Monitoring Change in Popularity using Sentiment Analysis – *Campaign Assistant*

Final Year Project
Report

Submitted by

Vivek Rao (A043)
Yash Tibrewal (A056)
Animesh Yadav (A060)

Under The Guidance Of

Dr. Preeja Babu,
Prof. Rejo Mathew

In fulfillment for the award of the degree of

B.TECH.

INFORMATION TECHNOLOGY

At

SVKM'S NMIMS Deemed to be UNIVERSITY

MUKESH PATEL SCHOOL OF TECHNOLOGY MANAGEMENT & ENGINEERING

Department of Information Technology
Mukesh Patel School of Technology Management & Engineering
NMIMS (Deemed –to-be University)
JVPD Scheme Bhaktivedanta Swami Marg,
Ville Parle (W), Mumbai-400 056.

March, 2019

CERTIFICATE

This is to certify that the project entitled “Monitoring Change in Popularity using Sentiment Analysis- Campaign Assistant“ is the bonafide work carried out by Vivek Rao, Yash Tibrewal & Animesh Yadav of B.Tech (IT), MPSTME, Mumbai, during the VIII Semester of the academic year 2018-2019, in complete fulfillment of the requirements for the award of the degree of Bachelors of Technology as per norms prescribed by NMIMS. The project work has been assessed and found to be satisfactory.

(Signature of Internal Mentor 1)
Name: Dr. Preeja Babu
Designation:Assistant Professor

(Signature of Internal Mentor 2)
Name: Rejo Mathew
Designation: Assistant Professor

(Signature of External Examiner)
Name:
Designation:

HOD (IT)

(Prof.Pintu Shah)

Dean

(Dr. N T Rao)

DECLARATION

We, Vivek Rao, Yash Tibrewal & Animesh Yadav roll numbers: A043, A056, A060 respectively, understand that plagiarism is defined as any one or the combination of the following:

1. Uncredited verbatim copying of individual sentences, paragraphs or illustrations (such as graphs, diagrams, etc.) from any source, published or unpublished, including the Internet.
2. Uncredited improper paraphrasing of pages or paragraphs (changing a few words or phrases, or rearranging the original sentence order)
3. Credited verbatim copying of a major portion of a paper (or thesis chapter) without clear delineation of who did or wrote what. (Source: IEEE, The Institute, Dec. 2004)

We have made sure that all the ideas, expressions, graphs, diagrams, etc., that are not a result of our work, are properly credited. Long phrases or sentences that had to be used verbatim from published literature have been clearly identified using quotation marks.

We affirm that no portion of our work can be considered as plagiarism and we take full responsibility if such a complaint occurs. We understand fully well that the guide of the seminar/project report may not be in a position to check for the possibility of such incidences of plagiarism in this body of work.

Name: Vivek Rao
Roll No.: A043
Date:
Signature:

Name: Yash Tibrewal
Roll No.: A056
Date:
Signature:

Name: Animesh Yadav
Roll No.: A060
Date:
Signature:

ACKNOWLEDGEMENTS

It is indeed a moment of great satisfaction and immense pleasure to express our profound gratitude to all those who have contributed in the development of this project. The careful supervision, critical suggestions and encouragement by Dr. Preeja Babu and Prof. Rejo Mathew, Department of Information Technology, SVKMs NMIMS Mukesh Patel School of Technology Management & Engineering has been a source of immense help.

The Department of Information Technology and its Faculty members have provided the support and equipment that has led to the successful completion of this project.

- Vivek Rao (A043), Yash Tibrewal (A056), Animesh Yadav (A060)

TABLE OF CONTENTS

List of Tables

List of Figures

1. Overview

1.1 Project Specification

Sentiment Analysis, also known as opinion mining, is defined as determining the polarity of a user generated opinion. Text Mining incorporates techniques from data mining, machine learning and natural language processing. It helps to judge the attitude and the sentiment towards the subject. The concept of sentiment analysis has broad use cases including comprehending the product review, analysing public sentiment on an issue, etc. The sentiments of the users can be analysed by using machine learning techniques (Supervised & Unsupervised), lexicon based, keyword and concept based approaches. This particular section introduces one of the use cases which shall be implemented in this project. In recent years, Twitter as a micro-blogging portal has become ubiquitous and important for social networking, content sharing and posting opinions. As clearly stated in [1], Twitter's network of 465 million users generate about 175 million tweets every single day. From an individual's perspective, the significance of user-generated data analysis might not be of interest. The knowledge which is acquired provides an organization a competitive advantage over its competitors by making informed decisions. One of the applications related to the theory of sentiment analysis lies within politics. Almost every social-media user posts his/her opinions and voice which revolves around a political party and/or candidate. Hence, deducing the polarity of the user-generated messages, reviews, posts and opinions will provide a political party useful strategy to drive its election propaganda.

This project intends to provide a holistic view of public sentiment on prospective election candidates who will contest the *Lok Sabha General Elections,2019* by deducing the polarity of user-generated tweets. Hence, Twitter is being utilized as the primary source of data acquisition. The same also applies for mathematically computing the popularity of respective political parties over a defined time-frame among the Twitter users. The primary purpose of this project is that it is equipping the end-user with a tool/product for analyzing the tweets and gain knowledge in the form of data visualization. The project will demonstrate as to how the power of Natural Language Processing, Machine Learning and Data Visualization can be used together for the benefits of the Public Relations (PR) departments of various political parties. The project demonstrates as to how social-media content can be used to predict real-world outcomes.

1.2 Literature Review

The existing techniques or approaches tend to identify sentiments or polarity of the opinions by applying pre-processing techniques on the text without any regard to the context. They all tend to be mostly text based approaches and have different levels of accuracy and usage. The techniques that we have surveyed were Keyword spotting, Lexical Affinity, Concept-based approaches, Sentiment diffusion model, Binary sentiment classification, Stop-words removal and Review scoring.

Key-word Spotting- [2] This is one of the most popular approaches used in Sentiment Analysis. The reason why it's so popular is because of its accessibility and economy. Now it involves the economic factor because the setup for this type of approach is very minimalistic. This approach classifies text based on affect categories. The affect categories are the emotions attaches with a word such as happy, sad, afraid and bored. But Key-word Spotting is weak in two areas. It can't reliably recognize negated sentences and relies heavily on surface features. For example it can correctly classify the sentence "today was a happy day" as a positive statement but it's likely to classify the sentence "today wasn't a happy day at all" as same. Also this approach is good for small set of statements only.

Lexical Affinity- [2] This approach also detects words like the keyword spotting technique but also it assigns the words a probable affinity towards particular emotions. For example Lexical Affinity might assign the word "accident" a 75% probability of indicating a negative sentiment. These affinities are assigned by studying or training from a large dataset of linguistic corpora. The Lexical Affinity often outperforms the Keyword Spotting technique but there are some attached weaknesses to this approach. Just like the Keyword spotting method, negated sentences can trick this approach. As well as Lexical Affinity is somewhat domain dependent. It all depends on the dataset that was used to assign the affinity to the words for Sentiment Analysis.

Concept-based approaches [2] These approaches make the use of Web ontologies or semantic networks to accomplish semantic text analysis. This is a self-learning algorithm where in given a particular language the algorithm can outperform the previous two approaches. The accuracy in the start is very less because of the limited learning that the algorithm has done. But given enough time the algorithm starts predicting more accurately. This is because with the help of the Web ontologies this approach takes into account the different usage of words in different scenarios. But as efficient and good this method is, with it come some drawbacks. Firstly it requires the Web ontologies to be very deep and extensive in nature. A poor ontology will directly reduce the accuracy of the sentiments extracted from the sentences. Also this algorithm is language dependent, i.e. Web Ontology can be prepared for only a single language.

Sentiment diffusion model – [3] This approach is limited to Twitter platform. What this approach does is basically detects opinion from the tweets that people post. Topics such as “Who will be the next Prime Minister” are studied on Twitter. Based on people’s tweets the researchers predict the output. Then the real world output is being matched with the predicted output. Also it detects even immediate neighbors tweets for more accurate prediction. What this means is it will also search for tweets from your friends on the same topic and then form a judgment. This is done to take into account the crowd mentality that sometimes landslides the judgments. This is a very effective technique in mining opinions but there are some cons attached to it. For this approach to work very effectively a very large data set is required. Also a quite significant number of neighboring tweets should be available

Binary sentiment classification [5] This approach comes under Sentiment Analysis. It’s a very basic and quick approach to get peoples opinion on particular topic. It assigns a positive or negative influence to the words. For example words such as “like” and “dislike” are used to get feedbacks on products or announcements. The featured words have a sentiment polarity associated with them. For Facebook platform it accurately predicts if people like or dislike a trend. But the major drawback for this approach is that it can’t detect neutral sentiments.

Stop-words removal [6] Stop-words removal is a bit different technique compared to the above techniques. In this technique the algorithm removes the frequently used words like “the”“a” so that the dataset to be analyzed is made small. The ideology behind this approach is that the frequently used word are usually meaningless. This approach can be very good for datasets which are very lengthy in nature and the sentiments contained in it are less. But it fails to take into account that some frequently used words may contain important sentiments that might be missed out.

Following is a table summarizing the concepts associated –

1.3 Comparative Study & Analysis

Comparative Analysis

Technique	Key-Features	Key Assumptions
Keyword spotting [2]	Classifies text based on affect categories	The dataset has only few words with sentiments associated to it
Lexical Affinity [2]	Assigns arbitrary words a probable affinity	Dataset is at least paragraph length and sentiment assignment to words is always same
Concept-based approaches [2]	The algorithm is self-learning	Web-ontologies used are vast and correct for a given particular language
Sentiment Diffusion model [3]	The model can be adapted for any topic	Neighboring tweets are available for every tweet
Binary Sentiment classification[5]	Assigns positive or negative influence to the words	The feature words have a sentiment polarity associated with them
Stop-words removal[6]	Removes frequently used words	The frequently used words are meaningless usually

Scheme	Pros	Cons	Analysis
Keyword Spotting [2]	Accessibility and Economy	Cant recognize negated words and depends on surface features	The most basic algo for sentiment analysis with least accuracy

Lexical Affinity [2]	It outperforms keyword spotting	Negated sentences sometimes trick & its biased towards text of particular genre	There's not much difference between key-word spotting and this algo. Assigning probabilities depends upon the domain used
Concept-based approaches [2]	The algorithm takes into account the usage of words in different scenarios	Requires a very deep Web Ontology and the accuracy is poor in start	A good method because the algorithm keeps on learning
Sentiment Diffusion Model [3]	Detects even immediate neighbors tweets for more accurate prediction	Very extensive dataset needed	Even though the authors claim it to be so accurate, it was because of the common topic that they chose. This may fail to work on uncommon topics or with limited tweets.
Review Scoring [4]	Reviews each and every word with its placement in the sentence	Fails to work if majority of words are not found in the dictionary	This algorithm is highly system dependant. Its not universal
Binary Sentiment Classification [5]	Good for very short reviews with few words.	Cant recognize neutral sentiments or negated sentiments	This method seems not useful for sentiment analysis because sentiments are of various types
Stop-words removal [6]	Makes the dataset to be analyzed small	It may remove words with frequent sentiments	This maybe useful for long dataset but removing words from tweets isn’t a good idea.
N-Gram preprocessing technique [7]	makes predictability easy	Highly dependent on the dataset used	Predictibilty adds for more unknown factors

Table I

As per the insights gained from the literature survey, it was concluded to employ the Binary Classification method which incorporates an opinion classifier that follows the Machine Learning approach. A tweet is limited to 140 characters. Therefore, a random tweet might contain many words (hence, many features) used in an informal English language. If a word is taken and analyzed independently, its true meaning may contribute negatively to the given context. In order to preserve the contextual meaning of each and every word in a sentence/opinion, it is intended to employ a Machine Learning approach to solve the problem. Dictionary based approaches on the other hand do not consider the influence of the context in which a sentence has been written. As the number of features is large in our dataset (both test data and training data), it was concluded to utilize Support Vector Machine (SVM) as the opinion classifier.

1.4 Selection Criteria of the Classifier

After having critically analyzed the above mentioned research literature (papers and a journal), it was realized that the accuracy of the project should reach approximately 80% in order to successfully deploy the end-product. Various algorithms have been employed by the authors in different as well as same context. The repercussions of using an inappropriate classifier for sentiment analysis are significant. Therefore, it was gradually comprehended that in this case, the number of features is significantly more than the problem instances itself. Moreover, this project deals mainly with textual data rather than numeric data. Taking all this into consideration, it was concluded with proper theoretical justification that Support Vector Machine outperforms the remaining classifiers in this particular context.

1.5 Comprehension & Culminating Efforts

In order to accurately implement the algorithm, Support Vector Machine (SVM) was studied and understood. SVM is a supervised machine learning approach used to build linear non-probabilistic binary classifier. Hence, in this case, the tweets will be classified into either positive or negative. Since it is a supervised algorithm, it must learn from an existing labeled corpus or training data. As the number of features is significantly higher than the number of instances in this project’s case, the hyper-plane will be used to classify the data points.

2. Analysis & Design

2.1 Requirement Analysis

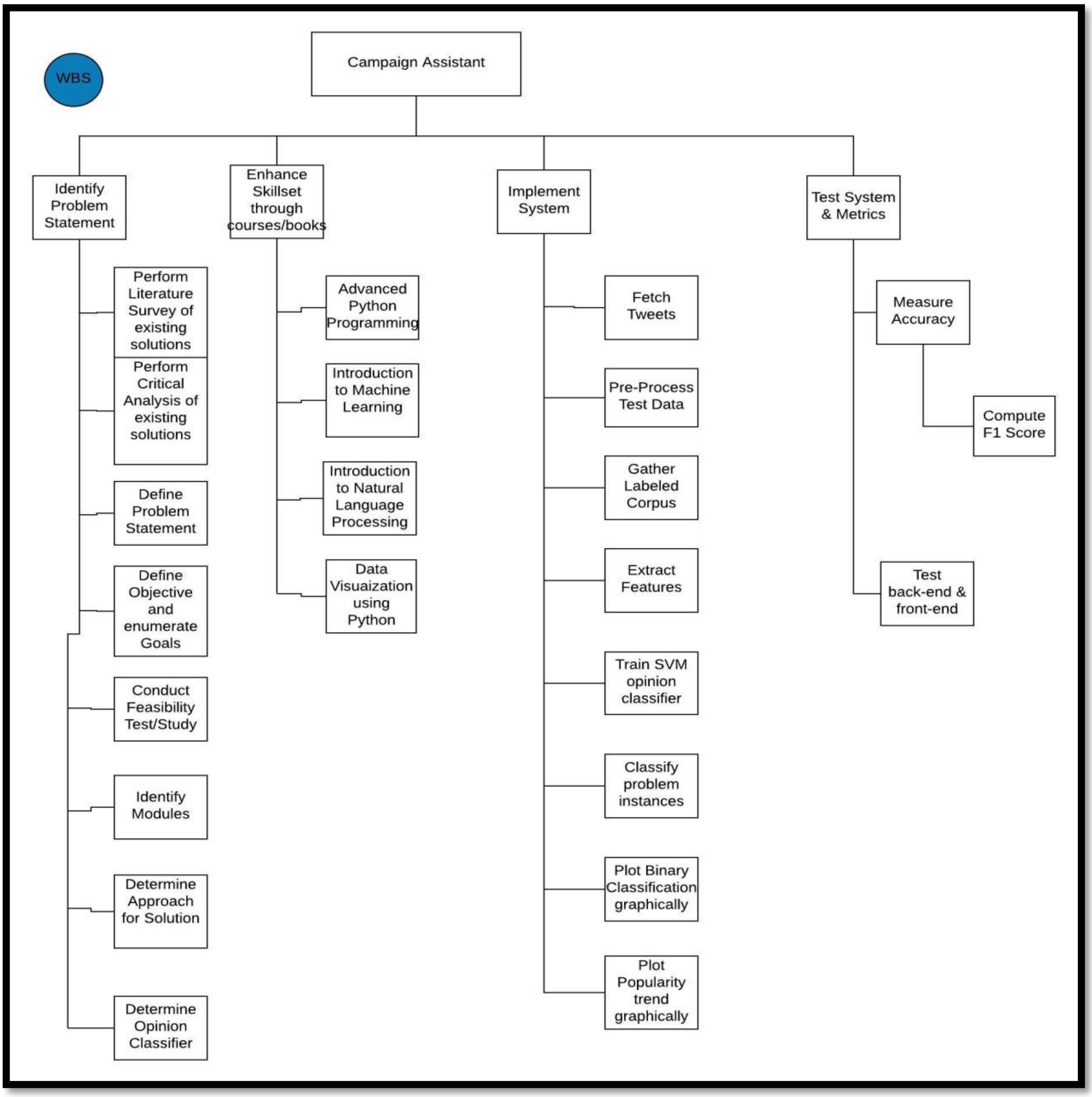

WBS
Campaign Assistant
Identify Problem Statement
Enhance Skillset through courses/books
Implement System
Test System & Metrics
Perform Literature Survey of existing solutions
Perform Critical Analysis of existing solutions
Define Problem Statement
Define Objective and enumerate Goals
Conduct Feasibility Test/Study
Identify Modules
Determine Approach for Solution
Determine Opinion Classifier
Advanced Python Programming
Introduction to Machine Learning
Introduction to Natural Language Processing
Data Visuaization using Python
Fetch Tweets
Pre-Process Test Data
Gather Labeled Corpus
Extract Features
Train SVM opinion classifier
Classify problem instances
Plot Binary Classification graphically
Plot Popularity trend graphically
Measure Accuracy
Compute F1 Score
Test back-end & front-end

