**Objective:**

This analysis describes a system for real-time analysis of public sentiment toward Indian Premier League , a professional Twenty20 cricket  in 2018 as expressed on Twitter, a microblogging service.

Twitter has become a central site where people express their opinions and views on ipl match, different team and players. It provides a unique opportunity to gauge the relation between expressed public sentiment and IPL events. In addition, sentiment analysis can help explore how these events affect public opinion.

The system demonstrated here analyses sentiment in the entire Twitter traffic about IPL, delivering results instantly and continuously.

The statistical classifier we use for sentiment analysis is a naïve Bayes model on unigram features. Based on the data we collected our classifier performs at 50.01% accuracy on the five categories classification of negative, positive, neutral, very positive, very negative.

**APPROACH**

Twitter is a popular microblogging service where users create status messages (called “tweets”). These tweets sometimes express opinions about different topics. We propose a method to automatically extract sentiment (positive or negative) from a tweet. Tweets are more casual and limited to 140 characters of text

With the help of the Twitter API,we extract large amounts of tweets in English along with emoticons. The Twitter API has a parameter that specifies which language to retrieve tweets in. We extract tweets containing hashtag #ipl2018 .We have extracted tweets for the period 1-May-2018 to 17-May-2018 for the analysis.

**Dataset description**

The dataset consists of one tweet per line. There are 19,301 tweets in the dataset

**Pre-processing**

The following tasks are performed during data pre-processing

* Transformed raw text with normalization techniques such as tokenization & case folding
* Only alphabetical & alphanumerical tokens are stored. Tokens with punctuation marks, strip whitespaces
* Remove retweet
* The effect of abbreviations and misspellings in the messages are ignored,
* Remove new line

Remove new line characters from the tweet

* Remove URL

The url's which are present in the tweet are shortened using TinyUrl due to the limitation on the tweet text. These shortened url's did not carry much information regarding the sentiment of the tweet. Thus these are removed.

* Remove Target   
  The target mentions in a tweet done using '@' are usually the twitter handle of people or organisation. This information is also not needed to determine the sentiment of the tweet. Hence they are removed.
* Hashtags   
  A hashtag is a type of metadata tag  in Twitter and hence are very critical. In order to capture the relevant information from hashtags, all special characters and punctuations are removed before using it as a feature.
* Numbers

Numbers are not useful when measuring sentiment. Thus, numbers which are obtained as tokenised unit from the tokeniser are removed in order to refine the tweet content

* Sequence of Repeated Characters

Tweets are written in random form, without any focus given to correct structure and spelling. Spell correction is an important part in sentiment analysis of user-generated content. People use words like 'coooool' and 'hunnnnngry' in order to emphasise the emotion. In order to capture such expressions, we replace the sequence of more than three similar characters by three characters.

* Stop-word Removal  
  Stop words don't carry any sentiment information and thus are of no use to us. We create a list of stop words like ipl,vivo,he etc and ignore them while scoring the sentiment.

**ANALYSIS AND VISUALIZATIONS**

Now we will provide the details of implemented analyses and their visualizations on the processed real-time tweets. We implemented four main group of data visualizations. For visualization purposes we utilized several R provided packages, such as tm,reshape, , ggplot, and wordcloud, dplyr and tidytext.

To analyse the sentiment of a tweet is to consider the text as a combination of its individual words and the sentiment content of the whole text as the sum of the sentiment content of the individual word.

 The tidytext package contains several sentiment lexicons. We use bing lexicon for our analysis. This lexicon is based on unigrams.  The bing lexicon categorizes words in a binary fashion into positive and negative categories

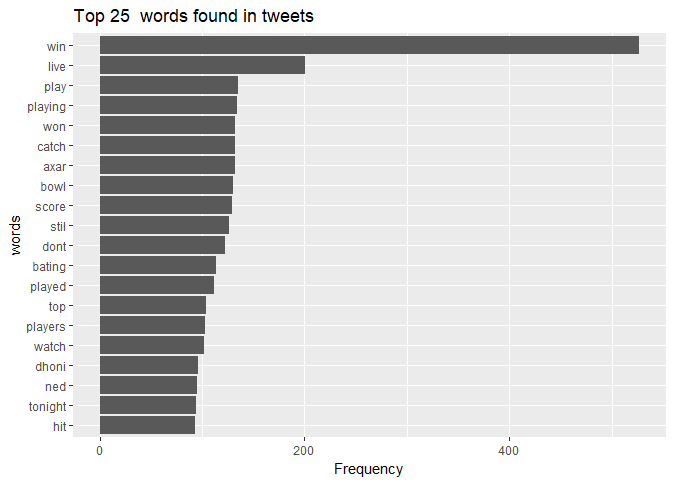
Dictionary-based methods calculate the total sentiment of a piece of text by adding up the individual sentiment scores for each word in the text.

