Twitter Sentiment Analysis

Journal:	*Engineering Computations*
Manuscript ID	Draft
Manuscript Type:	Research Article
Keywords:	Sentiment Analysis, Naive Bayes Classifier, Lexical Analysis

DECLARATION

We hereby declare that the project work entitled Twitter Sentiment Analysis and Research Paper is an authentic record of our work carried out as requirements of the Capstone Project for the award of B. Tech degree in Computer Science and Engineering from Lovely Professional University, Phagwara, under the guidance of Mr. Shevta, during Jan to May 2022. All the information furnished in this capstone project report is based on our tensive work and is genuine. Project Group Number: KC228.

VENKATA SRIRAM
11803046
MOUNIKA BUNGA
11814710
VENKATA GANESH
11802955
B.DEVA SAI HRITVIK
11802864
T.SAI SANTHOSH
11802338

VENKATA SRIRAM
Date: 06-04-2022

MOUNIKA BUNGA
Date: 06-04-2022

VENKATA GANESH
Date: 06-04-2022

B DEVA SAI HRITVIK
Date: 06-04-2022

T SAI SANTHOSH
Date: 06-04-2022

CERTIFICATE

This is to certify that the declaration statement made by this group of students is correct to the best of my knowledge and belief. They have completed this Capstone Project under my guidance and supervision. The present work is the result of their original investigation, effort, and study. No part of the work has ever been submitted for any other degree at any University. The Capstone Project is fit for the submission and partial fulfillment of the conditions for the award of a B.Tech degree in Computer Science and engineering from Lovely Professional University, Phagwara.

Signature and Name of the Mentor

Ms. Sheveta

Designation

School of Computer Science and Engineering,
Lovely Professional University,
Phagwara, Punjab.

Date : 19 April 2022

ACKNOWLEDGMENT

We take this opportunity to express our gratitude to everyone who has supported us throughout the B.Tech program. Thank you for their enthusiasm, constructive criticism, and advice during work. We truly thank them for sharing their genuine and enlightening ideas on the many issues related to this project.

Most of all I would like to thank Ms. Shevta (16856) for his support and guidance at Lovely Professional University which is his key guide in helping us put this work together and make it a complete success with his suggestions and instructions to serve as a major contribution to the implementation project.

Madam also helped us with their important suggestions and helpful guidance at various stages of project completion. We owe sincere thanks to all the faculty members in the department of Computer Science & Engineering for their kind guidance and encouragement from time to time.

Table of Contents

1. Introduction 7
1.1. Context 7
1.2. Why Sentiment Analysis? 7
1.3. What is sentiment analysis? 7
1.4. Twitter: 7
1.5. Steps for analysis 7
1.6. Hashtags and Handle 8
1.7. Software Technology Used (libraries and languages) 8
1.8. Hardware Technology Used 8
1.9. Challenges 8
1.10. Advantages 9
1.11. Aim of the Software 9
1.12. Overview 9
2. Profile of the Problem 9
2.1. Description of the project 9
2.2. Scope of the project 9
3. Analysis of the Existing System 10
3.1. Introduction: 10
3.2. Usability: 10
3.3. Applications: 10
3.4. Problem Analysis 10
3.4.1. Feasibility Analysis 10
3.4.2. Technical Feasibility 11
3.4.3. Operational Feasibility 11
3.4.4. Economic Feasibility 11
3.4.5. Schedule Feasibility 12
4. Software Requirement Analysis 12
4.1. Requirement Definition 12
4.1.1. Modules and its Functionality 12
4.1.2. Internal Interface requirement 12
4.1.3. External Interface Requirement 13
4.1.4. Non-Functional Requirements 14
4.1.5. Execution Requirements 14
4.1.6. Security Requirements 15
4.1.7. Programming Quality Attributes 15
5. System Design and Architecture 15
5.1. System Process Representation 15
5.2. Flow Diagrams 17
5.3. Data Flow Diagram 18
5.4. Flowchart 20

6. Testing 20
6.1. Unit Testing 20
6.2. System Testing 21
6.3. Performance Testing 21
6.4. Verification and Validation 21
7. IMPLEMENTATION 22
7.1. Machine Learning 22
7.2. NAIVE BAYES CLASSIFIER 22
7.2.1. Algorithm Dictionary generation 23
7.2.2. Feature set generation 23
7.3. Natural Language Processing 23
8. Programming Tools 24
8.1. R Programming 24
9. Project Legacy 24
9.1. Current status of the project 24
9.2. Remaining areas of concern 24
9.3. Technical lessons learnt 24
10. User Manual &Source Code 24
10.1. Packages Used 24
10.2. Project visualization and its code 25
10.2.3. Extraction of Tweets 25
10.2.4. CleaningTweets 26
10.2.5. Loading Database 27
10.2.6. Algorithms used 28

List of Tables and Figures

Figure 1 hastags and handle 8
Figure 2 Architecture 16
Figure 3 Process representation 17
Figure 4 flow diagrams 18
Figure 5 Data flow 19
Figure 6 flowchart 20
Figure 7 cleaning tweets 27
Figure 8 lexical analysis 30
Figure 9 final/_score 33
Figure 10 histogram of negative tweets 34
Figure 11 histogram of positive tweets 34
Figure 12 Sentiment analysis 35
Figure 13 hashtag ouputs 39
Figure 14 frequency in tweets 40

1. Introduction

1.1. Context

- Various outlets accessible for people to communicate opinions and emotions...positive,pessimistic, and unbiased.
- Need to advance positive news, respond to the negative, and make at least some difference well on unbiased news ... as close to constant as could really be expected
- Mining high volume, high speed information for significant bits of knowledge is difficult. to an extreme, excessively quick

1.2. Why Sentiment Analysis?

Assume we have an application that is extremely popular, and has around a billion clients and we chose to add another usefulness to our application, then how might we get criticism for it? With feeling examination, we can zero in on our pessimistic posts and work on our application.

1.3. What is sentiment analysis?

The process of computationally analyzing and differentiating opinions expressed in a context, particularly to determine whether the author's attention towards a content, product, etc. is understandable, confused, or neutral.

1.4. Twitter:

Twitter is an online social networking and micro-blogging service that enables users to make and peruse short messages, called "Tweets". It is a worldwide gathering with the presence of prominent characters from the field of amusement, industry and legislative issues. Individuals tweet about their life, occasions and offer viewpoint about different points utilizing instant messages restricted to 140 characters the selected individuals can dissect and impart their insights through tweets, however the unrolled individuals can understand them.

Objectives–

- To implement an algorithm for automatic classification of text in to positive, negative or unbiased.
- Feeling analysis to determine the attitude of the mass is whether justifiable, confused, or neutral towards the subject of interest.

1.5. Steps for analysis

- Data Pre-processing
- Data Cleaning
- Applying classification algorithms
- Classified tweets
- Sentiments in graphical representation
- Data collection using Twitter API.

1.6. Hashtags and Handle

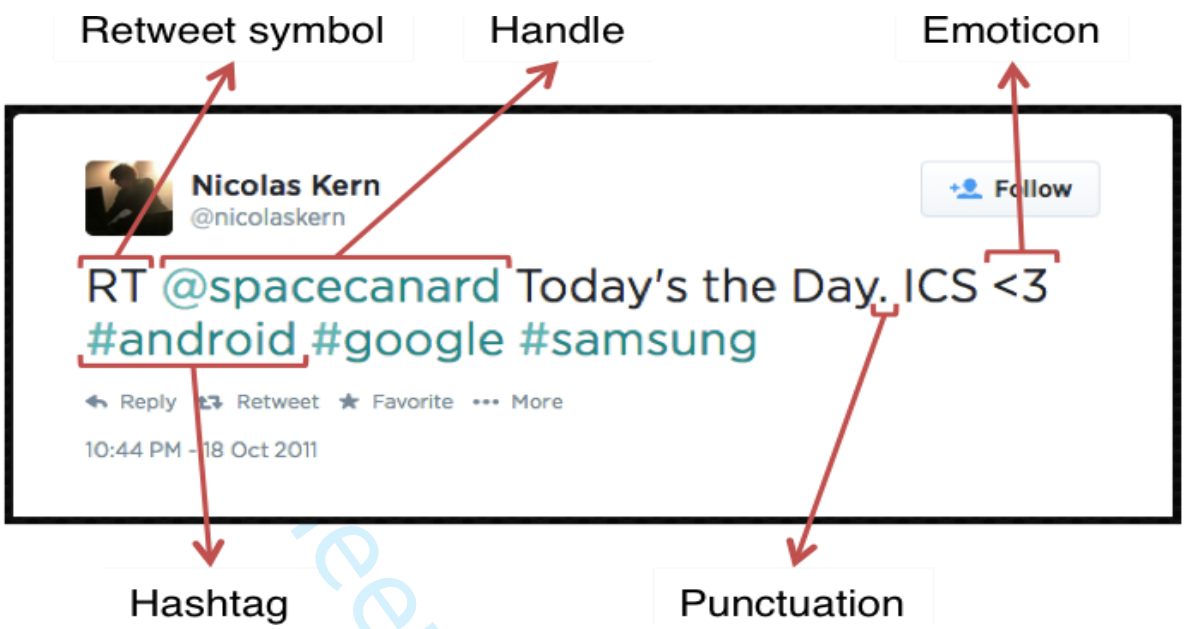


Figure 1 hastags and handle

1.7. Software Technology Used (libraries and languages)

- Windows 7 or higher
- R 3.4.4 or higher
- R studio with shiny package
- Notepad++
- Twitter API

1.8. Hardware Technology Used

- Processor – Dual Core
- Hard Disk – 50 GB
- Memory – 1GBRAM

1.9. Challenges

- A positive or negative sentiment word may have opposite orientation in different application domains.
- Accessing twitter API
- Sarcastic sentences with or without sentiment words are hard to deal.
- Many sentences without sentiment words can also imply opinions.

- A sentence containing sentiment words may not contains any sentiments.

1.10. Advantages

- Allows for automated sentiment analysis system.
- Adjust Marketing Strategy.
- Asentimentanalysisislikepersonallifeguardtomakesurethatwearetruly listening to what our customers think, want and need.
- Improve Customer Service

1.11. Aim of the Software

- This system can be used for online sentiment analysis system.
- Thesystemcanalsobeimplementedindifferentorganizationsthatwantsto read the mood of the users and improve its customer service.

1.12. Overview

Tweets are imported using R and the data is cleaned by removing emoticons and URLs. Lexical Analysis as well as Naïve Bayes Classifiers used to predict the sentiment of tweets and subsequently express the opinion graphically through ggplots, histogram, pie chart, word cloud and tables.

2. Profile of the Problem

The main objective of the application is to extract data from the twitter and clean it and then examination and visualize the outcome. At long last giving the general outcome whether the post got positive or negative reviews.

2.1. Description of the project

Tweets are imported using R and the data is cleaned by removing emoticons and URLs. Lexical Analysis as well as Naive Bayes Classifier is utilized to foresee the feeling of tweets and therefore offer the viewpoint graphically through ggplots, histogram, pie outline, word cloud and tables.

2.2. Scope of the project

- The applications for opinion examination are unending. It is incredibly helpful in friendly media monitoring as it allows us to gain an overview of the wider public opinion behind certain themes However, it is likewise viable for use in business investigation and circumstances in which text should be analyzed.
- Sentiment analysis is in demand because of its efficiency. Thousands of message reports can be handled for feeling like a flash, contrasted with the hours it would take a group of people to manually complete. Because it is so efficient (and accurate-Semantriahas 80% exactness for English substance) numerous organizations are embtacing message and opinion investigation and consolidating it into their cycles.

3. Analysis of the Existing System

3.1. Introduction:

- Twitter is an online news and social networking service that enables users to send and read short140-charactermessagescalled"tweets".
- Hence Twitter is a public platform with amine of public opinion of people all over the world and of all age categories.
- As of December 202, Twitter has more than 400 million monthly active users.
- Twitter Sentiment Analysis is the process of managing the sensitive nature after a series of information related to obtaining a conclusion of the views, opinions, and sentiments communicated inside an on-line mention.

3.2. Usability:

- The reasons for opinion investigation are boundless. It is strikingly valuable in web-based entertainment checking as it empowers us to acquire an outline of the more profound public judgment behind specific subjects Yet, it is likewise reasonable for use in advertising examination and conditions in which text should be made sense of.
- Feeling investigation is in need due to its skill large number of compositions archives can be handled for opinion right away, contrasted with the hours it would take at eam of individuals to finish physically. Since it is so productive (and exact - Semetria has 80% precision for English substance) numerous organizations are taking on message and feeling investigation and joining it into their cycles.

