

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“Jnana Sangama”, Belgaum – 590 018



A project report on

“TEXT BASED FAKE NEWS PREDICTION”

submitted in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

in

INFORMATION SCIENCE & ENGINEERING

by

ASHWINI .M (1CR16IS021)

LABHYA B. M(1CR16IS022)

MONISHA M.L(1CR16IS052)

Under the guidance of

Mrs. Dhanya Viswanath

Assistant Professor

Dept. of ISE, CMRIT, Bengaluru



CMR INSTITUTE OF TECHNOLOGY
DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

#132, AECS Layout, IT Park Road, Bengaluru-560037

2019-20

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“Jnana Sangama”, Belgaum – 590 018



DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

Certificate

This is to certify that the project entitled, “**Text Based Fake News Prediction**”, is a bonafide work carried out by **Ashwini .M (1CR16IS021)**, **B.M Labhya (1CR16IS022)** and **Monisha M.L (1CR16IS052)** in partial fulfillment of the award of the degree of Bachelor of Engineering in Information Science & Engineering of Visvesvaraya Technological University, Belgaum, during the year 2019-20. It is certified that all corrections/suggestions indicated during reviews have been incorporated in the report. The project report has been approved as it satisfies the academic requirements in respect of the project work prescribed for the said Degree.

Name & Signature of Guide
(Mrs. Dhanya Viswanath)

Name & Signature of HOD
(Dr.M.Farida Begam)

Signature of Principal
(Dr.Sanjay Jain)

External Viva

Name of the Examiners

Signature with date

- 1.
- 2.

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“Jnana Sangama”, Belgaum – 590 018



DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

DECLARATION

We, by **Ashwini .M (1CR16IS021)**, **B.M Labhya (1CR16IS022)** and **Monisha M.L (1CR16IS052)** bonafide students of CMR Institute of Technology, Bangalore, hereby declare that the report entitled “**Text Based Fake News Prediction**” has been carried out by us under the guidance of Mrs. Dhanya Viswanath, Assistant Professor CMRIT Bangalore, in partial fulfillment of the requirements for the award of the degree of Bachelor of Engineering in **Information Science Engineering**, of the Visvesvaraya Technological University, Belgaum during the academic year 2018-2019. The work done in this dissertation report is original and it has not been submitted for any other degree in any university.

Ashwini M(1CR16IS021)

B.M Labhya(1CR16IS022)

Monisha ML(1CR16IS052)

ABSTRACT

The large use of web and social media has tremendous impact on our society, culture, business with potentially positive and negative effects. The fake news for various commercial and political purposes has been emerging in large numbers and widely spread in the online world. Fake news is a type of yellow journalism or propaganda that consists of deliberate disinformation or hoaxes spread via traditional news media or online social media. The contents that claim people to believe with the falsification, sometimes with sensitive messages, gets rapidly dispersed to one other. The dissemination of fake news in today's digital world has effected beyond a specific group. Mixing both believable and unbelievable information on social media has made the confusion of truth, that is, the truth will be hardly classified. When the unexpected events happen there are also fake news that are broadcasted that creates confusion due to the nature of the events. Very few people know the real fact of the event while the most people believe the forwarded news from their credible friends or relatives. These are difficult to detect whether to believe or not when they receive the news information. The fake news is also designed to promote certain agenda or biased opinion.

An important goal is to improve the trustworthiness of information available on social platform. We propose to build a classification model that can predict the truthfulness of the news based on its content and its social-context. We investigate and compare different features extraction and machine classification techniques. Our goal is to develop a reliable model that classifies a given article as either fake or true

Keywords: Fake news, Feature Extraction, Machine Learning

Acknowledgment

The satisfaction and euphoria that accompany a successful completion of any task would be incomplete without the mention of people who made it possible, success is the epitome of hard work and perseverance, but steadfast of all is encouraging guidance.

So with gratitude, I acknowledge all those whose guidance and encouragement served as a beacon of light and crowned our effort with success.

I would like to thank **Dr. Farida Begam**, Associate Professor and HOD, Department of Information Science who shared her opinion and experience through which I received the required information crucial for the seminar.

I consider it a privilege and honor to express my sincere gratitude to my guide **Mrs Dhanya Viswanath**, Assistant Professor, Department of Information Science & Engineering, for her valuable guidance throughout the tenure of this review.

I consider it a privilege and honour to express my sincere gratitude to project coordinator **Dr Sudhakar K N**, Associate Professor, Department of Information Science & Engineering for her valuable guidance throughout the tenure of this review.

Finally, I would like to thank all my family members and friends whose encouragement and support was invaluable.

ASHWINI M (1CR16IS021)

B. M LABHYA(1CR16IS022)

MONISHA M.L (1CR16IS052)

TABLE OF CONTENTS

Title	Page No
Abstract.....	i
Acknowledgement.....	ii
List of Figures.....	iii
List of Tables.....	iv
Chapter 1	1-6
PREAMBLE.....	1
1.1 Introduction.....	1
1.2 Existing System.....	4
1.3 proposed System.....	4
1.4 Plan of implementation.....	5
1.5 problem statement.....	6
1.6 Objective of the project.....	6
Chapter 2	7-11
LITERATURE SURVEY.....	7
Chapter 3	12-14
THEORETICAL BACKGROUND.....	12
3.1 Existing Technique.....	12
3.2 Proposed Technique.....	13
Chapter 4	15-18
SOFTWARE REQUIREMENTS SPECIFICATION.....	15
4.1Functional Requirements.....	16
4.2 Non Functional Requirement.....	16
4.2.1 Product Requirement.....	17
4.2.2 Basic Operational Requirements.....	18
4.2.3 Hardware Requirements.....	18

4.2.4 Software Requirements.....	18
Chapter 5	19-22
SYSTEM ANALYSIS.....	19
5.1 Feasibility Study.....	19
5.1.1 Technical Feasibility.....	20
5.1.2 Operational Feasibility.....	20
5.1.3 Economic Feasibility	20
5.1.4 Schedule Feasibility	21
5.2 Analysis.....	21
5.2.1 Performance Analysis	21
5.2.2 Technical Analysis	21
5.2.3 Economical Analysis	22
Chapter 6	23-34
SYSTEM DESIGN.....	23
6.1 System Development Methodology.....	23
6.1.1 Model Phases.....	23
6.1.2 Advantages of the Incremental Model.....	24
6.2 Design Using UML.....	25
6.3 Data Flow Diagram.....	29
6.4 Component Diagram.....	30
6.5 Use Case Diagram.....	30
6.6 Activity Diagram.....	31
6.7 Sequence Diagram.....	33
6.8 Flow Chart.....	34
Chapter 7	35-42
IMPLEMENTATION.....	35
6.1 Naïve Bayes Classifier(NB).....	36
6.2 Natural Language Processing.....	38
6.3 Programming Tools.....	39

6.1.1 Python.....	39
5.2.2 NLTK(Natural Language Tool Kit).....	39
7.2.3 Code.....	40
Chapter 8	43-27
TESTING.....	43
8.1 Testing Methodologies.....	43
8.1.1 White Box Testing	43
8.1.1.1 Advantages of White Box Testing	44
8.1.1.2 Disadvantages of White Box Testing	45
8.1.2 Black Box Testing	45
8.1.2.1 Advantages of Black Box Testing.....	46
8.1.2.2 Disadvantages of Black Box Testing.....	46
8.2 Unit Testing.....	46
8.3 System Testing.....	47
8.4 Quality Assurance.....	47
8.4.1 Quality Factors.....	48
8.5 Functional Test.....	48
Chapter 9	49-52
RESULT AND PERFORMANCE ANALYSIS.....	49
9.1Snapshot of Input.....	49
9.1Snapshots of Output.....	50
Chapter 10	53
CONCLUSION AND FUTURE SCOPE.....	53
10.1 Conclusion.....	53
10.2 Limitation.....	53
10.2.1 Future Scope.....	53

CHAPTER 1

PREAMBLE

1.1 Introduction

The FAKE NEWS IDENTIFICATION is nothing but fake news detection. Fake news is passing wrong information through some media like social media, global news etc. With the development of media, it's getting hard to distinguish whether the news is true or not.

Fake news, one of the biggest new-age problems has the potential to mould opinions and influence decisions. The proliferation of fake news on social media and Internet is deceiving people to an extent which needs to be stopped. The existing systems are inefficient in giving a precise statistical rating for any given news claim. Also, the restrictions on input and category of news make it less varied

Example: During election.

In 2016, the Prime Minister of India, Mr. Narendra Modi declared that most of the cash that people possessed had become worthless and in the span of one month all this old currency had to be deposited in the banks. This led to a chain reaction of a series of fake news being published used mainly for click bait and political gains. News about the new paper bills having a GPS tracker or the daily limit of the amount that can be deposited in banks has increased, were spreading like wildfire. Now, this may not seem like a huge thing but the impact of such articles was so much that there was a point where the Ministry of the Finance had to officially release statements assuring citizens that what they were reading was false information. This is just a small instance of how the spread of false news can impact a much greater audience than it may seem.

