Homework3

Shraddha Hemant Kadam (sxk190069@utdallas.edu)

07/14/2020

```
pacman::p_load(caret, MASS, ggplot2, data.table, gains, dplyr)
options(digits = 3)
library(RCurl)
knitr::opts_chunk$set(echo = TRUE, fig.height=8, fig.width=12, fig.path = 'Figs/')
theme_set(theme_classic())
library(data.table)
```

```
fileURL <- "https://archive.ics.uci.edu/ml/machine-learning-databases/spambase/spambase.data"
spam_nspam  <- read.csv(fileURL, header = FALSE, sep = ",", quote = "\"'")
names(spam_nspam) <- c("word_freq_make", "word_freq_address", "word_freq_all",
                       "word_freq_3d", "word_freq_our", "word_freq_over",
                       "word_freq_remove", "word_freq_internet",
  "word_freq_order", "word_freq_mail", "word_freq_receive", "word_freq_will",
    "word_freq_people", "word_freq_report", "word_freq_addresses", "word_freq_free",
    "word_freq_business", "word_freq_email", "word_freq_you", "word_freq_credit",
    "word_freq_your", "word_freq_font", "word_freq_000", "word_freq_money",
    "word_freq_hp", "word_freq_hpl", "word_freq_george", "word_freq_650",
  "word_freq_lab", "word_freq_labs", "word_freq_telnet", "word_freq_857", "word_freq_data",
    "word_freq_415", "word_freq_85", "word_freq_technology", "word_freq_1999",
    "word_freq_parts", "word_freq_pm", "word_freq_direct", "word_freq_cs",
  "word_freq_meeting", "word_freq_original", "word_freq_project", "word_freq_re", "word_freq_edu",
    "word_freq_table", "word_freq_conference", "char_freq_ch;", "char_freq_(",
    "char_freq_[", "char_freq_!", "char_freq_$", "char_freq_#",
  "capital_run_length_average", "capital_run_length_longest", "capital_run_length_total", "class")
```

```
#normalizing the dataset
norm.values <- preProcess(spam_nspam[ ,-58], method = c("center", "scale"))
spam_nspam.norm<-predict(norm.values,spam_nspam)
set.seed(42)
training.index <- createDataPartition(spam_nspam.norm$class, p = 0.8, list = FALSE)
spam_nspam.train <- spam_nspam[training.index, ]
spam_nspam.valid <- spam_nspam[-training.index, ]
```

1 Ans: We will filter the data into Spam and Nonspam based on provided data and then take the average values based on the predictors in Spam as well as in Nonspam datasets. Then by finding the absolute difference between the averages we can sort the difference and get the top 10 predictors which are required as shown in the following code:

```
nspam <- filter(spam_nspam.norm,class==0)
spam <- filter(spam_nspam.norm,class==1)
```

```
nspamavg <- colMeans(nspam[,-58])
spamavg <- colMeans(spam[,-58])
avgdiff<- abs(nspamavg-spamavg)
avgdiff.vec <- as.vector(avgdiff)
names(avgdiff.vec) <- names(avgdiff)
top10predictors <- head(sort(avgdiff,decreasing=TRUE),10)
top10predictors_names <- as.data.frame(names(top10predictors))
colnames(top10predictors_names) <- c("Top 10 Predictors")
```

The top 10 Predictors are as follows:

```
top10predictors_names
```

```
##           Top 10 Predictors
## 1            word_freq_your
## 2             word_freq_000
## 3          word_freq_remove
## 4               char_freq_$
## 5             word_freq_you
## 6            word_freq_free
## 7        word_freq_business
## 8              word_freq_hp
## 9  capital_run_length_total
## 10            word_freq_our
```

