A Dissertation
On

" Sentiment Based Tool For Brand Management "

Submitted in partial fulfillment of the requirement
for the award of degree of

MASTER OF TECHNOLOGY
Software Engineering
Delhi Technological University, Delhi

SUBMITTED BY
Parul Hooda
2K15/SWE/12

Under the Guidance of

DR. AKSHI KUMAR

Assistant Professor
Department of Computer Engineering
Delhi Technological University

DEPARTMENT OF INFORMATION TECHNOLOGY
DELHI TECHNOLOGICAL UNIVERSITY
2017

Certificate

This is to certify that the dissertation entitled **"Sentiment Based Tool For Brand Management"** has been submitted by **Parul Hooda (Roll Number: 2K15/SWE/12),** in partial fulfillment of the requirements for the award of Master of Technology degree in Information Systems at **DELHI TECHNOLOGICAL UNIVERSITY.** This work is carried out by her under my supervision and has not been submitted earlier for the award of any degree or diploma in any university to the best of my knowledge.

AkshiKumar 23/6/17

DR. AKSHI KUMAR

Project Guide

Assistant Professor

Department of Computer Engineering

Delhi Technological University

Acknowledgement

I am very thankful to **DR. AKSHI KUMAR** (Assistant Professor, Deptt. of Computer Engineering) for providing immense support and guidance throughout the project.

I would also like to express gratitude to Mrs. Arunima (Research Scholar, Delhi Technological University) for providing me continuous support and guidance during this project.

I would also like to express my gratitude to the university for providing us with the laboratories, infrastructure, testing facilities and environment which allowed us to work without any obstructions.

I would also like to appreciate the support of my friends Aanchal, Indu and Swati. Also the help provded by our lab assistants, seniors who aided us with all the knowledge they had regarding various topics.

PARUL HOODA

(2K15/SWE/12)

Abstract

Sentiment analysis (also known as opinion mining) refers to the use of natural language processing (NLP), text analysis and computational linguistics to identify and extract subjective information from the source materials. Generally speaking, sentiment analysis aims to determine the attitude of a writer or a speaker with respect to a specific topic or the overall contextual polarity of a document.

Globally, business enterprises can leverage opinion polarity and sentiment topic recognition to gain deeper understanding of the drivers and the overall scope. Subsequently, these insights can advance competitive intelligence and improve customer service, thereby creating a better brand image and providing a competitive edge. The e-commerce industry is benefiting greatly by utilizing sentiment analysis. Generally, on e-commerce portals, buyers often express their opinions in the form of comments (positive or negative) for the products they have purchased, making this a huge data trove for sentiment analysis. Correspondingly, analysis of such opinion-related data (comments) can provide deep-insights to the key stakeholders. A thorough sentiment analysis reveals deep-insights on the product, quality and performance. Additional insights that can be extracted using sentiment analysis include.

List of Figures

Contents

Chapter 1. INTRODUCTION

1.1 Motivation

"Management is, above all, a practice where art, science, and craft meet", as quoted by Henry Mintzberg suffices enough to support the combination of technology and social media in today's 21st century to yield better results in the domain of business intelligence.

With the escalation of Web 2.0 applications which include various social networking sites and blogs amongst them, there came the relevance attached to the user generated data. The user generated data is virtually about anything and everything that surrounds us, like the products we buy, live news feeds, movies, people etc. Here, the relevance refers to the parameters with the content, like the ratings, recommendations, reviews and comments.

To analyse the dynamically increasing bulk content over the internet, the use for automated mining techniques is required, as it is almost impossible to do so manually. With this need, arrives the introduction of sentiment analysis or opinion mining, which allows the researchers and analysts to know how the public is being influenced by situations and events around them. The best approaches combine data from a diverse set of data resources and add customer sentiment data to the mix.In addition, information that includes geo-location data, customer-preference analysis, and market-trending information based on contextual text analytics can collectively provide a level of understanding of market dynamics previously unavailable to the industry.

Opinion Mining is spreading by the day by being so useful in everyday life. For example, Apple is diving deeper into artificial intelligence by acquiring the startup company Emotient, a cloud-based service that uses artificial intelligence to read emotion through the analysis of facial expressions. Also, researchers at the University of Arizona's Artificial Intelligence Laboratory use text mining to search for relevant terms and use sentiment analysis to study what motivates hackers to try to predict how they might act in future attempts at data breaching.Recently companies have started to

realize the potential of sentiment analysis and its power to deliver them smarter and better results.

Twitter is considered as one of the most prevalent micro-blogging platform which is used by all, from politicians to a common man, companies to activists, etc. Here, people can post their views on any topic and even re-tweet or reply to someone else's tweet, through which they are indirectly expressing their sentiment. The interest of this thesis is to do an analysis of a sports event called IPL (Indian Premier League' 17) for Twitter to understand the brand value of different teams and their popularity amongst users. The motivation of the thesis is to scrutinize the matches held amongst different teams, where various supporters and stakeholders are involved.

1.2 Scope

The technologies associated with Sentiment Analysis triggers us to confidently, and rather quickly act on customer data and trending social media issues .As it collects and combines the comments and data from various social media platforms, we can be sure that you're seeing the big picture in conversations about your brand, without missing "little things" that could become big customer-relations headaches or missing opportunities that could lead to increased conversions and customer loyalty. For eg:

- Which brands in your industry hold the highest share of voice on social media.
- Who is in conversations about your brand, products, industry, and competitors.
- What topics in your industry generate the most buzz.
- Which influences are talking about your brand and your competitors' brands.

Sentiment analysis can tell you the answers to all these questions and more. It's impossible to ignore the value that it adds to the industry.Sentiment analysis bases its results on factors that are inherently "human," so it is bound to be a major driver of many business decisions in future. It allows us to harness the wisdom of the crowd rather than only hear the opinions of a select few self-appointed "experts" who just happen to comment the loudest and most frequently.Taking sentiment analysis into account from a broad base provides a holistic view of how your customers (and potential customers) feel about you, allowing you to build a brand where every

opinion counts and every sentiment affects decision-making.This thesis provides an insight to the survey done on the different teams involved in the sports event of IPL. It takes into account the tweets of the users on Twitter as a dataset, from which we analyse their polarity and their sentiments related to those teams. Here, we further find the popularity of different players, which includes the tweets related to the most emerging and both Indian and foreigner players. Later we apply three different supervised machine learning techniques and compare their accuracy with each other. All the analysis is done on the R tool.

1.3 Project/Research Goals

Opinion mining is an area of computational study that is normally used to find the sentiment or views of a person or user by dealing with opinion-oriented natural language processing, and helps different firms and other facility providing businesses to get an understanding of what their consumers are thinking. Here, in this thesis, we tend to analyse the sentiments of the public on the event of IPL for some brand management firm. This tool will help the firm to know the sentiment and polarity of the attitude of the users towards the teams involved in the league of matches, thereby aiding them to make a decision as to with which particular player or team they can affiliate themselves. The objective of this thesis is to find the most popular and trending teams and players in the league so as to help in the brand management and endorsement, which eventually would be a bonus point in the field of business intelligence along with the research on opinion mining. It compares the accuracy of different machine learning techniques to lead to a more precise result. Another important aspect while is to understand how sentiment analysis is done on the analytic tool R.

1.4 Organization of the Report

The chapter wise distribution of this thesis described here which is as follows: After this, there are 5 more chapters. In chapter 2, literature survey and background work related to the thesis will be presented. Under literature review, basic whats' and hows'

of sentiment analysis, feature engineering, machine learning and its techniques, classification and how opinion mining is done on Twitter would be explored .In chapter 3,proposed framework which includes system architecture of the project is presented, it will discuss the research question of the thesis. Chapter 4, discusses implementation. Here data extraction and labeling, the different approaches of learning that are applied to the data, and its pre-processing are presented. Here three different supervised learning algorithms are used and their result on the data is depicted with the help of various formats. Chapter 5 walks us through the results of the project implemented and its analysis. Finally at the end, in chapter 6, we discuss the factors that affect the performance, challenges and conclusions along with future work. Followed by code snippets and snapshots of the system.

Chapter 2. Literature Survey and Background Work

2.1 Sentiment Analysis

Even though there exists ample and considerable amount of research on the subject of sentiment analysis,but according to Pang and Lee, the research carried out until now basically lays its focus on two things, ie. To categorize the given data into subjective or objective and then to find out the polarity of the former.

Ever since its introduction, the study done to contemplate Sentiment Analysis was mainly theoretical and survey-oriented but new forms of research have been initiated in this field due to the instigation of huge explosion of user-generated data on social media, discussion forums, blogs and reviews.As most SA studies have used or depended on machine learning approaches, the feasibility of the research lies in the fact that the amount of user generated content on the Web will provide unlimited data for training and testing of the models.The ever expanding data in the present day's situation demands us to be explored and understood. Since there is immense flow of

information in a random order, it needs to be sorted into a structured sequence from an unstructured format, which when analyzed helps us understand the direction of idea it is conveying.

”Sentiment Analysis or Opinion Mining is the computational study of people’s opinions, attitudes and emotions towards an entity .” It segregates the sentiments into basic polarity of negatives and positives, or otherwise neutral. The user’s perspective is then generalized on the basis of the scores obtained from the opinions. To sum up in simple words, “Sentiment Analysis describes a Natural Language Processing problem that attempts to differentiate opinionated text from the factual text, in case of former, determine its polarity.” Sentiment mining is quite feasible on the trending data of social web. From being coined in the mid 90’s, to gaining extensive popularity to current times, the term “Social Web” is still progressively fulfilling its purpose of bringing people together by increasing communication between them. It seems only valid to actually scrutinize the views that users hold on different portals of discussion which is where sentiment analysis plays its integral role of contemplating the behavior of the crowd. Since its inception, the web has been flooding constantly with the creativity of networking sites, thereby highlighting the impact of dynamicity of Web 2.0.

2.1.1 Classification and Approaches

Sentiment analysis is articulated as a problem of text-classification where this classification can be approached from various perspectives as suitable according to the situation. These different types of approaches can be divided into various parts such as : keyword-driven, discourse-driven, language-model-driven, or relationship-driven mostly depending on the circumstances and opinion of the person doing the sentiment analysis. These terms have been briefly described below.

2.1.1.1 Knowledge-based approach

Here, there are certain keywords whose functions are termed into sentiments. The initial task is to construct lexicons by discriminating the words on the basis of

sentiment that would distinguish into a particular class such of positive or negative.There are different ways to create lexicons. Certain publicly available word lexicons which are used in sentiment analysis for different kinds of domains are: http://twitrratr.com/ and http://www.cs.pitt.edu/mpqa/, etc. “http://twitrratr.com/” provides sentiment lexicons for Twitter sentiment analysis.

