VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“Jnana Sangama”, Belagavi – 590 018

A PROJECT REPORT ON

“SENTIMENT ANALYSIS ON TWITTER DATA”

Submitted in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

BY

PRAVIND THAKUR (1NH12CS091)

PRASANNA S (1NH12CS103)

SHUBHAM (1NH12CS115)

SOORAJ T.K (1NH12CS119)

Under the guidance of

Ms. Anjana Sharma

(Senior Assistant Professor, Dept. of CSE, NHCE)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NEW HORIZON COLLEGE OF ENGINEERING

(ISO-9001:2000 certified, Accredited by NAAC ‘A’,
Permanently affiliated to VTU)
Outer Ring Road, Panathur Post, Near Marathalli,
Bangalore – 560103

NEW HORIZON COLLEGE OF ENGINEERING

(ISO-9001:2000 certified, Accredited by NAAC 'A'
Permanently affiliated to VTU)
Outer Ring Road, Panathur Post, Near Marathalli,
Bangalore-560 103

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NHCE
In Pursuit of Excellence

CERTIFICATE

Certified that the project work entitled "**SENTIMENT ANALYSIS ON TWITTER DATA**" carried out by **PRAVIND THAKUR (1NH12CS091), PRASANNA S (1NH12CS103), SHUBHAM (1NH12CS115) and SOORAJ T.K (1NH12CS119)** bonafide students of NEW **HORIZON COLLEGE OF ENGINEERING** in partial fulfillment for the award of **Bachelor of Engineering** in **Computer Science and Engineering** of the Visvesvaraya Technological University, Belagavi during the year 2015-2016. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the department library. The project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the Degree.

Name & Signature of Guide	Name Signature of HOD	Signature of Principal
(Ms. Anjana Sharma)	(Dr. Prashanth C.S.R.)	(Dr. Manjunatha)

External Viva

Name of Examiner	Signature with date
1.	
2.	

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be, but impossible without the mention of the people who made it possible, whose constant guidance and encouragement crowned our efforts with success.

We thank the management, **Dr. Mohan Manghnani**, Chairman of NEW HORIZON EDUCTIONAL INSTITUTIONS for providing necessary infrastructure and creating good environment.

We also record here the constant encouragement and facilities extended to us by **Dr. Manjunatha**, Principal, NHCE and **Dr. Prashanth.C.S.R,** Dean Academics, Head of the Department of Computer Science and Engineering. We extend our sincere gratitude to them.

We express our gratitude to **Ms.Anjana Sharma**, our project guide for constantly monitoring the development of the project and setting up precise deadlines. Their valuable suggestions were the motivating factors in completing the work.

We would also like to express our gratitude to NHCE and to all our external guides at NHCE for their continuous guidance and motivation.

Finally a note of thanks to the teaching and non-teaching staff of Computer Science and Engineering Department for their cooperation extended to us and our friends, who helped us directly or indirectly in the course of the project work.

PRAVIND THAKUR (1NH12CS091)
PRASANNA S (1NH12CS103)
SHUBHAM (1NH12CS115)
SOORAJ T.K (1NH12CS119)

ABSTRACT

An important point of information gathering behaviour of humans has always been to find out what other people think. With the growing availability and popularity of opinion rich resources such as online review sites and personal blogs, new opportunities and challenges arise as people now can, and do, actively use information technologies to seek out and understand the opinions of others.

Sentiment analysis deals with identifying and classifying opinions or sentiments expressed in source text. Social media is generating a vast amount of sentiment rich data in the form of tweets, status updates, blog posts etc. Sentiment analysis of this user generated data is very useful in knowing the opinion of the crowd. Twitter sentiment analysis is difficult compared to general sentiment analysis due to the presence of slang words and misspellings.

This project contributes to the sentiment analysis for classification which is helpful to analyse the information in the form of the number of tweets, where opinions are highly unstructured and sentiment are either positive, negative or neutral. For this, we first pre-process the dataset, after that extract the dataset that have some meaning. This dataset is called feature vector. Then select the feature vector list and thereafter apply the machine learning based classification algorithms namely: **Naïve Bayes** along with six other algorithms namely: "*Multinomial Naïve Bayes, Bernoulli Naïve Bayes, Logistic Regression, Stochastic Gradient Descent, Support vector clustering, Linear Support vector clustering*" which is used to evaluate the confidence level of the sentiment.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 MACHINE LEARNING

"Optimizing a performance criterion using example data and past experience" gives an easy but faithful description about machine learning.

In machine learning, data plays an indispensable role, and the learning algorithm is used to discover and learn knowledge or properties from the data. The quality or quantity of the dataset will affect the learning and prediction performance. We might, for instance, be interested in learning to complete a task, make accurate predictions or behave intelligently. The learning that is being done is always based on some sort of observations or data, such as examples, direct experience, or instruction. So in general, machine learning is about learning to do better in the future based on what was experienced in the past. The emphasis of machine learning is to automate methods. In other words, the goal is to devise learning algorithms that do the learning automatically without human intervention or assistance.

In machine learning, we seek methods by which the computer will come up with its own program based on examples that we provide. Machine learning is a core subarea of artificial intelligence. It is very unlikely that we will be able to build any kind of intelligent system capable of any of the facilities that we associate with intelligence, such as language or vision, without using machine learning. These tasks are otherwise simply too difficult to solve. Further, we would not consider a system to be truly intelligent if it were incapable of learning since learning is the core of intelligence.

1.2 PROBLEM DEFINITION

Microblogging is an extremely prevalent broadcast medium amidst the Internet fraternity these days. People share their opinions and sentiments about variety of subjects like products, news, institutions, etc., every day on microblogging websites. Sentiment analysis plays a key role in prediction systems, opinion mining systems, etc. Twitter, one of the microblogging platforms allows a limit of 140 characters to its users. This restriction stimulates users to be very concise about their opinion and twitter an ocean of sentiments to analyze. Twitter also provides developer friendly streaming API for data retrieval purpose allowing the analyst to search real time tweets from various users.

Lots of effort has been conducted to analyze information of social networks, such as sentiment trend analysis of social network users. Our aim is to analyze the sentimental influence of posts and compare the result on various topics and different social media platforms. Large amounts of posts are generated on social networks every day which makes task more difficult.

1.3 PROJECT PURPOSE

Sentiment analysis deals with identifying and classifying opinions or sentiments expressed in source text. This project contributes to the analysis for classification of twitter sentiment based on machine learning algorithms. We implemented these algorithms for automatic classification of tweets sentiment into either positive, negative or neutral with highest accuracy possible. The tweet can be gathered to summarize the overall sentiment on particular topic. Hence by this way we can know about particular product, movie, person or topic.

1.4 PROJECT FEATURES

Our project was developed to train a machine about sentiment using machine learning algorithm and later use the trained machine to classify the sentiment of tweets. In our consideration classifier is our machine that will classify sentences or tweets based on sentiment. We took a good amount of data to train classifier for better accuracy. It takes huge amount of time to train classifier with good amount of data. We reduced training time by preprocessing the training dataset which removes unwanted words and symbols. The preprocessing generates list of important words called feature vector that affects sentiment.

Reducing the training time was not enough for our project. We also required best accuracy possible. For better accuracy we solved some of the problem such as negation problem. Negation problem is about sentiment of negation sentence having negation words like not, don’t, neither in the sentence. To be more confident on sentiment of tweet or sentence, we used total seven machine learning algorithm to classify sentiment. These seven algorithm were used to give the confidence about the outcome. We called it confidence level. For real time sentiment about particular topic or product, we used twitter streaming API that fetches live tweets from twitter.

1.5 MODULES DESCRIPTION

1.5.1 PROCESS TWEET

Preprocessing is done to remove all unwanted characters, which does not lead us to determine the sentiment of the tweet. Some of the actions performed in preprocessing module are:

Lower cases: We converted the tweets to lowercase.

URL’s: We eliminate the URL’s via regular expressions and replace it with the word URL.

@username: We eliminate “@username” and replace it with the word “AT_USER”.

#hashtag: We eliminate “#tag word” and replace it with the word “tag_word”.

Additional whitespaces: We eliminate all the additional white spaces.

1.5.2 GET FEATUREVECTOR

This function takes processed tweets and gives list of meaning words that has some sentiment called as feature vector. It refines processed tweet with the following functions to generate the feature vector:

Stop words: These words don't indicate any sentiment. Hence we remove it.

Repeating letters: In tweets, sometimes people repeat letters to stress the emotion. Two or more repetitive letters in words are replaced by two of the same letter.

Punctuation: We removed punctuation such as **:;,”, ’„?**

Words must start with an alphabet: We removed all those words which don’t start with an alphabet. The word starting with number or symbol usually don’t have any sentiment.

1.5.3 NEGATE_WORD

Negation identification is not a simple task and its complexity increases, since negation words such as not, nor etc., (syntactic negation) are not the only criterion for negation calculation. The linguistic patterns also introduce the context of negation in textual data. The negation rules are designed in order to improve the sentiment text analysis. The sentiment probability for any frequently used positive word is very high and hence even when used with negation words gives positive sentiment. For example “I am not a good boy” gives positive sentiment as sentiment probability of “not” is nearly equal in positive and negative class. So, this function changes “not good” into single word “not_good”. This single word will occur more only in negative sentence and hence gives “negative” as sentiment outcome.

1.5.4 EXTRACT_FEATURES

Extract features function just marks all the words in feature list as true or false for all class separately. This list is used for probability calculation for each word in the sentence. The words is marked as true for positive class if it is present in any sentence of positive class and false if it is not present. Similarly it is done for all classes.

1.5.5 TWITTER MODULE

The Twitter module connects the application with the worldwide conversation happening on Twitter. We make use of OAuth endpoints to connect users to Twitter and send secure, authorized requests to the Twitter API. OAuth authorization requires four different key (ACCESS_TOKEN, ACCESS_SECRET, CONSUMER_KEY, CONSUMER_SECRET) from a registered application. The response for the request gives tweets on the mentioned topic which is then used to identify the sentiment of the tweet.

CHAPTER 2

LITERATURE SURVEY

2.1 MACHINE LEARNING

Literature survey is the most important step in software development process. Before developing the tool it is necessary to determine the time factor, economy and team strength. Once these things are satisfied, then next steps is to determine which operating system and language can be used for developing the tool. Once the programmers start building the tool the programmers need lot of external support. This support can be obtained from senior programmers, from book or from websites. Before building the system the above consideration are taken into account for developing the proposed system. We have to analyze Machine learning and sentimental analysis:

With the rise of Social Media internet users became able to easily express and share their opinions about companies, products, services, events etc. Thus companies became interested in monitoring what people say about their brands in order to get feedback or enhance their marketing efforts.

Machine Learning has several interesting applications in Social Media Monitoring. It is used in order to evaluate the opinions of the users and classify them as positive, negative or neutral (Also known as Sentiment Analysis). In addition it can be used to detect whether the posts are objective or subjective, what is the natural language of the posts and whether the posts were written by a man or woman.

2.1.1 SENTIMENT ANALYSIS

Sentimental Analysis is the process of computationally identifying and categorizing opinions expressed in a piece of text, especially in order to determine whether the writer's attitude towards a particular topic, product, etc. is positive, negative, or neutral**.** Sentiment analysis is becoming a popular area of research and social media analysis, especially around user reviews and tweets. It is a special case of text mining generally focused on identifying opinion polarity, and while it’s often not very accurate, it can still be useful. For simplicity (and because the training data is easily accessible) we will be focusing on 3 possible sentiment classifications: positive , negative and neutral.

Benefits of Sentimental analysis:

Sentiment analysis on tweets can be an excellent source of information and can provide insights that can:

- Determine marketing strategy
- Improve campaign success
- Improve product messaging
- Improve customer service
- Test business KPIs

2.2 SOFTWARE DESCRIPTION

2.2.1 PYTHON

Python is a widely used high-level, general-purpose, interpreted, dynamic programming language. Python supports multiple programming paradigms, including object-oriented, imperative and functional programming or procedural styles. It features a dynamic type system and automatic memory management and has a large and comprehensive standard library. Python interpreters are available for many operating systems, allowing Python code to run on a wide variety of systems.

Python uses dynamic typing and a mix of reference counting and a cycle-detecting garbage collector for memory management. An important feature of Python is dynamic name resolution (late binding), which binds method and variable names during program execution. Python has a large standard library, commonly cited as one of Python's greatest strengths, providing tools suited to many tasks

The Python Package Index, the official repository of third-party software for Python, contains more than 72,000 packages offering a wide range of functionality, including: graphical user interfaces, web frameworks, multimedia, databases, networking and communications…test frameworks, automation and web scraping, documentation tools, system administration…scientific computing, text processing, image processing.

2.2.2 NLTK

The Natural Language Toolkit, or more commonly NLTK, is a suite of libraries and programs for symbolic and statistical natural language processing (NLP) for the Python programming language. NLTK includes graphical demonstrations and sample data. It is accompanied by a book that explains the underlying concepts behind the language processing tasks supported by the toolkit, plus a cookbook.

NLTK is intended to support research and teaching in NLP or closely related areas, including empirical linguistics, cognitive science, artificial intelligence, information retrieval, and machine learning. NLTK has been used successfully as a teaching tool, as an individual study tool, and as a platform for prototyping and building research systems.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 FUNCTIONAL REQUIREMENTS

In software engineering, a functional requirement defines a function of a software system or its component. A function is described as a set of inputs, the behavior, and outputs. Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish. Behavioral requirements describing all the cases where the system uses the functional requirements are captured in use cases. The following list shows functional requirement of our project:

- We should be able to train classifier with huge amount of data in sufficient time.
- Comparison graph for training time for different algorithm is required.
- Classifier should be able to give sentiment of input given by user or from twitter.
- Fetch live tweets from twitter on mentioned topic and classify their sentiment.
- Twitter will block streaming for frequent connection and streaming. So we should be able to store sufficient tweet for each execution.
- Program should have testing option with accuracy graph as outcome of different algorithm.
- Project must have functionality to store new training data to learn more every time.
- Classifier should be trained only once to work anytime.

3.2 NON-FUNCTIONAL REQUIREMENTS

In systems engineering and requirements engineering, a non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. This should be contrasted with functional requirements that define specific behavior or functions. The plan for implementing functional requirements is detailed in the system design. The plan for implementing non-functional requirements is detailed in the system architecture.

Other terms for non-functional requirements are "constraints", "quality attributes", "quality goals", "quality of service requirements" and "non-behavioral requirements".

Some of the quality attributes are as follows:

- ACCESSIBILITY
- MAINTAINABILITY
- SCALABILITY
- PORTABILITY

3.2.1 ACCESSIBILITY:

Accessibility is a general term used to describe the degree to which a product, device, service, or environment is accessible by as many people as possible.

In our project people who have registered with the apps on twitter and have all four OAuth key should be able to fetch tweets from twitter using internet anytime.

Our program interface is simple and efficient and easy to use.

3.2.2 MAINTAINABILITY:

In software engineering, maintainability is the ease with which a software product can be modified in order to:

- Correct defects
- Meet new requirements

New functionalities can be added in the project based on the user requirements just by adding the appropriate module and functions to existing project using simple Python GUI.

Since the programming is very simple, it is easier to find and correct the defects and to make the changes in the project.

