****

**COMSATS University Islamabad**

**Abbottabad, Pakistan**

**Sentiment Analysis on Android/IOS Applications**

***By***

**Abdul Hameed Khan CIIT/FA18-BCS-002/ATD**

**Daud Ahmed CIIT/FA18-BCS-010/ATD**

***Supervisor*Ma’am Ayesha Jadoon**

***Bachelor of Science in Computer Science (2018-2022)***

****

**COMSATS University, Islamabad Pakistan**

**Sentiment Analysis on Android/IOS Applications**

**A project presented to**

**COMSATS Institute of Information Technology, Islamabad**

**In partial fulfillment**

**of the requirement for the degree of**

***Bachelor of Science in Computer Science (2018-2022)***

**By**

**Abdul Hameed Khan CIIT/FA18-BCS-002/ATD**

**Daud Ahmed CIIT/FA18-BCS-010/ATD**

**CERTIFICATE OF APPROVAL**

It is to certify that the final year project of BS (CS) “Sentiment Analysis on Android/IOS Applications was developed by “**Abdul Hameed Khan (CIIT/FA18-BCS-002/ATD) and Daud Ahmed(CIIT/FA18-BCS-010/ATD)** under the supervision of “**Ma’am Ayesha Jadoon**” her opinion; it is fully adequate, in scope and quality for the degree of Bachelors of Science in Computer Sciences.

---------------------------------------

**Supervisor**

---------------------------------------

**External Examiner**

---------------------------------------

**Head of Department**

**(Department of Computer Science)**

**ACKNOWLEDGEMENT**

All praise is to Almighty Allah who bestowed upon us a minute portion of His boundless knowledge by virtue of which we were able to accomplish this challenging task.

We are greatly indebted to our project supervisor “Ma’am Ayesha Jadoon”. Without their personal supervision, advice and valuable guidance, completion of this project would have been doubtful. We are deeply indebted to them for their encouragement and continual help during this work.

And we are also thankful to our parents and family who have been a constant source of encouragement for us and brought us the values of honesty & hard work.

Student Name 1 Student Name 2

--------------------------- ---------------------------

## Table Of Contents

#### Study Project Page No.

Chapter 1 Introduction 9

Chapter 2 UML Diagrams 12

* 1. Sequence Diagram
  2. State Diagram
  3. Data Flow Diagram
  4. Flow Chart

Chapter 3 About the System 14

* 1. Problem Definition
  2. Technology Used
  3. Working Procedure
  4. Models Developed

Chapter 4 Study and Survey of Related Systems 19

Chapter 5 Conclusion and Future Scope 21

#### Chapter 1

**Introduction**

A sentiment is a view or opinion that is held or expressed. Sentiment means a view or opinion, but it can also mean an emotion. Maybe you prefer tragic movies because you enjoy the sentiment of sadness. This meaning of sentiment is taken to an extreme in yet another version of the word, meaning something like "overdone, exaggerated feelings, especially of sadness or nostalgia." For example, let’s take this sentence: ”I don’t find the app useful; it’s really slow and constantly crashing.” A sentiment analysis model automatically tags this as *Negative.*

**Sentiment analysis** (also known as **opinion mining** or **emotion AI**) refers to the use of natural language processing, text analysis, computational linguistics, and biometrics to systematically identify, extract, quantify, and study affective states and subjective information. Sentiment analysis is widely applied to voice of the customer materials such as reviews and survey responses, online and social media, and healthcare materials for applications that range from marketing to customer service to clinical medicine.

Sentiment analysis is contextual mining of text which identifies and extracts subjective information in source material, and helping a business to understand the social sentiment of their brand, product or service while monitoring online conversations. However, analysis of social media streams is usually restricted to just basic sentiment analysis and count

based metrics. This is akin to just scratching the surface and missing out on those high value insights that are waiting to be discovered.