2.1.1.2 Relationship-based approach

In this approach the classification task can be approached from the distinct relationships that might exist between features and components or among their own selves.These relationships may include relationships between discourse participants or between product features. For example, if a person wants to know the sentiment of customers about a particular movie, they can compute it as a function of the sentiments on different components/features of it.

2.1.1.3 Language models

Here,the classification is done by building n-gram language models. Where the frequency or occurrence of n-grams is used. Generally, in information retrieval type classification, frequency of n-grams has shown to give better results.This frequency or occurrence is converted to TF-IDF to take into account the importance of a term in a document. However, Pang et al. (2002)indicated through their research on movie reviews that uni-gram presence is more suited for sentiment analysis. But not long after that Dave et al. (2003) ,by working on product reviews found that bi-grams and tri-grams worked better than uni-grams in sentiment classification.

2.1.1.4 Discourse structures and semantics

In this approach, Classification is done by the discourse relation between text components.In such an approach, more weight is given to the sentiment of a paragraph that is at the end of a review in the determination of the sentiment of the

whole review. If there is a need,semantics can be used in role to identify agents.For example “India beat England” is different from “England beat India”.

2.2 Machine Learning

Machine Learning, we understand is a part of artificial intelligence where there is learning of the machine using past experiences and given data, on the basis of which the performance of the system is optimized. It is mainly divided into two parts, namely, supervised and unsupervised learning, along with semi-supervised learning and the recent discoveries of reinforcement and deep learning.

2.2.1 Types of Learning

2.2.1.1 Supervised Learning

These machine learning algorithms are the ones which classify the data and label them, thereby creating a function which helps to predict the correct output for a given input. This prediction is made on the grounds of the labeling that is done on the training dataset to anticipate an output, for the data used from testing dataset. We explain it further by saying that supervised learning is actually a two step process, namely, Training and Testing.
This can also be explained by the following diagram:

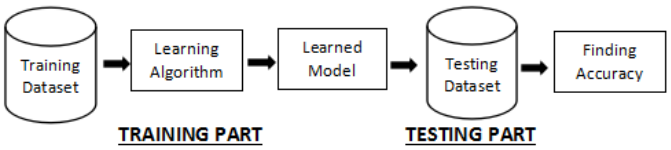


Fig.2 Concept of Supervised Algorithm

In the learning part, the model is fed with data, say dataset D, from which it learns to perform a task, say T, eventually measuring the performance of the system, using the

performance measure, say M. Here once, the training/learning part is done, we need a different set of data for the testing part of the model. It is always better to have a sufficient amount of data to achieve better accuracy. The accuracy of a model can be calculated by using the formula:

$$\text{Accuracy} = \frac{\text{No. of correctly classified observations}}{\text{Total no. of test case observations}}$$

2.2.1.2 Unsupervised Learning

In this type of machine learning, algorithms use a data driven approach, unlike the supervised algorithms which are task oriented. Another difference between the two is that, there is no labeling of data in unsupervised algorithms and hence it deals with more of data visualization. In other words, it is the Machine Learning task of inferring a function to describe hidden structure from unlabeled data. Since there is no prior classification of the data, it only deals with a simple flow of incoming data.

2.2.1.3 Semi-Supervised Learning

This learning deals with both labeled as well as unlabeled type of data. This type of learning helps one understand the behavior of the machine when both forms of data are dealt together in a combination. In simpler words, the semi-supervised learning can be considered as an extension to the supervised learning, where the goal is to train a classifier ‘f’ from both types of data such that it is better than supervised classifier trained on labeled data alone.[9] The practical nature of the algorithm adds to its advantages.

2.2.1.4 Reinforcement Learning

In this machine learning, the algorithms focuses on the maximization of rewards. It involves the presence of an agent taking actions in the given environment. Here, the learner is not notified about the actions it should be taking, rather it learns by discovering as to which particular actions lead the agent to make an increase in the reward signal. This also helps in providing a better exploratory view of the situation.

2.1.4.4 Feature Engineering

As sentiment analysis approaches generally use machine learning techniques, the main features of text are represented as feature vector.

The features used in Sentiment Analysis are:

Term presence/term frequency : In Information retrieval type classification, frequency of terms has shown to give better results.This frequency or occurrence is converted to TF-IDF to take into account the importance of a term in a document. However, Pang et al. (2002)indicated through their research on movie reviews that uni-gram presence is more suited for sentiment analysis. Pang et al. claim that this is one of the indicators which differentiates sentiment analysis from normal text classification as term frequency is taken to be a good indicator of a topic in the latter. Ironically, another study by Yang et al. (2006) proved in their study that words that appear just once in a given corpus tend to be good indicators of high-precision subjectivity.

The terms can be either uni-grams, bi-grams or n-grams where between uni-grams or n-grams, which ones give better results is still not clear. Pang et al. (2002) show that uni-grams outperform bi-grams in the study done on movie reviews, but Dave et al. (2003) claim that bi-grams and tri-grams give better product-review polarity classification.

POS (Part of speech) Tags : POS is used for dis-ambiguity which in turn helps in feature selection (Pang and Lee, 2008). With the help of POS tags, we can identify adjectives and adverbs which are usually used as sentiment indicators (Turney, 2002).

But later,he himself found that adjectives performed worse than uni-grams of same frequency.

Syntax and negation: Other features that can be employed to enhance performance are collocations and other syntactic features. In certain sentence-level classification tasks, algorithms that used syntactic features and algorithms using n-gram features found to give same kind of performance (Pang and Lee, 2008). Negation is also an important feature that should be taken into account as it has the potential to reverse the meaning of the sentiment (Pang and Lee, 2008).For a better performance,a attempts have been made to model the negation (Das and Chen, 2001, Na et al., 2004).Negation can also be expressed in other ways such as sarcasm, irony and other polarity turners.

Chapter 3. Proposed Framework

Opinion-oriented studies for sentiment analysis include genres of emotion and mood recognition, rankings through research, relevance, perspectives in text, identification of source of text etc.(Pang and Lee, 2008). Sentiment polarity analysis can be explained as the mapping of text to either of the labels that are taken out of a predefined set. These elements of the set are generally categorized into 'negative' and 'positive' or even ‘neutral’, but they can also be classified into categories of 'relevant' and 'irrelevant' and can also be range of numbers such as from 1 to 10. Sentiment analysis can be done at different levels - document, section, paragraph, sentence, or phrase levels where mostly it is done at a document level.(Wilson et al., 2005, Hatzi-vassiloglou and McKeown, 1997)

The maximum limit of characters in a tweet is 140 which becomes a limitation eventually, which twitter-users try to overcome by using abbreviated words and slang

language so as to convey their message . The language on social media is altogether different than what is found in traditional texts. For example, words like lol, rofl, wtf,etc.Some Twitter-specific terms are RT (for retweet), #(hashtag), @(at), etc. Even though sentiment analysis on twitter can be categorized as a general classification problem like any other sentiment analysis problem, it is considerably different from other studies because of the nature of the posts. Researchers usually extract Twitter posts which contain a certain term and then analyse the sentiment of these extracted tweets as Twitter literature is term based.

3.1 Research question

Sentiment analysis of tweets related to IPL'2017 sets out to do computation analysis of their sentiments. It attempts to extract tweets about this event from Twitter. In this study, the classification is done by taking three standard classes into consideration, which are positive, negative and neutral. It examines the ways of how the operations are performed on the dataset using different algorithms, how the output is represented and recommends to understand the comparison between them and analyse their accuracy to contemplate a better performance for the tool.

3.2 System Architecture

As we have used supervised algorithms on the data set, the architecture is divided into two parts: Training and Classification

The diagram for the architecture is given below:

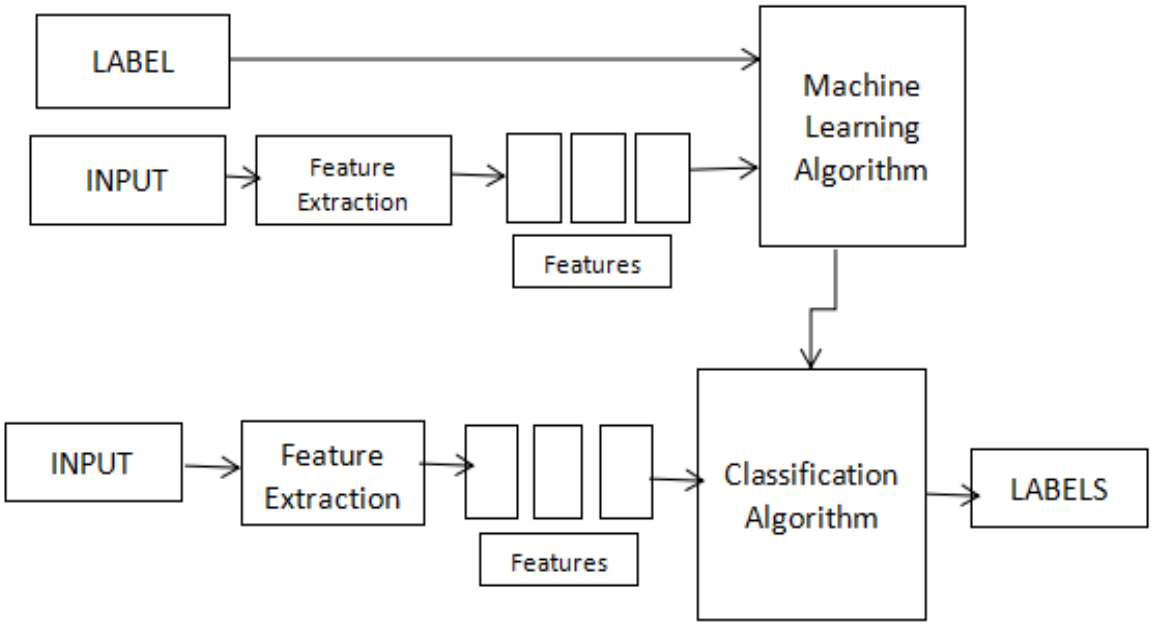


Fig 3.1 Architecture of the system

With the help of the above diagram we can explain our system and the way that the tool is going to work. The first part is: **Training.** Training of the model is also defined as a process where it "learns" from the data. Here both, labels and input are fed into the model on which later the algorithm is applied and part 2 of the model begins its working, this is called **Classification**. Here, after the model has learned from the prior data, it takes the new incoming data and predicts it into different classes on the basis of the training set. These two parts are the essential process of Machine Learning.

Chapter 4. Implementation

4.1 Platform/ Dataset Used

For this thesis the dataset that we have worked on is the data from Twitter Database. Using machine learning on this data is helpful as:

- There is huge data that can be used for the training purpose in the learning,
- And, the data that is available for testing is user-labeled according to the sentiments.