2 Ans. Performing Linear Discriminant Analysis using the training dataset by including only 10 predictors identified above.

```
top_pred <- names(top10predictors)
lda_pred <- c(top_pred,'class')
pred.data.train <- spam_nspam.train[,lda_pred]
pred.data.valid <- spam_nspam.valid[,lda_pred]
lda_model <-lda(class~., data = pred.data.train)
lda_model
```

```
## Call:
## lda(class ~ ., data = pred.data.train)
##
## Prior probabilities of groups:
##     0     1
## 0.604 0.396
##
## Group means:
##   word_freq_your word_freq_000 word_freq_remove `char_freq_$` word_freq_you
## 0          0.438       0.00682           0.0111        0.0124          1.28
## 1          1.365       0.24854           0.2657        0.1768          2.26
##   word_freq_free word_freq_business word_freq_hp capital_run_length_total
## 0         0.0702             0.0486       0.9161                      164
## 1         0.5430             0.2818       0.0203                      464
##   word_freq_our
## 0         0.179
## 1         0.506
```

```
##
## Coefficients of linear discriminants:
##                                LD1
## word_freq_your            0.334727
## word_freq_000             1.013978
## word_freq_remove          1.061004
## `char_freq_$`             1.159363
## word_freq_you             0.113383
## word_freq_free            0.426391
## word_freq_business        0.386598
## word_freq_hp             -0.140045
## capital_run_length_total  0.000634
## word_freq_our             0.364672
```

3 Ans. The probability which is calculated from the frequency distribution of various classes present in the dataset is known as Prior Probability.

```
lda_model$prior
```

```
##     0     1
## 0.604 0.396
```

From the above results we can see that the prior probability of an email being Non-Spam is 0.604 and that of being Spam is 0.396. We can see from the data that records which are classified as SPAM are 1813 out 4601, which has a relative frequency of 0.39 and same goes for the Non-spam records.

4 Ans: Linear Discriminant is as follows:

```
lda_model$scaling
```

```
##                                LD1
## word_freq_your            0.334727
## word_freq_000             1.013978
## word_freq_remove          1.061004
## `char_freq_$`             1.159363
## word_freq_you             0.113383
## word_freq_free            0.426391
## word_freq_business        0.386598
## word_freq_hp             -0.140045
## capital_run_length_total  0.000634
## word_freq_our             0.364672
```

The weighted average of each predictors are shown by coefficient of linear discriminant and by using this we can find the LD scores. For example, when we multiply each value of LD1 by the corresponding elements of variables and sum them, we get a LD score for each observation. (0.334727 x word_freq_your + 1.013978 x word_freq_your + 1.061004 x word_freq_your + …)

```
# Predcitions using validation dataset
spam_nspam.valid.pred  <- predict(lda_model,pred.data.valid)
head(spam_nspam.valid.pred$x,10)
```

```
##         LD1
## 5   0.12662
## 11  0.63895
## 21 -1.13507
## 28  1.01898
## 40  0.00965
## 42  1.19382
## 76  2.58618
## 85 -1.07483
## 88  0.84373
## 90 -1.12556
```

```
head(spam_nspam.valid.pred$posterior,10)
```

```
##         0      1
## 5  0.6323 0.3677
## 11 0.3995 0.6005
## 21 0.9469 0.0531
## 28 0.2474 0.7526
## 40 0.6811 0.3189
## 42 0.1921 0.8079
## 76 0.0177 0.9823
## 85 0.9410 0.0590
## 88 0.3127 0.6873
## 90 0.9460 0.0540
```

