**Build a naive Bayes model on the data set for classifying the ham and spam**

**Ans:**

> sms <- read.csv(file.choose())

> View(sms)

> str(sms)

'data.frame': 5559 obs. of 2 variables:

$ type: Factor w/ 2 levels "ham","spam": 1 1 1 2 2 1 1 1 2 1 ...

$ text: Factor w/ 5156 levels "'An Amazing Quote'' - Sometimes in life its difficult to decide whats wrong!! a lie that brings a smile or the "| \_\_truncated\_\_,..: 1651 2557 257 626 3308 190 357 3392 2726 1079 ...

> sms$type <- factor(sms$type)

> table(sms$type)

ham spam

4812 747

**Building a corpus using text mining**

> library(tm)

Loading required package: NLP

> library(textcat)

> table(textcat(x=sms$type),sms$type)

ham spam

latvian 0 747

norwegian 4812 0

**latvian messages are spam**

**Creating corpus**

> sms\_corpus <- Corpus(VectorSource(sms$text))

**Clean up the corpus using tm\_map()**

> corpus\_clean <- tm\_map(sms\_corpus,tolower)

Warning message:

In tm\_map.SimpleCorpus(sms\_corpus, tolower) :

transformation drops documents

> corpus\_clean <- tm\_map(corpus\_clean,removeNumbers)

Warning message:

In tm\_map.SimpleCorpus(corpus\_clean, removeNumbers) :

transformation drops documents

> corpus\_clean <- tm\_map(corpus\_clean,removeWords,stopwords())

Warning message:

In tm\_map.SimpleCorpus(corpus\_clean, removeWords, stopwords()) :

transformation drops documents

> corpus\_clean <- tm\_map(corpus\_clean,removePunctuation)

Warning message:

In tm\_map.SimpleCorpus(corpus\_clean, removePunctuation) :

transformation drops documents

> corpus\_clean <- tm\_map(corpus\_clean,stripWhitespace)

Warning message:

In tm\_map.SimpleCorpus(corpus\_clean, stripWhitespace) :

transformation drops documents

**Creating a Document term matrix**

> sms\_dtm <- DocumentTermMatrix(corpus\_clean)

**Splitting data to test and train**

> set.seed(101)

> split <- sample(1:nrow(sms),nrow(sms)\*0.7,F)

> tr\_sms <- sms[split,]

> ts\_sms <- sms[-split,]

**Splitting the document term matrix**

> tr\_dtm <- sms\_dtm[split,]

> ts\_dtm <- sms\_dtm[-split,]

> #splitting the corpus

> tr\_cor <- corpus\_clean[split]

> ts\_cor <- corpus\_clean[-split]

> round(prop.table(table(tr\_sms$type))\*100)

ham spam

87 13

> library(wordcloud)

Loading required package: RColorBrewer

> windows()

> wordcloud(tr\_cor,min.freq = 10,max.words = 100,colors = ifelse(sms$type=="spam","red","green"),

+ random.order = F)



**Reducing the number of columns based on lowest frequency**

> freq <- findFreqTerms(x=tr\_dtm,lowfreq = 10)

> tr\_dtm= DocumentTermMatrix(tr\_cor,list(dictionary = freq))

> ts\_dtm= DocumentTermMatrix(ts\_cor,list(dictionary = freq))

> nrow(tr\_dtm)

[1] 3891

> dim(tr\_dtm)

[1] 3891 604

> nrow(ts\_dtm)

[1] 1668

> dim(ts\_dtm)

[1] 1668 604

>counts=function(x){

x= ifelse(x>0,1,0)

x= factor(x,levels = c(0,1), labels = c("no","yes"))

}

> tr\_dtm = apply(tr\_dtm,MARGIN = 2,counts);table(tr\_dtm)

tr\_dtm

no yes

2331725 18439

> ts\_dtm =apply(ts\_dtm,MARGIN = 2,counts);table(ts\_dtm)

ts\_dtm

no yes

999748 7724

**Building naive bayes Model**

> library(e1071)

> model <- naiveBayes(tr\_dtm,tr\_sms$type)

> pred <- predict(model,ts\_dtm)

> table(pred,ts\_sms$type)

pred ham spam

ham 1422 37

spam 7 202

> mean(pred==ts\_sms$type)

[1] 0.9736211

**Model Accuracy is 97.36%**

> library(gmodels)

> CrossTable(pred,ts\_sms$type,prop.chisq = F,prop.r = F,prop.t = F)

Cell Contents

|-------------------------|

| N |

| N / Col Total |

|-------------------------|

Total Observations in Table: 1668

| ts\_sms$type

pred | ham | spam | Row Total |

-------------|-----------|-----------|-----------|

ham | 1422 | 37 | 1459 |

| 0.995 | 0.155 | |

-------------|-----------|-----------|-----------|

spam | 7 | 202 | 209 |

| 0.005 | 0.845 | |

-------------|-----------|-----------|-----------|

Column Total | 1429 | 239 | 1668 |

| 0.857 | 0.143 | |

-------------|-----------|-----------|-----------|

> model2 <- naiveBayes(tr\_dtm,tr\_sms$type,laplace = 1)

> pred2 <- predict(model2,ts\_dtm)

> mean(pred2==ts\_sms$type)

[1] 0.9730216

**Model Accuracy is 97.30%, Which is just fit Model**

> library(gmodels)

> CrossTable(pred,ts\_sms$type,prop.chisq = F,prop.r = F,prop.t = F)

Cell Contents

|-------------------------|

| N |

| N / Col Total |

|-------------------------|

Total Observations in Table: 1668

| ts\_sms$type

pred | ham | spam | Row Total |

-------------|-----------|-----------|-----------|

ham | 1422 | 37 | 1459 |

| 0.995 | 0.155 | |

-------------|-----------|-----------|-----------|

spam | 7 | 202 | 209 |

| 0.005 | 0.845 | |

-------------|-----------|-----------|-----------|

Column Total | 1429 | 239 | 1668 |

| 0.857 | 0.143 | |

-------------|-----------|-----------|-----------|