Due to the fixed amount of character-limit on Twitter, its users tend to use short-hand language and slangs, also expressing themselves with the help of emoticons. In a study done by Read in 2005, he said that, expression of sentiments on Twitter along with the use emoticons is helpful in the training of the data.

We first extract the Twitter data using the Twitter API, since we wanted to have a rich user-content to the analysis on, we extracted the tweets on a regular basis, also because of the recent constraint of Twitter allowing only to extract data of the last 7 days.

4.2 Platform: RStudio

R is considered as an integrated environment and provides a suite for various statistical analysis and graphical techniques and representation. R is an efficient language and since its an open source platform, it is easily available and is a free software. It can be compiled and can be run on a variety of platforms like Windows, UNIX and MacOS.

The biggest advantage of using this tool is its ease with which the user can use the in-built functions and the simpleness in the representations of results. Also, it as proven to be very effective in the handling and storage of the data, and there exists integrated tools in R for the data analysis.

4.3 Types of Approaches

The three different machine learning algorithms that were applied on the data are explained below:

4.3.1 Naive Bayes Classifier

This type of technique has a probabilistic view towards the learning problems and is based on the prominent Bayesian theorem. The Naive Bayes classifier is also one of the most frequently used techniques to classify text documents despite its fundamental comprehension. Here, the asset of the classifier is that it is robust to noisy data, also it calculates precise probabilities for the assumed conjectures. The expression for conditional probability that the classifier depends on is given by:

$$P(X|y_j) = \Pi_{i=1}^{m} P(x_i|y_j)$$

Here, X -> is a feature vector where it can be a function of x1, x2…xn

yj -> is a class label

Since Naive Bayes doesn't depend on connection between components and relies on individual assumptions between them, it can be successfully used on the large dataset that we extract from Twitter.

4.3.2) Support Vector Machines

This technique is a part of the linear classifiers in supervised learning and is based on the concept of finding a hyper-plane, that divides the data into two distinct classes, conveniently into positive and negative. It is generally considered one of the best classifiers for text classification. The major benefit of SVM lies in the fact that it can be used to perform non-linear classification as well, using the 'kernel-strategy' approach. The function is SVM can be expressed as:

$$g(X) = w^T \phi(X) + b$$

Here, the different terms used are-

X -> which is a feature vector

w -> which is the vector of weights, and

b -> which is the bias vector

where weight vector(w) and bias vector(b) are automatically found on the dataset that is being used for training and the 'kernel-strategy' that we mentioned above helps to widen the gap between the two classes.

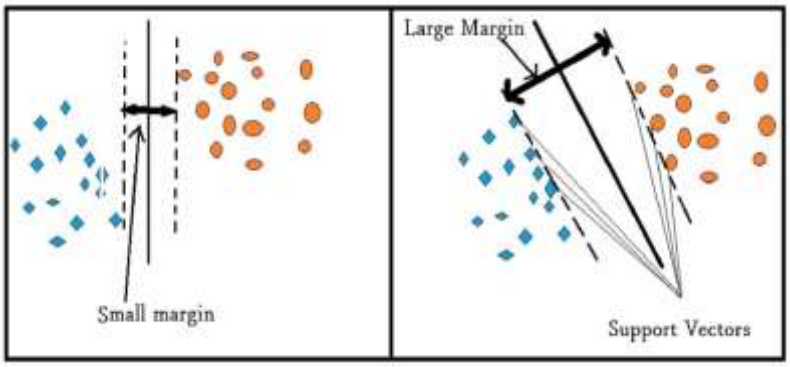


Fig 4.1 SVM Margin

4.3.3) Logistic Regression Approach

This is an statistical type of approach which analyses the data and determines the outcome on the basis of different types of independent variables. Here the measuring is done with the help of a dichotomous variable, which basically means that there would only be two possible outcomes. The expression to fit the regression curve in the data is:

$$y = f(x)$$

Here, y is the variable and x is the set of predictors, where the variables y can be categorical or continuous or both.

4.4) Implementation of the project

The implementation of the model/tool can be explained as follows:

1) Around 1,27,001 tweets were extracted related to IPL, where there existed duplicate tweets and re-tweets or noisy data.

2) After dividing the tweets according to individual teams, we do the pre -processing or the text processing, which is basically the removal of punctuation marks, stopwords, extra spaces, etc. This is the “cleaning” process of the data which left us with 48,615 tweets.

3) Then, the emotions of the tweets are also calculated, like: happy, sad, fear etc. But its only for the purpose of categorizing them into different classes, so it has not been displayed in the thesis.

4) After that, the polarity of the sentiments into positive, negative and neutral are assigned to the tweets.

5) This is followed by a graph representation in which it depicts the number of +ve, -ve and neutral tweets for the 8 individual teams.

6) The file is saved in .csv format, because its easier to read these files on the R tool.
Below are explained some expressions used in the code to help understand the main process of analysis:
Data3 <--- Data3[-1] : These kinds of statements are used to remove an extra row present in the data while converting the text to csv.
KKR <--- prop.table(table(data2$polarity) : This line finds out the proportion of the polarity(positive, negative and neutral) for all the tweets.
Big_data<--- This expression combines whole data for all the teams.

7) Unique data is used for further analysis, that is why the neutral tweets are removed as well, because doing so would refine the basis of our research. Hence, the rest of the analysis is done on positive and negative tweets only.

8) Then the text processing is done again, where certain terms can be explained as:

Dtml<--- DocumentTermMatrix(corpus1): This expression forms a document term matrix which converts into a matrix containing words.

Freq<--- findFreqTerms(c(dtm1), 10): Its used to find the frequency of occurrence of words, where words whose frequency is more than 10 are filtered.

Then the whole matrix is converted into a dataframe and this data is divided into two sets of training(75%) and testing(25%).

9) After converting all the variables (words) into factors, three different approaches like logistic regression, SVM and Naive Bayes are applied on the data to compare their accuracy and other measures where the classification is done using the Naive Bayes approach.

10) Eventually, occurrence of names of players have been counted and saved in the files to find out the overall, indian, foreign, and the youngest most popular player individually.

11) In the end, there is a graph obtained for the prediction of favorite teams rated from 0 to 1.

CHAPTER 5. RESULTS AND ANALYSIS

This section shows the results that we obtained from the implementation of the code on R.There were 8 different teams playing in the Indian Premier League 2017, namely:

- Delhi Daredevils.
- Royal Challengers Bangalore.
- Gujarat Lions.
- Rising Pune Supergiants.
- Kolkata Knight Riders.
- Sunrisers Hyderabad.
- Kings XI Punjab
- Mumbai Indians

The screenshots provide us with a glimpse of the number of positive, negative and neutral tweets that are given for all the 8 individual teams.

