**Sentiment Analysis Using Python**

A Project Report

Submitted in partial fulfillment of the

Requirements for the award of the Degree of

BACHELOR OF SCIENCE (INFORMATION

TECHNOLOGY)

By

Bodke Sahil Santosh

2011957

Under the esteemed guidance of

# Dr.Hiren Dand

Coordinator, Department of Information Technology



DEPARTMENT OF INFORMATION TECHNOLOGY

Parle Tilak Vidyalaya Association’s

MULUND COLLEGE OF COMMERCE

(Affiliated to University of Mumbai)

NAAC Re-Accredited A Grade – III Cycle

MULUND WEST, MUMBAI 400080

MAHARASHTRA, INDIA

2021 – 2022

# PROJECT PROPOSAL

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | PRN No | : | 2019016400446841 |
| 2 | Roll No | : | 2011957 |
| 3 | Name of the Student | : | Bodke Sahil Santosh |
| 4 | Title of the Project | : | Sentiment Analysis Using Python |
| 5 | Name of the Guide | : | Dr. Hiren Dand |
| 6 | Teaching Experience of  Guide | : | 29 years |
| 7 | Is this your first Submission | : | Yes |

Signature of the Student Signature of the Coordinator

Date: 29/07/21 Date: October 7, 2021

**Parle Tilak Vidyalaya Association’s**

**MULUND COLLEGE OF COMMERCE**

**(Affiliated to University Of Mumbai)**

**NAAC Re-Accredited A Grade – III Cycle**

**MULUND WEST, MUMBAI 400080**

**MAHARASHTRA, INDIA**

**DEPARTMENT OF INFORMATION TECHNOLOGY**



**Certificate**

This is to certify that the Project entitled, “ Sentiment Analysis Using Python”, is Bonafide work of Bodke Sahil Santosh bearing Seat NO : 2011957 submitted in partial fulfilment of the requirements for the award of degree of BACHELOR OF SCIENCE in INFORMATION TECHNOLOGY from University of Mumbai during the academic year 2021-2022

Name and Signature of Guiding Teacher

Date of submission: 16 APR 2022

**Declaration**

*Seal of the College*

I, hereby declare that the Project entitled, “Sentiment Analysis Using Python” done at Mulund College Of Commerce has not been in any case duplicated to submit to any other university for the award of any degree. To the best of my knowledge other than me, no one has submitted to any other university.

The Project is done in partial fulfilment of the requirements for the award of degree of BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY) to be submitted as final Semester project as part of our curriculum.

Name and signature of the Student

**Acknowledgment**

To list who all have helped me is difficult because they are so numerous and the depth is so enormous.

I would like to acknowledge the following as being idealistic channels and fresh dimensions in the completion of this project.

I take this opportunity to thank the **University of Mumbai** for giving me chance to do this project.

I would like to thank my **Principal**, DR. SONALI PEDNEKAR for providing the necessary facilities required for completion of this project.

I take this opportunity to thank our **Coordinator** Prof DR. Hiren Dand for his moral support and guidance.

I would like to thank my **College Library**, for having provided various reference books and magazines related to my project.

Lastly, I would like to thank each and every person who directly or indirectly helped me in the completion of the project especially **my Parents and Peers** who supported me throughout my project.

**INDEX**

|  |  |
| --- | --- |
| **Sr No** | **Description** |
| 1. | Introduction |
| 1.1 Theory behind the project concept |
| 1.2 Problem Definition |
| 2. | Literature Survey |
| 3. | Analysis & Design |
| 3.1 Analysis |
| 3.1.1 Feasibility Study |
| 3.1.2 Process Model |
| 3.2 Design |
| 3.2.1Data Flow Diagram |
| 3.2.2 UML Diagram |
| 3.2.3 EER Diagram |
| 3.2.4 Sequence Diagram |
| 3.2.5 Activity Diagram |
| 3.2.6 Deployment Diagram |
| 3.3 Technologies Used |
| 3.3.1 Hardware and Software |
| 3.3.2 Introduction to Programming Tools |
| 4 | Project Timeline |
| 5. | Implementation |
|  | 5.1 Pre-processing of Tweets |
|  | 5.2 Scoring Module |
|  | 5.3 Tweet Sentiment Scoring |
| 6. | Testing |
|  | 6.1 Unit Test |
|  | 6.2 System Testing |
|  | 6.3 User Acceptance Test |
| 7. | Conclusion & Future Support |
| 7.1 Conclusion |
| 7.2 Future Scope |

**List of Figures**

|  |  |  |
| --- | --- | --- |
| **Sr No** | **Figure Name** | **Page No** |
| 1 | Iterative Model | 13 |
| 2 | DFD Diagram | 15 |
| 3 | Use Case Diagram | 16 |
| 4 | EER Diagram | 18 |
| 5 | Interaction Diagram | 19 |
| 6 | Activity Diagram | 20 |
| 7 | Deployment Diagram | 21 |
| 8 | Gantt Chart | 24 |
| 7 | System Architecture | 26 |

1. **Introduction**

Ongoing increase in wide-area network connectivity promise vastly augmented opportunities for collaboration and resource sharing. Now-a-days, various social networking sites like Twitter, Facebook, MySpace, YouTube have gained so much popularity and we cannot ignore them. They have become one of the most important applications of Web. They allow people to build connection networks with other people in an easy and timely way and allow them to share various kinds of information and to use a set of services like picture sharing, blogs, wikis etc.

It is evident that the advent of these real-time information networking sites like Twitter have spawned the creation of an unequaled public collection of opinions about every global entity that is of interest. Although Twitter may provision for an excellent channel for opinion creation and presentation, it poses newer and different challenges and the process is incomplete without adept tools for analyzing those opinions to expedite their consumption.

* 1. **Theory behind the project concept.**

Due to the increase of hostile and negative communication over social networking sites like Facebook and Twitter, recently the Government of India tried to allay concerns over censorship of these sites where Web users continued to speak out against any proposed restriction on posting of content.

