**ABSTRACT**

Text memes have become a popular form of communicating ideologies on social networking websites. Some of these have led to drastic consequences in the society and has also influenced lives of several people. Text memes are pieces of text that are copied and spread rapidly by internet users.

We will be focusing on the social platform - Twitter. The goal of this project is to classify short Twitter messages with respect to their sentiment using data mining techniques. Twitter Lingo, or tweets, are limited to 140 characters. The sentiment can refer to two different types: emotions and opinions. This project is solely focused on the sentiment of opinions. These opinions can be divided into three classes: positive, neutral and negative. The tweets are then classified with an algorithm to one of these three classes.

We hope to infer from this project, the various categories of text memes circulated amongst users everywhere around the world. Also, this helps us calculate the over-all reaction or sentiment of the public.

**TABLE OF CONTENTS**

**CHAPTER NO. TITLE PAGE NO.**

ABSTRACT iv

LIST OF TABLES vii

LIST OF FIGURES viii

LIST OF ABBREVIATIONS x

**1. INTRODUCTION 1**

1.1 ‘Meme’tic Engineering 1

1.2 Sentiment Analysis 1

1.3 Applications Of Sentiment Analysis 2

1.4 Language- Models Approach 3

**2. LITERATURE SURVEY 4**

2.1 Sentiment Classification using Machine Learning 5 2.2 Twitter Sentiment Analysis 5

2.3 Evaluation of Datasets 5

**3. SYSTEM DESIGN 7**

3.1 Overall System Design 7

3.2 System Architecture 8

3.3 Data Acquisition 9

3.4 Human Labelling 11

3.5 Feature Extraction/ Pre-processing 12

**4. ALGORITHMS 14**

**Classification Models** 14

4.1 Naive Bayes Model 14

4.2 Support Vector Machine 15

4.3 Logistic Regression 16

4.4 K-Nearest Neighbours 16

**5. IMPLEMENTATION OF PROPOSED SYSTEM 18**

5.1 Proposed System 18

5.2 Datasets 19

5.3 Step by Step Processing 22

**6. RESULTS AND DISCUSSIONS 24**

6.1 Output 24

6.2 Estimation Measures 26

6.3 Handling Sarcasm 28

**7. CONCLUSION AND FUTURE WORK 29**

7.1 Conclusion 29

7.2 Future Work 29

**REFERENCES 30**

**LIST OF TABLES**

**NO. TABLE NAME PAGE NO.**

2.1 Literature Survey 4

6.2 Estimation Parameters 26

6.2 Accuracy Results 27

**LIST OF FIGURES**

**NO. FIGURE NAME PAGE NO.**

3.1 Overall System Design 7

3.2 System Architecture 8

3.3 Generic Benchmark Dataset 10

3.4 Demonetization Dataset 11

5.1 Process Flow of Proposed System 18

5.2 Flow Diagram of Twitter Sentiment Analysis Tool 19

5.3 Screenshot of real- time Tweet retrieval 20

5.4 Snapshot of real- time tweets csv file 21

5.5 Code snippet for pre-processing 22

5.6 Classified tweets screenshot 23

6.1 Search page screenshot 24

6.2 Page displaying overall sentiment and classification of tweets 25

**LIST OF ABBREVIATIONS**

1. TF-IDF - Term frequency-Inverse Document Frequency

2. DB - Database

3. API - Application Programming Interface

4. CSV - Comma Separated Values

5. URL - Uniform Resource Locator

6. MNB - Multinomial Naïve Bayes

7. KNN - K-Nearest Neighbours

8. SVM - Support Vector Machine

9. LR - Logistic Regression

**CHAPTER 1**

**INTRODUCTION**

In the last couple of years the social medium Twitter has become more and more popular. Since Twitter is the most used microblogging website with about 500 million users and 340 million tweets a day, it is an interesting source of information. The messages, or in Twitter terms the tweets, are a way to share interests publicly or among a defined group. Twitter distinguishes itself from other social media by the limited message size. The maximum size of 140 characters restricts users in their writing. Twitter is therefore challenging their users to express their view in one or two key sentences. Twitter is widely adopted through all strata, it can be seen as a good reaction of what is happening around the world.

* 1. **‘Meme’tic Engineering**

A MEME is an idea that spreads like a virus by word of mouth, email, blogs, etc.

Our project focuses on text memes which are pieces of text, copied and spread rapidly by internet users. ‘Meme’tic engineering refers to the process of handling memes using various methods such as searching, classifying, processing, etc. [5]

* 1. **Sentiment Analysis**

Sentiment analysis involves in several research fields - natural language processing, computational linguistics and text analysis. It refers to the extraction of subjective information from raw data, often in text form which consist of different kinds of sentiments. The sentiment can refer to opinions or emotions. These two types are slightly related but there is an evident difference. In sentiment analysis based on opinions, a distinction is made between positive and negative opinions. On the other hand, sentiment analysis based on emotions, is about the distinction between different kinds of emotions. The sentiment analysis that is considered in this project is based on opinions and is often referred to as opinion mining. Sentiment analysis aims to determine the attitude of the opinion holder with respect to a subject. Our application also determines the overall sentiment of a topic.

* 1. **Applications of Sentiment Analysis**

Following are the major applications of sentiment analysis in real world scenarios.

**Product and Service reviews** - The most common application of sentiment analysis is in the area of reviews of consumer products and services. There are many websites that provide automated summaries of reviews about products and about their specific aspects. A notable example of that is “Google Product Search”.

**Reputation Monitoring** - Twitter and Facebook are a focal point of many sentiment analysis applications. The most common application is monitoring the reputation of a specific brand on Twitter and/or Facebook.

**Result prediction** - By analysing sentiments from relevant sources, one can predict the probable outcome of a particular event. For instance, sentiment analysis can provide substantial value to candidates running for various positions. It enables campaign managers to track how voters feel about different issues and how they relate to the speeches and actions of the candidates.