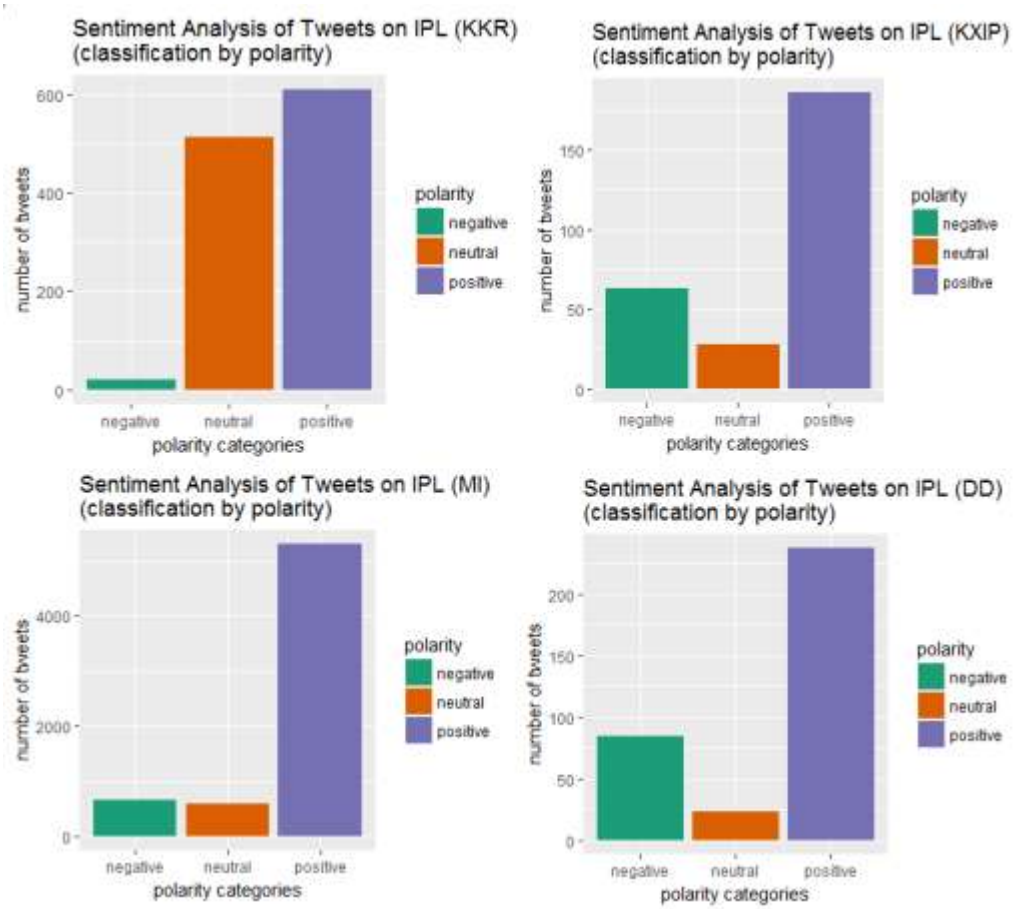


Fig.5.1 Bifurcation of No of Tweets for KKR, KXIP, MI and DD

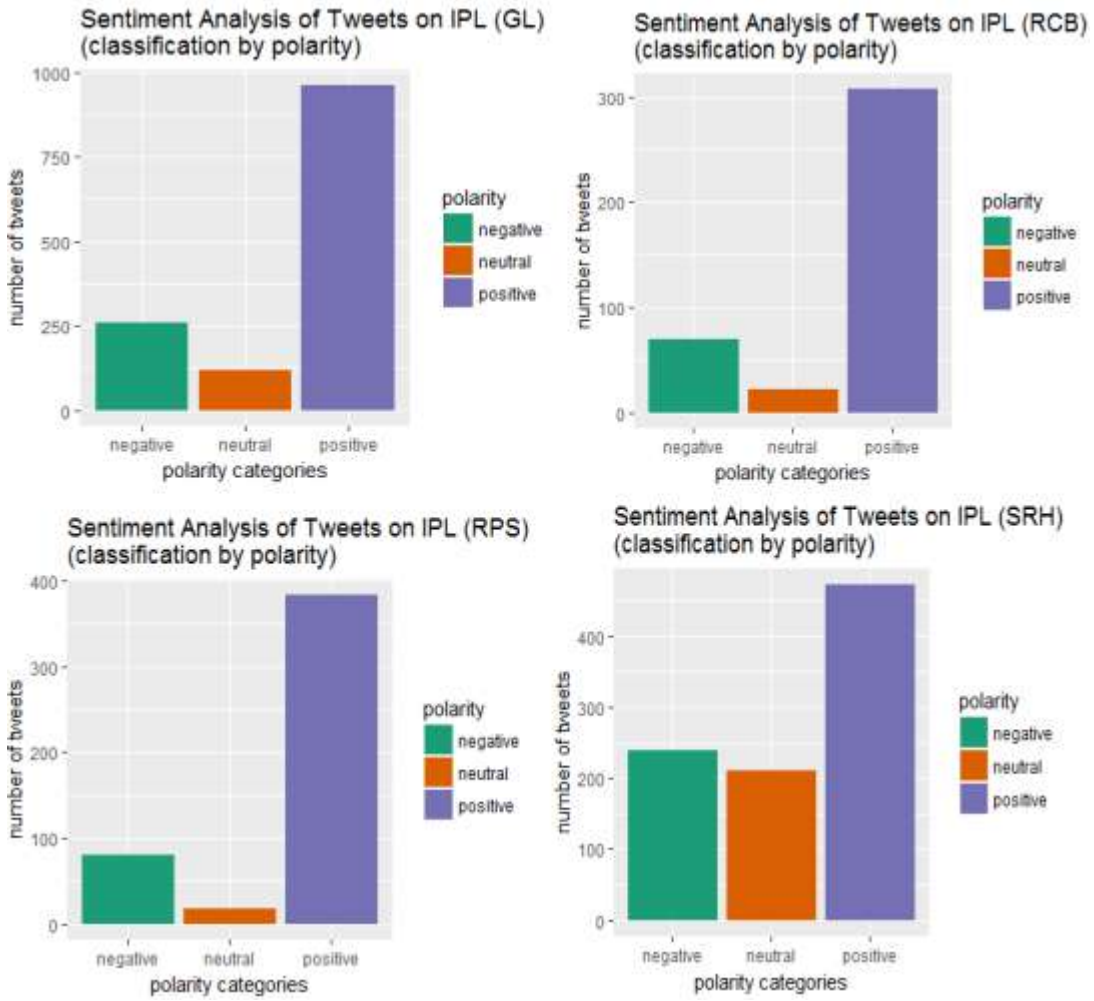
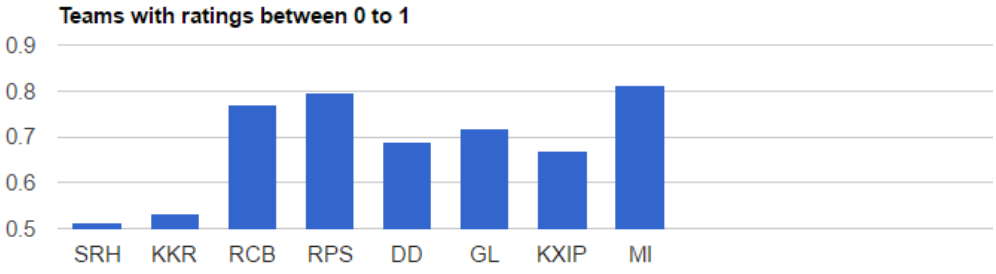


Fig.5.2 Bifurcation of No. of Tweets for GL, RCB, RPS and SRH



Fig() Team-wise rating of different teams.

The above graph shows the prediction of the most favorable team according to the tweets that we have extracted from Twitter, their ranking is given on the basis of ratings ranging from 0 to 1.

The output of different favorable and popular players is given below:

1) Overall Popular Player

	A	B
1	X1	X2
2	Angelo Mathews	0
3	Corey Anderson	1
4	Kagiso Rabada	18
5	Pat Cummins	0
6	Sam Billings	0
7	Quinton de Kock	2
8	Chris Morris	2
9	Carlos Brathwaite	12
10	Zaheer Khan	1
11	Sanju Samson	1
12	Karun Nair	2
13	Mohammed Shami	0
14	Shahbaz Nadeem	0
15	Suresh Raina	1
16	Ravindra Jadeja	1
17	Dinesh Karthik	4
18	Andrew Tye	0
19	Jason Roy	0
20	James Faulkner	0
21	Brendon McCullum	1
22	Dwayne Bravo	0
23	Aaron Finch	0
24	Dwayne Smith	1
25	Eoin Morgan	0
26	Hashim Amla	0
27	Martin Guptill	0
28	David Miller	0
29	Glenn Maxwell	1
30	Shaun Marsh	1
31	Manan Vohra	0
32	Axar Patel	0
33	Sandeep Sharma	0
34	Wriddhiman Saha	2
35	Mohit Sharma	0
36	Trent Boult	0
37	Chris Woakes	0
38	Shakib Al Hasan	0
39	Chris Lynn	0
40	Nathan Coulter-Nile	0
41	Sunil Narine	1
42	Gautam Gambhir	5
43	Manish Pandey	0

	A	B
43	Manish Pandey	0
44	Robin Uthappa	0
45	Umesh Yadav	5
46	Yusuf Pathan	0
47	Mitchell Johnson	10
48	Lendl Simmons	2
49	Mitchell McClenaghan	1
50	Kieron Pollard	6
51	Lasith Malinga	3
52	Jos Buttler	9
53	Tim Southee	0
54	Parthiv Patel	17
55	Harbhajan Singh	3
56	Rohit Sharma	63
57	Jasprit Bumrah	8
58	Hardik Pandya	11
59	Krunal Pandya	32
60	Ben Stokes	11
61	Steven Smith	0
62	Adam Zampa	0
63	Dan Christian	1
64	Manoj Tiwary	2
65	MS Dhoni	23
66	Ajinkya Rahane	14
67	Jaydev Unadkat	2
68	Tymal Mills	0
69	Shane Watson	0
70	Samuel Badree	0
71	Travis Head	0
72	AB de Villiers	0
73	Chris Gayle	0
74	Yuzvendra Chahal	0
75	Kedar Jadhav	0
76	Pawan Negi	23
77	Virat Kohli	26
78	Rashid Khan	3
79	David Warner	12
80	Kane Williamson	11
81	Moises Henriques	0
82	Shikhar Dhawan	12
83	Bhuvneshwar Kumar	5
84	Yuvraj Singh	2

Fig 5.3 Overall Popular Player

2) Youngest Popular Player

	A	B
1	X1	X2
2	Shreyas Iyer	2
3	Rishabh Pant	2
4	Nathu Singh	0
5	Basil Thampi	0
6	Ishan Kishan	0
7	Rahul Tewatia	0
8	T Natarajan	0
9	Shardul Thakur	0
10	KC Cariappa	0
11	Kuldeep Yadav	0
12	Ankit Singh Rajput	0
13	Vijay Shankar	0
14	Deepak Hooda	1
15	Mohammed Siraj	7
16	Sachin Baby	0
17	Avesh Khan	1
18	Aniket Choudhary	0
19	Harshal Patel	1
20	Rahul Tripathi	1
21	washington sundar	0
22	Nitish Rana	4

Fig 5.4 Youngest Player

3) Foreigner Popular Player

	A	B
2	Corey Anderson	1
3	Kagiso Rabada	18
4	Pat Cummins	0
5	Sam Billings	0
6	Chris Morris	2
7	Andrew Tye	0
8	Jason Roy	0
9	James Faulkner	0
10	Brendon McCullum	1
11	Aaron Finch	0
12	Dwayne Smith	1
13	Eoin Morgan	0
14	Hashim Amla	0
15	Martin Guptill	0
16	David Miller	0
17	Glenn Maxwell	1
18	Shaun Marsh	1
19	Trent Boult	0
20	Chris Woakes	0
21	Chris Lynn	0
22	Nathan Coulter-Nile	0
23	Sunil Narine	1
24	Mitchell Johnson	10
25	Lendl Simmons	2
26	Mitchell McClenaghan	1
27	Kieron Pollard	6
28	Lasith Malinga	3
29	Jos Buttler	9
30	Tim Southee	0
31	Ben Stokes	11
32	Steven Smith	0
33	Adam Zampa	0
34	Dan Christian	1
35	Rashid Khan	3
36	David Warner	12
37	Kane Williamson	11
38	Moises Henriques	0
39	Tymal Mills	0
40	Shane Watson	0
41	Samuel Badree	0
42	Travis Head	0
43	AB de Villiers	0
44	Chris Gayle	0

Fig 5.5 Foreigner Player

4) Indian Popular Player

	A	B
1	X1	X2
2	Zaheer Khan	1
3	Sanju Samson	1
4	Karun Nair	2
5	Mohammed Shami	0
6	Shahbaz Nadeem	0
7	Shikhar Dhawan	12
8	Bhuvneshwar Kumar	5
9	Yuvraj Singh	2
10	Yuzvendra Chahal	0
11	Kedar Jadhav	0
12	Pawan Negi	23
13	Virat Kohli	26
14	Manoj Tiwary	2
15	MS Dhoni	23
16	Ajinkya Rahane	14
17	Jaydev Unadkat	2
18	Parthiv Patel	17
19	Harbhajan Singh	3
20	Rohit Sharma	63
21	Jasprit Bumrah	8
22	Hardik Pandya	11
23	Krunal Pandya	32
24	Gautam Gambhir	5
25	Manish Pandey	0
26	Robin Uthappa	0
27	Umesh Yadav	5
28	Yusuf Pathan	0
29	Manan Vohra	0
30	Axar Patel	0
31	Sandeep Sharma	0
32	Wriddhiman Saha	2
33	Mohit Sharma	0
34	Suresh Raina	1
35	Ravindra Jadeja	1
36	Dinesh Karthik	4

In the next part, the results show the confusion matrix, the accuracy and other measuring parameters of the three different learning techniques, that are: Naive Bayes, SVM and Logistic Regression.

