



Research Paper on Tweet Sentiment Extraction

**A Sentiment Analysis Project on Social Media
Platform Twitter**

**Submitted To:
Jamileh Yousefi**

**PREPARED BY
CHANDRAYOG YADAV (20196914)
UMESH GARG (20193834)**

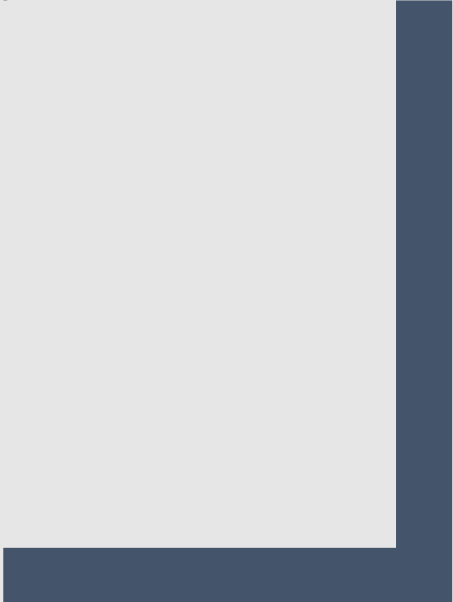


Table of Contents

<i>Abstract</i>	<i>3</i>
<i>Introduction:.....</i>	<i>3</i>
<i>Research Questions to be addressed:</i>	<i>4</i>
<i>Literature Review</i>	<i>4</i>
<i>Research Methods:</i>	<i>5</i>
<i>Implementation:.....</i>	<i>6</i>
<i>Observations from Exploratory Analysis:.....</i>	<i>13</i>
<i>Implementation of the SVM Model:.....</i>	<i>14</i>
<i>Conclusion:.....</i>	<i>16</i>
<i>References:</i>	<i>17</i>
<i>Papers:</i>	<i>17</i>
<i>Sites:.....</i>	<i>17</i>
<i>Appendix:.....</i>	<i>18</i>

Abstract

Sentiment is a thought, opinion, or idea based on a feeling about a situation, or a way of thinking about something (as per Cambridge dictionary). The process of analyzing such sentiments and determining which of them is positive, negative or neutral is called Sentiment Analysis or Opinion Mining. In this paper we have used an approach for sentiment analysis implementing support vector machines (SVMs) to analyse the tweets of microblogging site Twitter to classify the available data from the collected dataset. The sentiments are categorized into three categories which are positive, negative and neutral using a supervised machine learning algorithm for classification known as SVM. After applying the SVM model, we were able to classify the texts of a given tweet into positive, negative or neutral classes with the succession rate of 60%.

Keywords: Sentiment Analysis (SA), tweet, twitter, sentiment, social media, Opinion mining, Machine learning, Support Vector Machine (SVM)

Introduction:

This research paper aims to build a sentiment analysis model which will help us to categorize tweets from people based on their sentiments, whether they are positive, negative or neutral. Capturing sentiment in language is important in these times where decisions and reactions are created and updated in seconds. But which words actually lead to the sentiment description and may have impact on decision making for any organization will be developed by using this model.

Nowadays, to understand people's emotions is very much essential for businesses since customers express their thoughts and feelings more openly than ever before with the evolve of internet. Market research, customer

feedback, social media conversations and automatically analyzing customer feedback are few of the applications where sentiment analysis has been used extensively to provide help to businesses in order to gain insight of customers opinions, such as analyzing ratings in survey responses, allows brands to listen attentively to their customers, and tailor products and services to meet their needs.

In our paper, to perform sentiment analysis and produce some useful results, we have implemented supervised machine learning approach on social media microblogging site called Twitter and classifying texts as positive, negative or neutral using Support Vector Machines method. R language is used in this research to implement the classification algorithm on the collected data.

Research Questions to be addressed:

Q1: How does sentiment analysis work?

Q2: How can we classify the tweets among sentiments like positive, negative or neutral?

Q3: How will the sentiment analysis approach in this paper help any business in their decision making?

