|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **FINAL YEAR MAJOR PROJECT REPORT (PROJECT STAGE – II)** | | | | |
|  | | | | |
| **TWITTER SENTIMENT ANALYZER FOR T20 WORLDCUP** | | | | |
| **TRACK – 1 (RESEARCH AND DEVELOPMENT)** | | | | |
| Submitted in Partial Fulfillment for the Award of Degree of Bachelor of Technology in Computer Science and Engineering from Rajasthan Technical University, Kota | | | | |
|  | | | | |
|  | | | | |
|  | C:\xampp2\htdocs\Works\Pravah2015.tar\public_html\images\skitlogo.png | |  | |
| **COORDINATOR:** | | **SUBMITTED BY:** | | |
| **Mr. Saurabh Ranjan Srivastava**  (Dept. of Computer Science & Engineering) | | **Prashant Kathuria** | | **(12ESKCS080)** |
| **Rajat Gupta** | | **(12ESKCS091)** |
| **MENTOR:** | | **Subroto Biswas** | | **(12ESKCS115)** |
| **Mrs. Neha Janu**  (Dept. of Information Technology) | | **Tushar Mittal** | | **(12ESKCS120)** |
|  | | |
|  | | | | |
| DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING | | | | |
|  | | | | |
| **SWAMI KESHWANAND INSTITUTE OF TECHNOLOGY, MANAGEMENT & GRAMOTHAN**  **Ramnagaria (Jagatpura), Jaipur – 302017** | | | | |
| **SESSION 2015-16** | | | | |
|  | | | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| C:\xampp2\htdocs\Works\Pravah2015.tar\public_html\images\skitlogo.png | **SWAMI KESHWANAND INSTITUTE OF TECHNOLOGY, MANAGEMENT & GRAMOTHAN**  **Ramnagaria (Jagatpura), Jaipur – 302017** | | | | | |
|  | | |  | | |  |
|  | | |
| **CERTIFICATE** | | |
|  | | | | | | |
|  | | | | | | |
|  | | | | | | |
| This is to certify that Final Year Minor Project Report (Project Stage – II) entitled  “**TWITTER SENTIMENT ANALYZER FOR T20 WORLDCUP**” has been duly submitted by | | | | | | |
| * Prashant Kathuria | | | | (12ESKCS080) | | |
| * Rajat Gupta | | | | (12ESKCS091) | | |
| * Subroto Biswas | | | | (12ESKCS115) | | |
| * Tushar Mittal | | | | (12ESKCS120) | | |
| for partial fulfillment of the Degree of Bachelor of Technology of Rajasthan Technical University. It has been found satisfactory and hence approved for submission as Minor Project during academic session 2015-2016. | | | | | | |
|  | | | | | | |
| Date: 10 May 2016 | | | | | | |
|  | | | | | | |
|  | | | | | | |
|  | | | | | | |
| **COORDINATOR:** | | **MENTOR:** | | | **HEAD OF DEPARTMENT:** | |
| **Mr. Saurabh Ranjan Srivastava**  (Dept. of Computer Science & Engineering) | | **Mrs. Neha Janu**  (Dept. of Information Technology) | | | **Prof. Dr. C.M.Choudhary**  (Dept. of Computer Science & Engineering) | |
|  | |  | | |  | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| C:\xampp2\htdocs\Works\Pravah2015.tar\public_html\images\skitlogo.png | | **SWAMI KESHWANAND INSTITUTE OF TECHNOLOGY, MANAGEMENT & GRAMOTHAN**  **Ramnagaria (Jagatpura), Jaipur – 302017** | | | |
|  | | |  | |  |
|  | |
| **ABSTRACT** | |
| The growth in micro-blogging activity on sites over the last few years has been phenomenal. Platforms like Twitter offer an easy outlet for people to express their opinions and companies are increasingly getting interested in capturing these insights about customer behavior and preferences that could help generate more revenues. The staggering amount of data that these sites generate cannot be manually analyzed. Enter thus, Sentiment Analysis, the field where we teach machines to understand human sentiment.  Traditionally sentiment analysis under the umbrella term- ‘text mining’ focuses on larger pieces of text like movie reviews or news articles. On Twitter however, people post 140-character long informal messages called tweets. Analyzing sentiment from these tiny pieces of text is challenging due to their unstructured nature- internet slang, abbreviations, non-conventional spelling and grammar, hashtags, urls and emoticons are just some of the complexities that need to be addressed.  The T20 cricket world cup 2016 is without doubt the biggest sporting event in the world this year. During the 2015 World Cup, millions of fans and viewers from all over the globe used Social Media to share their thoughts and emotions about the games, teams and players and thus created massive amounts of data. Throughout, the tournament we will be analyzing tweets generated during the match. We will try to predict which player is getting more support, popularity of that team as compared to other teams etc. and give a visual representation for the analysis. | | | | | |
| **Domain Descriptors:** | | | |  | |
| 7 CS 1 | Cloud Computing | | |  | |
| 7 CS 3 | Database Mining and Warehousing | | |  | |
| 4 CS 3 | Software Engineering | | |  | |
| **Keywords:** | | | |  | |
| Big Data, Apache Spark, Machine Learning, Elastic Search, AWS, Twitter | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| C:\xampp2\htdocs\Works\Pravah2015.tar\public_html\images\skitlogo.png | | | | **SWAMI KESHWANAND INSTITUTE OF TECHNOLOGY, MANAGEMENT & GRAMOTHAN**  **Ramnagaria (Jagatpura), Jaipur – 302017** | | | | | | | | |
|  | | | | | | |  | | |  | | |
|  | | |
| **DECLARATION** | | |
|  | | | | | | | | | | | | |
| We hereby declare that the report of the project entitled **TWITTER SENTIMENT ANALYZER FOR T20 WORLDCUP** is a record of an original work done by us at **Swami Keshvanand Institute of Technology, Management & Gramothan, Jaipur** under the mentorship of **Mrs. Neha Janu** (Dept. of Information & Technology) and coordination of **Mr. Saurabh Ranjan Srivastava** (Dept. of Computer Science & Technology). This project report has been submitted as the proof of original work for the partial fulfillment of the requirement for the award of the degree of **Bachelor of Technology** (B.Tech) in the **Department of Computer Science & Technology.** It has not been submitted anywhere else, under any other program to the best of our knowledge and belief. | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
|  | | | | | | | | |  | | | |
|  | | | | | | | | |  | | | |
| **Team Members:** | | | | | | | | | **Signatures:** | | | |
| (12ESKCS080) | | | Prashant Kathuria | | | | | | | | | |
| (12ESKCS091) | | | Rajat Gupta | | | | | | | | | |
| (12ESKCS115) | | | Subroto Biswas | | | | | | | | | |
| (12ESKCS120) | | | Tushar Mittal | | | | | | | | | |
|  | | | | | | | | |  | | | |
|  | | | | | | | | | | | | |
|  | | |  | | | | | | | | | | |
|  | | | | | **ACKNOWLEDGMENT** | | | | | |  | | |
|  | | | | | |
|  | | | | | |
| A project of such a vast coverage cannot be realized without help from numerous sources and people in the organization. We take this opportunity to express our gratitude to all those who have been helping us in making this project successful. | | | | | | | | | | | | | |
| We are highly indebted to our faculty mentor **Mrs. Neha Janu.** She has been a guide, motivator & source of inspiration for us to carry out the necessary proceedings for the project to be completed successfully. We also thank our project coordinator **Mr. Saurabh Ranjan Srrivastava** for his co-operation, encouragement, valuable suggestions and critical remarks that galvanized our efforts in the right direction. | | | | | | | | | | | | | |