The main purpose of Fake news is making money. It is more essential to figure out fake news which delivers wrong information with lack of accuracy. These days' fake news is creating different issues from sarcastic articles to a fabricated news and plan government propaganda in some outlets.

Fake news and lack of trust in the media are growing problems with huge ramifications in our society. The term 'fake news' became common parlance for the issue,

particularly to describe factually incorrect and misleading articles published mostly for the purpose of making money through page views.

In this paper, we are trying to produce a model that can accurately predict the likelihood that a given article is fake news.

Facebook has been at the epicentre of much critique following media attention. They have already implemented a feature to flag fake news on the site when a user sees it; they have also said publicly they are working on to distinguish these articles in an automated way.

A given algorithm must be politically unbiased since fake news exists on both ends of the spectrum – and also give equal balance to legitimate news sources on either end of the spectrum. In addition, the question of legitimacy is a difficult one. However, in order to solve this problem, it is necessary to have an understanding on what Fake News is. Later, it is needed to look into how the techniques in the fields of machine learning, natural language processing help us to detect fake news.

Sentiment is an attitude, thought, or judgment prompted by feeling. Sentiment analysis, which is also known as opinion mining, studies people's sentiments towards certain entities. Internet is a resourceful place with respect to sentiment information. From a user's perspective, people are able to post their own content through various social media, such as forums, micro-blogs, or online social networking sites. From a researcher's perspective, many social media sites release their application programming interfaces (APIs), prompting data collection and analysis by researchers and developers. For instance, Twitter currently has three different versions of APIs available, namely the REST API, the Search API, and the Streaming API. With the REST API, developers are able to gather status data and user information; the Search API allows developers to query specific Twitter content, whereas the Streaming API is able to collect Twitter content in real time. Moreover, developers can mix those APIs to create their own applications. Hence, sentiment analysis seems having a strong fundament with the support of massive online data.

However, those types of online data have several flaws that potentially hinder the process of sentiment analysis. The first flaw is that since people can freely post their own content, the quality of their opinions cannot be guaranteed. For example, instead of sharing topic related opinions, online spammers post spam on forums. Some spam are meaningless at

all, while others have irrelevant opinions also known as fake opinions. The second flaw is that ground truth of such online data is not always available. A ground truth is more like a tag of a certain opinion, indicating whether the opinion is positive, negative, or neutral. The Stanford Sentiment 140 Tweet Corpus is one of the datasets that has ground truth and is also public available. The corpus contains 1.6 million machine- tagged Twitter messages.

Microblogging websites have evolved to become a source of varied kind of information. This is due to nature of micro blogs on which people post real time messages about their opinions on a variety of topics, discuss current issues, complain, and express positive sentiment for products they use in daily life. In fact, companies manufacturing such products have started to poll these microblogs to get a sense of general sentiment for their product. Many time these companies study user reactions and reply to users on microblogs. One challenge is to build technology to detect and summarize an overall sentiment.

Our project Tweezer resembles the analyse of tweets by the peoples on certain products of companies or brands or performed by political leaders. In order to do this we analysed tweets from Twitter. Tweets are a reliable source of information mainly because people tweet about anything and everything they do including buying new products and reviewing them. Besides, all tweets contain hash tags which make identifying relevant tweets a simple task. A number of research works has already been done on twitter data. Most of which mainly demonstrates how useful this information is to predict various outcomes. Our current research deals with outcome prediction and explores localized outcomes.

As the pre-processing phase was done in certain extent it was possible to guarantee that analysing these filtered tweets will give reliable results. Twitter does not provide the gender as a query parameter so it is not possible to obtain the gender of a user from his or her tweets. It turned out that twitter does not ask for user gender while opening an account so that information is seemingly unavailable.

1.2 EXSISTING SYSTEM

If a certain message contains these words or number of words these are addressed to as spam. These rules have also been used in social media platform with a success. Although the main drawback is that the process of developing new words are easy and constant and the use of shortened words are becoming more common on platform for example lol which means laugh out loud. Pattern matching techniques are being used to detect these shortened words on these platforms. For instance, from any account a tweet is published regarding trending information on the social media platform or a new account not more than a day old starts advertising about the trending topics is regarded as fake.

1.3 PROPOSED SYSTEM

A. Dataset Description

One of the most difficult issues to unravel in machine learning has nothing to do with complex calculations: it's the issue of getting the correct datasets in the correct organization. Getting the correct information implies assembling or distinguishing the information that relates with the results which needs to be foreseen; for example information that contains a flag about occasions which needs to be taken care about. The datasets should be lined up with the issue which is being attempted to explain. In the event that the correct information is not present, at that point the endeavours to assemble an AI arrangement must come back to the dataset gathering stage.

B. Pre-processing

Pre-preparing alludes to the changes connected to the information before nourishing it to the calculation. Data pre-processing is a method that is utilized to change over the crude information into a perfect informational index. At the end of the day, at whatever point the information is assembled from various sources it is gathered in a crude organization which isn't doable for the examination. Pre-processing is essential for accomplishing better outcomes from the applied model in Machine Learning project the configuration of the information must be in a legitimate way.

Another perspective is that the dataset ought to be arranged so that more than one Machine Learning and Deep Learning calculations are executed in one informational index, and best

out of them is picked. Before representing the data using various evaluating models, the data needs to be subjected to certain refinements. This will help us reduce the size of the actual data by removing the irrelevant information that exists in the data.

C. Train and Test Splitting

To make a valuable training set, the issue needs to be comprehended for which it is being settled for. For example what will the machine learning calculation do and what sort of yield is anticipated. Machine learning regularly works with two informational collections: training and test. Each of the two ought to arbitrarily test a bigger assortment of information. The principal set which is being used is the training set, the biggest of the two.

Running a training set through a machine learning system shows the net how to weigh diverse highlights, changing them coefficients as per their probability of limiting blunders in the outcomes. Those coefficients, otherwise called parameters, will be contained in tensors and together they are known as the model, since it encodes a model of the information on which it is being trained

1.4 Plan of Implementation

The problem at hand consists of two subtasks:

- Phrase Level Sentiment Analysis in Twitter
- Given a message containing a marked instance of a word or a phrase, determine whether that instance is positive, negative or neutral in that context.
- Sentence Level sentiment Analysis in Twitter

Given a message, decide whether the message is of positive, negative, or neutral sentiment. For messages conveying both a positive and negative sentiment, whichever is the stronger sentiment should be chosen.

1.5 Problem Statement

Problem	The issue of “fake news” has arisen recently as a potential threat to high-quality journalism and well-informed public discourse. People intentionally spread these counterfeit articles with the help of web-based social networking sites.
Effects	The fundamental objective of fake news sites is to influence the popular belief on specific issues.
Impact	Difficulty in detecting the genuineness of the news.
Solution	Aim is to find a reliable and accurate model that classifies a given tweet as either fake or true.

Table 1.5: Problem Statement with solution

1.6 Objective of the Project:

The objectives of the “Text Based Fake News Prediction “ can be stated as follows:

1. Read one or more tweet/message as input.
2. To provide users with a platform where they could check the veracity of the tweets.

CHAPTER 2

LITERATURE SURVEY

Sentiment analysis has been handled as a Natural Language Processing task at many levels of granularity. Starting from being a document level classification task (Turney, 2002; Pang and Lee, 2004) [4,5], it has been handled at the sentence level (Hu and Liu [2], 2004; Kim and Hovy, 2004 [1]) and more recently at the phrase level (Wilson et al., 2005; Agarwal et al., 2009). Microblog data like Twitter, on which users post real time reactions to and opinions about “everything”, poses newer and different challenges. Some of the early and recent results on sentiment analysis of Twitter data are by Go et al. (2009), (Birmingham and Smeaton, 2010) and Pak and Paroubek (2010) [3]. Go et al. (2009) use distant learning to acquire sentiment data. They use tweet sending in positive emotions like “:)” “:-)” as positive and negative emoticons like “:(” “:-)” as negative. They build models using Naive Bayes, Max Ent and Support Vector Machines (SVM), and they report SVM outperforms other classifiers. In terms of feature space, they try a Unigram, Bigram model in conjunction with parts-of-speech (POS) features. They note that the unigram model outperforms all other models. Specifically, bigrams and POS features do not help. Pak and Paroubek (2010) [3] collect data following a similar distant learning paradigm. They perform a different classification task though: subjective versus objective.