Sentiment analysis to gauge the public mood and detect any rising antagonistic or negative feeling on social medias. Although, we firmly believe that censorship is not right path to follow, this recent trend for research for sentiment mining in twitter can be utilized and extended for a gamut of practical applications that range from applications in business (marketing intelligence; product and service bench marking and improvement), applications as subcomponent technology (recommender systems; summarization; question answering) to applications in politics. This motivated us to propose a model which retrieves tweets on a certain topic through the Twitter API and calculates the sentiment orientation/score of each tweet.

The area of Sentiment Analysis intends to comprehend these opinions and distribute them into the categories like positive, negative, neutral. Till now most sentiment analysis work has been done on review sites. Review sites provide with the sentiments of products or movies, thus, restricting the domain of application to solely business. Sentiment analysis on Twitter posts is the next step in the field of sentiment analysis, as tweets give us a richer and more varied resource of opinions and sentiments that can be about anything from the latest phone they bought, movie they watched, political issues, religious views or the individuals state of mind. Thus, the foray into Twitter as the corpus allows us to move into different dimensions and diverse applications.

* 1. **Problem Definition**

People usually want to collect more information about a product before purchasing. They usually consider the opinion of other consumers to make decision on their purchase. Gathering all these reviews helps manufacturers to aware of the weakness and strength of their product to improve it. Web blogs are full of un-indexed and unprocessed text that reflects the opinions of people. It is not easy to walk through a lot of reviews and read the comments carefully in order to find which attribute or component of the product has received more feedbacks from the consumers.

Sometime reviews are directly talked about the product and sometime reviews are explicitly mentioned. For example, “it is not easy to carry.” This is a negative opinion on a cell phone. “weight” is an implicit feature of the cell phone which is implied from the sentence. Thus, there is a need to collect and process people’s opinions so that it can be used in decision making processes.

In this proposed system, it propose a Twitter blog mining system that will collects, parsed, processed, annotate and analyzed the movie reviews or comments from Twitter blogs which express either positive or negative sentiment and that will show web blog users what other people think about a particular product.

1. **Literature Survey**

Applying sentiment analysis on Twitter is the upcoming trend with researchers recognizing the scientific trials and its potential applications. The challenges unique to this problem area are largely attributed to the dominantly informal tone of the micro blogging.

Rationale the use micro blogging and more particularly Twitter as a corpus for sentiment analysis. They cited

* Micro blogging platforms are used by different people to express their opinion about different topics, thus it is a valuable source of people’s opinions.
* Twitter contains an enormous number of text posts and it grows every day. The collected corpus can be arbitrarily large.
* Twitter’s audience varies from regular users to celebrities, company representatives, politicians, and even country presidents. Therefore, it is possible to collect text posts of users from different social and interests groups.
* Twitter’s audience is represented by users from many countries.

Parikh and Movassate implemented two Naive Bayes unigram models, a Naive Bayes bigram model and a Maximum Entropy model to classify tweets. They found that the Naive Bayes classifiers worked much better than the Maximum Entropy model could. Go et al. proposed a solution by using distant supervision, in which their training data consisted of tweets with emoticons. This approach was initially introduced by Read. The emoticons served as noisy labels. They build models using Naive Bayes, MaxEnt and Support Vector Machines (SVM). Their feature space consisted of unigrams, bigrams and POS. The reported that SVM outperformed other models and that unigram were more effective as features.

Pak and Paroubek have done similar work but classify the tweets as objective, positive and negative. In order to collect a corpus of objective posts, they retrieved text messages from Twitter accounts of popular newspapers and magazine, such as “New York Times”, “Washington Posts” etc. Their classifier is based on the multinomial Naïve Bayes classifier that uses N-gram and POS-tags as features.

Barbosa et al. too classified tweets as objective or subjective and then the subjective tweets were classified as positive or negative. The feature space used included features of tweets like re tweet, hash tags, link, punctuation and exclamation marks in conjunction with features like prior polarity of words and POS of words.

Mining for entity opinions in Twitter, Batra and Rao used a dataset of tweets spanning two months starting from June 2009. The dataset has roughly 60 million tweets. The entity was extracted using the Stanford NER, user tags and URLs were used to augment the entities found. A corpus of 200,000 product reviews that had been labeled as positive or negative was used to train the model. Using this corpus the model computed the probability that a given unigram or bigram was being used in a positive context and the probability that it was being used in a negative context.

Bifet and Frank used Twitter streaming data provided by Firehouse, which gave all messages from every user in real-time. They experimented with three fast incremental methods that were well-suited to deal with data streams: multinomial naive Bayes, stochastic gradient descent, and the Hoeffding tree. They concluded that SGD-based model, used with an appropriate learning rate was the best.

1. **Analysis & Design**
   1. **Analysis**
      1. **Feasibility Study**

The very first phase in any system developing life cycle is preliminary investigation. The feasibility study is a major part of this phase. A measure of how beneficial or practical the development of any information system would be to the organization is the feasibility study.

The feasibility of the development software can be studied in terms of the following aspect:-

* OPERATIONAL FEASIBILITY :- The site will reduce the time consumed to maintain manual records and is not tiresome and cumbersome to maintain the records. Hence operational feasibility is assured
* TECHNICAL FEASIBILITY :-
* At least 166 MHz Pentium Processor or Intel compatible processor.
* At least 512 MB RAM.
* 14.4 kbps or higher modem.
* A mouse or other pointing device.
* At least 50 GB free hard disk space.
* Microsoft Internet Explorer 4.0 or higher.
* ECONOMICAL FEASIBILTY :-Once the hardware and software requirements get fulfilled, there is no need for the user of our system to spend for any additional overhead .For the user, the web site will be economically feasible in the following aspects:
  + The web site will reduce a lot of paper work. Hence the cost will be reduced.
  + Our web site will reduce the time that is wasted in manual processes.
  + The storage and handling problems of the registers will be solved.
* LEGAL FEASIBILITY :- The licensed copy of the required software is quite cheap and easy to get. So from legal point of view the proposed system is legally feasible.
  + 1. Process Model

A software development process is the process of dividing software development work into distinct phases to improve design, product management, and project management. It is also known as a software development life cycle. The methodology may include the pre-definition of specific deliverables and artifacts that are created and completed by a project team to develop or maintain an application.