Literature Review

Opinion Mining or Sentiment analysis has been handled as a Natural Language Processing (NLP) in various industries. Different machine learning approaches and methodologies implemented by researchers to analyse the sentiments and tweets from social media site Twitter. To begin with, from being a document level classification task (Turney, 2002; Pang and Lee, 2004), it has been handled at the sentence level (Hu and Liu, 2004; Kim and Hovy, 2004) and more recently at the phrase level (Wilson et al., 2005; Agarwal et al., 2009).

Recent results on sentiment analysis of Twitter data are by Alec Go, Richa Bhayani, and Lei Huang in their paper (2009). They applied distant learning to acquire sentiment data. They used tweets ending in positive emoticons like “:)” “:-)” as positive and negative emoticons like “:(” “:- (“ as negative. They built models using Naive Bayes, MaxEnt and Support Vector Machines (SVM), and they reported SVM performed better than other classifiers. One more research by Kamal Nigam, John Lafferty, Andrew McCallum used Maximum Entropy technique for text classification in their paper in 1999. It is a probability distribution technique widely used for various natural language processing task.

Another paper by Alexander Pak and Patrick Paroubek (2010) collected data following a similar distant learning paradigm. They performed a different classification which is subjective versus objective. For subjective data they collected the tweets ending with emoticons in the same manner as Go et al. (2009). They got objective data from the twitter accounts of popular newspapers like Washington Posts. They report that POS and bigrams both help.

Both of these approaches were primarily based on N-gram models. Moreover, the data they used for training and testing was collected by search queries and was therefore biased. In contrast, we obtained our data from trusted source which is Kaggle site. Dataset was extracted by Kaggle from twitter using Twitter's openAPIs. Also, we have performed extensive data exploration, processing and feature extraction using different methods in R language. We have used SVM as our base model for sentiment to determine the polarity of a tweet.

Research Methods:

In this section, we present the research methods that we used in order to answer our research questions.

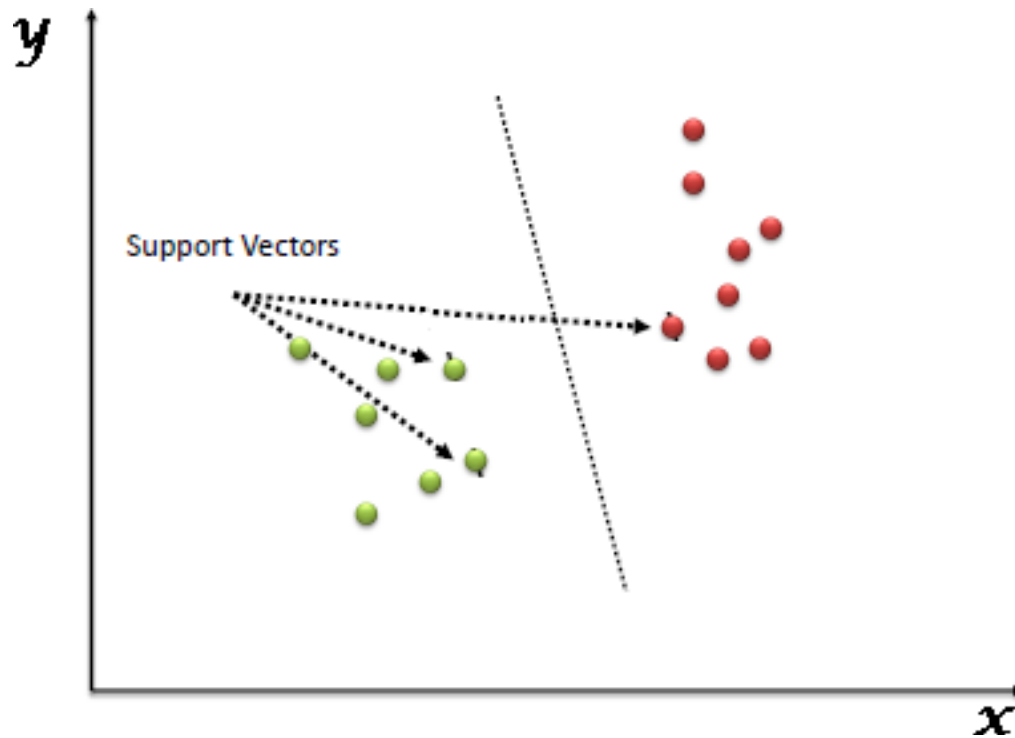
Classification Technique:

There are basically two approaches for text classification mostly used which are Machine learning approach and Lexicon based approach. We have implemented machine learning approach here.