3.3. Applications:

- The importance of sentiment analysis is wide and important.
- Variations under sentiment on social media have been given to associate with variation in stock market.
- Let us take, the Obama presidency used sentiment analysis to measure public opinion to policy posters and campaign information winning of the 2012 presidential election.
- The capacity to instantly know consumer opinions and react accordingly.
- Canada took advantage of when they mentioned that there was a constant increase.

3.4. Problem Analysis

3.4.1. Feasibility Analysis

An attainability study is a starter concentrate on which explores the data of planned clients and decides the assets necessities, expenses, advantages and plausibility of proposed framework. An attainability concentrate on considers different limitations inside which the framework ought to be executed and worked. In this stage, the asset required for the execution, for example, registering hardware, labor and expenses are assessed. The assessed are contrasted and accessible assets anda money saving advantage examination of the

framework is made. The attainability examination action includes the investigation of the issue and assortment of all significant data connecting with the undertaking. The principle targets of the attainability study are to decide if the venture would be doable as far as financial possibility, specialized feasibilityandoperationalfeasibilityandschedulefeasibilityornot. It is to ensure that the information which are expected for the undertaking are accessible. Accordingly, we assessed the achievability of the framework concerning the accompanying classifications:

- Technical feasibility
- Operational feasibility
- Economic feasibility
- Schedule feasibility

3.4.2. Technical Feasibility

Assessing the specialized possibility is the trickiest piece of a practicality study. This is on the grounds that, at the particular moment there is no any definite planned of the framework, making it hard to get to issues like execution, costs (by virtue of the sort of innovation to be conveyed) and so on. A few issues should be considered while doing a specialized investigation; comprehend the various innovations engaged with the proposed framework. Before starting the venture, we should be exceptionally clear about what are the innovations that are to be expected for the improvement of the new framework. Is the expected innovation accessible? Our framework "Twitter Sentiment Analysis" is actually practical in all the required device streakily accessible. Examination and Perception can be effortlessly taken care of with R. Albeit all instruments appear to be effectively accessible there are difficulties as well.

3.4.3. Operational Feasibility

Proposed project is helpful provided that it tends to be transformed into data frameworks that will meet the working prerequisites. Essentially expressed, this trial of feasibility asks on the off chance that the framework will work when it is created and introduced. Are there significant boundaries to Implementation? The proposed was to make an improved on web application. It is more straightforward to work and can be utilized in any pages. It is free and not expensive to work.

3.4.4. Economic Feasibility

Economic feasibility attempts to gauge the expenses of creating and implementing a new framework, against the advantages that would build from having the new framework set up. This plausibility concentrate on gives the top administration the monetary legitimization for the new framework. A basic financial investigation which gives the genuine correlation of expenses and advantages are significantly more significant for this situation. Moreover, this ends up being valuable perspective to think about real expenses as the undertaking advances. There could be different sorts of theoretical advantages because of robotization. As These could increment improvement in item quality, better direction, and practicality of data, speeding up exercises, further developed precision of tasks, better documentation and record keeping,

quicker recovery of data.

3.4.5. Schedule Feasibility

A project will fail if it takes too long to be done before it is necessary. Typically, this suggests considering how long the policy will take to evolve, and if it can be completed in a given period using some methods like payback period. Schedule feasibility is a measure how reasonable the project time table is. Given our technical expertise, are the project deadlines reasonable? Some project is initiated with specific deadlines. It is necessary to determine whether the deadlines are mandatory or desirable.

A minor deviation can be encountered in the original schedule decided at the beginning of the project. The application development is feasible in terms of schedule.

4. Software Requirement Analysis

4.1. Requirement Definition

After the extensive analysis of the problems in the system, we are familiarized with the requirement that the current system needs. The requirement that the system needs is categorized into the functional and non-functional requirements. These requirements are listed below:

4.1.1. Modules and its Functionality

Functional requirement are the functions or features that must be included in any system to satisfy the business needs and be acceptable to the users. Based on this, the functional requirements that the system must require are as follows:

System should be able to process new tweets stored in database after retrieval

System should be able to analyze data and classify each tweet polarity

4.1.2. Internal Interface requirement

Recognize the item whose product necessities are determined in this record, including the amendment or delivery number. Depict the extent of the item that is covered by this SRS, especially assuming this SRS portrays just piece of the framework or a solitary subsystem.

Depict any guidelines or typographical shows that were adhered to while composing this SRS, for example, text styles or featuring that have extraordinary importance. For instance, state whether needs for more significant level necessities are thought to be acquired by itemized prerequisites, or whether each prerequisite assertion is to have its own need.

Depict the various sorts of per usre that the report is expected for, like engineers, project directors, promoting staff, clients, analyzers, and documentation journalists. Portray what there of this SRS contains and the way things are coordinated. Recommend an arrangement for perusing the record, starting with the outline areas and continuing through the segments that are generally appropriate to every per user type.

Provide a short description of the software being specified and its purpose, including pertinent advantages, targets, and objectives. Relate the product to corporate objectives or business methodologies. In the event that a different vision and extension record is accessible, allude to it as opposed to copying its substance here. The new blast in information relating to clients via web-based entertainment has made an extraordinary

interest in performing opinion examination on this data using Big Data and Machine Learning principles to understand people's inclinations. This venture plans to play out similar errands. The contrast between this undertaking and other feeling investigation instruments is that it will perform an ongoing examination of tweets in view of hashtags and not on a put-away chronicle.

Depict the unique circumstance and beginning of the item being indicated in this SRS. For instance, state whether this item is a; follow-on individual from an item family, a substitution for specific existing framework, or a new, independent item. In the event that the; SRS characterizes a part of a bigger framework, relate the necessities of the bigger framework to the usefulness of this; product and distinguish interfaces between the two. A basic graph that shows the significant parts of the general framework, subsystem interconnections, and outside connection points can be useful.

The Product capacities are

- Gather tweets in an ongoing design i.e., from the twitter live stream in light of indicated hashtags
- Eliminate excess data from these gathered tweets.
- Store the arranged tweets in MongoDB data set
- Perform Sentiment Analysis on the tweets put away in the data set to arrange their tendency viz. positive, negative and soon.
- Utilize an AI calculation which will anticipate the 'mind-set' individuals concerning that theme.

Sum up the significant capacities the item should perform or should allow the client to perform. Subtleties will be given in Section 3, so just an undeniable level outline (such as a bullet list) is required here. Put together the capacities to make the maunders and ready to; any per user of the SRS. An image of the significant gatherings of related prerequisites and how they relate, like a high level information; stream graph or item class outline, is frequently compelling.

Distinguish the different client classes that you expect will utilize this item. Client classes might bed if forensicated in view of recurrence of purpose, subset of item works

utilized, specialized ability, security or honor levels, instructive level, or experience. Portray the appropriate attributes of every client class. Certain necessities might relate just to specific client classes. Recognize the main client classes for this item from the people who are less critical to fulfill.

Depict the climate where these of these will work, including the equipment stage, working framework and forms, and some other programming parts or applications with which it should calmly coincide.

4.1.3. External Interface Requirement

We classify External Interface in 4 types:

4.1.3.1. User Interface

Depict the coherent qualities of every connection point between the product item and the clients. This might incorporate example screen pictures, any GUI guidelines or item family style directs that are to be followed, screen design imperatives, standard fastens and works (e.g., help) that will show up on each screen, console alternate ways, mistake message show norms, etc. Characterize the product parts for which a UI is required. Subtleties of the UI configuration ought to be archived in a different UI detail.

4.1.3.2. Equipment interface

Depict the intelligent and actual attributes of every point of interaction between the product item and the hard product parts of the framework. This might incorporate the upheld gadget types, the idea of the information and control collaborations between the product and the equipment, and correspondence conventions to be utilized.

4.1.3.3. Programming Interface

Depict the associations between this item and other explicit programming parts (name and form), including information bases, working frameworks, apparatuses, libraries, and coordinated business parts. Distinguish the information things or messages coming into the framework and going out and depict the motivation behind each. Depict the administrations required and the idea of correspondences. Allude to reports that portrayed etailed application programming point of interaction conventions. Recognize information that will be shared across programming parts. On the off chance that the information sharing component should be executed with a certain goal in mind (for instance, utilization of a worldwide information region in a performing various tasks working framework), determine this as an execution imperative.

4.1.3.4. Correspondence Interface

Depict the necessities related with any correspondences capacities expected by this item, including email, internet browser, network server interchanges conventions, electronic structures, etc. Characterize any relevant message arranging. Recognize any correspondence guidelines that will be utilized, like FTP or HTTP. Determine any correspondence security or encryption issues, information move rates, and synchronization instruments.

4.1.4. Non-Functional Requirements

Non-practical prerequisites are a portrayal of elements, qualities and property of the framework as well as any requirements that might restrict the limits of the proposed framework.

The non-useful necessities are basically founded on the exhibition, data, economy, control and security productivity and administrations. In view of these the non-practical prerequisites are as per the following:

- Easy to understand.
- Framework ought to give better precision.
- To perform with effective throughput and reaction time.

4.1.5. Execution Requirements

Assuming there are execution prerequisites for the item under different conditions, state them here and make

sense of their reasoning, to assist the engineers with getting the plan and settle on appropriate plan decisions. Indicate the circumstance connections for continuous frameworks. Make such prerequisites as unambiguous as could be expected. You might have to state execution necessities for individual useful prerequisites or highlights.

4.1.6. Security Requirements

Indicate those necessities that are worried about conceivable misfortune, harm, or mischief that could result from the utilization of the item. Characterize any protections or moves that should be made, as well as activities that should be forestalled. Allude to any outer arrangements or guidelines that state wellbeing issues that influence the item's plan or use. Characterize any security accreditations that should be fulfilled. Determine any prerequisites with respect to security or protection issues encompassing utilization of the item or assurance of the information utilized or made by the item. Characterize any client personality validation prerequisites. Allude to any outside approaches or guidelines containing security gives that influence the item.

Characterize any security or protection certificates that should be fulfilled.

4.1.7. Programming Quality Attributes

Indicate any extra quality attributes for the item that will be critical to either the clients or the engineers. Some to consider are: versatility, accessibility, accuracy, adaptability, interoperability, viability, conveyability, unwavering quality, reusability, heartiness, testability, and ease of use. Compose these to be explicit, quantitative, and obvious whenever the situation allows. At any rate, explain the general inclinations for different characteristics, like convenience over simplicity of learning.

