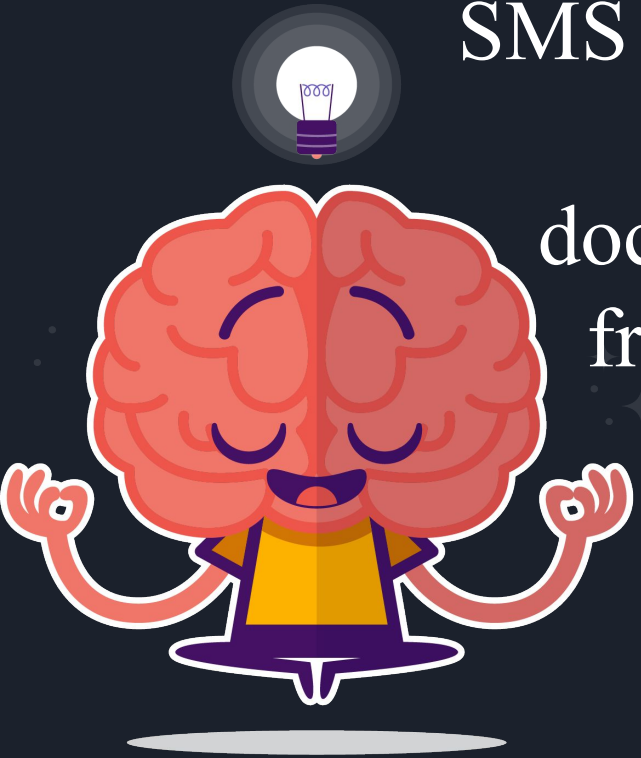
SMS - Spam or Ham

- Gagan Kumar Soni
- Adarsh Shukla
- Parul Negi
- Anubhav Singh

Problem Statement



SMS text classification by extracting word frequencies from the text of the documents, and then integrating those frequencies into predictive models.

Dataset for the problem

```
ham Go until jurong point, crazy.. Available only in bugis n grea
ham Ok lar... Joking wif u oni...
spam    Free entry in 2 a wkly comp to win FA Cup final tkts 21st
question(std txt rate)T&C's apply 08452810075over18's
ham U dun say so early hor... U c already then say...
ham Nah I don't think he goes to usf, he lives around here though
spam    FreeMsg Hey there darling it's been 3 week's now and no w
Tb ok! XxX std chgs to send, £1.50 to rcv
ham Even my brother is not like to speak with me. They treat me I
ham As per your request 'Melle Melle (Oru Minnaminunginte Nurungu
Callers. Press *9 to copy your friends Callertune
spam    WINNER!! As a valued network customer you have been selec
09061701461. Claim code KL341. Valid 12 hours only.
spam    Had your mobile 11 months or more? U R entitled to Update
Free! Call The Mobile Update Co FREE on 08002986030
ham I'm gonna be home soon and i don't want to talk about this st
spam    SIX chances to win CASH! From 100 to 20,000 pounds txt> C
TsandCs apply Reply HL 4 info
spam    URGENT! You have won a 1 week FREE membership in our £100
81010 T&C www.dbuk.net LCCLTD POBOX 4403LDNW1A7RW18
ham I've been searching for the right words to thank you for this
```

The 2 columns are -

- Text Message (char type)
- Label (categorical type)

Junk message - spam
Legitimate message - ham

Source of the dataset

UCI Machine learning Repository

Flow of the Presentation

Step 1 - Analysis and Preprocessing

EDA, creating corpus of the data and removing stop words

Step 2 - Finding Word Frequencies and Removing Sparse Terms

Stemming , Document term Matrix, sorting the words

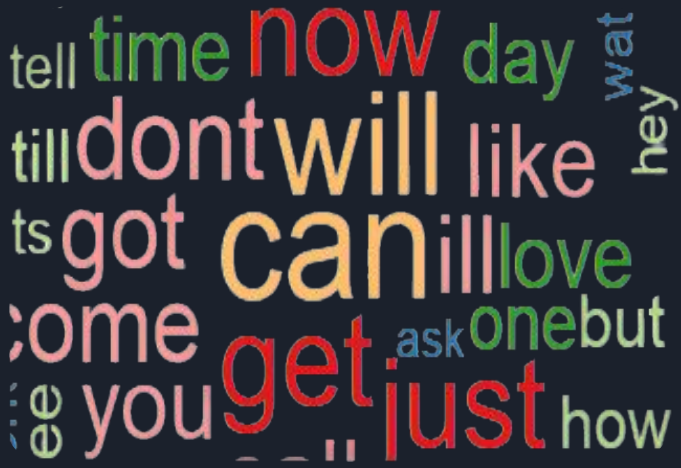
Step 3 - Building Model and Finding their accuracies