The SVM method of supervised machine learning used as a classifier in the existing approach for the sentiment analysis. The SVM classifier only classify data into three classes and also it has accuracy approximately of 80 percent which increase efficiency of sentiment analysis.

Support Vector Machine Classifier:

Support-vector machines are supervised learning models with associated learning algorithms that analyse data used for classification and regression analysis. In the SVM algorithm, we plot each data item as a point in n-dimensional space where n is number of features with the value of each feature being the value of a particular coordinate. Then, we perform classification technique by finding the hyper-plane that differentiates the two classes very well.



Implementation:

We have performed Exploratory Data Analysis first in the paper to understand the data before data preparation. We obtained two datasets which are Training and Test data. Both contains around 30K tweets from the twitter in csv format which will be used as input to the model.

The extracted data in the dataset contains certain amount of irrelevant data from Twitter. We had to filtered out any kinds of arbitrary characters or useless information from the tweet information. R language tool is applied for filtering out this useless data. Also, any missing values, invalid values and redundant values have been treated by the Rlanguage tool using various in-built libraries.

Exploratory Data Analysis:

No of observations or records are 27481 in both training and test datasets.

Training dataset - Features Name and Data types

Column specification
textID = col_character(),
text = col_character(),
selected_text = col_character(),
sentiment = col_character()

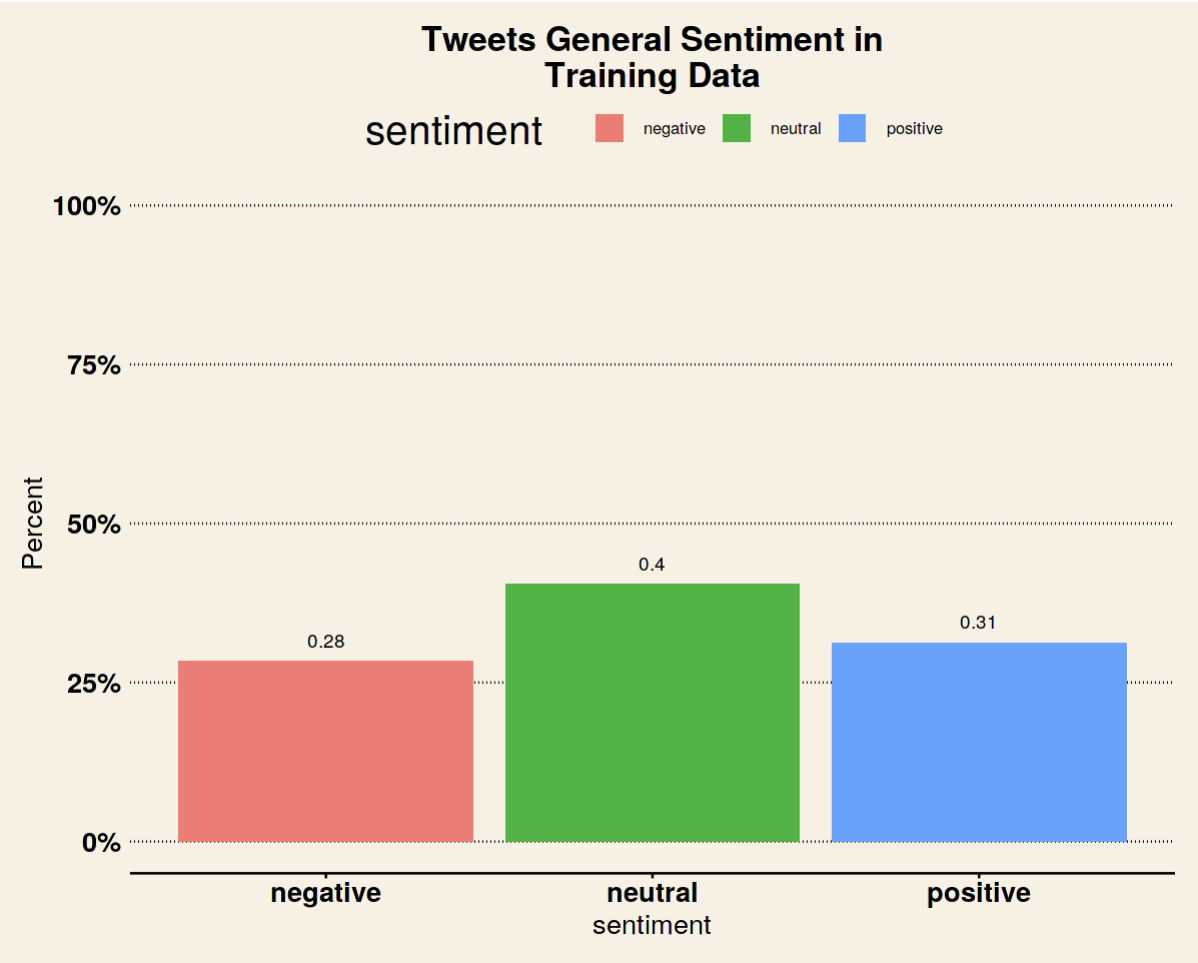
Test dataset - Features Name and Data types