5. System Design and Architecture

5.1. System Process Representation

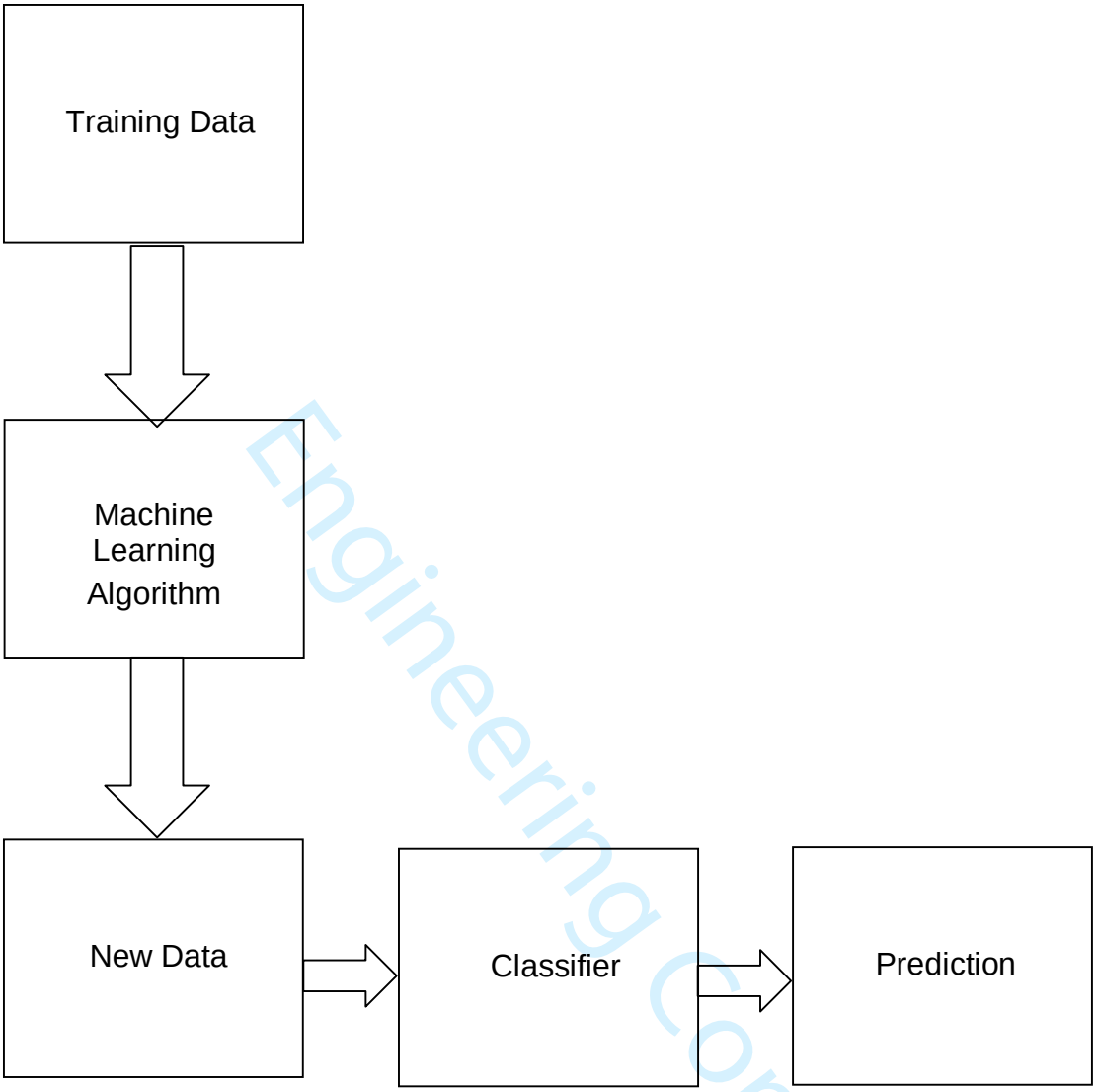


Figure 2 Architecture

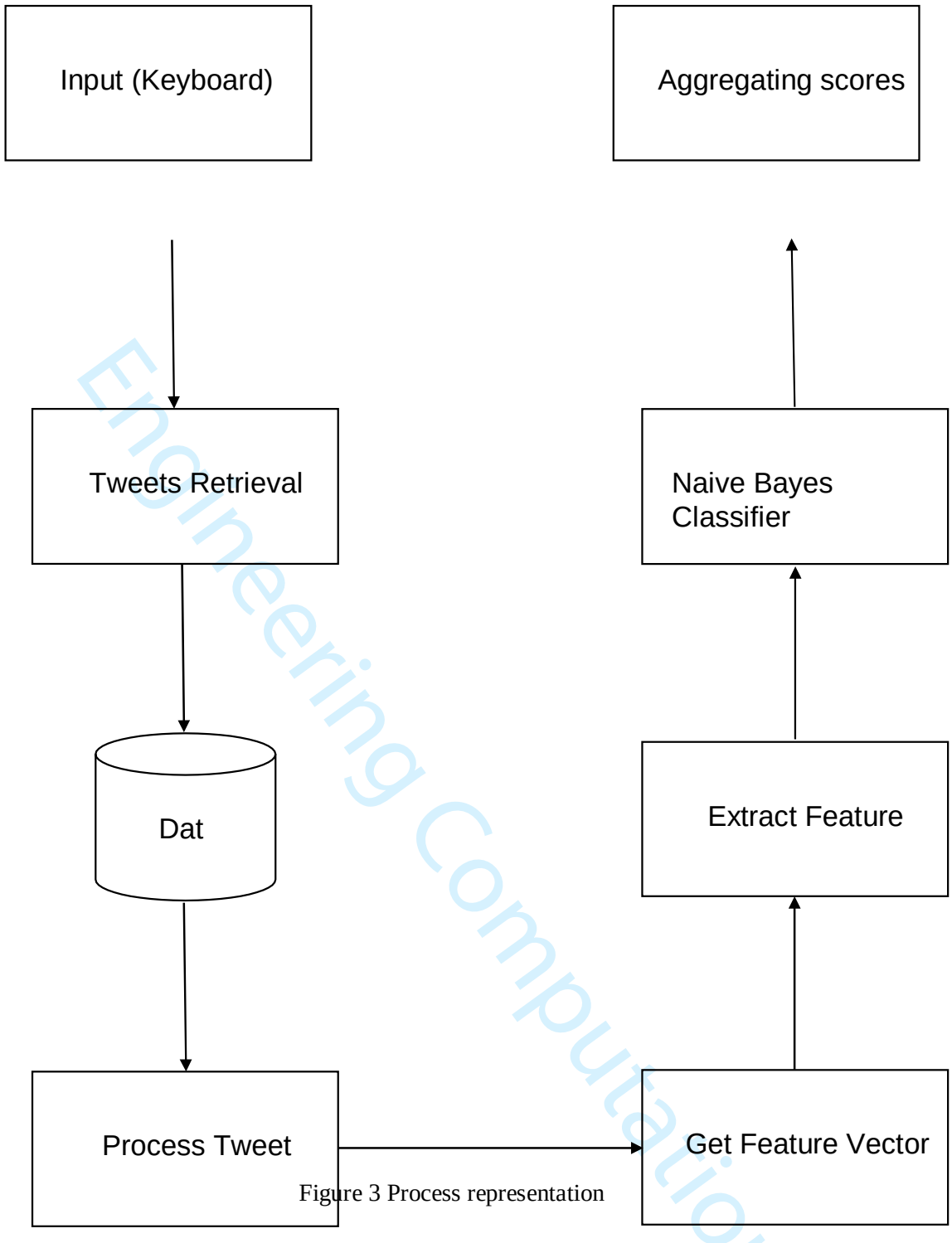


Figure 3 Process representation

5.2. Flow Diagrams

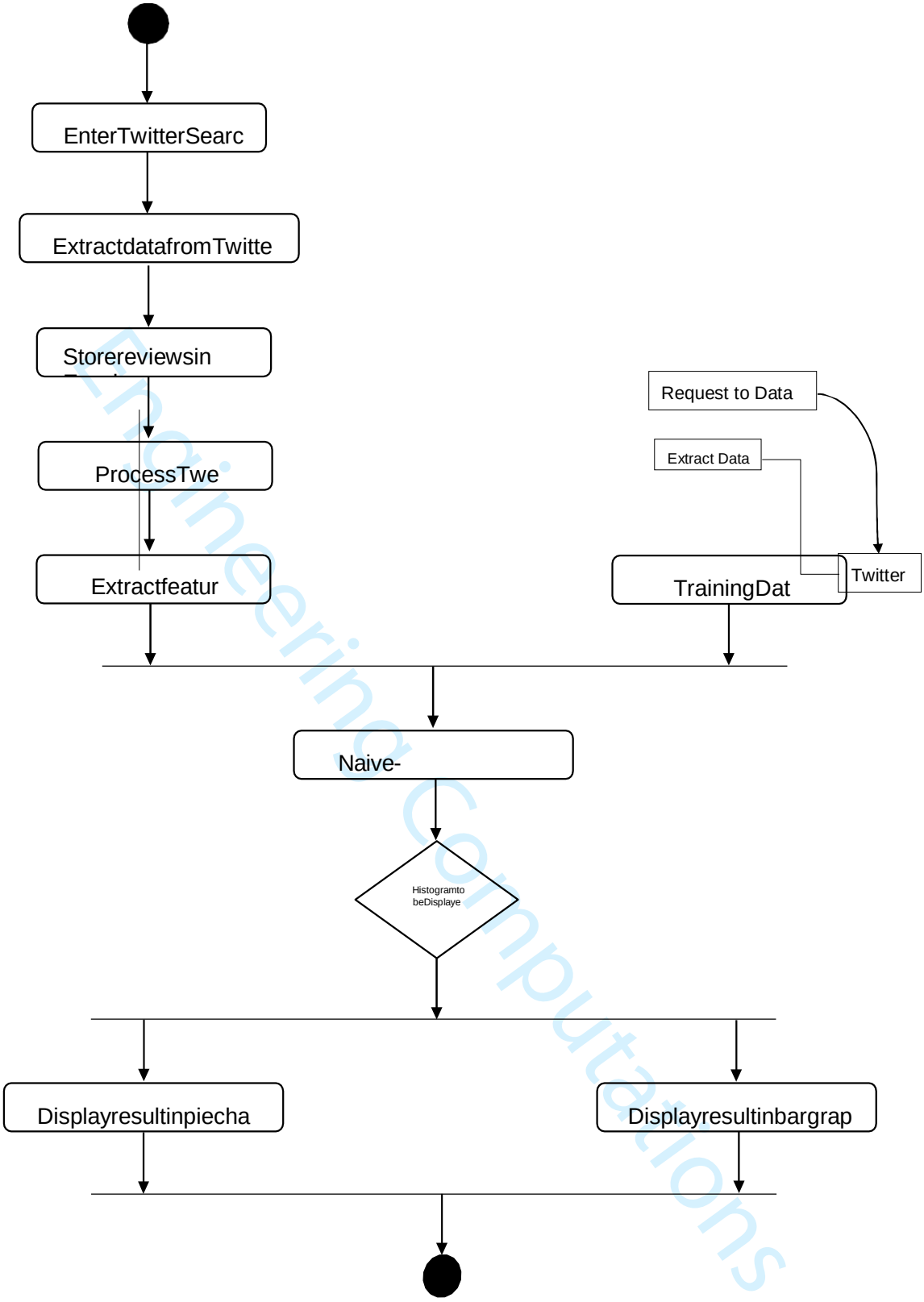


Figure 4 flow diagrams

5.3. Data Flow Diagram

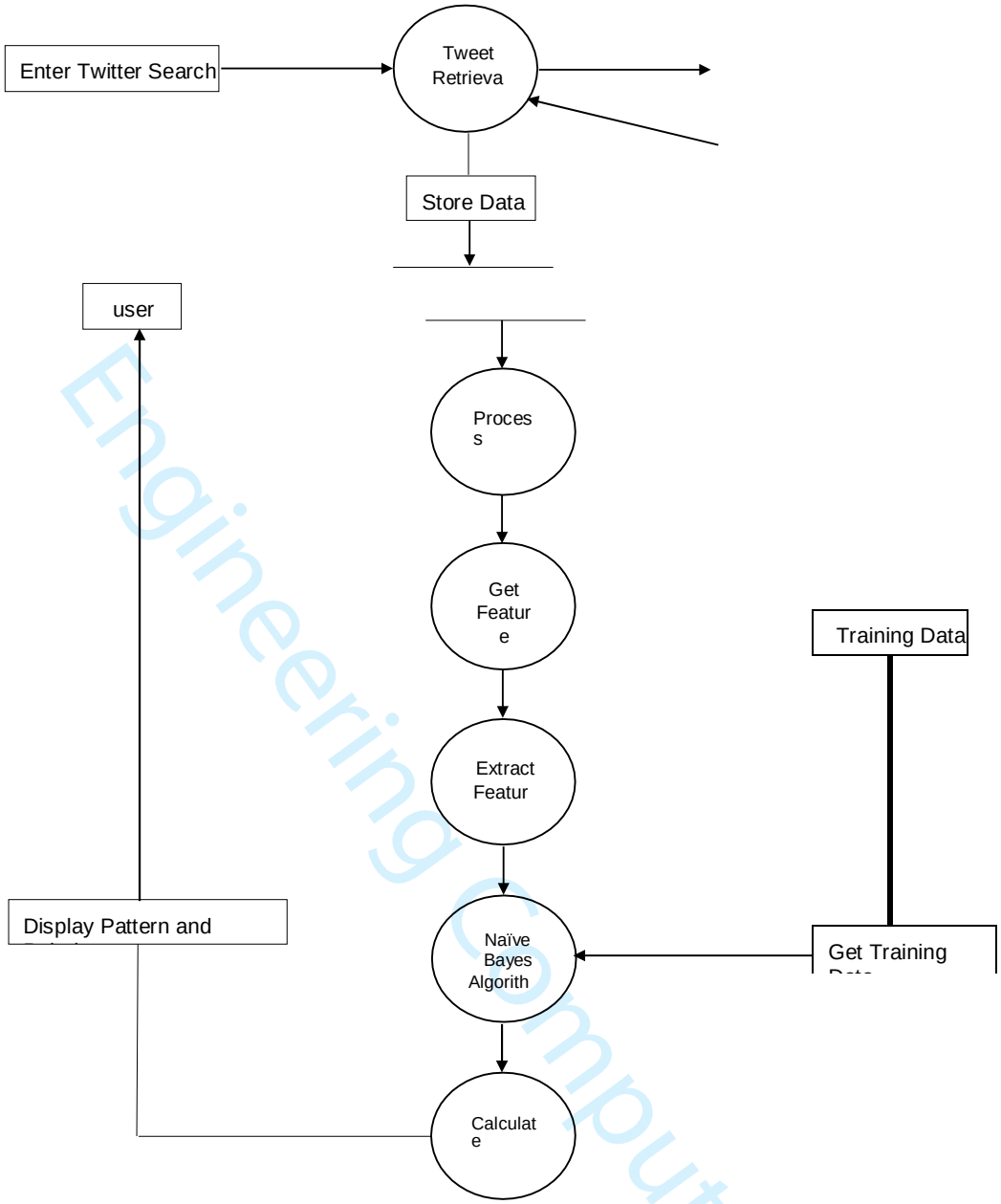


Figure 5 Data flow

5.4. Flowchart

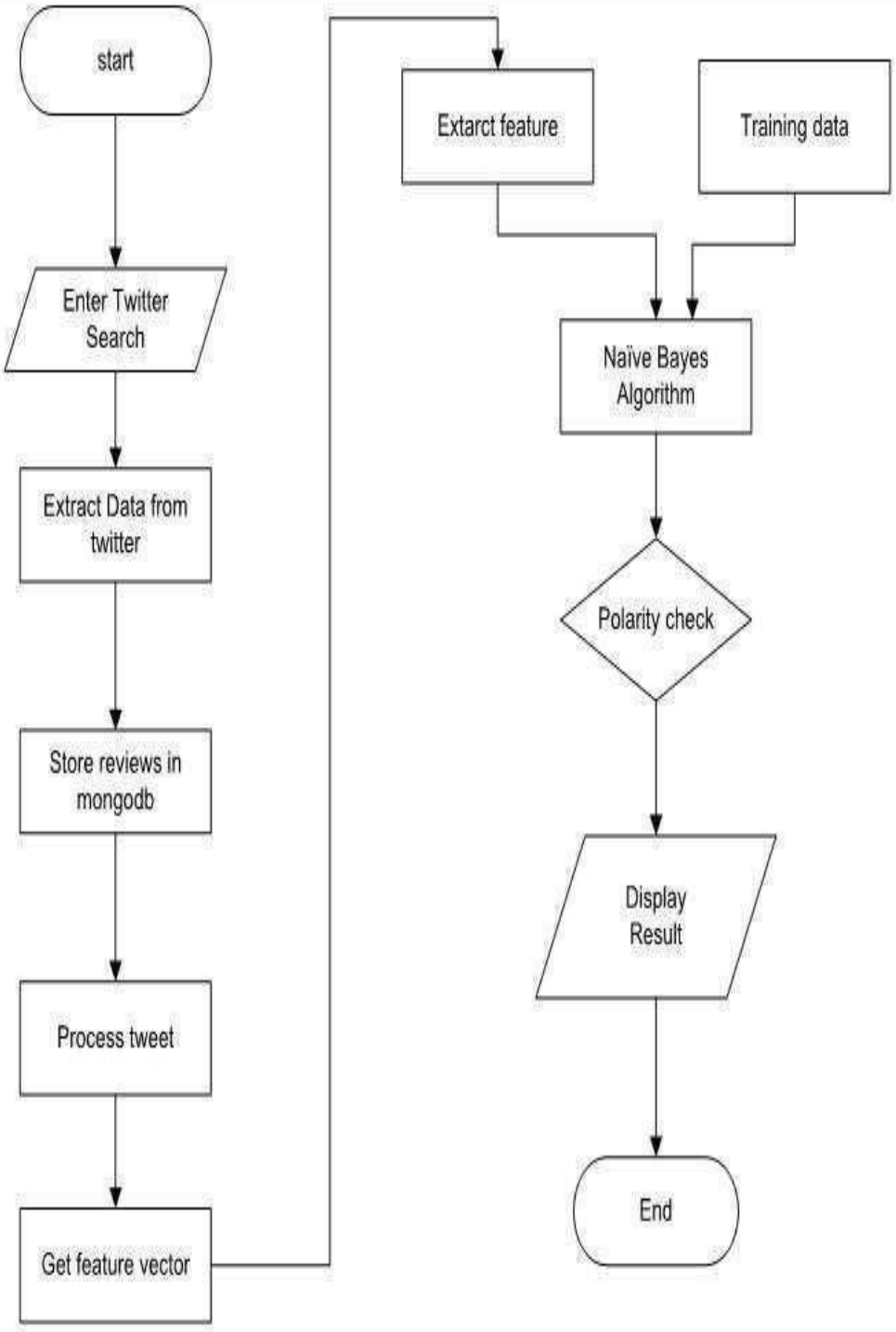


Figure 6 flowchart

6. Testing

6.1. Unit Testing

Unit testing is performed for testing modules against detailed design. Inputs to the process are usually compiled modules from the coding process. Each module is assembled into a larger unit during the unit testing process.

Testing has been performed on each phase of project design and coding. We carry out the testing of module

interface to ensure the proper flow of information into and out of the program unit while testing. We make sure that the temporarily stored data maintains its integrity throughout the algorithm's execution by examining the local data structure. Finally, all error- handling paths are also tested.

6.2. System Testing

We usually perform system testing to find errors resulting from unanticipated interaction between the sub-system and system components. Software must be tested to detect and rectify all possible errors once the source code is generated before delivering it to the customers. For finding errors, a series of test cases must be developed which ultimately uncover all the possibly existing errors. Different software techniques can be used for this process. These techniques provide systematic guidance for designing a test that

- Exercise the internal logic of the software components,
- `Exercise the input and output domains of a program to uncover errors in program function, behavior and performance.

We test the software using two methods:

i) White Box testing: Internal program logic is exercised using this test case design techniques.
ii) Black Box testing: Software requirements are exercise dusing this test case design techniques.

Both techniques help in finding maximum number of errors with minimal effort and time.

6.3. Performance Testing

It is done to test the run-time performance of the software within the context of integrated system. These tests are carried out throughout the testing process. For example, the performance of individual module is accessed.

6.4. Verification and Validation

The testing process is a part of broader subject referring to verification and validation. We must acknowledge the system specifications and try to meet the customer’s requirements and for this sole purpose, we must verify and validate the product to make sure everything is in place. Verification and validation are two different things. One is performed to ensure that the software correctly implements a specific functionality and other is done to ensure if the customer requirements are properly met or not by the product.

Verification is more like 'are we building the product, right?' and validation is more like 'are we building the right product.

7. IMPLEMENTATION

There are primarily two types of implementations for sentiment classification of opinionated texts

- Using a Machine learning based text classifier such as Naive Bayes
- Using Natural Language Processing

7.1. Machine Learning

It depends on text classifiers, they are a sort of managed AI model, where the classifier requests to be directed by some marked preparation information be for it tends to be applied to genuine characterization task. The preparation information is normally a separated piece of the first information hand named physically. After reasonable preparation they can be utilized on the real test information. The Naive Bayes is a factual classifier though Support Vector Machine is a sort of vector space classifier. The factual message classifier plan of Naïve Bayes (NB)can be adjusted to be utilized for feeling grouping issue

As it very well may be picture dasa2-class text order issue: in certain and negative classes. Support Vector machine (SVM) is a sort of vector space model-based classifier which expects that the text reports ought to be changed to include vectors before they are utilized for order. Normally, the text reports are changed to multi-faceted vectors. The whole problemofclassificationisthenclassifyingeverytextdocumentrepresented as a vector into a class. It is at type of large edge classifier. Here the goalies to observe a choice limit between two classes that is maximally distant from any report in the preparation information.

This approach needs

- A decent classifier like Naïve Byes
- A preparation set for each class

There are different preparation sets accessible on Internet, for example, Movie Reviews informational index, twitter dataset, and so forth. Class can be Positive, negative. For both the classes we really want preparing informational indexes.

7.2. NAIVE BAYES CLASSIFIER