Naive Bayes, Decision Tree, Conclusion

Analysis & PreProcessing

```r
library(wordcloud)
spam <- subset(messages, label == "spam")
wordcloud(spam$message, max.words = 60, colors = brewer.pal(7, "Paired"), random.order = FALSE)

ham <- subset(messages, label == "ham")
wordcloud(ham$message, max.words = 60, colors = brewer.pal(7, "Paired"), random.order = FALSE)
```

Spam



Ham

```
print(paste("Longest message has: ",max(nchar(messages$message))," Characters"))
cat("\n")
print(paste("shortest message has: ",min(nchar(messages$message))," Characters"))
```

OUTPUT:-

```
"Longest message has:  910  Characters"

"shortest message has:  2  Characters"
```

```r
print(paste("Index of messages with shortest length"))
print(which(nchar(messages$message) == 2))
print(paste("Shortest message is: ",messages$message[which(nchar(messages$message) == 2)[1]]))
print(paste("Longest message is: ",messages$message[which.max(nchar(messages$message))]))
```

OUTPUT

```
[1] "Index of messages with shortest length"
[1] 1926 3052 4499 5360
[1] "Shortest message is:  Ok"
[1] "Longest message is:  For me the love should start with attraction.i should
 feel that I need her every time around me.she should be the first thing which
 comes in my thoughts.I would start the day and end it with her.she should be t
here every time I dream.love will be then when my every breath has her name.my
 life should happen around her.my life will be named to her.I would cry for he
r.will give all my happiness and take all her sorrows.I will be ready to fight
 with anyone for her.I will be in love when I will be doing the craziest things
 for her.love will be when I don't have to proove anyone that my girl is the mo
st beautiful lady on the whole planet.I will always be singing praises for her.
love will be when I start up making chicken curry and end up makiing sambar.lif
e will be the most beautiful then.will get every morning and thank god for the
 day because she is with me.I would like to say a lot..will tell later.."
```

CORPUS

Corpus is a collection of text documents over which we can apply data mining or NLP routines to drive inferences.

```r
#creating a corpus
corpus <- Corpus(VectorSource(messages$message))
print(corpus)
corpus = tm_map(corpus, content_transformer(tolower))
corpus = tm_map(corpus, removePunctuation)
corpus <- tm_map(corpus, removeWords, stopwords("english"))
```

```r
library(SnowballC)
corpus <- tm_map(corpus, stemDocument)
```

Stemming

Stemming is **a technique used to extract the base form of the words by removing affixes from them**. It is just like cutting down the branches of a tree to its stems. For example, the stem of the words eating, eats, eaten is eat.

Analysing word Frequencies

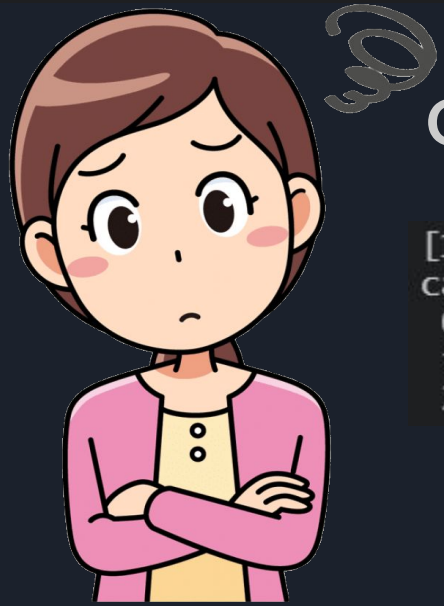
Bags of words:-

- the rows correspond to documents, and
- the columns correspond to words .

```
dtm = DocumentTermMatrix(corpus)
print(dtm)

#remove sparse terms
spdtm = removeSparseTerms(dtm, 0.98)
print(spdtm)
```

```
<<DocumentTermMatrix (documents: 5574, terms: 7744)>>
Non-/sparse entries: 44451/43120605
Sparsity           : 100%
Maximal term length: 52
Weighting          : term frequency (tf)
<<DocumentTermMatrix (documents: 5574, terms: 50)>>
Non-/sparse entries: 9880/268820
Sparsity           : 96%
Maximal term length: 5
Weighting          : term frequency (tf)
```

```
messagesSparse = as.data.frame(as.matrix(spdtm))
print(dim(messagesSparse))

colnames(messagesSparse) = make.names(colnames(messagesSparse))

#sorting word and finding more frequent
print(head(sort(colSums(messagesSparse),decreasing = TRUE),n = 20))
```