For subjective data they collect the tweets ending with emoticons in the same manner as Go et al. (2009). For objective data they crawl twitter accounts of popular newspapers like “New York Times”, “Washington Posts” etc. They report that POS and bigrams both help (contrary to results presented by Go et al. (2009)). Both these approaches, however, are primarily based on ngram models. Moreover, the data they use for training and testing is collected by search queries and is therefore biased. In contrast, we present features that achieve a significant gain over a unigram baseline. In addition we explore a different method of data representation and report significant improvement over the unigram models. Another contribution of this paper is that we report results on manually annotated data that does not suffer from any known biases. Our data will be a random sample of streaming tweets unlike data collected by using specific queries. The size of our hand-labeled data

- 1 Fake News: A Survey of Research, Detection Methods, and Opportunities Author: XINYI ZHOU, USA REZA ZAFARANI Published in: journalism.org/2017/09/07/news-use-across-social-media-platforms-2017/ Manuscript submitted to ACM

➤ **Abstract:** The explosive growth in fake news and its erosion to democracy, justice, and public trust has increased the demand for fake news analysis, detection and intervention. This survey comprehensively and systematically reviews fake news research. The survey identifies and specifies fundamental theories across various disciplines, e.g., psychology and social science, to facilitate and enhance the interdisciplinary research of fake news. Current fake news research is reviewed, summarized and evaluated. These studies focus on fake news from four perspective: (1) the false knowledge it carries, (2) its writing style, (3) its propagation patterns, and (4) the credibility of its creators and spreaders. We characterize each perspective with various analyzable and utilizable information provided by news and its spreaders, various strategies and frameworks that are adaptable, and techniques that are applicable. By reviewing the characteristics of fake news and open issues in fake news studies, we highlight some potential research tasks at the end of this survey.

2. Fake News Detection

Author: Manisha Gahirwal, Sanjana Moghe, Tanvi Kulkarni, Devansh Kakar, Jayesh Bhatia Published In: 2018, IJARIT

➤ **Abstract:** Fake news, one of the biggest new-age problems has the potential to mould opinions and influence decisions. The proliferation of fake news on social media and Internet is deceiving people to an extent which needs to be stopped. The existing systems are inefficient in giving a precise statistical rating for any given news claim. Also, the restrictions on input and category of news make it less varied. This paper proposes a system that classifies unreliable news into different categories after computing an F-score. This system aims to use various NLP and classification techniques to help achieve maximum accuracy.

3. Fake News or Truth? Using Satirical Cues to Detect Potentially Misleading News

Author: Victoria L. Rubin, Niall J. Conroy Publications: 2016 Association for Computational Linguistics

- **Abstract:** Satire is an attractive subject in deception detection research: it is a type of deception that intentionally incorporates cues revealing its own deceptiveness. Whereas other types of fabrications aim to instill a false sense of truth in the reader, a successful satirical hoax must eventually be exposed as a jest. This paper provides a conceptual overview of satire and humor, elaborating and illustrating the unique features of satirical news, which mimics the format and style of journalistic reporting. Satirical news stories were carefully matched and examined in contrast with their legitimate news counterparts in 12 contemporary news topics in 4 domains (civics, science, business, and “soft” news). Building on previous work in satire detection, we proposed an SVM based algorithm, enriched with 5 predictive features (Absurdity, Humor, Grammar, Negative Affect, and Punctuation) and tested their combinations on 360 news articles. Our best predicting feature combination (Absurdity, Grammar and Punctuation) detects satirical news with a 90% precision and 84% recall (F-score=87%). Our work in algorithmically identifying satirical news pieces can aid in minimizing the potential deceptive impact of satire.

4. Early Detection of Fake News on Social Media Through Propagation Path Classification with Recurrent and Convolutional Networks

5. Author: Yang Liu, Yi-Fang Brook Wu Published in: 2018, Association for the Advancement of Artificial Intelligence

- **Abstract:** In the midst of today’s pervasive influence of social media, automatically detecting fake news is drawing significant attention from both the academic communities and the general public. Existing detection approaches rely on machine learning algorithms with a variety of news characteristics to detect fake news. However, such approaches have a major limitation on detecting fake news early, i.e., the information required for

detecting fake news is often unavailable or inadequate at the early stage of news propagation. As a result, the accuracy of early detection of fake news is low. To address this limitation, in this paper, we propose a novel model for early detection of fake news on social media through classifying news propagation paths. We first model the propagation path of each news story as a multivariate time series in which each tuple is a numerical vector representing characteristics of a user who engaged in spreading the news. Then, we build a time series classifier that incorporates both recurrent and convolutional networks which capture the global and local variations of user characteristics along the propagation path respectively, to detect fake news. Experimental results on three real-world datasets demonstrate that our proposed model can detect fake news with accuracy 85% and 92% on Twitter and Sina Weibo respectively in 5 minutes after it starts to spread, which is significantly faster than state-of-the-art baselines.

6. Prominent Features of Rumor Propagation in Online Social Media Author: Sejeong Kwon, Meeyoung Cha, Kyomin Jung, Wei Chen Published in: International Conference on Data Mining. IEEE.

➤ **Abstract:** The problem of identifying rumors is of practical importance especially in online social networks, since information can diffuse more rapidly and widely than the offline counterpart. In this paper, we identify characteristics of rumors by examining the following three aspects of diffusion: temporal, structural, and linguistic. For the temporal characteristics, we propose a new periodic time series model that considers daily and external shock cycles, where the model demonstrates that rumor likely have fluctuations over time. We also identify key structural and linguistic differences in the spread of rumors and non-rumors.

7. Towards a highly effective and robust Web credibility evaluation system.
8. Author: Xin Liu a, Radoslaw Nielek , Paulina Adamska , Adam Wierzbicki , Karl Aberer Published in: Elsevier,2015

➤ **Abstract:** By leveraging crowdsourcing, Web credibility evaluation systems (WCESs) have become a promising tool to assess the credibility of Web

content, e.g., Web pages. However, existing systems adopt a passive way to collect users' credibility ratings, which incurs two crucial challenges: (1) a considerable fraction of Web content have few or even no ratings, so the coverage (or effectiveness) of the system is low; (2) malicious users may submit fake ratings to damage the reliability of the system. In order to realize a highly effective and robust WCES, we propose to integrate recommendation functionality into the system. On the one hand, by fusing Matrix Factorization and Latent Dirichlet Allocation, a personalized Web content recommendation model is proposed to attract users to rate more Web pages, i.e., the coverage is increased. On the other hand, by analysing a user's reaction to the recommended Web content, we detect imitating attackers, which have recently been recognized as a particular threat to WCES to make the system more robust. Moreover, an adaptive reputation system is designed to motivate users to more actively interact with the integrated recommendation functionality. We conduct experiments using both real datasets and synthetic data to demonstrate how our proposed recommendation components significantly improve the effectiveness and robustness of existing

CHAPTER 3

THEORETICAL BACKGROUND

Fake news is passing wrong information through some media like social media, global news etc. With the development of media, it's getting hard to distinguish whether the news is true or not. Fake news can be simply explained as a piece of article which is usually written for economic, personal or political gains. Detection of such fake news is possible by using various NLP techniques, Machine learning. In this paper, we are trying to produce a model that can accurately predict the likelihood the news by taking twitter as a source of information

Twitter is a social networking site which allows its registered members to post updates, status, messages etc. These are known as tweets and the length of such tweets are limited to 280 characters (previously 140). There is also a retweet mechanism which is efficient for redistribution of Tweets. Twitter is microblogging in nature which means its different from traditional blog and its contents are smaller in both actual and aggregated file size. Microblog allows users to exchange small elements of contents such as short sentences, images or video link, this feature is twitter may be reason for its popularity. The service of twitter gained worldwide popularity in 2012, more than 100 million users posted 340 million tweets a day, currently in 2020: there are 330 million monthly active users and 145 million daily active users on twitter.

3.1 Existing technique

The main drawback is that the process of developing new words are easy and constant and the use of shortened words are becoming more common on platform for example gtg which means got to go There are main flaws in twitter which apparently led to emergence of fake news few of the reasons are listed here. The first flaw is that since people can freely post their own content, the quality of their opinions cannot be guaranteed. For example, instead of sharing topic related opinions, online spammers post spam on forums. Some spam are meaningless at all, while others have irrelevant opinions also known as fake opinions The

second flaw is that truth of such online data is not always available. A ground truth is like a tag of a certain opinion, indicating whether the opinion is true, false, or neutral.

3.2 Proposed Technique

We collected data using the Twitter public API which allows developers to extract tweets from twitter programmatically. The collected data, because of the random and casual nature of tweeting, it needs to be filtered to remove unnecessary information. Filtering out such problematic tweets which are redundant ones, and ones with no proper sentences was done next. This step is called pre-processing. As the pre-processing phase was done in certain extent it was possible to guarantee that analyzing these filtered tweets will give reliable results

With a lot of research on spam mails based on the way they are written, words used, researchers have come up with a database of the words most probably seen in the spam messages or emails. If a certain message contains these words or number of words, these are addressed to as spam. Such rules have also deployed on social media platform with a notable success rate. Although the main drawback is that the process of developing new words are easy and constant and the use of shortened words are becoming more common on platform for example lol which means laugh out loud. Pattern matching techniques are being used to detect these shortened words on these platforms. For instance, from any account a tweet is published regarding trending information on the social media platform or a new account not more than a day old starts advertising about the trending topics is regarded as fake.

In machine learning, naive Bayes classifiers is one of the Bayesian variety is a family of simple probabilistic classifiers. The tautological Bayesian Machine Learning algorithm is the Naive Bayes classifier, which utilizes Bayes Rule with the strong independence assumption that features of the dataset are conditionally independent of each other, given we know the class of data. We can apply this to spam filtering. Naive Bayes is a straightforward technique for constructing classification models that assign class labels to problem instances, represented as vectors of feature values, where the class labels are drawn from some finite set. It is not a single algorithm for training such classifiers, but a family of algorithms based

on a common principle: all naive Bayes classifiers assume that the value of a particular feature is independent of the value of any other feature, given the class variable.