Let’s say a company has just launched a new product feature and you notice a sharp increase in mentions on Twitter. However, receiving tons of mentions does not *necessarily* mean a good thing. Are customers reviewing more because they are expressing good things about this new product feature? Or, are customers actually complaining about the feature having lots of bugs? Performing twitter sentiment analysis can be excellent way to understand the tone of those mentions and obtain real-time insights on how users are perceiving your new product.

Because of sentiment analysis, companies can understand the reputation of their brand. By analyzing social media posts, customer feedback, or NPS(Net promoter Score) responses(among other sources of unstructured business data), they can be aware of how their customers *feel* about their product. They can also track specific topics and get relevant insights on how people are talking about those topics. Sentiment analysis is particularly useful for social media monitoring because it goes beyond metrics that focus on the number of likes or re-reviews , and provides a qualitative point of view.

With the recent advances in deep learning, the ability of algorithms to analyze text has improved considerably. Creative use of advanced artificial intelligence techniques can be an effective tool for doing in-depth research. Sentiment Analysis is the most common text classification tool that analyses an incoming message and tells whether the underlying sentiment is positive, negative our neutral. Performing sentiment analysis

on data from Twitter using machine learning can help companies understand how people are talking about their brand.

Twitter allows businesses to reach a broad audience and connect with customers without intermediaries. On the downside, it’s harder for brands to quickly detect negative content, and if it goes viral you might end up with an unexpected PR crisis on your hands. This is one of the reasons why social listening – monitoring conversation and feedback in social media – has become a crucial process in social media marketing. Monitoring Twitter allows companies to understand their audience, keep on top of what’s being said about their brand and their competitors, and discover new trends in the industry.

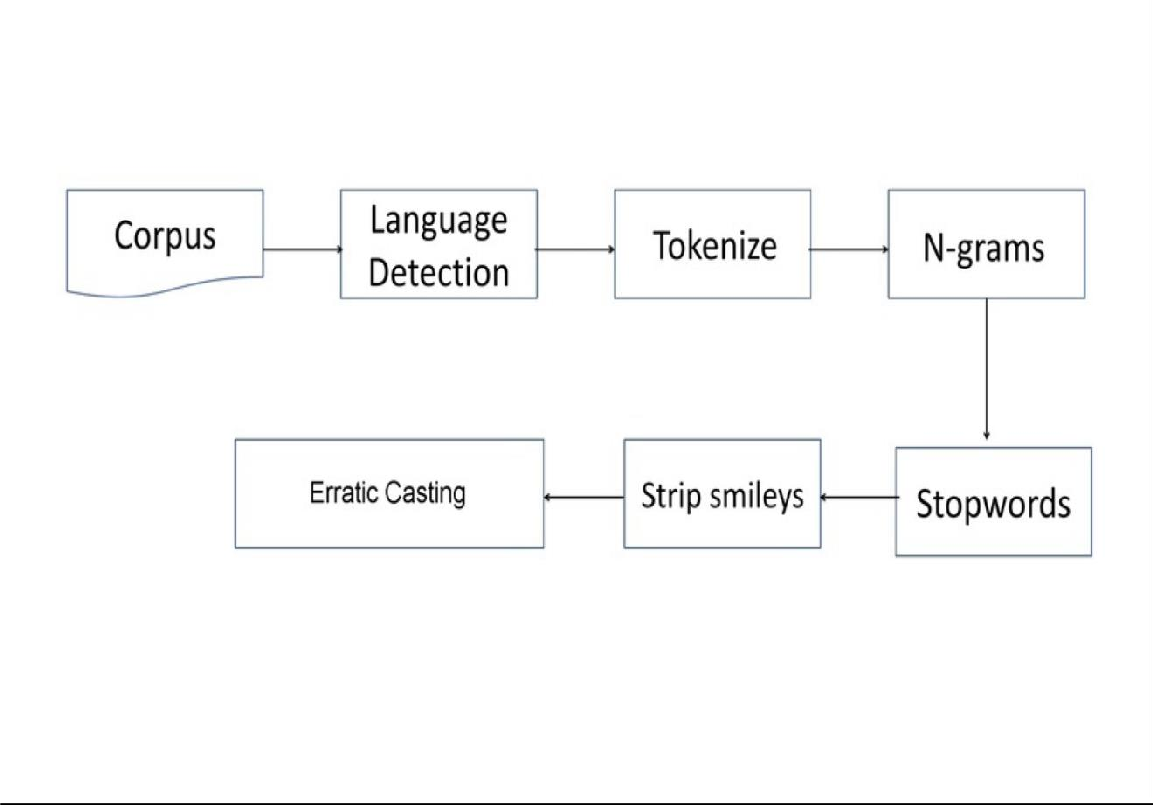
Nearly 80% of the world’s digital data is unstructured , and data obtained from social media sources is no exception to that. Since the information is not organized in any predefined way, it’s difficult to sort and analyze. Twitter sentiment analysis systems allow you to sort large sets of reviews and detect the polarity of each statement automatically.