3.2.3 SCALABILITY:

System is capable of handling increase total throughput under an increased load of training data when resources (typically hardware) are added.

System can work normally under situations such as low bandwidth and using less hardware resources.

3.2.4 PORTABILITY:

Portability is one of the key concepts of high-level programming. Portability is the software code base feature to be able to reuse the existing code instead of creating new code when moving software from an environment to another.

Project can be executed under different operation conditions provided it meet its minimum configurations. Only python environment and common libraries would have to be configured in such case.

3.3 HARDWARE REQUIREMENTS

Processor	: Any Processor above 1.5 GHz
RAM	: 2 GB
Hard Disk	: 10 GB
Input device	: Standard Keyboard and Mouse
Output device	: VGA or High Resolution Monitor

3.4 SOFTWARE REQUIREMENTS

- Operating system : Windows XP or any higher version
- IDE : Python 2.7 (64bit)
- Internet : Low Bandwidth is enough

CHAPTER 4

DESIGN

4.1 DESIGN GOALS

To implement an algorithm for automatic classification of tweets into Positive, Negative or Neutral with highest accuracy possible. The tweets cab be gathered to summarize the overall sentiment on a particular topic.

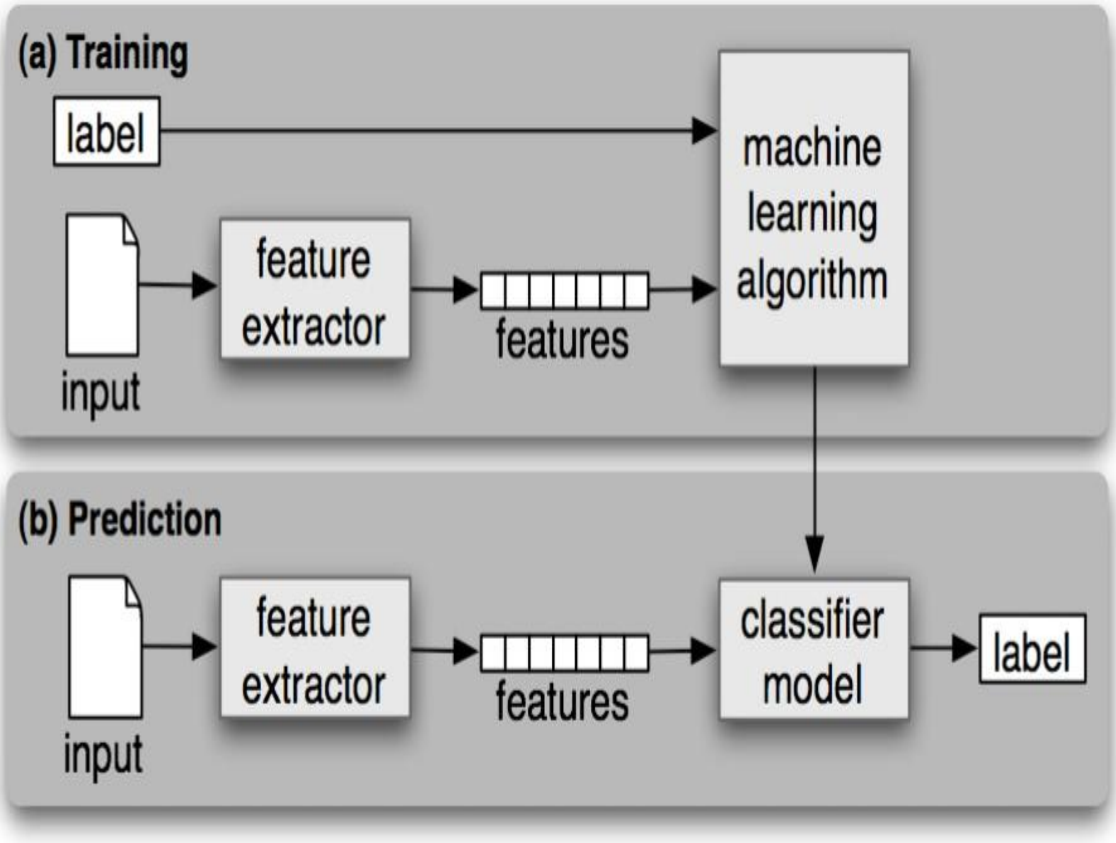


Fig 4.1 System Model

4.1.1 TRAINING DATA

Training data should consist of tweets and its sentiment. In our case we have csv file with two columns. First column is sentiment and second column is tweet.

4.1.2 FEATURES

All the symbols, URL are removed and word starting with number and word without sentiment is removed. The remaining list of word is called features which is list of word that have some sentiment.

4.1.3 MACHINE LEARNING ALGORITHM

The algorithm that is used to train classifier is called machine learning algorithm. It creates trained classifier. In our case we have 7 algorithm with Naive Bayes as important and six other are Multinomial Naive Bayes, Bernoulli Naive Bayes, Logistic Regression, Stochastic Gradient Descent, Support vector clustering, Linear Support vector clustering.

4.1.4 INPUT

The input to program is sentence or tweet.

4.1.5 CLASSIFIER MODEL

We have seven classifier model. One for each machine learning algorithm that is used to classify input and gives output as label.

4.1.6 LABEL

Label is the sentiment output generated by classifier for the given input.

4.2 DATA FLOW DIAGRAM

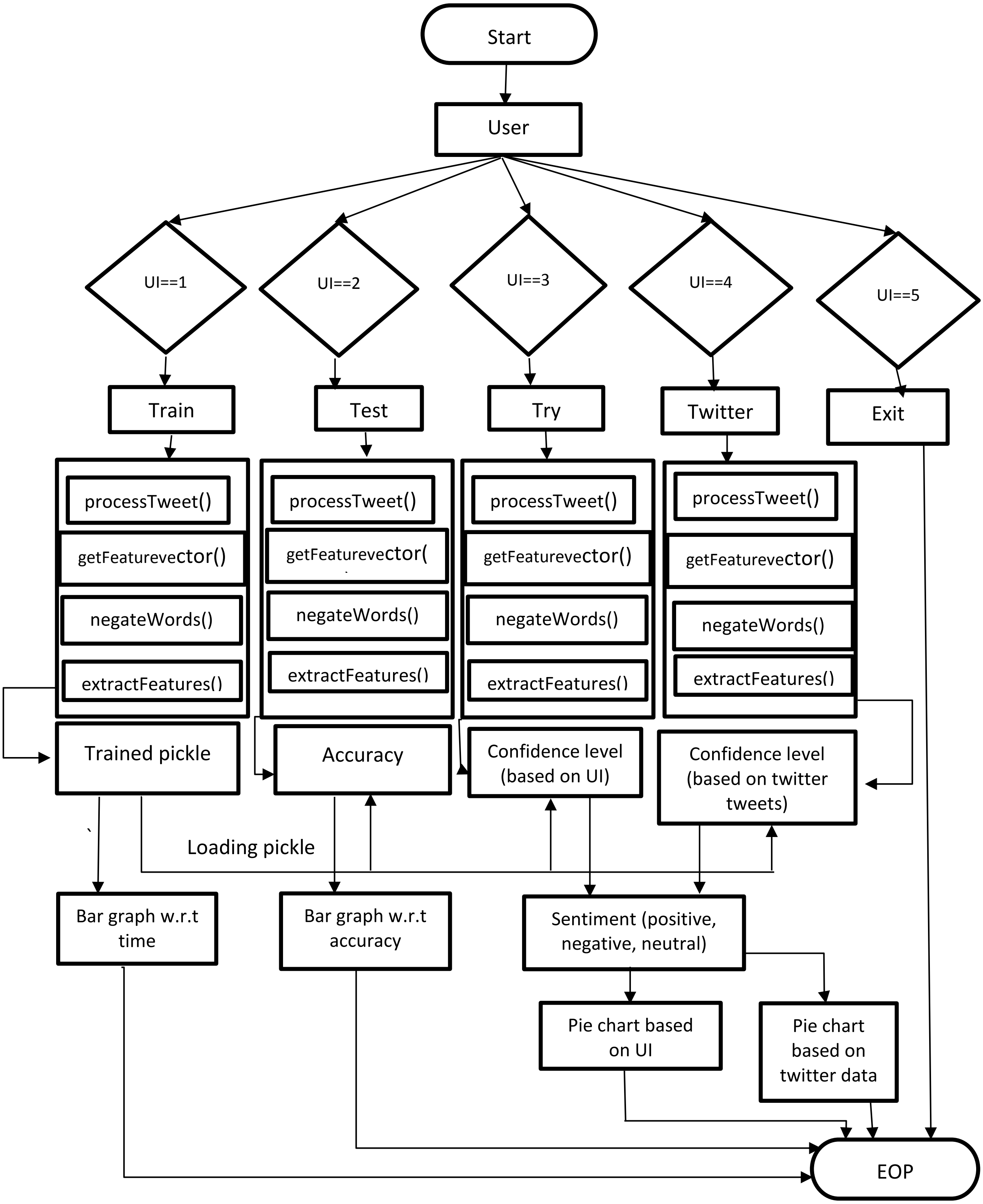


Fig 4.2: DATA FLOW DIAGRAM

4.3 SEQUENCE DIAGRAM FOR TWITTER STREAMING

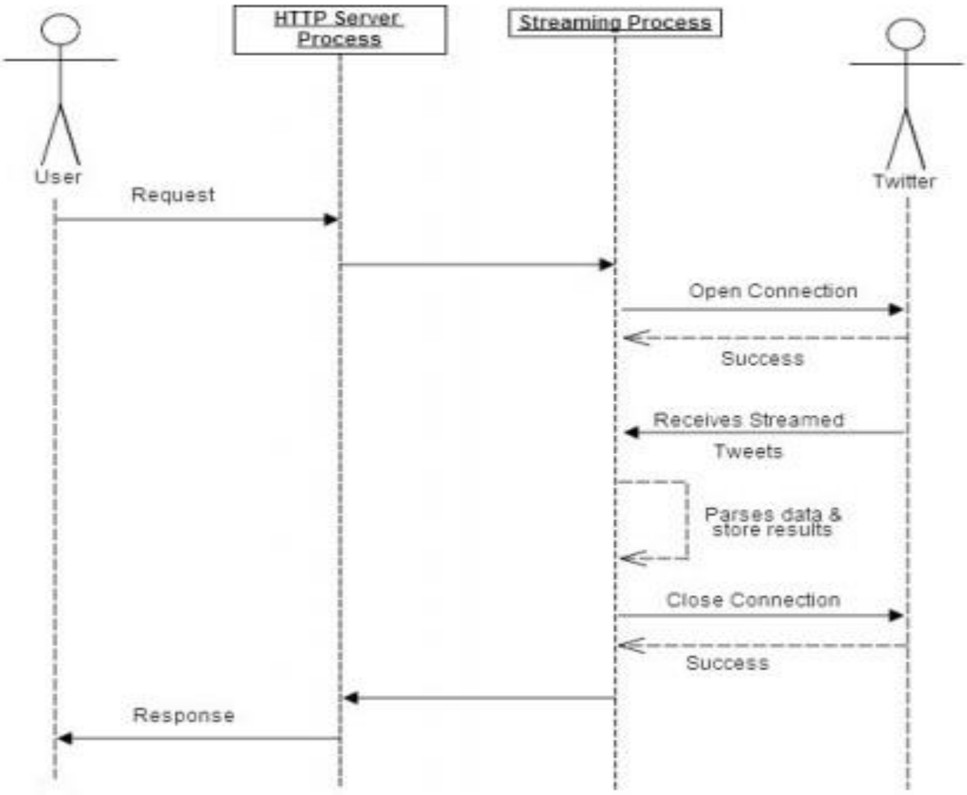


Fig 4.3: SEQUENCE DIAGRAM FOR TWITTER STREAMING

The above diagram shows how to fetch tweet from twitter to use it as input to classifier and calculate sentiment.

CHAPTER 5

IMPLEMENTATION

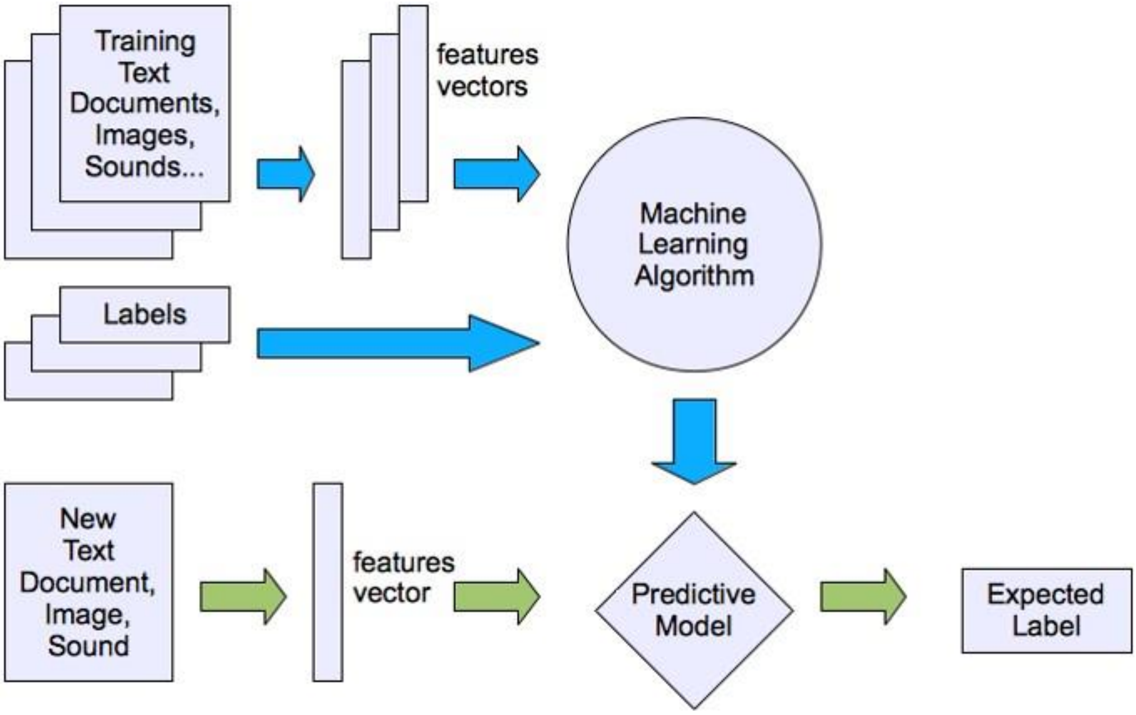
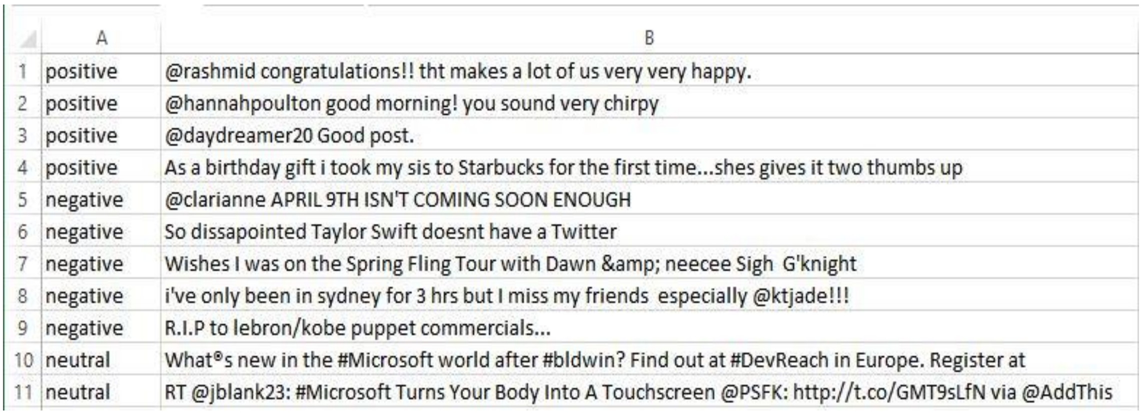

Training Text Documents, Images, Sounds...
features vectors
Machine Learning Algorithm
Labels
New Text Document, Image, Sound
features vector
Predictive Model
Expected Label

Fig 5.1: Implementation Steps of Sentiment Analysis

The figure above shows implementation steps that is required in sentiment analysis problem. The detail explanation about each step in sentiment analysis on twitter data are given below.