Natural language processing (NLP) is a branch of artificial intelligence that helps computers understand, interpret and manipulate human language. NLP draws from many disciplines, including computer science and computational linguistics, in its pursuit to fill the gap between human communication and computer understanding. While natural language processing isn't a new science, the technology is rapidly advancing thanks to an increased interest in human-to-machine communications, plus an availability of big data, powerful computing and enhanced algorithms. Natural language processing includes many different techniques for interpreting human language, ranging from statistical and machine learning methods to rules-based and algorithmic approaches

Chapter 4

SYSTEM REQUIREMENT SPECIFICATION

A System Requirement Specification (SRS) is basically an organization's understanding of a customer or potential client's system requirements and dependencies at a particular point prior to any actual design or development work. The information gathered during the analysis is translated into a document that defines a set of requirements. It gives the brief description of the services that the system should provide and also the constraints under which, the system should operate. Generally, SRS is a document that completely describes what the proposed software should do without describing how the software will do it. It's a two-way insurance policy that assures that both the client and the organization understand the other's requirements from that perspective at a given point in time.

SRS document itself states in precise and explicit language those functions and capabilities a software system (i.e., a software application, an ecommerce website and so on) must provide, as well as states any required constraints by which the system must abide. SRS also functions as a blueprint for completing a project with as little cost growth as possible. SRS is often referred to as the "parent" document because all subsequent project management documents, such as design specifications, statements of work, software architecture specifications, testing and validation plans, and documentation plans, are related to it.

Requirement is a condition or capability to which the system must conform. Requirement Management is a systematic approach towards eliciting, organizing and documenting the requirements of the system clearly along with the applicable attributes. The elusive difficulties of requirements are not always obvious and can come from any number of sources. There exists a large body of research on the topic of machine learning methods for deception detection, most of it has been focusing on classifying online reviews and publicly available social media posts. The simple content-related n-grams and shallow parts-of-speech (POS) tagging have proven insufficient for the classification task, often failing to account for important context information. Rather, these methods have been shown useful only in tandem with more complex methods of analysis.

4.1 Functional Requirement

Functional Requirement defines a function of a software system and how the system must behave when presented with specific inputs or conditions. These may include calculations, data manipulation and processing and other specific functionality. In this system following are the functional requirements: -

Following are the functional requirements on the system:

1. System should be able to process new tweets stored in database after retrieval.
2. System should be able to analyse data and classify each tweet polarity

4.2 Non-Functional Requirements

Non-functional requirements is a description of features, characteristics and attribute of the system as well as any constraints that may limit the boundaries of the proposed system. The non-functional requirements are essentially based on the performance, information, economy, control and security efficiency and services. Based on these the non-functional requirements are as follows:

- **User friendly:** User friendly describes a hardware device or software interface that is easy to use. It is friendly to the user, meaning it is not difficult to learn or understand. While "user-friendly" is a subjective term, the following are several common attributes found in user-friendly interfaces.
- **System should provide better accuracy:** In simpler terms, given a set of data points from repeated measurements of the same quantity, the set can be said to be **accurate** if their average is close to the true value of the quantity being measured, while the set can be said to be precise if the values are close to each other.
- **To perform with efficient throughput:** performing or functioning in the best possible manner with the least waste of time and effort; having and using requisite knowledge, skill, and industry; competent; capable: a reliable, efficient assistant. satisfactory and economical to use. The effective throughput ratio tells you how much of the available bandwidth your single flow uses. The effective throughput ratio for a single flow is highest near the data source and decreases with distance. This is

orthogonal to bandwidth utilization, which is a measure of capacity consumed by many flows.

- **Response Time:** Response time, in the context of computer technology, is the elapsed time between an inquiry on a system and the response to that inquiry. Used as a measurement of system performance, response time may refer to service requests in a variety of technologies. Low response times may be critical to successful computing.

4.2.1 Product Requirements

Platform Independency: Standalone executables for embedded systems can be created so the algorithm developed using available products could be downloaded on the actual hardware and executed without any dependency to the development and modeling platform.

Correctness: It followed a well-defined set of procedures and rules to compute and also rigorous testing is performed to confirm the correctness of the data.

Ease of Use: Model Coder provides an interface which allows the user to interact in an easy manner.

Modularity: The complete product is broken up into many modules and well-defined interfaces are developed to explore the benefit of flexibility of the product.

Robustness: This software is being developed in such a way that the overall performance is optimized and the user can expect the results within a limited time with utmost relevancy and correctness

Non-functional requirements are also called the qualities of a system. These qualities can be divided into execution quality & evolution quality. Execution qualities are security & usability of the system which are observed during run time, whereas evolution quality involves testability, maintainability, extensibility or scalability.

4.2.2 Basic Operational Requirements

The customers are those that perform the eight primary functions of systems engineering, with special emphasis on the operator as the key customer. Operational requirements will define The basic need and, at a minimum, will be related to these following points: -

- **Mission profile or scenario:** It describes about the procedures used to accomplish mission objective. It also finds out the effectiveness or efficiency of the system.
- **Performance and related parameters:** It points out the critical system parameters to accomplish the mission Simulink Model Coder for control system Operation Models
- **Utilization environments:** It gives a brief outline of system usage. Finds out appropriate environments for effective system operation.
- **Operational life cycle:** It defines the system lifetime.

4.2.3 Hardware Requirements

Processor : Pentium IV 2.4Ghz
Hard Disk : 250GB
RAM : 1GB
Monitor : 15 VGA Colour
Mouse : Optical

4.2.4 Software Requirements

Operating System : Windows 10
Coding Language : Python
Database GUI : Python
Database stored in .csv files.

Chapter 5

SYSTEM ANALYSIS

Analysis is the process of finding the best solution to the problem. System analysis is the process by which we learn about the existing problems, define objects and requirements and evaluates the solutions. It is the way of thinking about the organization and the problem it involves, a set of technologies that helps in solving these problems. Feasibility study plays an important role in system analysis which gives the target for design and development.

5.1 Feasibility Study

A feasibility study is a preliminary study which investigates the information of prospective users and determines the resources requirements, costs, benefits and feasibility of proposed system. A feasibility study takes into account various constraints within which the system should be implemented and operated. In this stage, the resource needed for the implementation such as computing equipment, manpower and costs are estimated. The estimated are compared with available resources and a cost benefit analysis of the system is made. The feasibility analysis activity involves the analysis of the problem and collection of all relevant information relating to the project. The main objectives of the feasibility study are to determine whether the project would be feasible in terms of economic feasibility, technical feasibility and operational feasibility and schedule feasibility or not. It is to make sure that the input data which are required for the project are available. Thus we evaluated the feasibility of the system in terms of the following categories:

- Technical feasibility
- Operational feasibility
- Economic feasibility
- Schedule feasibility

5.1.1 Technical Feasibility

Evaluating the technical feasibility is the trickiest part of a feasibility study. This is because, at the point in time there is no any detailed designed of the system, making it difficult to access issues like performance, costs (on account of the kind of technology to be deployed) etc. A number of issues have to be considered while doing a technical analysis; understand the different technologies involved in the proposed system. Before commencing the project, we have to be very clear about what are the technologies that are to be required for the development of the new system. Is the required technology available? Our system "Text Based Fake News Prediction" is technically feasible since all the required tools are easily available. Python and Php with JavaScript can be easily handled. Although all tools seems to be easily available there are challenges too.

5.1.2 Operational Feasibility

Proposed project is beneficial only if it can be turned into information systems that will meet the operating requirements. Simply stated, this test of feasibility asks if the system will work when it is developed and installed. Are there major barriers to Implementation? The proposed was to make a simplified web application. It is simpler to operate and can be used in any webpages. It is free and not costly to operate.

5.1.3 Economic Feasibility

Economic feasibility attempts to weigh the costs of developing and implementing a new system, against the benefits that would accrue from having the new system in place. This feasibility study gives the top management the economic justification for the new system. A simple economic analysis which gives the actual comparison of costs and benefits are much more meaningful in this case. In addition, this proves to be useful point of reference to compare actual costs as the project progresses. There could be various types of intangible benefits on account of automation. These could increase improvement in product quality, better decision making, and timeliness of information, expediting activities, improved accuracy of operations, better documentation and record keeping, faster retrieval of information. This is a web based application. Creation of application is not costly.

5.1.4 Schedule Feasibility

A project will fail if it takes too long to be completed before it is useful. Typically, this means estimating how long the system will take to develop, and if it can be completed in a given period of time using some methods like payback period. Schedule feasibility is a measure how reasonable the project timetable is. Given our technical expertise, are the project deadlines reasonable? Some project is initiated with specific deadlines. It is necessary to determine whether the deadlines are mandatory or desirable. A minor deviation can be encountered in the original schedule decided at the beginning of the project. The application development is feasible in terms of schedule.