If we want to perform Twitter sentiment analysis, the first step is to gather the data. This data is the data that we will use for training the model and running the actual sentiment analysis on Twitter data. There are two main type of reviews we can extract from Twitter: Current Reviews and Historical Reviews. Current Review is useful to track keywords or hashtags in real-time. Historical Reviews are used to search past reviews during a predefined time frame.

Note that no any algorithm or technique provides 100% accuracy or prediction on Sentiment analysis.

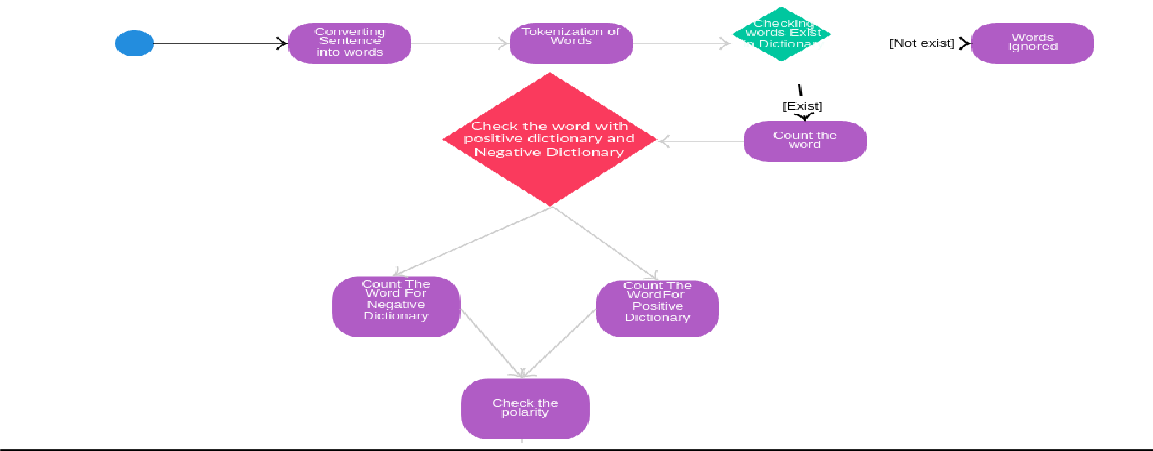
#### Chapter 2 UML Diagrams

##### Sequence Diagram:



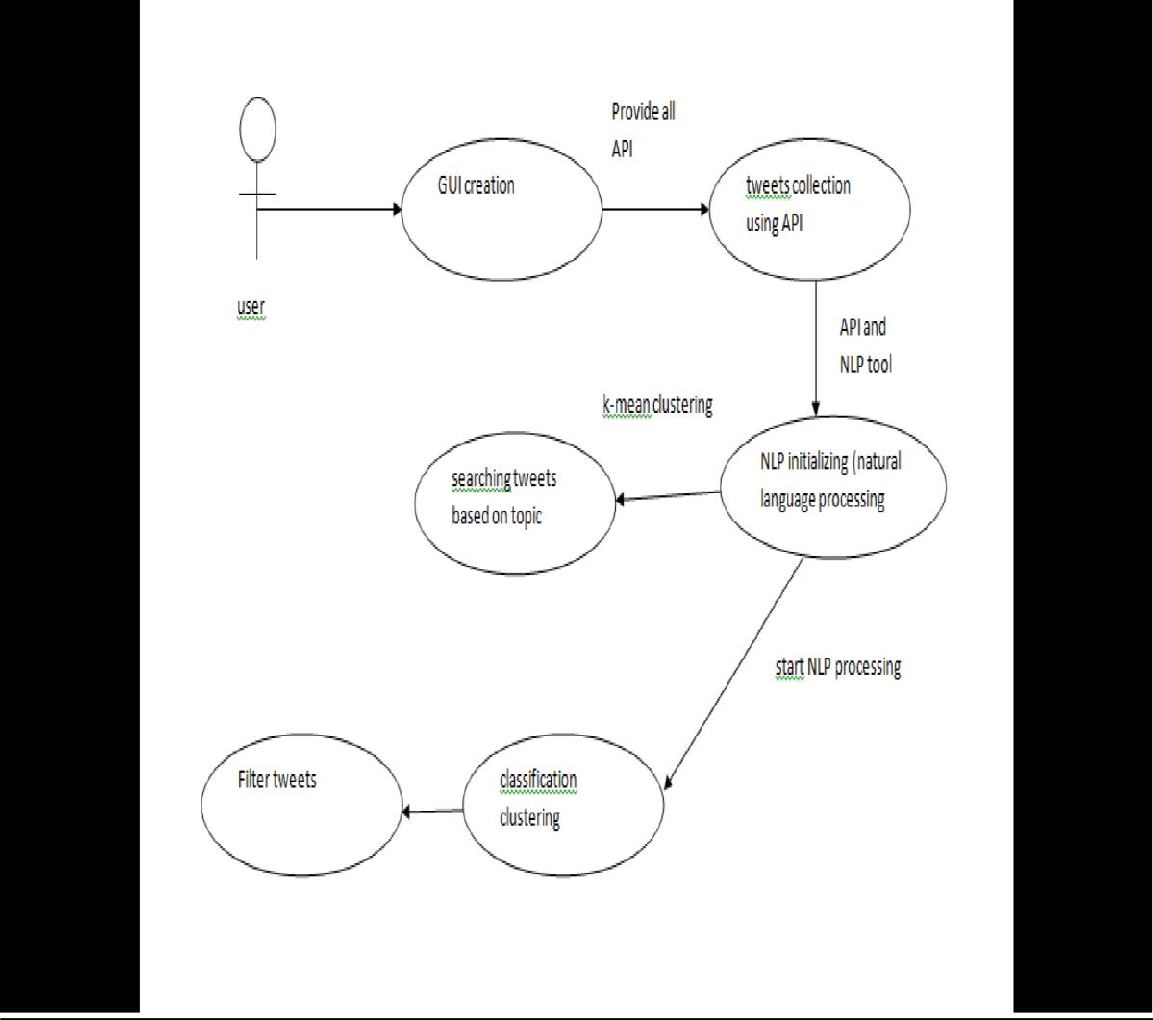
[Figure: 1] Sequence Diagram

##### State Diagram:



[Figure: 2] State Diagram

##### DFD(Data Flow Diagram):



[Figure: 3] Data Flow Diagram

#### Chapter 3 About the System

**Problem Definition:**

Sentiment analysis of Twitter in the domain of micro-blogging is a relatively new research topic so there is still a lot of room for further research in this area. Decent amount of related prior work has been done on sentiment analysis of user reviews, documents, web blogs/articles and general phrase level sentiment analysis. These differ from twitter mainly because of the limit of 280 characters per review which forces the user to express opinion compressed in very short text. The best results reached in sentiment classification use supervised learning techniques such as Naive Bayes and Support Vector Machines, but the manual labeling required for the supervised approach is very expensive. Some work has been done on unsupervised and semi-supervised approaches, and there is a lot of room of improvement. Various researchers testing new features and classification techniques often just compare their results to base-line performance. There is a need of proper and formal comparisons between these results arrived through different features and classification techniques in order to select the best features and most efficient classification techniques for particular applications.