**Decision making** - Another important application is that sentiment analysis can be used as an important factor assisting the decision making systems. For instance, in the financial markets investment. There are numerous news items, articles, blogs, and tweets about each public company. A sentiment analysis system can use these various sources to find articles that discuss the companies and aggregate the sentiment about them as a single score that can be used by an automated trading system. One such system is The Stock Sonar. [9]

* 1. **Language Models Approach**

In this approach, the classification is done by building n-gram language models. A gram is a token or lexicon taken into consideration for training and classification. N-gram represents a set of such chosen lexicons. Generally, in this approach frequency of n-grams are used. In traditional information retrieval and topic-oriented classification, frequency of n-grams gives better results. The frequency is converted to TF-IDF [3.5] to take term's importance in the document to be classified. [9]

**CHAPTER 2**

**LITERATURE SURVEY**

|  |  |  |
| --- | --- | --- |
| **S.NO.** | **PAPER** | **DESCRIPTION** |
| 1 | Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan. "Thumbs up?: sentiment classification using machine learning techniques." In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pp. 79-86. Association for Computational Linguistics, 2002. | * classifying documents not by topic, but by overall sentiment * e.g: determining whether a review is positive or negative. |
| 2 | Kouloumpis, Efthymios, Theresa Wilson, and Johanna D. Moore. "Twitter sentiment analysis: The good the bad and the omg!." *Icwsm* 11 (2011): 538-541. | * investigate the utility of linguistic features * detecting the sentiment of Twitter messages |
| 3 | Saif, Hassan, Miriam Fernandez, Yulan He, and Harith Alani. "Evaluation datasets for Twitter sentiment analysis: A survey and a new dataset, the STS-Gold." (2013). | * organisations and individuals a fast and effective way to monitor the public’s feelings * investigate the pair-wise correlation among these dimensions |

*Table 2.1. Literature Survey*

**2.1 Sentiment classification using Machine Learning**

In the report ‘Thumbs up? Sentiment Classification using Machine Learning Techniques’, the documents are classified by overall sentiment rather than by topic. The Machine Learning techniques employed by them include Naive Bayes, maximum entropy classification, and support vector machines. Movie reviews have been grouped into positive and negative based on individual connotations. This gives a clear understanding of making use of appropriate algorithms for the required application. [7]

**2.2 Twitter Sentiment Analysis**

In the study, “Twitter sentiment analysis: The good the bad and the omg!” investigation of the linguistic features has been carried out for detecting the sentiment of twitter messages. Research has been performed to evaluate the usefulness of existing lexical resources as well as features that capture information about the informal and creative language used in microblogging. The conclusion derived from the study shows that the parts-of-speech method may not be useful for sentiment analysis in the microblogging domain. [4]

**2.3 Evaluation of Datasets**

Eight publicly available and manually annotated datasets have been evaluated in the report, “Evaluation datasets for Twitter sentiment analysis: A survey and a new dataset, the STS-Gold” to assess the performance of sentiment analysis on Twitter. STS-Gold is a new evaluation dataset that has been created where tweets and targets (entities) are annotated individually and therefore may present different sentiment labels. This paper also provides a comparative study of the various datasets along several dimensions including: total number of tweets, vocabulary size and sparsity which enables efficient procedures for sentiment analysis. [8]

**CHAPTER 3**

**SYSTEM DESIGN**

**3.1 Overall System Design**



*Fig. 3.1. Overall System Design*

The corpus refers to the dataset which has the stored tweets that were retrieved from Twitter. These tweets are then cleaned and pre-processed. The sentiment analysis is performed using appropriate classification algorithms in order to group them into positive, negative and neutral labels.

**3.2 System Architecture**

Analysis

* Retrieve
* Pre-processing

Classification

(Algorithm- Nearest neighbour & Multinomial)

* Positive (1)
* Negative(-1)
* Neutral (0)

Twitter DB-Dataset

*Fig. 3.2. System Architecture*

The architecture of the system can be broken down into four components as shown in figure 3.2. The modules are as follows, Twitter Database module, Analysis module, Classification and Result modules. The tweets are retrieved from the Twitter DB and then cleaning and pre-processing stages follow. Once this is done, the text memes are categorised using appropriate classification algorithms. This results in a set of labelled tweets.