5.2 Analysis

5.2.1 Performance Analysis

For the complete functionality of the project work, the project is run with the help of healthy networking environment. Performance analysis is done to find out whether the proposed system. It is essential that the process of performance analysis and definition must be conducted in parallel.

5.2.2 Technical Analysis

System is only beneficial only if it can be turned into information systems that will meet the organization's technical requirement. Simply stated this test of feasibility asks whether the system will work or not when developed & installed, whether there are any major barriers to implementation. Regarding all these issues in technical analysis there are several points to focus on: -

- **Changes to bring in the system:** All changes should be in positive direction, there will be increased level of efficiency and better customer service.
- **Required skills:** Platforms & tools used in this project are widely used. So the skilled manpower is readily available in the industry.
- **Acceptability:** The structure of the system is kept feasible enough so that there should not be any problem from the user's point of view.

5.2.3 Economical Analysis

Economic analysis is performed to evaluate the development cost weighed against the ultimate income or benefits derived from the developed system. For running this system, we need not have any routers which are highly economical. So the system is economically feasible enough.

Chapter 6

SYSTEM DESIGN

Design is a meaningful engineering representation of something that is to be built. It is the most crucial phase in the developments of a system. Software design is a process through which the requirements are translated into a representation of software. Design is a place where design is fostered in software Engineering. Based on the user requirements and the detailed analysis of the existing system, the new system must be designed. This is the phase of system designing. Design is the perfect way to accurately translate a customer's requirement in the finished software product. Design creates a representation or model, provides details about software data structure, architecture, interfaces and components that are necessary to implement a system. The logical system design arrived at as a result of systems analysis is converted into physical system design.

6.1 System development methodology

System development method is a process through which a product will get completed or a product gets rid from any problem. Software development process is described as a number of phases, procedures and steps that gives the complete software. It follows series of steps which is used for product progress. The development method followed in this project is incremental model.

6.1.1 Model phases

The Incremental Model is a process of software development where requirements are broken down into multiple standalone modules of software development cycle. Incremental development is done in steps from requirement analysis, design and development, implementation, testing/verification.

1. Requirement analysis: In the first phase of the incremental model, the product analysis expertise identifies the requirements. And the system functional requirements are understood by the requirement analysis team. To develop the software under the incremental model, this phase performs a crucial role.

2. Design & Development: In this phase of the Incremental model of SDLC, the design of the system functionality and the development method are finished with success. When software develops new practicality, the incremental model uses style and development phase.

3. Testing: In the incremental model, the testing phase checks the performance of each existing function as well as additional functionality. In the testing phase, the various methods are used to test the behaviour of each task.

4. Implementation: Implementation phase enables the coding phase of the development system. It involves the final coding that design in the designing and development phase and tests the functionality in the testing phase. After completion of this phase, the number of the product working is enhanced and upgraded up to the final system product.

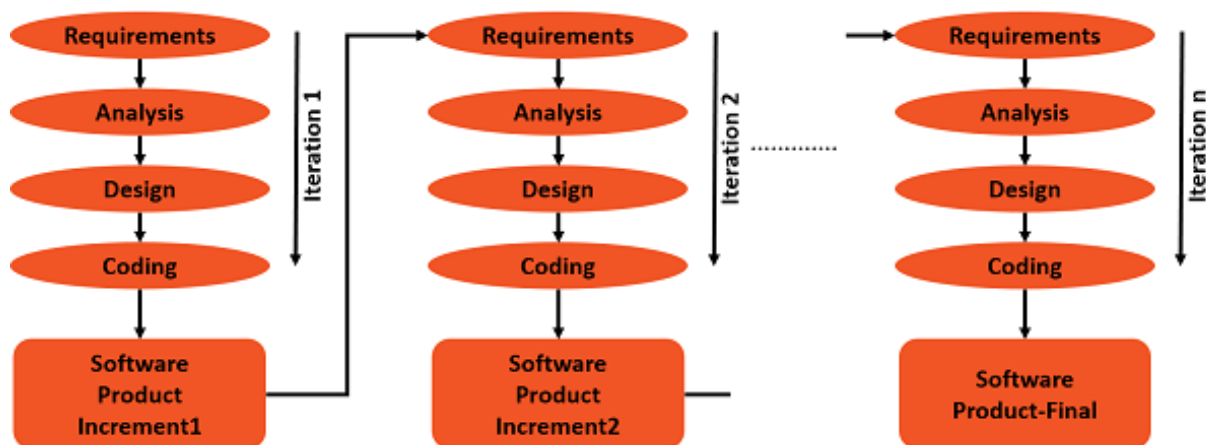


Fig 6.1: Incremental Model

6.1.2 Advantages of the Incremental Model

- Errors are easy to be recognized.
- Easier to test and debug
- More flexible.
- Simple to manage risk because it handled during its iteration.
- The Client gets important functionality early.

6.2 Design Using UML

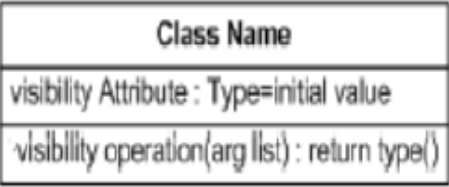
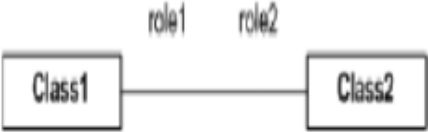



Designing UML diagram specifies, how the process within the system communicates along with how the objects within the process collaborate using both static as well as dynamic UML diagrams since in this ever-changing world of Object Oriented application development, it has been getting harder and harder to develop and manage high quality applications in reasonable amount of time. As a result of this challenge and the need for a universal object modelling language every one could use, the Unified Modeling Language (UML) is the Information industries version of blue print. It is a method for describing the systems architecture in detail. Easier to build or maintains system, and to ensure that the system will hold up to the requirement changes.

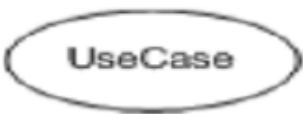
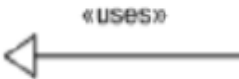




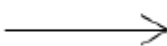

A picture is worth a thousand words, this idiom absolutely fits describing UML. Object-oriented concepts were introduced much earlier than UML. At that point of time, there were no standard methodologies to organize and consolidate the object-oriented development. It was then that UML came into picture.

There are a number of goals for developing UML but the most important is to define some general purpose modeling language, which all modelers can use and it also needs to be made simple to understand and use.

UML diagrams are not only made for developers but also for business users, common people, and anybody interested to understand the system. The system can be a software or non-software system. Thus it must be clear that UML is not a development method rather it accompanies with processes to make it a successful system.

The goal of UML can be defined as a simple modeling mechanism to model all possible practical systems in today's complex environment.

Sl. No	Symbol Name	Symbol	Description
1	Class		Classes represent a collection of similar entities grouped together.
2	Association		Association represents a static relation between classes.
3	Aggregation		Aggregation is a form of association. It aggregates several classes into a single class.
4	Composition		Composition is a special type of aggregation that denotes a strong ownership between classes.
5	Actor		Actor is the user of the system that reacts with the system.

6	Use Case		A use case is an interaction between system and the external environment.
7	Relation (Uses)		It is used for additional purpose communication.
8	Communication		It is the communication between use cases.
9	State		It represents the state of process. Each state goes through various flows.
10	Initial State		It represents initial state of object.
11	Final State		It represents final state of object.
12	Control Flow		It represents decision making process for object.
13	Decision Box		It represents the decision making process from a constraint.


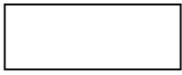

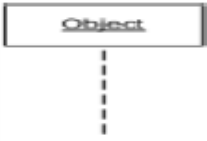
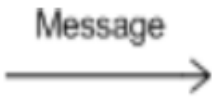
14	Data Process/ State		A circle in a DFD represents a state or process which has been triggered due to some other event or action.
15	External Entity		It represents external entity such as Keyboard, sensors, etc which are used in the system.
16	Transition		It represents any communication that occurs between processes.
17	Object Lifeline		Object lifeline represents the vertical dimension that object communicates.
18	Message		It represents messages exchanged.

Table 6.1: Symbols used in UML

6.3 Data Flow Diagram

The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of the input data to the system, various processing carried out on these data, and the output data is generated by the system. A **data-flow diagram** (DFD) is a way of representing a flow of a data of a process or a system. The DFD also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow, there are no decision rules and no loops. Specific operations based on the data can be represented by a flowchart.

There are several notations for displaying data-flow diagrams. For each data flow, at least one of the endpoints (source and / or destination) must exist in a process. The refined representation of a process can be done in another data-flow diagram, which subdivides this process into subprocesses. The data-flow diagram is part of the structured-analysis modelling tools. When using UML, the activity diagram typically takes over the role of the data-flow diagram. A special form of dataflow plan is a site-oriented data-flow plan.

Data-flow diagrams can be regarded as inverted Petri nets, because places in such networks correspond to the semantics of data memories.