So in our system we are able to do sentimental analysis on reviews by some keywords like: good, bad, worst, happy etc.

##### About Technology we Used:

***Jupyter:***

Jupyter is a micro web framework written in Python. It is classified as a micro-framework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions.

##### NLTK(Natural Language Processing Toolkit):

The Natural Language Toolkit, or more commonly NLTK, is a suite of libraries and programs for symbolic and statistical natural language processing for English written in the Python programming language.

##### joblib:

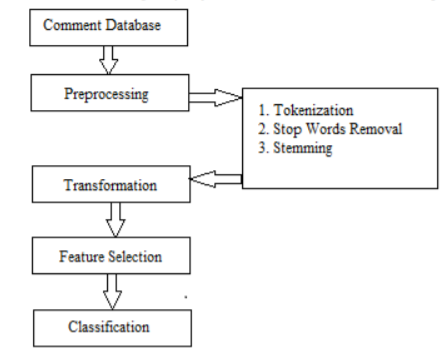
Joblib is a Python (2 and 3) library for processing textual data. It provides a simple API for diving into common natural language processing (NLP) tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation and more.



***WORKING PROCEDURE OF SENTIMENTAL ANALYSIS:***

An overview of sequential steps and techniques commonly used in sentiment classification approaches, as shown in Figure 4.Parts of speech is a model which aims to classify roles that means according to parts of speech has also been explored. In this model , information is used as part of a feature set which leads to sentiment classification on a dataset.

The model parts of speech is supposed to be the significant indicator of sentiment expression and which works on subjectivity detection that represents the close relationship between presence of adjectives and sentence subjectivity. But, many experimental results show that using only adjectives as features leads to worse performance.



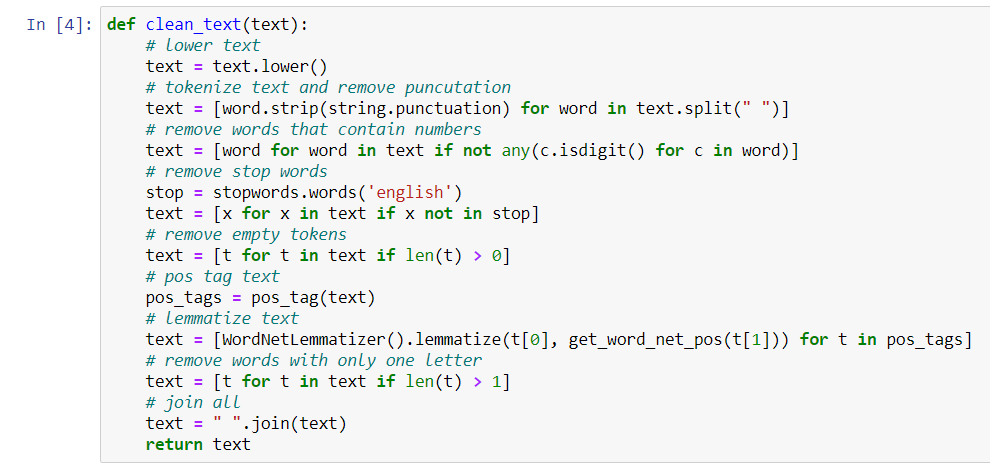
[Figure: 4] Steps and techniques used in sentiment classification.

1. **Text Preprocessing**

Pre processing of data is the process of preparing and cleaning the data of dataset for classification. Here is the hypothesis of having the data properly pre-processed: to reduce the noise in the text should help improve the performance of the classifier and speed up the classification process, thus aiding in real time sentiment analysis.

**1. Tokenization:**

Given input as character sequence, tokenization is a task of chopping it up into pieces called tokens and at the same time removing certain characters such as punctuation marks. A token is an instance of sequence of characters that are grouped together as a useful semantic unit for processing.