**3.3 Data Acquisition**

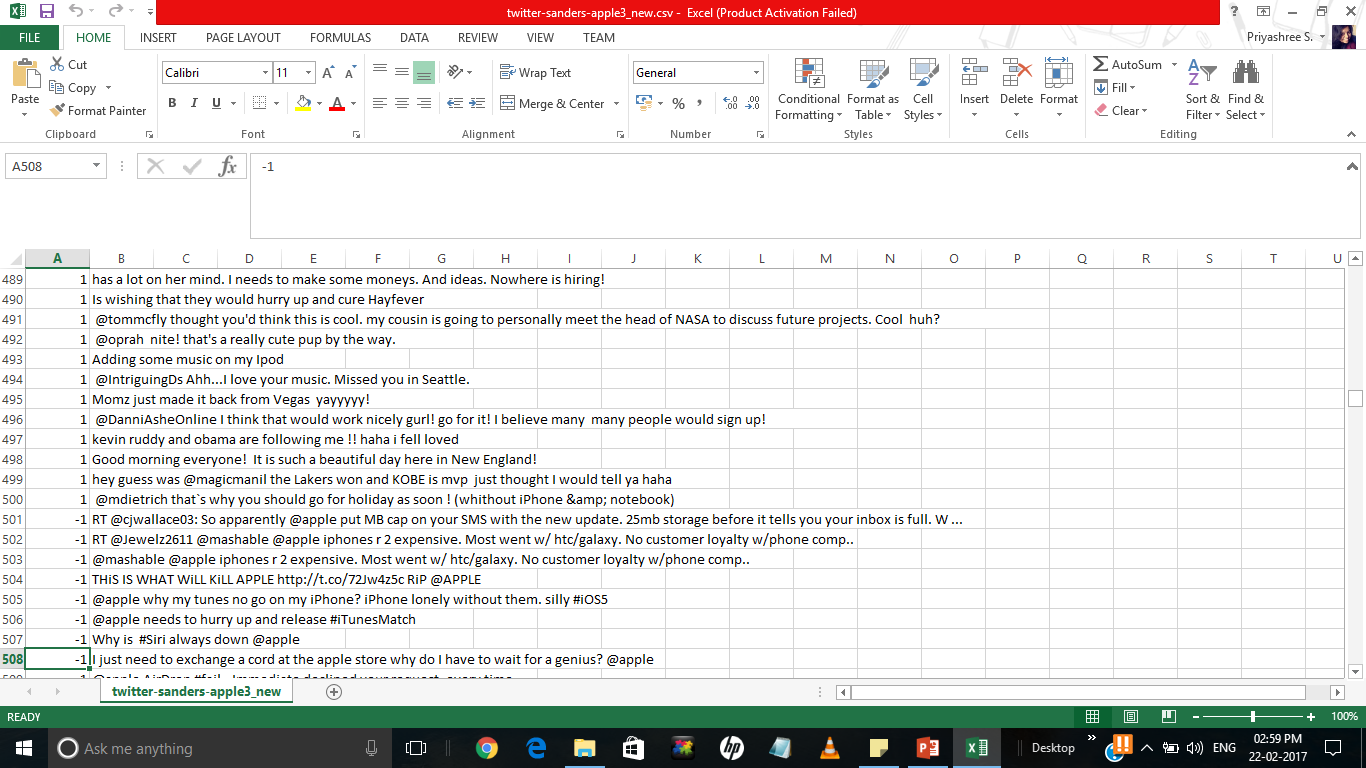
Both the Twitter [REST API](https://dev.twitter.com/docs/api/1/get/search) and [streaming API](https://dev.twitter.com/docs/streaming-apis) are available in “tweepy”. Data in the form of raw tweets is acquired by using the python library tweepy which provides a package for simple twitter RESTful API. Tweepy is open-sourced, which enables Python to communicate with the Twitter platform and use its API. An easy-to-use Python library for accessing the Twitter API [14]. Tweepy supports OAuth authentication. Retrieval of tweets is done using a search function available in tweepy. The required keyword is specified along with the number of tweets to be extracted in order to obtain the necessary set of tweets.

Our training dataset is as follows:

**Name:** **twitter-sanders.csv [8]**

The dataset contains 1524 tweets

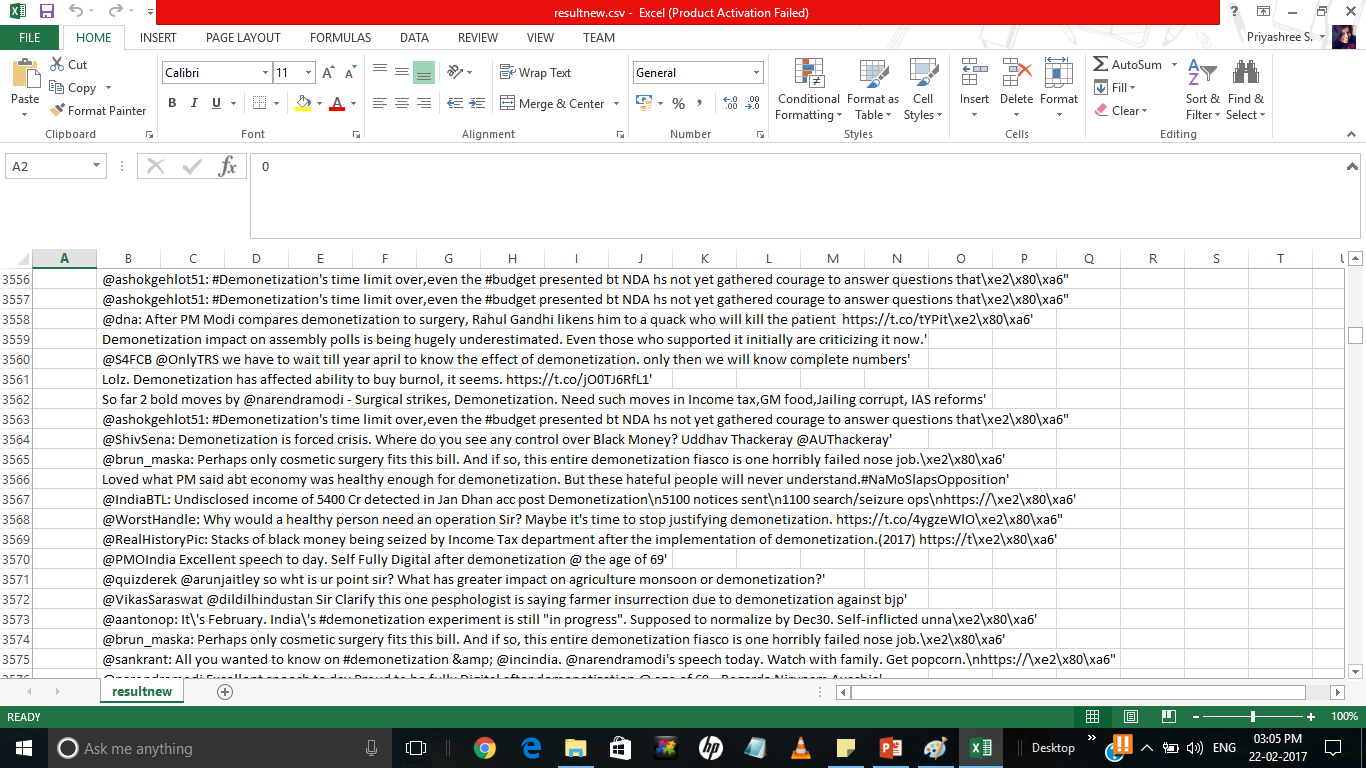
Format: semicolon-separated CSV file

Columns: **class:** positive(1), negative(-1), neutral(0); **tweet**: the tweet text  *Fig. 3.3. Generic Benchmark Dataset*

Our test dataset is as follows:

**Name:** **demonetization.csv**

Format: semicolon-separated CSV file



*Fig. 3.4. Demonetization Dataset*

**3.4 Human Labelling**

We labelled the tweets into three classes according to sentiments expressed/observed in the tweets: positive, negative and neutral. We used the following guidelines for the labelling process:

* **Positive**: If the entire tweet 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 tweet but the positive sentiment is more dominant.
* **Negative**: If the entire tweet has a negative/sad/displeased attitude or if something is mentioned with negative connotations. Also if more than one sentiment is expressed in the tweet but the negative sentiment is more dominant.
* **Neutral**: If the creator of tweet expresses no personal sentiment/opinion in the tweet and merely transmits information. Advertisements of different products would be labelled under this category. [1]

**3.5 Feature Extraction/Pre-processing**

We need to extract useful features from the dataset which can be used in the process of classification. The following text formatting techniques will aid us in feature extraction:

• **Tokenization:** It is the process of breaking up a stream of text into words, symbols and other meaningful elements called “tokens”. Tokens can be separated by whitespace characters and/or punctuation characters. It is done so that we can look at tokens as individual components that make up a tweet.

• **URL’s and user references** (identified by tokens “http” and “@”) are removed since we are interested in only analysing the text of the tweet.

• **Lowercase Conversion:** Tweet is normalized by converting it to lowercase which makes it’s comparison with an English dictionary easier.

• **Stop-words removal:** Stop words are class of some extremely common words which hold no additional information when used in a text and are thus claimed to be useless. Examples include “a”, “an”, “the”, “he”, “she”, “by”, “on”, etc. It is sometimes convenient to remove these words because they hold no additional information since they are used almost equally in all classes of text.

For this process we make use of **TF-IDF**:

The tf-idf (term frequency-inverse document frequency) measure is a statistic that reflects the importance of a word across a set of documents. This measure is composed of two individual measures: the term frequency (Eq. 3.1) and the inverse document frequency (Eq. 3.2).  It is often used as a [weighting factor](https://en.wikipedia.org/w/index.php?title=Weighting_factor&action=edit&redlink=1) in information retrieval, [text-mining](https://en.wikipedia.org/wiki/Text_mining), and [user modelling](https://en.wikipedia.org/wiki/User_modeling). The tf-idf value (Eq. 3.3) increases [proportionally](https://en.wikipedia.org/wiki/Proportionality_(mathematics)) to the number of times a word appears in the document. Tf–idf can be successfully used for [stop-words](https://en.wikipedia.org/wiki/Stop-words) filtering in various subject fields including [text summarization](https://en.wikipedia.org/wiki/Automatic_summarization) and classification.[1][3][9]

**Term Frequency:**

*tf(w,d)=|{w ∈ d}| (3.1)*

**Inverse Document Frequency:**

*idf(w,D)=log |D| (3.2)*

*df(w,D)*

**Tf-idf:**

*tf -idf(w, d, D) = tf (w,d) \* idf(w,D) (3.3)*

**CHAPTER 4**

**ALGORITHMS**

**Classification Models**

In this chapter, the various classification algorithms implemented and their functionalities in this application are described in detail.

**4.1 Naive Bayes Model**

The Naive Bayesian classifier is based on Bayes’ theorem with independence assumptions between predictors. A Naive Bayesian model is easy to build, with no complicated iterative parameter estimation which makes it particularly useful for very large datasets. Despite its simplicity, the Naive Bayesian classifier often does surprisingly well and is widely used because it often outperforms more sophisticated classification methods.

Bayes theorem provides a way of calculating the posterior probability, P(c|x), from P(c), P(x), and P(x|c). Naive Bayes classifier assume that the effect of the value of a predictor (x) on a given class (c) is independent of the values of other predictors. This assumption is called class conditional independence.



P(c|x) is the posterior probability of class (target) given predictor (attribute).

P(c) is the prior probability of class.

P(x|c) is the likelihood which is the probability of predictor given class.

P(x) is the prior probability of predictor.

There are two naive Bayes' models: Multinomial and Bernoulli. The multinomial naive Bayes model generates one term from the vocabulary in each position of the document. The Bernoulli model generates an indicator for each term of the dataset. The value 1 indicates that a term is present and a 0 indicates absence. The Bernoulli model does not count the multiple occurrence of terms whereas multinomial naive Bayes does.

We have used Multinomial naïve Bayes classifier since it performs faster than other classifiers apart from weighing features independently. We have utilised the ‘MultinomialNB’ package available on scikit learn (sklearn) in Python. [9] [10]

**4.2 Support Vector Machine**

Support Vector Machine (SVM) is a supervised learning method that can be used to carry out general regression and classification. A support vector machine constructs a hyper plane or set of hyper planes in a high or infinite dimensional space, which can be used for classification, regression or other tasks. Intuitively, a good separation is achieved by the hyper plane that has the largest distance to the nearest training data points of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier.

We checked the working of the SVM classifier in our application and observed that the results obtained were less accurate when compared to other classifiers. The reason behind this is that SVM is fundamentally a two-class classifier whereas our project deals with three different classes namely positive, negative and neutral. We made use of the ‘SVC’ package available on scikit learn (sklearn) in Python. [3] [11]

**4.3 Logistic Regression**

Logistic regression measures the relationship between the categorical dependent variable and one or more independent variables by estimating probabilities using a [logistic function](https://en.wikipedia.org/wiki/Logistic_function), which is the cumulative [logistic distribution](https://en.wikipedia.org/wiki/Logistic_distribution).

It is a discriminative model which uses a class boundary membership just like SVM. The probability of (y|x) is computed directly from the training data. For a small training dataset, overfitting of data takes place and hence logistic regression is not suitable for such cases.

We made use of this classifier from ‘LogisticRegression()’ in the package ‘linear\_model’ of sklearn in Python. [12]

**4.4 K-Nearest Neighbours**

In [pattern recognition](https://en.wikipedia.org/wiki/Pattern_recognition), the k-nearest neighbours algorithm (k-NN) is a [non-parametric](https://en.wikipedia.org/wiki/Non-parametric_statistics) method used for [classification](https://en.wikipedia.org/wiki/Statistical_classification). The input consists of the k closest training examples in the [feature space](https://en.wikipedia.org/wiki/Feature_space).