Column specification
textID = col_character(),
text = col_character(),
sentiment = col_character()

Summary of Train Data

textID	text	selected_text	sentiment
Length:2481	Length:2481	Length:2481	Length:2481
Class :character	Class :character	Class :character	Class :character
Mode :character	Mode :character	Mode :character	Mode :character

Plot General Sentiments in Training Data



Observations:

- We got to know that `selected_text` is a subset of text in train set
- `selected_text` contains only one segment of the sentence
- We can see in plot that 40% tweets are neutral, 31% positive and 28% are negative

Data Cleaning:

Clean the Tweets:

We have treated the missing, null and blank values in this step. Remove the unwanted text, special characters, white spaces, punctuations, hashtags etc. in the tweets.

We have made user defined function named **`cleanCorpus()`** to treat different types of data inconsistency. It contains small sub programs to further treat data irregularities.

For example: `removeHtmlTags()`, `removeHashTags()`

See appendix section 1

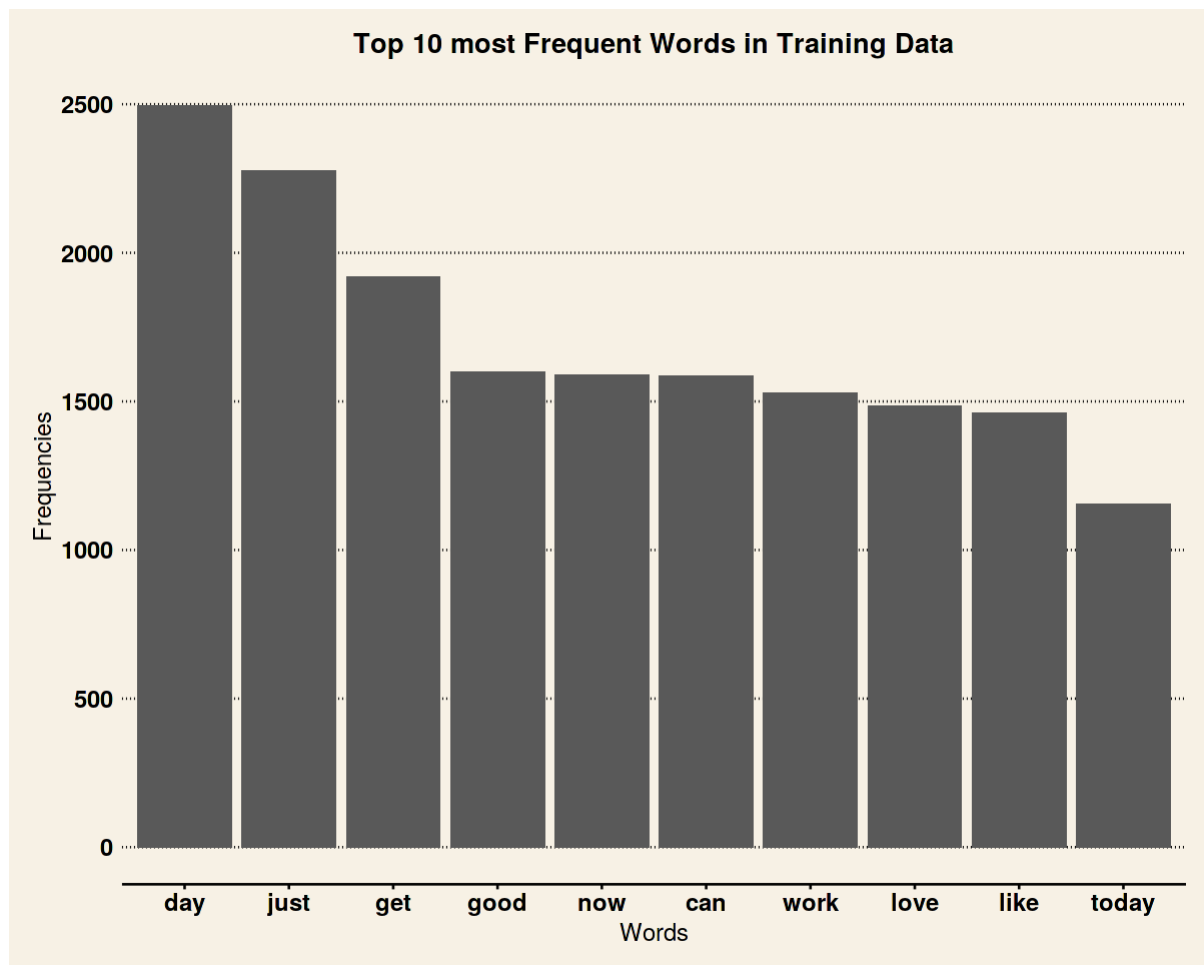
To analyze the word **frequency** we have created function.

See appendix section 2

Top 10 words

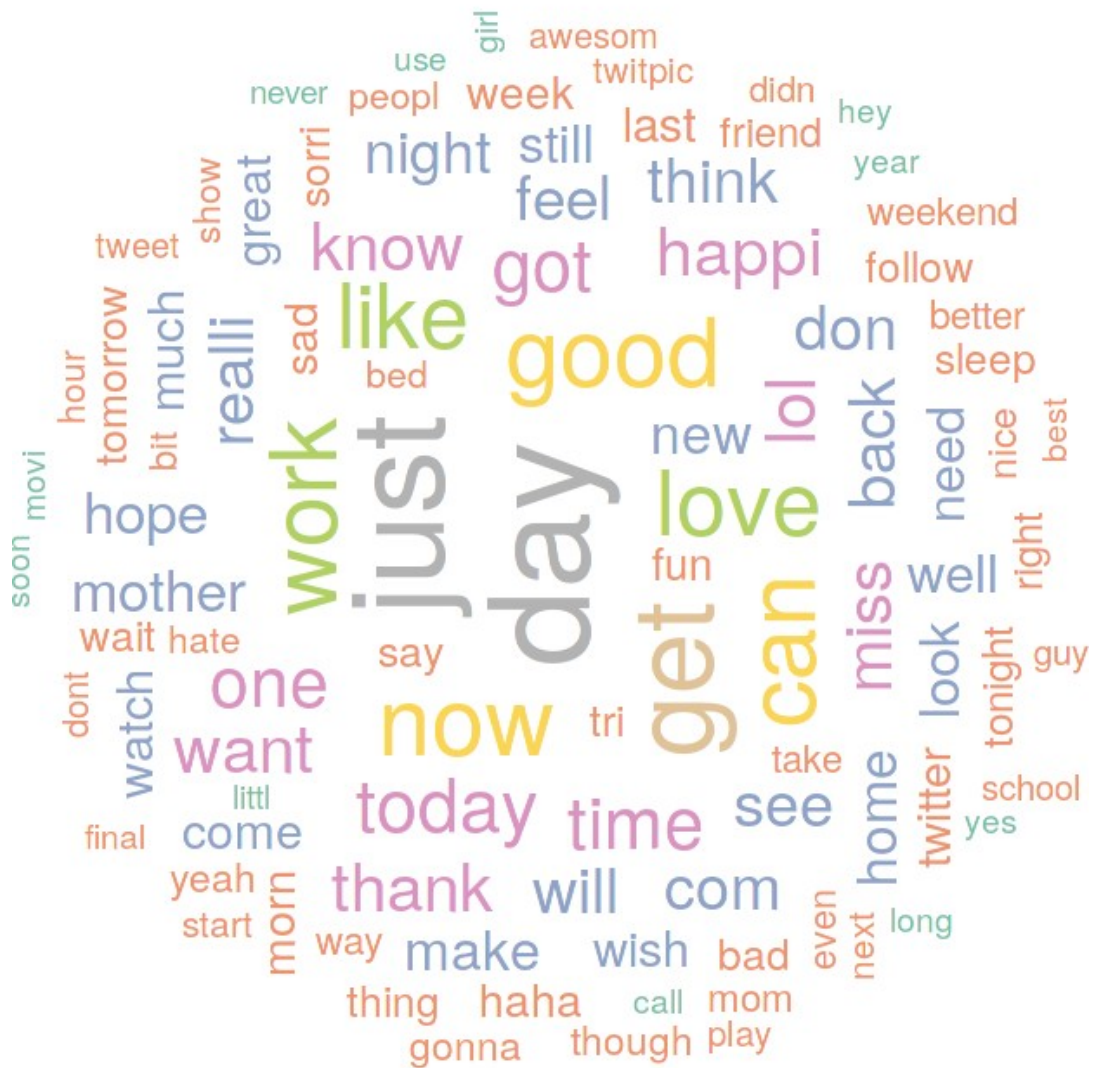
word	freq
day	2497
just	2278
get	1918
good	1600
now	1590
can	1587
work	1530
love	1486
like	1462
today	1155

