VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

SENTIMENT ANALYSIS

A J Component Report

submitted by

NISHANT PANDEY

17BCE0780

in partial fulfillment for the award of the degree of

B.Tech

in

COMPUTER SCIENCE ENGINEERING

Under the guidance

of

Faculty: Prof. Delhi Babu R

School of Computing Science and Engineering

MARCH 2019

VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

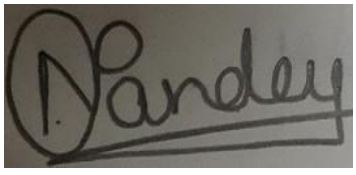
School of Computer Science and Engineering

DECLARATION

I hereby declare that the J Component report entitled

"**Sentiment Analysis"** submitted

by me to Vellore Institute of Technology, Vellore-14 in partial fulfillment of the requirement for the award of the degree of **B.Tech** in **Computer science and engineering** is a record of bonafide undertaken by me under the supervision of **Dr. R. Delhi Babu** I further declare that the work reported in this report has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.



Signature

Name: NISHANT PANDEY

Reg. Number: 17BCE0780

TABLE OF CONTENTS

LIST OF FIGURES

1. Introduction

Sentiment Analysis refers to the use of natural language processing, text analysis and statistical learning to identify and extract subjective information in source materials.

In simple terms, sentiment analysis aims to determine the attitude of a speaker or a writer with respect to some topic or the overall contextual polarity of a document. In this project I have used it to determine if a text has a positive or a negative mood.

Imagine for example that we apply it to entries in Twitter about the hashtag #Windows10. We will be able to determine how people feels about the new version of Microsoft operating system. Of course, this is not a big deal applied to an individual piece of text. I believe that the average person will always make a better judgement than what we will build here. However my model will show its benefits when automatically processing large amounts of text very quickly, or processing a large number of entries.

- ## Abstract

So in the sentiment analysis process there are a couple of stages more or less differentiated. The first is about *processing natural language*, and the second about *training a model*. The first stage is in charge of processing text in a way that, when we are ready to train our model, we already know what variables the model needs to consider as inputs. The model itself is in charge of learning how to determine the sentiment of a piece of text based on these variables. This might sound a bit complicated right now, but I promise it will be crystal-clear by the end of the tutorial.
For the model part I will use linear models. They aren't the most powerful methods in terms of accuracy, but they are simple enough to be interpreted in their results as we will see. Linear methods allow us to define our input variable as a linear combination of input variables. In tis case we will introduce logistic regression.

- **REQUIREMENTS**

software specification:

- R studio
- R 3.3.0 or above

- Windows 8 or above

hardware specification

- 4 GB RAM
- 20 GB Disk Space
- GPU is a plus!

libraries

- dplyr
- tm
- ggplot2
- Rcurl
- CaTools

- **INPUT**

- Raw Text with labelled Sentiments for training.