In k-NN classification, the output is a class membership. An object is classified by a majority vote of its neighbours, with the object being assigned to the class most common among its k nearest neighbours (k is a positive [integer](https://en.wikipedia.org/wiki/Integer), typically small). If k = 1, then the object is simply assigned to the class of that single nearest neighbour.

The training examples are vectors in a multidimensional feature space, each with a class label. The training phase of the algorithm consists only of storing the [feature vectors](https://en.wikipedia.org/wiki/Feature_vector) and class labels of the training samples. In the classification phase, k is a user-defined constant, and an unlabelled vector (a query or test point) is classified by assigning the label which is most frequent among the k training samples nearest to that query point. A commonly used distance metric for [continuous variables](https://en.wikipedia.org/wiki/Continuous_variable) is [Euclidean distance](https://en.wikipedia.org/wiki/Euclidean_distance). For discrete variables, such as for text classification, another metric can be used, such as the overlap metric ([Hamming distance](https://en.wikipedia.org/wiki/Hamming_distance)). In text classification nearest neighbour achieves reasonable results in combination with the tf-idf measure. [3]

We have utilised this classifier from ‘KNeighborsClassifier’ package of sklearn in Python. [13] We noticed that the maximum accuracy was obtained using this classifier because:

* Finding the nearest neighbours in a smaller dataset is much easier
* It uses a similarity measure to classify new cases
* The similarity between the training and test dataset will be effective in predicting the class of the data

**CHAPTER 5**

**IMPLEMENTATION OF PROPOSED SYSTEM**

**5.1 Proposed System**

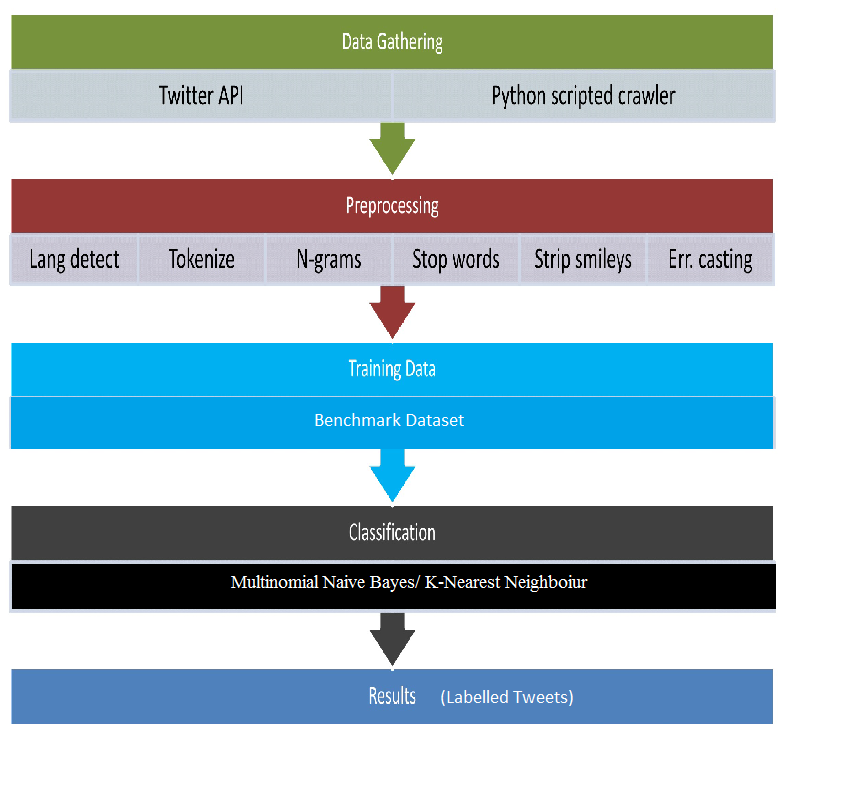


Fig. 5.1 Process flow of proposed system

The Twitter Lingo analysis starts with data collection using Twitter API, tweepy which acts as an interface between Python and Twitter. The data is searched using the Python scripted Crawler in the web. The gathered data is then sent to the pre-processing unit where the tweets are cleaned by eliminating stop words apart from converting to lowercase, tokenizing and limiting the language to English only. The classification of the pre-processed data is carried out using different algorithms in Python. The various accuracies obtained are compared and we make use of the best algorithm to classify the tweets in real time.