5.1 META-DATA GENERATION

Let the user wishes to use program to train classifier with training data for sentiment analysis with three classes namely positive, negative and neutral. Let this file be CSV file which consist of two columns containing sentiment (positive, negative, neutral) in first column (A) and Tweet of max 140 character in second column (B). Each row is training data and we can have huge number of training data.



	A	B
1	positive	@rashmid congratulations!! tht makes a lot of us very very happy.
2	positive	@hannahpoulton good morning! you sound very chirpy
3	positive	@daydreamer20 Good post.
4	positive	As a birthday gift i took my sis to Starbucks for the first time...shes gives it two thumbs up
5	negative	@clarianne APRIL 9TH ISN'T COMING SOON ENOUGH
6	negative	So dissapointed Taylor Swift doesnt have a Twitter
7	negative	Wishes I was on the Spring Fling Tour with Dawn &amp; neecee Sigh G'knight
8	negative	i've only been in sydney for 3 hrs but I miss my friends especially @ktjade!!!
9	negative	R.I.P to lebron/kobe puppet commercials...
10	neutral	What®s new in the #Microsoft world after #bldwin? Find out at #DevReach in Europe. Register at
11	neutral	RT @jblank23: #Microsoft Turns Your Body Into A Touchscreen @PSFK: http://t.co/GMT9sLfN via @AddThis

Fig 5.2: CSV file

We initially read the file using csv reader and create list of dataset name it as Tweets. Each element in list have sentiment and a Tweet.

Each row is then separated into two part. First as sentiment and second as Tweet. Consider first row.

Sentiment = “positive”

Tweet = “@sooraj \\ I can't afffford flight tickets of http://www.jetways.com but I can buy train ticket. #sad”

5.2 CREATION OF PROCESSED TWEETS AND FEATURE LIST

5.2.1 PRE-PROCESSING

1. Convert to lowercase
2. Replace @user with at_user.
3. Replace #Tag with Tag.
4. Replace http://www.url.com with URL.
5. Remove special characters that doesn’t affect sentiment.

Tweet before preprocessing is as follows:

Tweet = “@sooraj \\ I can't afffford flight tickets of http://www.jetways.com but I can buy train ticket. #sad”

Tweet after preprocessing is as follows:

Tweet =“at_user i can't afffford flight tickets of URL but i can buy train ticket. sad”

5.2.2 CREATING FEATURE VECTOR

1. Tokenization
2. Replace two or more with two occurrences
3. Remove the word that doesn’t start with alphabet
4. Remove all stop word which doesn’t have any sentiment

Tweet after preprocessing before creating feature vector is as follows:

Tweet =“at_user i can't afffford flight tickets of URL but i can buy train ticket. sad”

Tweet after creating feature vector is as follows:

FeatureVector = “ can't, afford, but, can, buy., sad ”

5.2.3 CREATE NEGATED FEATURE VECTOR

1. Attach “not_” to words which have negation impact.
2. Remove negation words.
3. Remove the remaining punctuation.

Tweet after creating feature vector is as follows:

FeatureVector = “ Not_afford, can, buy, sad ”

Sentiment = “positive”

5.2.4 CREATE FEATURE LIST

FeatureList = FeatureList + FeatureVector

FeatureList = FeatureList + [Not_afford, can, buy, sad]

5.2.5 APPENDING SENTIMENT AND FEATURE VECTOR

Tweets = Tweets + [“positive, Not_afford, can, buy, sad”]

5.2.6 REPEAT STEP 5.2.1 TO 5.2.5

Step from 5.2.1 till 5.2.5 is repeated for each tweet in training set to get complete list of Processed Tweets and FeatureList.

5.3 USE EXTRACT FEATURES TO GET FEATURES

Extract features function just marks all the words in feature list as true or false for all class separately. This list is used for probability calculation for each word in the sentence.

The words is marked as true for positive class if it is present in any sentence of positive class and false if it is not present. Similarly it is done for all classes. This outcome is called as Features.

5.4 APPLY FEATURES TO GET TRAINING SET

The apply feature function from NLTK (Natural Language Processing Tool) is used to create Training set from Features.

5.5 USE TRAINING SET TO TRAIN CLASSIFIER

The function train() in Naïve Bayes Classifier class of NLTK library takes training set created in 5.2.6 as parameter and returns trained classifier. Similarly other six classifier (Multinomial Naïve Bayes, Bernoulli Naïve Bayes, Logistic Regression, Stochastic Gradient Descent, Support vector clustering, Linear Support vector clustering) are created using train() function from sklearn library of different machine learning classes.

5.6 TRAINED CLASSIFIER IS STORED IN PICKLE

The trained classifier is stored in the system to avoid training classifier every time. Whenever it is required to classify the tweet or sentence then the classifier is loaded from pickle and used to give the output without training once again.

5.7 USER INPUT SENTENCE TO CLASSIFY

Now comes the user part. User enters a sentence and the preprocessing that is 5.2.1 to 5.2.3 is done to create features. The function classify() takes features and does the classification to give sentiment as output. All seven classifier gives they output as positive or negative or neutral. The output may vary of different classifier which is based on methodology used to classify.

5.8 USER INPUT TOPIC TO FETCH TWEET AND CLASSIFY

The user can give topic name as input. Twitter streaming API authenticate using OAuth function using four key as parameter. After authentication twitter stream function is used which takes topic name and language as parameter to create iterator. Iterator gives tweet on mentioned topic on every iteration. This tweet is used as input and classifier gives sentiment as output like in previous case.

5.9 CALCULATING CONFIDENCE

Sentiment output from seven classifier can be used for voting to give which is the best sentiment. The sentiment with the majority vote wins. For example 4 out of seven classifier gives positive as output and other 3 as negative or neutral then confidence is calculated as follows:

No. of positive = 4

No. of negative = 3

No. of neutral = 0

Output = Max (No. of positive, No. of negative, No. of neutral)

Output = 4 (positive)

Confidence Level = (output\Total) * 100 %

Confidence Level = (4\7) * 100 % = 57.14 %

Hence the output will be positive with 57.14 confidence.

5.10 STORING MORE TRAINING DATA

If output given of the classifier is wrong then we can teach them the correct output by storing correct output of the tweet. After every execution it ask whether outcome was correct or not. If user enters as wrong then control ask the correct answer and stores it for next training. After training classifier next time it learns about that tweet and increases its correctness. Hence this is a continuous learning process like human.

CHAPTER 6

TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product it is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

TYPES OF TESTS

6.1 UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

6.2 INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

6.3 VALIDATION TESTING

An engineering validation test (EVT) is performed on first engineering prototypes, to ensure that the basic unit performs to design goals and specifications. It is important in identifying design problems, and solving them as early in the design cycle as possible, is the key to keeping projects on time and within budget. Too often, product design and performance problems are not detected until late in the product development cycle — when the product is ready to be shipped. The old adage holds true: It costs a penny to make a change in engineering, a dime in production and a dollar after a product is in the field.

Verification is a Quality control process that is used to evaluate whether or not a product, service, or system complies with regulations, specifications, or conditions imposed at the start of a development phase. Verification can be in development, scale-up, or production. This is often an internal process.

Validation is a Quality assurance process of establishing evidence that provides a high degree of assurance that a product, service, or system accomplishes its intended requirements. This often involves acceptance of fitness for purpose with end users and other product stakeholders.

The testing process overview is as follows:

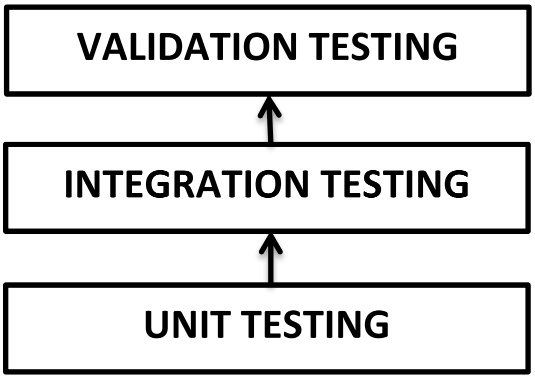


Fig 6.1: The testing process

6.4 SYSTEM TESTING

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black box testing, and as such, should require no knowledge of the inner design of the code or logic.

As a rule, system testing takes, as its input, all of the "integrated" software components that have successfully passed integration testing and also the software system itself integrated with any applicable hardware system(s).

System testing is a more limited type of testing; it seeks to detect defects both within the "inter-assemblages" and also within the system as a whole.

System testing is performed on the entire system in the context of a Functional Requirement Specification(s) (FRS) and/or a System Requirement Specification (SRS).

System testing tests not only the design, but also the behavior and even the believed expectations of the customer. It is also intended to test up to and beyond the bounds defined in the software/hardware requirements specification(s).

6.5 TESTING OF INITIALIZATION AND UI COMPONENTS

Table 6.1: Test case for train option

Serial Number of Test Case	TC 01
Module Under Test	Train
Description	When the train option is selected, it takes size as input and trains the classifiers.
Output	If the training size is valid, it generates bar chart, which shows training time taken for training the classifiers. If size is incorrect, an exception is thrown.
Remarks	Test Successful.

Table 6.2: Test Case for test classifier option

Serial Number of Test Case	TC 02
Module Under Test	Test
Description	An option where user enter the size for testing the classifier and it generates the accuracy score.
Input	Size for testing the classifier
Output	If the size is valid, it generates the accuracy of all the classifiers, along with bar chart for the same. If the size is invalid, an exception is thrown.
Remarks	Test Successful.

Table 6.3: Test Case for try option

Serial Number of Test Case	TC 03
Module Under Test	Try
Description	When user enter the tweet, it generates the corresponding sentiment along with the confidence level of the user tweet.
Input	User sentence(tweet)
Output	The user is shown with the sentiment of the sentence and pie chart for the same is displayed.
Remarks	Test Successful.

Table 6.4: Test Case for tweet option

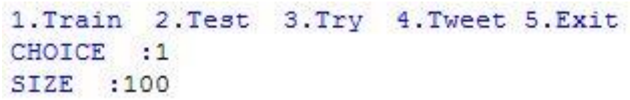
Serial Number of Test Case	TC 04
Module Under Test	Tweet
Description	When the user selects this option, user is asked for tweet topic. This topic is fetched from Twitter and corresponding sentiment is generated
Input	User enters the topic to be fetched from Twitter.
Output	The sentiment of the tweet fetched from Twitter based on user entered topic and pie chart are generated.
Remarks	Test Successful.

Table 6.5: Test Case for store option

Serial Number of Test Case	TC 05
Module Under Test	Store
Description	User is provided with option to correct the sentiment generated, if user feels the generated sentiment is wrong.
Input	Save option, Correct sentiment
Output	Correct sentiment of the tweet is saved
Remarks	Test Successful.

CHAPTER 7

SNAPSHOT



```
1.Train  2.Test  3.Try  4.Tweet 5.Exit
CHOICE  :1
SIZE  :100
```

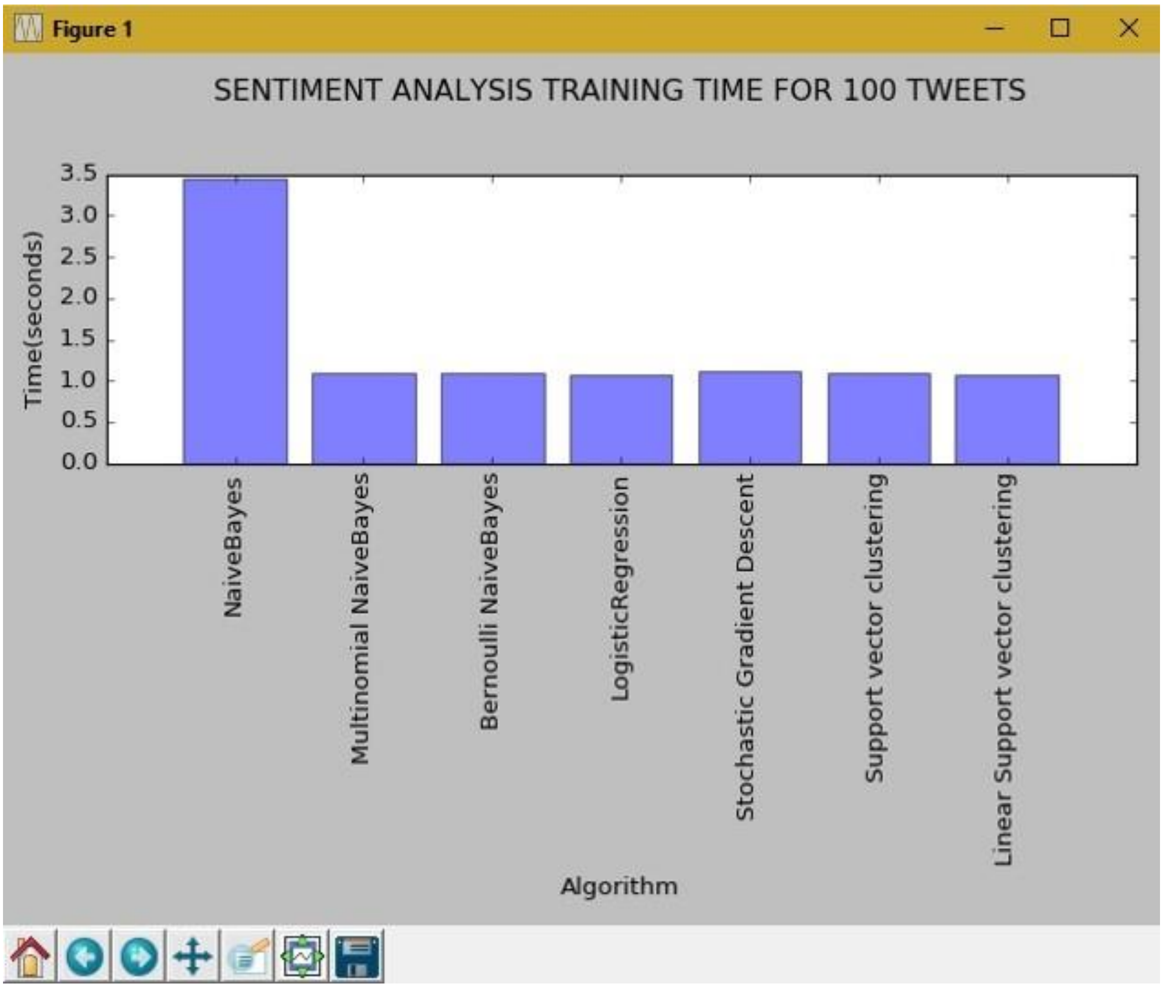
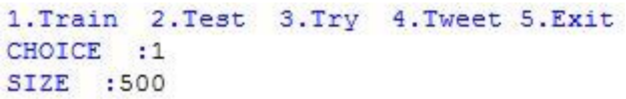