**2. Stop word removal:**

A stop-list is the name commonly given to a set or list of stop words. It is typically language specific, although it may contain words. A search engine or other natural language processing system may contain a variety of stop-lists, one per language, or it may contain a single stop-list that is multilingual. Some of the more frequently used stop words for English include "a", "of", "the", "I", "it", "you", and ”and” these are generally regarded as 'functional words' which do not carry meaning. When assessing the contents of natural language, the meaning can be conveyed more clearly by ignoring the functional words .Hence it is practical to remove those words which appear too often that support no information for the task.

**3. Stemming:**

It is the process for reducing derived words to their stem, or root form. Stemming programs are commonly referred to as stemmers or stemming algorithms. A simple stemmer looks up the inflected form in a lookup table, this kind of approach is simple and fast. The disadvantage is that all inflected forms must be explicitly listed in table.eg. “developed”, “development” , ”developing” are reduced to the stem “develop”.



**Transformation**

The weight of each word in the corpus is calculated with the help of TF-IDF, so that it is easy to determine what words in the corpus of documents might be more favorable to use in a further processing. TF-IDF calculates [9] values for each word in a document.

**Feature Selection**

Feature Selection is used to make classifiers more efficient by reducing the amount of data to be analyzed as well as identifying relevant features to be considered in classification process. Ideally, feature selection stage will refine features, which are input into a classification / learning process.

* Identify the parts of corpus to contribute to positive and negative sentiment.
* Join these parts of corpus in such a way that the document falls into one of these polar categories.

**Classification**

Goal of text classification is to classify data into predefined classes. Here they are positive and negative classes. Text classification is supervised learning problem.

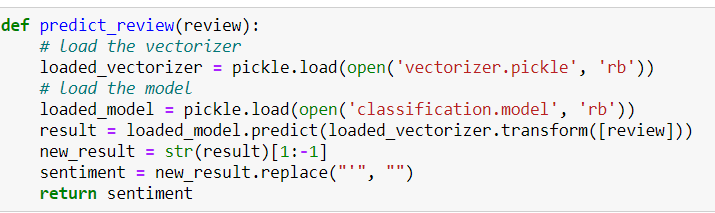
First step in text classification is transforming document which is in string format into format suitable for learning algorithm and classification task. In information retrieval it is found that word stem works well as representation unit. This leads to attributed value representation of text. Each word corresponds to feature with, number of times word occurs in document, as its value. Words are considered as features only if they are not stop words (like “and”, “or”, etc). Scaling the dimension of feature with IDF improves the performance[8].

*SVM*- Support vector machines are universal learners[8]. Remarkable property of SVM is that their ability to learn can be independent of dimensionality of feature space. SVM measures the complexity of Hypothesis based on margin that separates the plane and not number of features[8].

**SVM learning Algorithms for Text Categorization -**

SVM has defined input and output format. Input is a vector space and output is 0 or 1 (positive/negative).

Text document in original form are not suitable for learning. They are transformed into format which matches into input of machine learning algorithm input. For this preprocessing on text documents is carried out.Then we carryout transformation.Each word will correspond to one dimension and identical words to same dimension. As mentioned before we will see TF-IDF for this purpose. Now a machine learning algorithm is used for learning how to classify documents, i.e. creating a model for input-output mappings.SVM has been proved one of the powerful learning algorithm for text categorization[8].



**SVM Benefits:**

1. High Dimension Input Space - while text classification we have to deal with many features (may be more then 1000). Since SVM uses over fitting protection[12], which does not depend on number of features so they have ability to handle large number of features.
2. Document Vector Space - despite the high dimensionality of the representation, each of the document vectors contain only a few non-zero element[12].More Text Categorization problems are linearly separable[8].

**SVM Characteristics-**

1. ML algorithms typically use a vector space (attribute-value) [10] representation of examples, mostly the attributes correspond to words. However word-pairs or the position of a word in the text may have considerable information, and practically infinitely many features can be constructed which can enhance classification accuracy.