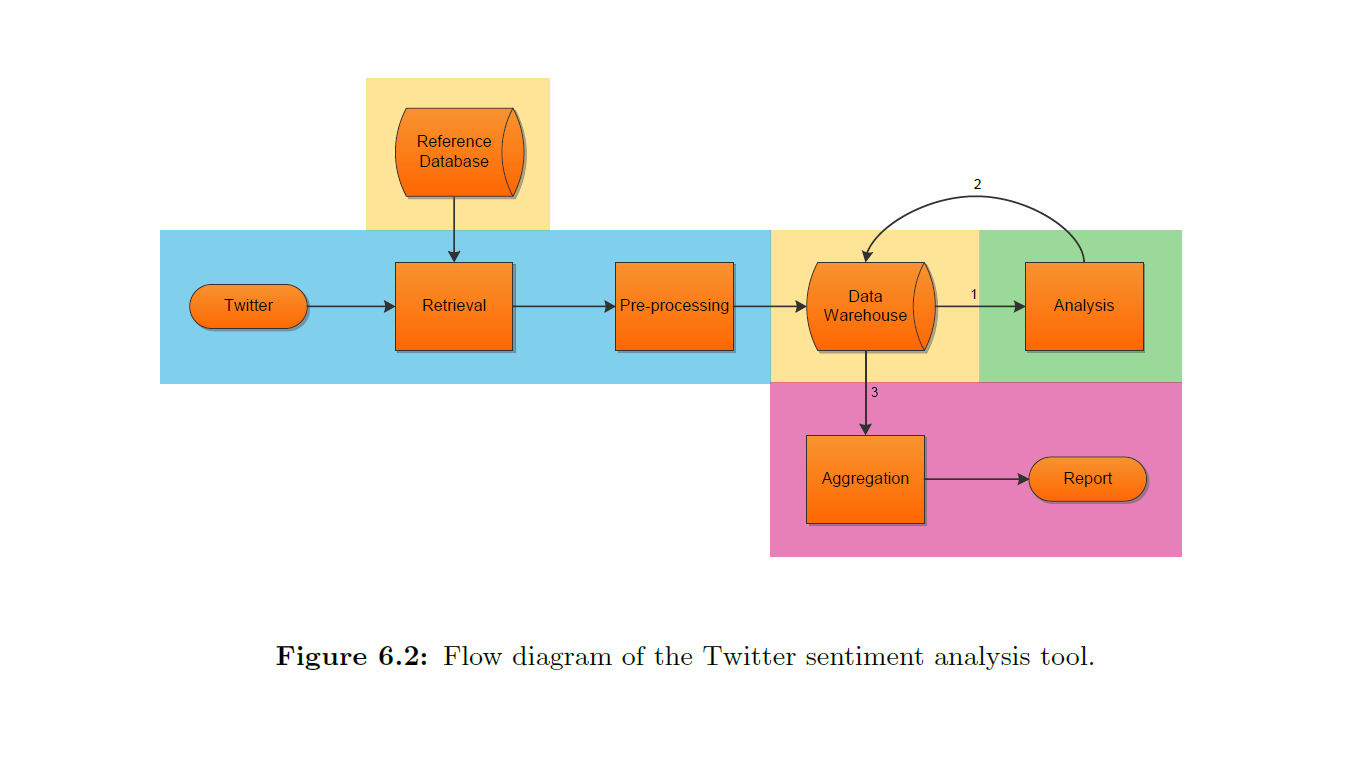


Fig. 5.2 Flow diagram of twitter sentiment analysis tool

**5.2 Datasets**

Our training dataset consists of 1524 tweets which comprises of a combination of positive, negative and neutral labels. This dataset is a Comma Separated (CSV) file and was extracted from generic benchmark datasets. [8]

**Real time dataset using Twitter API:**

****

Fig. 5.3 Screenshot of real time tweet retrieval

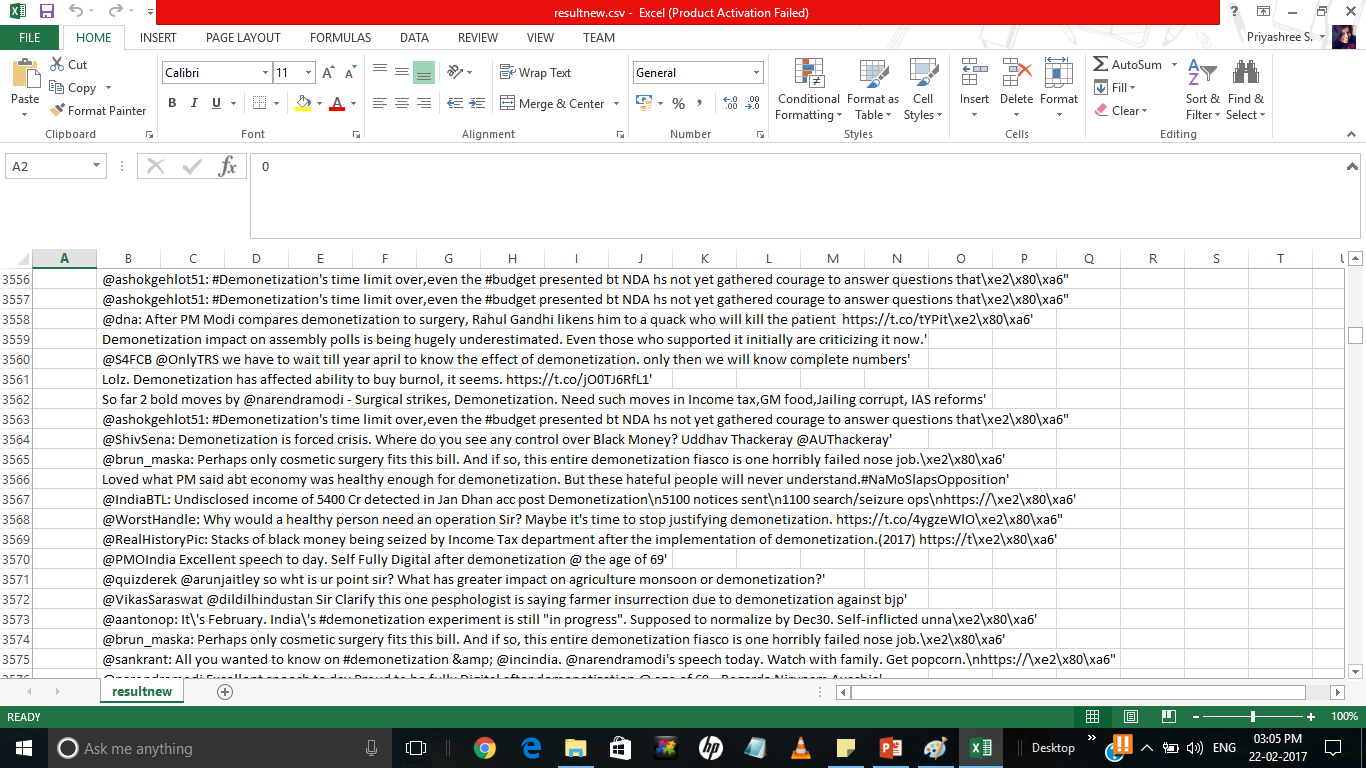
****

Fig. 5.4 Snapshot of real time tweets csv file

The tweets are gathered at real time using the search function available in tweepy. The access tokens and consumer keys have been used to connect the Twitter handle. This helps us retrieve various tweets based on a particular search keyword entered by the user. The collected tweets are stored in a csv file, then pre-processed and classified.