OUTPUT:-

```
[1] 5574   50
call  now  get  can will just come dont free know like love want  got
 647  476  435  392  379  363  294  276  271  261  249  247  241  236
 ill time good  day text send
 236  234  231  227  222  198
```

Building Model

Dividing into training and testing sets

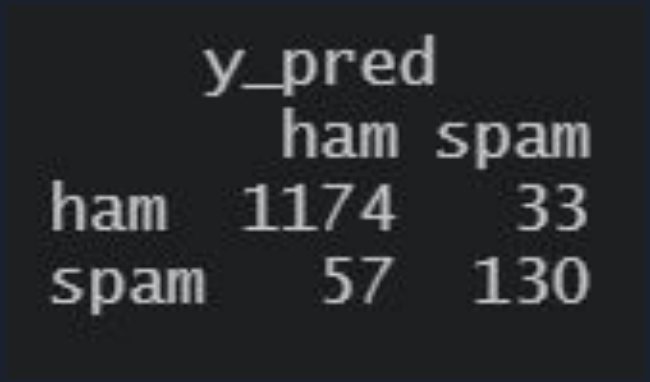
```
library(caTools)
set.seed(123)
split<-sample.split(messagesSparse$label,SplitRatio = 0.75)

training_set = subset(messagesSparse,split==TRUE)
testing_set = subset(messagesSparse,split==FALSE)
```

Naive Bayes

```
library(e1071)
classifier_naive <- naiveBayes(msg_train,training_set$label)
y_pred <- predict(classifier_naive, newdata = msg_test)

cm <- table(testing_set$label, y_pred)
print(cm)
print(confusionMatrix(cm))
print(confusionMatrix(cm)$overall["Accuracy"]*100)
```

Splitting Test data and training data
and Building Model

Confusion Matrix



	y_pred	
	ham	spam
ham	1174	33
spam	57	130

Accuracy

93.54376

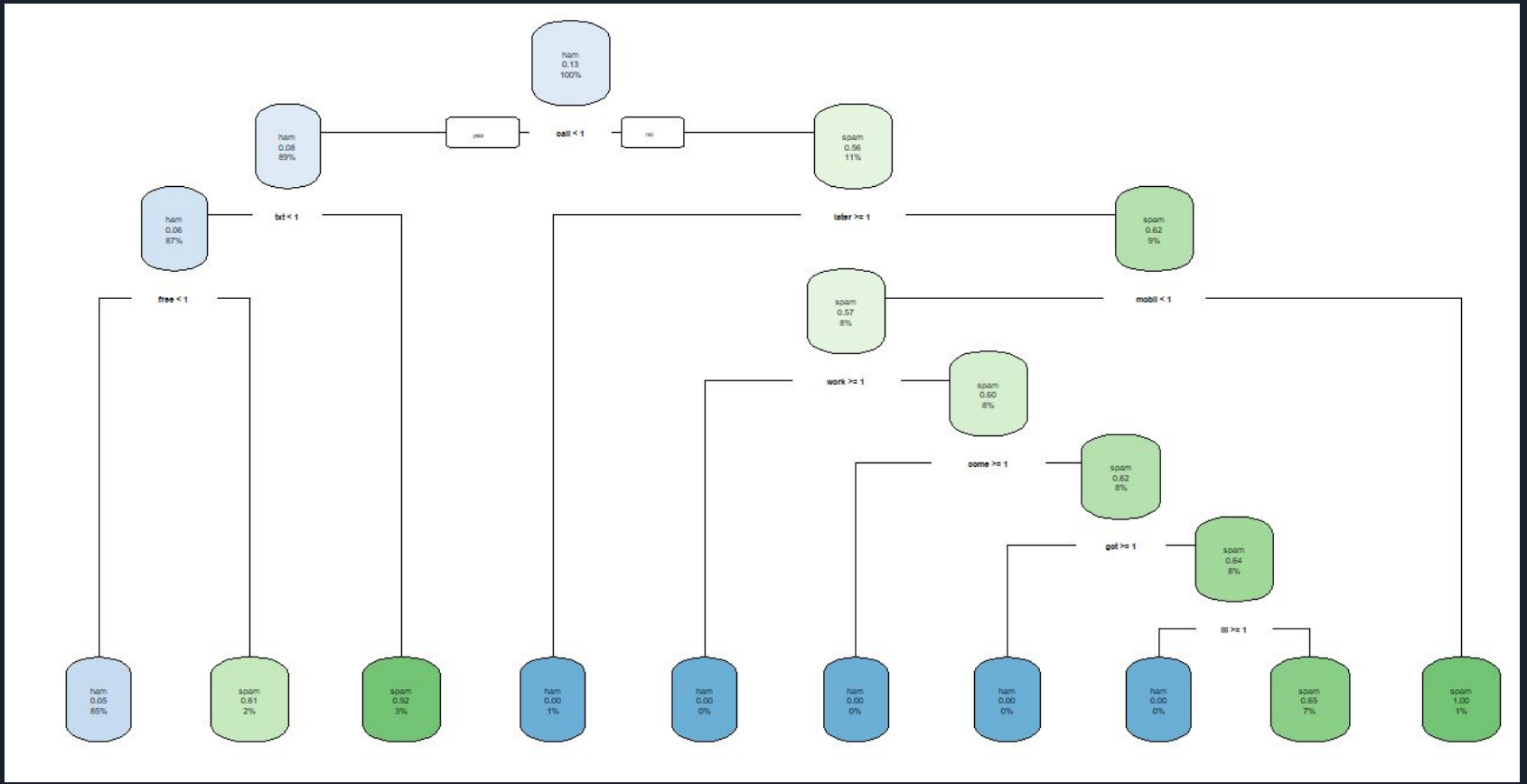
- Sensitivity : 0.9537
- Specificity : 0.7975
- Pos Pred Value : 0.9727
- Neg Pred Value : 0.6952