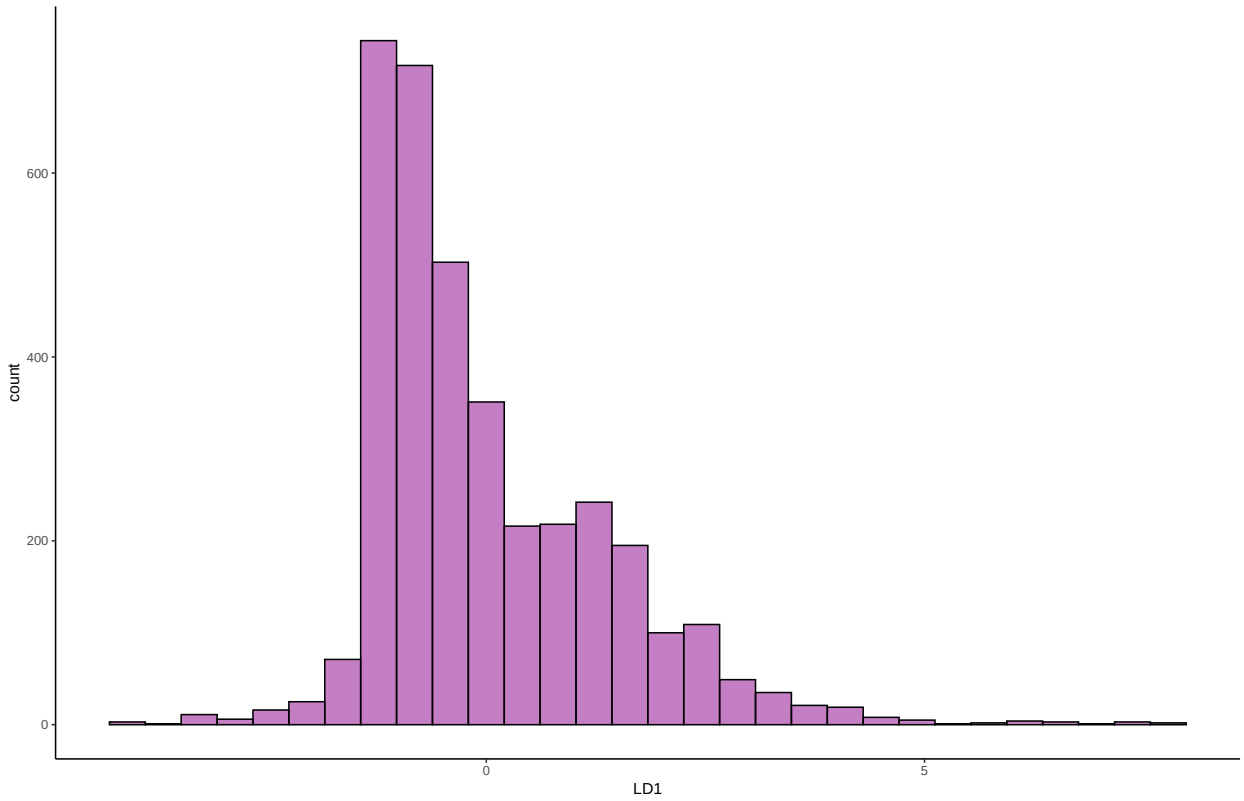
5 Ans: Spam and Nonspam can be classified on the basis posterior probability which tells us the probability of an observation belonging to a class. For example, the first record in output has the probability of belonging to Nonspam class is 0.6323 and the probability of belonging to Spam class is 0.3677. So if we use the default 0.5 cut off then we can identify this email as Nonspam. The second observation tells us that the probability of belonging to Nonspam class is 0.3995 and that of belonging to Spam class is 0.6005. So if we us the default 0.5 cut off then we can identify this email as Spam. Same goes for the rest of the observations.