**Top 25 words in the tweet.**

The following 5 words are extensively used in the tweets.

* win
* live
* play
* playing
* won

The plot for top 25 words used in the tweet is given below

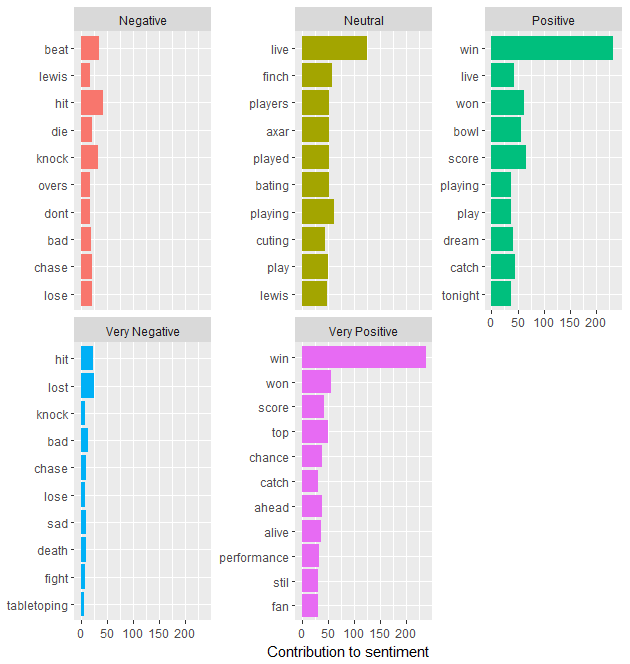


**Most common positive and negative words**

The below table shows most common positive ,negative, very positive ,very negative and neutral words

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Negative | Very Negative | Neutral | Positive | Very Positive |
| beat | hit | live | win | win |
| lewis | lost | finch | live | won |
| hit | knock | players | won | score |
| die | bad | axar | bowl | top |
| Knock | chase | played | score | chance |
| overs | lose | bating | playing | ahead |
| dont | sad | playing | play | alive |
| bad | death | cuting | dream | performance |
| chase | fight | play | catch | stil |
| lose | tabletoping | lewis | tonight | fan |

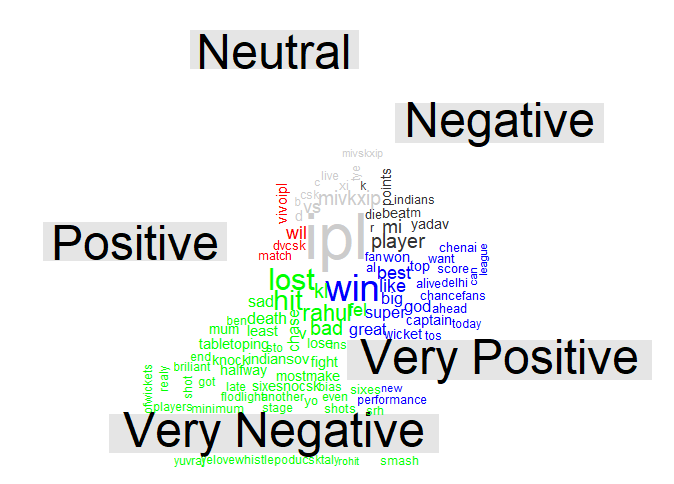
The graphical representation of most common positive ,negative, very positive ,very negative and neutral words is given below.



**Plot a cloud comparing the frequencies of words across documents.**

A comparison cloud compares the relative frequency with which a term was used in two or more tweet . It does not simply merge two word clouds. Rather, it plots the difference between the word usage in the tweet.

The size of a word’s text in the below picture is in proportion to its frequency within its sentiment. We can use this visualization to see the most important positive and negative words, but the sizes of the words are not comparable across sentiments.