1) Naive Bayes

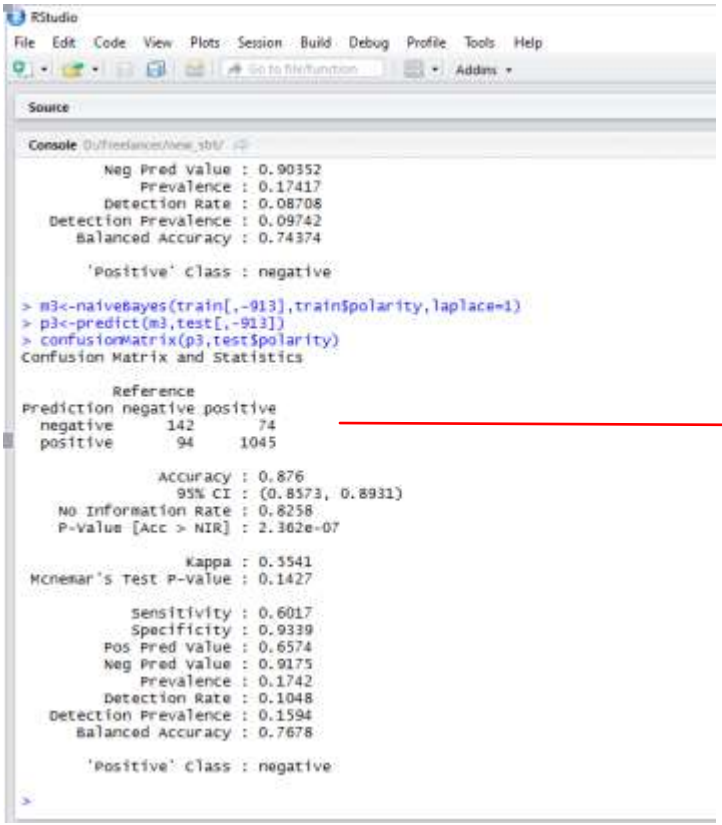


Fig 5.6 Naive Bayes Output

2) Support Vector Machine

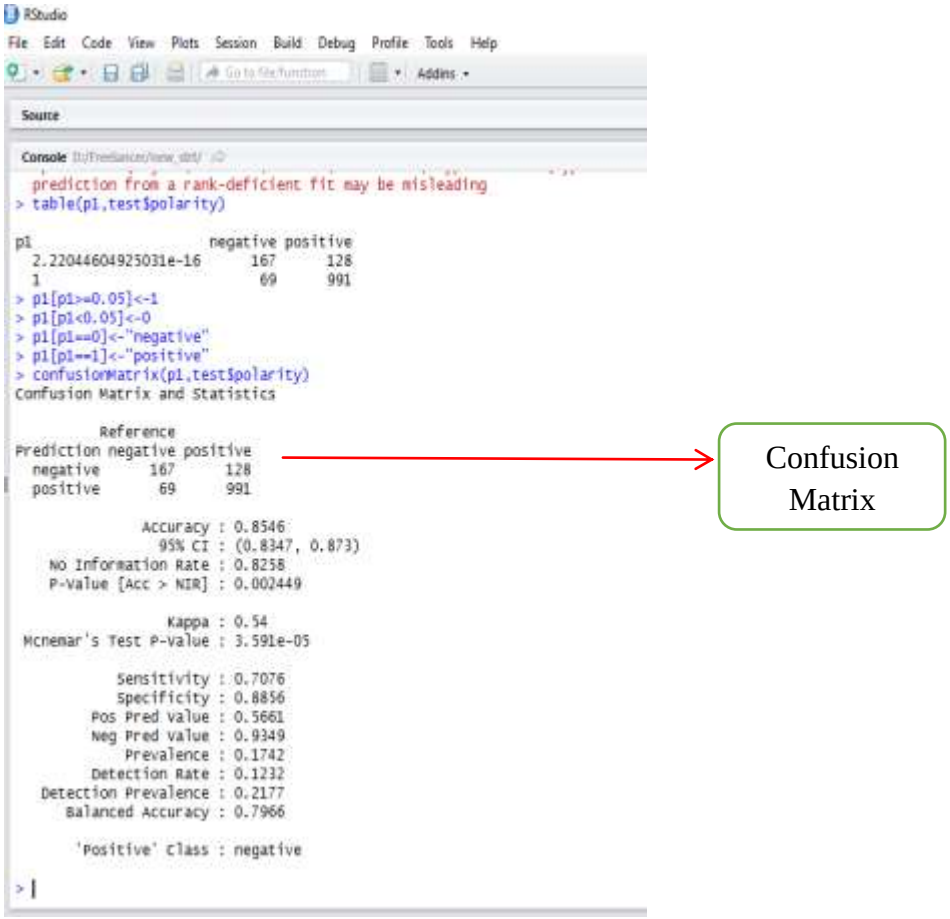


Fig 5.7 SVM Output

3) Logistic Regression

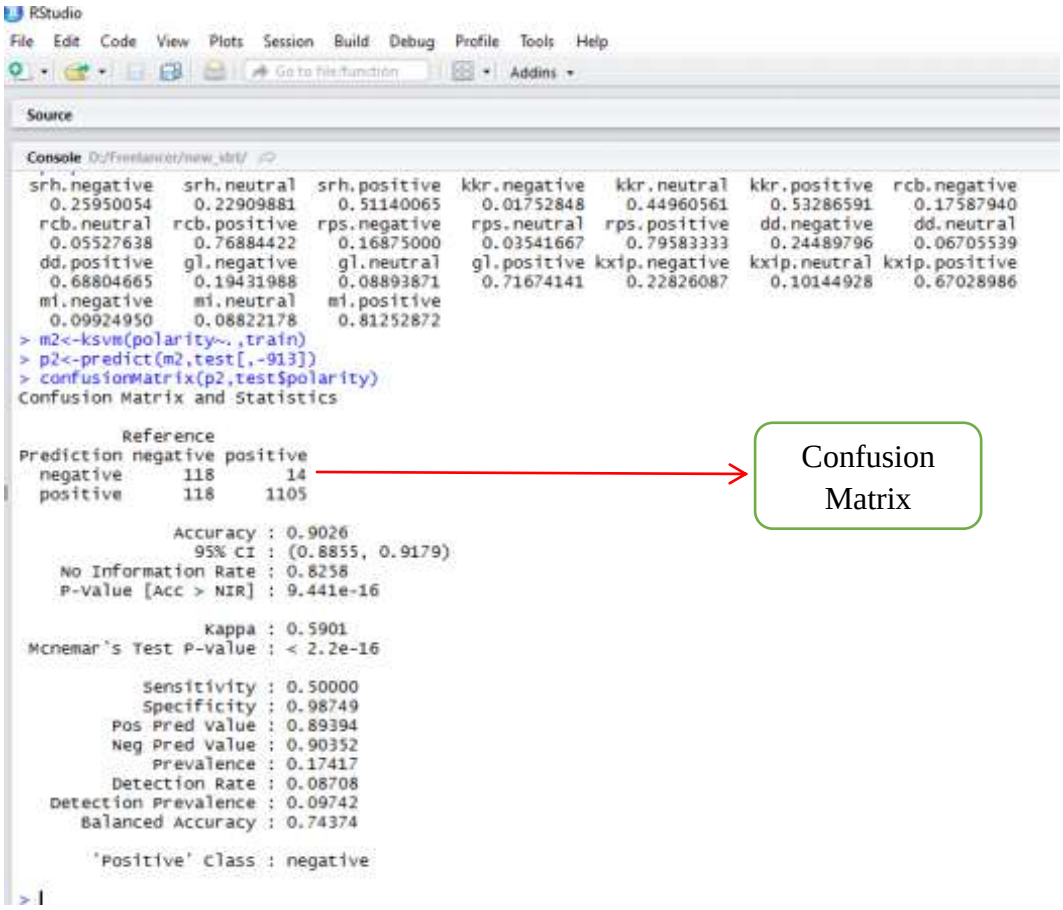


Fig 5.8 Logistic Regression Output

Chapter 6. Conclusion

6.1 Conclusion

With the presentation of this thesis, we come to the conclusion that sentiment analysis can prove to be a boon in the field of business intelligence. Here, according to the representations of our results from our experimentation in the previous section and considering to give this data to a brand endorsment company at the user-end, we can infer that the firm would like to affiliate itself to either of the trending teams that were ranked on the basis of ratings of 0-1.

The ranking of the teams is :

1) Mumbai Indians
2) Rising Pune Supergiants
3) Royal Challengers Bangalore
4) Gujarat Lions
5) Delhi Daredevils
6) Kings XI Punjab
7) Kolkata Night Riders
8) Sun Risers Hyderabad

We have also calculated the amount of tweets and the polarity associated with them, to help understand the positive or negative trend of the team. This is represented by the bar graphs in the former section.

We use the Naive Bayes classifier for the classification of the tweets into the different categories. Then we apply the different machine learning algorithms to learn the text patterns and then show prediction by this learning, in the testing dataset.
The accuracy of those different approaches were found and compared to detect which classifier can be a better predictor in the model.

Classifier	Regression Model	SVM	Naive Bayes
Accuracy	85.46%	90.26%	87.6%

So, according to the survey done on this data, SVM is the best classifier that can be used for prediction.

6.2) Limitations

According to our research done using Twitter for the analysis of the sports event, we found out that there are certain constraints associated with it.

1) Initially, we could extract tweets using the Twitter API by using the keyword on which we wanted to do the analysis on, but now twitter has limited the access of data for only past 7 days. This **limits the user's extraction of content**, hence one needs to carry out this process over and over again to get a sentiment-rich corpus.

2) **Trends are dynamic.** The fluctuating data effects the trends that we analyse, making it difficult to infer a strong decision, specially in the application domain of business intelligence, where one wrong decision of affiliation may lead to huge losses.

3) Different **outliers** involved in the data. Since the limit of tweets is just 140 characters, it is both an advantage and disadvantage for the user. One can express himself in limited words but also, it might be difficult sometimes for people who are

not good in communicating well with words. Use of slangs and abbreviations and short hand typing makes it difficult to analyse the sentiments of the tweets.

4) In this particular project, another limitation that we came across was the wavering of rankings that may occur according to the performances of the players and that can differ the predictions of the results.

6.3) Future Work

Considering the conclusions and limitations of our project, we derive that a lot of work using supervised techniques have been done in the past. In this thesis as well, we use 3 different supervised learning techniques. For further exploration and study in this area, we suggest future researchers the following scope of work:

1) Build a hybrid learning technique for better prediction results. In this thesis, even though we have used different technique for classification and various ensemble techniques for prediction, there is still scope to improve the accuracy of the predictors using hybrid models.

2) One can further conduct survey on this event with the help of already available data on the web, ie. The rankings of the teams that are provided by different newspapers and online sports site can be compared with our rankings, and a co-relation between the two can be found out, to contemplate the accuracy of our results with the real-time results.

REFERENCES

[1] L. Carstens, "Sentiment analysis – a multinational approach", Imperial College London, September 2011.

[2] B. Pang and L.Lee, "Opinion mining and sentiment analysis",Foundations and Trends in Information Retrieval, Vol. 2, No 1-2, pp. 1–135, 2008

[3] J. Khairnar and M. Kinikar, "Machine learning algorithms for opinion mining and sentiment classification", International Journal of Scientific and Research Publications, Vol. 3, pp. 1-6, 2013.

