.LDA <- data.frame(spam.valid.norm$spam, pb[,2])
x <- as.data.frame(pred.valid$posterior)
y <- data.frame(spam.valid.norm$spam, x[,2])
colnames(y) <- c("x1","y1")
lift.ld <- lift(x1 ~ y1, data = y, cuts=10, class="Spam")
xyplot(lift.ld, main="LDA Lift Chart", type=c("l","g"), lwd=1,
scales=list(x=list(alternating=FALSE,tick.number = 10),
y=list(alternating=FALSE,tick.number = 10)))
```




```
# Decile chart

prob <- ifelse(spam.valid.norm$spam == "Spam", 1 ,0)
df_numeric <- data.frame(prob, pb$Spam)
colnames(df_numeric) <- c("Act","Probabilities")
#df_numeric
gain <- gains(df_numeric$Act, df_numeric$Probabilities)
barplot(gain$mean.resp / mean(df_numeric$Act), names.arg = gain$depth, xlab = "Percentile", space = 1.3
```



#Interpretation 9 : From LDA liftchart we get to know that our model outperforms the naive model.The % of samples which are predicted as SPAM are greater than with the baseline model with no predictors. However the blue line becomes flat only after crossing 80% of samples and hence the model prediction is ok. From the decile chart we get to know that model prediction is greater from the first few bars. So in the beginning we are able to predict model more accurately.Inorder to say that the model prediction is very good, the bars should slide from left to right. In our case since there is an exception to this condition in the first and second bar and because all the other bars are descending from left to right only we can conclude that our model prediction is good upto some extent

####*Question 10 :Does accuracy of model changes if you use a probability threshold of 0.2. Explain your answer

```
acc <- table(ifelse(pred.valid$posterior[,2] > 0.2, 1, 0), ifelse(as.numeric(spam.valid.norm$spam) >1 ,
confusionMatrix(acc)
```

```
## Confusion Matrix and Statistics
##
##
##       0   1
##   0 356  34
##   1 201 328
##
##                Accuracy : 0.744
##                  95% CI : (0.715, 0.772)
##     No Information Rate : 0.606
##     P-Value [Acc > NIR] : <0.0000000000000002
```

```
##
##                   Kappa : 0.504
##
##  Mcnemar's Test P-Value : <0.0000000000000002
##
##             Sensitivity : 0.639
##             Specificity : 0.906
##          Pos Pred Value : 0.913
##          Neg Pred Value : 0.620
##              Prevalence : 0.606
##          Detection Rate : 0.387
##    Detection Prevalence : 0.424
##       Balanced Accuracy : 0.773
##
##        'Positive' Class : 0
##
```

#*Interpretation 10 : Yes the accuracy of model comes down to 0.744. The senstivity gets reduced to 0.639 and specificity gets increased to 0.906.This is due to the cutoff value or threshold that we changed which does not capture the class that has more records.