Most modern development processes can be vaguely described as agile. Other methodologies include waterfall, prototyping, iterative and incremental development, spiral development, rapid application development, and extreme programming.

**The Iterative model**

The waterfall model derives its name due to the cascading effect from one phase to the other as is illustrated in Figure1.1. In this model each phase well defined starting and ending point, with identifiable deliveries to the next phase.

This model is sometimes referred to as the linear sequential model or the software life cycle.

The model consists of six distinct stages, namely:

1.      In the **requirements analysis** phase

1. The problem is specified along with the desired service objectives (goals)
2. The constraints are identified

2.      In the **specification phase** the system specification is produced from the detailed definitions of (a) and (b) above. This document should clearly define the product function.

3.      In the system and software **design phase**, the system specifications are translated into a software representation. The software engineer at this stage is concerned with:

* Data structure
* Software architecture
* Algorithmic detail
* Interface representations

The hardware requirements are also determined at this stage along with a picture of the overall system architecture. By the end of this stage should the software engineer should be able to identify the relationship between the hardware, software and the associated interfaces. Any faults in the specification should ideally not be passed ‘down stream.

4.      In the **implementation and testing**phase stage the designs are translated into the software domain

* Detailed documentation from the design phase can significantly reduce the coding effort.
* Testing at this stage focuses on making sure that any errors are identified and that the software meets its required specification.

5.      In the **integration and system testing** phase all the program units are integrated and tested to ensure that the complete system meets the software requirements. After this stage the software is delivered to the customer [**Deliverable – The software product is delivered to the client for acceptance testing**.]

6.      The **maintenance** phase the usually the longest stage of the software. In this phase the software is updated to:

* Meet the changing customer needs
* Adapted to accommodate changes in the external environment
* Correct errors and oversights previously undetected in the testing phases
* Enhancing the efficiency of the software

Observe that feed back loops allow for corrections to be incorporated into the model. For example a problem/update in the design phase requires a ‘revisit’ to the specifications phase. When changes are made at any phase, the relevant documentation should be updated to reflect that change.

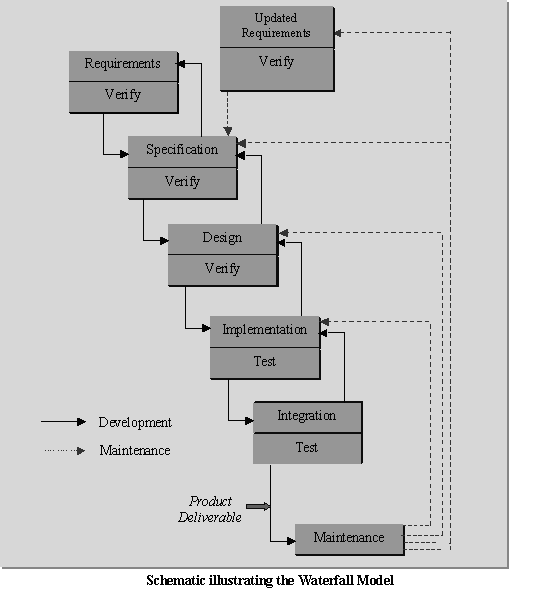
**Advantages of the Iterative Model:-**

* Testing is inherent to every phase of the Iterative model
* It is an enforced disciplined approach
* It is documentation driven, that is, documentation is produced at every stage

**Disadvantages of the Iterative Model:-**

The waterfall model is the oldest and the most widely used paradigm. However, many projects rarely follow its sequential flow. This is due to the inherent problems associated with its rigid format. Namely:

* It only incorporates iteration indirectly, thus changes may cause considerable confusion as the project progresses.
* As The client usually only has a vague idea of exactly what is required from the software product, this IM has difficulty accommodating the natural uncertainty that exists at the beginning of the project.
* The customer only sees a working version of the product after it has been coded. This may



**Figure 1- Iterative Model**

* 1. **Design**

Software design is the process of implementing software solutions to one or more sets of problems. One of the main components of software design is the software requirements analysis (SRA). SRA is a part of the software development process that lists specifications used in software engineering.

If the software is "semi-automated" or user centered, software design may involve user experience design yielding a storyboard to help determine those specifications. If the software is completely automated (meaning no user or user interface), a software design may be as simple as a flow chart or text describing a planned sequence of events.

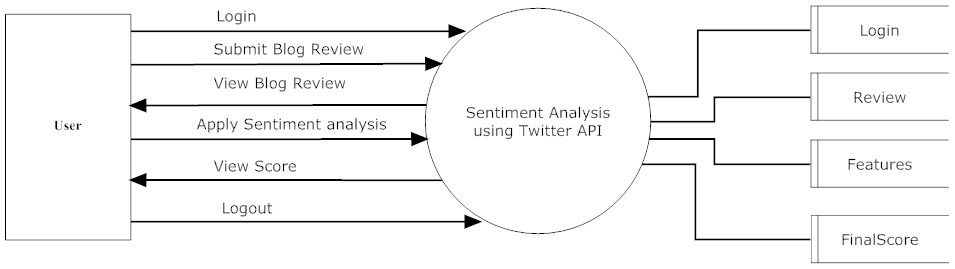
There are also semi-standard methods like Unified Modeling Language and Fundamental modeling concepts. In either case, some documentation of the plan is usually the product of the design. Furthermore, a software design may be platform-independent or platform-specific, depending upon the availability of the technology used for the design.