| We would also like to convey our sincere thanks to **Prof. C.M. Choudhary,** HOD, Department of Computer Science & Engineering, for facilitating, motivating and **s**upporting us during each phase of development of the project. Also, we pay our sincere gratitude to all the **Faculty Members** of Swami Keshvanand Institute of Technology, Management & Gramothan, Jaipur and all our Colleagues for their co-operation and support. | | | | | | | | | | | | | |
| Last but not least we would like to thank all those who have directly or indirectly helped and cooperated in accomplishing this project. | | | | | | | | | | | | | |
|  | | | | | | | |  | | | | | |
| **Team Members:** | | | | | | | |  | | | | | |
| (12ESKCS080) | | | Prashant Kathuria | | | | | | | | | | |
| (12ESKCS091) | | | Rajat Gupta | | | | | | | | | | |
| (12ESKCS115) | | | Subroto Biswas | | | | | | | | | | |
| (12ESKCS120) | | | Tushar Mittal | | | | | | | | | | |
|  | | | | | | | |  | | | | | |
| |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | |  |  | | | | **INDEX** |  |  | | **UNIT** | **DESCRIPTION** | | | |  | | **PAGE No.** | | I. | TITLE PAGE | | | | | |  | | II. | CERTIFICATE | | | | | | 1 | | III. | ABSTRACT | | | | | | 2 | | IV. | DECLARATION | | | | | | 3 | | V. | ACKNOWLEDGEMENT | | | | | | 4 | | 1. | PROJECT CHARTER | | | | | | 9-14 | |  | 1.1 | Problem Statement & Objective | | | | | 9 | |  | 1.2 | Literature Survey | | | | | 9-11 | |  | 1.3 | Introduction to Project | | | | | 11 | |  | 1.4 | Proposed Algorithm | | | | | 12-13 | |  | 1.5 | Scope of the Project | | | | | 14 | | 2. | SYSTEM REQUIREMENT SPECIFICATION | | | | | | 15-20 | |  | 2.1 | Overall Description | | | | | 15 | |  |  | 2.1.1 | Product Perspective | | | | 15 | |  |  |  | 2.1.1.1 | System Interfaces | | | 15 | |  |  |  | 2.1.1.2 | User Interfaces | | | 15 | |  |  |  | 2.1.1.3 | Hardware Interfaces | | | 15-17 | |  |  |  | 2.1.1.4 | Software Interfaces | | | 17 | |  |  |  | 2.1.1.5 | Communiations Interfaces | | | 17 | |  |  |  | 2.1.1.6 | Memory Constraints | | | 18 | |  |  |  | 2.1.1.7 | Operations | | | 18 | |  |  |  | 2.1.1.8 | Site Adaptations Requirements | | | 18 | |  |  | 2.1.2 | Project Functions | | | | 18-19 | |  |  | 2.1.3 | User Characteristics | | | | 19 | |  |  | 2.1.4 | Constraints | | | | 19 | |  |  | 2.1.5 | Assumptions & Dependencies | | | | 19 | |  | 2.2 | Specific Requirements | | | | | 19 | |  |  | 2.2.1 | User Interface Requirements | | | | 20 | |  |  | 2.2.2 | System Product Features | | | | 20 | |  |  |  | 2.2.2.1 | Security | | | 20 | |  |  |  | 2.2.2.2 | Maintainability | | | 20 | |  |  |  | 2.2.2.3 | Portability | | | 20 | | 3 | SYSTEM DESIGN SPECIFICATION | | | | | | 21-31 | |  | 3.1 | System Architecture | | | | | 21 | |  | 3.2 | Module Decomposition Description | | | | | 21-27 | |  | 3.3 | High Level Design Diagrams | | | | | 28 | |  |  | 3.3.1 | Usecase Diagram | | | | 28 | |  |  | 3.3.2 | Sequence Diagram | | | | 29 | |  |  | 3.3.3 | Data-Flow Diagram | | | | 30 | |  |  | 3.3.4 | PERT Chart | | | | 31 | | 4. | METHODOLOGY & TEAM | | | | | | 32-34 | |  | 4.1 | Introduction to Waterfall Framework | | | | | 32-34 | |  | 4.2 | Team Members, Roles & Responsibilities | | | | | 34 | | 5. | SYSTEM TESTING | | | | | | 35-36 | |  | 5.1 | Functionality Testing | | | | | 35 | |  | 5.2 | Performance Testing | | | | | 35 | |  | 5.3 | Usability Testing | | | | | 36 | |  | 5.4 | Server Side Interfacing | | | | | 36 | |  | 5.5 | Client Side Compatibility | | | | | 36 | | 6. | TEST EXECUTION SUMMARY | | | | | | 37 | | 7. | PROJECT SCREENSHOTS | | | | | | 38-41 | | 8. | PROJECT SUMMARY AND CONCLUSIONS | | | | | | 42 | | 9. | FUTURE SCOPE | | | | | | 43 | | 10. | REFERENCES | | | | | | 44 | | 11. | PROJECT PAPER | | | | | | 45 | | | | | | | | | | | | | | |
|  | **INDEX OF FIGURES** | | | | | | | | | |  | | |
| **S.No** | **DESCRIPTION** | | | |  | | | | | | **PAGE No.** | | |
|  | PROCEDRAL FLOW CHART | | | | | | | | | | 13 | | |
|  | PIE-CHART SHOWING THE DISTRIBUTION OF SENTIMENTS ANALYSED | | | | | | | | | | 25 | | |
|  | GRAPH SHOWING MAXIMUM NUMBER OF TWEETS COLLECTED AT A PARTICULAR INSTANT | | | | | | | | | | 26 | | |
|  | GRAPH SHOWING THE USER SUPPORT ANALYSED FOR A PARTICULAR TEAM | | | | | | | | | | 26 | | |
|  | GRAPH SHOWING THE RANGE OF FOLOWERS OF THE USER WHO TWEETED (HENCE POPULARITY) | | | | | | | | | | 27 | | |
|  | GRAPH SHOWING THE LOCATION OF TWEETS ORIGINATION | | | | | | | | | | 27 | | |
|  | USECASE DIAGRAM | | | | | | | | | | 28 | | |
|  | SEQUENCE DIAGRAM | | | | | | | | | | 29 | | |
|  | LEVEL 0 DATAFLOW DIAGRAM | | | | | | | | | | 30 | | |
|  | P.E.R.T CHART | | | | | | | | | | 31 | | |
|  | BASIC WATERFALL MODEL | | | | | | | | | | 32 | | |
|  | APACHE SPARK CLUSTER RUNNING INTERFACE | | | | | | | | | | 38 | | |
|  | ELASTIC SEARCH PLUGIN SHOWING ALL DATA | | | | | | | | | | 39 | | |
|  | ELASTIC SEARCH PLUGIN SHOWING DATA OF A PARTICULAR TABLE | | | | | | | | | | 39 | | |
|  | KIBANA SHOWING THE DATA IMPORTED | | | | | | | | | | 40 | | |
|  | KIBANA’S INTERFACE TO IMPORT NEW DATA | | | | | | | | | | 40 | | |
|  | KIBANA’S INTERFACE TO CREATE A NEW VISUALISATION | | | | | | | | | | 41 | | |

|  |  |  |  |
| --- | --- | --- | --- |
|  | **INDEX OF TABLES** | |  |
| **S.No** | **DESCRIPTION** |  | **PAGE No.** |
|  | MINIMUM CLIENT SIDE HARDWARE INTERFACES | | 16 |
|  | MINIMUM SERVER SIDE HARDWARE INTERFACES | | 16 |
|  | RECOMMENDED CLIENT SIDE HARDWARE INTERFACES | | 16 |
|  | RECOMMENDED SERVER SIDE HARDWARE INTERFACES | | 17 |
|  | MINIMUM SOFTWARE INTERFACES | | 17 |
|  | AWS RESOURCE SPECIFICATION | | 24 |
|  | ROLES AND RESPONSIBILITIES | | 34 |

|  |  |  |
| --- | --- | --- |
| **ABBREVIATIONS USED** | | |
|  | MLLib | Machine Learning library |
|  | DBMS | Database Management System |
|  | OS | Operaiting System |
|  | NLP | Nuero Linguistic Programming |
|  | DMW | Data Mining & Warehousing |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **UNIT – 1** | | | | | | | **PROJECT CHARTER** |
|  | | | | | | | |
| **1.1** | **Problem Statement** **& Objective** | | | | | | |
|  | | | | | | | |
| **Problem:** To Analyze the support of a cricket team based on related tweets  **Reasons:** There are several reasons to support this analysis   * Twitter is one of the most important social media platform where a majority of people give their views about a particular topic , here Cricket. * Marketing and investment firms need a specific platform to advertise and invest and hence must have a good knowledge about people’s perspective . * Teams need to understand the mass perspective in order to strategize their game and choose players accordingly.   **Solution:** Keeping in mind these situations and to develop marketing and business investment strategies Twitter Sentiment Analyzer is thought to be a solution, in conjunction with the large number of people using Twitter.  So the basic idea behind this application is to know about the popularity of a particular team or player. This analysis would help the companies strategize marketing policies as to what product should be advertised through which team, Which player would be the best fit for endorsing a particular product relative to the product popularity in the specific location.  This analysis would also help the team management understand the people’s perspective and strategize accordingly as to what would be a good at which batting order. | | | | | | | |