[4] C. Mosha, "Combining Dependency Parsing with Shallow Semantic Analysis for

Chinese Opinion-Element Relation Identification", IEEE, pp. 299-305, 2010

[5] S. Choudhury and J. G. Breslin, "User sentiment detection: a YouTube use case", Proceedings of the 21st National Conference on Artificial Intelligence and Cognitive Science, 2010

[6] P.Songram and C.Jareanpon, "Opinion Mining of Thai Politics on Facebook Status Updates", Proceedings of the 3rd IIAE International Conference on Intelligent Systems and Image Processing, 2015

[7]http://www.dcc.ufrj.br/~valeriab/DTM-supervised-learning_altJonice.pdf, 12/12/2016

[8] P. Schultes, V. Dorner and F. Lehner, "Leave a comment! An in-depth analysis of user comments on YouTube", 11th International Conference on Wirtschaftsinformatik, pp. 659-673, 2013.

[9]X.Zhu and A.B.Goldberg,"Introduction to Semi-Supervised Learning", Morgan & Claypool, 2009

[10] M. Z. Asghar, S. Ahmed, A. Marwat and F.M. Kundi, "Sentiment analysis on youtube: A brief survey", pp. 1-6, 2015

[11] A.Krishnay, J. Zambreno and S. Krishnan, "Polarity Trend Analysis of Public Sentiment on YouTube", Proceedings of the 19th International Conference on Management of Data (COMAD), pp. 125-128,2013

[12] D. Daviov, O. Tsur and A. Rappoport, "Enhanced sentiment learning using twitter hashtags and smileys", Proceedings of the 23rd International Conference on Computational Linguistics: Posters, COLING'10, pp. 241-249, 2010

[13] L. Barbosa and J. Feng, "Robust sentiment detection on twitter from biased and noisy data", Proceedings of the 23rd International Conference on Computational Linguistics: Posters, Coling'10, pp. 36-44, 2010

[14] A. Kumar, and T. M. Sebastian, "Sentiment analysis on twitter", International Journal of Computer Science Issues, vol. 9, issue 4, pp. 372-378, 2012.

[15] N. A. M. Zamini, S. Z.Z. Abidin, N. Omar and M.Z.Z. Zabiden, "Sentiment analysis: determining people's emotions in facebook", Applied Computational Science, pp. 111-116, 2014

[16] G. Vashisht and S. Thakur, "Facebook as a corpus for emoticons-based sentiment analysis", International Journal of Engineering and Advanced Engineering, vol. 4, issue. 5, pp. 904-908, 2014.

[17] N. Nigam, K. Bansal, R. Sharma, R. K. Trivedi, “Opinion mining of online shopping sites through facebook pages using graph API and FQL query”, International Journal of Advanced Research In Science and Engineering, vol. 4, issue. 2, pp. 64-68, 2015.

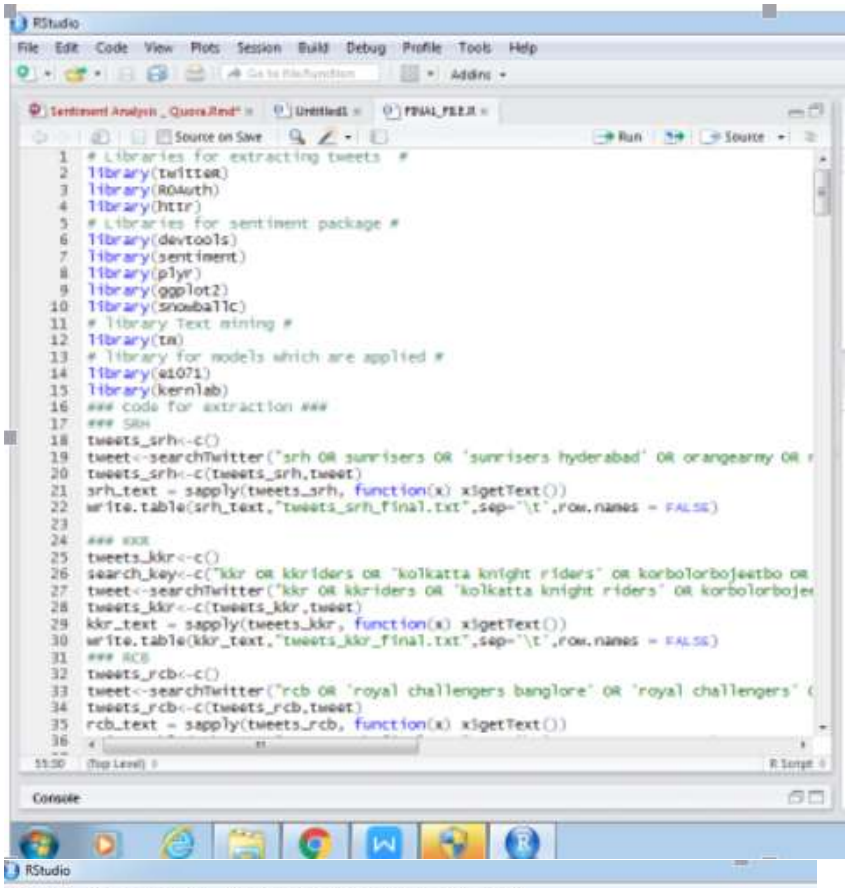
[18] C. Müller and I.Gurevych, “Using Wikipedia and Wiktionary in domain-specific information retrieval”, Proceeding CLEF' 08, Springer-Verlag Berlin, Heidelberg, 2009

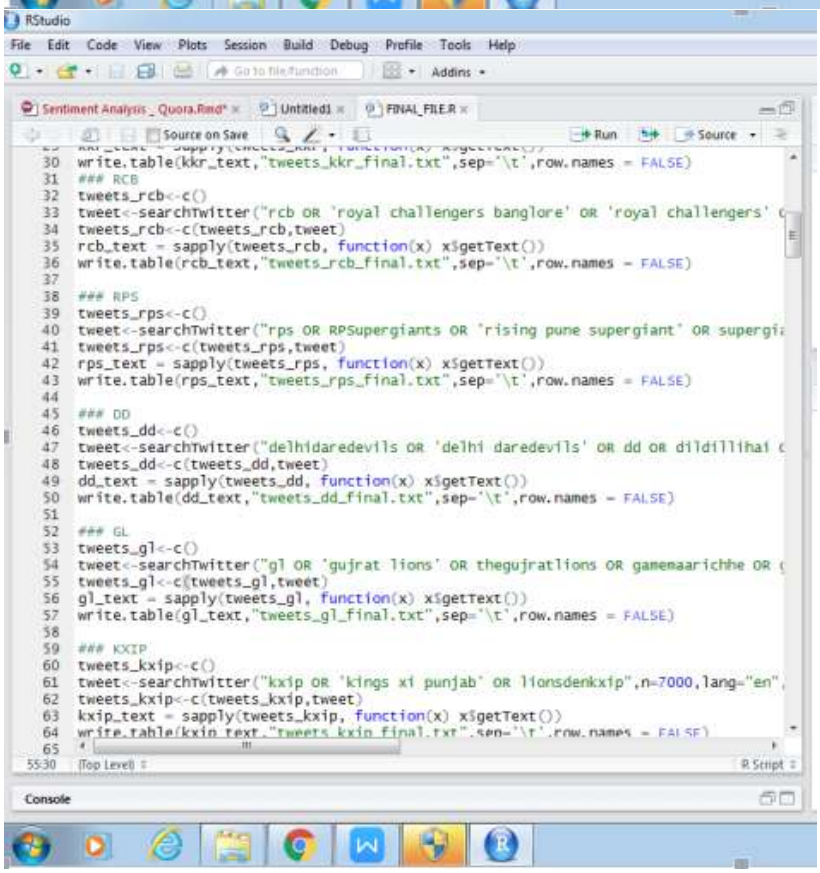
[19] J.M Conroy and D. P. O'leary, “Text summarization via hidden markov models”, In Proceedings of SIGIR '01, pages 406-407, 2001

[20] B. Cui, C. Zhang and G. Cong, “Content-enriched classifier for web video classification”, Proceedings of the 33rd international ACM SIGIR, 2010

[21] DM. Bikel and J. Sorensen, “If We Want Your Opinion”, International Conference on Semantic Computing (ICSC 2007), 2007

APPENDIX A


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Addins
Sentiment Analysis _ Quora.Rmd* Untitled1 FINAL_FILE.R
Source on Save
Run
Source
# Libraries for extracting tweets #
library(twitteR)
library(ROAuth)
library(httr)
# Libraries for sentiment package #
library(devtools)
library(sentiment)
library(plyr)
library(ggplot2)
library(snowballc)
# library Text mining #
library(tm)
# library for models which are applied #
library(e1071)
library(kernlab)
code for extraction
SRH
tweets_srh<-c()
tweet<-searchTwitter("srh OR sunrisers OR 'sunrisers hyderabad' OR orangearmy OR
tweets_srh<-c(tweets_srh,tweet)
srh_text = sapply(tweets_srh, function(x) x$getText())
write.table(srh_text,"tweets_srh_final.txt",sep="\t",row.names = FALSE)
KKR
tweets_kkr<-c()
search_key<-c("kkr OR kkriders OR "kolkatta knight riders" OR korbolorbojeetbo OR
tweet<-searchTwitter("kkr OR kkriders OR 'kolkatta knight riders' OR korbolorboje
tweets_kkr<-c(tweets_kkr,tweet)
kkr_text = sapply(tweets_kkr, function(x) x$getText())
write.table(kkr_text,"tweets_kkr_final.txt",sep="\t",row.names = FALSE)
RCB
tweets_rcb<-c()
tweet<-searchTwitter("rcb OR 'royal challengers banglore' OR 'royal challengers'
tweets_rcb<-c(tweets_rcb,tweet)
rcb_text = sapply(tweets_rcb, function(x) x$getText())
(Top Level)
R Script
Console