The design process is a sequence of steps that enables the designer to describe all aspects of the software for building. Creative skill, experience, a sense of what makes "good" software, and an overall commitment to quality are examples of critical success factors for a competent design. It is important to note, however, that the design process is not always a straightforward procedure.

The design model can be compared to an architect’s plans for a house. It begins by representing the totality of the thing that is to be built (e.g., a three-dimensional rendering of the house); slowly, the thing is refined to provide guidance for constructing each detail (e.g., the plumbing lay). Similarly, the design model that is created for software provides a variety of different views of the computer software.

Software design usually involves problem solving and planning a software solution. This includes both a low-level component and algorithm design and a high-level, architecture design. Basic design principles enable the software engineer to navigate the design process.

* + 1. **Data flow Diagram**



*Figure 2 - DFD Diagram*

* + 1. **UML Diagram**

The Unified Modeling Language (UML) is a general-purpose, developmental, modeling language in the field of software engineering that is intended to provide a standard way to visualize the design of a system.

The creation of UML was originally motivated by the desire to standardize the disparate notational systems and approaches to software design.

UML offers a way to visualize a system's architectural blueprints in a diagram, including elements such as:

* any activities (jobs);
* individual components of the system;
* and how they can interact with other software components;
* how the system will run;
* how entities interact with others (components and interfaces);
* external user interface.

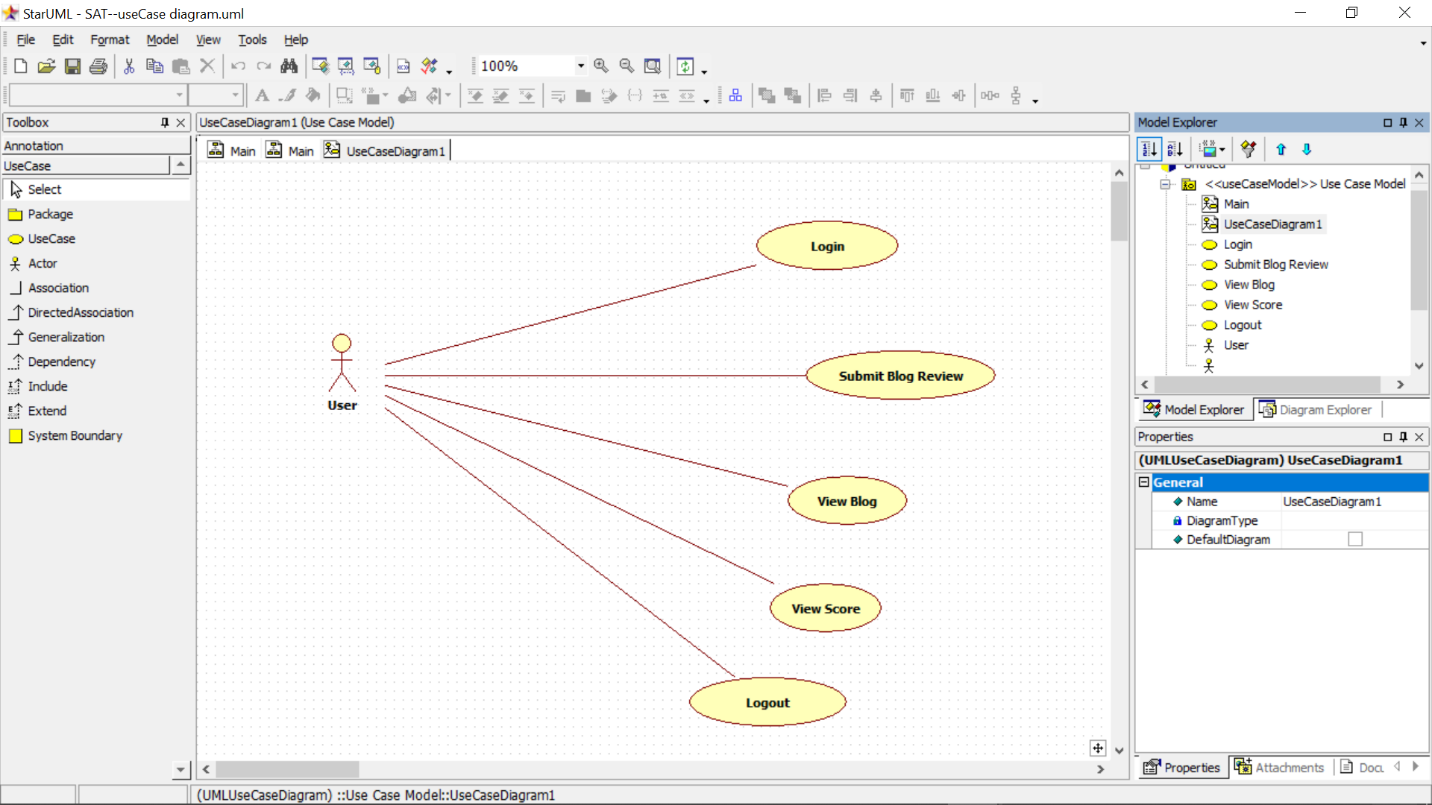
Although originally intended for object-oriented design documentation, UML has been extended to a larger set of design documentation (as listed above) and been found useful in many contexts.

UML diagrams represent two different views of a system model:

• Static (or structural) view: emphasizes the static structure of the system using objects, attributes, operations and relationships. It includes class diagrams and composite structure diagrams.

• Dynamic (or behavioral) view: emphasizes the dynamic behavior of the system by showing collaborations among objects and changes to the internal states of objects. This view includes sequence diagrams, activity diagrams and state machine diagrams.