#### Classified Reviews:

We labeled the reviews in three classes according to sentiments expressed/observed in the reviews: positive, negative and neutral We gave the following guidelines to our labellers to help them in the labeling process:

**Positive:**

If the entire review has a positive/happy/excited/joyful attitude or if something is mentioned with positive connotations. Also if more than one sentiment is expressed in the review but the positive sentiment is more dominant. Example: “4 more years of being in shithole Australia then I move to the USA!

:D”.

##### Negative:

If the entire review has a negative/sad/displeased attitude or if something is mentioned with negative connotations. Also if more than one sentiment is expressed in the review but the negative sentiment is more dominant. Example: “I want an android now this iPhone is boring :S”.

##### Neutral:

If the user who wrote a review, expresses no personal sentiment/opinion in the review and merely transmits information. Advertisements of different products would be labeled under this category.

So let us discuss first command line models:- In command line models first we have created,

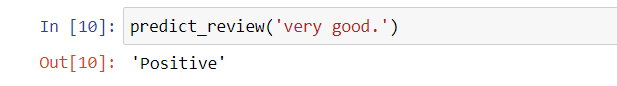
##### Keyword and count wise:-

In this we have made a python model in which user can provide a particular keyword along with limited no of counts to limit the no of fetching reviews.

##### Polarity and subjectivity of individual reviews:-

Here it fetched all real time reviews on stored keyword based on recent timestamp.

It prints time stamp, sentimental polarity of individual reviews along with its subjectivity to user and content of review. Like as shown below,



[Figure: 10]Command line Output

#### Chapter 4 Study and Survey of Related Systems

The research and analysis work took a good two three days to find out if the project could be completed within the allotted time frame or not. We went through different articles of Machine Learning and also went through different other articles to know which other technologies would be required to complete the project.

The development approach we took as a team to work out our project was to first research and study as many articles and content available from reliable sources. This was needed to be done as to check whether the project we took in hand was feasible or not. Also the research was to be done so as to find whether we could complete the given project in a given amount of time. Because time was also a priority and selecting a project which could not be completed in a given time was of no use.

We also divided our work into different parts so as to efficiently run the project. So first we decided to work on the backend portion as a whole and keeping the frontend portion of the project for the later part of the working stage. The idea was to initially develop a model which was a very basic model in the command prompt which would just accept the word and the number of reviews to be fetched and then give the output of the reviews in the prompt itself. And after developing the basic model, we then started further to add various features and make the frontend portion.

The main module of the system is the Twitter API. The Twitter API allows you to access the features of Twitter without having to go through the website interface. This can be useful for doing things like posting reviews or sending directed messages in an automated way with scripts. The Twitter platform offers access to corpus of data, via the API. It's important to note that the Twitter APIs are constantly evolving, and developing on the Twitter Platform is not a one-off event. The Twitter API (the term stands for Application Programming Interface) enables software developers to access and interact with public Twitter data. Developers can interact with this API by writing their own scripts or by using one of the open source libraries available in different programming languages.

The use of Twitter API was to fetch the real time reviews from the internet. After getting the keyword from the user, the reviews are fetched accordingly. The number of reviews to be analyzed is also taken from the user. So after applying our model, the exact number of reviews are fetched according the API and then the model works to produce the output of the review being positive, negative or neutral. The reviews fetched are stored in the database.

The main issue of the development was the approval of the developers at Twitter as it was a sensitive deal to allow us to fetch the reviews and also analyze them as giving the authority to anyone without any true meaning would cause damage and also be used for the wrong reasons. The sentiment analysis could be used for any politically motivated reason and hence the team at Twitter took lot of time to grant the access of the API. It took around a week of exchanging of emails to properly make them understand of why we wanted to use the Twitter API and what we wanted to do with the reviews fetched through the API.