6 Ans. There is only 1 Linear Discriminant in the model as the number of classes are 2. This is because from the rule of thumb we can say that Linear Discremination is number of classes -1. Therefore 2-1 = 1.

```
 lda.model.train.plot <- data.frame(pred.data.train, predict(lda_model)$x)
ggplot(lda.model.train.plot, aes(LD1,fill=class)) +
  geom_histogram(color ="black", fill= "darkmagenta", alpha=0.5) +
  ggtitle("LDA plot for the training data set:")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

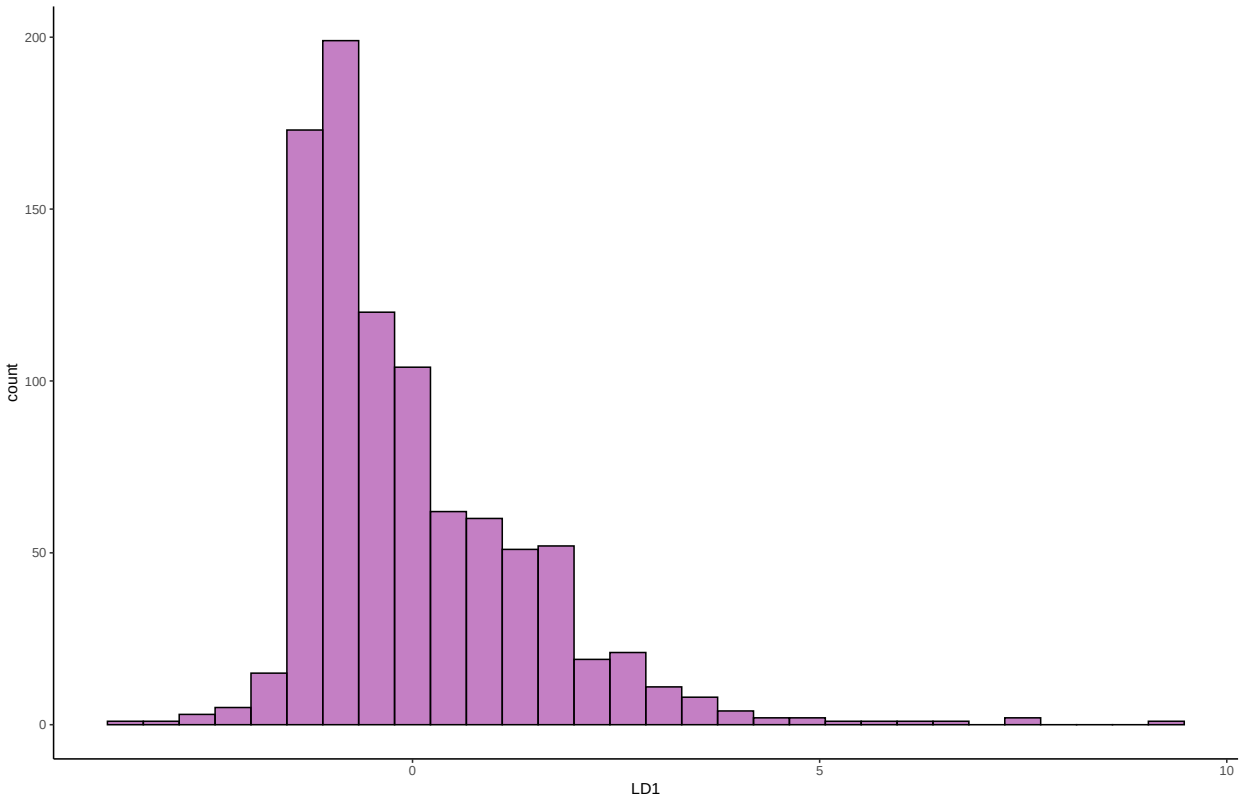
LDA plot for the training data set:



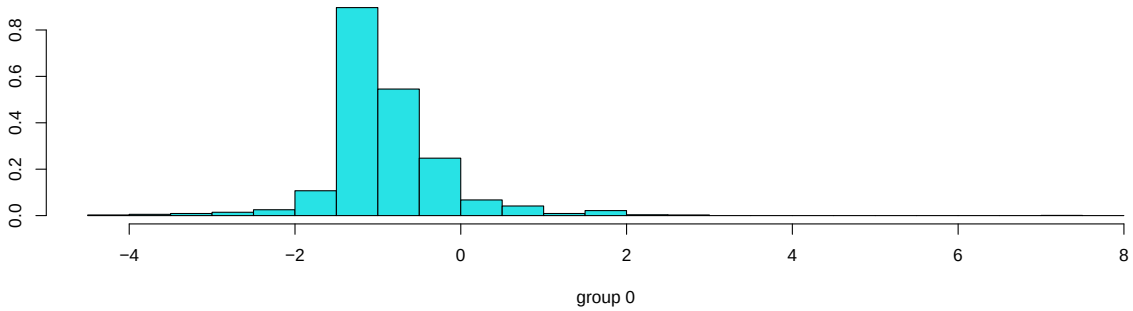