**Use Case Diagram**



*Figure 3 - Use Case Diagram*

3.2.3 EER Diagram

An entity–relationship model is usually the result of systematic analysis to define and describe what is important to process in an area of a business. It does not define the business processes; it only presents a business data schema in graphical form. It is usually drawn in a graphical form as boxes (entities) that are connected by lines (relationships) which express the associations and dependencies between entities. An ER model can also be expressed in a verbal form, for example: one building may be divided into zero or more apartments, but one apartment can only be in one building.

Entities may be characterized not only by relationships, but also by additional properties (attributes), which include identifiers called "primary keys". Diagrams created to represent attributes as well as entities and relationships may be called entity-attribute-relationship diagrams, rather than entity–relationship models.

An ER model is typically implemented as a database. In a simple relational database implementation, each row of a table represents one instance of an entity type, and each field in a table represents an attribute type. In a relational database a relationship between entities is implemented by storing the primary key of one entity as a pointer or "foreign key" in the table of another entity

**EER Model:**

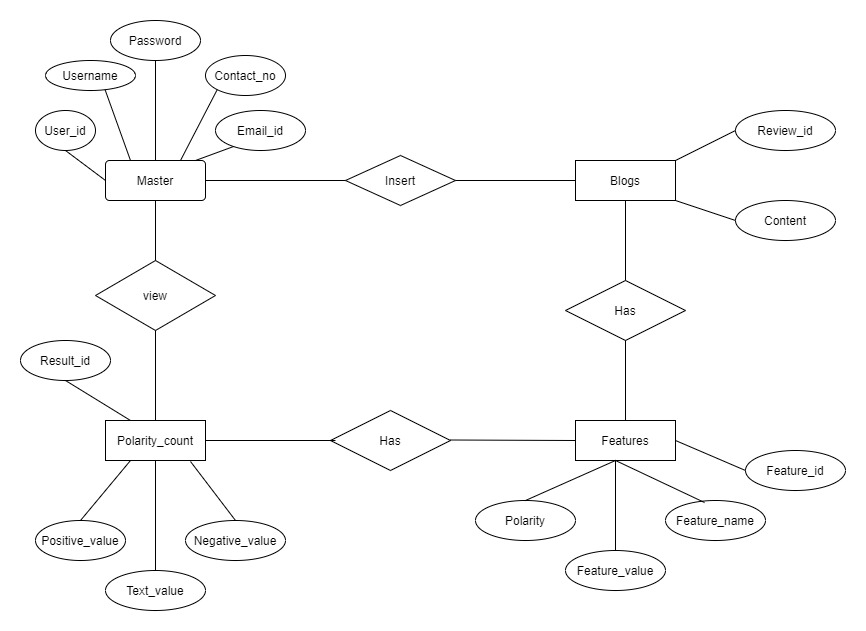
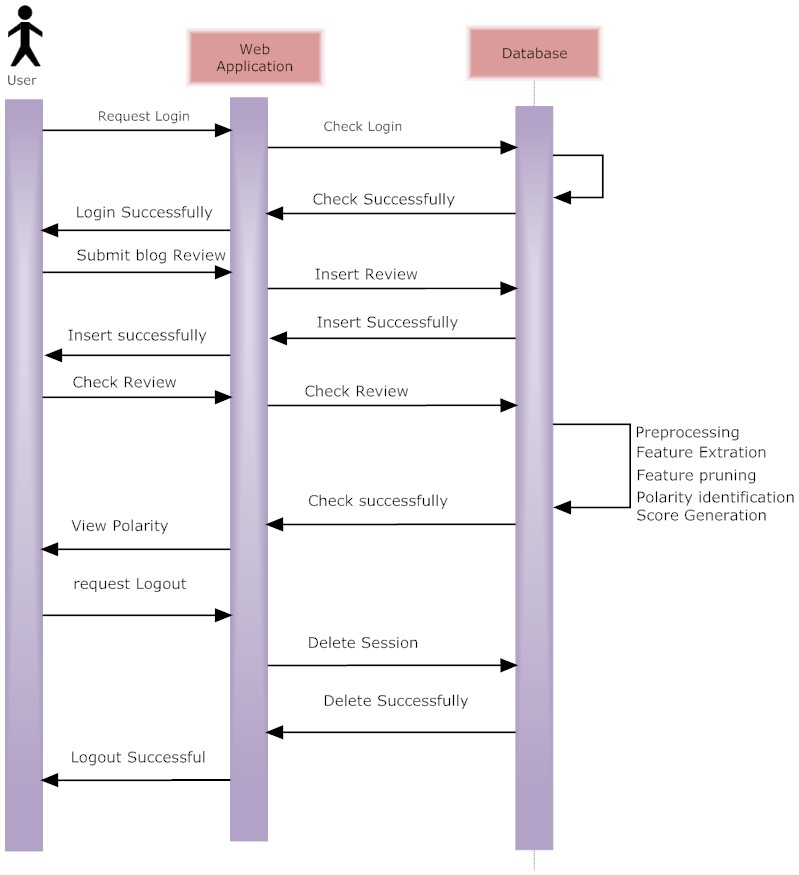


Figure 4- EER Diagram

3.2.4 **Interaction Diagram**

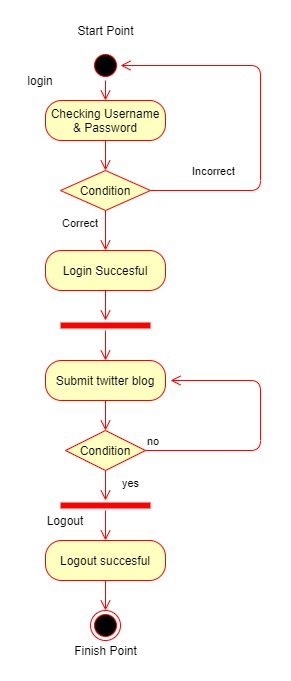
An interaction diagram is used to show the interactive behavior of a system. Since visualizing the interactions in a system can be a cumbersome task, we use different types of interaction diagrams to capture various features and aspects of interaction in a system.