|  | | | | | | | |
| **Objective:** To create a big data application to find the support of a team for T20 worldcup. A lot of tweets are tweeted during T20 world cup matches. This vast amount of data is used to train the system to analyze a tweet at real time and thus find its polarity i.e. positive, negative or neutral. This system provides facility to the user to analyze the match and find the support of a particular team. | | | | | | | |
|  | | | | | | | |
| **1.2** | **Literature Survey** | | | | | | |
|  | | | | | | | |
| The motivation for this research is taken from recent advancement in the field of sentiment analtsis usig natural language processing and various machine learning algorithms such as Naïve Bayes model, SVM classifier etc. The methodology is derived from previous studies which model the tweets and label them with a sentiment according to the training data | | | | | | | |
|  | | | | | | | |
| There have been many papers written on sentiment analysis for the domain of blogs and product reviews. (Pang and Lee 2008) gives a survey of sentiment analysis. Researchers have also analyzed the brand impact of microblogging (Jansen). We could not find any papers that analyzes machine learning techniques in the specific domain of cricket, probably because the popularity of Twitter is very recent. | | | | | | | |
|  | | | | | | | |
| In recent years, opinion mining, aka sentiment analysis, attracted a lot of interest and has been studied by many researchers. In their early work, Hatzivassiloglou and McKeown reported that it is possible to identify sentiment words (adjectives) and their polarity in sentences with a high accuracy of 82%. Following this finding, various sentiment analysis algorithms have been proposed. For example, Turney introduced one of the first algorithms for document level sentiment analysis, which achieved an average accuracy of 74% for product reviews; but on movie reviews the performance was much worse only 66%. | | | | | | | |
|  | | | | | | | |
| In his design, rather than focusing on isolated adjectives, Turney proposed to detect sentiments based on selected phrases, which are chosen via a number of Part-Of-Speech (POS) patterns. Generally speaking, POS information is frequently exploited in sentiment analysis systems. In particular, POS tagging helps with the word sense disambiguation problem and provides the ability to better understand the surrounding context. For another example, Cambria et al. proposed Sentic Computing which explores the usage of Common Sense Computing to significantly enhance computers' emotional intelligence, i.e., their capability of perceiving and expressing emotions | | | | | | | |
|  | | | | | | | |
| As it stands, the design of sentiment analysis systems could be divided into two schools | a lexicon-based approach and a learning-based approach Pang et al have evaluated and compared several different supervised machine learning algorithms for classifying the sentiments of movie reviews. The learning algorithms they used include Naïve Bayes (NB), Maximum Entropy (ME), and Support Vector Machine (SVM), with SVM slightly outperforming other learning algorithms. In their early work [15], they achieved 82.9% accuracy with a relatively simple design using SVM trained on bag-of-words (unigram) features; this was further increased in their later work to 87.2%. This performance increase was achieved by employing graph min-cut based subjectivity detection before the classification step, thus removing objective text from the final sentiment classification. However, as other researchers have found, such a simple design, solely based on supervised machine learning, sufiers from style, domain, or even time dependencies. Furthermore, it only provides an overall sentiment score for each review without any further explanation or justification. People often express multiple opinions in their reviews, therefore by just detecting that a given review is positive or negative we cannot obtain much knowledge about which speci\_c aspects (e.g., product features) people liked or disliked and to what degree. To address these concerns, researchers have proposed various sentence-level opinion mining techniques. | | | | | | | |
|  | | | | | | | |
| A recent hot topic in opinion mining is domain dependency. As Owsleys et al. have found, to achieve good sentiment analysis results you have to build a domain specific lexicon that is related to both the entities and their sentiment expressions. Particularly, in different domains, the same word could have completely opposite meanings or very different sentiment strengths. To build domain-specific lexicons, researchers have proposed various techniques. One common approach is to start from a small initial sentiment lexicon and gradually expand it during the processing of reviews. | | | | | | | |
|  | | | | | | | |
| Some researchers have successfully utilised WordNet for the construction of sentiment lexcions where Word- Net provides initial seed information for the sentiment lexicon which will then be expanded using a sentiment corpus .However, our experience is that WordNet is not a very reliable source to build sentiment lexicons, since it introduces too much noise. Furthermore, it is worthy to mention that their method does not adjust the sentiment value for each sentiment word in the lexicon it merely expands the lexicon with previously unknown sentiment words. Another common approach is bootstrapping. For example, Rilofi and Wiebe employed a classifier to extract subjective patterns from text which could be used to build a sentiment lexicon. | | | | | | | |
|  | | | | | | | |
| For text mining, Fang et al. proposed the contrastive opinion modeling, which models the opinions of the individual perspective on the topic. Likewise, our work also focuses on a particular topic and opinion. However, our work and the work of Fang et al. have different purposes. Fang’s work aims at finding subjective information. In contrast, our work aims at analyzing the information. | | | | | | | |
|  | | | | | | | |
|  | | | | | | | |
| **1.3** | **Introduction to Project** | | | | | | |
| A Sentiment analyser classifies a particular text into categories based on emotion and opinion. This technology has many interesting consequences not just for businesses but to countries as a whole .For businesses, online opinion has turned into a kind of virtual currency that can make or break a product in the marketplace. On the other hand the aggregate of emotions can very effectively measure the temperature of a country in response to real world events like a mass increase in the level of worriedness around major weather phenomena a mass increase in the level of distress and sadness after terror attacks the same with political changes and so on. | | | | | | | |
|  | | | | | | | |