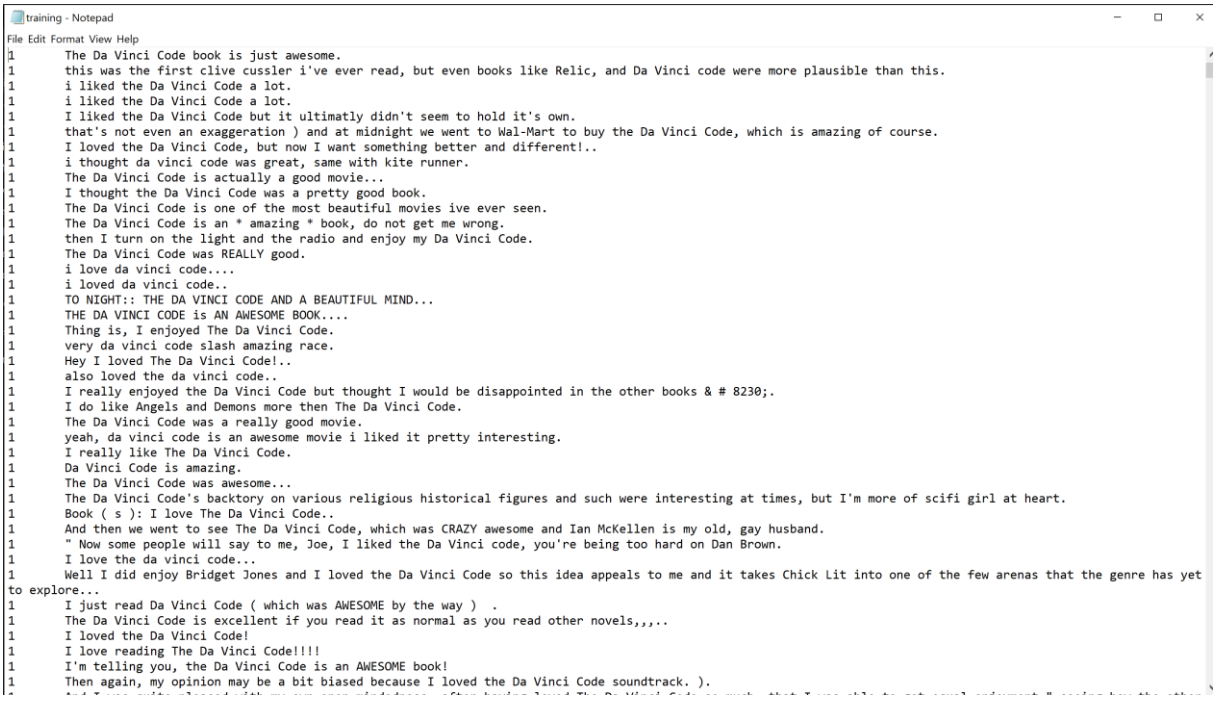


Figure 1 (Input)

- For test data, need raw data from user-

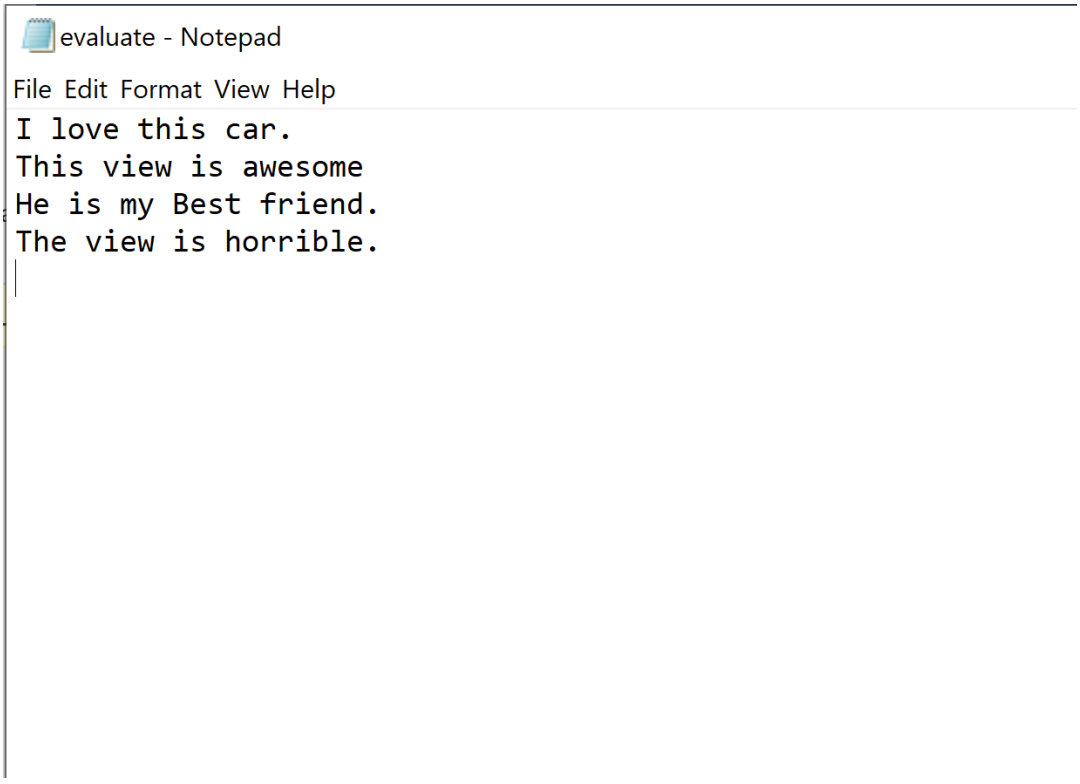


Figure 2(test data)

- **OUTPUT**
  - Sentiment of the statements typed by the user.
  - Confusion matrix
  - Best and worst Sentences
  - Accuracy