**5.3 Step by step processing**

**Original tweet (sample):**

*@myindmakers: Modi has reenergized the party cadres with a resounding rebuttal in the Parliament. https://t.co/81chOtjdtT'*

****

Fig. 5.5 Code snippet for pre-processing

Classified as positive (1)

**Pre-processed tweet:**

*modi has reenergized the party cadres with a resounding rebuttal in the parliament*

**Classified tweet:**

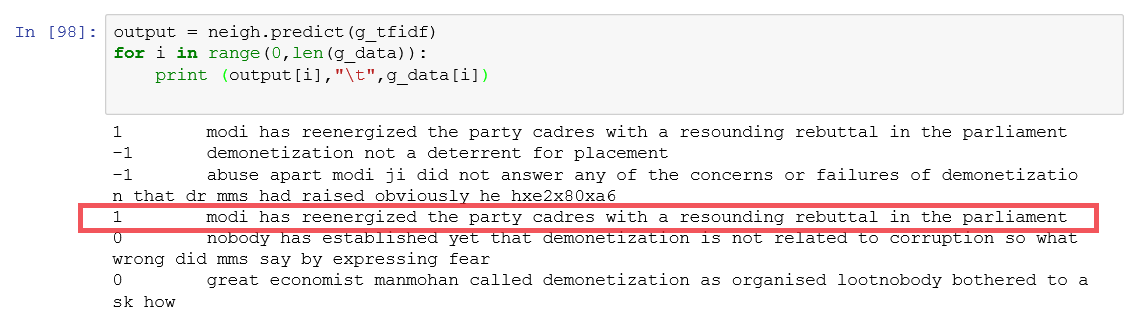


Fig. 5.6 Classified tweets screenshot

Classified -positive (1)

Fig. 5.5 and 5.6 show the step by step processing of the tweets and its classification along with the corresponding label.

**CHAPTER 6**

**RESULTS AND DISCUSSIONS**

The various experimental results that were recorded during the development of this application are presented in this chapter.

**6.1 Output**



Fig. 6.1 Search page screenshot

This page is where the user enters the required search keyword, based on which the Twitter API collects the corresponding set of Tweets. In Fig.6.1, we have entered the keyword as ‘demonetization’ in order to retrieve the relevant data.

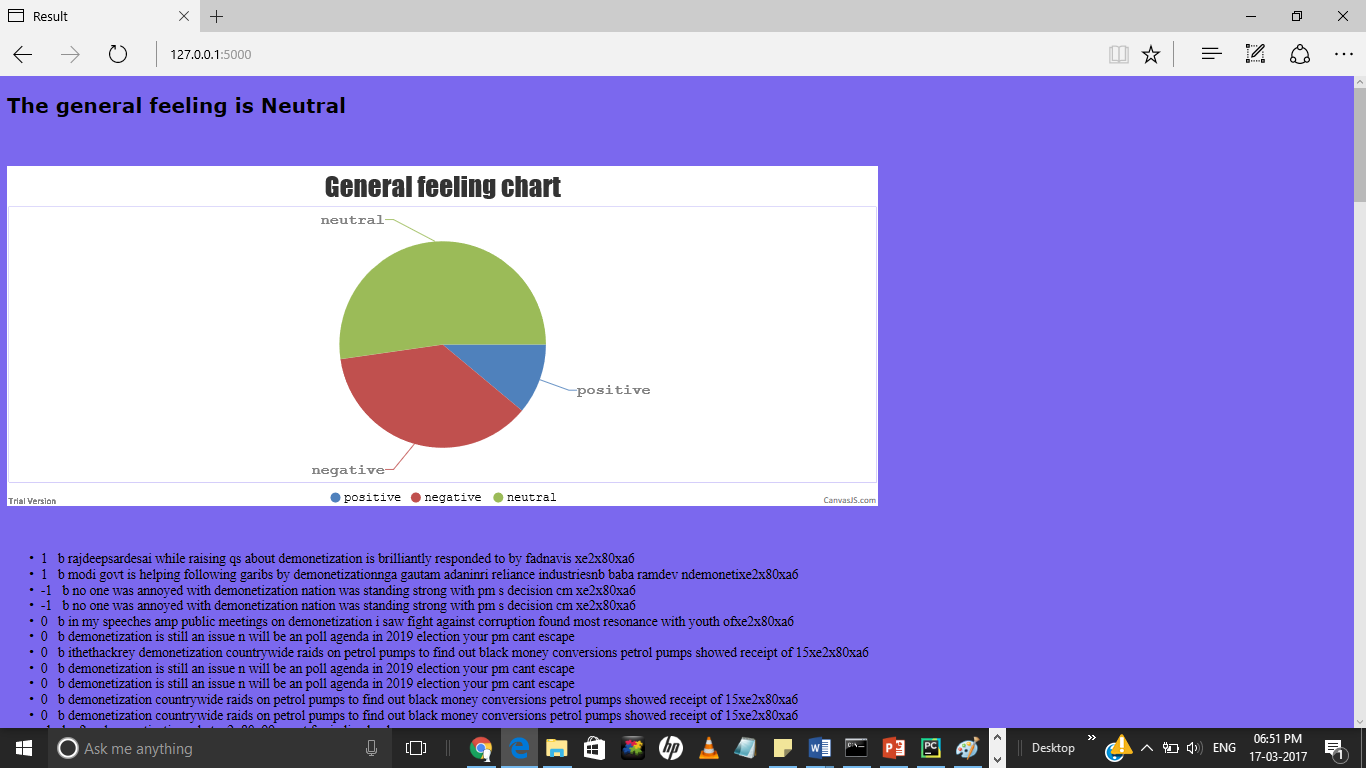


Fig. 6.2 Page displaying overall sentiment and classification of tweets

Fig.6.2, shows the pie-chart which provides the visualization of the overall sentiment on the topic ‘demonetization’. The three different class labels positive, negative and neutral have been indicated with appropriate colour codes and percentages. The classified set of tweets have also been incorporated in the output with 1 denoting a positive tweet, -1 for negative and 0 for neutral.