Fig. 1 Work Breakdown Structure

As depicted in the Work Breakdown Structure (WBS) diagram, the initial few weeks (i.e. August,2018 onwards) were spent in performing the literature review. The exact problem statement was defined after having critically analyzed multiple research papers and journals. *IEEE Xplore* was the primary source of literature database as reliability and quality of work was the primary concern. A keyword based search led to multiple existing solutions / literature on this particular topic

After having the problem statement clearly defined, the approach to solve it was being looked upon. Various existing solutions as mentioned above were analyzed. In order to handle the context of the tweet, machine learning approach was justified. Among the machine learning algorithms, SVM outperforms the rest in case of political tweets as mentioned in [1]. Simultaneously, online courses were taken up by the members of the team in order to enhance the skill set which will be required for the implementation phase. This includes – *programming in python, Introduction to Natural Language Processing with NLTK and Introduction to Machine Learning.*

The next set of steps/works falls within the implementation phase of the project. The implementation will proceed linearly and gradually by the implementation of the required features and modules. Every module integrated will be comprehensively tested individually as well as undergo integration testing. During the last phase of the project, the back-end will be integrated with an appropriate GUI of the end-product. The accuracy measure of the computation will also be conducted using standard like F-Score

2.2 Feasibility Study

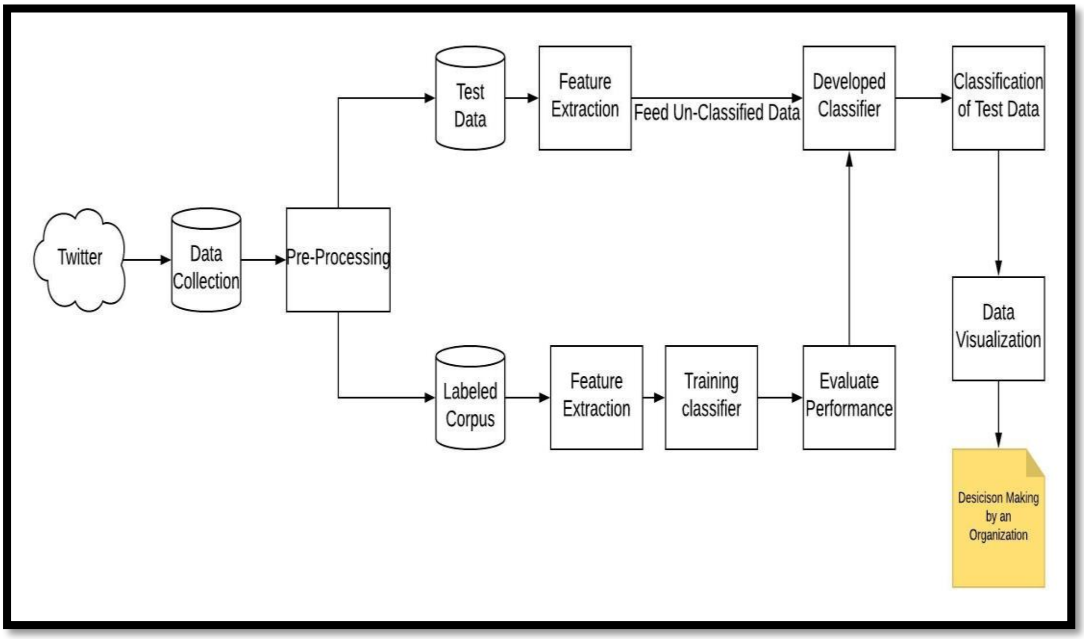


Fig. 2 Machine Learning Approach

As depicted in the flowchart, this particular project entails the following phases – *Data Acquisition (Test Data), Pre-Processing (Test Data), Training Corpus Acquisition, Pre-Processing (Training Data), Feature Extraction(test & training corpus), Training of classifier on pre-labeled Corpus, Classification using opinion classifier, Plotting of Popularity as per the subject, Verification & Validation (i.e. Accuracy) and development of an appropriate GUI.*

Since the theme of this project revolves around social media, the project starts of by fetching tweets from the micro-blogging portal – Twitter. It is highly important to note that only relevant datasets are to be fetched rather than irrelevant samples. Therefore, the fetched datasets should be represented by a search-term or a keyword of desire/interest. The collected tweets about a particular search term/keyword will be referred to as Test Data as the labels need to be computed at run-time. The tweets need to undergo pre-processing and cleaning in order to remove inconsistencies and noise. Typical instances include removal of stop words, punctuation marks, “#”/”@”symbols, emoticons as well as tokenization and conversion into lowercase. Unnecessary repetition of characters which users supply in order to demonstrate impact should also be taken care of.

This project follows the Machine Learning approach where a particular algorithm is first trained on a pre-labeled Corpus and then run on problem instances (i.e test data) in order to perform the binary classification. Such a classifier is referred to as a Supervised classifier as training of the model is necessary before running it on the actual unknown test data (problem instances) Hence, a pre-labeled Corpus will be referred and the suitable features will be extracted. In this project, the number of features will be more than the number of problem instances. Hence, Support Vector Machine (SVM) will outperform the rest of the Machine Learning classifiers in this context [1]. After having the SVM (Support Vector Machine) trained, the test problem instances will undergo binary classification. The statistical results will then be used to mathematically compute the popularity for the political parties and candidates and plotted/visualized accordingly.

2.3 Technology and Software/Language/Tool Details

Technology utilized – This particular project implements the machine learning technology. As stated in the literature review section, various specific solutions are available for the problem of opinion mining. However, the suitability of a particular solution depends on the context in which the problem is being solved. The context for this particular project implementation is the Indian politics. As the data being fetched is from a social media platform, the number of features will be much higher than the total number of problem instances. Hence, Support Vector Machine is an ideal choice for the visualization of popularity.

The Machine Learning approach within Sentiment Analysis comprises of a training stage and the computation stage. The training stage involves gathering of a labeled corpus for the purpose of leaning. Here, the learning stage comprises of generation of a feature space. Already labeled corpus exist which can be utilized. Typical examples include, *Niek Sanders-5000 labeled tweets*. Here, already fetched tweets that fall in the context/domain of technology are labeled as positive, negative and neutral. Since the domain of an already available open-source corpus is different, this project required the generation of newly hand annotated corpus for the purpose of training and learning of the algorithm.

The first step in the generation of labeled corpus is fetching of relevant tweets that are related to Indian political context. "Python Twitter" module was utilized to create an API object which can make use of its search function to fetch a maximum of 100 tweets per 15 min. The function takes in the search string as an argument. All the tweets are contain that particular keyword are returned in the form of a list. In order to store the returned tweets in a file, the tweets are directed in an excel file with columns like tweet, polarity and number. Once a suitable amount of data was generated which included tweets related to political parties and candidates, human intervention was required to manually annotate the individual tweets as positive or negative. Here, the context plays a very crucial role , especially in the political context. A positive sentiment on the particular issue for a BJP supporter might not be considered as positive by the other member/supporter of the same party. As a neutral user, the generated tweets were manually annotated as positive or negative considering the current scenario of Indian politics.

As the collected tweets comprised of noise such as *multiple characters , capital letters, hashtags, stop words such as 'is, am, the , a,' , links to other websites or articles* , the data needed to undergo pre-processing. Here, pre-processing includes tasks such as converting into lower case, removal of stopwords, removal of links and tokenization. For the tasks such as lowercase, removal of url links, removal of hashtags, the regular expression was utilized. A RegEx, or Regular Expression is a sequence of characters that forms a particular search pattern. RegEx can be used to check if a given string contains the specified search pattern which can be then further computed. The python language has a pre-defined package called 're' which can be then used for the purpose of extracting specific patterns within strings of tweets. For the remaining of the tasks such as removal of stopwords and tokenization, the NLTK tooklit was utilised.

SVM (Support Vector Machine) is an algorithmic solution available under the machine learning set. The following section describes the mathematical foundation behind the working of this solution. SVM (Support Vector Machine) represents points in an n-dimensional hypercube. A hypercube is a 3-Dimensional space in geometry. A hyper-plane is also a mathematical and a geometric shape. It has (n-1) dimensions where 'n' represents the total number of features available in the feature space/set. Each point on hyper-plane is represented by (n-1) co-ordinates. The thickness of the hyper-plane does not exist in 1-Dimension.

Consider the following equation. These points define the hyper-plane,

'$Ax + By + Cz= D$'. All the points will satisfy the equation. All points on one side of the hyper-plane will satisfy the condition, $Ax + By + Cz< D$ and all points on the other side of the hyper-plane will satisfy the condition, $Ax + By + Cz> D$. During the computation, the incoming problem instances are passed into the linear function in order to be classified. Here, the probability of the feature is not assumed. The choice of the linear function/ hyper-plane depends on the distance of point from the hyper-plane. For example, the data point , X=(X1,Y1,Z1). The distance of the data point to the hyper-plane is defined *by* $[Ax+By+Cz-D]/[A^2+B^2+C^2]^{1/2}$. The best hyper-plane maximizes the sum of distance of nearest points on either side. The typical solution is provided by Maximum Margin hyper-plane. Challenge arises when the points are not linearly separable. With the provision of kernel trick, the soft margin method finds a hyper-plane that does as clean a separation possible. Soft margin method finds the best hyper-plane with few errors. The feature vector is transformed into a higher dimension space. In the context of machine learning, a kernel is a function that takes two points , uses a non-linear function (dot product in higher dimension) to classify the test problem instances.

The mathematical working of the Support Vector Machine (SVM) can be described as follows. First, a vocabulary is generated. A vocabulary here refers to the set of all distinct features that are extracted from the trained corpus. A feature in this context resembles a particular 'word' from an individual string/tweet. For example- If 100 tweets are present in the corpus, every word (distinct) that exists is pushed into the feature space. That becomes the vocabulary of distinct features. Next step is to generate the feature vector for each individual problem instance that needs to be classified into one of the classes – positive or negative. For every individual tweet, a feature vector is generated during run-time. Here, the feature vector can be defined as – {[feature1], [feature2], [feature3], …..[featureN]}. Every element in this vector indicates the presence or absence of that particular feature. In case of presence, '1' is defined or else '0'. Similarly, every incoming problem instance has a corresponding vector associated with it. The specification of the feature vector is by default 'uni-gram'. Hence, only a single feature is represented by a single element. In case of 'Bi-gram', the feature vector comprises of a pair of features represented by a single vector element. As per the literature review conducted, the highest accuracy obtained is via the implementation of 'uni-gram'. Upon the generation of feature vectors, the linear kernel accepts the vector, runs the mathematical computation and then places the data point in either side of the hyper-plane , .i.e. positive or negative. Support vector machines (SVMs) are a set of supervised learning solutions used for classification. The advantages of support vector machines are:

- Effective in high dimensional spaces where there is a large feature space
- Still effective in cases where number of dimensions is greater than the number of samples.
- Uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.
- Versatile: different Kernel functions can be specified for the decision function. Common kernels are provided, but it is also possible to specify custom kernels.

The disadvantages of support vector machines include:

- If the number of features is much greater than the number of samples, avoid over-fitting in choosing Kernel functions and regularization term is crucial.
- SVMs do not directly provide probability estimates, these are calculated using an expensive five-fold cross-validation

The Sci-Kit learn library is utilised for the purpose of deploying the SVM solution. It is a simple and efficient tool for data mining and data analysis. It is easy to access and reusable in various contexts. It is built upon NumPy, SciPy and matplotlib. Also, its open-source nature makes it very easy to use for solving machine learning problems.

2.4 Design & Development

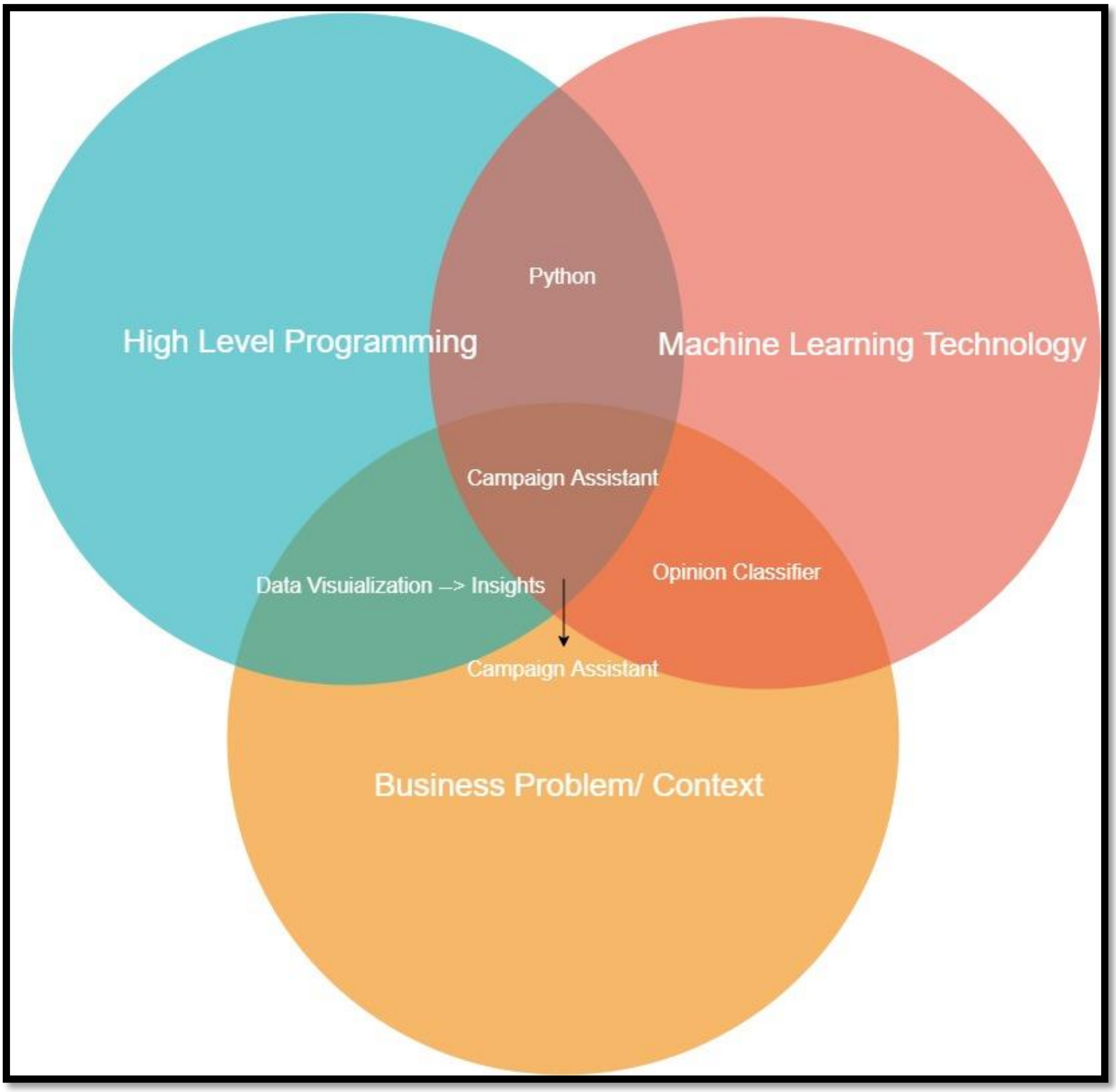

Python
High Level Programming
Machine Learning Technology
Campaign Assistant
Data Visuialization –> Insights
Opinion Classifier
Campaign Assistant
Business Problem/ Context