Decision Tree

```
library(rpart)
library(rpart.plot)

dt<-rpart(label~.,training_set,method = "class")

rpart.plot(dt)
rpart.plot(dt,type = 4, extra = 101)

p<-predict(dt,testing_set,type = "class")
```

Splitting Test data and training data
and Building Model

```
rpart.plot(dt)
```



Confusion Matrix

	p	
	ham	spam
ham	1139	68
spam	65	122

Accuracy

90.45911

- Sensitivity : 0.9460
- Specificity : 0.6421
- Pos Pred Value : 0.9437
- Neg Pred Value : 0.6524

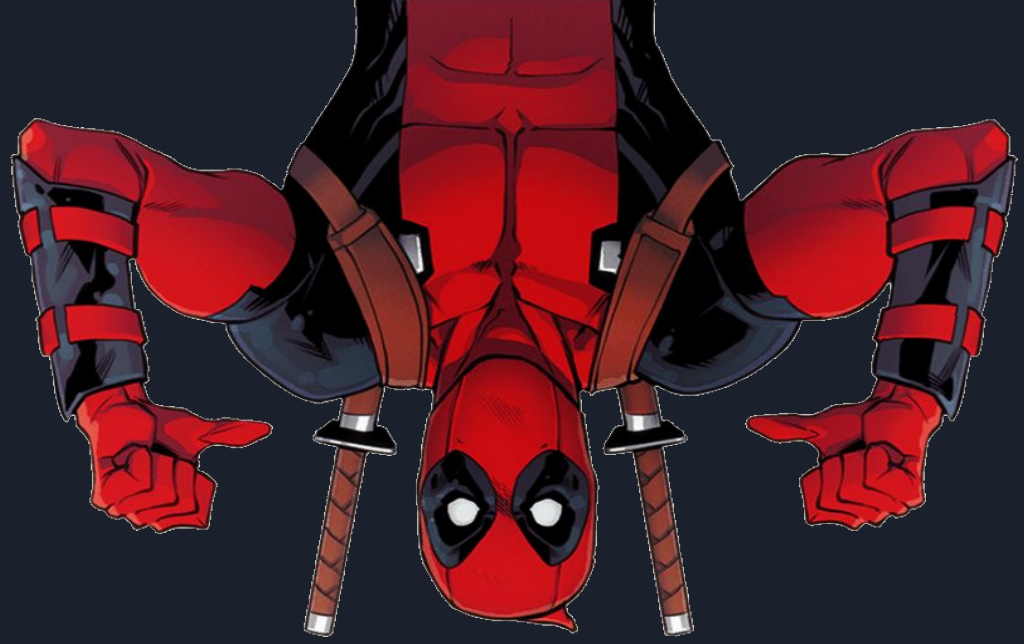
Conclusion

As we can see the Accuracy of Naive Bayes Model is greater than that of Decision tree.

Why Naive bayes Model Worked so Well ?

- Occurrence of words in a text document are independent of each other
- The location of one word does not depends on another

This satisfying the independence assumption of Naive Bayes model . Hence, it is most commonly used for Spam filtering



THANK YOU!