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Addins
Sentiment Analysis _ Quora.Rmd* Untitled1 FINAL_FILE.R
Source on Save
Run
Source
write.table(kkr_text,"tweets_kkr_final.txt",sep="\t",row.names = FALSE)
RCB
tweets_rcb<-c()
tweet<-searchTwitter("rcb OR 'royal challengers banglore' OR 'royal challengers'
tweets_rcb<-c(tweets_rcb,tweet)
rcb_text = sapply(tweets_rcb, function(x) x$getText())
write.table(rcb_text,"tweets_rcb_final.txt",sep="\t",row.names = FALSE)
RPS
tweets_rps<-c()
tweet<-searchTwitter("rps OR RPSupergiants OR 'rising pune supergiant' OR supergi
tweets_rps<-c(tweets_rps,tweet)
rps_text = sapply(tweets_rps, function(x) x$getText())
write.table(rps_text,"tweets_rps_final.txt",sep="\t",row.names = FALSE)
DD
tweets_dd<-c()
tweet<-searchTwitter("delhidaredevils OR 'delhi daredevils' OR dd OR dildillihai
tweets_dd<-c(tweets_dd,tweet)
dd_text = sapply(tweets_dd, function(x) x$getText())
write.table(dd_text,"tweets_dd_final.txt",sep="\t",row.names = FALSE)
GL
tweets_gl<-c()
tweet<-searchTwitter("gl OR 'gujrat lions' OR thegujratlions OR gamemaarichhe OR
tweets_gl<-c(tweets_gl,tweet)
gl_text = sapply(tweets_gl, function(x) x$getText())
write.table(gl_text,"tweets_gl_final.txt",sep="\t",row.names = FALSE)
KXIP
tweets_kxip<-c()
tweet<-searchTwitter("kxip OR 'kings xi punjab' OR lionsdenkxip",n=7000,lang="en"
tweets_kxip<-c(tweets_kxip,tweet)
kxip_text = sapply(tweets_kxip, function(x) x$getText())
write.table(kxip_text,"tweets_kxip_final.txt",sep="\t",row.names = FALSE)
(Top Level)
R Script
Console

```
# Libraries for extracting tweets    #

library(twitteR)

library(ROAuth)

library(httr)

# Libraries for sentiment package #

library(devtools)

library(sentiment)

library(plyr)

library(ggplot2)

library(SnowballC)

# library Text mining #

library(tm)

# library for models which are applied #

library(e1071)

library(kernlab)

### Code for extraction ###

### SRH

tweets_srh<-c()
```

```
tweet<-searchTwitter("srh OR sunrisers OR 'sunrisers hyderabad' OR orangearmy OR riseoforange",n=3000,lang="en",since='2017-05-31',until = '2017-06-01')

tweets_srh<-c(tweets_srh,tweet)

srh_text = sapply(tweets_srh, function(x) x$getText())

write.table(srh_text,"tweets_srh_final.txt",sep='\t',row.names = FALSE)


### KKR

tweets_kkr<-c()

search_key<-c("kkr OR kkriders OR 'kolkatta knight riders' OR korbolorbojeetbo OR amikkr")

tweet<-searchTwitter("kkr OR kkriders OR 'kolkatta knight riders' OR korbolorbojeetbo OR amikkr",n=4000,lang="en",since='2017-05-31',until='2017-06-01')

tweets_kkr<-c(tweets_kkr,tweet)

kkr_text = sapply(tweets_kkr, function(x) x$getText())

write.table(kkr_text,"tweets_kkr_final.txt",sep='\t',row.names = FALSE)

### RCB

tweets_rcb<-c()

tweet<-searchTwitter("rcb OR 'royal challengers banglore' OR 'royal challengers' OR rcbtweets OR playbold",n=4000,lang="en",since='2017-05-31',until='2017-06-01')
```

```
tweets_rcb<-c(tweets_rcb,tweet)

rcb_text = sapply(tweets_rcb, function(x) x$getText())

write.table(rcb_text,"tweets_rcb_final.txt",sep='\t',row.names = FALSE)


### RPS

tweets_rps<-c()

tweet<-searchTwitter("rps OR RPSupergiants OR 'rising pune supergiant'
OR
supergiants",n=5000,lang="en",since='2017-05-31',until='2017-06-01')

tweets_rps<-c(tweets_rps,tweet)

rps_text = sapply(tweets_rps, function(x) x$getText())

write.table(rps_text,"tweets_rps_final.txt",sep='\t',row.names = FALSE)


### DD

tweets_dd<-c()

tweet<-searchTwitter("delhidaredevils OR 'delhi daredevils' OR dd OR
dildillihai OR
ddtweets",n=5000,lang="en",since='2017-05-31',until='2017-06-01')

tweets_dd<-c(tweets_dd,tweet)

dd_text = sapply(tweets_dd, function(x) x$getText())

write.table(dd_text,"tweets_dd_final.txt",sep='\t',row.names = FALSE)
```

```
### GL

tweets_gl<-c()

tweet<-searchTwitter("gl OR 'gujrat lions' OR thegujratlions OR gamemaarichhe OR gujratlions OR roaroncemore",n=5000,lang="en",since='2017-05-31',until='2017-06-01')

tweets_gl<-c(tweets_gl,tweet)

gl_text = sapply(tweets_gl, function(x) x$getText())

write.table(gl_text,"tweets_gl_final.txt",sep='\t',row.names = FALSE)


### KXIP

tweets_kxip<-c()

tweet<-searchTwitter("kxip OR 'kings xi punjab' OR lionsdenkxip",n=7000,lang="en",since='2017-05-31',until = '2017-06-01')

tweets_kxip<-c(tweets_kxip,tweet)

kxip_text = sapply(tweets_kxip, function(x) x$getText())

write.table(kxip_text,"tweets_kxip_final.txt",sep='\t',row.names = FALSE)


### MI
```

```
tweets_mi<-c()

tweet<-searchTwitter("mi OR 'mumbai indians' OR mipaltan OR cricketmerijaan",n=6000,lang="en",since='2017-05-31',until='2017-06-01')

tweets_mi<-c(tweets_mi,tweet)

mi_text = sapply(tweets_mi, function(x) x$getText())

write.table(mi_text,"tweets_mi_final.txt",sep='\t',row.names = FALSE)


######### EXRACTION COMPLETED AND NOW READING EACH FILES #######

####### READING TEXT FILES

srh_text=read.table("tweets_srh_final.txt",sep='\t')

kkr_text=read.table("tweets_kkr_final.txt",sep='\t')

rps_text=read.table("tweets_rps_final.txt",sep='\t')

rcb_text=read.table("tweets_rcb_final.txt",sep='\t')

dd_text=read.table("tweets_dd_final.txt",sep='\t')

gl_text=read.table("tweets_gl_final.txt",sep='\t')

kxip_text=read.table("tweets_kxip_final.txt",sep='\t')

mi_text=read.table("tweets_mi_final.txt",sep='\t')

####### SRH #######
```

```
srh_text1<-srh_text

srh_text<-apply(srh_text,1,function(x)
gsub("(RT|via)((?:\\b\\W*@\\w+)+)", "",x))

srh_text = gsub("@\\w+", "", srh_text)

srh_text = gsub("[[:punct:]]", "", srh_text )

srh_text = gsub("[[:digit:]]", "", srh_text)

srh_text = gsub("http\\w+", "", srh_text)

srh_text = gsub("[ \t]{2,}", "", srh_text)

srh_text = gsub("^\\s+|\\s+$", "", srh_text)

try.error = function(x)

{

   y = NA

   try_error = tryCatch(tolower(x), error=function(e) e)

   if (!inherits(try_error, "error"))

      y = tolower(x)

   return(y)

}

srh_text = sapply(srh_text, try.error)

srh_text = srh_text [!is.na(srh_text)]

names(srh_text) = NULL
```

```
class_emo = classify_emotion(srh_text, algorithm="bayes", prior=1.0)

emotion = class_emo[,7]

emotion[is.na(emotion)] = "unknown"

class_pol = classify_polarity(srh_text, algorithm="bayes")

polarity = class_pol[,4]

sent_df = data.frame(text= srh_text1, emotion=emotion,polarity=polarity, stringsAsFactors=FALSE)

sent_df = within(sent_df,emotion <- factor(emotion, levels=names(sort(table(emotion), decreasing=TRUE))))

ggplot(sent_df, aes(x=polarity)) +

   geom_bar(aes(y=..count.., fill=polarity)) +

   scale_fill_brewer(palette="Dark2") +

   labs(x="polarity categories", y="number of tweets") +

   labs(title = "Sentiment Analysis of Tweets on IPL (SRH)\n(classification by polarity)")

abc<-sent_df[c(1,3)]

write.csv(abc,"srh_csv_senti.csv",row.names = FALSE)

####### KKR #######

kkr_text1<-kkr_text

kkr_text<-apply(kkr_text,1,function(x)
gsub("(RT|via)((?:\\b\\W*@\\w+)+)", "",x))
```

```
kkr_text = gsub("@\\w+", "", kkr_text)

kkr_text = gsub("[[:punct:]]", "", kkr_text )

kkr_text = gsub("[[:digit:]]", "", kkr_text)

kkr_text = gsub("http\\w+", "", kkr_text)

kkr_text = gsub("[ \t]{2,}", "", kkr_text)

kkr_text = gsub("^\\s+|\\s+$", "", kkr_text)

try.error = function(x)

{

   y = NA

   try_error = tryCatch(tolower(x), error=function(e) e)

   if (!inherits(try_error, "error"))

      y = tolower(x)

   return(y)

}

kkr_text = sapply(kkr_text, try.error)

kkr_text = kkr_text [!is.na(kkr_text)]

names(kkr_text) = NULL

class_emo = classify_emotion(kkr_text, algorithm="bayes", prior=1.0)

emotion = class_emo[,7]

emotion[is.na(emotion)] = "unknown"
```

```
class_pol = classify_polarity(kkr_text, algorithm="bayes")

polarity = class_pol[,4]

sent_df = data.frame(text= kkr_text1, emotion=emotion,polarity=polarity, stringsAsFactors=FALSE)

sent_df = within(sent_df,emotion <- factor(emotion, levels=names(sort(table(emotion), decreasing=TRUE))))

ggplot(sent_df, aes(x=polarity)) +

   geom_bar(aes(y=..count.., fill=polarity)) +

   scale_fill_brewer(palette="Dark2") +

   labs(x="polarity categories", y="number of tweets") +

   labs(title = "Sentiment Analysis of Tweets on IPL (KKR) \n(classification by polarity)")

abc<-sent_df[c(1,3)]

write.csv(abc,"kkr_csv_senti.csv",row.names = FALSE)

####### RCB #######

rcb_text1<-rcb_text

rcb_text<-apply(rcb_text,1,function(x) gsub("(RT|via)((?:\\b\\W*@\\w+)+)", "",x))

rcb_text = gsub("@\\w+", "", rcb_text)

rcb_text = gsub("[[:punct:]]", "", rcb_text )

rcb_text = gsub("[[:digit:]]", "", rcb_text)
```

```
rcb_text = gsub("http\\w+", "", rcb_text)

rcb_text = gsub("[ \t]{2,}", "", rcb_text)

rcb_text = gsub("^\\s+|\\s+$", "", rcb_text)

try.error = function(x)

{