The Naïve Bayes classifier is the simplest and most commonly used classifier. Naïve Bayes classification model computes the posterior probability of a class, based on the distribution of the words in the document. The model works with the BOWs feature extraction which ignores the position of the word in the document. It uses Bayes Theorem to predict the probability that a given feature set belongs to a label.

$$P(\text{label} \mid \text{features}) = \frac{P(\text{label})\ P(\text{features} \mid \text{label})}{}$$

P(features)$P(label)$ is the prior probability of a label or the likelihood that a

random feature set the label. $P(features \mid label)$ is the prior probability that a given

feature set is being classified as a label. *P*(*features*) is the prior probability that a given feature set is occurred. Given the Naïve assumption which states that all features are independent, the equation could be rewritten as follows:

$$P(label|features) = \frac{P(label)\, P(f1|label) \ldots * \ldots P(fn|label)}{P(features)}$$

7.2.1. Algorithm Dictionary generation

Count occurrence of all word in our whole data set and make a dictionary of some most frequent words.

7.2.2. Feature set generation

- All document is represented as a feature vector over the space of dictionary words.
- For each document, keep track of dictionary words along with their number of occurrences in that document.

7.3. Natural Language Processing

Regular language handling (NLP) is a field of software engineering, man-made consciousness, and etymology worried about the collaborations between PC sand human (natural)languages. This approach uses the freely accessible library of Senti Word Net, which gives an opinion extremity values to each term happening in the archive. In this lexical asset each term t happening in WordNet is related to three mathematical scores obj(t), pos(t)and neg(t), portraying the goal, positive and negative polarities of the term, separately. These three scores are figured by consolidating the outcomes delivered by eight ternary classifiers. Word Netisalar gelexical data set of English. Things, action words, modifiers and qualifiers are gathered into sets of mental equivalents (synsets), each communicating a particular idea.

WordNet is additionally uninhibitedly and freely accessible for download. WordNet's design makes it a helpful instrument for computational etymology and regular language handling. Its gatherings words together in view of their implications. Synet is only a bunch of at least one Synonyms. This approach utilizes Semantics to get the language.

Significant errands in NLP that aides in extricating opinion from a sentence:

- Removing part of the sentence that mirrors the feeling
- Getting the construction of the sentence
- Various devices which assist with handling the text based information

Fundamentally, Positive and Negatives centers got from Senti Word Net as per its grammatical feature tag and afterward by counting the all out sure and negative scores we decide the feeling extremity in light of

which class (for example either sure or negative) has gotten the most noteworthy score.

8. Programming Tools

8.1. R Programming

R is a programming language developed by Ross Ihaka and Robert Gentleman in 1993. R possesses an extensive catalogue of statistical and graphical methods. It includes machine learning algorithm, line a regression, time series and statistical inference to name a few. It is one of the most popular languages used by statisticians, data analysts, researchers and marketers to retrieve, clean, analyze, visualize and present data. Due to its expressive syntax and easy-to-use interface, it has grown in popularity in recent years. R is free to download as it is licensed under the terms of GNU
General Public license. You can look at the source to see what's happening under the hood.

9. Project Legacy

9.1. Current status of the project

The Twitter sentiment analysis is developed but some changes are to be made. But 95% of the application is completely developed.

9.2. Remaining areas of concern

For the emoji analysis there is a problem with importing dictionaries. R text tools package is not working as expected. But the user can analyze the tweets which are having the plain text. We need to request for the twitter access key 48 hours before the execution of the project, sometimes we may even get void response.

9.3. Technical lessons learnt

Technical lessons learnt; we have learnt to import the data from external systems. We learnt how to analyze bulk amount of data, we have learnt clean and visualize the imported data effectively. When developing the Project, a lot of issues we found like unavailability of the required dictionaries and packages. But after trying different versions of R and trying different functionality we can successfully complete the module. Managerial lesson that we learnt is how to develop a project in team while discussing with the group members.