While searching and analyzing the different articles and projects, we found out many different projects regarding the sentiment analysis of Twitter. We found a open-source project which had many different functionalities and features than our current project. In that project it was available that the users can know the sentiment of reviews based on a particular location. For example, the user enters the region as Gujarat, then the prompt appears to enter the number of reviews to be fetched and perform sentiment analysis. The model then would fetch the latest reviews irrespective of the keyword and would fetch the latest reviews of that particular region and perform sentiment analysis. We studied the project and found it useful to implement but because of time constraints and also the model being very complicated we decided not to pursue the model.

While searching and analyzing the different articles and projects, we found out many different projects regarding the sentiment analysis of Twitter.

We also found a model which could tell the machine that the user has used for the reviewing purpose. There are different machines or systems available to review such as from the Android phone, it could be an iPhone or it could be the standard Twitter Desktop version. The model could tell after the reviews have been fetched about the system that has been used for reviewing. And in turn this could be used for determining the different types of machines and hardware available in the particular area for the time being. By knowing the number of different types of machines and hardware available in the region, it could be found out where the twitter is most used from.

We also found different models which had many nice features for the Frontend which made it the model looked very attractive. One of the project had the 3D view of the world which could be rotated and pointed to a particular place for performing the sentiment analysis of the reviews of that particular region. Also it had a nice graphic for the plotting of the graph which had an effect of fading in the screen.

We also found the model where in the sentiments of reviews of different users can be compared regarding the same key-word. This feature is useful for knowing the opinion of two different users on the same topic or key-word. It could also be used to derive what the different leaders have to say on a particular topic.

#### Chapter 6 Conclusion and future Scope

Nowadays, sentiment analysis or opinion mining is a hot topic in machine learning. We are still far to detect the sentiments of corpus of texts very accurately because of the complexity in the English language and even more if we consider other languages such as Chinese.

In this project we tried to show the basic way of classifying reviews into positive or negative category using Naive Bayes as baseline and how language models are related to the Naive Bayes and can produce better results. We could further improve our classifier by trying to extract more features from the reviews, trying different kinds of features, tuning the parameters of the naïve Bayes classifier, or trying another classifier all together.

The task of sentiment analysis, especially in the domain of micro-bloging, is still in the developing stage and far from complete. So we propose a couple of ideas which we feel are worth exploring in the future and may result in further improved performance.

One more feature we have that is worth exploring is whether the information about relative position of word in a review has any effect on the performance of the classifier. Although studies conducted explored a similar feature and reported negative results, their results were based on reviews which are very different from reviews and they worked on an extremely simple model.

we are focusing on general sentiment analysis. There is potential of work in the field of sentiment analysis with partially known context. For example we noticed that users generally use our website for specific types of keywords which can be divided into a various distinct classes, namely: politics/politicians, celebrities, products/brands, sports/sportsmen and journalism/movies/music. So we can attempt to perform separate sentiment analysis on reviews that only belong to one of these classes (i.e. the training data would not be general but specific to one of these categories) and compare the results which we obtained if we apply general sentiment analysis on it instead.

**References:-**

1. Efthymios Kouloumpis and Johanna Moore,IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 4, No 3, July 2012
2. S. Batra and D. Rao, ”Entity Based Sentiment Analysis on Twitter”, Stanford University,2010
3. Saif M.Mohammad and Xiaodan zhu ,Sentiment Analysis on of social media texts:,2014
4. Ekaterina kochmar,University of Cambridge,at the Cambridge coding Academy Data Science.2016
5. Manju Venugopalan and Deepa Gupta ,Exploring Sentiment Analysis on Twitter Data, IEEE 2015
6. Brett Duncan and Yanqing Zhang, Neural Networks for Sentiment Analysis on Twitter.2017
7. Afroze Ibrahim Baqapuri, Twitter Sentiment Analysis: The Good the Bad and the OMG!, Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media.2011
8. Jin Bai, JianYun Nie. Using Language Models for Text Classification.
9. Apoorv Agarwal, Boyi Xie, Ilia Vovsha, Owen Rambow, Rebecca Passonneau. Sentiment Analysis of Twitter Data.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_