```
 lda.model.valid.plot <- data.frame(pred.data.valid, spam_nspam.valid.pred$x)
ggplot(lda.model.valid.plot, aes(LD1,fill=class)) +
  geom_histogram(color ="black", fill= "darkmagenta", alpha=0.5) +
  ggtitle("LDA plot for the validation data set:")
```

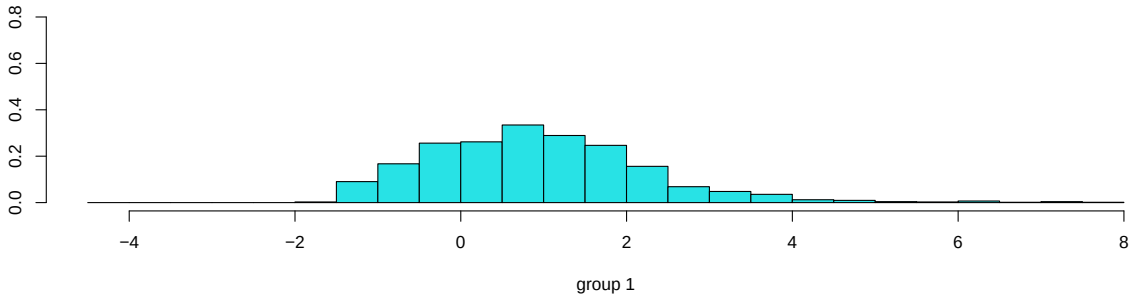
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

LDA plot for the validation data set:

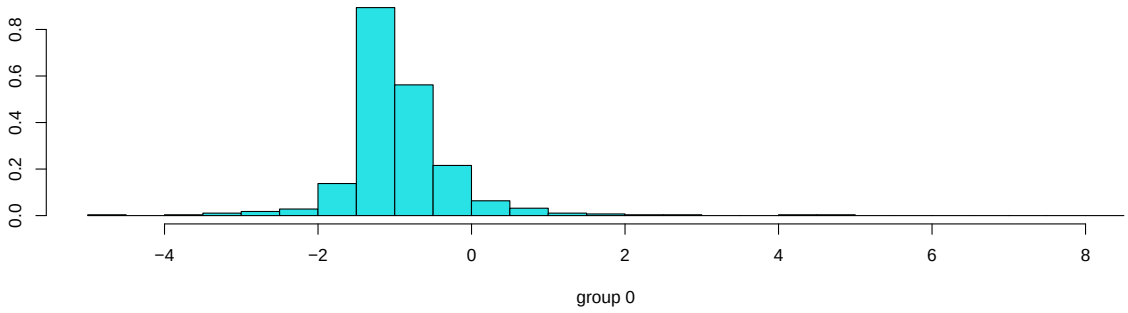


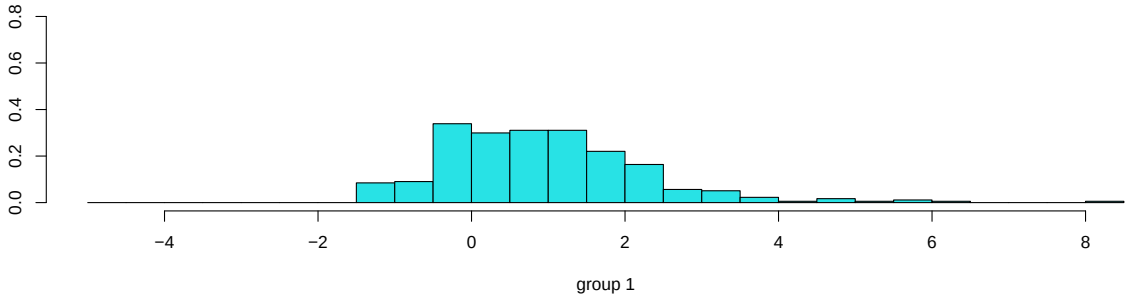