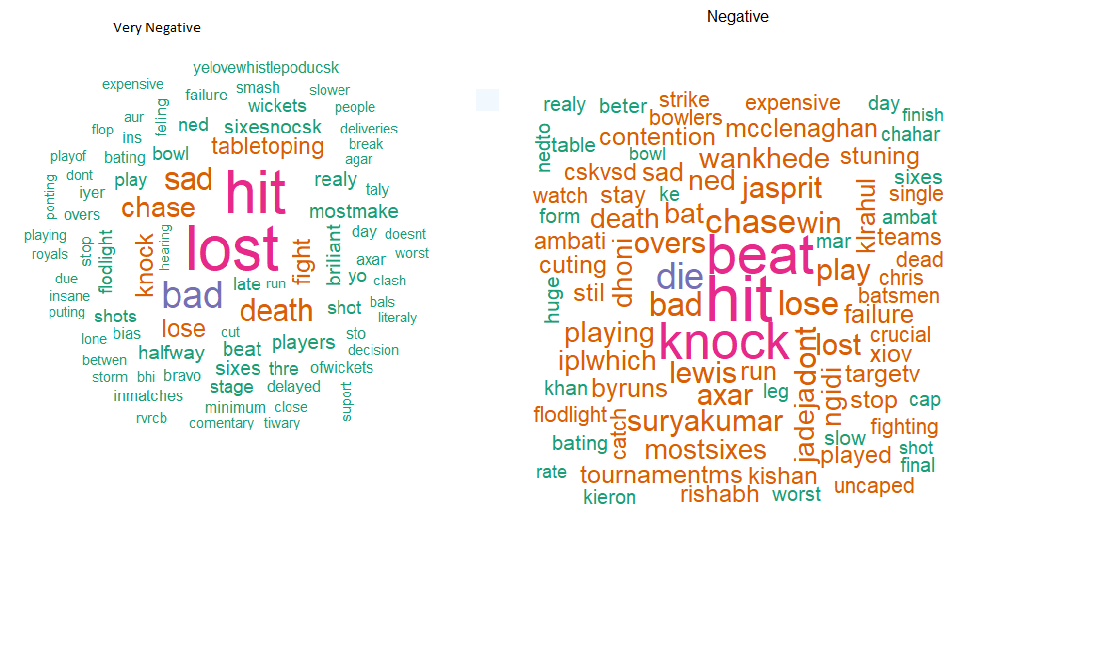
**Building the Word clouds**

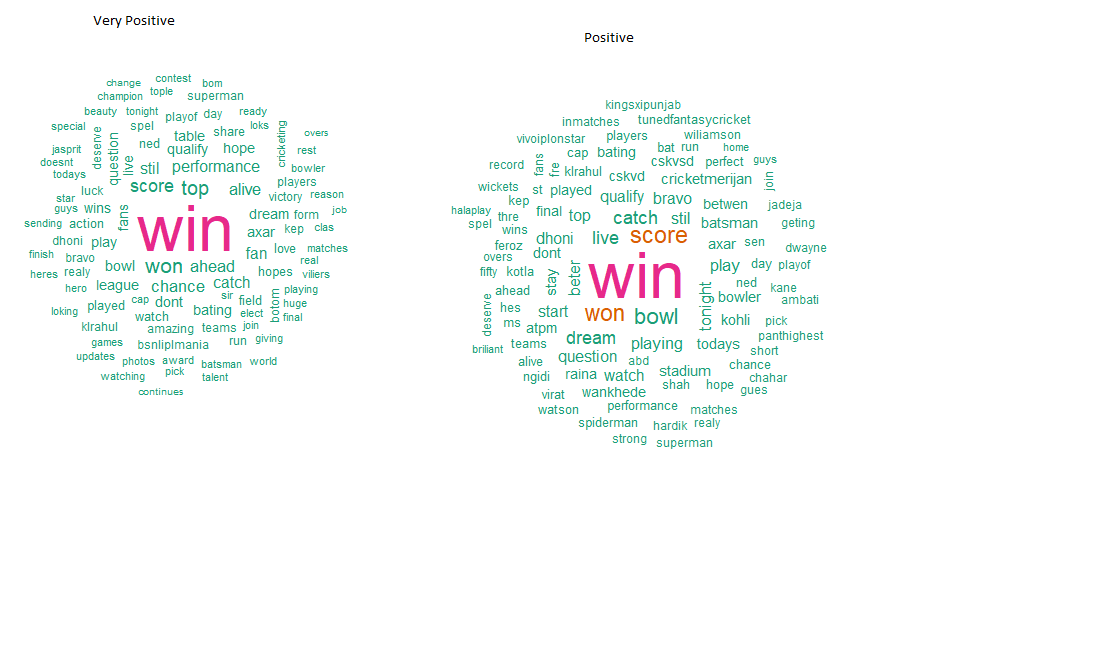
The following steps are performed

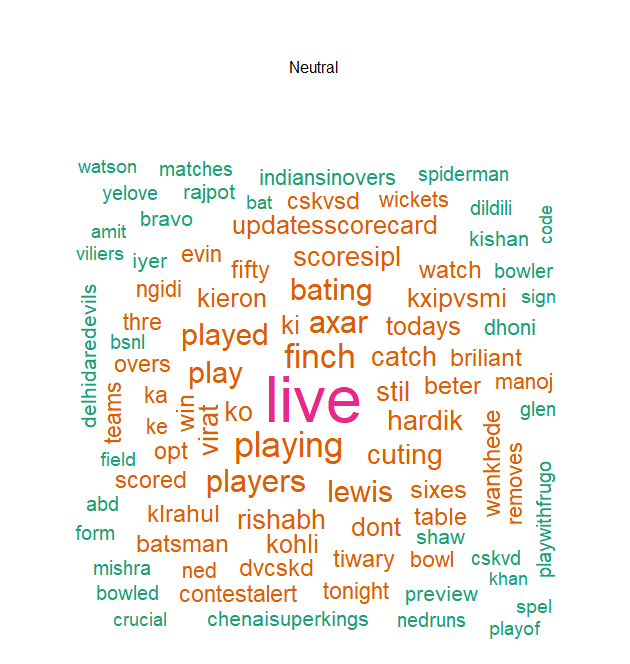
* Subset and filter all the tokens based on positive, negative,very positive, very negative and neutral
* Generate a document feature matrix which is a sparse matrix consisting of the frequency of words that occur in a document
* Transformed raw text with normalization techniques such as tokenization & case folding
* Only the alphabetical & alphanumerical tokens are stored. Tokens with punctuation marks, strip whitespaces, special characters, url & numbers, words like lt ,gt etc ignored
* Perform language-specific pre-processing techniques such as stop word removal.
* The effect of abbreviations and misspellings in the messages are ignored,
* Frequency count of each token per message is stored as Bag of Words
* Picks and selects the most commonly occurring words in the sentences i.e the words having the highest frequencies and plots them, the more the frequency of a particular word the greater is the size of the word in the word-cloud.