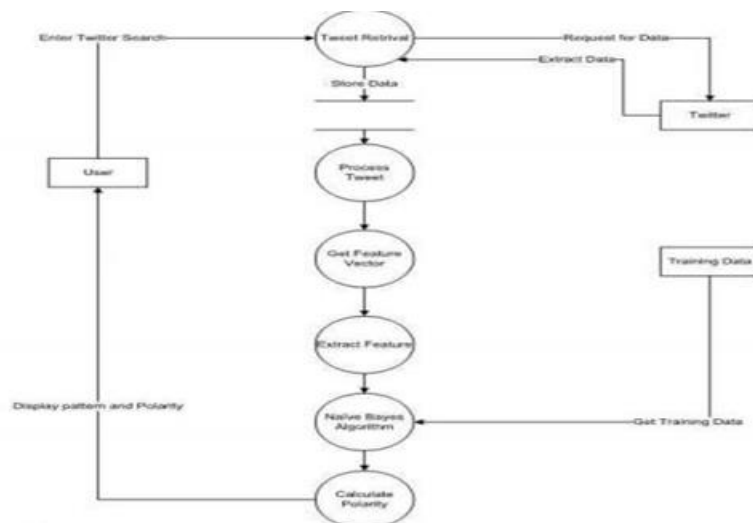


Fig 6.2: Data flow diagram

6.4 Component Diagram

In the Unified Modeling Language, a component diagram depicts how components are wired together to form larger components and or software systems. They are used to illustrate the structure of arbitrarily complex systems.

The component diagram for the decentralized system ideally consists of different modules that are represented together via a common module for the user. The user is required to have the input files in the current folder where the application is being used. It is interesting to note that all the sequence of activities that are taking place are via this module itself, i.e. the parsing and the process of computing the final sequence. The parsing redirects across the other modules till the final code is generated.

6.5 Use case Diagram

A use case defines a goal-oriented set of interactions between external entities and the system under consideration. The external entities which interact with the system are its actors. A set of use cases describe the complete functionality of the system at a particular level of detail and it can be graphically denoted by the use case diagram.

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well.

In software and systems engineering, a use case is a list of steps, typically defining interactions between a role (known in Unified Modeling Language (UML) as an "actor") and a system, to achieve a goal. The actor can be a human, an external system, or time. In systems engineering, use cases are used at a higher level than within software engineering, often representing missions or stakeholder goals. The detailed requirements may then be captured in Systems Modeling Language (SysML) or as contractual statements. The Sequence of activities that are carried out are the same as the other diagrams. Use case for this module indicates the users interaction with the system as a whole rather than individual modules .All the encryption mechanisms are carried out via the login page that redirects the user to the particular functionality that he or she wishes to implement .

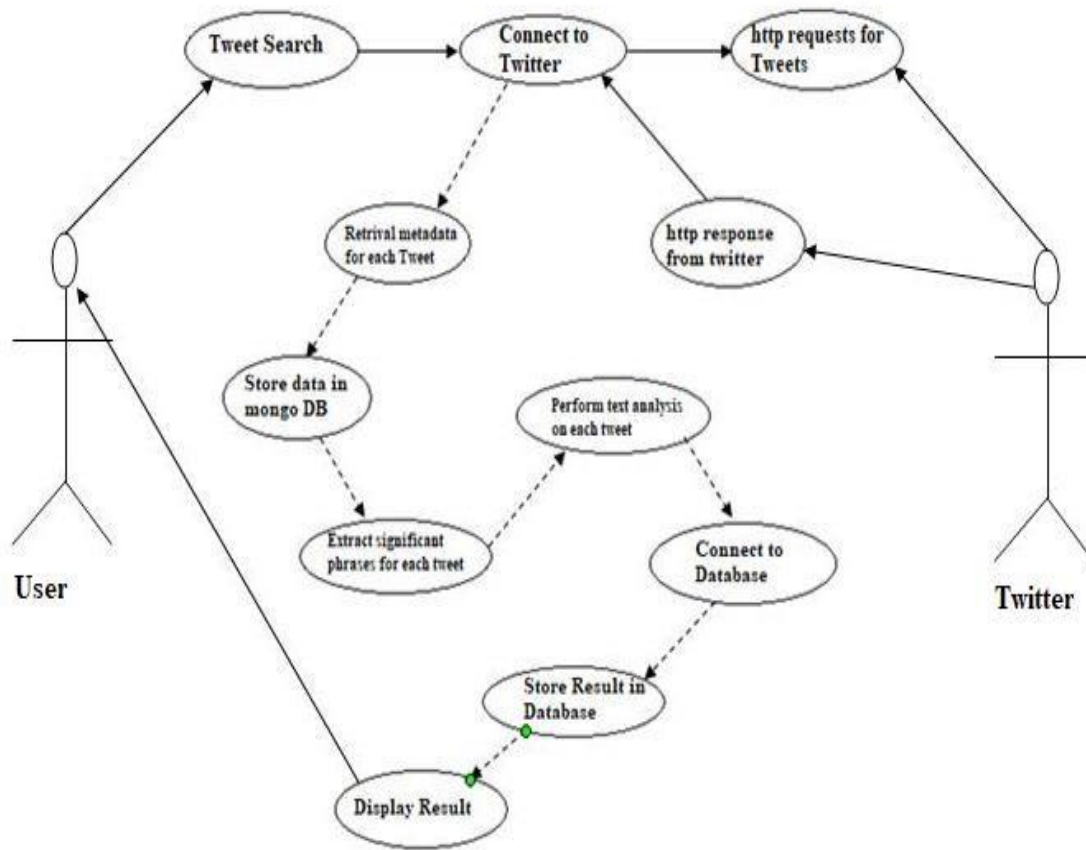


Fig 6.3: Use Case Diagram

6.6 Activity Diagram

An activity diagram shows the sequence of steps that make up a complex process. An activity is shown as a round box containing the name of the operation. An outgoing solid arrow attached to the end of the activity symbol indicates a transition triggered by the completion. Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modelling Language, activity diagrams are intended to model both computational and organisational processes (i.e. workflows). Activity diagrams show the overall flow of control. Activity diagrams are constructed from a limited number of shapes, connected with arrows. The most important shape types:

- rounded rectangles represent actions;
- diamonds represent decisions;
- bars represent the start (split) or end (join) of concurrent activities;

- a black circle represents the start (initial state) of the workflow;
- an encircled black circle represents the end (final state).

The basic purposes of activity diagrams are similar to other four diagrams. It captures the dynamic behavior of the system. Other four diagrams are used to show the message flow from one object to another but activity diagram is used to show message flow from one activity to another. Activity is a particular operation of the system. Activity diagrams are not only used for visualizing dynamic nature of a system but they are also used to construct the executable system by using forward and reverse engineering techniques.

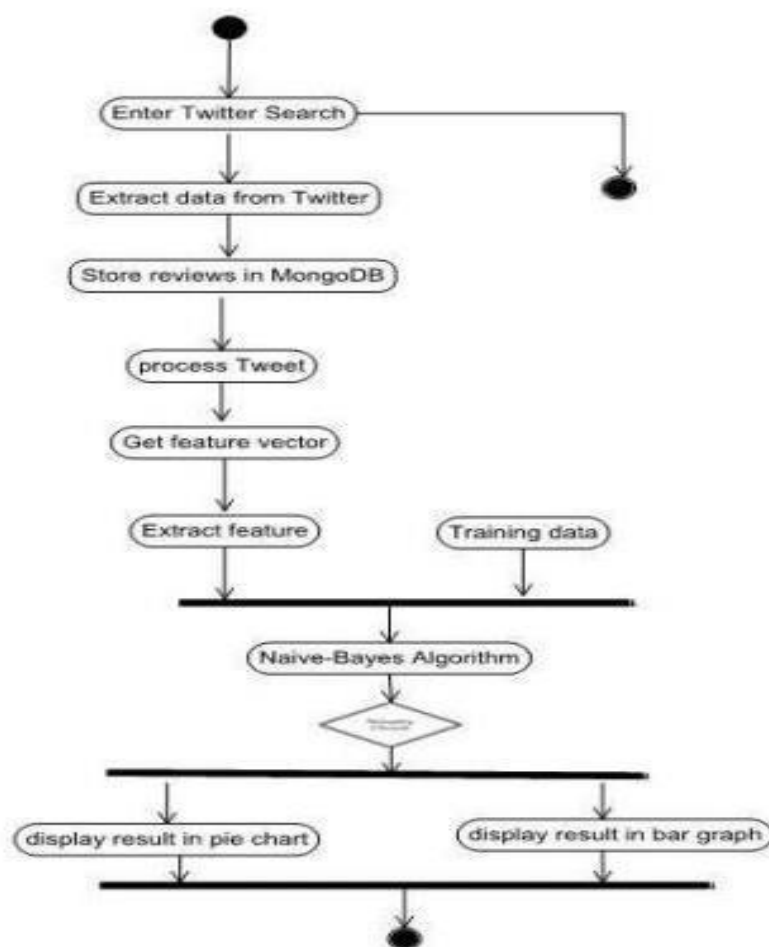


Fig 6.4 : Activity Diagram

6.7 Sequence Diagram

Sequence diagram are an easy and intuitive way of describing the behaviour of a system by viewing the interaction between the system and the environment. A sequence diagram shows an interaction arranged in a time sequence. A sequence diagram has two dimensions: vertical dimension represents time; the horizontal dimension represents the objects existence during the interaction.