Fig 7.1 Screen Layout of Train with 100 Tweets



```
1.Train  2.Test  3.Try  4.Tweet 5.Exit
CHOICE  :1
SIZE  :500
```

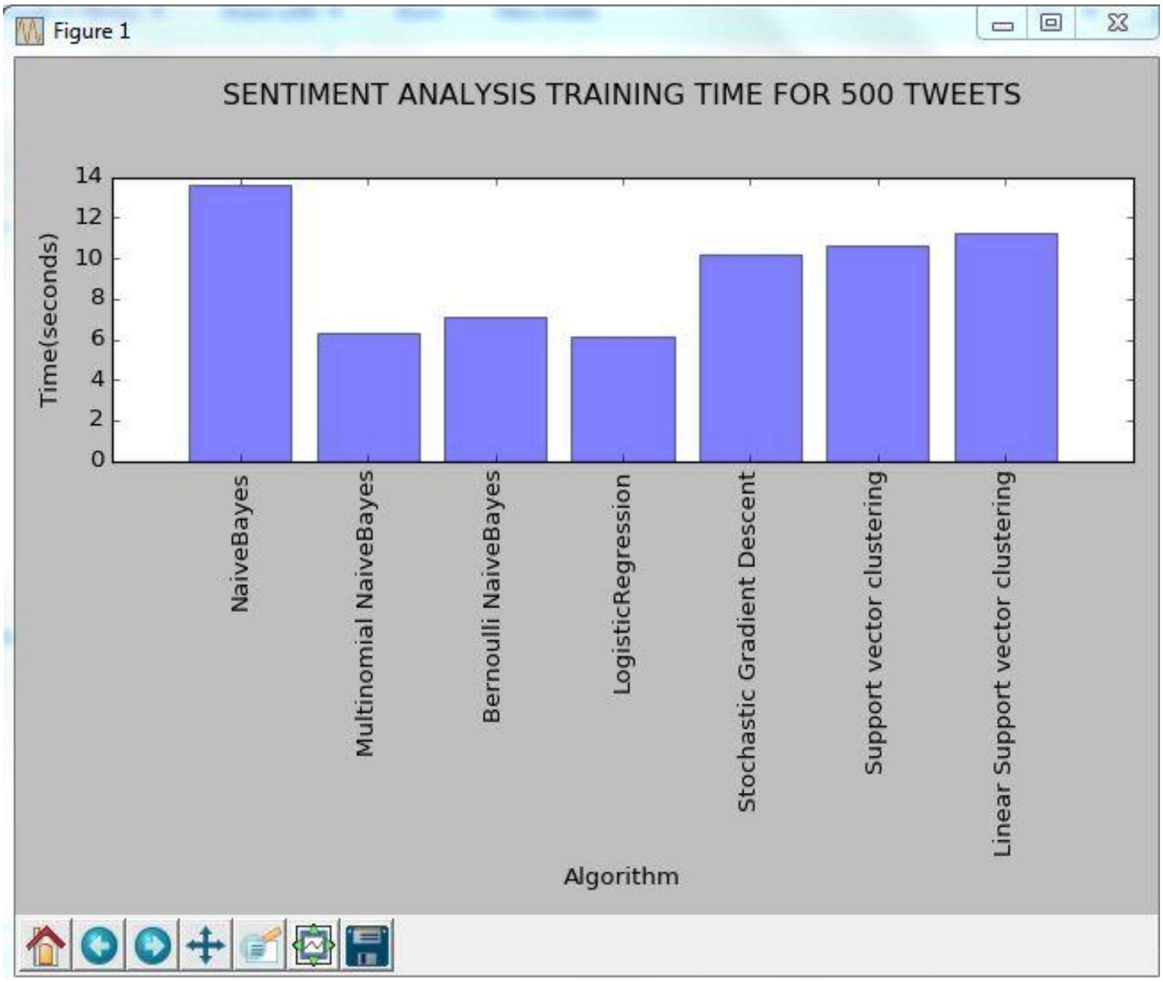
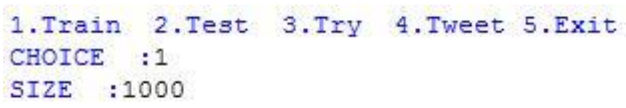


Fig 7.2 Screen Layout of Train with 500 Tweets



```
1.Train  2.Test  3.Try  4.Tweet 5.Exit
CHOICE  :1
SIZE  :1000
```

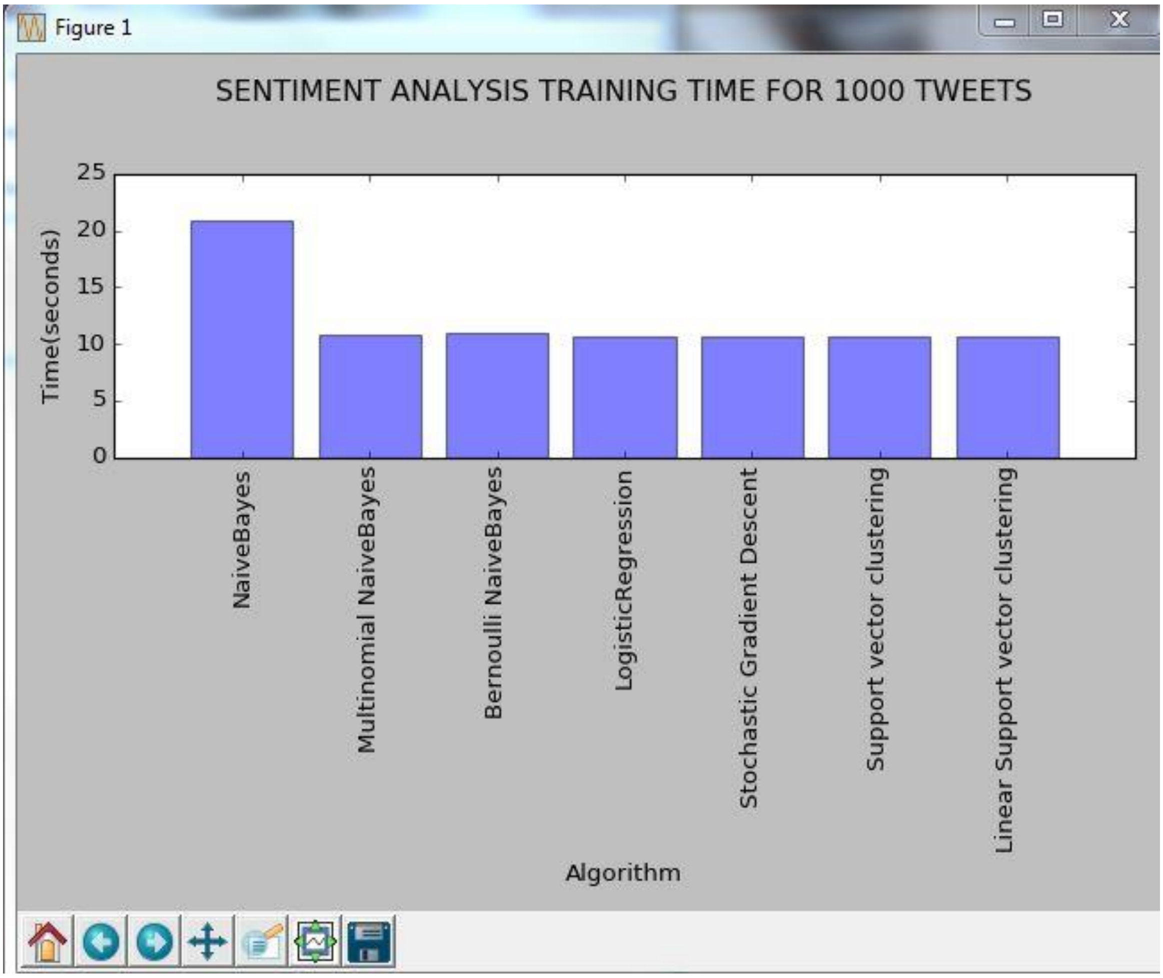
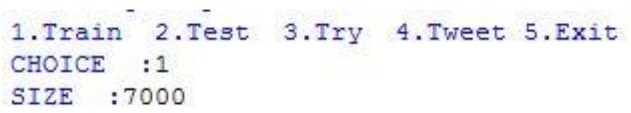


Fig 7.3 Screen Layout of Train with 1000 Tweets



```
1.Train  2.Test  3.Try  4.Tweet 5.Exit
CHOICE  :1
SIZE  :7000
```