Here is the summary.

* The group of tokens (hit, lost, bad , knock,bad ,death) are related to very negative sentiment
* The group of tokens(beat,hit,die,don’t) are related to negative sentiment
* The group of tokens(win,live,won,bowl) are related to positive sentiment
* The group of tokens(win,won,score,top,chance) are related to very positive sentiment
* The group of tokens(live,finch,players,axar) are related to neutral sentiment





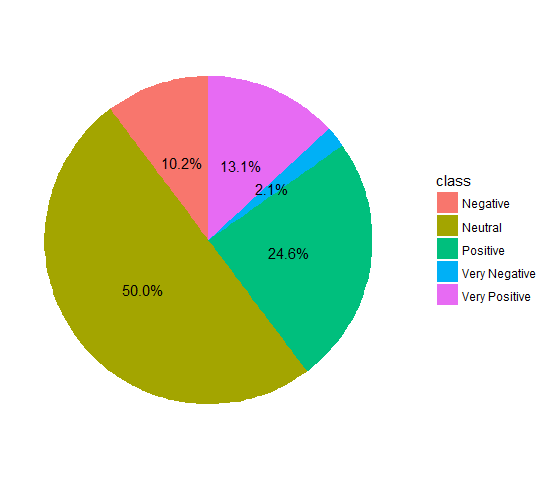


# We use “RSentiment” package to analyse sentiment (Positive, Negative, Very Positive, Very Negative, Neutral, Sarcasm) of the each tweet .

The number of sentences which are positive, negative, very positive, very negative, neutral and sarcasm is given below. There is no sarcasm.

|  |  |
| --- | --- |
| Category | # of sentences |
| Negative | 908 |
| Very Negative | 184 |
| Neutral | 4440 |
| Positive | 2180 |
| Very Positive | 1166 |

Out of 8878 tweets 50.27% tweets are neutral, 24.55% tweets are positive, 13.13% tweets re very positive and 2.07 % tweets are very negative. The graphical representation is given below.