Plot of word frequencies



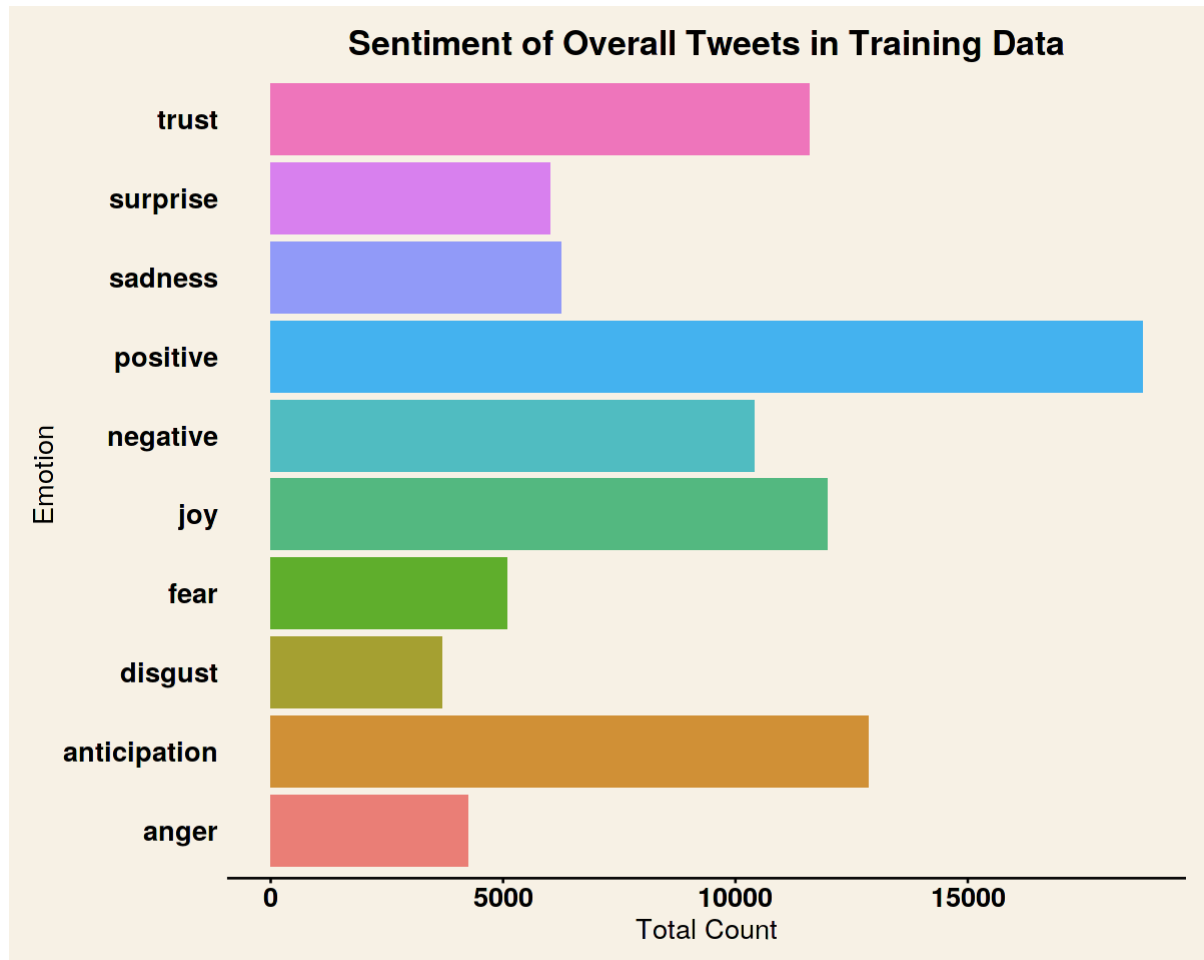
Print the Word Cloud

Overall Tweets Wordcloud of Training Data



Overall Tweets Emotion

We used NRC general-purpose lexicons dictionaries tidytext package To analyse the emotions in the train dataset.



Observations from Exploratory Analysis:

- 40% of tweets are neutral, 31% positive, and 28% negative out of 27481 records
- Top 10 Frequent words are day, just, get, good, now, can, work, love, like, today
- Top 10 frequent positive words are day, love, good, happy, thank, mother, just, hope, great, like
- Top 10 frequent negative words are just, miss, get, can, now, work, like, feel, sad, day
- Top 10 frequent neutral words are just, get, day, work, can now, lol, like, time, god.
- As per NRC sentiment dictionary, mostly tweets showed positive emotion, followed by anticipation, joy.
- Least of the tweets showed disgust, anger, and fear
- In the selected text, most of the tweets showed positive emotion, followed by anticipation and joy.
- There seems no difference between tweets and selected text in terms of emotion.
- There was a difference in the emotion of positive, negative, and neutral tweets.
- Positive tweet's highest emotion was positive, followed by joy and anticipation

Implementation of the SVM Model:

We have applied the Support Vector Machine classifier to the datasets in R language using the library `e1071`.

We have trained the model first then calculate the accuracy based on confusion matrix.

After that, we have validated the model using the test data.

Confusion Matrix of Model:

	Negative	Neutral	Positive
Negative	94	50	24
Neutral	110	225	72
Positive	22	39	170

Accuracy of the Model:

As per the confusion matrix, the calculate accuracy of the model is **61.39%**.

Test the Model:

Input data to model is `df_test` data which is the test data set.

Top 10 input rows-

text.text	
1	I`d have responded, if I were going
2	Sooo SAD I will miss you here in San Diego!!!
3	my boss is bullying me...
4	what interview! leave me alone
5	Sons of ****, why couldn`t they put them on the releases we already bought
6	http://www.dothebouncy.com/smf - some shameless plugging for the best Rangers forum on earth
7	2am feedings for the baby are fun when he is all smiles and coos
8	Soooo high
9	Both of you
10	Journey!? Wow... u just became cooler. hehe... (is that possible!?)