Fig. 3 Project Abstract Venn diagram

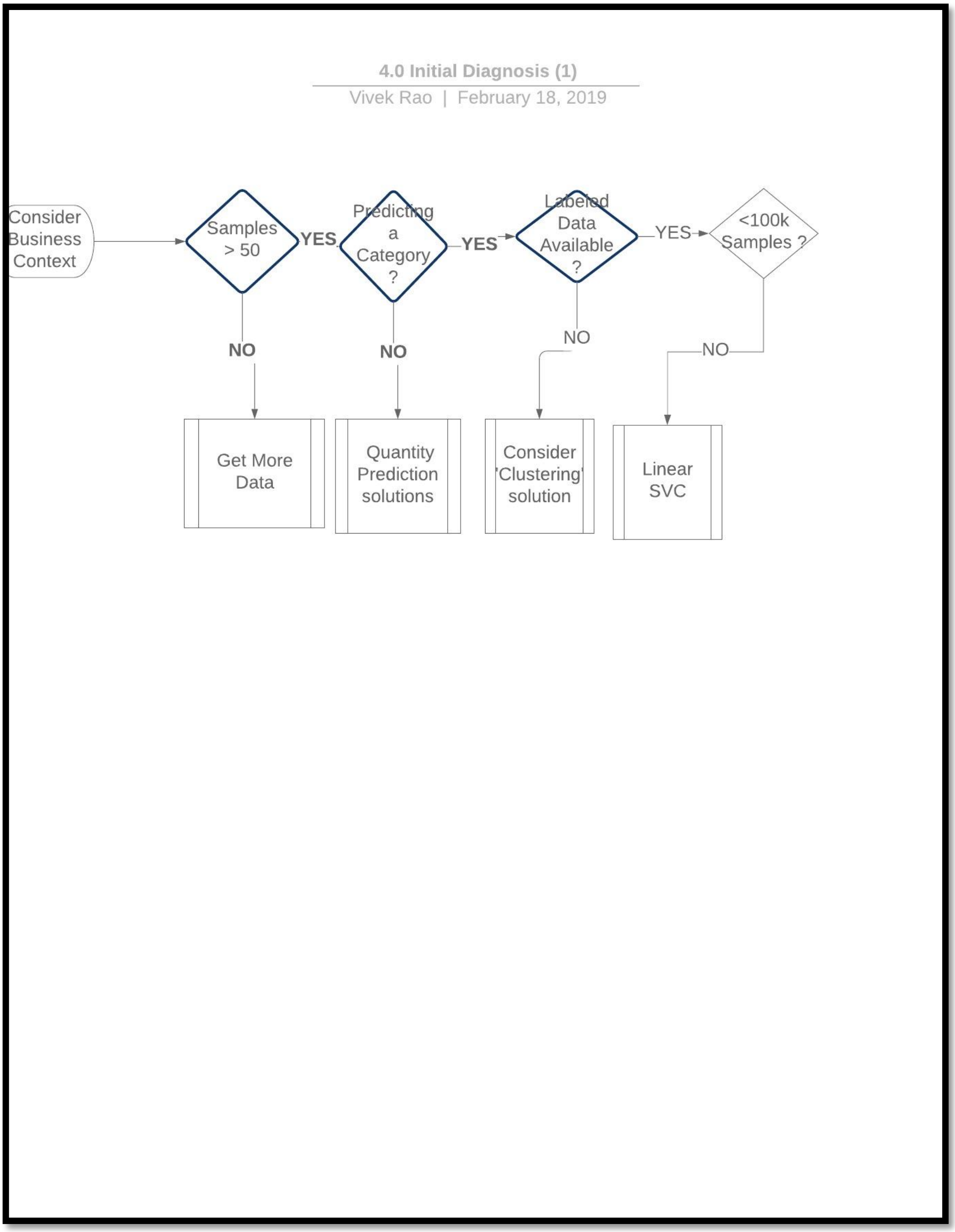

4.0 Initial Diagnosis (1)
Vivek Rao | February 18, 2019
Consider Business Context
Samples > 50
YES
Predicting a Category ?
YES
Labeled Data Available ?
YES
<100k Samples ?
NO
NO
NO
NO
Get More Data
Quantity Prediction solutions
Consider 'Clustering' solution
Linear SVC

Fig 4 Sci-Kit learn Algorithm flowchart

-

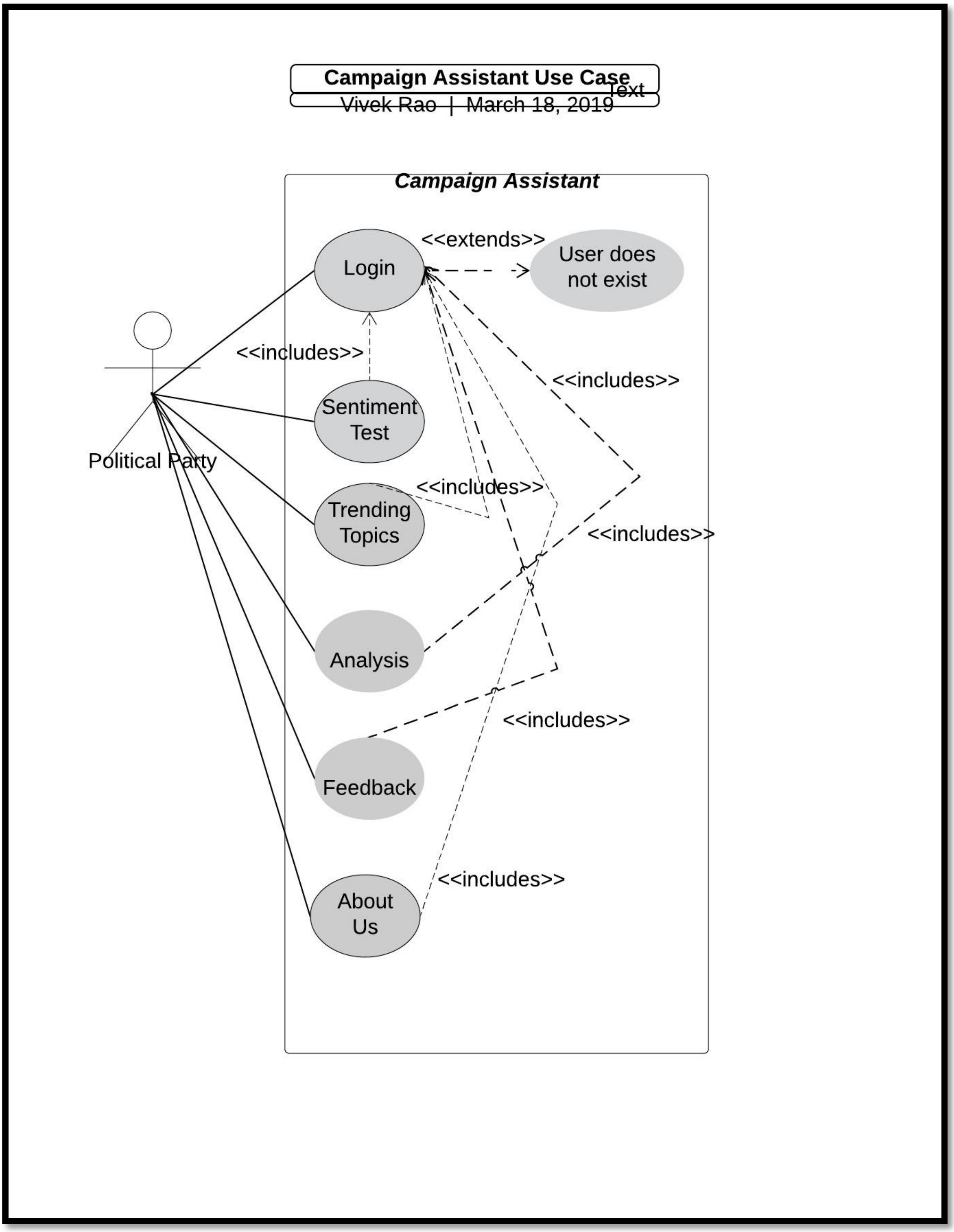


Fig. 5 Use Case Diagram

A sequence diagram is a type of interaction diagram because it describes how-and in what order-a group of objects works together.

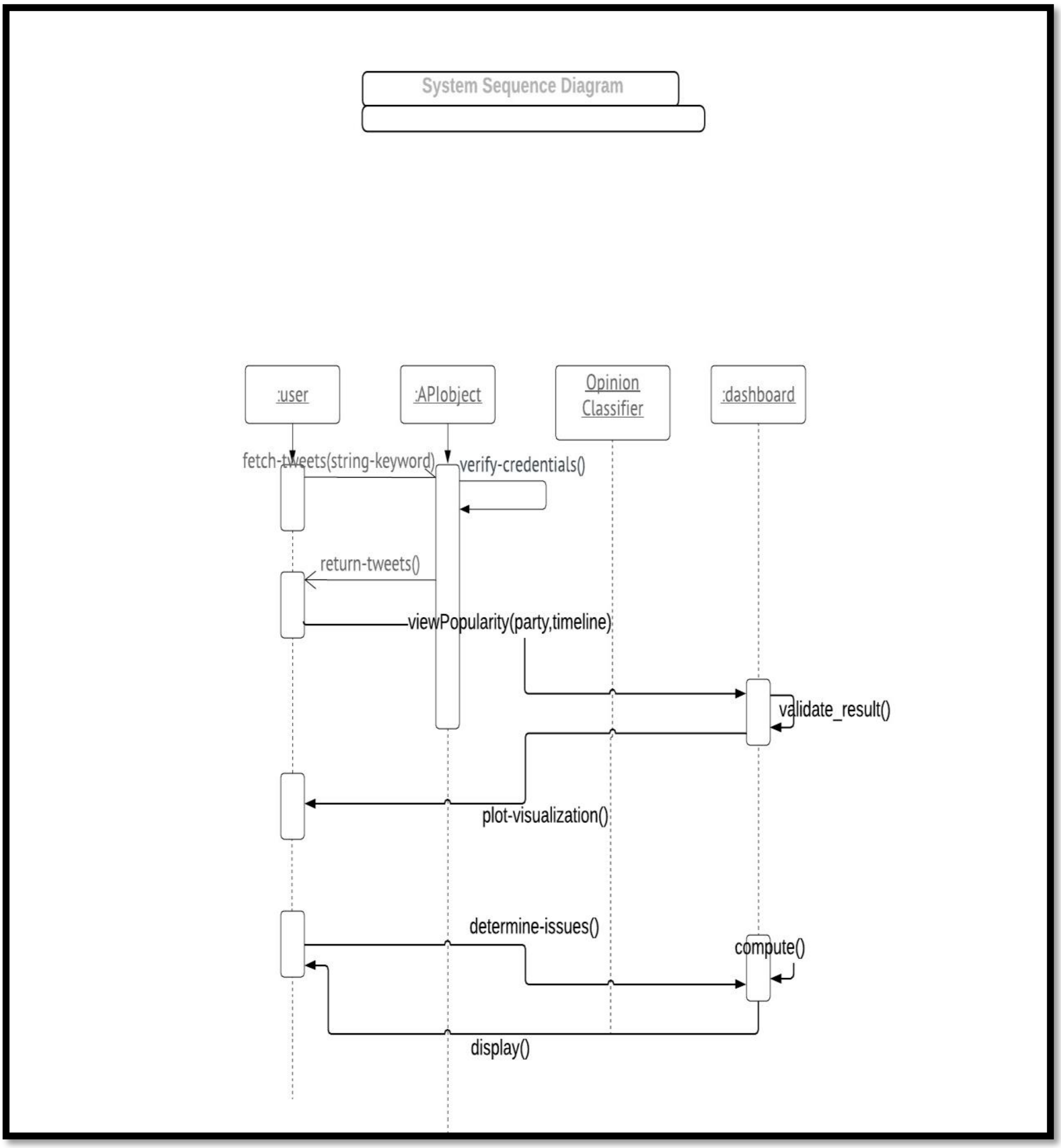


Fig. 6 Sequence Diagram

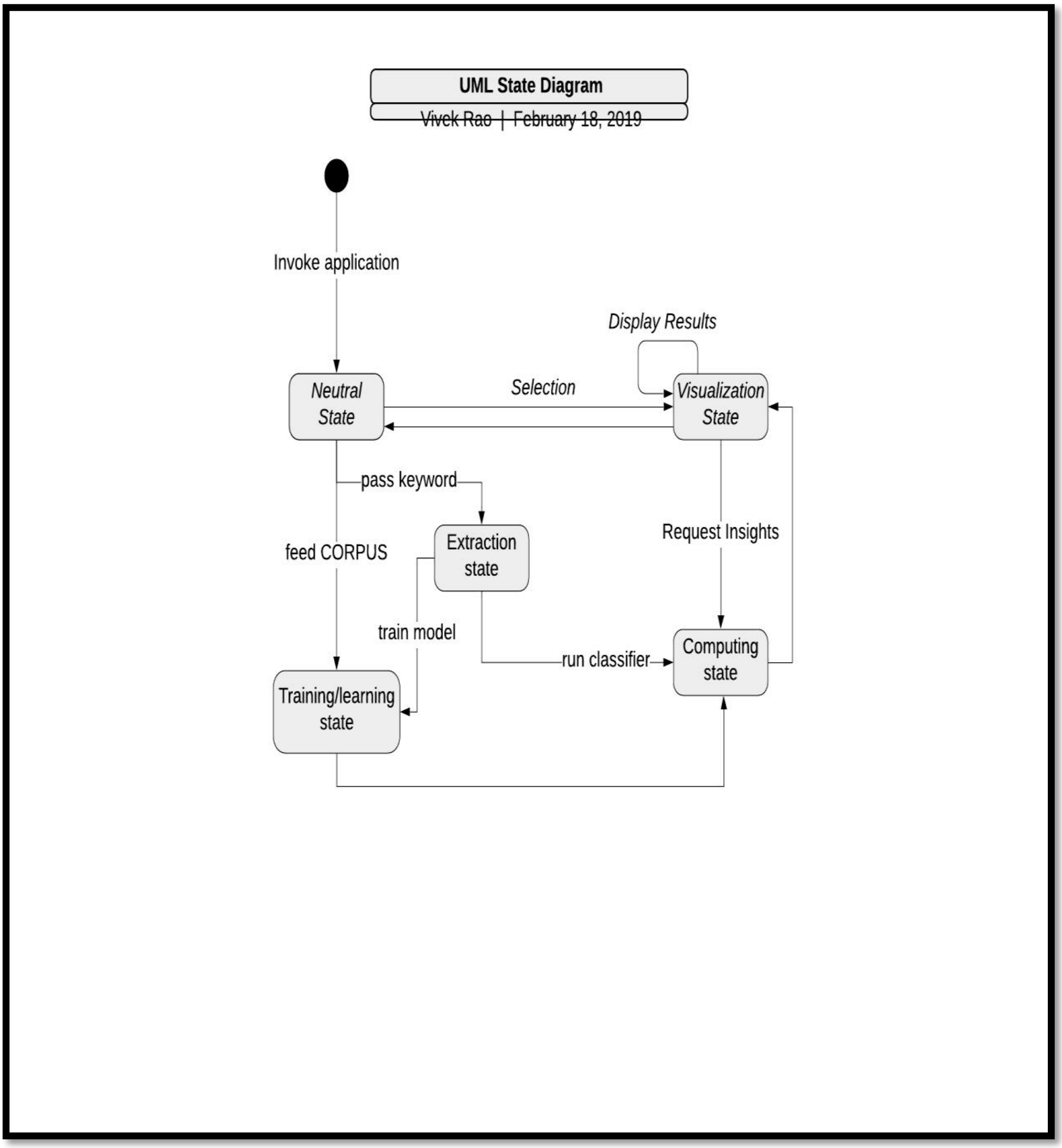


Fig. 7 State Diagram

The following data flow diagram (DFD) is of the type – 'physical'. This particular DFD is a physical DFD, which shows how the system is currently (or will be) implemented. For example, in a logical DFD, the processes would be programs and manual procedures rather than business events, which is what a logical DFD focuses on. DFDs use defined symbols and shapes to show data inputs, outputs, storage points, and the routes between each destination. This diagram uses the Gane and Sarson symbols.