We also calculate sentiment of each tweet by using calculate sentiment ()

 This function uses the score predicted by calculate score method to classify the sentences. It classifies sentences into 6 categories:

* Positive
* Negative
* Very Positive
* Very Negative
* Sarcasm
* Neutral

The score of each sentence of the text fed on basis of following conditions.

* Parts of Speech tagging of each word of the sentence
* Identifying various conditions of occurrences and presence of Verb, Adverb, Adjective, Noun etc in each sentence
* Accordingly, marking the presence of words of positive and negative sentiment and order of their occurrence
* Considering negation
* Checking for sarcasm on basis of presence of punctuation
* Checking presence of emoticons in the text

Here is the summary of the score

* 0 indicates neutral sentiment.
* Positive value indicates positive sentiment.
* Negative value indicates negative sentiment.
* 99 indicates sarcasm

**Naïve Bayes Classification**

Naive Bayes text classifier is used to compute the probabilities of a tweets being (positive, negative, very positive, very negative, neutral)

Naive Bayes classifiers are a class of simple linear classifiers which use conditional probability models based on **Bayes Theorem**

The naive Bayes Classifier combines this model with a decision rule, which picks up the hypothesis with maximum probability. In simple words, we pick the class which has maximum value for

Where  are the number of inputs and Y is a categorical response variable and are the number of class labels.

* Extracting tokens for all tweets in the dataset will result in 70,635 tokens. However, not all of these tokens are useful in the classification. Going through the extracted tokens Going through the extracted tokens Going through the extracted tokens, we removed the ones with less than 5 and more than 500 times frequency in the dataset, since those tokens are either too rare or too common, and do not contribute to the content of the messages.
* Divide the corpus in two parts: the first 70% of the tweets were separated for training and the remaining 30% for testing
* As all the messages are fairly short, we did not use any kind of method to reduce the dimensionality of the training space

The overall accuracy of the model is **50.01%.** The confusion matrix is given below.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Negative | Neutral | Positive | Negative | Very Positive |
| Sensitivity | 0.00000 | 1.00000 | 0.00000 | 0.00000 | 0.00000 |
| Specificity | 1.00000 | 0.00000 | 1.00000 | 1.00000 | 1.00000 |
| Pos Pred Value | NaN | 0.5002 | NaN | NaN | NaN |
| Neg Pred Value | 0.8979 | NaN | 0.7544 | 0.97935 | 0.8686 |
| Prevalence | 0.1021 | 0.5002 | 0.2456 | 0.02065 | 0.1314 |
| Detection Rate | 0.00000 | 0.5002 | 0.00000 | 0.00000 | 0.00000 |
| Detection Prevalence | 0.00000 | 1.00000 | 0.00000 | 0.00000 | 0.00000 |
| Balanced Accuracy | 0.50000 | 0.50000 | 0.50000 | 0.50000 | 0.50000 |

**Conclusion:**

In this study Naïve Byes Classification model is used for categorizing the tweets as positive, negative, very positive, very negative, neutral. The unigram feature extractor is used to calculate the sentiment. The overall accuracy of the model is 50.01%

This proposed system would be easy for user to obtain the summarized report about the opinion related to IPL 2018 from Twitter.. We tentatively conclude that sentiment analysis for Twitter data is not that different from sentiment analysis for other genres.

We will also need to explore even richer linguistic analysis, for example, parsing, semantic analysis and topic modelling.

Proposed model can be enhanced in following ways:

* Parts Of Speech tag can be utilized to interpret emotions in a better way.E.g. ’over’ as a verb conveys a negative emotion, but as a noun it is neutral
* Usage of bigrams need to be included as classifiers handling negations always

produce unexpected results.

* Need to translate acronym before analysis. For example, lol is translated to laughing out loud.
* We focus only on English sentences, but Twitter has many international users. It should be possible to use our approach to classify sentiment in other languages.

**ANNEXTURE**

1. R Code for Sentiment Analysis (Text\_Mining\_Assignment-I\_Rajib\_Mandal\_TA17002.R)
2. R Code for Twitter Data Extraction(Textmining\_Project\_Rajb\_Mandal\_Twitter\_Data\_Extraction\_TA17002.R)
3. Program Output(Text\_Mining\_Assignment-I\_Console\_Rajib\_Mandal\_TA17002.txt)
4. Dataset(IPL-2018-11.csv)