10. User Manual &Source Code

10.1. Packages Used

- **TwitteR:** Provides an interface to the Twitter web API
- **String:** String operations in R.
- **R Curl:** Provides functions to allow one to compose general HTTP requests and provides convenient functions to fetch URIs, get & post forms, etc. and process the results returned by the Web serve

- **tm:** A framework for text mining applications within R.
- **RJSONIO:** This is a package that allows conversion to and from data in JavaScript object notation (JSON) format. This allows R objects to be inserted into JavaScript/ECMAScript/ActionScript code and allows R programmers to read and convert JSON content to R objects
- **word cloud:** visual representation in the form of word cloud where size of the word is proportional to the frequency of words used in the tweets
- **grid Extra**: Provides several user-level functions to work with "grid" graphics, notably to arrange multiple grid-based plots on a page and draw tables.
- **plyer**: Tools for Splitting, Applying and Combining Data

10.2. Project visualization and its code

10.2.3. Extraction of Tweets

I. Create twitter application

II. twitteR - Provides an interface to the Twitter webAPI

III. ROAuth - R Interface forOAuth

IV. Create twitter authenticated credential object (using key from step (ii) and cacert. pem certificate): It is done using consumer key, consumer secret, access token, access secret.

V. During authentication, we are redirected to a URL automatically where we click on Authorize app as shown in the image below and enter the unique 7-digit number to get linked to the account from which feeds are being taken.

Related Code

```
apiKey <-  "Nejxm88eaJ6F9D5sfI7VojuzF"
apiSecret <- "grWBLLxTj4ZscRbIjPo0mirYEayDcjbJV7w6cgXY8o8kP8uu6V"
accessToken <- "1097661052945195009-Yv0YYpx6930nycbMYcerxucPUfyAcV"
accessTokenSecret <- "IlYSKWWS6Bnjd2VgUZb9P8i1lMLvAZwgb5TKaAE8rGQKL"
setup_twitter_oauth(apiKey,
                    apiSecret,
                    accessToken,
                    accessTokenSecret)
```

```
tweets <- searchTwitter("RRR", n=100) # Max we can have 1500 tweets
tweets
class(tweets)
```

Output (First 5 only):

```
[[1]]
[1] "Sreenu71505458: RT @LMKMovieManiac: #RRR #RRRMovie Week1 total Chennai city gross is 5.24
CR 🤘\n\nVery good 2nd weekend ahead now."

[[2]]
[1] "KABITAG45642896: RT @pushpakchowdary: Terrific performances from #NTR &amp;amp; #RamCharan,
long live KING #SSRajamouli - Bollywood Superstar #KanganaRanaut after w…"

[[3]]
[1] "beinghpl: RT @JamesKL95: #RRR wine shop ah ..□□□□\nWhatte creativity😐
https://t.co/J0KvUX5U18"

[[4]]
[1] "Gg70XOG3cdo5JOS: @chi_rrr__ だったらヘアアイロン買うわって使ってるわ！笑笑\n毎日20分位www\nそ
の時間利用して色んな音楽聴いてる♪♪♪♪"

[[5]]
[1] "aaaa_aaa_rrr: 左腕なんか痛い"
```

10.2.4. CleaningTweets

The tweets are cleaned in R by removing:

- Extrapunctuation
- Stop words
- Redundant Blankspaces
- Emoticons
- URLS

Related Code:

```
df=do.call("rbind",lapply(tweets,as.data.frame))
```

```
class(df)


df
View(df)
df$text # to derive only tweet

df$text <- sapply(df$text,function(row) iconv(row, "latin1", "ASCII", sub="")) #remove emoticon, convert latin1 to ASCII and then substitute it with blank
df$text = gsub("(f|ht)tp(s?)://(.*)[.][a-z]+", "", df$text) #remove URL, here we used regular expression of URL, we replace it with nothing
sample <- df$text


head(sample)
```

Output:

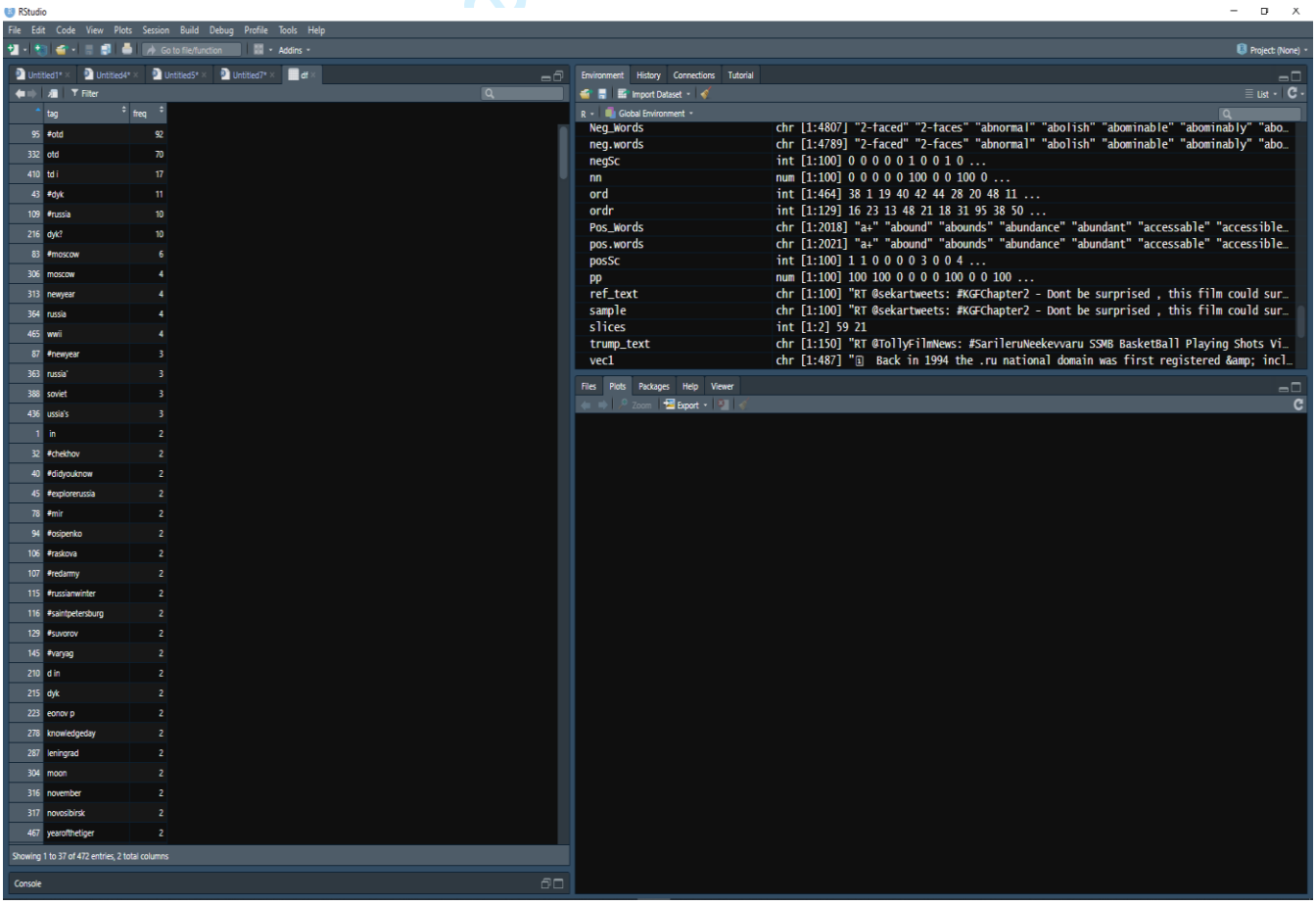


Figure 7 cleaning tweets

10.2.5. Loading Database

Form movie tweets, Naïve-Bayes

MachineLearningAlgorithmisused.AFINNisalistofEnglishwordsratedforvalence with an integer

between minus five (negative) and plus five (positive). The wordshave been manually labelled by Finn Arup Nielsen in 2009-2011. The file is tab-separated. The version used is: AFINN-111: Newest version with 2477 words an

Related Code:

```
getwd()
setwd("D:/Capstone Project/cp/Capstone")

Pos_Words=scan('D:/Capstone Project/cp/Capston/positive-words.txt',what='character',comment.char = ';')
#what tells what kind of data stored, ; depicts comment

Neg_Words=scan('D:/Capstone Project/cp/Capstone/negative-Words.txt',what='character',comment.char =
';')

#Adding words to positive and negative databases

Pos_Words=c(Pos_Words, 'fantastic','amazing','marvellous'
,'best','Congrats','massive','terrific','fabulous','sensational','magneficient' 'outstanding', 'blockbuster',
'thanks', 'thnx', 'Grt', 'gr8', 'trending', 'recovering', 'brainstorm', 'leader','like','love','good','nice')

Neg_Words = c(Neg_Words, 'Fight', 'fighting', 'wtf', 'arrest', 'no',
'not','bad','horrible','worst','hate','rude','abusive','terrible','ugly','sad','harmful','scary','worse')
```

10.2.6. Algorithms used

Lexical Analysis: By comparing uni-grams to the pre-loaded word database, the tweet is assigned sentiment score-positive, negative or neutral and overall score is calculated.

Related Code:

```
score.sentiment = function(sentences, Pos_Words, Neg_Words, .progress='none') #Function for lexical
analysis
{
 require(plyr)
 require(stringr)
 list=lapply(sentences, function(sentence, Pos_Words, Neg_Words)
 {
  sentence = gsub('[[:punct:]]',' ',sentence) #Convert punctuation to nothing
  sentence = gsub('[[:cntrl:]]','',sentence)  #Convert cntrl to nothing
  sentence = gsub('\\d+','',sentence)  #removes decimal number
  sentence = gsub('\n','',sentence)    #removes new lines
  sentence = tolower(sentence) # Convert sentences to lowercase
  word.list = str_split(sentence, '\\s+') #For this we need stringr package
  words = unlist(word.list)  #changes a list to character vector
  pos.matches = match(words, Pos_Words) #matching one word of words with Postive words database, if
```

```
matches then return data else na
    neg.matches = match(words, Neg_Words)
    pos.matches = !is.na(pos.matches) #remove na data, contains only positive
    neg.matches = !is.na(neg.matches)
    pp = sum(pos.matches)
    nn = sum(neg.matches)
    score = sum(pos.matches) - sum(neg.matches)
    list1 = c(score, pp, nn)
    return (list1)
  }, Pos_Words, Neg_Words)
  score_new = lapply(list, `[[`, 1) #Extract the first column from returned list
  pp1 = lapply(list, `[[`, 2) #Stores positive scores
  nn1 = lapply(list, `[[`, 3) #Stores negative scores

  scores.df = data.frame(score = score_new, text=sentences)
  positive.df = data.frame(Positive = pp1, text=sentences)
  negative.df = data.frame(Negative = nn1, text=sentences)

  list_df = list(scores.df, positive.df, negative.df)
  return(list_df)
}

require(plyr)
result=score(sample,Pos_Words,Neg_Words)

result
require(reshape)
```