- **SAMPLE ALGORITHM TO PROCESS NATURAL LANGUAGE.**

```
test_data_file <- "testdata.txt"
train_data_file <- "training.txt"
e_file<-"evaluate.txt"
train_data_df <- read.csv(
  file  = train_data_file,
```

```
  sep='\t',
  header=FALSE,
  quote = "",
  stringsAsFactor=F,
  col.names=c("Sentiment", "Text"))
test_data_df <- read.csv(
  file  = test_data_file,
  sep='\t',
  header=FALSE,
  quote = "",
  stringsAsFactor=F,
  col.names=c("Text"))
e_data_df <- read.csv(
  file  = e_file,
  sep='\t',
  header=FALSE,
  quote = "",
  stringsAsFactor=F,
  col.names=c("Text"))

# we need to convert Sentiment to factor
train_data_df$Sentiment <- as.factor(train_data_df$Sentiment)
head(train_data_df)


table(train_data_df$Sentiment)


mean(sapply(sapply(train_data_df$Text, strsplit, " "), length))

library(tm)



corpus <- Corpus(VectorSource(c(train_data_df$Text, test_data_df$Text,e_data_df$Text)))

corpus[1]$content


corpus <- tm_map(corpus, content_transformer(tolower))


corpus <- tm_map(corpus, removePunctuation)
corpus <- tm_map(corpus, removeWords, stopwords("english"))
corpus <- tm_map(corpus, stripWhitespace)
corpus<- tm_map(corpus, stemDocument,language="english")

corpus
corpus[1:3]$content

dtm <- DocumentTermMatrix(corpus)
dtm
```

```
sparse <- removeSparseTerms(dtm, 0.99)
sparse

important_words_df <- as.data.frame(as.matrix(sparse))
colnames(important_words_df) <- make.names(colnames(important_words_df))

log_model <- glm(Sentiment~., data=eval_train_data_df, family=binomial)

#summary(log_model)

log_pred <- predict(log_model, newdata=eval_test_data_df, type="response")
table(eval_test_data_df$Sentiment, log_pred>.5)

log_pred_test <- predict(log_model, newdata=test_data_words_df, type="response")
test_data_df$Sentiment <- log_pred_test>.5
```

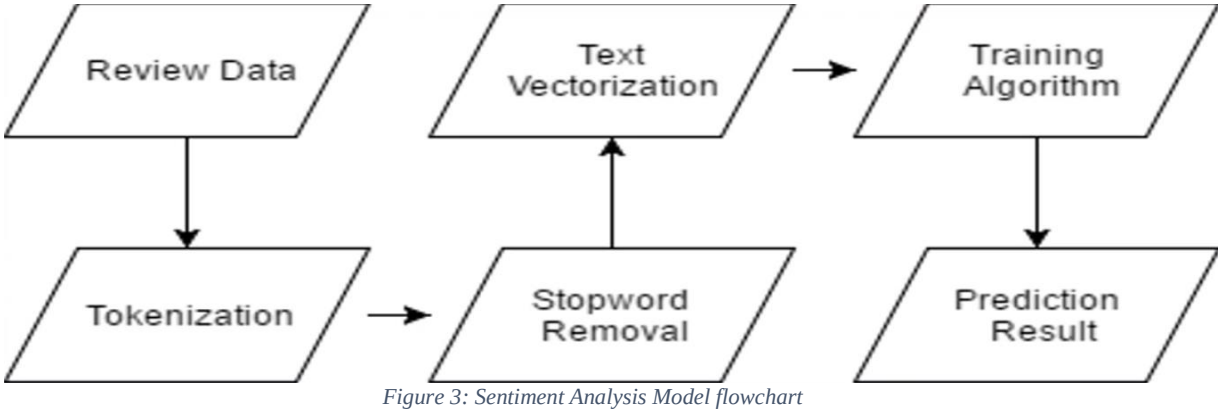


Figure 3: Sentiment Analysis Model flowchart

OUTPUT-

```
                    Terms
Docs                 amazing best car friend horrible like love view
  I love this car          0    0   1      0        0    0    1    0
  This view is amazing     1    0   0      0        0    0    0    1
  He is my best friend     0    1   0      1        0    0    0    0
  I do not like this car   0    0   1      0        0    1    0    0
  The view is horrible     0    0   0      0        1    0    0    1
>
```

Figure 4 Document matrix

```
> e_data_df
                  Text sentiment              Value
1      I love this car.      TRUE Positive Sentiment
2  This view is awesome      TRUE Positive Sentiment
3 He is my Best friend.      TRUE Positive Sentiment
4 The view is horrible.     FALSE Negative sentiment
>
```

Figure 5: Sample Sentiment output

LOGISTIC REGRESSION IMPLEMENTATION

In statistics, the logistic model (or logit model) is a widely used statistical model that, in its basic form, uses a logistic function to model a binary dependent variable; many more complex extensions exist. In regression analysis, logistic regression (or logit regression) is estimating the parameters of a logistic model; it is a form of binomial regression.

