Sentiment Analysis & Prediction

```
The packages being used are: -readr
-dplyr
-tm
-SnowballC
-caret
Dataset being used: IMDB dataset (Sentiment analysis)
Loading the Data
library(readr)
library(dplyr)
library(tm)
library(SnowballC)
  dF <- read.csv(choose.files())</pre>
  dF \leftarrow dF[1:10,]
  glimpse(dF)
## Rows: 10
## Columns: 2
## $ i..text <chr> "I grew up (b. 1965) watching and loving the Thunderbirds. All~
## $ label <int> 0, 0, 0, 0, 1, 0, 1, 0, 1, 1
Creating a Corpus
corp <- Corpus(VectorSource(dF$\"i..text))</pre>
corp[[1]][1]
## $content
## [1] "I grew up (b. 1965) watching and loving the Thunderbirds. All my mates at school watched. We pl
# Lowercase
corp <- tm_map(corp, PlainTextDocument)</pre>
## Warning in tm_map.SimpleCorpus(corp, PlainTextDocument): transformation drops
## documents
corp <- tm_map(corp, tolower)</pre>
## Warning in tm_map.SimpleCorpus(corp, tolower): transformation drops documents
```

```
corp[[1]][1]
```

As you can see below, the Corpus text has been converted to lowercase for simplification and processing

```
## $content
## [1] "i grew up (b. 1965) watching and loving the thunderbirds. all my mates at school watched. we pl
```

Further Preprocessing

```
corp = tm_map(corp, removePunctuation)

## Warning in tm_map.SimpleCorpus(corp, removePunctuation): transformation drops

## documents

corp = tm_map(corp, removeWords, c("movie", stopwords("english")))

## Warning in tm_map.SimpleCorpus(corp, removeWords, c("movie",

## stopwords("english"))): transformation drops documents

corp = tm_map(corp, stemDocument)

## Warning in tm_map.SimpleCorpus(corp, stemDocument): transformation drops

## documents

freq = DocumentTermMatrix(corp)

##Cleaning freq

freq = removeSparseTerms(freq, 0.995)
```

Creating Dataframe

```
based = as.data.frame(as.matrix(freq))
colnames(based) = make.names(colnames(freq))
based$id = dF$label
based$id = as.factor(based$id)
```

```
set.seed(469)
library(caret)

split = createDataPartition(based$id, p = 0.7, list = F)
train <- based[split,]
test <- based[-split,]

ldamod <- train(id ~ ., data = train, method = "symLinear", tuneLength = 2,preProcess = c("center", "sca</pre>
```

ldamod

```
## Support Vector Machines with Linear Kernel
##
     70 samples
## 3740 predictors
      2 classes: '0', '1'
##
## Pre-processing: centered (3740), scaled (3740)
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 70, 70, 70, 70, 70, 70, ...
## Resampling results:
##
##
     Accuracy
                Kappa
     0.6039992 0.09893723
##
##
## Tuning parameter 'C' was held constant at a value of 1
pred <- predict(ldamod, based)</pre>
confusionMatrix(based$id, pred)
## Confusion Matrix and Statistics
##
##
             Reference
## Prediction 0 1
##
            0 57 3
            1 9 31
##
##
                  Accuracy: 0.88
##
##
                    95% CI: (0.7998, 0.9364)
##
       No Information Rate: 0.66
       P-Value [Acc > NIR] : 4.427e-07
##
##
##
                     Kappa: 0.7436
```

Specificity: 0.9118

Mcnemar's Test P-Value: 0.1489

Sensitivity: 0.8636

Pos Pred Value : 0.9500 ## Neg Pred Value: 0.7750 ## Prevalence: 0.6600 Detection Rate: 0.5700 ## Detection Prevalence : 0.6000 ## ## Balanced Accuracy: 0.8877

##

##

##

'Positive' Class: 0 ## ##

Word Cloud

library(wordcloud)

```
freq = DocumentTermMatrix(corp)
a <- as.data.frame(freq$dimnames$Terms)
b <- as.matrix(freq)
c <- as.data.frame(b)
c <- t(c)
c <- as.data.frame(c)
c$sum <- c$'1' + c$'2' + c$'3' + c$'4' + c$'5' + c$'6' + c$'7' + c$'8' + c$'9' + c$'10'
c <- as.data.frame(c)
d <- cbind(a,c$sum)
d
wordcloud(d$'freq$dimnames$Terms', d$'c$sum')</pre>
```

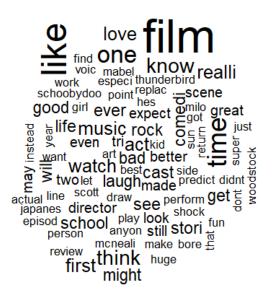


Figure 1: Word Plot