**6.2 Estimation Measures**

True positive – TP

False positive - FP

True negative - TN

False negative - FN

Table 6.1 Estimation Parameters

The different performance evaluation metrics are[3]:

* Accuracy - It is a measure of statistical bias and a description of systematic errors. It is given as:

Accuracy = (TP+TN)/(TP+FP+TN+FN)

* Precision - Proportion of instances that are truly of a class divided by the total instances classified as that class also called Positive Predictive Value(PPV)

PPV = TP/(TP+FP)

* Recall - proportion of instances classified as a given class divided by the actual total in that class (equivalent to TP rate)

Recall = TP/(TP+FN)

* F-Score - A combined measure for precision and recall calculated as

F = (2\*PPV\*Recall)/(PPV+Recall)

We have opted for Accuracy measure as the performance estimation metric as it considers all the parameters while evaluating.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Algorithm/Data** | **Support Vector Classifier** | **Logistic Regression** | **Multinomial Naïve Bayes** | **K-Nearest Neighbours Classifier** |
| **twitter-sanders**  **(training set)** | 31.80% | 72.45% | 73.77% | 62.29% |
| **demonetization (test set)** | 38.77% | 54.08% | 65.30% | 66.32% |

Table 6.2 Accuracy Results

From Table 6.2, we observe that the accuracy for K-Nearest Neighbours (KNN) classifier is the highest followed by Multinomial naïve Bayes (MNB) classifier.

KNN works well for comparatively smaller datasets like the one used in our project since it is easier to find the nearest neighbours. This makes the computation more efficient for the text dataset used. Also MNB works well when compared to SVM or Logistic Regression because it is not just a two-class classifier like SVM. And, each feature is weighed independently unlike LR and does not over fit data. MNB is a generative model since it selects the most likely features from the given data.

**6.3 Handling Sarcasm**

In the application we have implemented, the sarcastic comments are classified under the neutral class label. Handling sarcasm, irony and double negatives separately is difficult. This is caused by the difficulty in recognizing those figures of speech purely based on text. Although it is possible to recognize some figures of speech, it results in a low accuracy. The low accuracy means that more tweets are recognized as a figure of speech than they actually are. Therefore tweets without figures of speech are indicated as tweets with a figure of speech, which leads to a higher false positive rate. Sarcasm and irony are almost impossible to classify correctly, since these are very subtle in text. The only way these could be found is by adding a special hash-tag to the tweet.[3]

**CHAPTER 7**

**CONCLUSION AND FUTURE WORK**

**7.1 Conclusion**

Twitter is a platform where highly unstructured, short messages are available. This project successfully classifies tweets based on a search topic and analyses the overall impact that an issue has on the society. [3] Most companies and businesses these days focus on evaluating the sentiments of people on various issues. We have developed a small-scale working application of such an analyser and studied its functionalities. Classification of text memes into positive, negative and neutral gives us a clearer perspective of reactions of the general public on different areas. The KNN and MNB classifier algorithms prove to be accurate in making the necessary classifications in this application. Thus, we have devised an efficient sentiment evaluator which is in demand today in all enterprises.

**7.2 Future Work**

This project could be further expanded by trying to handle ambiguous tweets which do not fit into the positive, negative or neutral label. Sarcastic messages involve conveying a negative meaning by using positive words or even intensified positive words. Detecting sarcasm will help in refining a sentiment analysis model, but there are a number of conceptual and technical challenges. It is currently being researched on, in the field of Deep Learning. [3]

Due to the lack of a multi-lingual lexical dictionary, it is currently not feasible to develop a multi-language based sentiment analyser. Further research can be carried out in making the classifiers language independent. [9]

**REFERENCES**

1. Baqapuri, Afroze Ibrahim. "Twitter Sentiment Analysis." *arXiv preprint arXiv:1509.04219* (2015).
2. Cervesato, Iliano, Majd Sakr, Mark Stehlik, Bernardine Dias, and Lynn Carter. "2009 Senior Thesis Project Reports." (2010).
3. Groot, R. de. "Data mining for tweet sentiment classification." Master's thesis, 2012.
4. Kouloumpis, Efthymios, Theresa Wilson, and Johanna D. Moore. "Twitter sentiment analysis: The good the bad and the omg!.” *Icwsm* 11 (2011): 538-541.
5. meburn.com/2010/09/text-mining-reveals-best-and-worst-words-used-in-social-media/
6. Mejova, Yelena Aleksandrovna. "Sentiment analysis within and across social media streams." (2012).
7. Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan. "Thumbs up?: sentiment classification using machine learning techniques." In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pp. 79-86. Association for Computational Linguistics, 2002.Domingos, Pedro. "A few useful things to know about machine learning." *Communications of the ACM* 55.10 (2012): 78-87.
8. Saif, Hassan, Miriam Fernandez, Yulan He, and Harith Alani. "Evaluation datasets for Twitter sentiment analysis: A survey and a new dataset, the STS-Gold." (2013).
9. Thakkar, Mr Harsh Vrajesh. "Twitter Sentiment Analysis using Hybrid Naïve Bayes."
10. “Naïve Bayes Model” ,

http://scikit-learn.org/stable/modules/naive\_bayes.html

1. “Support Vector Machine Classifier”

http://scikit-learn.org/stable/modules/svm.html

1. “Logistic Regression”,

http://scikitlearn.org/stable/modules/generated/sklearn.linear\_model.LogisticRegression.html

1. “K-Nearest Neighbours”,

<http://scikitlearn.org/stable/modules/generated/sklearn.neighbors.KNeighboursClassifier>. html

1. “Twitter API”,

http://www.tweepy.org/

1. “PyCharm”,

https://www.jetbrains.com/pycharm/

1. “Pandas”,

https://pypi.python.org/pypi/pandas/

1. “NLTK Tokenizer”,

http://www.nltk.org/

1. “Twitter Application”,

https://dev.twitter.com/

https://apps.twitter.com/