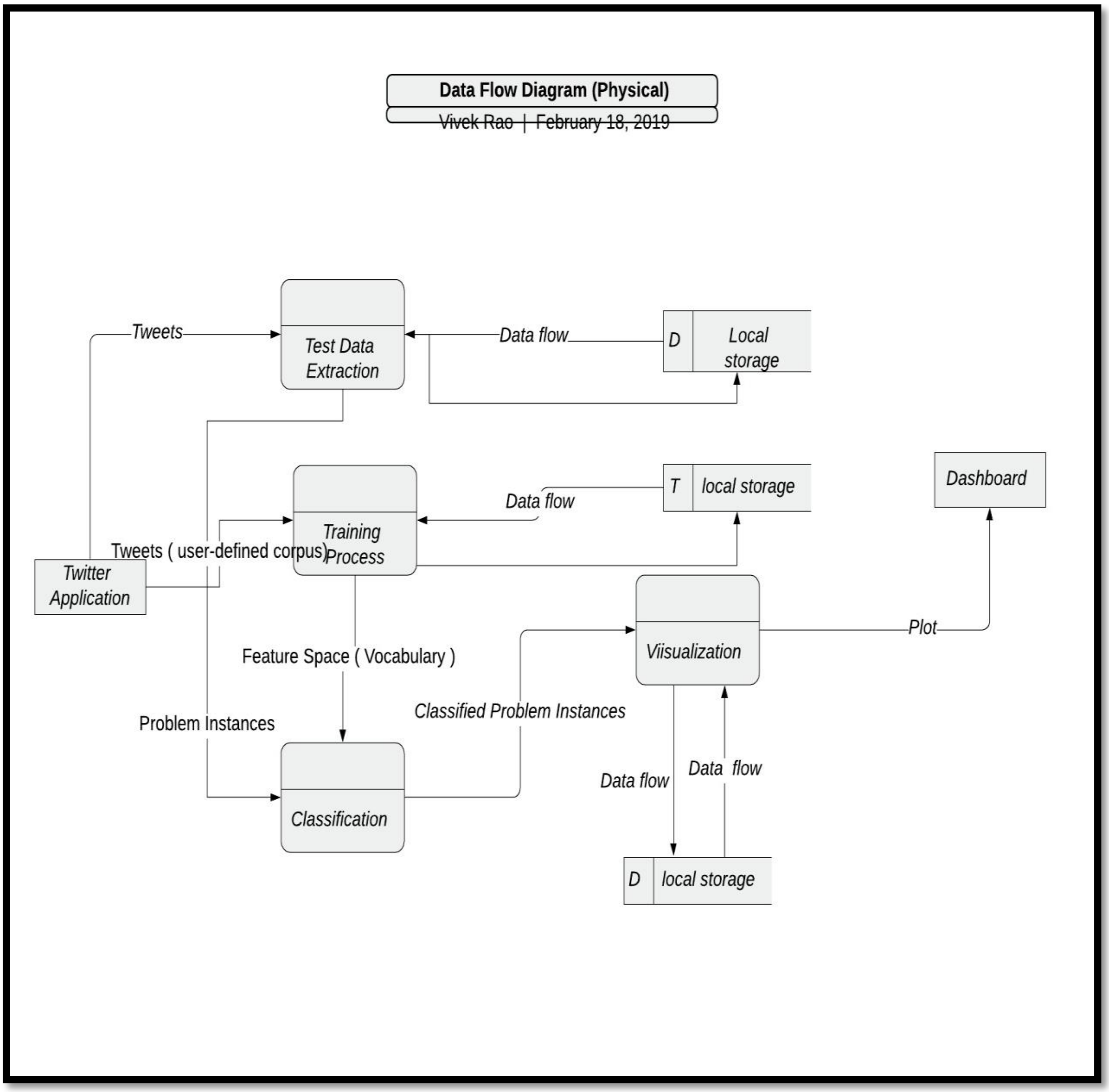
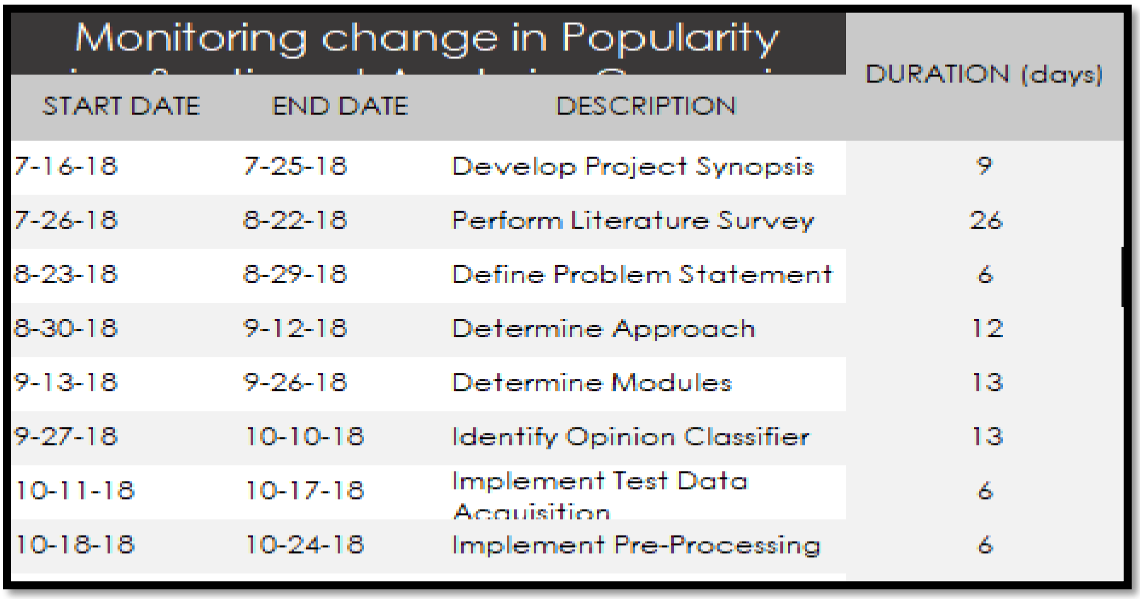


Fig. 8 Data flow diagram

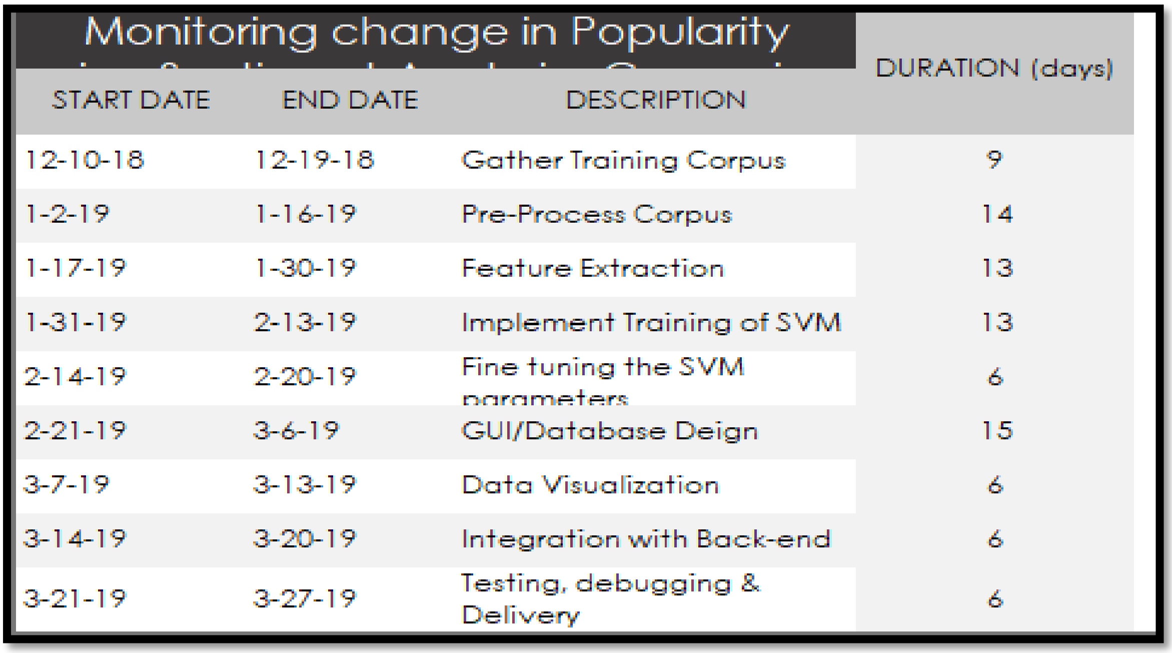
2.5 Project Planning

Fig. 9 Gantt chart



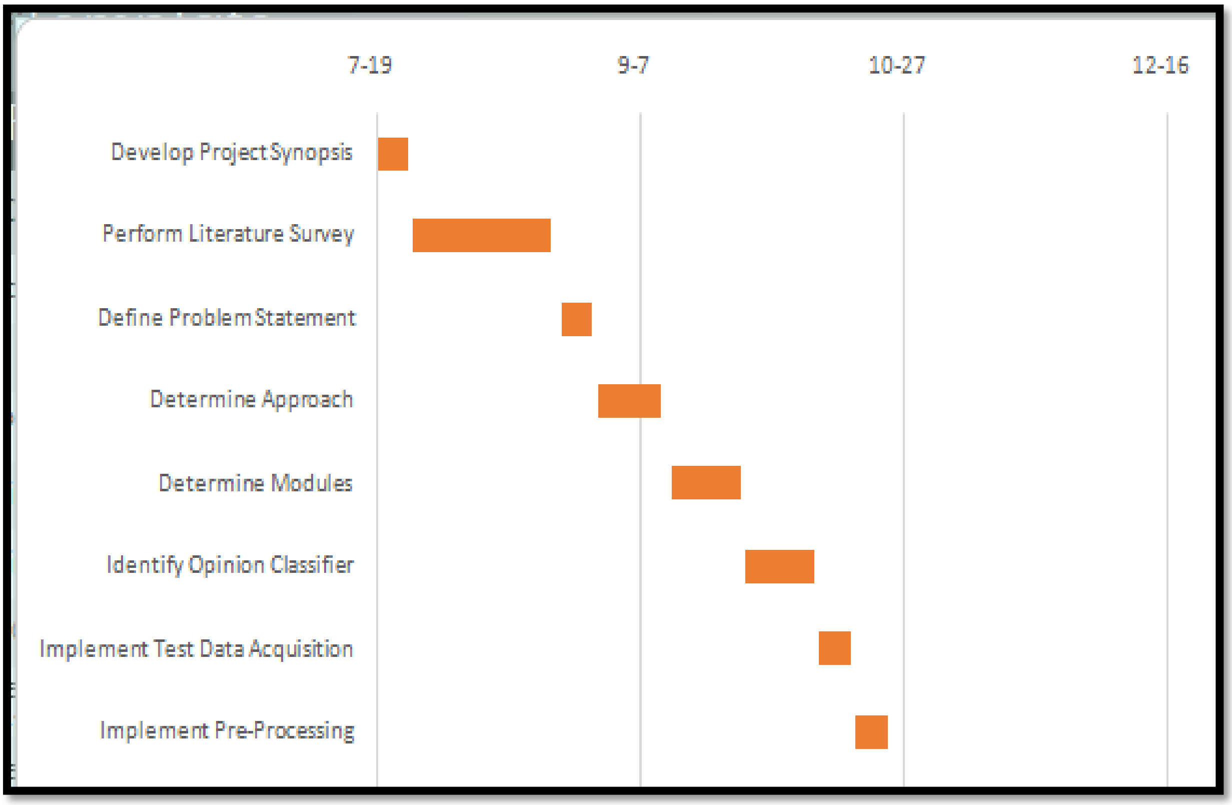
Monitoring change in Popularity

START DATE	END DATE	DESCRIPTION	DURATION (days)
7-16-18	7-25-18	Develop Project Synopsis	9
7-26-18	8-22-18	Perform Literature Survey	26
8-23-18	8-29-18	Define Problem Statement	6
8-30-18	9-12-18	Determine Approach	12
9-13-18	9-26-18	Determine Modules	13
9-27-18	10-10-18	Identify Opinion Classifier	13
10-11-18	10-17-18	Implement Test Data Acquisition	6
10-18-18	10-24-18	Implement Pre-Processing	6



Monitoring change in Popularity

START DATE	END DATE	DESCRIPTION	DURATION (days)
12-10-18	12-19-18	Gather Training Corpus	9
1-2-19	1-16-19	Pre-Process Corpus	14
1-17-19	1-30-19	Feature Extraction	13
1-31-19	2-13-19	Implement Training of SVM	13
2-14-19	2-20-19	Fine tuning the SVM parameters	6
2-21-19	3-6-19	GUI/Database Deign	15
3-7-19	3-13-19	Data Visualization	6
3-14-19	3-20-19	Integration with Back-end	6
3-21-19	3-27-19	Testing, debugging & Delivery	6


7-19
9-7
10-27
12-16
Develop Project Synopsis
Perform Literature Survey
Define Problem Statement
Determine Approach
Determine Modules
Identify Opinion Classifier
Implement Test Data Acquisition
Implement Pre-Processing