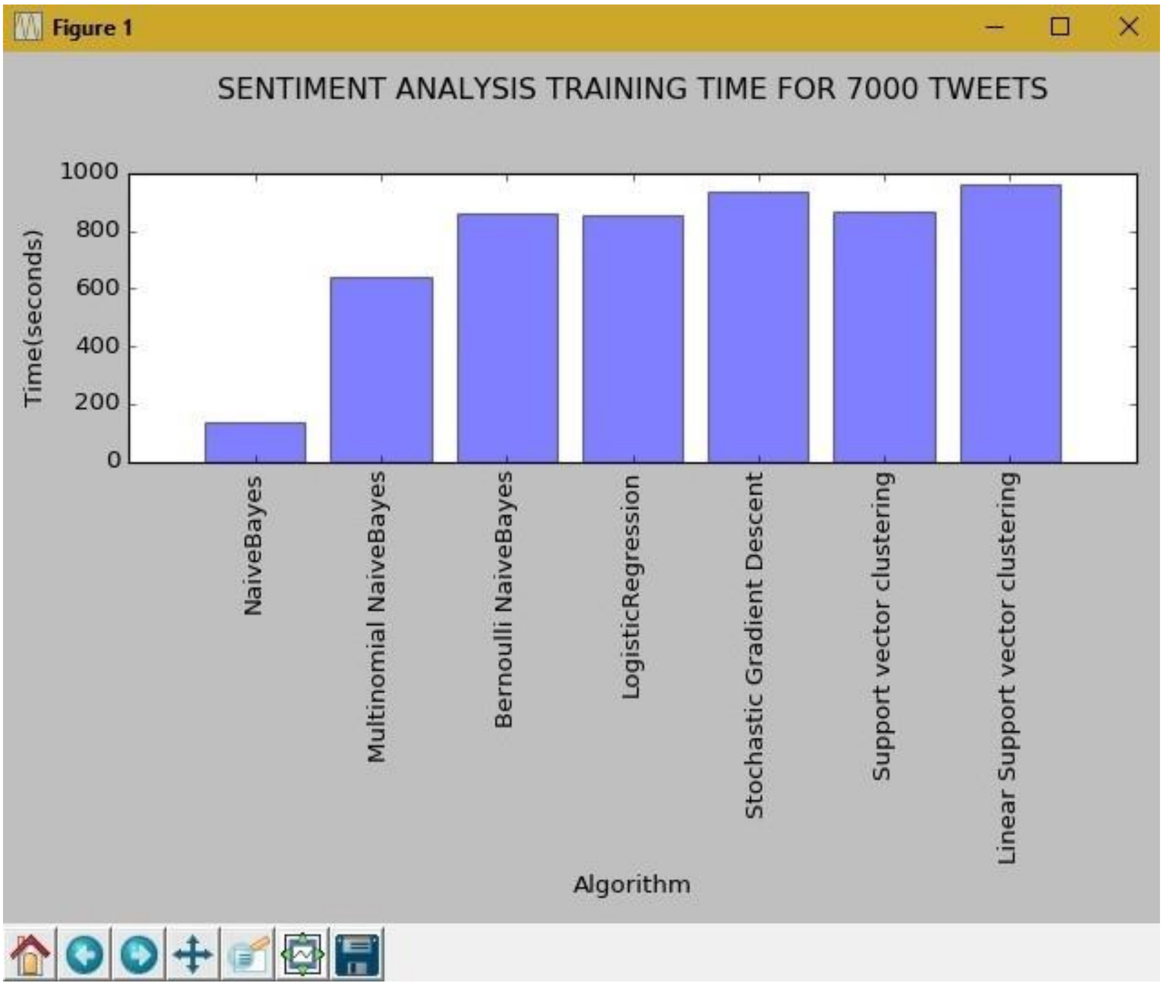


Fig 7.4 Screen Layout of Train with 7000 Tweets

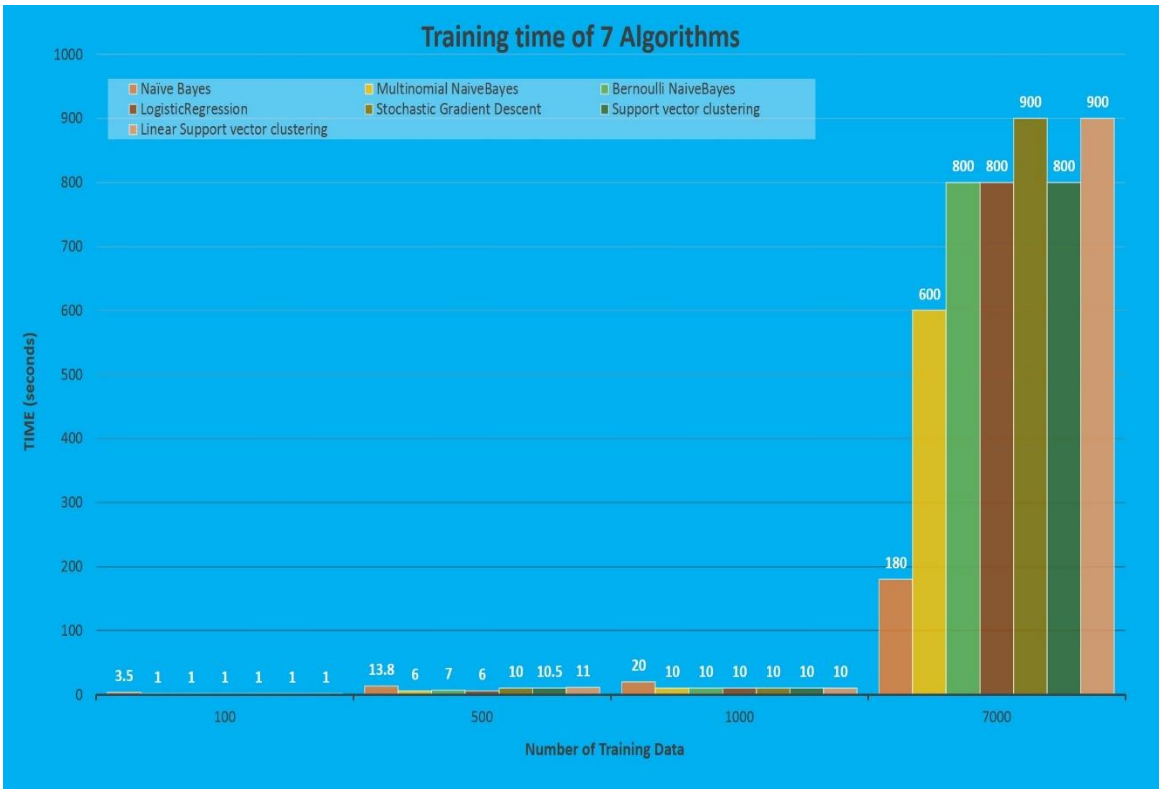
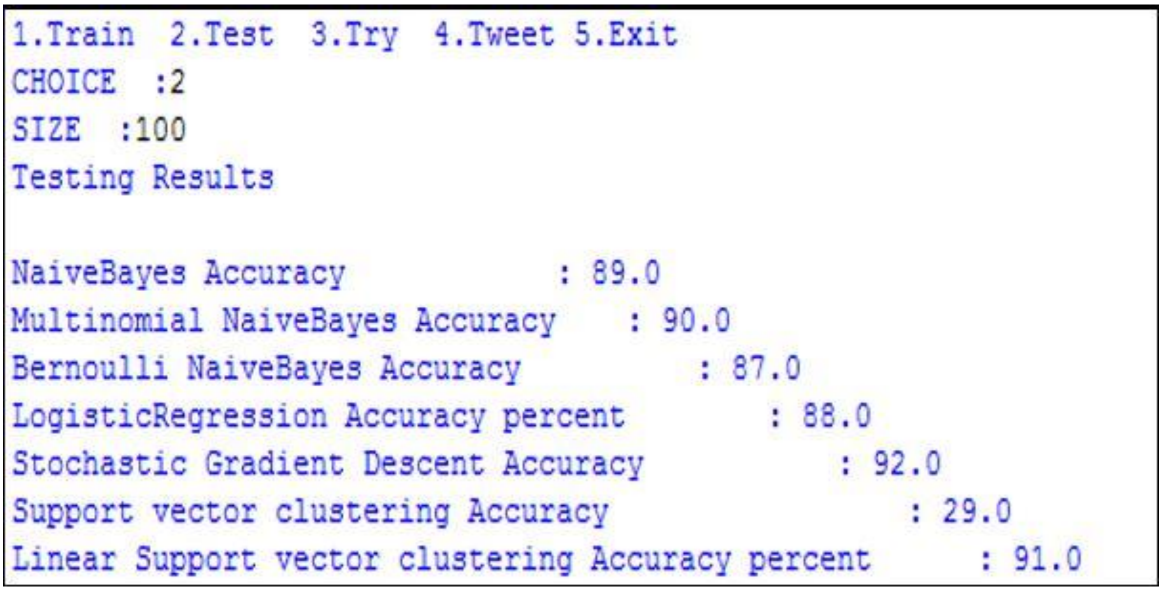


Fig 7.5 Training time comparison of Seven Algorithm



```
1.Train  2.Test  3.Try  4.Tweet 5.Exit
CHOICE  :2
SIZE  :100
Testing Results

NaiveBayes Accuracy                   : 89.0
Multinomial NaiveBayes Accuracy         : 90.0
Bernoulli NaiveBayes Accuracy             : 87.0
LogisticRegression Accuracy percent           : 88.0
Stochastic Gradient Descent Accuracy              : 92.0
Support vector clustering Accuracy                    : 29.0
Linear Support vector clustering Accuracy percent        : 91.0
```

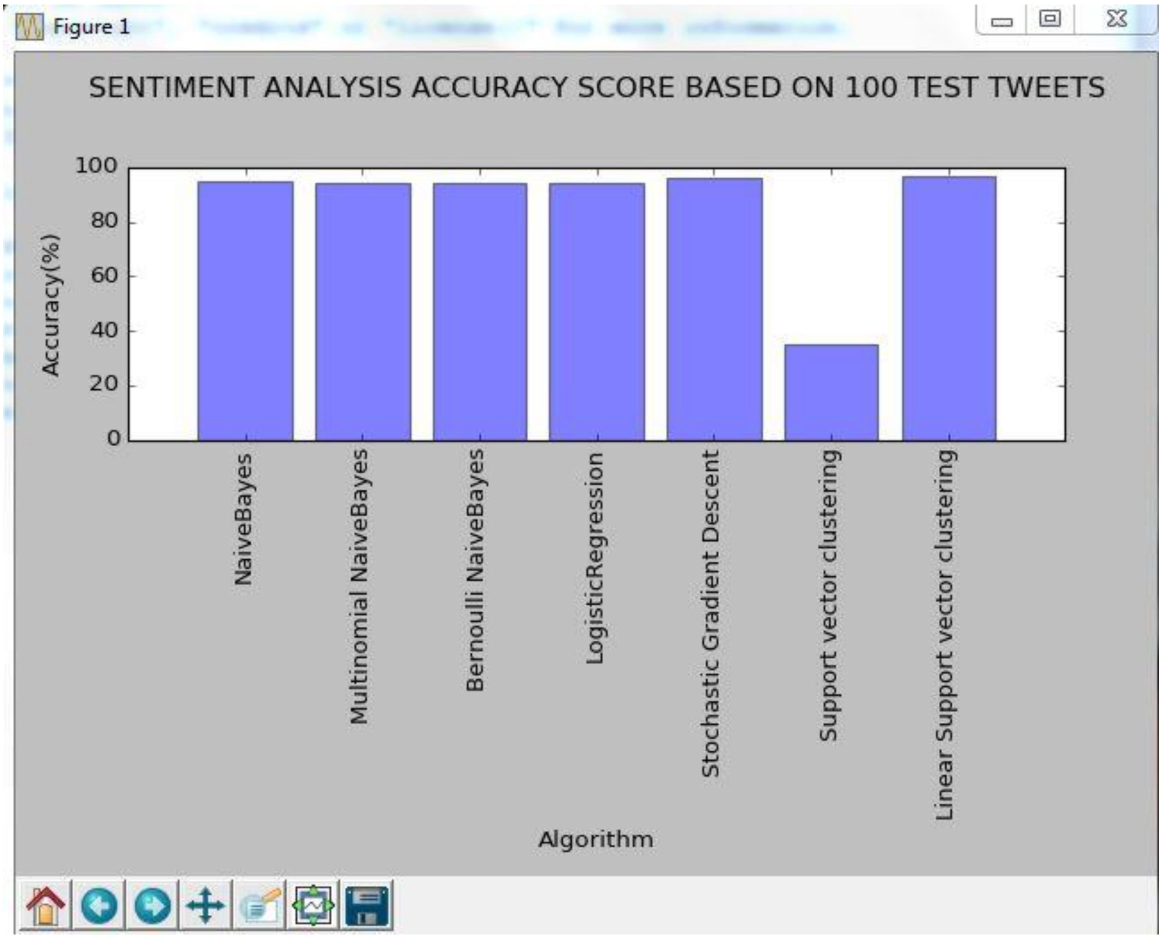
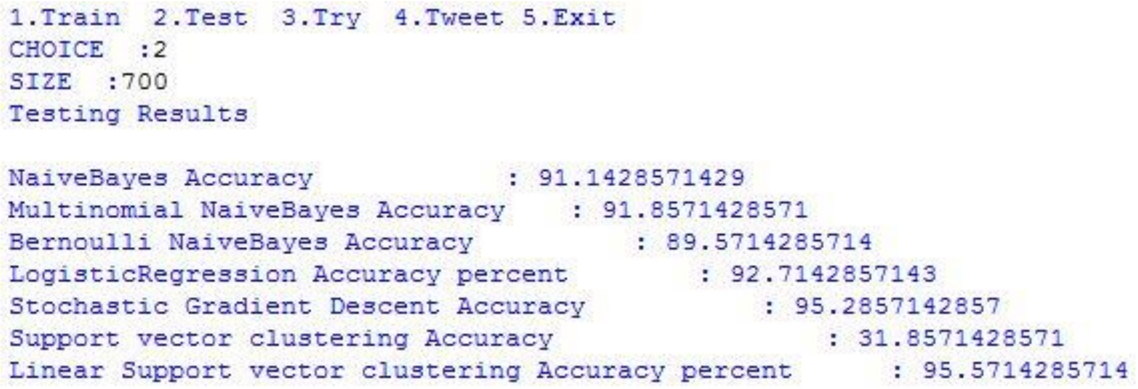


Fig 7.6 Test with 100 Tweets Accuracy Comparison



```
1.Train  2.Test  3.Try  4.Tweet 5.Exit
CHOICE  :2
SIZE  :700
Testing Results

NaiveBayes Accuracy            : 91.1428571429
Multinomial NaiveBayes Accuracy    : 91.8571428571
Bernoulli NaiveBayes Accuracy        : 89.5714285714
LogisticRegression Accuracy percent        : 92.7142857143
Stochastic Gradient Descent Accuracy           : 95.2857142857
Support vector clustering Accuracy                : 31.8571428571
Linear Support vector clustering Accuracy percent      : 95.5714285714
```

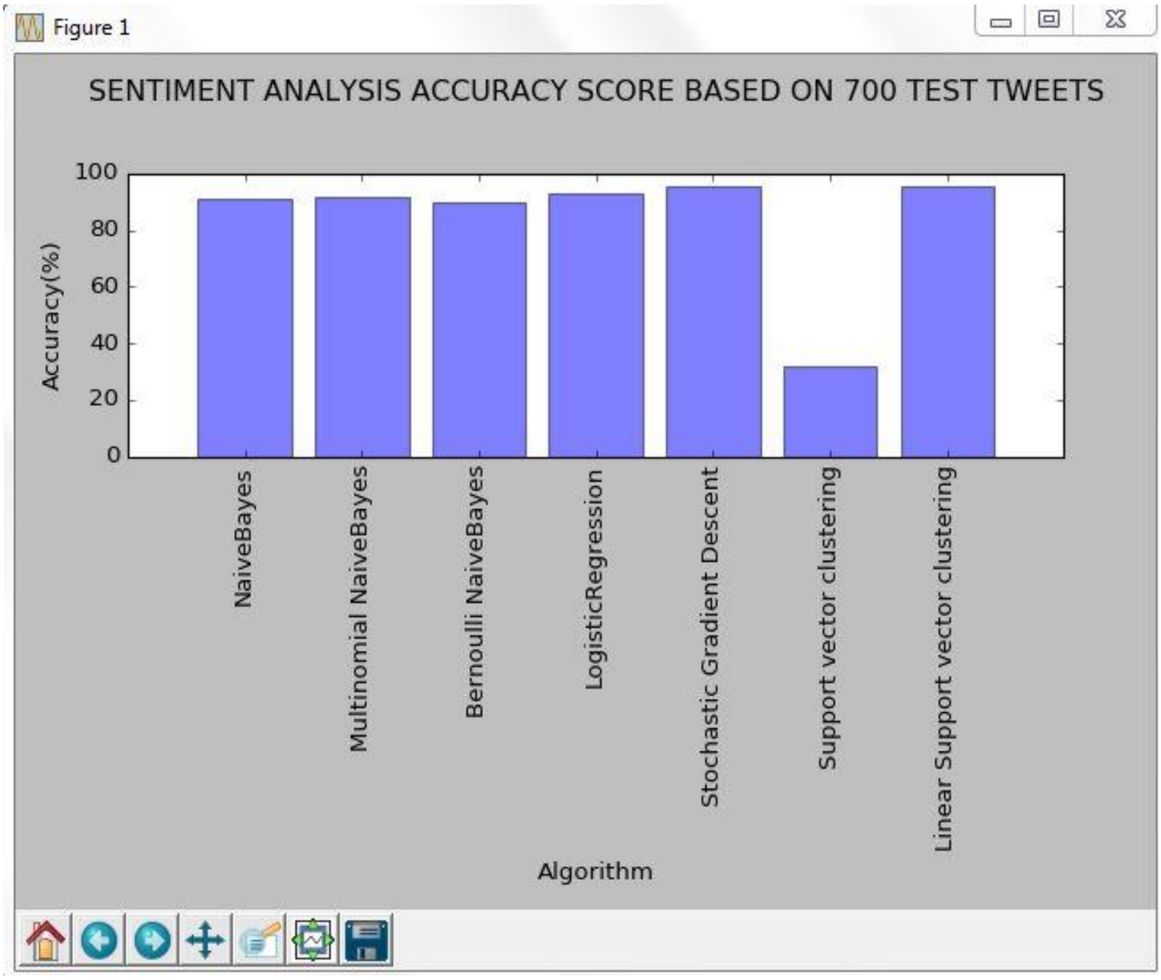
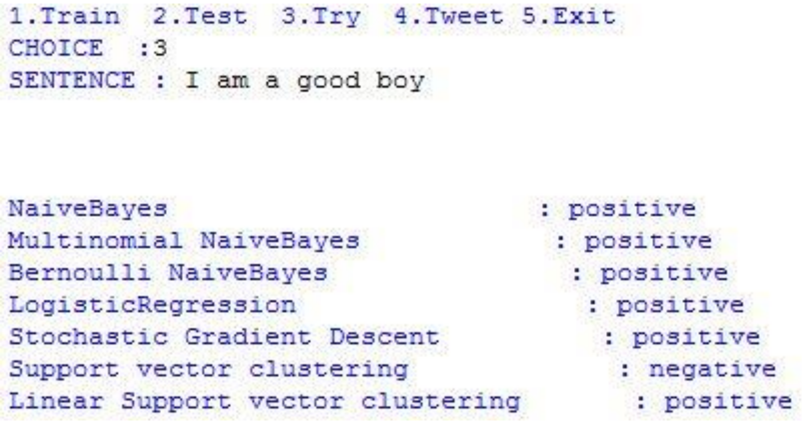


Fig 7.7 Test with 700 Tweets Accuracy Comparison



```
1.Train  2.Test  3.Try  4.Tweet 5.Exit
CHOICE  :3
SENTENCE : I am a good boy


NaiveBayes                          : positive
Multinomial NaiveBayes               : positive
Bernoulli NaiveBayes                  : positive
LogisticRegression                     : positive
Stochastic Gradient Descent             : positive
Support vector clustering                : negative
Linear Support vector clustering          : positive
```