| As sentiment analysis tools begin to take shape, they not only help businesses improve their bottom lines, but also eventually transform the experience of searching for information online. The simplest algorithms work by scanning keywords to categorize a statement as positive or negative, based on a simple binary analysis love evaluates to good, while hate to bad). But that approach fails to capture the subtleties that bring human language to life: irony, sarcasm, slang and other idiomatic expressions. Reliable sentiment analysis requires parsing many linguistic shades of gray. For example, several adjectives often signal a high degree of subjectivity, while noun- and verb-heavy statements tend toward a more neutral point of view view. As sentiment analysis algorithms become more sophisticated, they begin to yield more accurate results that may eventually point the way to more sophisticated filtering mechanisms. It gives its users the ability to research any topic on blogs, social media sites, and in traditional news media reports. The rise of blogs and social networks has also affected the bull market (the stock exchanges) by reviews, ratings, recommendations and other forms of online expression. For computer scientists, this fastgrowing mountain of data is opening a tantalizing window onto the collective consciousness of Internet users. | | | | | | | |
|  | | | | | | | |
| **1.4** | **Proposed Logic or Algorithm** | | | | | | |
| Our concept-level sentiment analysis system, our algo, is developed by combining lexicon-based and learning-based approaches. As shown in Figure, the supervised machine learning component is not just responsible for small tasks such as adjusting sentiment values or finding more sentiment words, but is actually responsible for evaluating all the ingredients of the sentiment system, including semantic rules used to derive the final output. The final component in Algo measures and reports the overall sentiment of a given opinionated text, such as a customer review, as a realvalued score between -1 and +1, which can then be easily transformed into a positive/negative classification or into a scale of 1-5 stars.  The main advantage of our hybrid approach using a lexicon/ learning symbiosis, is to attain the best of both worlds stability as well as readability from a carefully designed lexicon, and the high accuracy from a powerful supervised learning algorithm. Due to the built-in sentiment lexicon and linguistic rules, this algo can detect and measure sentiments at the concept level, providing structured and readable aspect-oriented outputs. Furthermore, algo is less sensitive to changes in topic domain or writing style. The system can even be extended after it has already been trained, by introducing new linguistic rules or expanding the sentiment lexicon at any time, so as to further improve the system's performance. | | | | | | | |
| We start of by regulating a twitter stream of relevant information related to the T-20 Cricket World cup help from 16 March 2016 to 3 April 2016. We collected results from the important matches and the fileds required for some meaningful analysis. The tweets and other information in the stream is not still properly formatted for human interpretation so we had to perform pre-processing on the information obtained. The tweets are known to be in a language which is fast for typing but differs from the normal language convention and uses lingos for shortning of the message as well as the tweets may contain many emoticons and other such data. To parse such a variety of information the pre-processing step was used. In this we separated/removed data which was not relevant to the discussion and transformed improper data to a useful form which could be easily trained as well as tested and analysed later on.  C:\Users\Prashant\Downloads\a.png  **Figure 1- Procedral flow chart**  After we have obtained a filter tweets text and extracted it without any ambiguous content we pass it into our analytical engine where we process it into a naïve bayes classifier as well as a NLP system. We combine there scores to obtain a final label for our tweet. | | | | | | | |
| The tweets with matching score greater than some pre-specified threshold are considered to represent this tweet’s label or analysed sentiment. A label implies what opinion the tweet contained or the sentiment analysed. We have used five different label for this process namely Extremely Negative, Moderately Negative,Neutral,Moderately Positive and Extremely Positive. | | | | | | | |
|  | | | | | | | |
| **1.5** | **Scope of the Project** | | | | | | |
| The project entitled **“Twitter Sentiment Analysis during T20 World Cup”** will provide user to analyse the sentiment for a particular event in the field of cricket. The user can observe and visualize the sentiment which was analysed, the support to teams, and other useful information like the source of the tweets, the language it was in, the device used to tweet and so on.   * The system provides user to analyse tweets on the basis of any event. * The system passes the tweet collected via the stream first through a preprocessing unit. * The system then passes the filtered tweet to the analytical engine. * The system lets the user know about the sentiment of the tweet analysed. * The system shows visualization of any particular event or a part of an event.   The features that are described in this document are used in the future phases of the software development cycle. The features described here meet the needs of all the users. The success criteria for the system are based in the level up to which the features described in this document are implemented in the system. | | | | | | | |
| **UNIT – 2** | | | | **SYSTEM REQUIREMENT SPECIFICATION** | | | |
|  | | | | | | | |
| **2.1** | **Overall Description** | | | | | | |
| This section and its subsections contain the description of the project compoents such as interfaces, performance requirements, design constraints, assumptions and dependencies etc. | | | | | | | |
| **2.1.1** | **Product Perspective** | | | | | | |
|  | The application will be a Windows / Linux based, self contained and independent product. | | | | | | |
|  | | | | | | | |
| **2.1.1.1** | **System Interfaces** | | | | | | |
|  | List each system interface and identify the functionality of the system (hardware and software both) to accomplish the system requirement and interface description to match the system. | | | | | | |
|  |  | | | | | | |
| **2.1.1.2** | **User Interfaces** | | | | | | |
|  | The application will have a user friendly interface. Following output will be provided:   1. A blog which will display the analysis using the representation tool Kibana. 2. The blog would contain piecharts and graphs in order to display the analysis which would present the support of a particular team, the most mentioned player, device operating system from which tweet was made, location of polarized tweets in groups of positive, negative and neutral etc. | | | | | | |
| **2.1.1.3** | **Hardware Interfaces** | | | | | | |