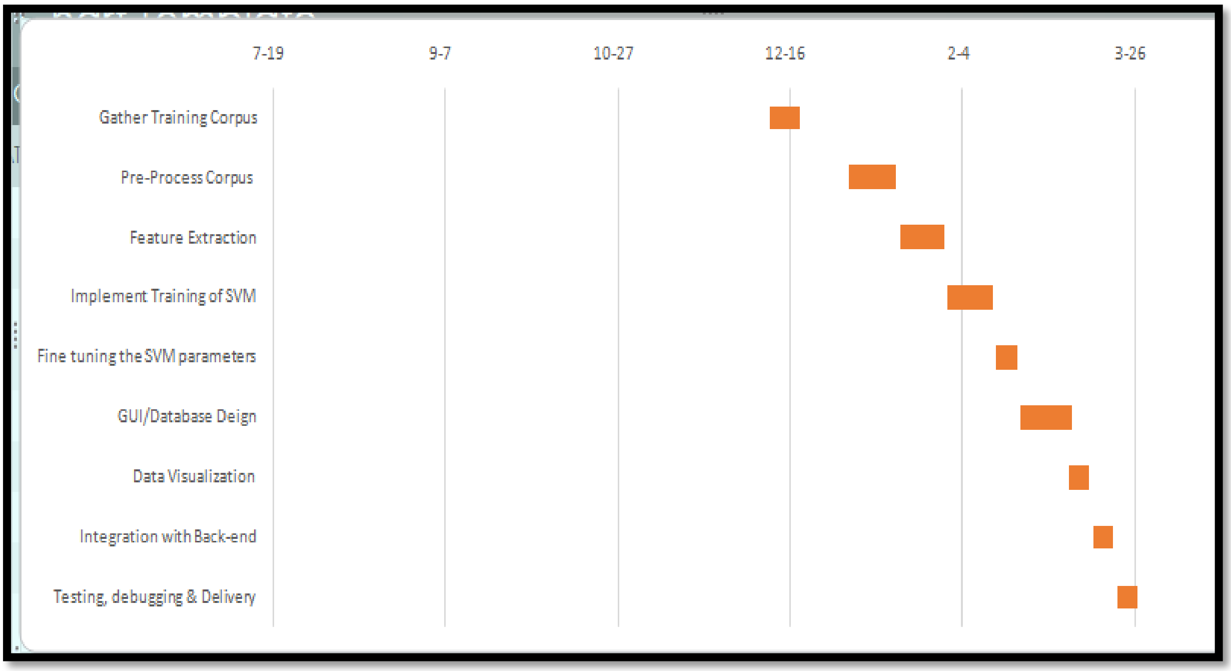
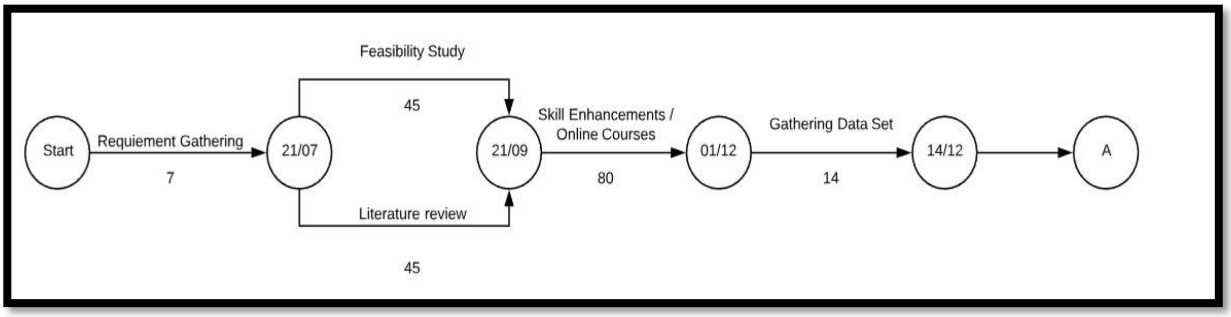
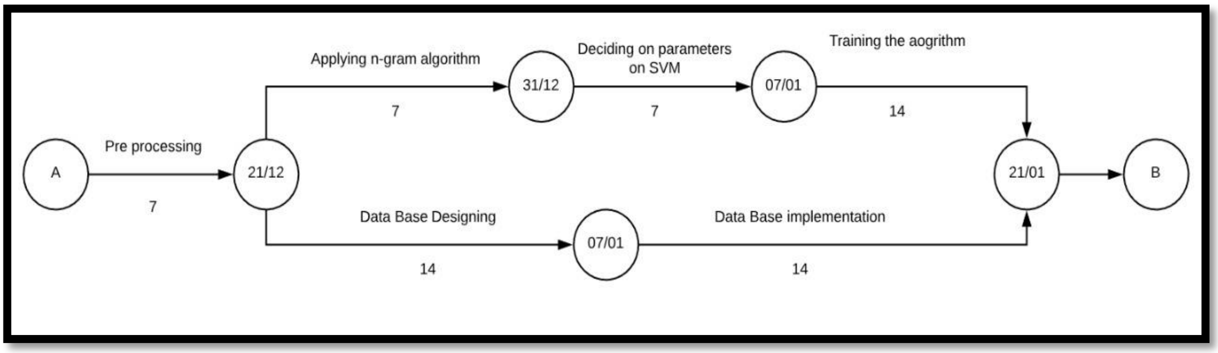
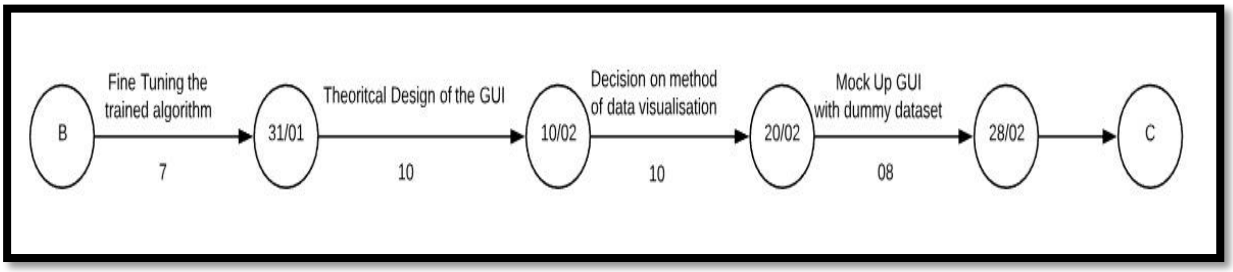
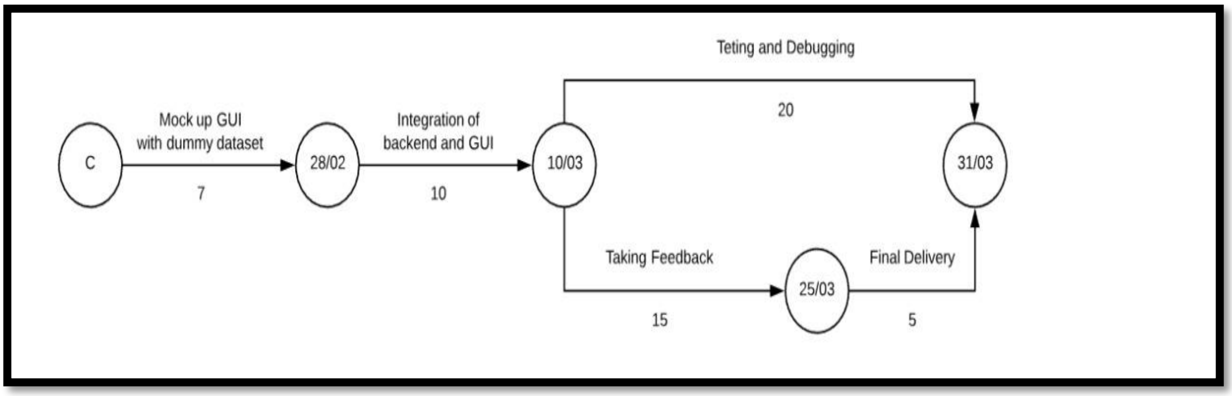

7-19
9-7
10-27
12-16
2-4
3-26
Gather Training Corpus
Pre-Process Corpus
Feature Extraction
Implement Training of SVM
Fine tuning the SVM parameters
GUI/Database Deign
Data Visualization
Integration with Back-end
Testing, debugging & Delivery

Fig. 10 PERT (Programme Evaluation Review Technique) diagram




Applying n-gram algorithm
7
31/12
Deciding on parameters on SVM
7
07/01
Training the aogrithm
14
A
Pre processing
7
21/12
21/01
B
Data Base Designing
14
07/01
Data Base implementation
14


B
Fine Tuning the trained algorithm
7
31/01
Theoritcal Design of the GUI
10
10/02
Decision on method of data visualisation
10
20/02
Mock Up GUI with dummy dataset
08
28/02
C


C
Mock up GUI with dummy dataset
7
28/02
Integration of backend and GUI
10
10/03
Teting and Debugging
20
31/03
Taking Feedback
15
25/03
Final Delivery
5

3. Project Description

3.1 Problem Statement

To determine the popularity of political parties and candidates and visualize the popularity trend as a function of time. The computation of the popularity will entail the classification of user-generated opinions in the form of tweets. Hence, ‘twitter’ has been utilized as the primary source of data for this project.

3.1.1 Justification

In recent years, the social media platform has proven to be a promising stage for stating an individual’s opinion. An opinion can be for a particular product, item, commodity, service, company or even political bodies. Hence, in order to take the advantage of the current Indian political scenario (Lok Sabha General Elections, 2019), this project will exploit the abundance of twitter data (in form of tweets that represent opinions) towards election contesting parties and candidates such as *Bhartiya Janta Party (BJP), Indian National Congress Party, Narendra Modi & Rahul Gandhi*. Now-a-days, the users of twitter post opinions regarding various issues such as employment, policies, jobs, healthcare, etc. Hence, after popularity has been deduced, the key issues from the set of classified tweets can be determined which can help a political party to develop its election strategy.

The users post direct opinions or sometimes sarcastic comments. Some typical examples include, “ *Congratulations to dear Modiji* !! *You are the best…*” , “Movies are being made on dear *Modiji* and people complaint of unemployment :)”. There is a significant difference between the former tweets. The first one is a simple comment and an opinion towards *Narendra Modi*. On the other hand, the second tweet is a sarcastic comment towards *Narendra Modi*. Even the presence of a smiley leads to a shift in the polarity of the message. It is therefore for this project to neglect the neutral comments, sarcastic comments from the twitter dataset (for corpus as well as the problem instances pool). For this purpose, human intervention is required as the invalid dataset can be then eliminated from the computation (learning stage and the classification stage).

3.2 Project Objective & Goals

Objective: To mathematically determine the polarity of user-generated opinions as tweets and subsequently compute the popularity of the respective political bodies (candidates /or parties). This project intends to provide a holistic view of public sentiment on prospective election candidates who will contest the *Lok Sabha General Elections,2019* by deducing the polarity of user-generated tweets. Hence, Twitter is being utilized as our primary source of data acquisition. The same also applies for mathematically computing the popularity of respective political parties over a defined time-frame among the Twitter users. The primary purpose of this project is that it is equipping us with a tool/product for analysing the tweets and gain knowledge in the form of data visualization. The project will demonstrate as to how the power of Natural Language Processing, Machine Learning and Data Visualization can be used together for the benefits of the Public Relations (PR) departments of various political parties. We demonstrate as to how social-media content can be used to predict real-world outcomes and how sentiments extracted from social media can be further utilized to improve the forecasting power of social media. Though uninteresting individually, knowledge from social-media platforms can provide an accurate reflection of public sentiment when taken in aggregation. As far as the use case is concerned, our product can be used by the Public Relations (PR) officers of the respective parties so that they can keep a track of their party's and/or their candidates' popularity and act accordingly by establishing better strategies to serve the citizens of the country.

Goals:

- ✓ To fetch desired problem instances which are related to the given context (Indian Politics)
- ✓ To store the retrieved data set for the purpose of computation
- ✓ To remove noise from the collected data set
- ✓ To gather training data for the purpose of defining a unique classifier (model)
- ✓ To classify the problem instances (test data) as either positive or negative
- ✓ To fine-tune the parameters of the classifier
- ✓ To plot the visualizations in the graphically understandable format
- ✓ To gain insights from every outcome of the machine learning model
- ✓ To visualize popularity as a function of time
- ✓ To compare popularity between two competing political bodies

3.3 Benefits while serving on the project

As stated in the proposal of the project, the approach adopted is Machine Learning. Hence, online courses were taken up in order to enhance the skill-set in terms of knowledge of latest technologies. Various different topics were explored such as classification, clustering, mathematical foundations for various algorithms, applications of algorithms and Natural Language Processing with Python. Certificate courses were taken up in order to bridge the knowledge gap. The existing skills were also enhanced via the online certificate courses in 'Python' and 'Data Visualization'.

Following benefits were gained during the phase of the development of the project-

- ✓ The concept of Sentiment Analysis
- ✓ Recent work in this field of research
- ✓ Available solutions for generic problem statements
- ✓ Specific solution for this particular problem statement
- ✓ Mathematical foundation behind the selected opinion classifier
- ✓ Knowledge certified as per certificate from online portal
- ✓ Enhancement in python programming skills
- ✓ Various data visualization techniques
- ✓ Core components in a ML based project
- ✓ An opportunity to write a technical paper based on results of this project
- ✓ Opportunities to improve in future

4. Project Implementation

4.1 Major Project Components

- Test Data Acquisition

This project started with the extraction of test problem instances from the twitter database. For this purpose, the team members had to register themselves with the "Twitter Developer Platform". The portal requires the team members to fill in important details such as name of the application, purpose of developing the application, data handling intentions and such similar questions. After answering all such questions, the request was submitted. Upon success, the credentials were returned. These credentials basically verify and validate the API object for every such application account. The credentials contained the following values – API Key, API Secret Key, Access Token & Access Token Secret. Random values are assigned to each of these keys. Since python programming language was utilised to develop the data extraction module, the "python-twiiter" module was imported and utilised. This project required the installation of the python-twitter module. Unfortunately this module works only with Python 2 currently and the Python 3 support is still under development. There are other modules that are similar though and some are listed on the Twitter API documentation website https://dev.twitter.com/overview/api/twitter-libraries. The following command-line utility was used to install the module- pip install python-twitter to install python-twitter for Python 2. This is a module that provides a python like interface to the Twitter API. The Twitter API is fairly straightforward to use like the REST APIs. A REST API provides information in the form of a JSON which application will have to parse once it gets it. python-twitter does this work and abstracts from having to know the nitty-gritty of the Twitter API. In case the module provides a json output; the json library could have been used to parse the tweets. This would be an additional step.

- Pre-processing of test data

As the previous module was developed, the required amount of test problem instances were fetched by the API object. The API object comprises of a ".getSearch()" which processes the request of tweets to the Twitter. A number of parameters are associated with this particular function. Some include- search-string, count, max-id, min-id. Here, the search-string is the important consideration for this project. As the keywords like "bjp, #congressParty, Modi" was passed as arguments, all those tweets that contain that keyword were returned. The "count" parameter can be set as per choice. However, due to recent changes in the twitter policies, every user (uniquely distinguished by the credentials) can fetch a maximum of 100 tweets per 15 minutes. This will return a list with twitter. Status objects. These have attributes for text, hashtags etc of the tweet that are being fetched. The full documentation again, can be seen by typing pydoc twitter.Status at the command prompt of terminal. As the required problem instances contained noise, pre-processing module was required to be developed. As stated before as well, the basic pre-processing activities include conversion into lower case, removal of hashtags, removal of links and tokenization. For this purpose, the NLTK and regular expression were used. Not only the pre-processing tasks were carried out, also the tweets which were not in English language were discarded. For this purpose, an independent function for the extraction of "English-only" tweets was also defined. So first the required keyword based tweets were fetched which were then fed into the language checker function.

As the keyword based tweets were fetched and stored, human intervention was required for the purpose of annotation. Here, annotation refers to reading each and every tweet from the file, understanding the context and then marking it as either positive or negative based on the current context. As the data belongs to Indian political context, the real time environment and scenario had to be taken into consideration for the purpose of annotation. This project fetched data based on recent issues as well. Hence, the tweets had topics such as "*kisan seva yojna, pulwama attack, biopic of Mr. Modi, etc*". Based on the present sentiment in that particular tweet string, the string was annotated as either positive or negative. Hence, every feature within that string will then contribute to the respective polarity. Once the annotated list of tweets was ready, the model had to be trained. For this purpose, the scientific library provided in the python programming language, "Sci-Kit Learn" was utilized. The various characteristics of the library justify the reason for its use :

- ✓ Open Source
- ✓ Commercially useable
- ✓ Built on existing libraries such as Matplotlib, SciPy, NumPy
- ✓ Simple and effective for the purpose of machine learning/data mining
- ✓ Accessible to everybody
- ✓ Reusable in various contexts

- Development into a Fully Trained Prototype

Once the model underwent training under the kernel provided by the machine learning, scientific library in python programming language, it was time to feed the classifier with the actual tweets as well as issues. Hence, the model was fed with various campaign related keyword such as "*Modi, Modiji, Pulwama, Pappu, Kisan, etc*". After having the classifier compute the incoming test data, the insights gained were significant. It was noted that the online community used the term "*Modi*" in a negative sentiment while the term "*Modiji*" was utilized in a positive sentiment. On the similar ground, it was observed that issue like the recent "Pulwama Attack" attracted a lot of negativity by the twitter community due to which that issue became a negative word for a successful election campaign. As and when the training data will be updated with the current political environment, the corresponding insights will be gained accordingly. Hence, this project exhibits high traits of software engineering by enabling flexibility and reusability of the existing source code. Not only the issues and campaign related keywords were tested, but also the various test tweets that were fetched as part of the test data were also fed into the classifier. Depending on the presence or absence of strong features, the corresponding polarity was detected. This particular feature or functionality of this project is high useful in crafting a campaign related slogan for a particular political party. Hence, the title of this project/tool/utility justifies its name, i.e. "Campaign Assistant". This project will assist the political party (in this case, BJP) is creating attractive slogans for the purpose of influencing the prospective voters. For example, a slogan that contains keywords such as "*Modiji, Kisan, Yojana, Award*" will have a positive impact on the voters which in turn will lead to higher votes in the upcoming Lok Sabha General Elections, 2019.

- Developing a GUI supported Final Deliverable

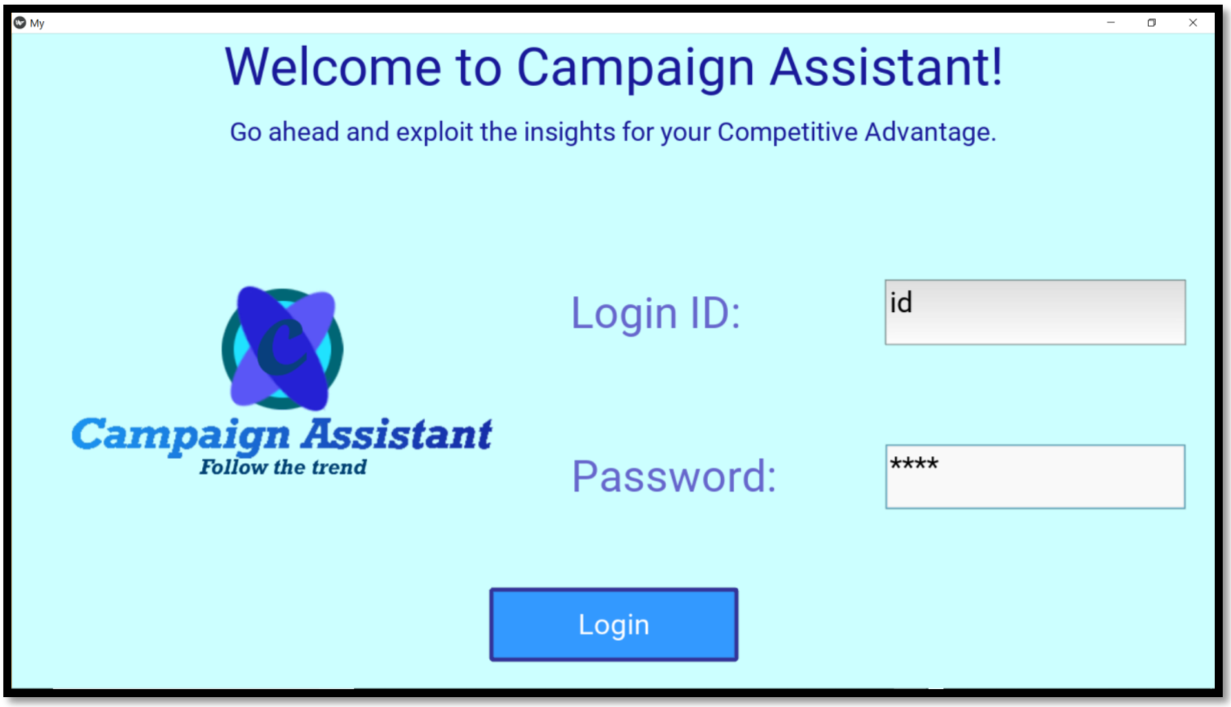
This project timely proceeded as per the stated schedule in the Gantt chart. The back-end mathematical code was developed within the expected deadline. However, this project required an appropriate Graphical User Interface (GUI) in order to be truly utilized as a tool by the end-user. For this goal to be achieved, a corresponding framework had to be chosen. Available options included tkinter, PyQt & Kivy. These frameworks are compatible with the development environment of Python. The front-end development team chose to go ahead with the development of the front-end/GUI by utilizing the Kivy framework.