Sequence Diagrams – A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function. These diagrams are widely used by businessmen and software developers to document and understand requirements for new and existing systems.

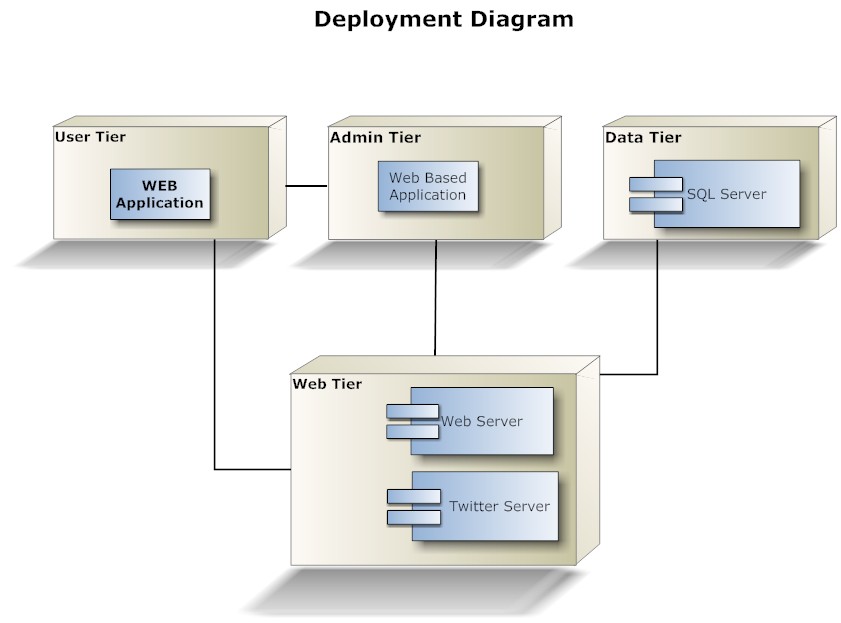


*Figure 5- Sequence Diagram*

3.2.5 Activity Diagram

**  
*Figure 6- Activity Diagram*

3.2.6 Deployment Diagram



*Figure 7- Deployment Diagram*

3.3 Technologies Used

3.3.1 Hardware & Software

* Hardware Requirements:
  + 1 GB Ram
  + 200 GB HDD
  + Intel 1.66 GHz Processor Pentium 4
* Software Requirements:
  + Windows 8,10
  + Visual Studio 2010
  + MS SQL Server 2008

3.3.2 Introduction to Programming Tools

Python

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

Python interpreters are available for many operating systems. CPython, the reference implementation of Python, is open source software and has a community-based development model, as do nearly all its variant implementations. CPython is managed by the non-profit Python Software Foundation.

**Python Libraries**

Since all the libraries are open sourced, we have added commits, contributors count and other metrics from github, which could be served as a proxy metrics for library popularity.

1. NumPy

When starting to deal with the scientific task in Python, one inevitably comes for help to Python’s SciPy Stack, which is a collection of software specifically designed for scientific computing in Python (do not confuse with SciPy library, which is part of this stack, and the community around this stack). This way we want to start with a look at it. However, the stack is vast, there is more than a dozen of libraries in it, and we want to put a focal point on the core packages (particularly the most essential ones).

The most fundamental package, around which the scientific computation stack is built, is NumPy (stands for Numerical Python). It provides an abundance of useful features for operations on n-arrays and matrices in. The library provides vectorization of mathematical operations on the NumPy array type, which ameliorates performance and accordingly speeds up the execution.

1. Tweepy

Twitter is a popular social network where users share messages called tweets. Twitter allows us to mine the data of any user using Twitter API or Tweepy. The data will be tweets extracted from the user. The first thing to do is get the consumer key, consumer secret, access key and access secret from twitter developer available easily for each user. These keys will help the API for authentication.

1. Re (Regular expression):-

Specifies a set of strings that matches it; the functions in this module let you check if a particular string matches a given regular expression (or if a given regular expression matches a particular string, which comes down to the same thing).

Regular expressions can be concatenated to form new regular expressions; if *A* and *B* are both regular expressions, then *AB* is also a regular expression. In general, if a string *p* matches *A* and another string *q* matches *B*, the string *pq* will match AB. This holds unless *A* or *B* contain low precedence operations; boundary conditions between *A* and *B*; or have numbered group references. Thus, complex expressions can easily be constructed from simpler primitive expressions like the ones described here

d) Pandas:-

Pandas is an open source library that allows to you perform data manipulation in Python. Pandas library is built on top of Numpy, meaning Pandas needs Numpy to operate. Pandas provide an easy way to create, manipulate and wrangle the data. Pandas is also an elegant solution for time series data.

Data scientists use Pandas for its following advantages:

* Easily handles missing data
* It uses **Series for one-dimensional data structure** and **DataFrame for multi-dimensional data structure**
* It provides an efficient way to slice the data
* It provides a flexible way to merge, concatenate or reshape the data
* It includes a powerful time series tool to work with

**4. Project Timeline**

In project management, a schedule is a listing of a project's milestones, activities, and deliverables, usually with intended start and finish dates. Those items are often estimated by other information included in the project schedule of resource allocation, budget, task duration, and linkages of dependencies and scheduled events. A schedule is commonly used in the project planning and project portfolio management parts of project management. Elements on a schedule may be closely related to the work breakdown structure (WBS) terminal elements, the Statement of work, or a Contract Data Requirements List.

A Gantt chart is a type of bar chart that illustrates a project schedule. Modern Gantt charts also show the dependency relationships between activities and current schedule status.

A Gantt chart is a type of bar chart that illustrates a project schedule. This chart lists the tasks to be performed on the vertical axis, and time intervals on the horizontal axis. The width of the horizontal bars in the graph show the duration of each activity.Gantt charts illustrate the start and finish dates of the terminal elements and summary elements of a project. Terminal elements and summary elements constitute the work breakdown structure of the project. Modern Gantt charts also show the dependency (i.e., precedence network) relationships between activities. Gantt charts can be used to show current schedule status using percent-complete shadings and a vertical "TODAY" line as shown here. Gantt charts are sometimes equated with bar charts.