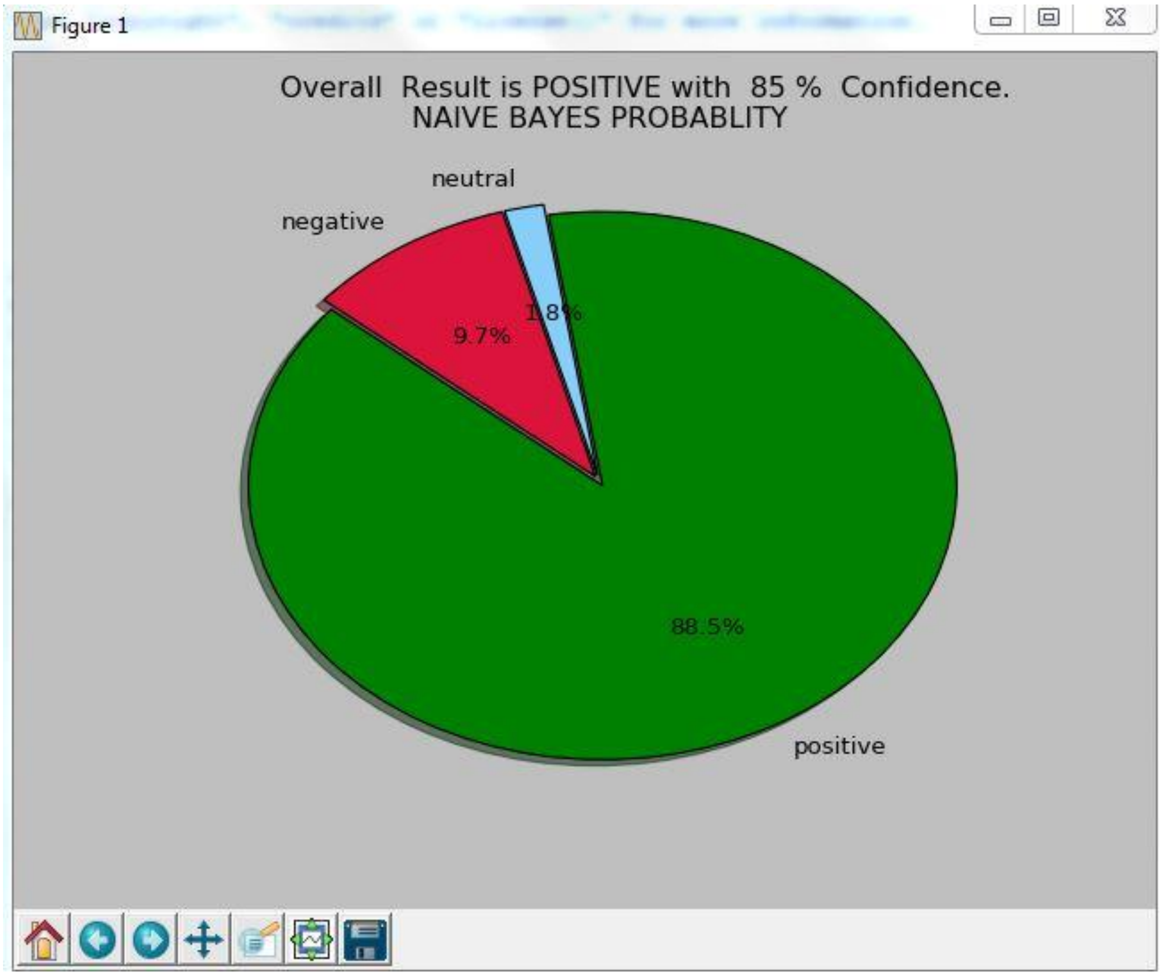
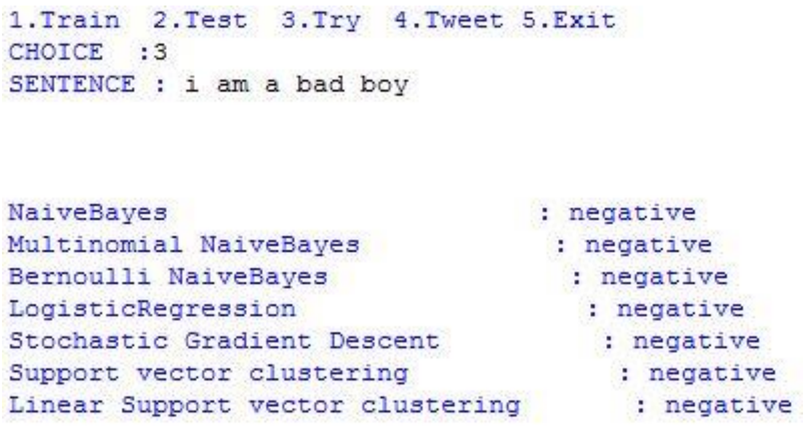


Fig 7.8 Try for positive with confidence and Naïve Bayes probability Pie chart



```
1.Train  2.Test  3.Try  4.Tweet 5.Exit
CHOICE  :3
SENTENCE : i am a bad boy


NaiveBayes                         : negative
Multinomial NaiveBayes              : negative
Bernoulli NaiveBayes                 : negative
LogisticRegression                    : negative
Stochastic Gradient Descent            : negative
Support vector clustering               : negative
Linear Support vector clustering         : negative
```

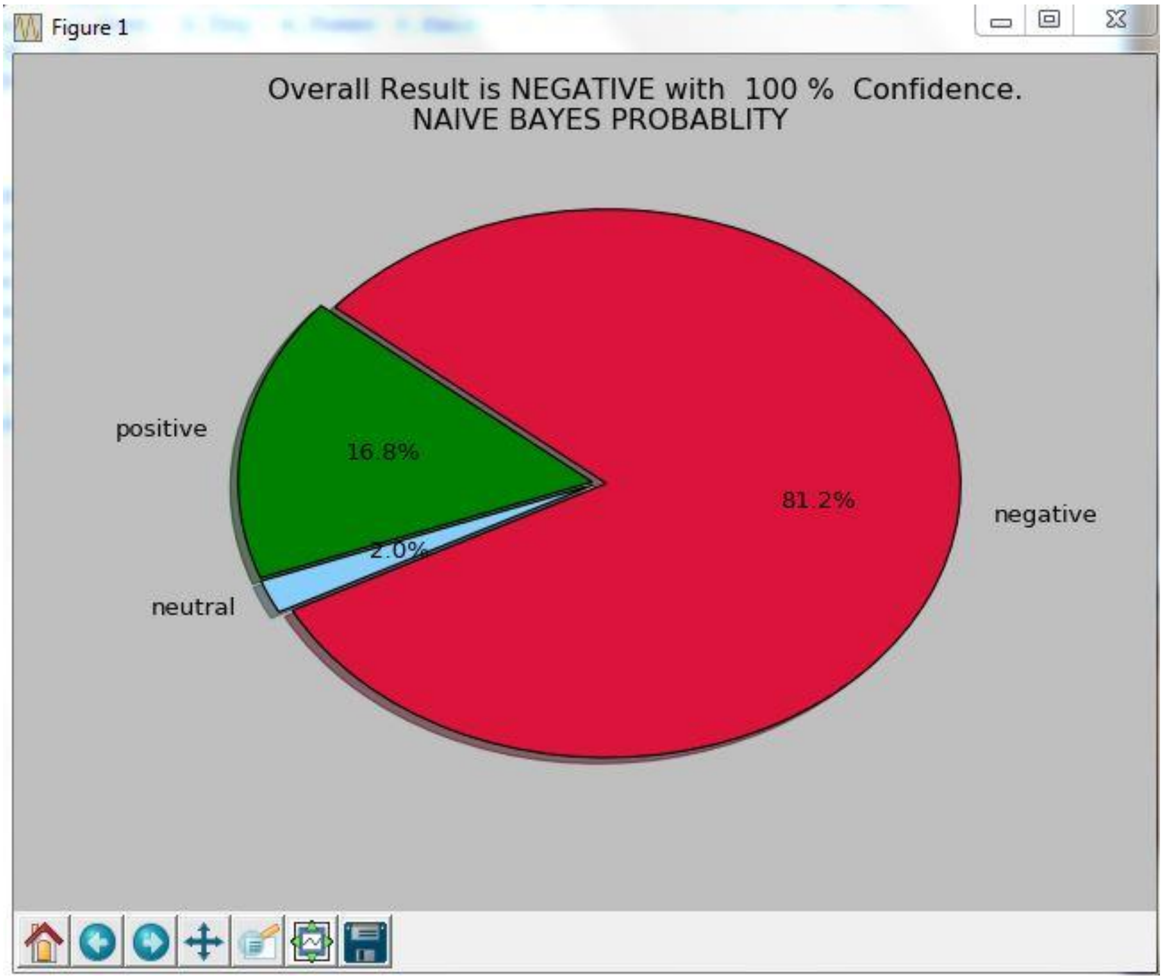
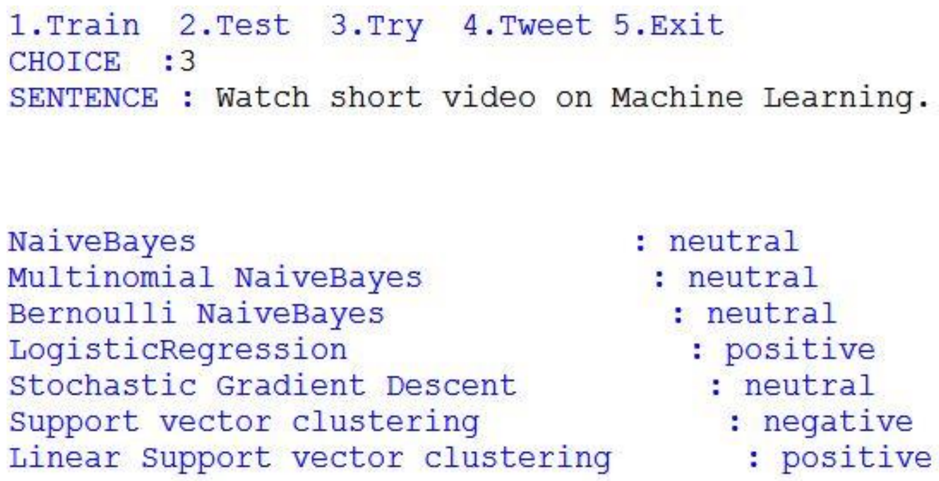


Fig 7.9 Try for Negative with confidence and Naïve Bayes probability Pie chart



```
1.Train  2.Test  3.Try  4.Tweet 5.Exit
CHOICE  :3
SENTENCE : Watch short video on Machine Learning.


NaiveBayes                        : neutral
Multinomial NaiveBayes             : neutral
Bernoulli NaiveBayes                : neutral
LogisticRegression                   : positive
Stochastic Gradient Descent           : neutral
Support vector clustering              : negative
Linear Support vector clustering        : positive
```

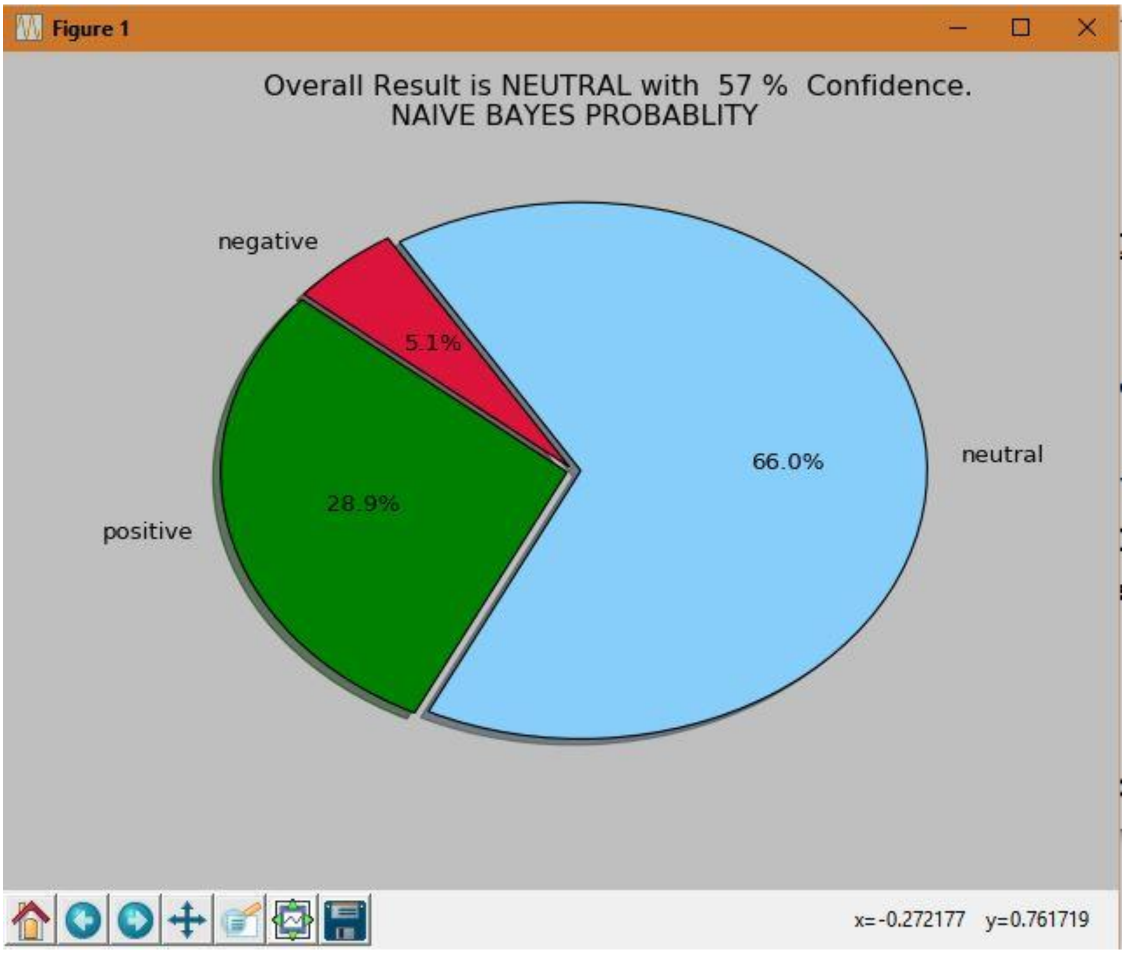
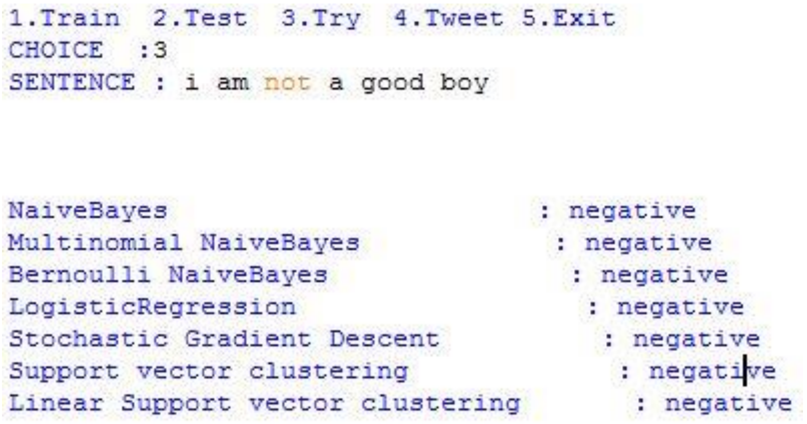


Fig 7.10 Try for Neutral with confidence and Naïve Bayes probability Pie chart



```
1.Train  2.Test  3.Try  4.Tweet 5.Exit
CHOICE  :3
SENTENCE : i am not a good boy


NaiveBayes                              : negative
Multinomial NaiveBayes                   : negative
Bernoulli NaiveBayes                      : negative
LogisticRegression                         : negative
Stochastic Gradient Descent                 : negative
Support vector clustering                    : negative
Linear Support vector clustering              : negative
```