```
plot(lda_model)
```






```
lda_model_2 <-lda(class~., data = pred.data.valid)
plot(lda_model_2)
```





7 Ans: From the plots above, we can see that the LDA plots for training and validation datasets are similar. The observations in the plots present distribution of spam and non-spam classes in the training and validation dataset.

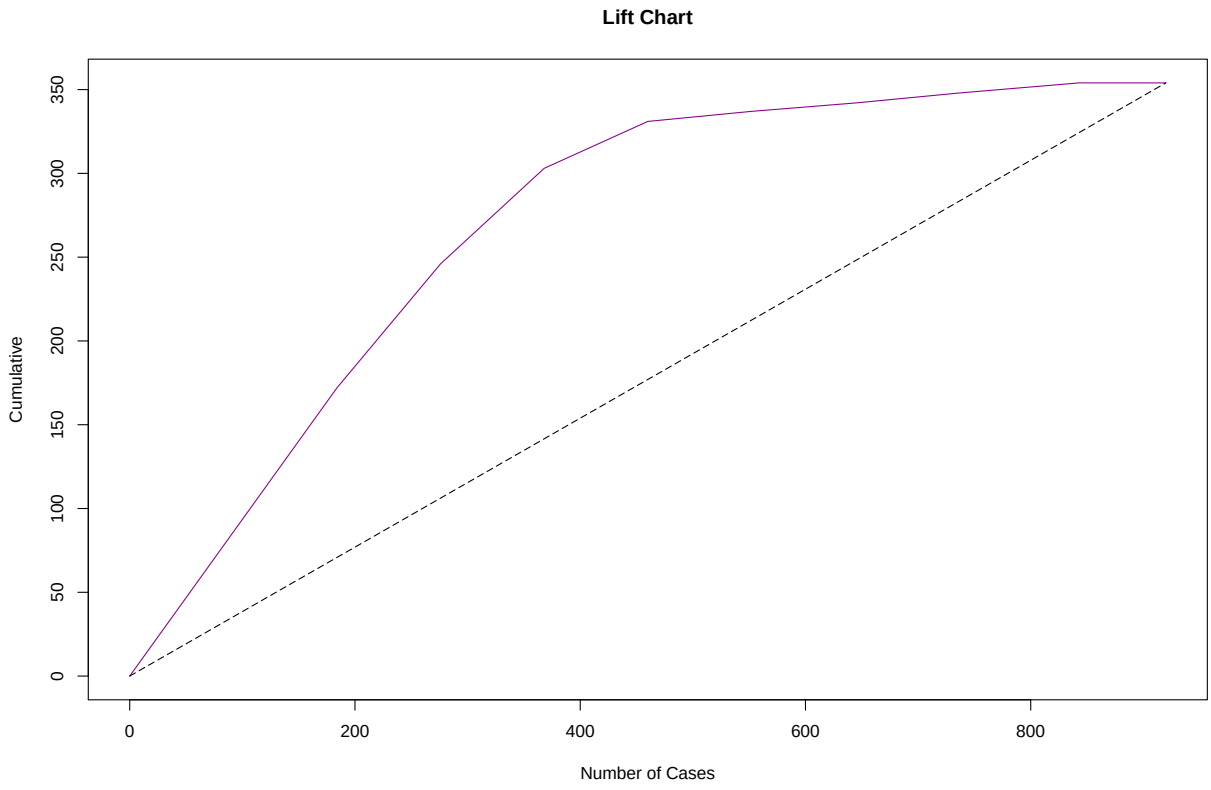
```
confusion_m_data <- table(spam_nspam.valid.pred$class,spam_nspam.valid$class)
confusionMatrix(confusion_m_data)
```