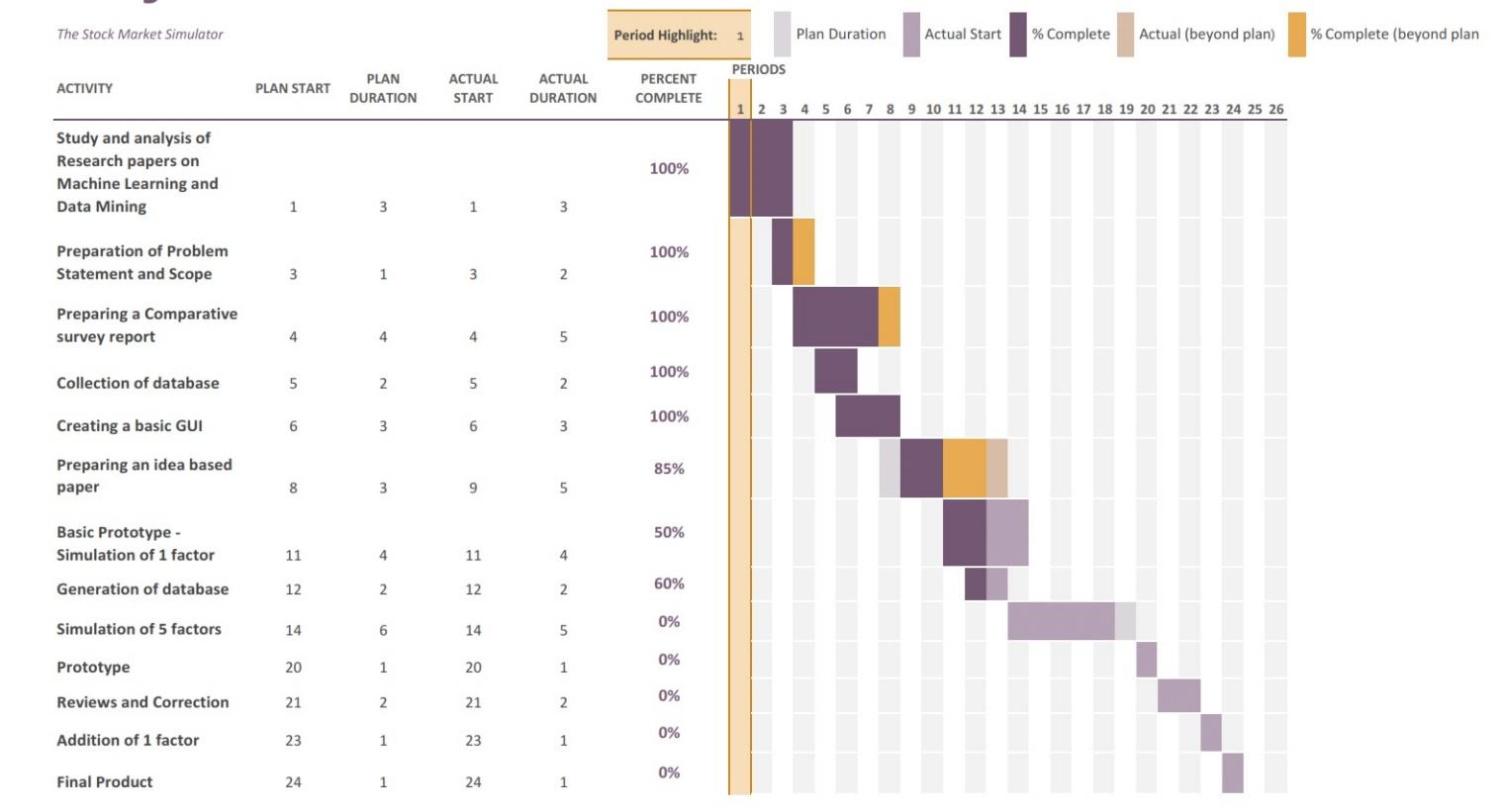
Gantt charts are usually created initially using an early start time approach, where each task is scheduled to start immediately when its prerequisites are complete. This method maximizes the float time available for all tasks.

Figure 6- Gantt Chart

**5. Implementation**

5.1 Pre-processing of Tweets

We prepare the transaction file that contains opinion indicators, namely the adjective, adverb and verb along with emoticons (we have taken a sample set of emoticons and manually assigned opinion strength to them). Also we identify some emotion intensifiers, namely, the percentage of the tweet in Caps, the length of repeated sequences & the number of exclamation marks, amongst others. Thus, we pre-process all the tweets as follows:

1. Remove all URLs (e.g. www.example.com), hash tags (e.g. #topic), targets (@username), special Twitter words (“e.g. RT”).
2. Calculate the percentage of the tweet in Caps.
3. Correct spellings; A sequence of repeated characters is tagged by a weight. We do this to differentiate between the regular usage and emphasized usage of a word.
4. Replace all the emoticons with their sentiment polarity (Table 1).
5. Remove all punctuations after counting the number of exclamation marks.
6. Using a POS tagger, the NL Processor linguistic Parser [15], we tag the adjectives, verbs and adverbs.