|  | * Screen resolution of at least 800 x 600 pixels is required for proper and complete viewing of screens. Higher resolutions in wide-screen mode will be better for a better view. * A network connection (internet / intranet) is required to make the web service accessible on other systems connected over the network. * Other hardware interface specifications are as follows  |  |  |  | | --- | --- | --- | | **HARDWARE INTERFACES - CLIENT SIDE (Minimum)** | | | | **HARDWARE** | **RAM** | **DISK SPACE** | | Intel Pentium 4 and Higher Processor | 256 MB | 2 GB | | QWERTY Keyboard (U.S. Design) | | USB 2.0 / PS2 Mouse | | **Table 1 – Minimum Client Side Hardware Interfaces** | | |  |  |  |  | | --- | --- | --- | | **HARDWARE INTERFACES - SERVER SIDE (Minimum)** | | | | **HARDWARE** | **RAM** | **DISK SPACE** | | Quad core processor and higher | 4096 MB | 15 GB | | **Table 2 – Minimum Server Side Hardware Interfaces** | | | | | | | | | |
|  | |  |  |  | | --- | --- | --- | | **HARDWARE INTERFACES - CLIENT SIDE (Recommended)** | | | | **HARDWARE** | **RAM** | **DISK SPACE** | | Intel Core i3 / i5 / i7 2.27 GHz and higher  Or  AMD 4XXX and higher | 1024 MB | 2 GB | | QWERTY Keyboard (U.S. Design) | | USB 2.0 Optical Mouse | | **Table 3 – Recommended Client Side Hardware Interfaces** | | |  |  |  |  | | --- | --- | --- | | **HARDWARE INTERFACES - SERVER SIDE (Recommended)** | | | | **HARDWARE** | **RAM** | **DISK SPACE** | | Octa core processor or higher | 8192 MB | 30 GB | | **Table 4 – Recommended Server Side Hardware Interfaces** | | | | | | | | | |
| **2.1.1.4** | **Software Interfaces** | | | | | | |
|  | |  |  |  | | --- | --- | --- | | **SOFTWARE INTERFACES (Minimum)** | | | | **Software Tool** | **Version** | **Purpose of Use** | | Analysis Tool | Apache Spark 1.6.0 | Analysis and operational platform | | Visualization Tool | Kibana 4.4.1 | Representation and visualizing of data as graphs, pie-charts etc | | Machine Learning package | Core NLP 3.6.0 | Package containing methods to implement machine learning algorithms | | ProgrammiBuild Tool | Scala Build Tool 0.13.11 | Programming tool for coding using Scala | | Database | Elastic Search 2.2.2 | Unstructured database which is very fast in data retrieval | | **Table 5 – Minimum Software Interfaces** | | | | | | | | | |
| **2.1.1.5** | **Communication Interfaces** | | | | | | |
|  | * Client (customer) on Internet will be using HTTP/HTTPS protocol. * Client (system user) on Internet will be using HTTP/HTTPS protocol. | | | | | | |
| **2.1.1.6** | **Memory Constraints** | | | | | | |
|  | * At least 256 MB of RAM and 2 GB of space on hard disk will be required for running the application on client end. * Similarily, a minimum of 4096 MB of RAM and 15 GB of space on hard disk will be required for running the application on server end. | | | | | | |
|  |  | | | | | | |
| **2.1.1.7** | **Operations** | | | | | | |
|  | * This machine is trained with manually polarized tweets so that the system automates the analysis process at real time. * Libraries consisting of team names and their players are manually built used for preprocessing. * This fetched tweet is preprocessed and then polarized using NLP. * The data is hosted at the AWS cloud which consists of Elastic Search Database in which tweets are streamed through Twitter API stream. * The blog consists of visual represetations using Kibana Tool. * The number of tweets can be checked at the AWS console where the instance only runs for 4 hours during the match. | | | | | | |
|  |  | | | | | | |
| **2.1.1.8** | **Site Adaption Requirements** | | | | | | |
|  | The computing terminals conneted to network (internet / intranet) at the client end will be required to support the hardware and software interfaces specified in above sections. | | | | | | |
|  | | | | | | | |
|  | | | | | | | |
| **2.1.2** | **Project Functions** | | | | | | |
|  | The system will allow access to all users having access to the blog.  A summary of the major functions that the software will perform:   1. Gather tweets at real time and store in the elastic search. 2. Gathered tweets are preprocessed. 3. Machine is trained for processing new tweets according to the training data. 4. The polarized tweets are then used to find the required support. 5. This analysis is presented using graphs and pie-charts using Kibana. | | | | | | |
|  | | | | | | | |
| **2.1.3** | **User Characteristics** | | | | | | |
|  | * *Educational Level:* User should be at least graduate and comfortable with English. * *Experience:* User should be well versed / informed about the purpose of the analysis and read graphs and pie-charts effectively. * *Technical Expertise:* User should be comfortable in browsing web on a computer. | | | | | | |
|  | | | | | | | |
| **2.1.4** | **Constraints** | | | | | | |
|  | * Since the cloud service for data streaming, AWS is being used in this project, so the efficient working of the AWS server is a major factor for data streaming at real time and storing as analysis can only be done if data is available. * As very large amount of tweets are streamed during a match so the tweets are only streamed for 4 hours during the match. * Most of the time is comsumed by the preprocessing step of the analysis. * Changes in team and team players leads to faulty analysis as the training is done on the old information about the teams. | | | | | | |
| **2.1.5** | **Assumptions & Dependencies** | | | | | | |
|  | * Training data is made according to the latest information. * AWS server works without any crash. * The team names, player names etc do not change abruptly. * The main aspect of the project i.e. data depends efficient working of Twitter server. * The polarity of tweet can only be positive, negative or neutral. | | | | | | |
|  |  | | | | | | |
| **2.2** | **Specific Requirements** | | | | | | |
| This section presents the software requirements to a level of detail sufficiency to enable designers to design and testers to test the system. | | | | | | | |
| **2.2.1** | **User Interface Requirements** | | | | | | |
|  | **Static Blog:**  A blog can be accessed using a web browser assuming internet connectivity at client side which shows graphs and pie-charts one after another on a single web page. These graphs and pie-charts show the support or percentage whenever mouse pointer is hovered over the pie-chart sector. | | | | | | |
|  |  | | | | | | |
| **2.2.2** | **System Product Features** | | | | | | |
| **2.2.2.1** | **Security** | | | | | | |
|  | The analysis can be viewed by anyone on the blog. But users can’t access the raw data stored at AWS cloud which is password protected. | | | | | | |
|  |  | | | | | | |
| **2.2.2.2** | **Maintainability** | | | | | | |
|  | The application will be designed in a manner to make it easy to incorporate the required types of analysis as required by a user. Elastic Search and Spark helps in dealing with big data and hosting it on AWS server makes the application reliable. | | | | | | |
|  |  | | | | | | |
| **2.2.2.3** | **Portabilty** | | | | | | |
|  | The application can be hosted on any server and hence is quite portable but a good internet connection is required. | | | | | | |
| **UNIT– 3** | | **SYSTEM DESIGN SPECIFICATION** | | | | | |
|  | |  | | | | | |
| **3.1** | **System Architecture** | | | | | | |