Output

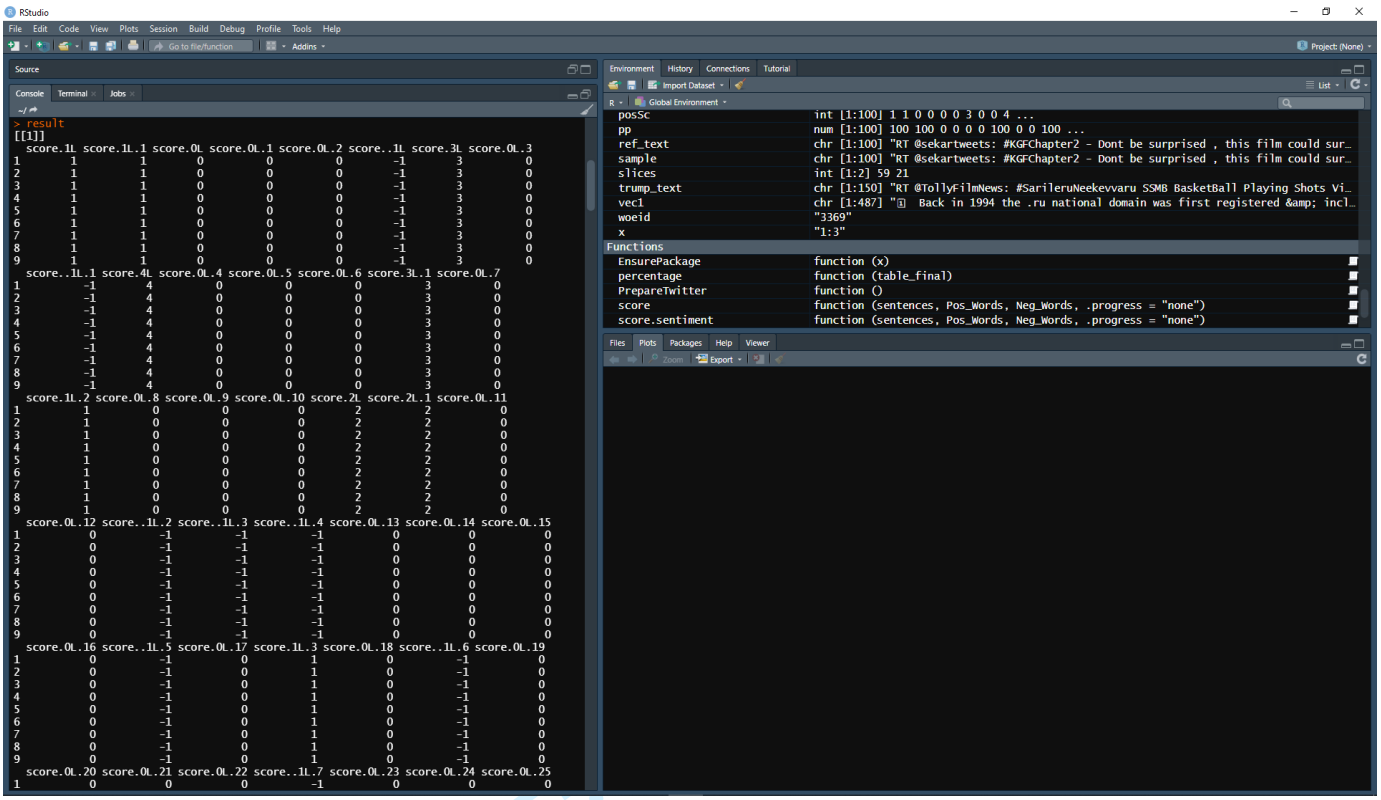


Figure 8 lexical analysis

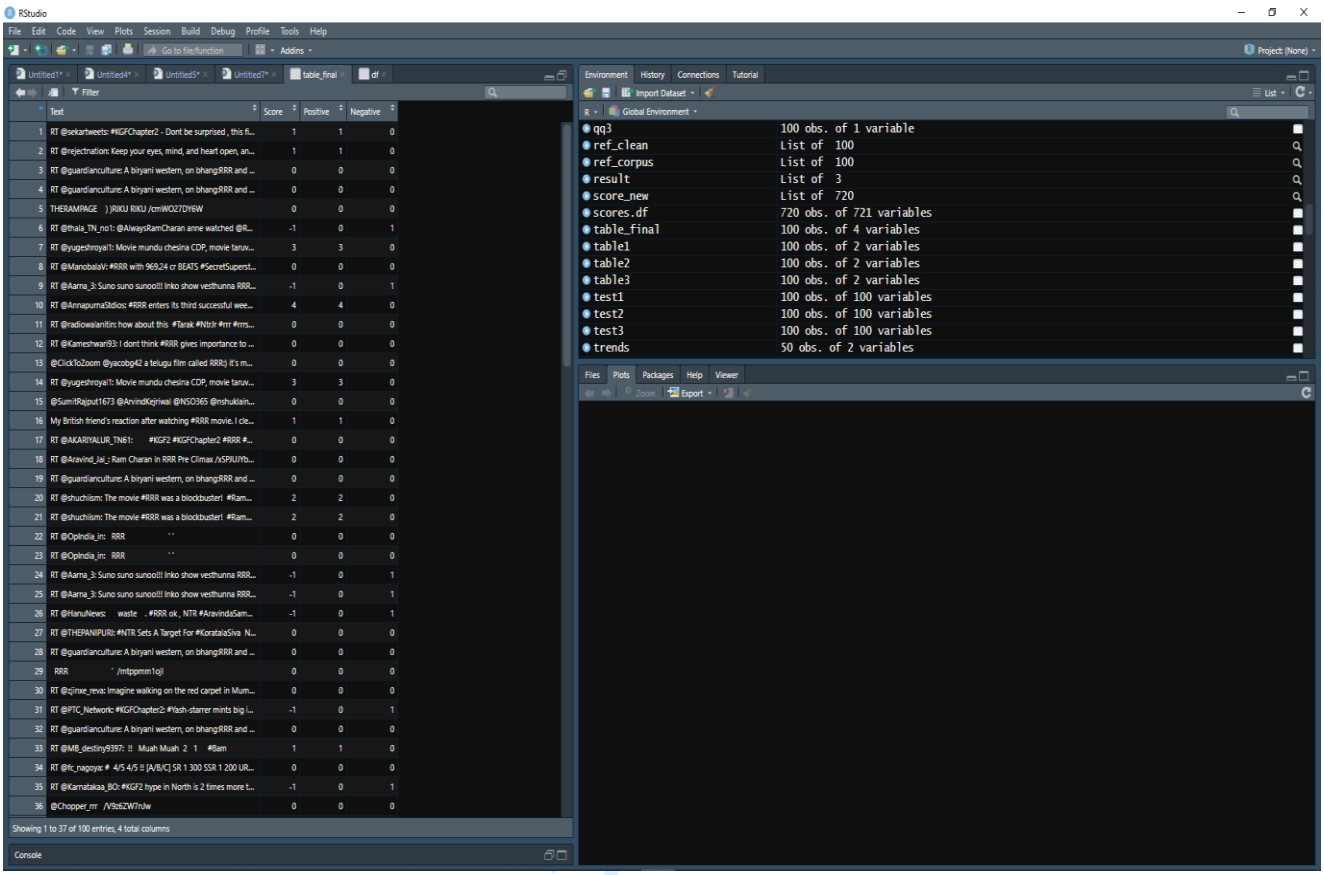
Naive Bayes Machine Learning Algorithm: Training data sets are used to teach the machine what kind of sentences are categorized as positive and what kind are categorized as negative. On arrival of a new tweet or sentence, the machine uses this algorithm to give the correct category to the new data and adds level to the motion.

Related Code:

```
result = score.sentiment(sample, Pos_Words, Neg_Words)


library(reshape)
test1=result[[1]] #Stores the score that is merged and tweets
test2=result[[2]] #stores the positive scores and tweets
test3=result[[3]] #stores the negative scores and tweets


#Creating three different data frames for Score, Positive and Negative
#Removing text column i.e tweet from data frame
test1$text=NULL
test2$text=NULL
test3$text=NULL


#Storing the first row(Containing the sentiment scores) in variable q
q1=test1[1,]
```

```
q2=test2[1,]
q3=test3[1,]
qq1=melt(q1, var='Score') #It melts the score in one column

qq2=melt(q2, var='Positive')
qq3=melt(q3, var='Negative')

qq1['Score'] = NULL
qq2['Positive'] = NULL
qq3['Negative'] = NULL

#Creating data frame
table1 = data.frame(Text=result[[1]]$text, Score=qq1)
table2 = data.frame(Text=result[[2]]$text, Score=qq2)
table3 = data.frame(Text=result[[3]]$text, Score=qq3)

#Merging three data frames into one
table_final=data.frame(Text=table1$Text,Score=table1$value,Positive=table2$value,
Negative=table3$value)
View(table_final)
table_final
```

Output:



Calculating percentage

Here we have presented the scores, the tweets as well as the percentage of positive/negative emotion in the text. This calculated using simple arithmetic to understand the overall sentiment in a better manner.

Related Code

```
#Positive Percentage
#Renaming
posSc=table_final$Positive
negSc=table_final$Negative
#Adding column
table_final$PosPercent = posSc/ (posSc+negSc)*100
#Replacing Nan(Not a number when divided by 0) with zero
pp = table_final$PosPercent
pp[is.nan(pp)] <- 0
table_final$PosPercent = pp
#Negative Percentage
#Adding column
table_final$NegPercent = negSc/ (posSc+negSc)*100

#Replacing Nan with zero
nn = table_final$NegPercent
nn[is.nan(nn)] <- 0
```

```
table_final$NegPercent = nn
View(table_final)
table_final
```

Histogram tab: histogram plot

Histograms of positive, negative and overall score are found under the Histogram tab for graphically analyzing the intensity of emotion in the tweeters.

Related Code:

```
hist(table_final$Positive, col=rainbow(10))
hist(table_final$Negative, col=rainbow(10))
hist(table_final$Score, col=rainbow(10))
```

Output:

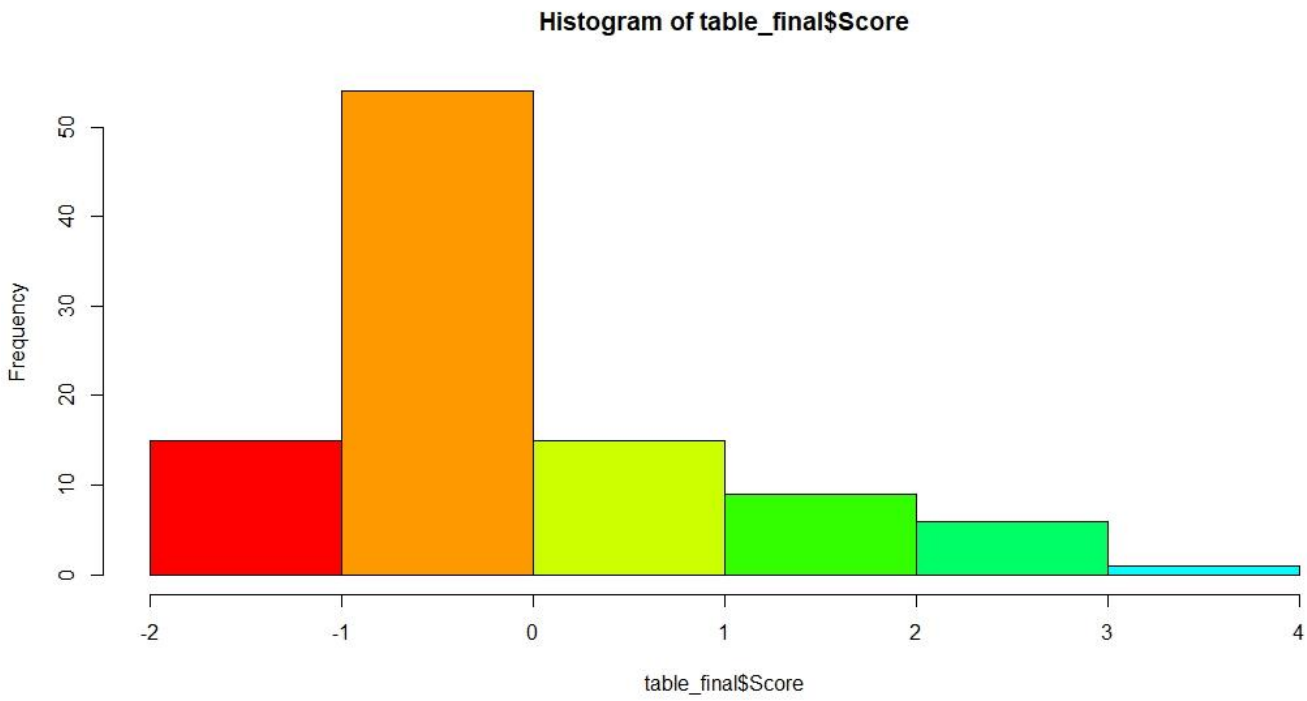


Figure 9 final/_score

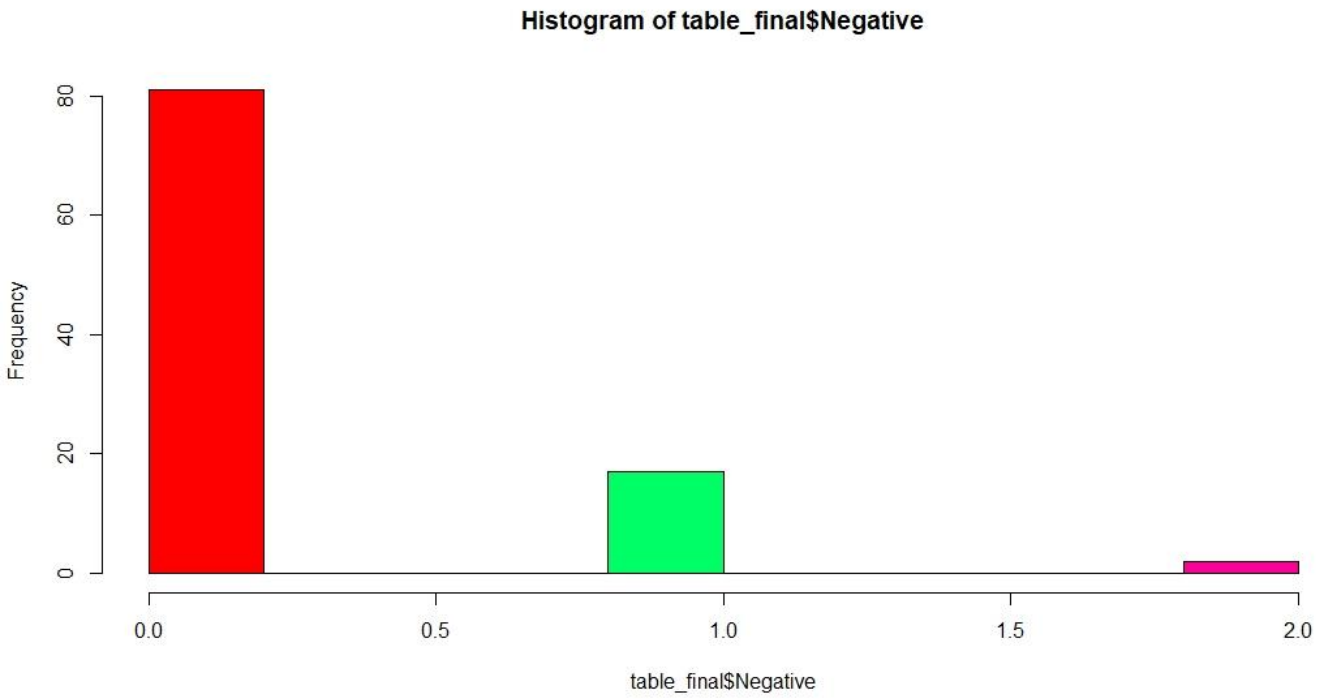


Figure 10 histogram of negative tweets

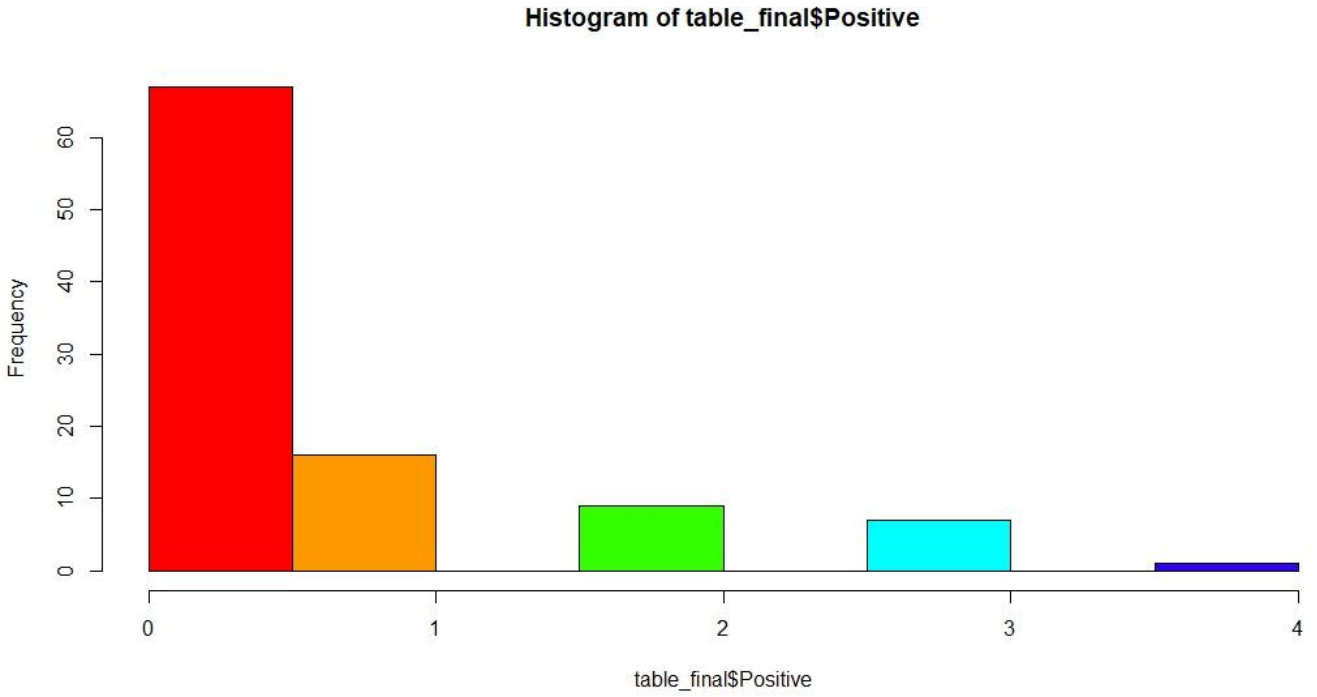


Figure 11 histogram of positive tweets

Pie Chart tab:

Pie chart plot A pie chart is a circular statistical graphic, which is divided into slices to illustrate the sentiment of the hashtag. In a pie chart, the arc length of each slice (and consequently its central angle and area), is proportional to the quantity it represents.

Related Code:

```
slices <- c(sum(table_final$Positive), sum(table_final$Negative))
labels <- c("Positive", "Negative")

library(plotrix)
#pie(slices, labels = labels, col=rainbow(length(labels)), main="Sentiment Analysis")
pie3D(slices, labels = labels, col=rainbow(length(labels)),explode=0.00, main="Sentiment Analysis")
ref_text = sapply(tweets, function(x) x$getText()) #sapply returns a vector
df <- do.call("rbind", lapply(tweets, as.data.frame)) #lapply returns a list
ref_text <- sapply(df$text,function(row) iconv(row, "latin1", "ASCII", sub=""))
str(ref_text) #gives the summary/internal structure of an R object

library(tm) #tm: text mining
ref_corpus <- Corpus(VectorSource(ref_text)) #corpus is a collection of text documents
ref_corpus
inspect(ref_corpus[1])

docs<-ref_corpus
```

Sentiment Analysis Output:

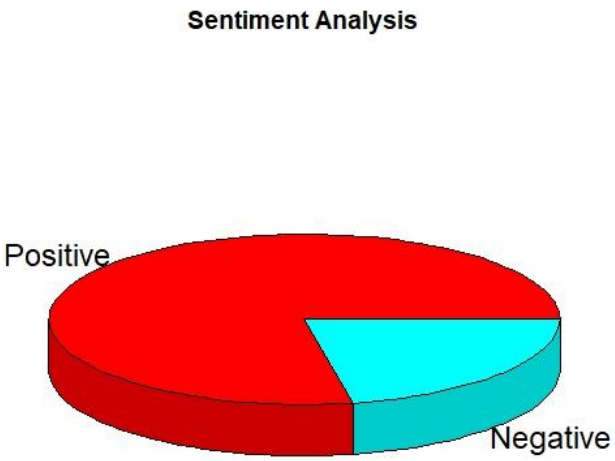


Figure 12 Sentiment analysis

Word Cloud tab

A word cloud is a visual representation of text data, typically used to depict key word meta data (tags) on

websites, or to visualize free form text. This format is useful for quickly perceiving the most prominent terms and for locating a term alphabetically to determine its relative prominence. We have used tmand word cloud package to depict the most used words associated with the hashtag in a pictorial representation under the Word cloud tab.

Related Code:

```
library(wordcloud)
ref_clean <- tm_map(ref_corpus, removePunctuation)
ref_clean <- tm_map(ref_clean, removeWords, stopwords("english"))
ref_clean <- tm_map(ref_clean, removeNumbers)
ref_clean <- tm_map(ref_clean, stripWhitespace)
wordcloud(ref_clean, random.order=F,max.words=80, col=rainbow(50), scale=c(1.5,1))
```

Output:

Top Trending Tweets Today tab: Table

The table is shown which displays the top trending hashtags on Twitter of the location that has been selected.

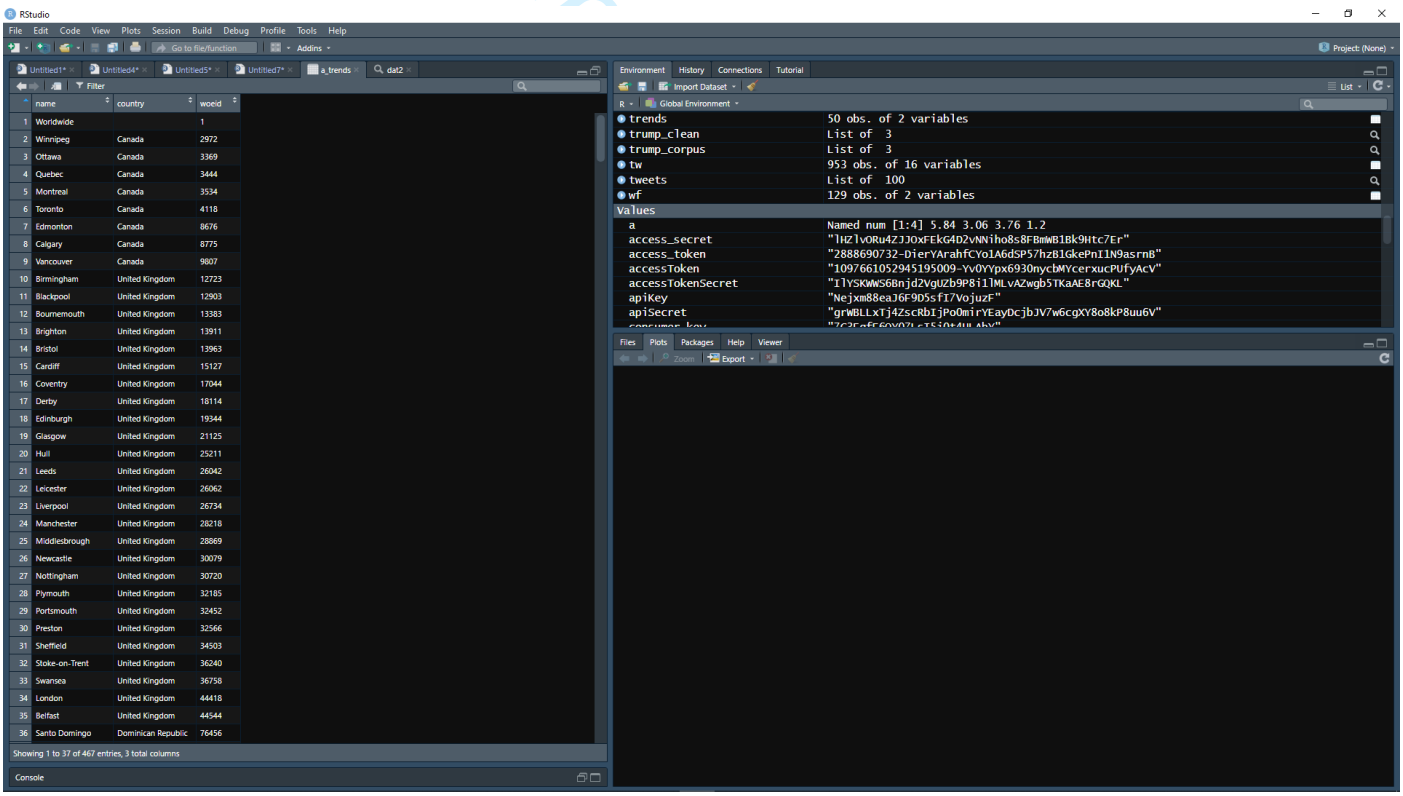
AWOEID (Where on Earth Identifier) isaunique32-bitreferenceidentifier, which is generated, and R uses the WOEID of the selected place to obtain the trending hashtags from that location.

Related Code:

```
a_trends = availableTrendLocations()
```

```
a_trends
View(a_trends)
#woeid = a_trends[which(a_trends$name=="Ottawa"),3] # 3 is serial number of ottawa
locat_trend = getTrends(woeid = 2295414)
trends = locat_trend[1:2]


#To clean data and remove Non English words:
dat <- cbind(trends$name)
dat2 <- unlist(strsplit(dat, split=", "))
dat3 <- grep("dat2", iconv(dat2, "latin1", "ASCII", sub="dat2"))
dat4 <- dat2[-dat3]
dat4
View(dat2)
```

Output:



Top Hashtags of user:ggplot

The ggplot shows the top hashtags of tweeter along with the frequency of each hashtag. This considers the entire user timeline of the tweeter.