Reasons that justify the incorporation of Kivy framework –

- ✓ Open Source
- ✓ Cross Platform
- ✓ Enables development of visually appealing graphical user interfaces (GUIs)
- ✓ A wide variety of widgets available
- ✓ Free to user
- ✓ A stable framework
- ✓ A well-documented Application Programming Interface (API)
- ✓ A programming guide to help get started

The first initial time was spent in setting-up the environment for the development work. For this purpose, the “Anaconda Virtual Environment” was utilized. The graphical user interface (GUI) should enable the easy usage of the tool by the political party, hence the golden rules of UX-design were considered. Many tabs were developed including the Sentiment Test tab, Summary Tab, Report Tab and a Help/FAQ tab. In the Sentiment Test tab, the end-user can test the polarity of the keywords which could be used in an election campaign. The output will be either positive or negative. In the Summary tab, the end-user can gain qualitative insights from the test data collected so far. A typical example is that of a Word Cloud. In the Report tab, the end-user can monitor the performance of the popularity of the political party via various dashboard-based graph objects such as pie-chart, bar graph and line chart. The insights gained here will be quantitative in nature. The Help tab will provide useful information about the tool and simplified meaning of the jargons.

5. Screenshots



Welcome to Campaign Assistant!

Go ahead and exploit the insights for your Competitive Advantage.

Login ID: id

Password: ****

Fig. 11 Login Module

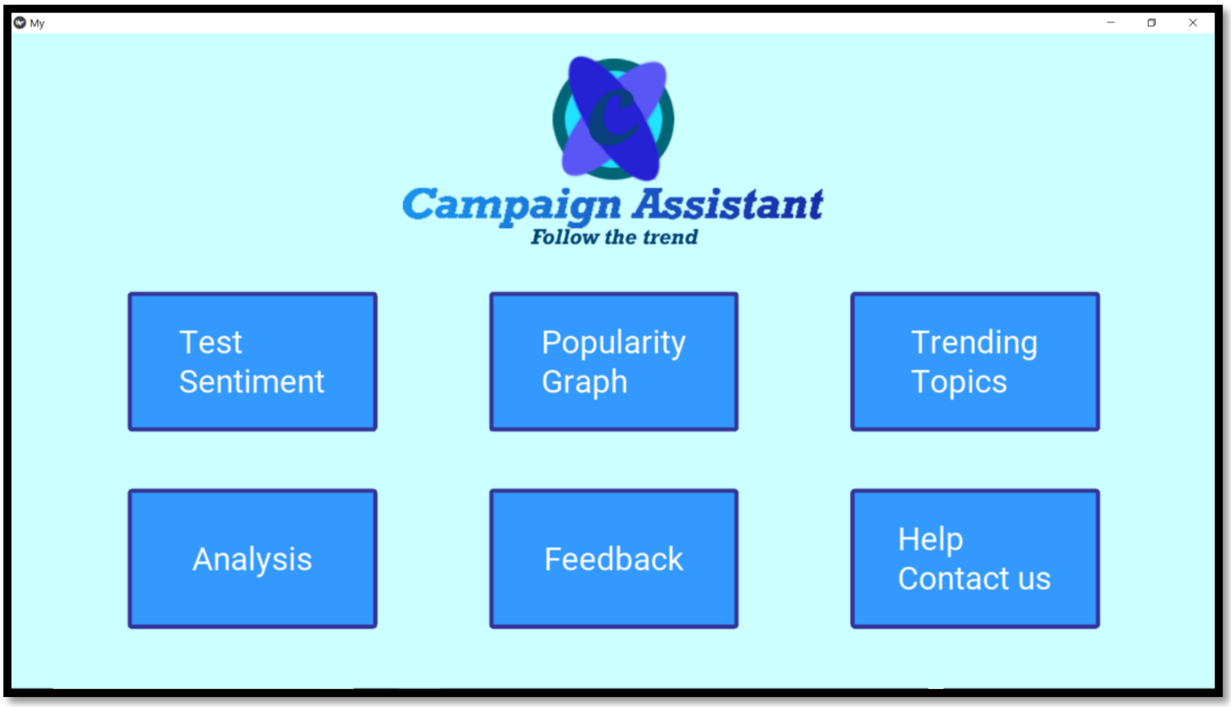
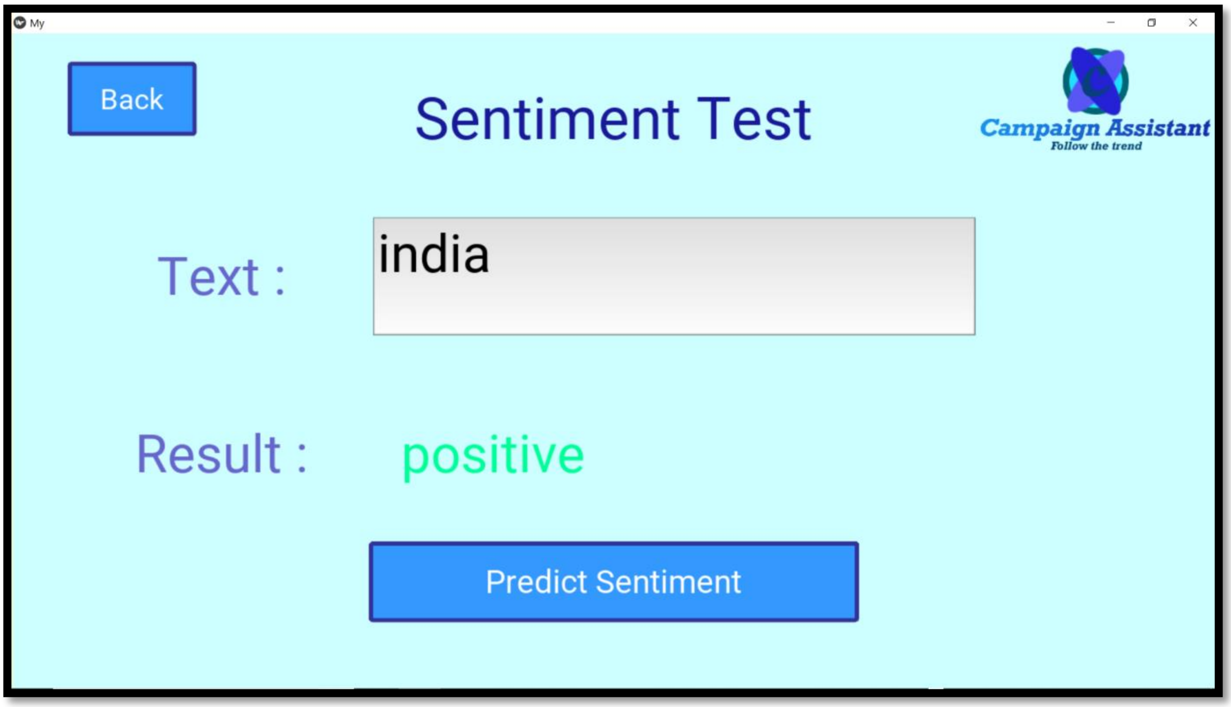


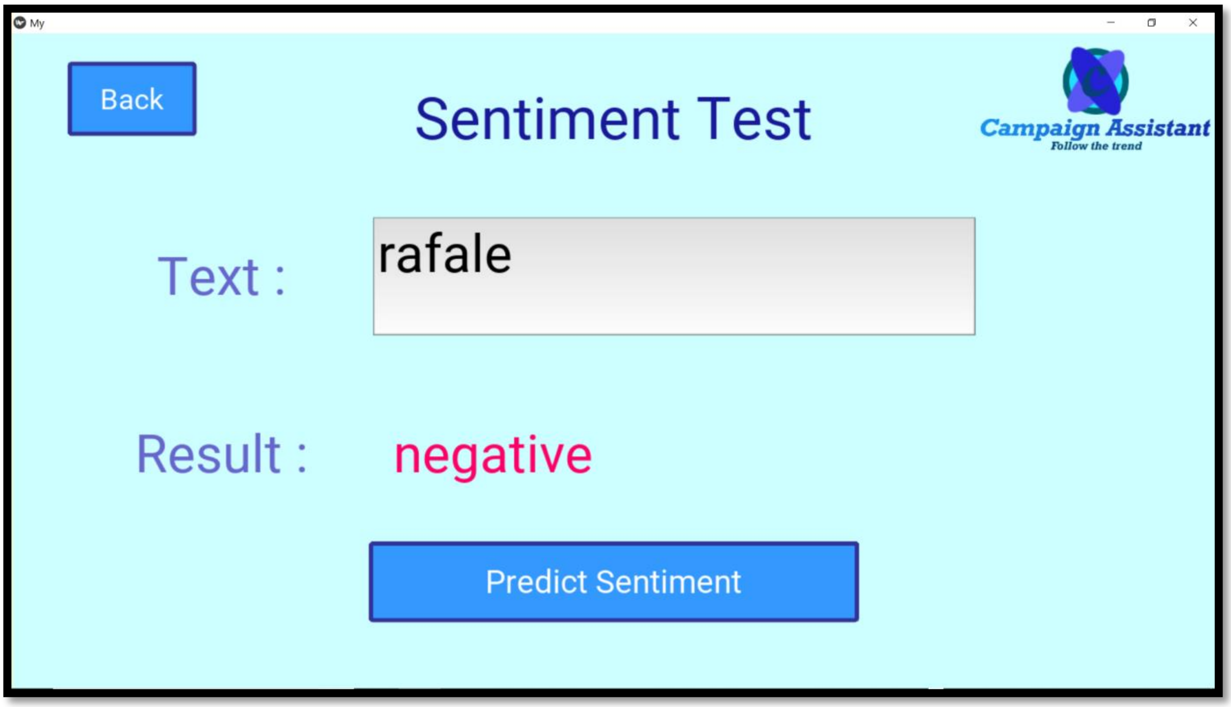
Fig. 12 Homepage of the Tool



Text : india

Result : positive

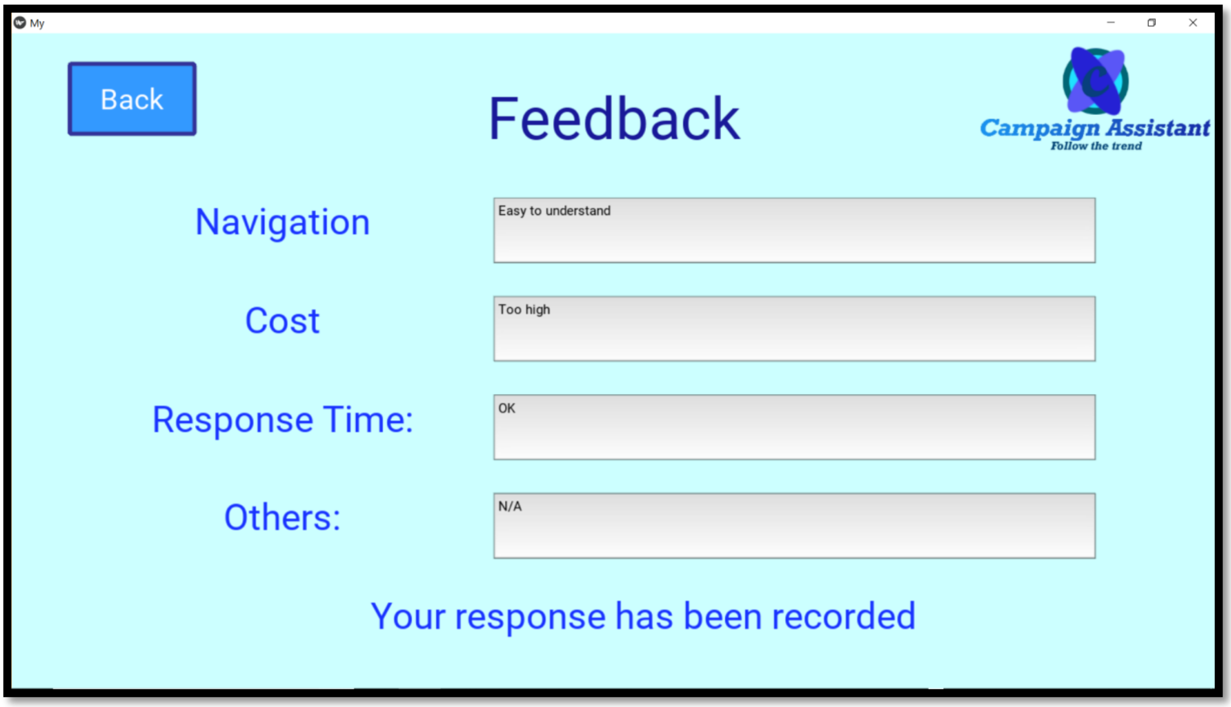
Fig. 13 Module to check the polarity of a keyword