|  | System architecture presents the schematic view of the complete system along with its major components and their connectivities. The overall architecture of the proposed system will be as follows. | | | | | | |
|  | | | | | | | |
| **3.2** | **Module Decomposition** | | | | | | |
|  | The proposed system can be decomposed into following major modules :   1. Requirement Analysis and Algorithm Proposition 2. Data Gathering and Preprocessing 3. AWS cloud and Elastic Search setup and configuration 4. Apache Spark and NLP implementation 5. Final Analysis and result visualization | | | | | | |
|  | **Module - 1: Requirement Analysis and Algorithm proposition** | | | | | | |
|  | **Requirement analysis** requires that what kind of softwares and resources will be required to implement the project.  A switch was made from Python to Scala as Spark is built on Scala and hence offers faster performance as compared to Python.  A scalable processor was needed which could process very large amount of data and so the AWS cloud was used which helped us to scale the processing relative to data amount and used the Elastic Search database for storing streamed data as it is unstructured , has low latency and gives fast retrieval.  Apache Spark is used as in contrast to Hadoop's two-stage disk-based MapReduce paradigm, Spark's multi-stage in-memory primitives provides performance up to 100 times faster for certain applications.By allowing user programs to load data into a cluster's memory and query it repeatedly, Spark is well-suited to machine learning algorithms. Spark enables applications in Hadoop clusters to run up to 100x faster in memory, and 10x faster even when running on disk.  MLlib is Spark’s machine learning (ML) library. Its goal is to make practical machine learning scalable and easy. It consists of common learning algorithms and utilities, including classification, regression, clustering, collaborative filtering, dimensionality reduction, as well as lower-level optimization primitives and higher-level pipeline APIs.  Kibana is an open source data visualization plugin for Elasticsearch. It provides visualization capabilities on top of the content indexed on an Elasticsearch cluster. Users can create bar, line and scatter plots, or pie charts and maps on top of large volumes of data.  Core NLP NLP stands for **Neuro-Linguistic Programming,** a name that encompasses the three most influential components involved in producing human experience: *neurology, language* and *programming*. The neurological system regulates how our bodies function, language determines how we interface and communicate with other people and our programming determines the kinds of models of the world we create. Neuro-Linguistic Programming describes the fundamental dynamics between mind (neuro) and language (linguistic) and how their interplay affects our body and behavior (programming).  **Algorithm Proposition**  Our concept-level sentiment analysis system, our algo, is developed by combining lexicon-based and learning-based approaches. As shown in Figure, the supervised machine learning component is not just responsible for small tasks such as adjusting sentiment values or finding more sentiment words, but is actually responsible for evaluating all the ingredients of the sentiment system, including semantic rules used to derive the final output. The final component in Algo measures and reports the overall sentiment of a given opinionated text, such as a customer review, as a realvalued score between -1 and +1, which can then be easily transformed into a positive/negative classification or into a scale of 1-5 stars. | | | | | | |
|  | **Module - 2: Data Gathering and Preprocessing** | | | | | | |
|  | **DATA GATHERING**  Twitter’s Streaming API was used for mining tweet. Data Gathering was made up of two steps:  The first one was collecting the data to use as a training set to build the model which consisted of 4000 tweets manually labelled “positive”, “negative” or “neutral”.  The second step was collecting tweets during the T20 World Cup tournament and processing them by filtering some of the official World Cup hasthags (e.g. #T20WorldCup), as well as team code hashtags (e.g. #IND and #AUS). In addition, the Twitter usernames of teams and players were used to extract some more tweets related to events that occurred during the tournament. The data was in JSON format as a set of documents, one for each tweet.  **DATA PRE-PROCESSING**  The live tweets which are live streamed are full of unwanted and noisy contents which must be removed before using the data for training as well as testing. Pre-processing can be looked here as cleaning and filtration of data. Pre-processing helps in reducing size of data which helps in increasing computational speeds thereby decreasing time to train the NLP and getting faster results.   1. Removing all characters like “ ”,’ ‘, ( , ) , < , > , ; etc and other non-relevant symbols which are not required for analysis. 2. Removing stop words from tweets like the, it, a, is etc. These type of stop words are not important and just tend to consume space increasing size of given data. 3. Replacing the URL’s in the tweets using “URL” as they don’t help in deciding the polarity of a tweet. These URL’s were extracted and then removed. 4. To check for retweeted tweets which are the tweets shared by persons of one another. The “RT” keyword was the token for retweeted tweets which were available in JSON format from the spark stream. 5. To find the username or the username of the person to whom the tweet was focused on after the @ symbol. Ex. @imkohli, @YuviStrong etc. These usernames help to analyze the support for a particular team or player making them significantly important for analysis. 6. Removing the emoticons and emojis and replacing them with ASCII code relevant to the particular emoticon thereby helpful for further processing. These emoticons are pictorial representations of sentiments which are very important to be considered for analysis. | | | | | | |
|  | **Module - 3: AWS cloud and Elastic Search setup and configuration**  Preference for AWS:   1. Amazon Web Services (AWS) is a one year free cloud service provider for it’s free service. 2. Also, it provides various types of Storage and management systems, like EBS, EMR etc. 3. It also provides with numerous choices for RAM and IOPs configurations for various framework in today’s scenario.   Problems faced during Cloud setup:   1. We had to create our instance multiple times so as to make it functional for Spark Framework and Cluster Computing. 2. At first, We tried to create a normal instance without any particular type of storage with created problems for Spark Framework installation. 3. We then had to buy a more powerful instance with following specifications (approx.):     Total instances created - 3   |  |  | | --- | --- | | RAM | 8 GB each | | Processor Cores | 4 each | | Disk Space | 80 GB SSD each | | IOPs | (1000 IOPS), journal (250 IOPS), and log (100 IOPS) | | Cost | Rs. 80 per match |   **Table 6 – AWS Resource Specification** | | | | | | |