A Sequence diagram is an interaction diagram that shows how processes operate with one another and what is their order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios.

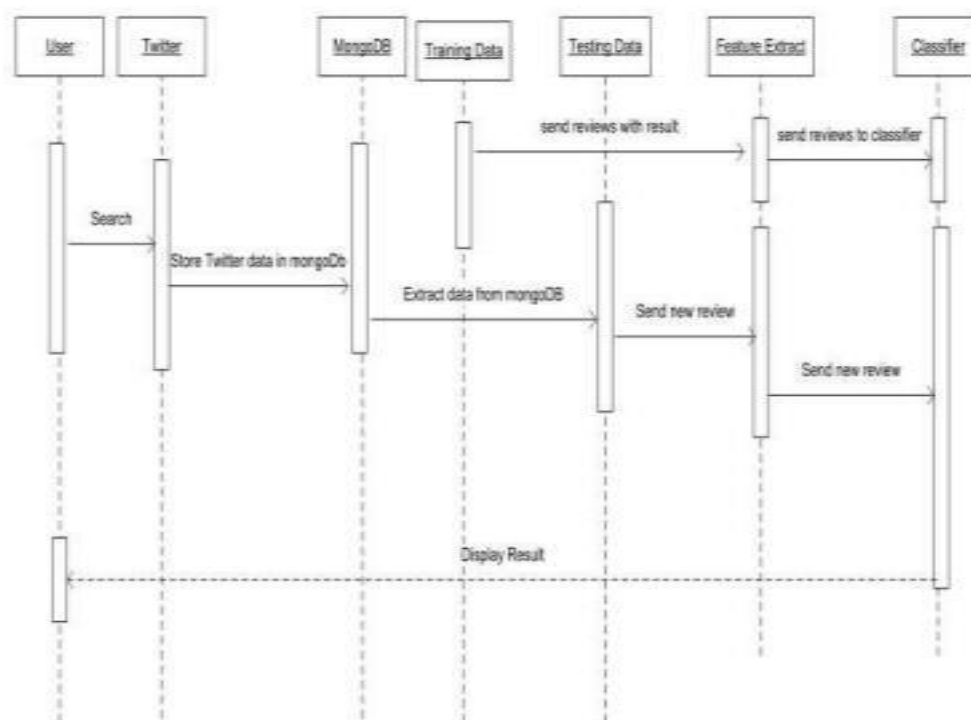


Fig 6.5 : Sequence Diagram

6.8 Flow Chart

A diagram of the sequence of movements or actions of people or things involved in a complex system or activity. A graphical representation of a computer program in relation to its sequence of functions (as distinct from the data it processes). It is a type of diagram that represents a workflow or process. The flowchart shows the steps as boxes of various kinds, and their order by connecting the boxes with arrows. This diagrammatic representation illustrates a solution model to a given problem.



Fig 6.6: Flow Chart

CHAPTER 7

IMPLEMENTATION

The aim of this project is to accurately determine the authenticity of the contents of a particular news article. For this purpose, we have devised a procedure which is intended to fetch favourable results. We first take the URL of the article that the user wants to authenticate, after which the text is extracted from the URL. The extracted text is then passed on to the data pre-processing unit. The data pre-processing unit consists of various processes like the Tokenization and Generation of the word cloud. The outputs from these processes play an important role in further analyzing the data. The core deciding factors that we use to determine the output of our project i.e. if a particular news article is fake or not are the stance of the article and comparison of the article with top google search results. The first method is by using stance detection to in order to analyze the stance of the author. Stance is a mental or an emotional position adopted by the author with respect to something. Stance detection is an important part of NLP and has wide applications. The stance of the author can be divided into various categories like Agree, Disagree, Neutral or Unrelated with respect to the title. Giving each of these categories weights can help us in the final conclusion of whether a news article is fake or not. The second method is to use document similarity or tf-idf to know how similar a document is to top search results. This too can give us an insight into the authenticity of a news article. Next, we need to classify the output into various output classes for which we can use classification algorithms or regression models. The output classes can be true, mostly true, false, and mostly false or we can just present it with a number. For example, 68% true or the score is 7 out of 10 where 1 is completely true and 10 is completely false.

There are primarily two types of approaches for sentiment classification of opinionated texts:

- Using a Machine learning based text classifier such as Naive Bayes
- Using Natural Language Processing

We will be using those machine learning and natural language processing for sentiment analysis of tweet. Machine Learning The machine learning based text classifiers are a kind of supervised machine learning paradigm, where the classifier needs to be trained on some labelled training data before it can be applied to actual classification task. The training data is

usually an extracted portion of the original data hand libelled manually. After suitable training they can be used on the actual test data. The Naive Bayes is a statistical classifier whereas Support Vector Machine is a kind of vector space classifier. The statistical text classifier scheme of Naive Bayes (NB) can be adapted to be used for sentiment classification problem as it can be visualized as a 2-class text classification problem: in positive and negative classes. Support Vector machine (SVM) is a kind of vector space model based classifier which requires that the text documents should be transformed to feature vectors before they are used for classification. Usually the text documents are transformed to multidimensional vectors. The entire problem of classification is then classifying every text document represented as a vector into a particular class. It is a type of large margin classifier. Here the goal is to find a decision boundary between two classes that is maximally far from any document in the training data. This approach needs

- A good classifier such as Naive Byes
- A training set for each class

There are various training sets available on Internet such as Movie Reviews data set, twitter dataset, etc. Class can be Positive, negative. For both the classes we need training data sets.

7.1 Naïve Bayes Classifier (NB)

The Naïve Bayes classifier is the simplest and most commonly used classifier. Naïve Bayes classification model computes the posterior probability of a class, based on the distribution of the words in the document. The model works with the BOWs feature extraction which ignores the position of the word in the document. It uses Bayes Theorem to predict the probability that a given feature set belongs to a particular label.

$$P(\text{label}|\text{features}) = \frac{P(\text{label}) \cdot P(\text{features}|\text{label})}{P(\text{features})}$$

$P(\text{label})$ is the prior probability of a label or the likelihood that a random feature set the label. $P(\text{features}|\text{label})$ is the prior probability that a given feature set is being classified as a label. $P(\text{features})$ is the prior probability that a given feature set is occurred. Given the Naïve assumption which states that all features are independent, the equation could be rewritten as follows:

$$P(\text{label}|\text{features}) = \frac{P(\text{label}) \cdot P(f_1|\text{label}) \cdot \dots \cdot P(f_n|\text{label})}{P(\text{features})}$$

5.1.1.1 Multinomial Naïve Bayes Classifier

Accuracy – around 75%

Algorithm:

i. Dictionary generation

Count occurrence of all word in our whole data set and make a dictionary of some most frequent words

ii. Feature set generation

All document is represented as a feature vector over the space of dictionary words. For each document, keep track of dictionary words along with their number of occurrence in that document.

Formula used for algorithms:

$$\phi_{k|\text{label}=y} = P(x_j = k | \text{label} = y)$$

$$\phi_{k|\text{label}=y} = \frac{\sum_{i=1}^m \sum_{j=1}^{n_i} \mathbb{1}\{x_j^{(i)} = k \text{ and } \text{label}^{(i)} = y\} + 1}{\left(\sum_{i=1}^m \mathbb{1}\{\text{label}^{(i)} = y\} n_i \right) + |V|}$$

$\phi_{k|\text{label}=y}$ = probability that a particular word in document of label (neg/pos) = y will be the kth word in the dictionary.

m = Number of words in i^{th} document.

n_i = Total Number of documents.

Training in this phase We have to generate training data (words with probability of occurrence in positive/negative train data files).

Calculate $\phi_k | label = y$ for each label .
 Calculate $\phi_k | label = y$ for each dictionary words and store the result (Here: label will be negative and positive).

7.2 Natural Language Processing

Natural language processing (NLP) is a field of computer science, artificial intelligence, and linguistics concerned with the interactions between computers and human (natural) languages. This approach utilizes the publicly available library of Senti Word Net, which provides a sentiment polarity values for every term occurring in the document. In this lexical resource each term t occurring in WordNet is associated to three numerical scores $obj(t)$, $pos(t)$ and $neg(t)$, describing the objective, positive and negative polarities of the term, respectively. These three scores are computed by combining the results produced by eight ternary classifiers. WordNet is a large lexical database of English. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept. WordNet is also freely and publicly available for download. WordNet's structure makes it a useful tool for computational linguistics and natural language processing. It groups words together based on their meanings. Synet is nothing but a set of one or more Synonyms. This approach uses Semantics to understand the language. Major tasks in NLP that helps in extracting sentiment from a sentence:

- Extracting part of the sentence that reflects the sentiment
- Understanding the structure of the sentence
- Different tools which help process the textual data

Basically, Positive and Negative scores got from Senti Word Net according to its part-of-speech tag and then by counting the total positive and negative scores we determine the sentiment polarity based on which class (i.e. either positive or negative) has received the highest score.