Text : rafale

Result : negative

Fig. 14 Module to check polarity of a keyword



Feedback

Navigation: Easy to understand

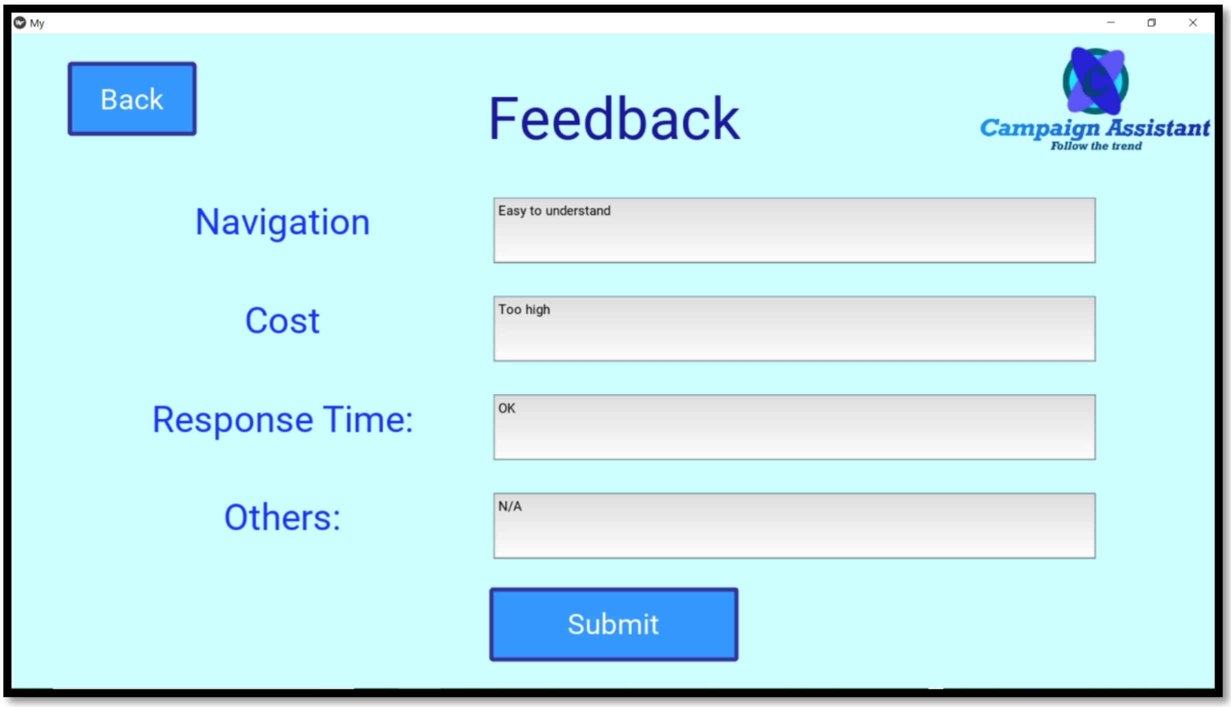
Cost: Too high

Response Time: OK

Others: N/A

Your response has been recorded

Fig. 15 Feedback Module



Feedback

Navigation	Easy to understand
Cost	Too high
Response Time:	OK
Others:	N/A

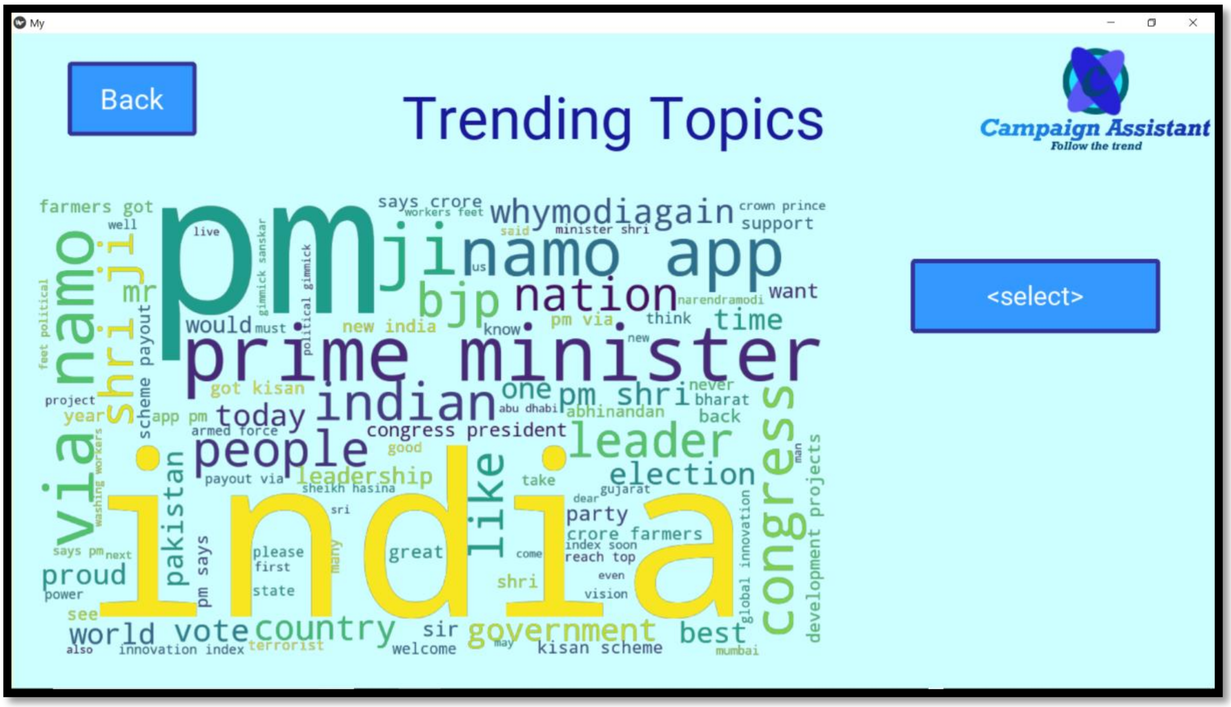


Fig. 16 Data Visualization Module – Word Cloud

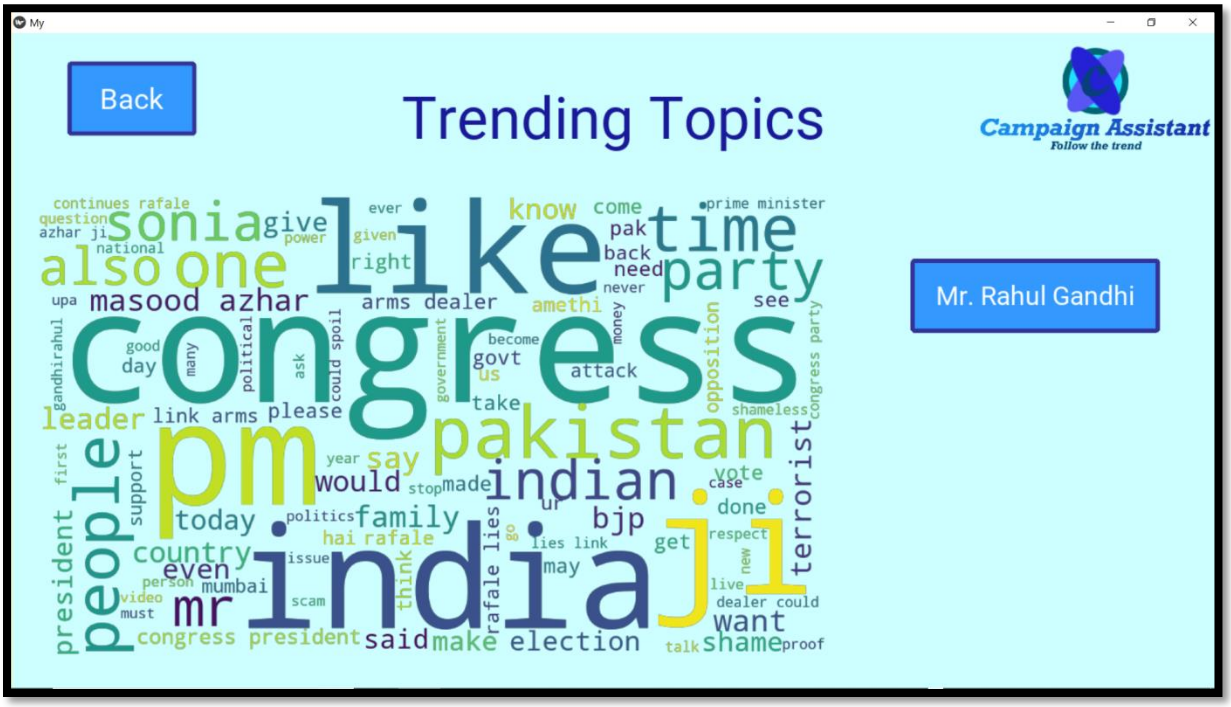


Fig. 17 Data Visualization Module – Word Cloud

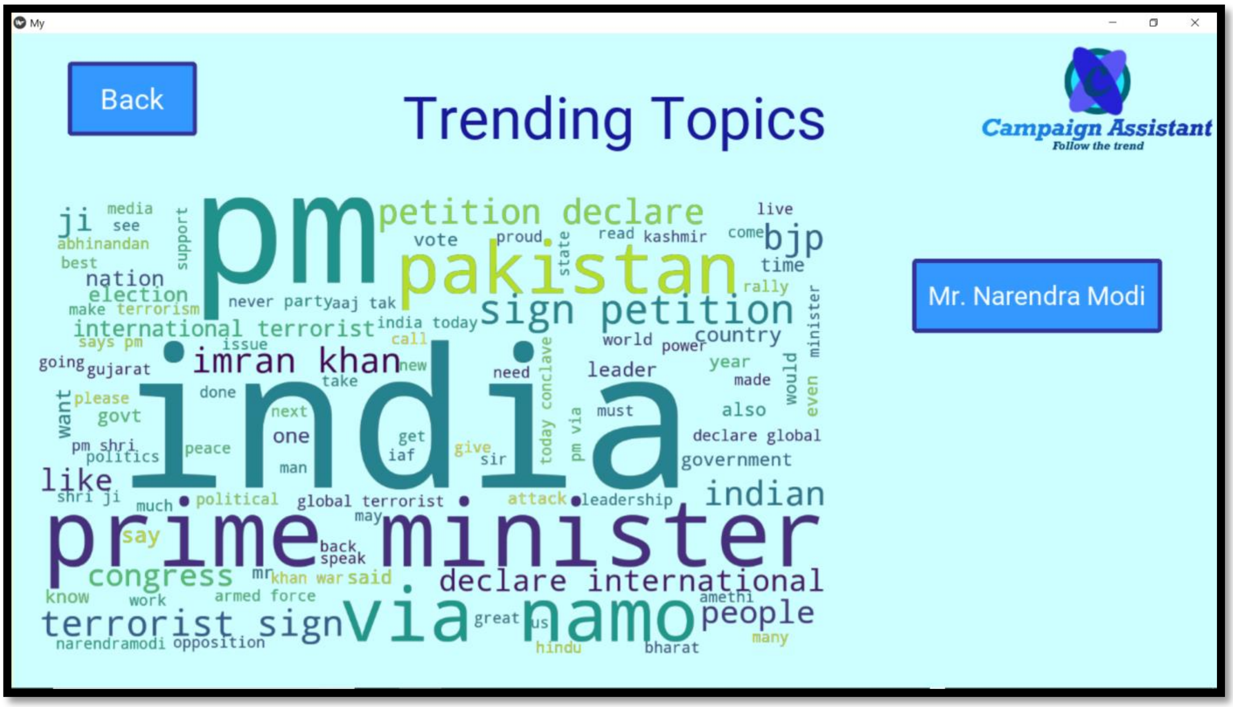
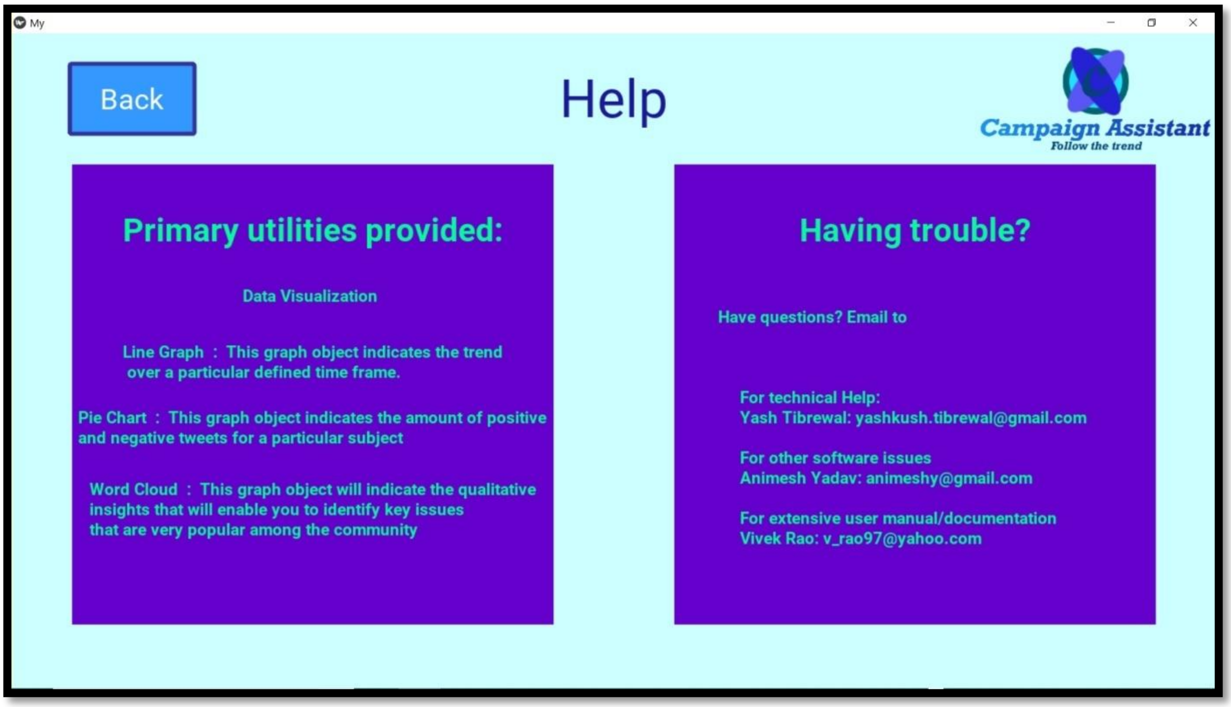


Fig. 18 Data Visualization Module – Word Cloud



Help

Primary utilities provided:

Data Visualization

Line Graph : This graph object indicates the trend over a particular defined time frame.

Pie Chart : This graph object indicates the amount of positive and negative tweets for a particular subject

Word Cloud : This graph object will indicate the qualitative insights that will enable you to identify key issues that are very popular among the community

Having trouble?