|  | **Module - 4: Apache Spark and NLP implementation**  Spark is mainly used to perform two things .  1. NLP (Natural Language Preocessing)  2. Naive Bayes Implementation  We have used stanford-coreNLP library to perform text analysis over tweets.  To find whether a tweet is negative or positive we defined a metric system which takes account of both the algorithm NLP and Naive Bayes . We send tweets to both the procedures to compute the result and then we classify that tweet for following categories.  1. Neutral  2. Mild Negative  3. Negative  4. Mild Positive  5. Positive  NLP computes score for each tweet by first breaking it into small tokens and then calculating score over each keywords by removing the irrelevant keywords and then adding up their scores , then it generates the score.  Spark Machine learning library gives us a implementation of naive bayes algorithm . We first train Naive bayes using our training data set and then applying it on a tweet and then producing a score .  After performing both the operations , we merge the result generated by both the functions and classifying them into above mentioned categories . | | | | | | |
|  | **Module - 5: Final Analysis and result visualization**  The hypothesis of the project focuses stated that the Indian cricket fans’ would experience more negative emotions after the Indian team tends to lose and that the fans would experience positive emotions after the indian team scored. We examined the patterns of the tweets to see whether they were consistent with our expectations. It should be acknowledged that during cricket games, there are multiple wickets,boundries and incidents of being attacked by the opponents and fouls that may also influence one’s emotional responses. Fans may even start celebrating or experiencing joy when a wicket is taken a boundry scored. We focused also concentrated on support based analysis and from where the tweets were originating.  The following visualizations were obtained for the matches we analysed :  sentimentanalysed**Figure 2- Pie-chart shows the distribution of sentiments analysed**  **createdAt**  **Figure 3 - Graph shows maximum number of tweets collected at a particular instant**  **usersupport**  **Figure 4 - Graph shows the user support analysed for a particular team**  **followers**  **Figure 5 - Graph shows the ranges of follower of the user who tweeted(hence popularity)**  **location**  **Figure 6 - Graph shows the loction of tweets origination** | | | | | | |
| **3.3** | **High Level Design Diagrams** | | | | | | |
|  |  | | | | | | |
| **3.3.1** | **Usecase Diagrams** | | | | | | |
| **Figure 7 - Usecase Diagram** | | | | | | | |
|  | | | | | | | |
| **3.3.2** | **Sequence Diagram** | | | | | | |
| **Figure 8 - Sequence Diagram** | | | | | | | |
| **3.3.3** | **Data-Flow Diagram** | | | | | | |
| **Figure 9 - Level 0 Dataflow Diagram** | | | | | | | |
| **3.3.7** | **PERT Chart** | | | | | | |
| **Figure 10 – P.E.R.T. Chart** | | | | | | | |
| **UNIT– 4** | | **METHODOLOGY & TEAM** | | | | | |
|  | | | | | | | |
| **4.1** | **Introduction to Waterfall Model** | | | | | | |
|  | The Waterfall Model was first Process Model to be introduced. It is also referred to as a linear-sequential life cycle model. It is very simple to understand and use. In a waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases.  The waterfall Model illustrates the software development process in a linear sequential flow; hence it is also referred to as a linear-sequential life cycle model. This means that any phase in the development process begins only if the previous phase is complete. In waterfall model phases do not overlap. In "The Waterfall" approach, the whole process of software development is divided into separate phases. In Waterfall model, typically, the outcome of one phase acts as the input for the next phase sequentially.  Following is a diagrammatic representation of different phases of waterfall model.  Untitled  **Figure 11 – Basic Waterfall Model**  The sequential phases in Waterfall model are:   * **Requirement Gathering and analysis:** All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification doc. * **System Design:** The requirement specifications from first phase are studied in this phase and system design is prepared. System Design helps in specifying hardware and system requirements and also helps in defining overall system architecture. * **Implementation:** With inputs from system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality which is referred to as Unit Testing. * **Integration and Testing:** All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures. * **Deployment of system:** Once the functional and non functional testing is done, the product is deployed in the customer environment or released into the market. * **Maintenance:** There are some issues which come up in the client environment. To fix those issues patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.   All these phases are cascaded to each other in which progress is seen as flowing steadily downwards (like a waterfall) through the phases. The next phase is started only after the defined set of goals are achieved for previous phase and it is signed off, so the name "Waterfall Model". In this model phases do not overlap.  Every software developed is different and requires a suitable SDLC approach to be followed based on the internal and external factors. Some situations where the use of Waterfall model is most appropriate are:   * Requirements are very well documented, clear and fixed. * Product definition is stable. * Technology is understood and is not dynamic. * Ample resources with required expertise are available to support the product. * The project is short.  **Waterfall Model Pros & Cons**Advantage The advantage of waterfall development is that it allows for departmentalization and control. A schedule can be set with deadlines for each stage of development and a product can proceed through the development process model phases one by one.  Development moves from concept, through design, implementation, testing, installation, troubleshooting, and ends up at operation and maintenance. Each phase of development proceeds in strict order. Disadvantage The disadvantage of waterfall development is that it does not allow for much reflection or revision. Once an application is in the testing stage, it is very difficult to go back and change something that was not well-documented or thought upon in the concept stage. | | | | | | |
| **4.2** | **Team Members, Roles & Responsibilities** | | | | | | |
| |  |  |  | | --- | --- | --- | | **Team Member** | **Project Role** | **Responsibilities** | | Prashant Kathuria | Developer | Machine Learning, Apache Spark, Scala | | Rajat Gupta | Developer | Core NLP, Spark streaming, Scala | | Subroto Biswas | Developer | Twitter API Streaming, Elastic Search, AWS setup | | Tushar Mittal | Developer | Data Preprocessing, Library formation, Kibana |   **Table 7 – Roles and responsibilities** | | | | | | | |
| **UNIT– 5** | | | **SYSTEM TESTING** | | | | |
|  | | | | | | | |
| The designed system has been testing through following test parameters. | | | | | | | |
|  | | | | | | | |
| **5.1** | **Functionality Testing** | | | | | | |