Hashtags of user: ggplot

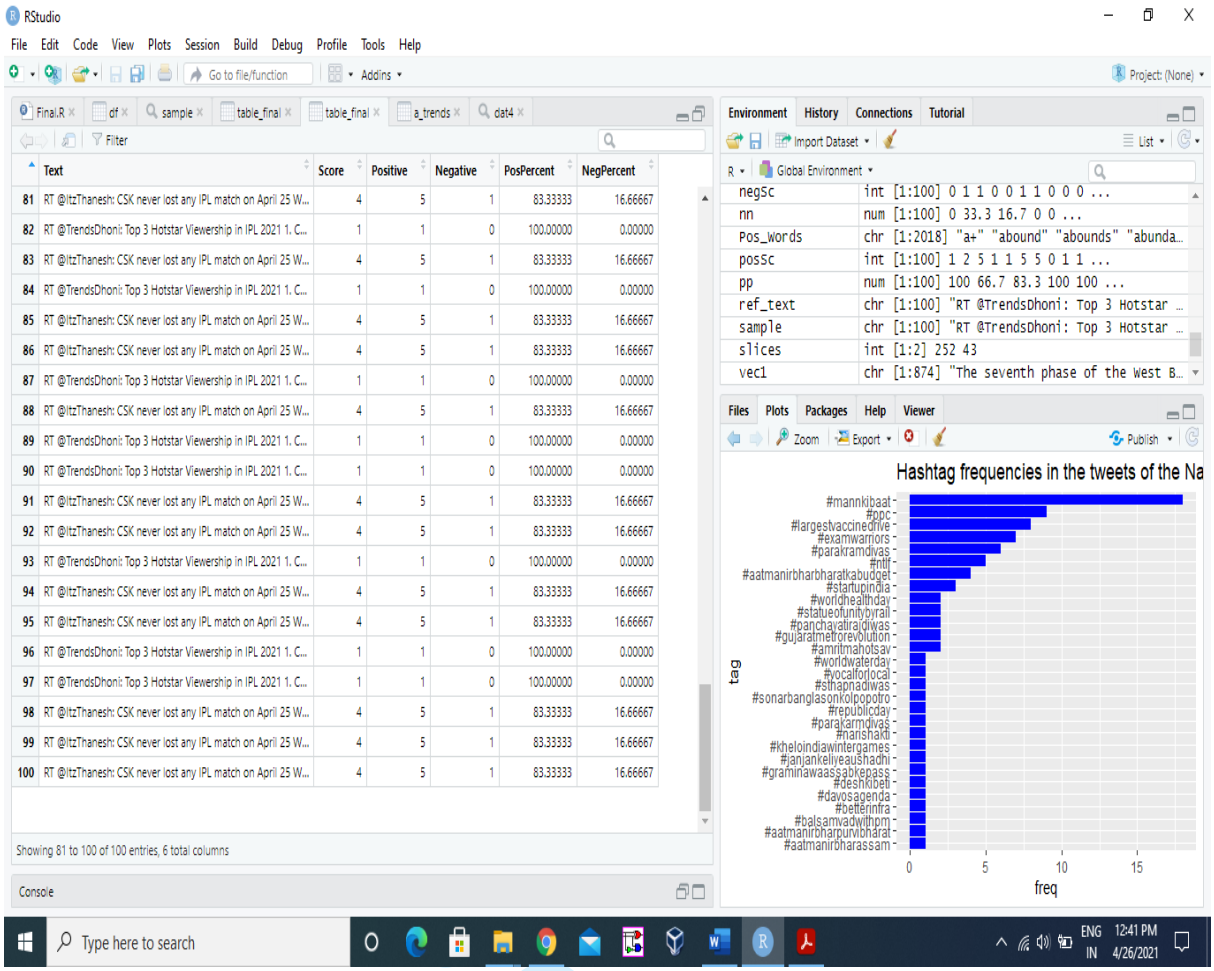


Fig 16

Related Code:

```
library(twitteR)
tw = userTimeline("ipl", n = 1000)
tw = twListToDF(tw) #Covert list to data frame
tw
vec1 = tw$text
vec1
#Extract the hashtags:
hash.pattern = "#[[:alpha:]]+"
have.hash = grep(x = vec1, pattern = hash.pattern) #stores the indices of the tweets which have hashes

hash.matches = gregexpr(pattern = hash.pattern,
                        text = vec1[have.hash]) #Matching hash with hash.pattern
extracted.hash = regmatches(x = vec1[have.hash], m = hash.matches) #the actual hashtags are stored here

df = data.frame(table(tolower(unlist(extracted.hash)))) #dataframe formed with var1(hashtag), freq of hashtag
```

```
colnames(df) = c("tag","freq")
df = df[order(df$freq,decreasing = TRUE),] #for ordering the frequencies
dat = head(df,50)
dat2 = transform(dat,tag = reorder(tag,freq)) #reorder it so that highest freq is at the top
library(ggplot2)
p = ggplot(dat2, aes(x = tag, y = freq)) + geom_bar(stat="identity", fill = "green")+theme_gray()
q=p + coord_flip() + labs(title = "Hashtag frequencies in the tweets")+theme_grey()
q
```

Hashtag Frequency Output

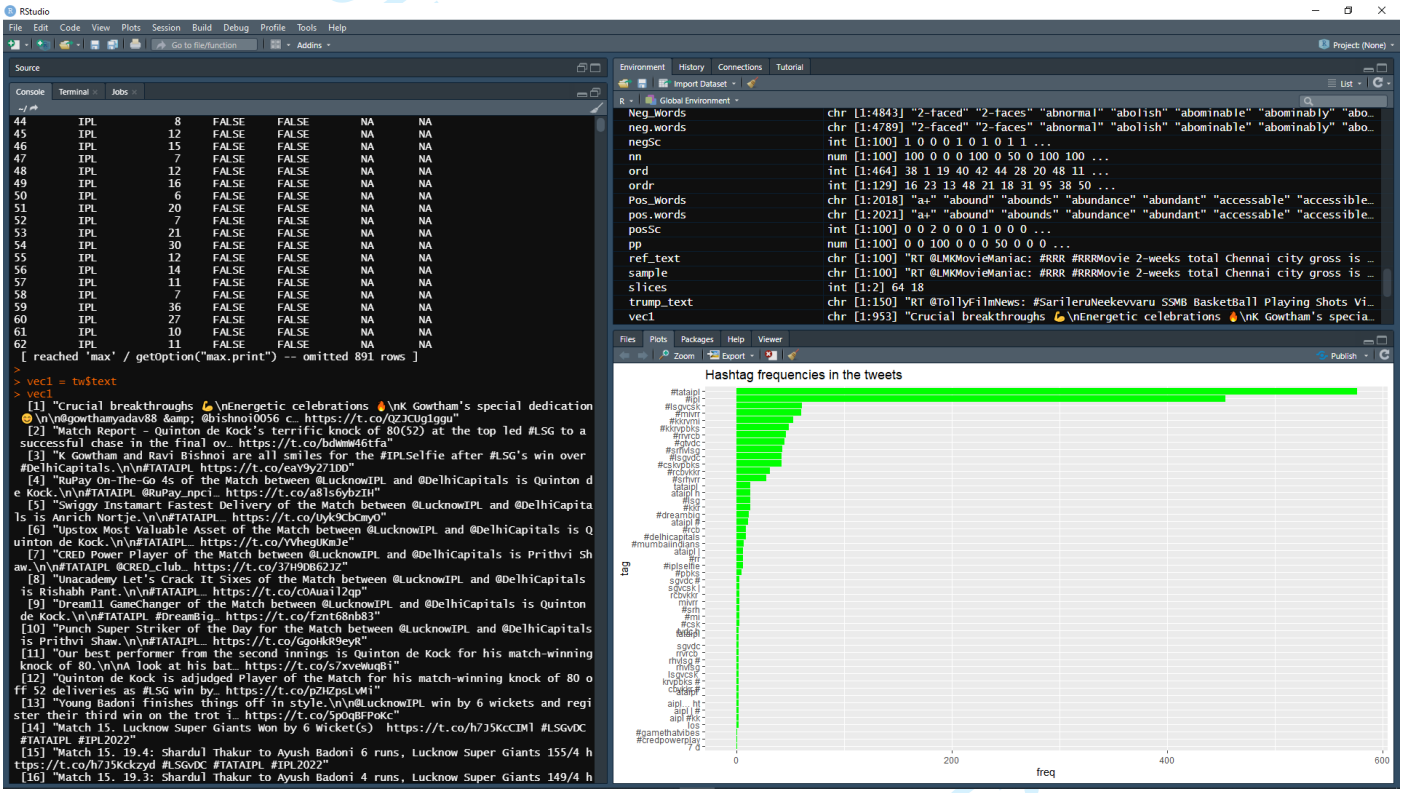


Figure 13 hashtag ouputs

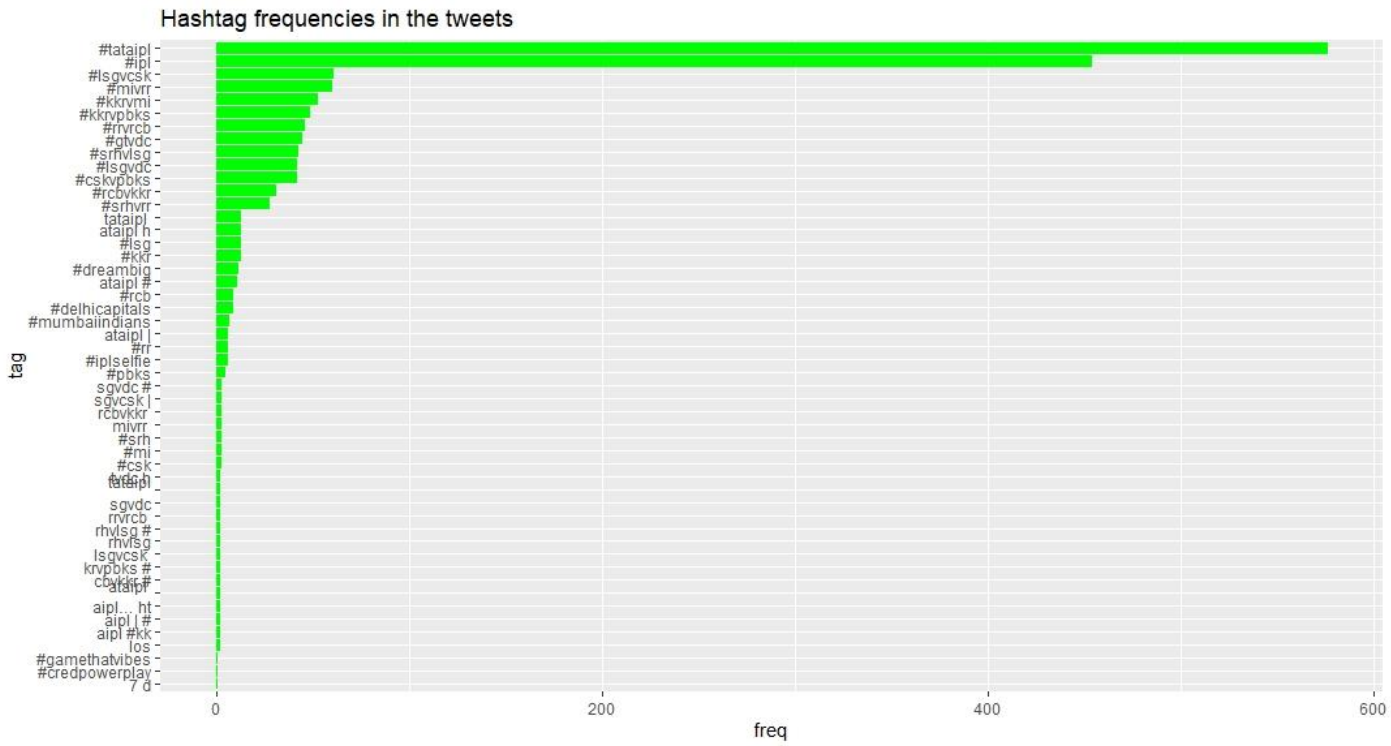


Figure 14 frequency in tweets

11.Limitations

- The Twitter Search API can get tweets up to a maximum of 7 days old.
- Not effective in detecting sarcasm.
- Cannot get 100% efficiency in analysing sentiment oftweets.
- Can only retrieve a maximum of 1000 tweets per query without authenticating viaOAuth before receiving a 403 error or timeout.
- Giving a hash tag under the wrong category will still give results: No error message.