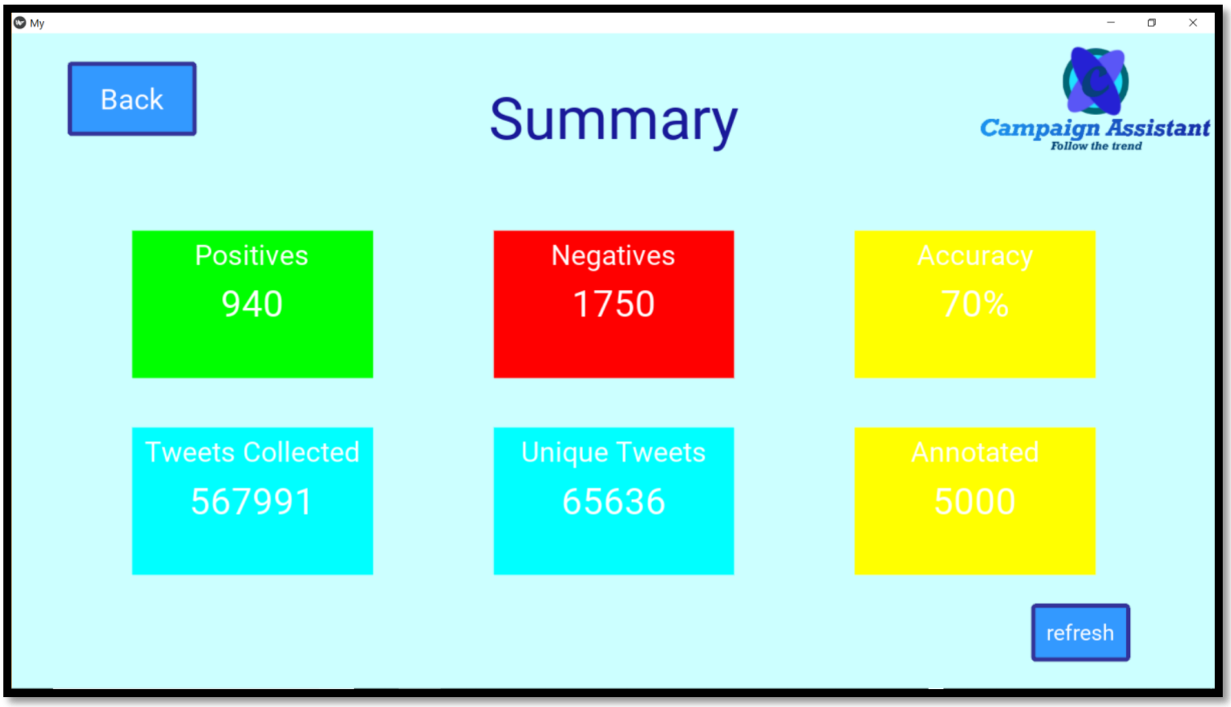
Have questions? Email to

For technical Help:
Yash Tibrewal: yashkush.tibrewal@gmail.com

For other software issues
Animesh Yadav: animeshy@gmail.com

For extensive user manual/documentation
Vivek Rao: v_rao97@yahoo.com

Fig. 19 Help Module



Summary

Positives 940

Negatives 1750

Accuracy 70%

Tweets Collected 567991

Unique Tweets 65636

Annotated 5000

Fig. 20 Analysis-Summary Module

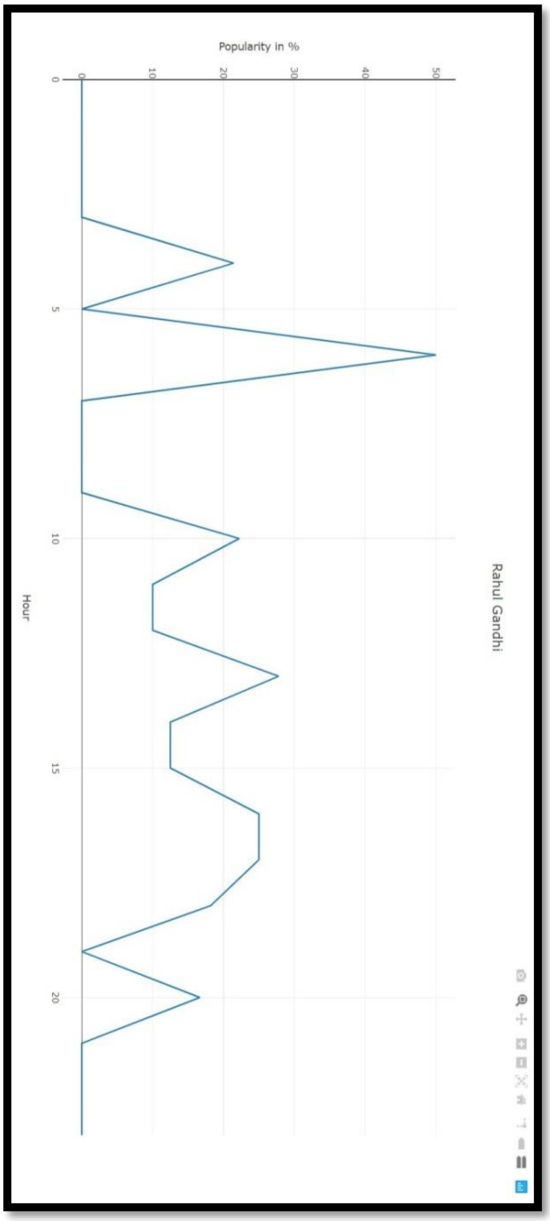


Fig. 21 Line Graph Showing Trend

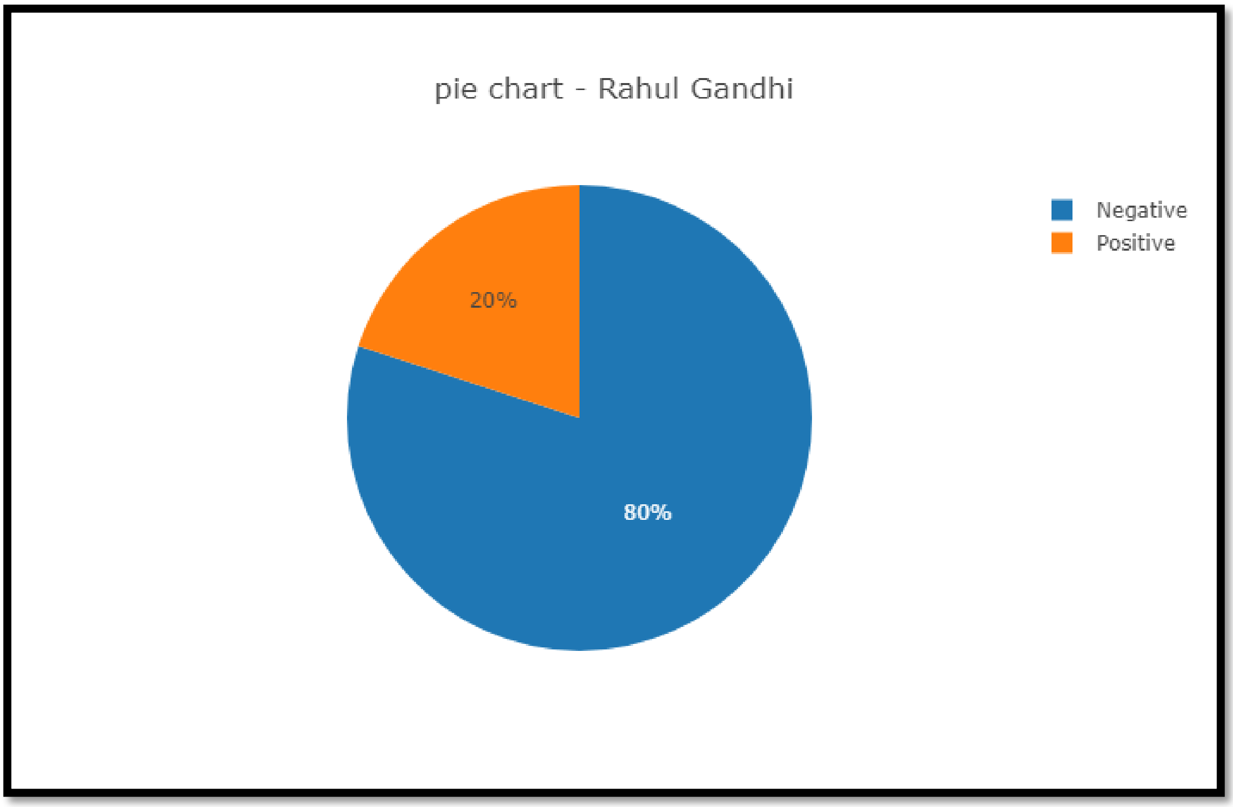
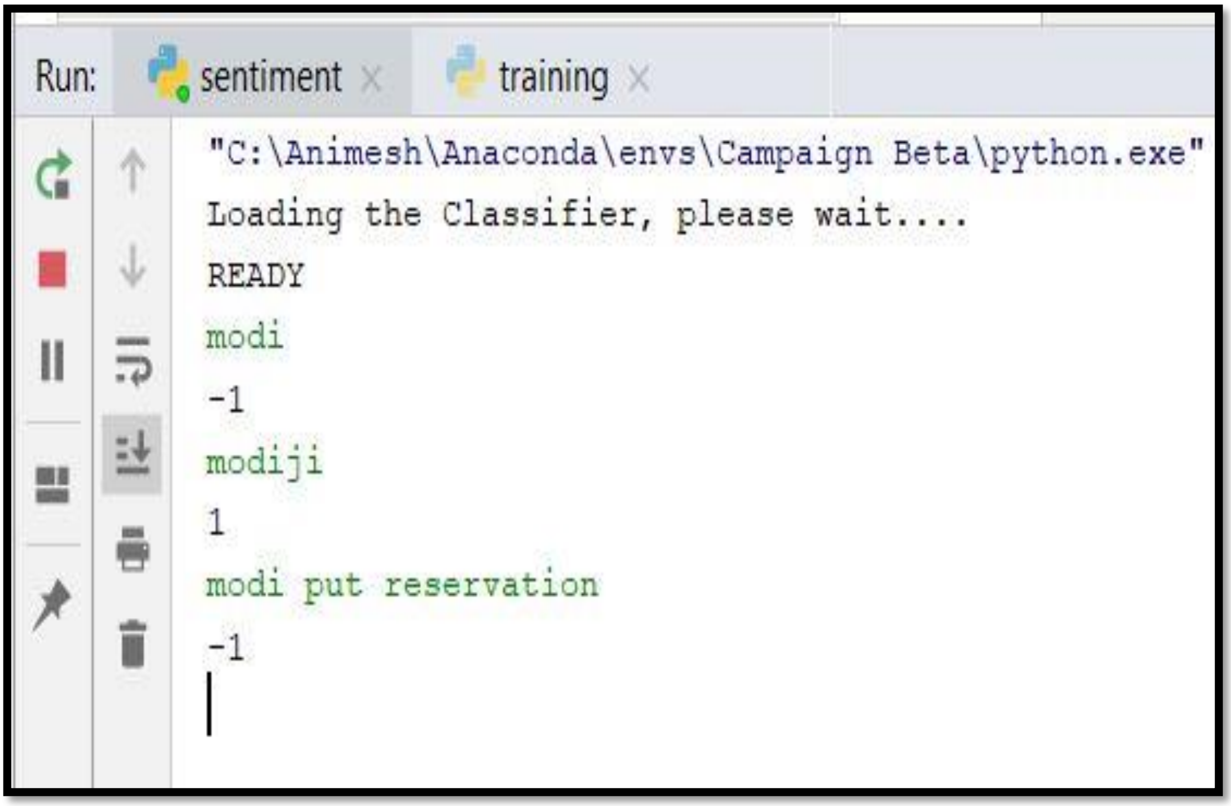


Fig. 22 Pie Chart



```
"C:\Animesh\Anaconda\envs\Campaign Beta\python.exe"
Loading the Classifier, please wait....
READY
modi
-1
modiji
1
modi put reservation
-1
```

Fig. 23 Initial Prototype

6. Project Test Report

6.1 White Box Testing

When "White Box testing" was incorporated, the development team itself engaged in the entire testing process. Every module that was developed from scratch underwent thorough testing as follows –

White-Box Test Sheet

Functional Module	Module Description	Test Case No.	Test Case
Data fetching	Fetches Tweets as per the keyword argument	Test Case #1	Check if relevant tweets are fetched
Data storage	Inserts fetched tweets into excel sheet as rows	Test Case #2	Check if each row contains tweet, timestamps
Pre-Processing	Removes noise from the fetched tweets	Test Case #3	Validate if noisy data is cleaned if passed as input to the module
		Test Case #3.1	Check if emoticons are converted into characters
		Test Case #3.2	Check if punctuation marks are removes
		Test Case #3.3	Check if string is converted into lowercase
		Test Case #3.4	Check if input string is tokenized as expected
		Test Case #3.5	Check if hashtags are removed
Training Model	Trains the mathematical model for the purpose of learning	Test Case #4	Check if corpus contains neutral-free binary annotated tweets related to political subjects

Opinion classifier	Classifies the test data as either positive or negative	Test Case #5	Validate if the passed keyword associates with the expected sentiment
		Test Case #5.1	Check if tweets/keywords relevant to Modi/Rahul in the respective context associate with expected polarity
Graphical User Interface	Provides visually appealing front-end to use the tool	Test Case #6	Validate if all the utilities provided by the GUI function as per its definition
		Test Case #6.1	Check if the tabs enable switching from one window to another by single click
		Test Case #6.2	Check if the on-click events execute on user-click
		Test Case #6.3	Check if input field of sentiment test passes the argument to the classifier
		Test Case #6.4	Check if output field of sentiment test produces a result in the form of sentiment polarity
Data Visualization	Plots visually appealing graphs, charts in order to help the end-user gain meaningful insights	Test Case #7	Validate if plots are generated upon end-user click

Table-II

Results & Discussions

The machine learning approach stated in the previous sections was adopted to compute and predict the popularity of political candidates of India such as *Narendra Modi* and *Rahul Gandhi*. After the training corpus was collected and the training model was successfully trained, key issues and keywords were identified which will have influence on the popularity. The Twitter data was fetched and exploited to visualize the popularity of *Lok Sabha* candidates.

In order to gain insights from the test data, visually appealing graph objects were incorporated. For the purpose of qualitative insights, a word cloud was developed. The visual output in the form of screenshots indicates the outcome of the implementation. It can be clearly observed that the popularity generated by means of a mathematical equation of *Rahul Gandhi* varies significantly within a single day as compared to over months. However, the overall popularity of *Rahul Gandhi* is lower than that of *Narendra Modi*. Hence, the popularity of *Rahul Gandhi* is on a decline as per the insights gained from the tweets collected so far. On the other hand, the online community expresses its sentiment for Modi in varying form as he is currently in power which lead to various issues or events over the timeline. The quality of comprehensive tweets towards Narendra Modi is higher as compared to that of Rahul Gandhi. One of the major insights gained during the successful run of this project is that the Twitter community of India expresses positivity towards Narendra Modi by using the word, "*Modiji*". Therefore, the Modi government can exploit this insight in its election campaigns in order to influence and motivate the citizens of India to vote for them. The pie-chart shows the amount of positive & negative tweets for a particular leader.

8. Conclusions

8.1 Conclusion

This project incorporated machine learning approach for the purpose of solving a very specific problem. Sentiment Analysis of social media is an upcoming trend and now employed by almost every firm to know about their customers/clients and hence act accordingly to remain competitive in the market. Similarly, the political party's social media team can use this novel tool to test the polarity of keyword which could be used in their election campaigns and determine to whether go ahead with it or not. The method considers only positive and negative tweets and discards neutral ones. Human intervention was required to eliminate the neutral tweets in order to increase the accuracy. As long as the context of the corpus remains stable, the average accuracy obtained fluctuates between 65% to 70% where the maximum theoretical accuracy possible is 80%. However, accuracy cannot be defined and relied upon as the outcome is in terms of local sentiment and it is highly dependent on the environment variables such as location, culture and language.

Upon the actual deployment of the project on real world data, the two political subjects considered were "*Rahul Gandhi*" & "*Narendra Modi*". These two leaders were chosen as they are the prospective candidates for *Lok Sabha General Elections, 2019*. The classifier was trained on almost 2000 tweets that were unique and annotated. It was observed that the popularity of *Rahul Gandhi* was significantly lower as compared to that of *Narendra Modi's.* As far as numbers are concerned, *Rahul Gandhi's* highest popularity went upto only 50% within a single day. Over a few months, it was as low as 10%. The resulting quantitative results were demonstrated by the utilization of visual charts and graphs. The numbers indicated by the line graphs convey the trend of their popularity. In order to convey the qualitative insights which will actually enable the political parties for decision making, a Word Cloud was utilized which helped to visualize all the issues the online community is upset or not happy with. By considering significant actions on these issues, the respective political party can work on effective solutions and regain their higher popularity. Hence, as long as the training data is updated with the latest issues, local sentiments and needs, the sentiment of the online community towards their followed organizations can be easily visualized.

8.2 Future Avenues

The scope of additional work on this project is almost endless. First and foremost, the accuracy remains the primary objective. For this purpose, the corpus needs to be extremely clean, comprehensive and limited to the desired subject. As and when the current environment changes, the corpus needs to be updated so that the hidden sentiments are truly captured. This project incorporated a corpus of just a few thousand tweets while the online database is gigantic. Hence, the corpus needs to accommodate as many contexts, local sentiments, issues, keywords and style as much as possible with respect to the desired subjects.

Secondly, the software tool can be made to run in real-time where the visual graphs will update themselves as and when the new and unique tweets relevant to the desired subjects are fetched. For this purpose, the tool needs to be deployed on a server where all modules run in real-time. For every 15 minutes, (as enforced by the policies of Twitter) the server would fetch 100 tweets. The redundancy check module could eliminate the redundant tweets and would pass on the unique ones to the pre-processing module. As soon as the test data is ready for computation, the classifier would be fed with it and the outcome would be visualized using a variety of graphical objects.

This tool can be made flexible enough to accommodate a wide variety of languages as well. In India, every state is different in terms of local language as well. Most common is the Hindi language. Hence, the tweets posted in Hindi language would not undergo pre-processing by undergoing removal by the function when fetched For this purpose, either a context based translation service could be provided or the corpus could be populated with the characters of Hindi language.

Detecting sarcasm and handling the same remain one of the biggest challenges of this project. While annotating tweets for the purpose of populating the corpus, a lot of user-generated tweets were observed to be sarcastic. Sarcasm basically means to convey an opposite meaning while the words in that sentence mean just the opposite. The online community, especially on Twitter, uses a lot of sarcastic comments. Hence, manipulating the overall popularity effectively by detecting sarcasm is in itself a major project which involves a lot of literature review on various available algorithmic solutions.

REFERENCES

[1] Shayaa, S., Jaafar, N., Bahri, S., Sulaiman, A., Seuk Wai, P., Wai Chung, Y., Piprani, A. and Al-Garadi, "Sentiment Analysis of Big Data: Methods, Applications, and Open Challenges". *IEEE Access*, 6, pp.37807-37827.

[2] E. Cambria, B. Schuller, Y. Xia and C. Havasi, "New Avenues in Opinion Mining and Sentiment Analysis", *IEEE Intelligent Systems*, vol.28, pp.15-21, Feb,2013.

[3] V. Kagan, A. Stevens and V.S. Subrahmanian, "Using Twitter Sentiment to Forecast the 2013 Pakistani Election and the 2014 Indian Election", *IEEE Intelligent Systems*, vol.30, pp.2-5, Feb 2015.

[4] L. Banic and M. Brakus, "Using Big Data and Sentiment Analysis in Product Evaluation", *in 36th International Convention on Information Communication Technology, Electronics and Microelectronics (MIPRO)*, 2013, pp.1149-1154.

[5] E. Cambria, "Affective Computing and Sentiment Analysis", *IEEE Intelligent Systems*, vol.31, pp.102-107, Mar 2016.

[6] A. Krouska, C. Troussas and M. Vivou. "The Effect of Pre-processing Techniques on Twitter Sentiment Analysis", *in 7th International Conference on Information, Intelligence, Systems & Applications (IISA)*, 2016, pp.1-5.

[7] Z. Jianqiang and G. Xiaolin. "Research on Text Pre-processing Methods", *IEEE Access*, vol.5, pp.2870-2879, Feb 2017.

[8] L. Wang and J. Q. Gan, "Prediction of the 2017 French Election Based on Twitter Data Analysis", *in 9th Computer Science and Electronic Engineering (CEEC),* Colchester, UK, 2017.