|  | The test case we performed to check functionality was to deploy our analytical engine to a cloud. The initial attempt to do that met with a failure as big data components cannot be run on clouds that easily.  We had to use Amazon MapReduce Component to be able to run a big data technology such as spark to be able to easily run it on the cloud.  Another test case we performed was the speed of Apache Spark framework and to utilize the most of its in-memory capability. To do that we have to optimize the code as well as the Spark configuration.We used persist() operations and increased the memory allocated to such operations to increase the speed of the spark components. | | | | | | |
| **5.2** | **Performance Testing** | | | | | | |
|  | Performance testing can be applied to understand the website’s scalability, or to benchmark the performance in the environment of third party products such as servers and middleware for potential purchase. This can only be done once it is put into use on the actual internet server and tested by the users.  The test case generated was to check the accuracy of the Naïve Bayes classifier. To do this we splitted the training dataset into ratio 80/20 where 80% of it was to train the classifier and the rest that is 20% of it was to test the classifier we just trained. By such method we obtained an accuracy of the model as 68%. Which is considered optimal for data obtained from tweets. | | | | | | |
| **5.3** | **Usability Testing** | | | | | | |
|  | Usability testing is the process by which the human-computer interaction characteristics of a system are measured, and weaknesses are identified for correction.   1. Ease of learning 2. Navigation 3. Subjective user satisfaction 4. General appearance   As system is not put into the real time use so it’s not yet tested for usability. | | | | | | |
| **5.4** | **Server Side Interfacing** | | | | | | |
|  | In this we tested the server side interface. This was done by verifying that communication is done properly. Also the compatibility of server with software, hardware, network and database was tested. | | | | | | |
| **5.5** | **Client Side Compatibility** | | | | | | |
|  | The client side compatibility is also tested using various browsers like Google Chrome, Mozilla Firefox and Internet Explorer. | | | | | | |
|  | | | | | | | |
| **UNIT– 6** | | **TEST EXECUTION SUMMARY** | | | | | |
|  | | | | | | | |
|  | Execution Test Summary Report is an overall view of Testing Process from start to end. Test Plan comes at the starting of project while Test Summary Report comes at the end of testing process. This report is given to the client for his understanding purpose.      The Test Summary Report contents are : We computed test case analysis on the two analytical components i.e. apache spark’s naïve bayes algorithm and CoreNLP’s Natural language processing engine and obtained results as follows:  **Apache Spark MLLib (Naïve Bayes)**  We splitted the test data into 80/20 ratio and trained data on 80% part and analysed on the test cases from the remaining 20% of the data.  Doing this we obtained an accuracy of 68% on this data.  **CoreNLP (Natural Language Processing)**  In this we performed analysis on the data by comparing the label obtained on the whole data with the previous data we used for training.  Doing this we obtained an accuracy of 71% on our data.  In all we obtained an accuracy of 69% on our data with the whole process of combining our two analytical approaches. | | | | | | |
|  |  | | | | | | |
|  | | | | | | | |
| **UNIT– 7** | | **PROJECT SCREENSHOTS** | | | | | |
|  | | | | | | | |
| Untitled **Figure 12 - Apache spark cluster running interface**  Untitled1  **Figure 13 – Elastic Search Plugin showing all data** | | | | | | | |
| C:\Users\Prashant\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Untitled3.png  **Figure 14 - Elastic search plugin showing data of a particular table** | | | | | | | |
| Screenshot from 2016-05-13 19_24_31  **Figure 15 - Kibana showing the data imported**  Capture44  **Figure 16 - Kibana’s interface to import new data**  **Capture444**  **Figure 17 – Kibana’s interface to create a new visualisation** | | | | | | | |
|  | | | | | | | |
| **UNIT– 8** | | **PROJECT SUMMARY AND CONCLUSIONS** | | | | | |
|  | | | | | | | |
|  | The object of this project is to harness the power of Internet for our practical and potential one. This report explains to extensively cover this concept and plant a seed of inquisitiveness in the mind of developers.  This application aims in inculcating data mining and warehousing functionality on vast amount of data so that a domain oriented analysis of a particular topic or product helps in developing business and technical strategies to improve business and help in marketing.  Our project is currently on track with the exception of a couple of problems that were encountered. The first is that we have not yet figured out how to calculate the support of a specific player i.e. his/her popularity. Implementation of this function will require further research in the field of Text Analytics, NLP, Machine Learning, Data Mining, Pattern matching etc and enhancement of cricketing keyword library.  The second problem is determining the reason during the computation we found that ratio of neutral tweets was way more than other categories (positive , negative, mildly positive, mildly negative).To troubleshoot this problem we have to improve the training set of Naïve Bayes classifier and further research in NLP is required. | | | | | | |
|  | | | | | | | |
| **UNIT– 9** | | **FUTURE SCOPE** | | | | | |
|  | | | | | | | |
|  | The possible future scope of this application will be on mobile platform with following enhancements:  1. Designing of mobile application for this project on Android / iPhone platforms.  2. Generalize for any type of event or tournament.  3. Use for business analysis and develop a product where a company can subscribe and analyze data that is required for business scaling.  4. Analyze the popularity of a player according to the support. | | | | | | |
|  | | | | | | | |
| **UNIT– 10** | | | | | | **REFERENCES** | |
| **PAPERS:**  [1] GOAALLL!: Using Sentiment in the World Cup to Explore Theories of Emotion Jonathan Gratch, Gale Lucas, Nikolaos Malandrakis, Evan Szablowski, Eli Fessler and Jeffrey Nichols  [2] Text Analysis and Sentiment Polarity on FIFA World Cup 2014 Tweets Peiman Barnaghi ,Parsa Ghaffari ,John G. Breslin  [3] A. Bifet and E. Frank, ”Sentiment knowledge discovery in twitter streaming data,” in Discovery Science, 2010, pp. 1-15.  [4] D. Terrana, A. Augello, and G. Pilato, ”Automatic Unsupervised Polarit Detection on a Twitter Data Stream,” in Semantic Computing (ICSC), 2014 IEEE International Conference on, 2014  [5] Manning, Christopher D., Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations  **WEBSITES:**  1. www.stackoverflow.com  2. www.developers.google.com  3. www.spark.apache.org  4. www.aws.amazon.com  5. www.elastic.co    6. www.stanfordnlp.github.io | | | | | | | |
|  | | | | | | | |
| **UNIT– 11** | | | | | **PROJECT PAPER** | | |
|  | | | | | | | |