   y = NA

   try_error = tryCatch(tolower(x), error=function(e) e)

   if (!inherits(try_error, "error"))

      y = tolower(x)

   return(y)

}

rcb_text = sapply(rcb_text, try.error)

rcb_text = rcb_text [!is.na(rcb_text)]

names(rcb_text) = NULL

class_emo = classify_emotion(rcb_text, algorithm="bayes", prior=1.0)

emotion = class_emo[,7]

emotion[is.na(emotion)] = "unknown"

class_pol = classify_polarity(rcb_text, algorithm="bayes")

polarity = class_pol[,4]
```

```
sent_df = data.frame(text= rcb_text1, emotion=emotion,polarity=polarity, stringsAsFactors=FALSE)

sent_df = within(sent_df,emotion <- factor(emotion, levels=names(sort(table(emotion), decreasing=TRUE))))

ggplot(sent_df, aes(x=polarity)) +

   geom_bar(aes(y=..count.., fill=polarity)) +

   scale_fill_brewer(palette="Dark2") +

   labs(x="polarity categories", y="number of tweets") +

   labs(title = "Sentiment Analysis of Tweets on IPL (RCB) \n(classification by polarity)")

abc<-sent_df[c(1,3)]

write.csv(abc,"rcb_csv_senti.csv",row.names = FALSE)

######## RPS ########

rps_text1<-rps_text

rps_text<-apply(rps_text,1,function(x) gsub("(RT|via)((?:\\b\\W*@\\w+)+)", "",x))

rps_text = gsub("@\\w+", "", rps_text)

rps_text = gsub("[[:punct:]]", "", rps_text )

rps_text = gsub("[[:digit:]]", "", rps_text)

rps_text = gsub("http\\w+", "", rps_text)

rps_text = gsub("[ \t]{2,}", "", rps_text)
```

```
rps_text = gsub("^\\s+|\\s+$", "", rps_text)

try.error = function(x)

{

   y = NA

   try_error = tryCatch(tolower(x), error=function(e) e)

   if (!inherits(try_error, "error"))

      y = tolower(x)

   return(y)

}

rps_text = sapply(rps_text, try.error)

rps_text = rps_text [!is.na(rps_text)]

names(rps_text) = NULL

class_emo = classify_emotion(rps_text, algorithm="bayes", prior=1.0)

emotion = class_emo[,7]

emotion[is.na(emotion)] = "unknown"

class_pol = classify_polarity(rps_text, algorithm="bayes")

polarity = class_pol[,4]

sent_df = data.frame(text= rps_text1,

emotion=emotion,polarity=polarity, stringsAsFactors=FALSE)
```

```
sent_df = within(sent_df,emotion <- factor(emotion, levels=names(sort(table(emotion), decreasing=TRUE))))

ggplot(sent_df, aes(x=polarity)) +

   geom_bar(aes(y=..count.., fill=polarity)) +

   scale_fill_brewer(palette="Dark2") +

   labs(x="polarity categories", y="number of tweets") +

   labs(title = "Sentiment Analysis of Tweets on IPL (RPS)\n(classification by polarity)")

abc<-sent_df[c(1,3)]

write.csv(abc,"rps_csv_senti.csv",row.names = FALSE)

######### DD ########

dd_text1<-dd_text

dd_text<-apply(dd_text,1,function(x) gsub("(RT|via)((?:\\b\\W*@\\w+)+)", "",x))

dd_text = gsub("@\\w+", "", dd_text)

dd_text = gsub("[[:punct:]]", "", dd_text )

dd_text = gsub("[[:digit:]]", "", dd_text)

dd_text = gsub("http\\w+", "", dd_text)

dd_text = gsub("[ \t]{2,}", "", dd_text)

dd_text = gsub("^\\s+|\\s+$", "", dd_text)
```

```
try.error = function(x)
{
   y = NA
   try_error = tryCatch(tolower(x), error=function(e) e)
   if (!inherits(try_error, "error"))
      y = tolower(x)
   return(y)
}
dd_text = sapply(dd_text, try.error)
dd_text = dd_text [!is.na(dd_text)]
names(dd_text) = NULL
class_emo = classify_emotion(dd_text, algorithm="bayes", prior=1.0)
emotion = class_emo[,7]
emotion[is.na(emotion)] = "unknown"
class_pol = classify_polarity(dd_text, algorithm="bayes")
polarity = class_pol[,4]
sent_df = data.frame(text= dd_text1, emotion=emotion,polarity=polarity, stringsAsFactors=FALSE)
sent_df = within(sent_df,emotion <- factor(emotion, levels=names(sort(table(emotion), decreasing=TRUE))))
```

```
ggplot(sent_df, aes(x=polarity)) +
   geom_bar(aes(y=..count.., fill=polarity)) +
   scale_fill_brewer(palette="Dark2") +
   labs(x="polarity categories", y="number of tweets") +
   labs(title = "Sentiment Analysis of Tweets on IPL (DD)\n(classification by polarity)")
abc<-sent_df[c(1,3)]
write.csv(abc,"dd_csv_senti.csv",row.names = FALSE)
########## GL ##########
gl_text1<-gl_text
gl_text<-apply(gl_text,1,function(x) gsub("(RT|via)((?:\\b\\W*@\\w+)+)",
"",x))
gl_text = gsub("@\\w+", "", gl_text)
gl_text = gsub("[[:punct:]]", "", gl_text )
gl_text = gsub("[[:digit:]]", "", gl_text)
gl_text = gsub("http\\w+", "", gl_text)
gl_text = gsub("[ \t]{2,}", "", gl_text)
gl_text = gsub("^\\s+|\\s+$", "", gl_text)
try.error = function(x)
{
```

```
    y = NA
    try_error = tryCatch(tolower(x), error=function(e) e)
    if (!inherits(try_error, "error"))
        y = tolower(x)
    return(y)
}
gl_text = sapply(gl_text, try.error)
gl_text = gl_text [!is.na(gl_text)]
names(gl_text) = NULL
class_emo = classify_emotion(gl_text, algorithm="bayes", prior=1.0)
emotion = class_emo[,7]
emotion[is.na(emotion)] = "unknown"
class_pol = classify_polarity(gl_text, algorithm="bayes")
polarity = class_pol[,4]
sent_df = data.frame(text= gl_text1, emotion=emotion,polarity=polarity,
stringsAsFactors=FALSE)
sent_df = within(sent_df,emotion <- factor(emotion,
levels=names(sort(table(emotion), decreasing=TRUE))))
ggplot(sent_df, aes(x=polarity)) +
    geom_bar(aes(y=..count.., fill=polarity)) +
```

```
   scale_fill_brewer(palette="Dark2") +

   labs(x="polarity categories", y="number of tweets") +

   labs(title = "Sentiment Analysis of Tweets on IPL (GL)\n(classification
by polarity)")

abc<-sent_df[c(1,3)]

write.csv(abc,"gl_csv_senti.csv",row.names = FALSE)

######## KXIP #######

kxip_text1<-kxip_text

kxip_text<-apply(kxip_text,1,function(x)
gsub("(RT|via)((?:\\b\\W*@\\w+)+)", "",x))

kxip_text = gsub("@\\w+", "", kxip_text)

kxip_text = gsub("[[:punct:]]", "", kxip_text )

kxip_text = gsub("[[:digit:]]", "", kxip_text)

kxip_text = gsub("http\\w+", "", kxip_text)

kxip_text = gsub("[ \t]{2,}", "", kxip_text)

kxip_text = gsub("^\\s+|\\s+$", "", kxip_text)

try.error = function(x)

{

   y = NA

   try_error = tryCatch(tolower(x), error=function(e) e)
```

```
    if (!inherits(try_error, "error"))
        y = tolower(x)
    return(y)
}
kxip_text = sapply(kxip_text, try.error)
kxip_text = kxip_text [!is.na(kxip_text)]
names(kxip_text) = NULL
class_emo = classify_emotion(kxip_text, algorithm="bayes", prior=1.0)
emotion = class_emo[,7]
emotion[is.na(emotion)] = "unknown"
class_pol = classify_polarity(kxip_text, algorithm="bayes")
polarity = class_pol[,4]
sent_df = data.frame(text= kxip_text1, emotion=emotion,polarity=polarity, stringsAsFactors=FALSE)
sent_df = within(sent_df,emotion <- factor(emotion, levels=names(sort(table(emotion), decreasing=TRUE))))
ggplot(sent_df, aes(x=polarity)) +
    geom_bar(aes(y=..count.., fill=polarity)) +
    scale_fill_brewer(palette="Dark2") +
    labs(x="polarity categories", y="number of tweets") +
```

```
    labs(title = "Sentiment Analysis of Tweets on IPL (KXIP)\n(classification
by polarity)")

abc<-sent_df[c(1,3)]

write.csv(abc,"kxip_csv_senti.csv",row.names = FALSE)

######## MI #########

mi_text1<-mi_text

mi_text<-apply(mi_text,1,function(x)
gsub("(RT|via)((?:\\b\\W*@\\w+)+)", "",x))

mi_text = gsub("@\\w+", "", mi_text)

mi_text = gsub("[[:punct:]]", "", mi_text )

mi_text = gsub("[[:digit:]]", "", mi_text)

mi_text = gsub("http\\w+", "", mi_text)

mi_text = gsub("[ \t]{2,}", "", mi_text)

mi_text = gsub("^\\s+|\\s+$", "", mi_text)

try.error = function(x)

{

    y = NA

    try_error = tryCatch(tolower(x), error=function(e) e)

    if (!inherits(try_error, "error"))

        y = tolower(x)
```

```
    return(y)
}
mi_text = sapply(mi_text, try.error)
mi_text = mi_text [!is.na(mi_text)]
names(mi_text) = NULL
class_emo = classify_emotion(mi_text, algorithm="bayes", prior=1.0)
emotion = class_emo[,7]
emotion[is.na(emotion)] = "unknown"
class_pol = classify_polarity(mi_text, algorithm="bayes")
polarity = class_pol[,4]
sent_df = data.frame(text= mi_text1, emotion=emotion,polarity=polarity,
stringsAsFactors=FALSE)
sent_df = within(sent_df,emotion <- factor(emotion,
levels=names(sort(table(emotion), decreasing=TRUE))))
ggplot(sent_df, aes(x=polarity)) +
    geom_bar(aes(y=..count.., fill=polarity)) +
    scale_fill_brewer(palette="Dark2") +
    labs(x="polarity categories", y="number of tweets") +
    labs(title = "Sentiment Analysis of Tweets on IPL (MI)\n(classification
by polarity)")
```

```
abc<-sent_df[c(1,3)]

write.csv(abc,"mi_csv_senti.csv",row.names = FALSE)

#########################################################

############ SENTIMENT ANALYSIS DONE ##############

############ Model comparison different file ###########
```

APPENDIX B

List of Publications

Communicated

1. Kumar A., Hooda P., Dabas V.,(2017, February),"Text Classification Algorithm for Mining Unstructured Data: A SWOT Analysis", IJIT.