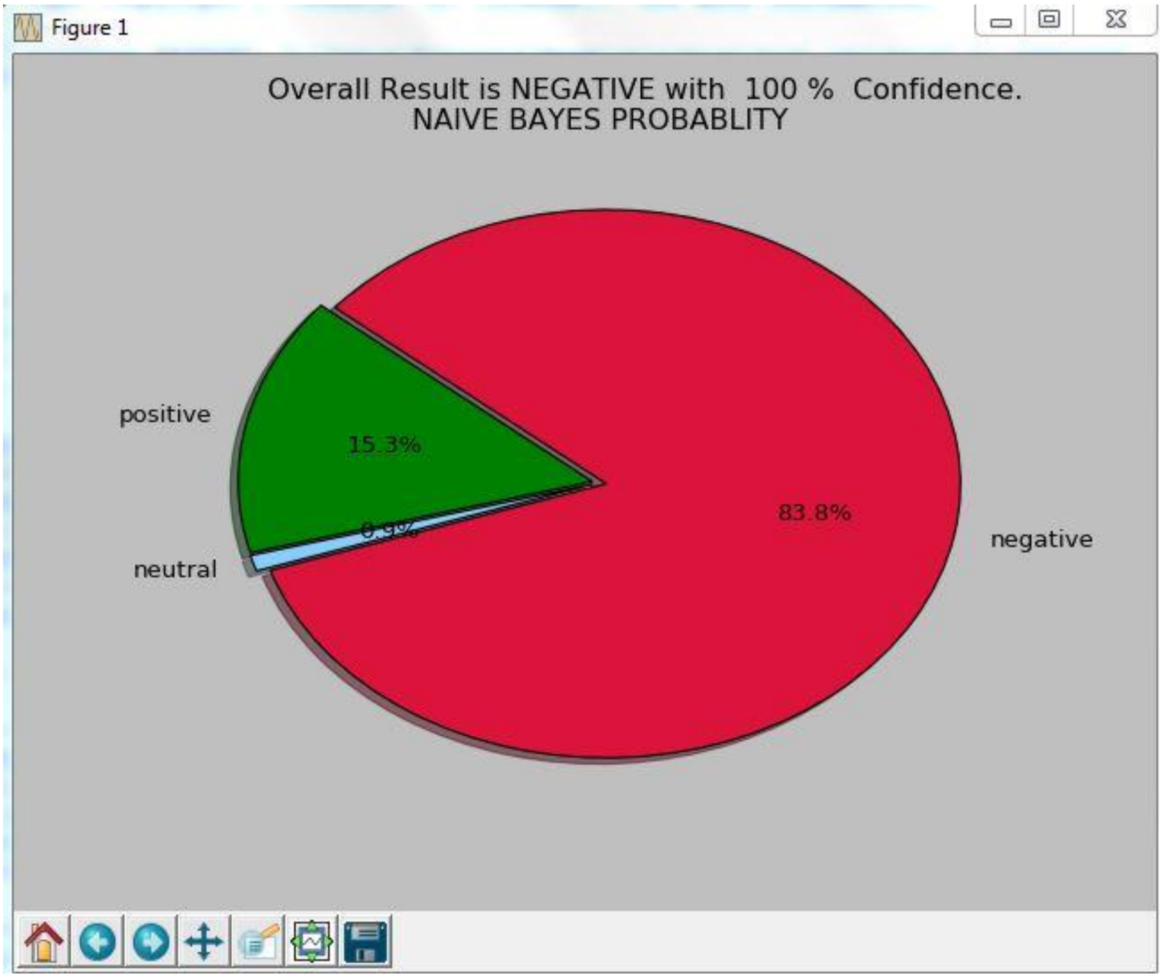
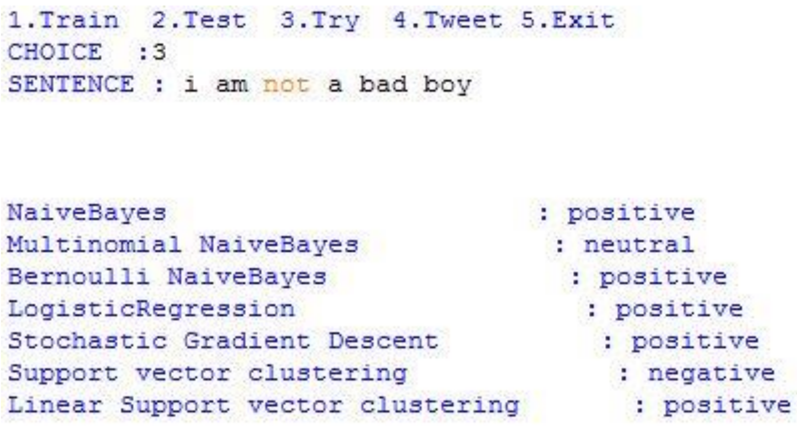


Fig 7.11 Try for not positive with confidence and Naïve Bayes probability Pie chart



```
1.Train  2.Test  3.Try  4.Tweet 5.Exit
CHOICE  :3
SENTENCE : i am not a bad boy


NaiveBayes                          : positive
Multinomial NaiveBayes               : neutral
Bernoulli NaiveBayes                  : positive
LogisticRegression                     : positive
Stochastic Gradient Descent             : positive
Support vector clustering                : negative
Linear Support vector clustering          : positive
```

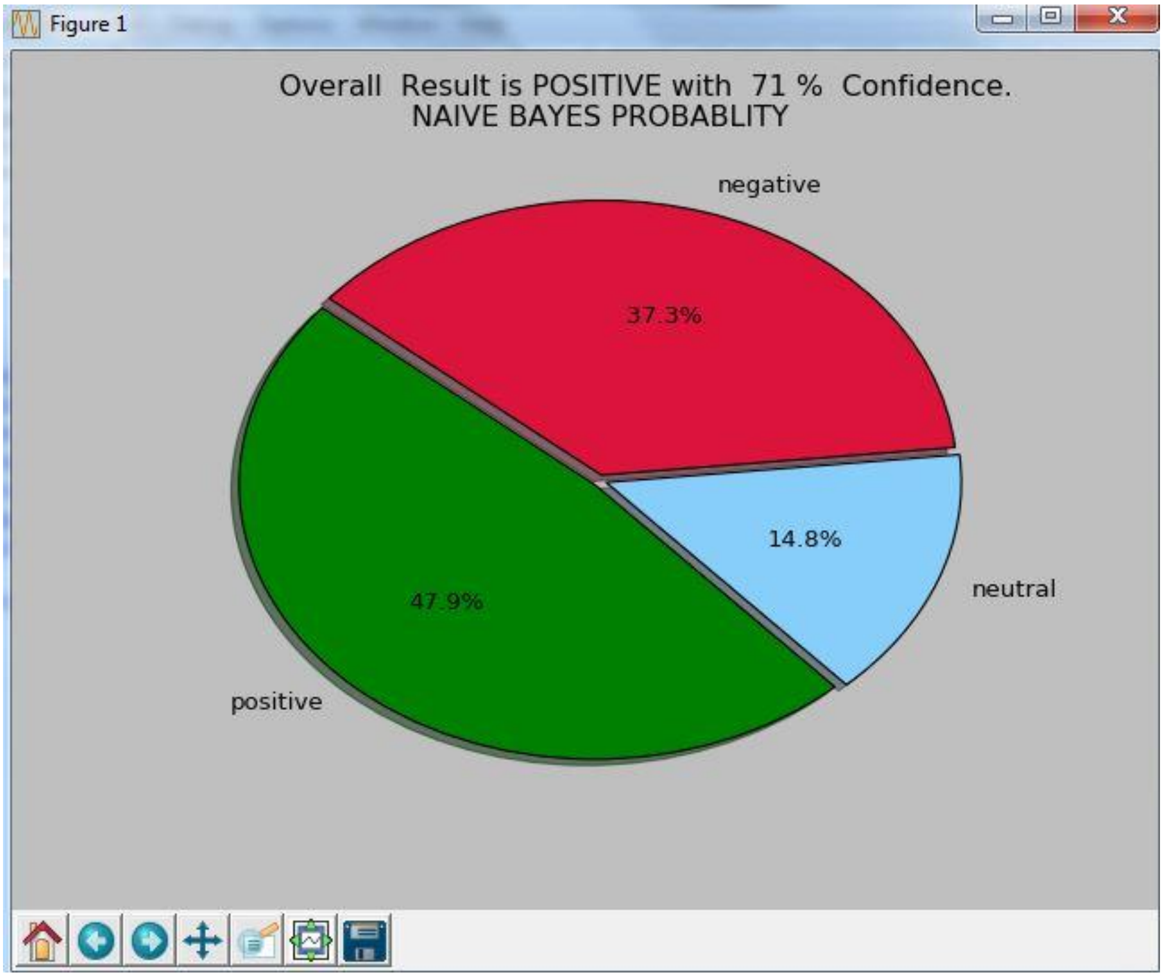
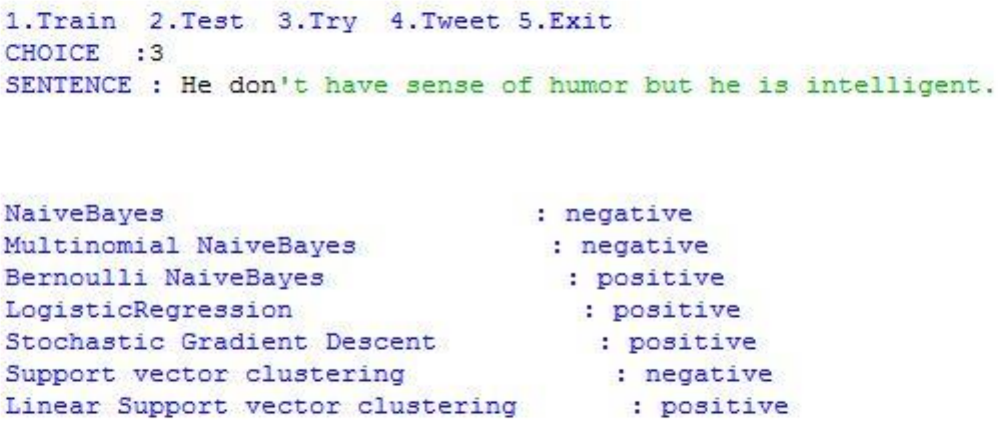


Fig 7.12 Try for not Negative with confidence and Naïve Bayes probability Pie chart



```
1.Train  2.Test  3.Try  4.Tweet 5.Exit
CHOICE  :3
SENTENCE : He don't have sense of humor but he is intelligent.


NaiveBayes                          : negative
Multinomial NaiveBayes               : negative
Bernoulli NaiveBayes                  : positive
LogisticRegression                     : positive
Stochastic Gradient Descent             : positive
Support vector clustering                : negative
Linear Support vector clustering          : positive
```

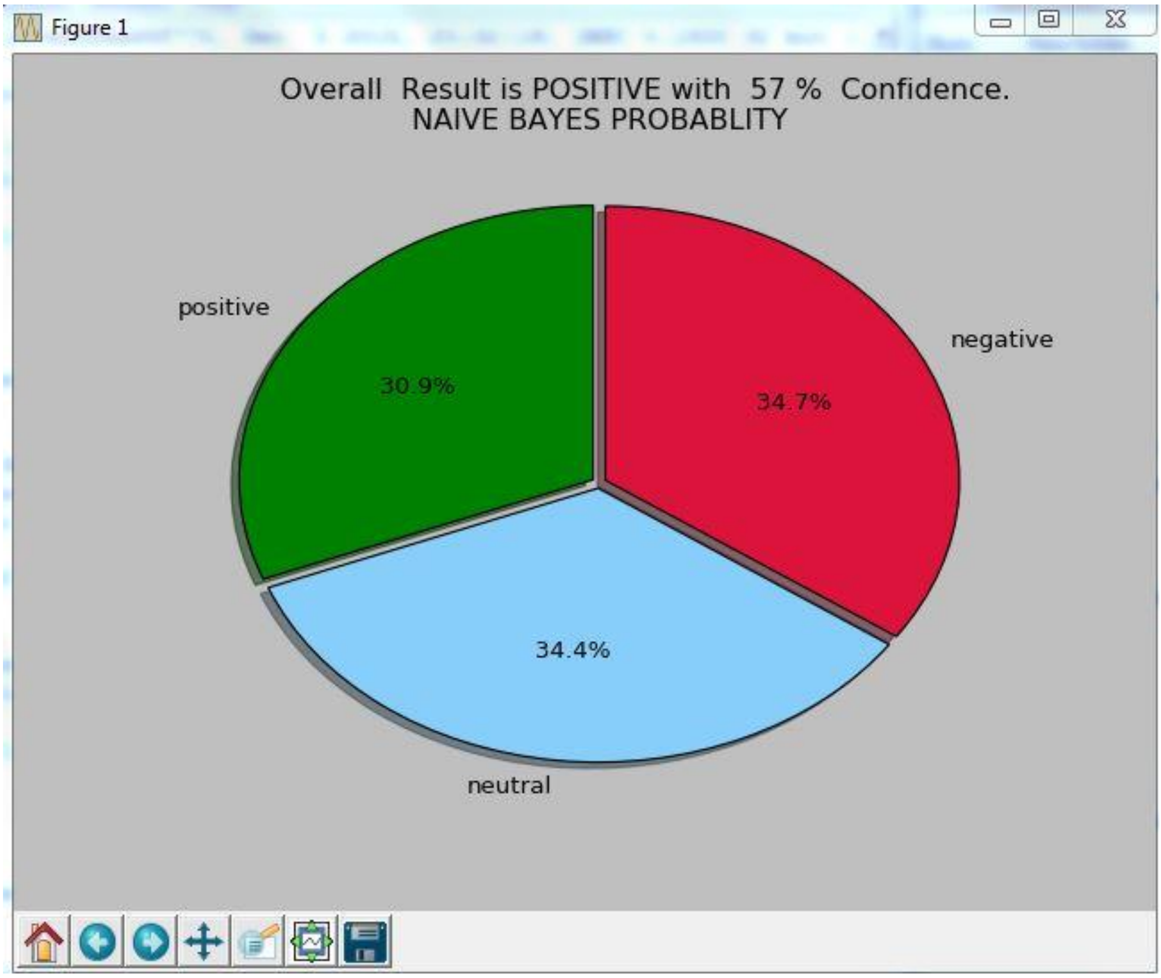
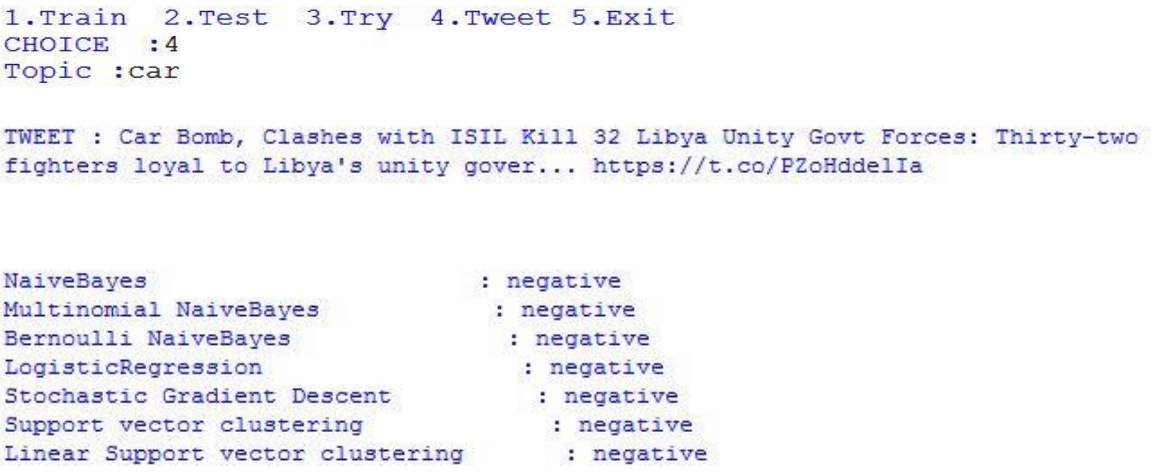