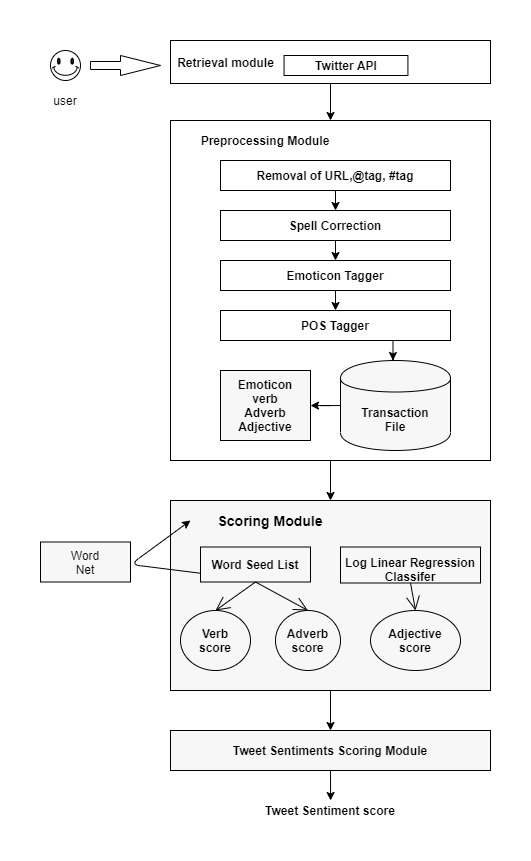
5.2 Scoring Module

The next step is to find the semantic score of the opinion carriers i.e. the adjectives, verbs and adverbs. As mentioned previously, in our approach we use corpus based method to find the semantic orientation of adjectives and the dictionary-based method to find the semantic orientation of verbs and adverbs.

5.3 .Tweet Sentiment Scoring

As adverbs qualify adjectives and verbs, we group the corresponding adverb and adjective together and call it the adjective group; similarly we group the corresponding verb and adverb together and call it the verb group. The adjective group strength is calculated by the product of adjective score (adji) and adverb (advi) score, and the verb group strength as the product of verb score (vbi) and adverb score (advi). Sometimes, there is no adverb in the opinion group, so the S (adv) is set as a default value 0.5 To calculate the overall sentiment of the tweet, we average the strength of all opinion indicators like emoticons, exclamation marks, capitalization, word emphasis, adjective group and verb group as shown below:

=



**Figure 7-System Architecture**

**6 Testing:-**

**6.1 Unit Testing:-**

Unit testing is **a software development process in which the smallest testable parts of an application, called units, are individually and independently scrutinized for proper operation**. This testing methodology is done during the development process by the software developers and sometimes QA staff.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. No. | Scenario | System Output | Expected Output | Difference | Comments |
| 1 | Module: Authenticate Twitter API  To check if the API credentials are correct | If API credentials are unauthentic, system provides error message | If API credentials are unauthentic, system provides error message | No. | Testing Completed. System Output matches with Expected Output. |
| 2 | Module: Twitter account,  To check whether username exists in the twitter | If username does not exist, the system provides error message. | If username does not exist, the system should provide error message. | No. | Testing Completed. System Output matches with Expected Output. |
| 3 | Module: Take specified number of tweets  To check whether the provided number of tweets user gets or not. | If the provided number of tweets not getting. it generates an error message. | If the provided number of tweets not getting. it generates an error message. | No. | Testing Completed. System Output matches with Expected Output. |
| 4 | Module: Checking Polarity & Subjectivity.  To check whether system gives correct output or not. | All data are correct. | All the data correct. | No. | Testing Completed. System Output matches with Expected Output. |

**6.2 System Testing:-**

System Testing is **a type of software testing that is performed on a complete integrated system to evaluate the compliance of the system with the corresponding requirements**. In system testing, integration testing passed components are taken as input.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr. No. | Scenario | Expected Output | System Output | Result(pass/fail) |
| 1 | Testing the entire system with incorrect dummy data. | The system should process the data and provide error message for the incorrect data. | System generated the error message. | Pass |
| 2 | Testing the entire system with correct dummy data. | System should process the given data and provide the correct output. | System provides the correct output. | Pass |
| 3 | Testing the entire system with correct actual data. | Should provide correct data. | Correct Analytic data provided. | Pass |
| 4 | Testing entire system using polarity & subjectivity. | All data should correct and pychart & graph plotted successfully. | All the data correct. | Pass |

**6.3 User Acceptance Test:-**

User Acceptance Testing (UAT) is **one of the last stages of the software development life cycle**. It is performed after the software has been thoroughly tested. It is sometimes known as End User Testing.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. No. | Scenario | System Output | Expected Output | Difference | Comments |
| 1 | Testing users API credential. | If API credentials are unauthentic, system provides error message | If API credentials are unauthentic, system provides error message | No. | Testing Completed. System Output matches with Expected Output. |
| 2 | Checking twitter account | If username does not exist, the system provides error message. | If username does not exist, the system should provide error message. | No. | Testing Completed. System Output matches with Expected Output. |
| 3 | Checking tweets are cleaned | If the tweets are not cleaned with all punctuation & stopwords. | If the tweets are not cleaned with all punctuation & stopwords. | No. | Testing Completed. System Output matches with Expected Output. |
| 4 | Getting the correct pychart & graph. | All data are correct. | All the data correct. | No. | Testing Completed. System Output matches with Expected Output. |

**7 Conclusion & Future-scope:-**

**7.1 Conclusion:-**

The work presented in this paper specifies a novel approach for sentiment analysis on Twitter data. To uncover the sentiment, we extracted the opinion words (a combination of the adjectives along with the verbs and adverbs) in the tweets. The corpus-based method was used to find the semantic orientation of adjectives and the dictionary-based method to find the semantic orientation of verbs and adverbs. The overall tweet sentiment was then calculated using a linear equation which incorporated emotion intensifiers too. This work is exploratory in nature and the prototype evaluated is a preliminary prototype. The initial results show that it is a motivating technique.

**7.2 Future scope:-**

The future of sentiment analysis is going to continue to dig deeper, far past the surface of the number of likes, comments and shares, and aim to reach, and truly understand, the significance of social media interactions and what they tell us about the consumers behind the screens. This forecast also predicts broader applications for sentiment analysis – brands will continue to leverage this tool, but so will individuals in the public eye, governments, nonprofits, education centers and many other organizations.