Mathematically, a binary logistic model has a dependent variable with two possible values, such as pass/fail, win/lose, alive/dead or healthy/sick; these are represented by an indicator variable, where the two values are labeled "0" and "1".

In the logistic model, the log-odds (the logarithm of the odds) for the value labeled "1" is a linear combination of one or more independent variables ("predictors"); the independent variables can each be a binary variable (two classes, coded by an indicator variable) or a continuous variable (any real value). The corresponding probability of the value labeled "1" can vary between 0 (certainly the value "0") and 1 (certainly the value "1"), hence the labeling; the function that converts log-odds to probability is the logistic function, hence the name.

The unit of measurement for the log-odds scale is called a logit, from logistic unit, hence the alternative names. Analogous models with a different sigmoid function instead of the logistic function can also be used, such as the probit model; the defining characteristic of the logistic model is that increasing one of the independent variables multiplicatively scales the odds of the given outcome at a constant rate, with each dependent variable having its own parameter; for a binary independent variable this generalizes the odds ratio.

$$f(x) = \frac{L}{1 + e^{-k(x-x_0)}}$$

FLOWCHARTS-

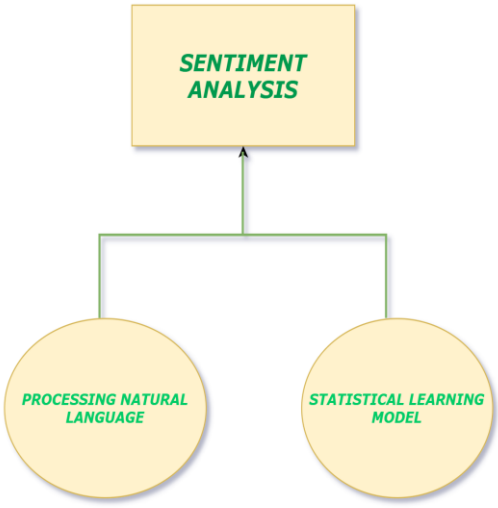


Figure 6: splitting sentiment analysis

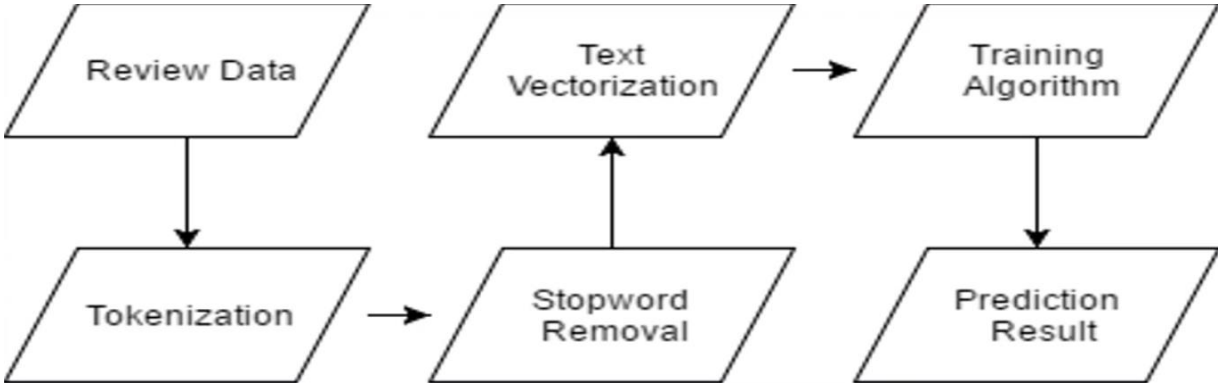


Figure 7: Sentiment analysis process as whole

SCREENSHOTS-
Output-

```
> e_data_df
                  Text sentiment              Value
1      I love this car.      TRUE Positive Sentiment
2  This view is awesome      TRUE Positive Sentiment
3 He is my Best friend.      TRUE Positive Sentiment
4 The view is horrible.     FALSE Negative sentiment
5      I hate this show     FALSE Negative sentiment
6 I like playing games.      TRUE Positive Sentiment
>
```

Figure 8: Sentiment classified of input given by user.