Output for top 10 rows:

	model_sentiment	model_prob	text.textID
1	neutral	0.6209012	cb774db0d1
2	negative	0.7934668	549e992a42
3	negative	0.6089539	088c60f138
4	negative	0.6140321	9642c003ef
5	negative	0.6173819	358bd9e861
6	neutral	0.6447695	28b57f3990
7	positive	0.6680402	6e0c6d75b1
8	neutral	0.6277401	50e14c0bb8
9	neutral	0.5495634	e050245fbd
10	positive	0.6640122	fc2cbefa9d

text.selected_text		text.sentiment
1	I`d have responded, if I were going	neutral
2	Sooo SAD	negative
3	bullying me	negative
4	leave me alone	negative
5	Sons of ****,	negative
6	http://www.dothebouncy.com/smf - some shameless plugging for the best Rangers forum on earth	neutral
7	fun	positive
8	Soooo high	neutral
9	Both of you	neutral

Conclusion:

After applying the SVM model, we were able to classify the texts of a given tweet into positive, negative or neutral classes with the succession rate of 60%. We also performed the testing of the model with the test dataset and predicted the sentiments based on developed model.

Our results show that the features that enhance the performance of our classifiers the most are features that combine polarity of words. The tweet syntax features help but only marginally.

At the end of study, we can say that large data can be used as data that produces more useful information with more accuracy. By applying sentiment analysis, significant information will be easier to obtain. Therefore, based on the studies and implementation that have been done in this paper, it can be concluded that this study produces an approach that can be used to perform sentiment analysis for the tweets/texts by doing positive, negative, and neutral classification on input data.

References:

Papers:

- Priyanka Tyagi, R.C. Tripathi, A Review towards the Sentiment Analysis Techniques for the Analysis of Twitter Data, February 8, 2019
- Mika V. Mäntylä, Daniel Graziotin, Miikka Kuutila, The Evolution of Sentiment Analysis, 22 March 2017,
- Tony Mullen and Nigel Collier, Sentiment analysis using support vector machines with diverse information sources, Jan 2004
- Alexander Pak, Patrick Paroubek, Twitter as a Corpus for Sentiment Analysis and Opinion Mining, Jan 2010
- Cristianini and J. Shawe-Taylor, An Introduction to Support Vector Machines and other Kernel-based Learning Methods. Cambridge University Press, 2000.
-

Sites:

- [https://monkeylearn.com/sentiment-analysis/#:~:text=Sentiment%20analysis%20\(or%20opinion%20mining,b rand%20reputation%2C%20and%20understand%20customers](https://monkeylearn.com/sentiment-analysis/#:~:text=Sentiment%20analysis%20(or%20opinion%20mining,b rand%20reputation%2C%20and%20understand%20customers)
- <https://monkeylearn.com/blog/sentiment-analysis-applications/>
- http://rstudio-pubs-static.s3.amazonaws.com/380625_ee3fca1c653f466486edd7c512fd3e47.html
- <https://rpubs.com/mbhargav68/TweetClassificationModel>
- https://cran.r-project.org/web/packages/dataPreparation/vignettes/train_test_prep.html
- <https://www.lexalytics.com/technology/sentiment-analysis>

Appendix:

Section 1. Sub programs for tweet cleanup- removeHtmlTags(), removeHashTags()

```
# helper functions for text cleaning
removeHtmlTags <- function(x)
  (gsub("<.*?>", "", x))
removeHashTags <- function(x)
  gsub("#\\S+", " ", x)
removeTwitterHandles <- function(x)
  gsub("@\\S+", " ", x)
removeURL <- function(x)
  gsub("http:[[:alnum:]]*", " ", x)
removeApostrophe <- function(x)
  gsub("'", "", x)
removeNonLetters <- function(x)
  gsub("[^a-zA-Z\\s]", " ", x)
removeSingleChar <- function(x)
  gsub("\\s\\S\\s", " ", x)
```

Section 2. Analyze word frequency

```
# function get word frequency
wordFrequency <- function(corpus) {
  dtm <- TermDocumentMatrix(corpus)
  rm(corpus)
  # convert to matrix
  m <- as.matrix(dtm)
  rm(dtm)
  # sort by word frequency
  v <- sort(rowSums(m), decreasing=TRUE)
  rm(m)
  # calculate word frequency
  word_frequencies <- data.frame(word = names(v), freq=v)
  return(word_frequencies)
}
```