```
## Confusion Matrix and Statistics
##
##
##       0   1
##   0 537 111
##   1  29 243
##
##                Accuracy : 0.848
##                  95% CI : (0.823, 0.87)
##     No Information Rate : 0.615
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.664
##
##  Mcnemar's Test P-Value : 7.61e-12
##
##             Sensitivity : 0.949
##             Specificity : 0.686
##          Pos Pred Value : 0.829
##          Neg Pred Value : 0.893
```

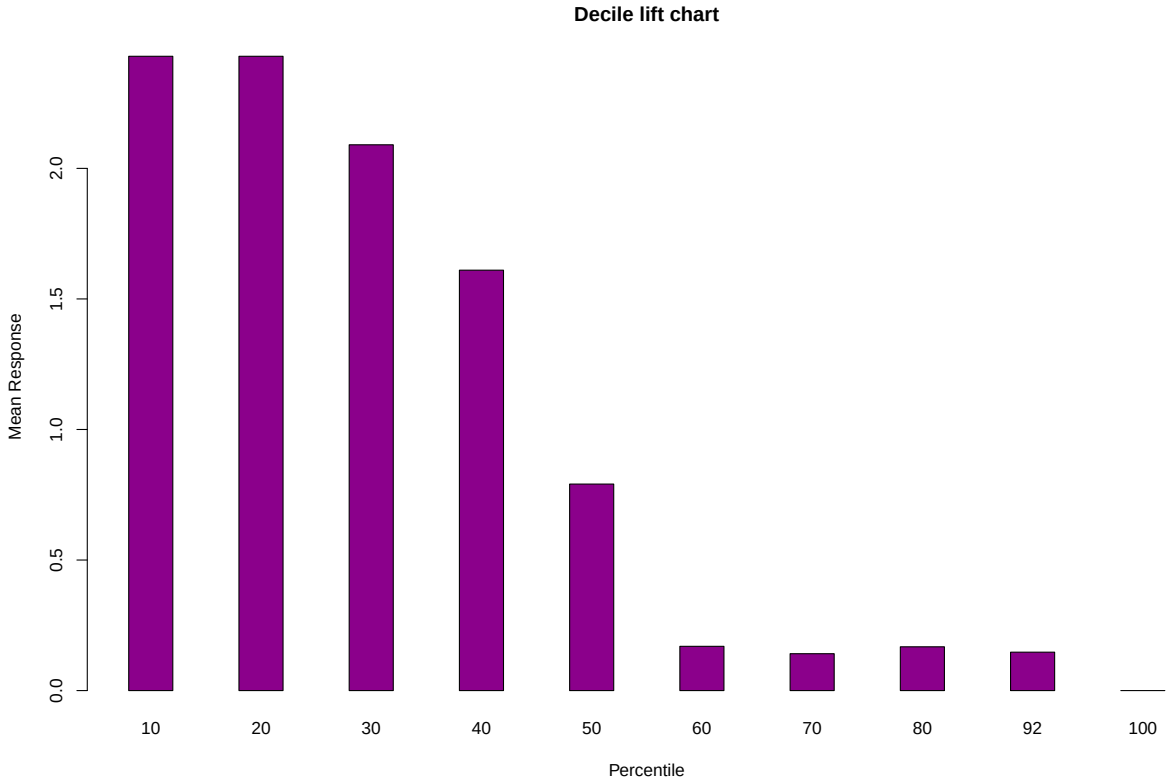
```
##              Prevalence : 0.615
##          Detection Rate : 0.584
##    Detection Prevalence : 0.704
##       Balanced Accuracy : 0.818
##
##        'Positive' Class : 0
##
```

8 Ans: From the above results we can see that the Sensitivity is 0.949 and the Specificity is 0.686.

```
gain <- gains(pred.data.valid$class,spam_nspam.valid.pred$x)
plot(c(0,gain$cume.pct.of.total*sum(as.numeric(pred.data.valid$class)))
     ~c(0,gain$cume.obs),
     xlab = 'Number of Cases', ylab = 'Cumulative',
     main = "Lift Chart",
     col = "darkmagenta",
     type = "l")
lines(c(0,sum(pred.data.valid$class))~c(0,dim(pred.data.valid)[1]), lty = 5)
```




```
barplot(gain$mean.resp/mean(pred.data.valid$class), names.arg = gain$depth, space = 1.5,
        xlab = "Percentile", ylab = "Mean Response", main = "Decile lift chart",
        col = "darkmagenta", border = "black")
```



9 Ans: From the above Lift Chart we can see that our model has a higher lift than No Information Rate. This can also be seen in the confusion matrix where the No Information Rate is 61% and Accuracy of our model is 84% justifying the effectiveness of our model in identifying spams.

```
confusionMatrix(table(as.numeric(spam_nspam.valid.pred$posterior[,2] > 0.2),
pred.data.valid$class), positive = "1")
```

```
## Confusion Matrix and Statistics
##
##
##       0   1
##   0 461  33
##   1 105 321
##
##                Accuracy : 0.85
##                  95% CI : (0.825, 0.872)
##     No Information Rate : 0.615
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.695
##
##  Mcnemar's Test P-Value : 1.5e-09
##
##             Sensitivity : 0.907
##             Specificity : 0.814
##          Pos Pred Value : 0.754
##          Neg Pred Value : 0.933
```

```
##              Prevalence : 0.385
##          Detection Rate : 0.349
##    Detection Prevalence : 0.463
##       Balanced Accuracy : 0.861
##
##        'Positive' Class : 1
##
```

10 Ans: When we decrease the probability threshold from 0.5 to 0.2 that means the probability of classifying the emails as spam will increase. Hence, the emails which were previously Nonspam will now be classified as Spam . This increases the false positive value.