Fig 7.13 Try for not and but with confidence and Naïve Bayes probability Pie chart



```
1.Train  2.Test  3.Try  4.Tweet 5.Exit
CHOICE  :4
Topic :car

TWEET : Car Bomb, Clashes with ISIL Kill 32 Libya Unity Govt Forces: Thirty-two
fighters loyal to Libya's unity gover... https://t.co/PZoHddelIa


NaiveBayes                          : negative
Multinomial NaiveBayes               : negative
Bernoulli NaiveBayes                  : negative
LogisticRegression                     : negative
Stochastic Gradient Descent             : negative
Support vector clustering                : negative
Linear Support vector clustering          : negative
```

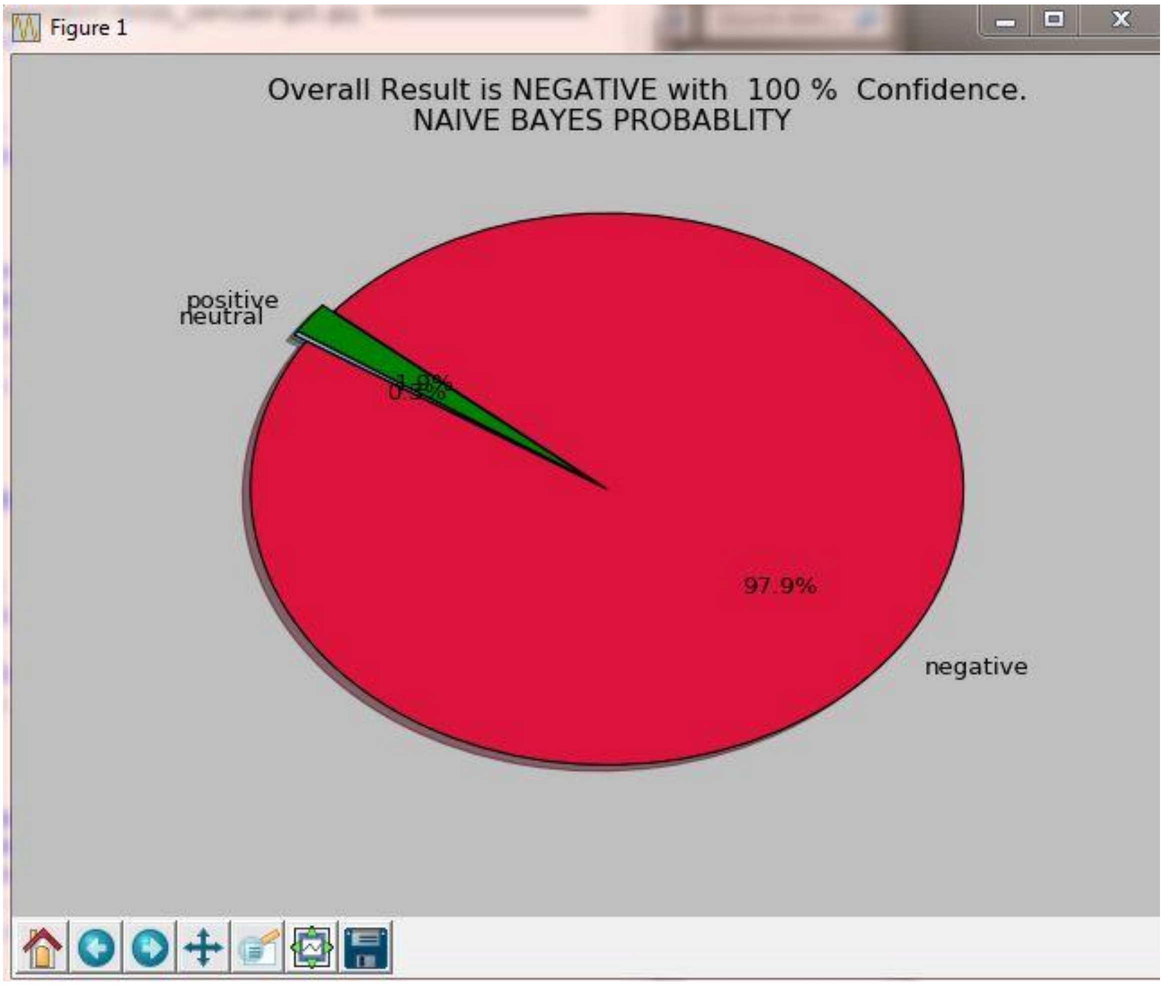
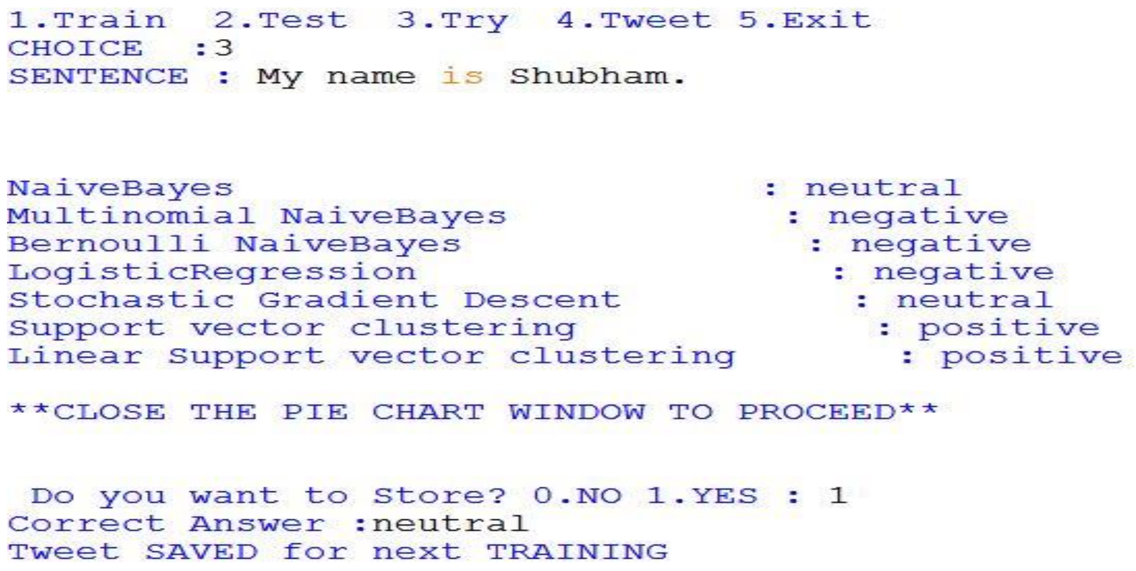


Fig 7.14 Tweet option with confidence and Naïve Bayes probability Pie chart



```
1.Train  2.Test  3.Try  4.Tweet 5.Exit
CHOICE  :3
SENTENCE : My name is Shubham.


NaiveBayes                                : neutral
Multinomial NaiveBayes                     : negative
Bernoulli NaiveBayes                        : negative
LogisticRegression                           : negative
Stochastic Gradient Descent                   : neutral
Support vector clustering                      : positive
Linear Support vector clustering                : positive

**CLOSE THE PIE CHART WINDOW TO PROCEED**


 Do you want to Store? 0.NO 1.YES : 1
Correct Answer :neutral
Tweet SAVED for next TRAINING
```

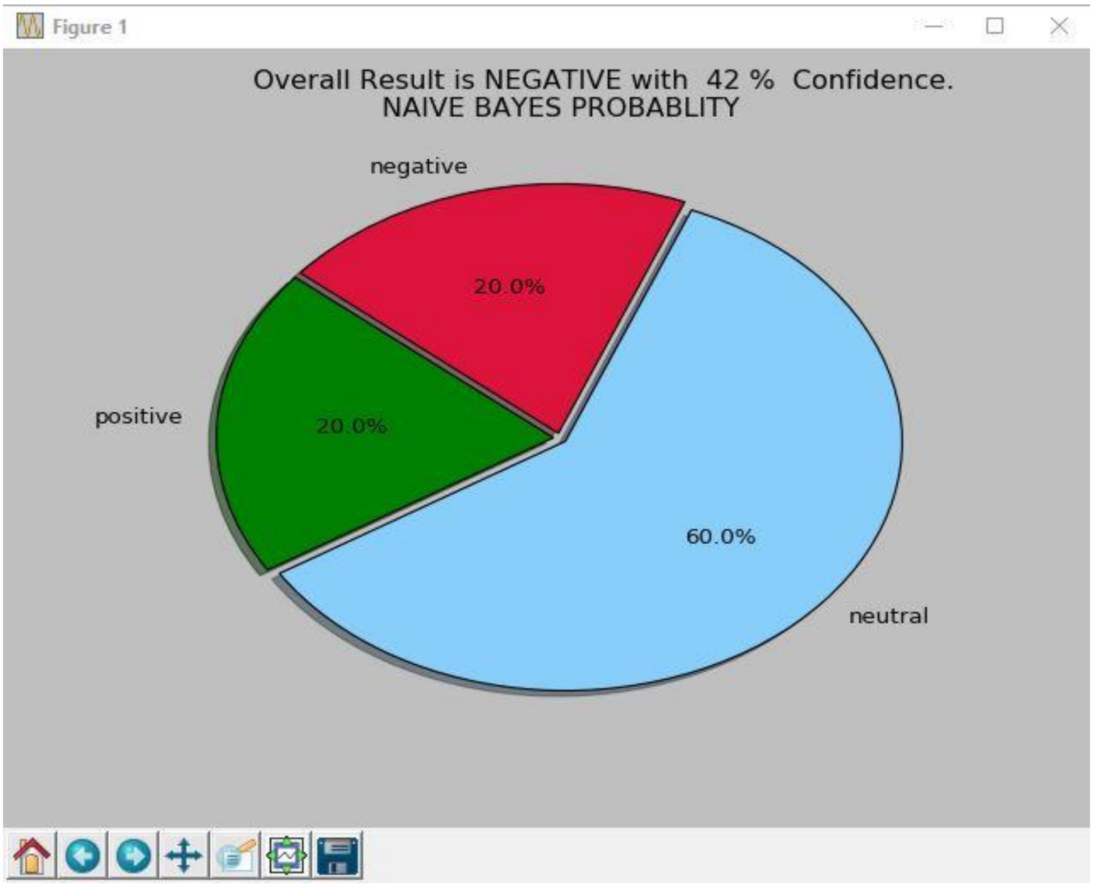


Fig 7.15 Save option for next Training and Naïve Bayes probability Pie chart

CHAPTER 8

CONCLUSION AND FUTURE ENHANCEMENT

8.1 CONCLUSION

The applications for sentiment analysis are endless. It is used in social media monitoring and Voice of Customers (VOC) to track customer reviews, survey responses, competitors, humanoid robots etc. However, it is also practical to use it in business analytics and situations for text analysis.

As we know every coin have two sides, sentiment analysis is great but it’s a difficult task. The difficulty increases with increase in complexity of opinions expressed. In some of the fields like product reviews, face recognition, span filter etc. are relatively easy whereas fields like books, movies, art, music, indirect expressions of opinion are more difficult.

Sentiment analysis is in demand because of its efficiency. Thousands of text documents can be processed for sentiment in seconds, compared to the hours by a team of people to manually complete it. It is so efficient, accurate and fast that many businesses are adopting text and sentiment analysis and incorporating it into their business processes.

The great thing about social media sentiment analysis is that you're not looking for the needle in the hay. Sentiment mining is looking into trends and large numbers of people. It means that you can account for some degree of fuzziness in sentiment classification with the raw amount of data otherwise we will come to know that trends we searching is not popular or important.

8.2 FUTURE ENHANCEMENT

We took existing algorithm and tried to enhance the accuracy with different preprocessing work. We build a confidence level to increase accuracy and solved negation problem. But there are lot more enhancement that we can do with this project. We removed a problem of training every time using pickle which have ability to store trained classifier. Some of the enhancement that we were not able to do due to time constraint are as follows:

- Training with large and better data for best accuracy.
- Enhancing Naïve Bayes using bi-gram, tri-gram, n-gram etc.
- Training with dataset of different lexicon for improving accuracy of varieties of sentences.
- There are some complex machine learning algorithm like neural network that have great accuracy.
- Reducing the execution time by training the different algorithm classifier in parallel way.
- Creating a better user interface.
- Creating an executable file that can be executed without requiring python environment.

BIBLIOGRAPHY

Good Teachers are worth more than thousand books, we have them in Our Department. Specially our guide **Ms. Anjana Sharma**.

REFERENCE PAPERS:

- Twitter Sentiment Analysis: The Good the Bad and the OMG! By Efthymios Kouloumpis, Theresa Wilson, Johanna Moore
- Sentiment Analysis of Twitter Data by Apoorv Agarwal, Boyi Xie, Ilia Vovsha, Owen Rambow and Rebecca Passonneau
- Modeling and Representing Negation in Data-driven Machine Learning-based Sentiment Analysis by Robert Remus
- Sentiment Analysis of Twitter Data Using Machine Learning Approaches and Semantic Analysis by Geetika Gautam
- A Framework for Fast-Feedback Opinion Mining on Twitter Data Streams by Lokmanyathilak Govindan Sankar Selvan and Teng-Sheng Moh
- Sentiment Analysis on Twitter by Akshi Kumar and Teeja Mary Sebastian
- Sentiment Analysis on Twitter through Topic-based Lexicon Expansion by Zhixin Zhou, Xiuzhen Zhang, and Mark Sanderson
- Serendio: Simple and Practical lexicon based approach to Sentiment Analysis by Prabu Palanisamy,Vineet Yadav and Harsha Elchuri

TEXT BOOKS:

1. Learning Python, Fifth Edition by Mark Lutz
2. Python Essential Reference by David Beazley
3. Python for Data Analysis by Wes McKinney
4. Introduction to Python for Econometrics, Statistics and Data Analysis by Kevin Sheppard from University of Oxford
5. A Python 2.7 programming tutorial Version 2.0 by John W. Shipman

SITES REFFERED:

- https://www.python.org/
- http://scikit-learn.org/
- http://www.nltk.org/
- http://www.lfd.uci.edu/~gohlke/pythonlibs/
- https://dev.twitter.com/overview/documentation
- http://matplotlib.org/
- http://nlp.stanford.edu