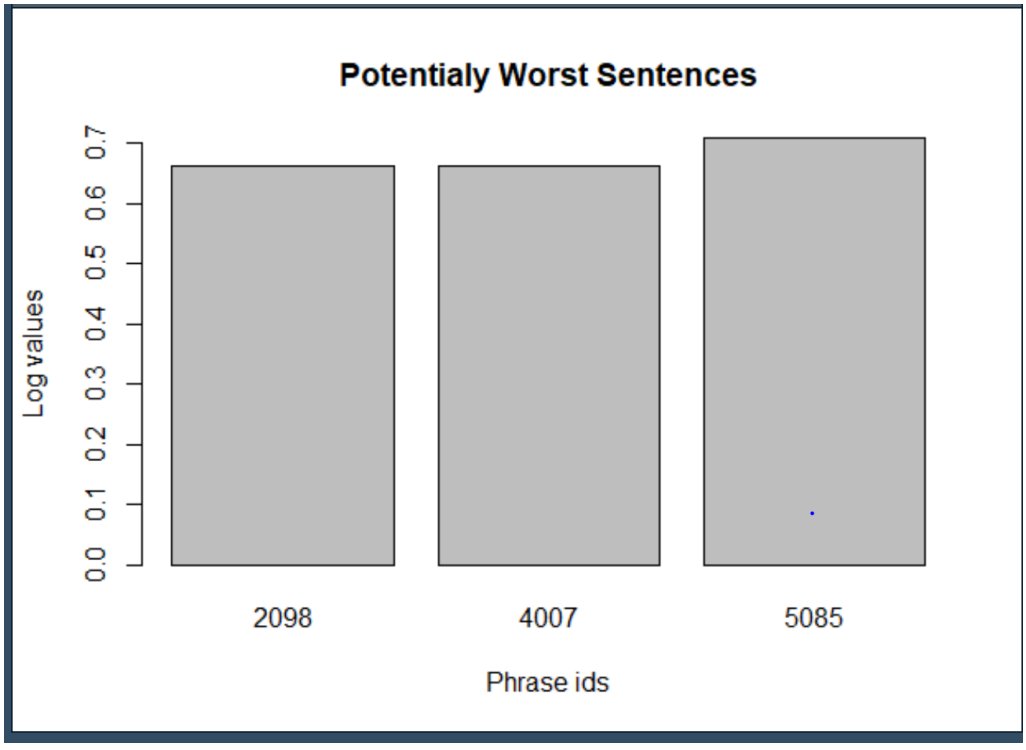


Figure 9: Worst Sentences from the test set (index values) with the Logistic predicted value.

Confusion Matrix-

```
> table(eval_test_data_df$Sentiment, log_pred>.69)

    FALSE TRUE
  0   455    9
  1    10  589
> 
```

Figure 10: Confusion Matrix

CONCLUSION

We have seen that Sentiment Analysis can be used for analyzing opinions in blogs, articles, Product reviews, Social Media websites, Movie-review websites where a third person narrates his views. We also studied NLP and Machine Learning approaches for Sentiment Analysis. We have seen that is easy to implement Sentiment Analysis via SentiWordNet approach than via Classier approach. We haveseen that sentiment analysis has many applications and it is important field to study.Sentiment analysis has Strong commercial interest because Companies want to knowhow their products are being perceived and also Prospective consumers want toknow what existing users think.

REFERENCES

References Pang B., L. Lee and S. Vaithyanathan. “Thumbs up?: sentiment classification using machine learning techniques”. In EMNLP ’02: Proc. of the ACL-02 conf. on Empirical methods in natural language processing, pages 79–86. ACL, 2002 Russell Stuart J. and Peter Norvig. 2003. “Artificial Intelligence: A Modern Approach (2 ed.). Pearson Education”. p. 499

Jason Rennie, Lawrence Shih, Jaime Teevan, David Karger , Tackling the Poor Assumptions of Logistic Regression Text Classifiers, presentation slides(http://cseweb.ucsd.edu/~elkan/254/LogisticForText.pdf