7.3 Programming tools

7.3.1 Python

Python is a widely used high-level, general-purpose, interpreted, dynamic programming language. Its design philosophy emphasizes code readability, and its syntax allows programmers to express concepts in fewer lines of code than possible in languages such as C or Java. The language provides constructs intended to enable writing clear programs on both a small and large scale. Hence, you can use the programming language for developing both desktop and web applications. Also, you can use Python for developing complex scientific and numeric applications. Python is designed with features to facilitate data analysis and visualization.

Python's run time must work harder than Java's. For these reasons, Python is much better suited as a "glue" language, while Java is better characterized as a low-level implementation language. In fact, the two together make an excellent combination.

7.3.2 NLTK (Natural Language Tool Kit)

NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries, and an active discussion forum. NLTK has been called “a wonderful tool for teaching, and working in, computational linguistics using Python,” and “an amazing library to play with natural language.” NLTK is suitable for linguists, engineers, students, educators, researchers, and industry users alike. Natural Language Processing with Python provides a practical introduction to programming for language processing. Written by the creators of NLTK, it guides the reader through the fundamentals of writing Python programs, working with corpora, categorizing text, analyzing linguistic structure, and more

7.3.3 Code

```
from sklearn import svm
import urllib
import numpy as np
import matplotlib, pyplot as plt
import pandas
filename = 'dataset1.csv'
names =
['WC','WPS','Sizltr','Dic','Numerals','harmvirtue','harmvice','fairnessvirtue','fairnessvice','ingrouvirtue','ingrouvice','authorityvirtue','authorityvice','purityvirtue','purityvice','moralitygeneral','AllPct','depression']

data_set=pandas.read_csv(filename,names=names)

#print(data_set)

#print("Dataset::", data_set['WC'])
X=pandas.DataFrame(data_set)

#dec=data_set['decision']

d=pandas.Series(data_set['depression'])

y=pandas.DataFrame(d)

#y = column(y, warn=True)

del X [data_set. columns [17]]

X.columns=[data_set.columns[0],data_set.columns[1],data_set.columns[2],data_set.columns[3],data_set.columns[4],data_set.columns[5],data_set.columns[6],data_set.columns[7],data_set.columns[8],data_set.columns[9],data_set.columns[10],data_set.columns[11],data_set.columns[12],data_set.columns[13],data_set.columns[14],data_set.columns[15],data_set.columns[16]]
```



```
ns[12],data_set.columns[13],data_set.columns[14],data_set.columns[15],data_set.columns[16  
]]
```

```
//Get tweets
```

```
#http://www.tweepy.org/
```

```
import tweepy
```

```
import sys
```

```
import csv
```

```
from textblob import TextBlob
```

```
#Get your Twitter API credentials and enter them here
```

```
consumer_key = "IgM5R8xVMGNJBZuqQ42RbEH15"
```

```
consumer_secret = "Myu6ernLGk2ODUrsBNWr6deZu01MzlnVMTkvz93JARFJVHuoqU"
```

```
access_key = "339802190-clJpgJofm8n6tXUPVwrnWDILUZFdEGvecKqynjbY"
```

```
access_secret = "qqOr9Fx5Sh3uIxS0nf8Y0ic1rnoK2HJhQg0uAIPeGL5fh"
```

```
#method to get a user's last tweets
```

```
def get_tweets(username):
```

```
    #http://tweepy.readthedocs.org/en/v3.1.0/getting_started.html#api
```

```
    auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
```

```
    auth.set_access_token(access_key, access_secret)
```

```
    api = tweepy.API(auth)
```

```
    print('HI')
```

```
    #set count to however many tweets you want
```

```
    number_of_tweets = 100
```

```
    #get tweets
```

```
    tweets_for_csv = []
```

```
    for tweepy.Cursor(api.user_timeline, screen_name = username). items(number_of_tweets):
```

```
        #create array of tweet information: username, tweet id, date/time, text
```

```
        #tweets_for_csv.append([username,tweet.id_str,tweet.created_at, tweet.text.encode("utf-  
8")])
```

Text Based Fake News Prediction

```
    tweets_for_csv.append([tweet.text.encode("utf-8")])
#write to a new csv file from the array of tweets
outfile = username + ".txt"
print("writing to " + outfile)
with open (outfile, 'w+') as file:
    writer = csv.writer(file, delimiter=',')
    writer.writerows(tweets_for_csv)

#if we're running this as a script
if __name__ == '__main__':
    name = input ("Enter the name:")
    get_tweets(name)
    """#get tweets for username passed at command line

if len(sys.argv) == 2:
    get_tweets(sys.argv[1])
else:
    print ("Error: enter one username)"""
```

Chapter 8

TESTING

System testing is actually a series of different tests whose primary purpose is to fully exercise the computer-based system. Although each test has a different purpose, all work to verify that all the system elements have been properly integrated and perform allocated functions. The testing process is actually carried out to make sure that the product exactly does the same thing what is supposed to do. In the testing stage following goals are tried to achieve: -

- To affirm the quality of the project.
- To find and eliminate any residual errors from previous stages.
- To validate the software as a solution to the original problem.
- To provide operational reliability of the system.

8.1 Testing Methodologies

There are many different types of testing methods or techniques used as part of the software testing methodology. Some of the important testing methodologies are:

8.1.1 White box testing

White box testing (clear box testing, glass box testing, and transparent box testing or structural testing) uses an internal perspective of the system to design test cases based on internal structure. It requires programming skills to identify all paths through the software. The tester chooses test case inputs to exercise paths through the code and determines the appropriate outputs. While white box testing is applicable at the unit, integration and system levels of the software testing process, it is typically applied to the unit. While it normally tests paths within a unit, it can also test paths between units during integration, and between subsystems during a system level test.

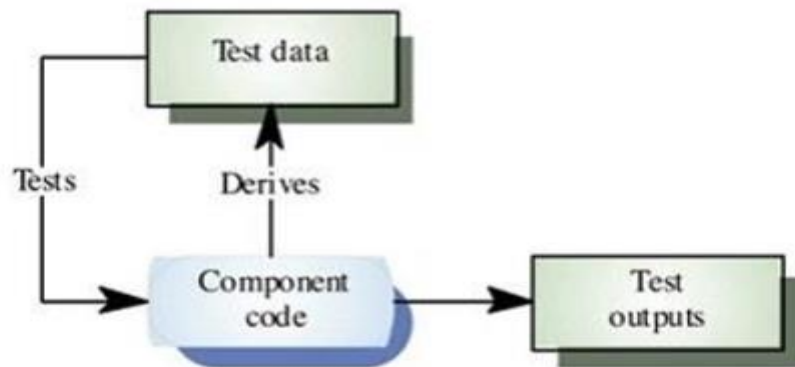


Fig 8.1.1: White Box Testing

Though this method of test design can uncover an overwhelming number of test cases, it might not detect unimplemented parts of the specification or missing requirements, but one can be sure that all paths through the test object are executed.

Using white box testing we can derive test cases that:

- Guarantee that all independent paths within a module have been exercised at least once. Exercise all logical decisions on their true and false sides.
- Execute all loops at their boundaries and within their operational bounds.
- Execute internal data structure to assure their validity

8.1.1.1 Advantages of White Box Testing

- To start the white box testing of the desired application there is no need to wait for user face (UI) to be completed. It covers all possible paths of code which will ensure a thorough testing.
- It helps in checking coding standards.
- Tester can ask about implementation of each section, so it might be possible to remove unused/deadlines of codes helps in reducing the number of test cases to be executed during the black box testing.
- White box testing allows you to help in code optimization.

8.1.1.2 Disadvantages of White Box Testing

- To test the software application a highly skilled resource is required to carry out testing who has good knowledge of internal structure of the code which will increase the cost.
- Updating the test script is required if there is change in requirement too frequently.
- If the application to be tested is large in size, then exhaustive testing is impossible.
- It is not possible for testing each and every path/condition of software program, which might miss the defects in code.
- White box testing is a very expensive type of testing.
- To test each paths or conditions may require different input conditions, so in order to test full application, the tester need to create range of inputs which may be a time consuming.

8.1.2 Black box testing

Black box testing focuses on the functional requirements of the software. It is also known as functional testing. It is a software testing technique whereby the internal workings of the item being tested are not known by the tester. For example, in a black box test on software design the tester only knows the inputs and what the expected outcomes should be and not how the program arrives at those outputs.

The tester does not ever examine the programming code and does not need any further knowledge of the program other than its specifications. It enables us to derive sets of inputs that will fully exercise all functional requirements for a program.

Black box testing is an alternative to white box technique. Rather it is a complementary approach that is likely to uncover a different class of errors in the following categories: -

- Incorrect or missing function.
- Interface errors.
- Performance errors.
- Initialization and termination errors.
- Errors in objects.



Fig 8.1.2: Black Box Testing

8.1.2.1 Advantages of Black Box Testing

- The test is unbiased as the designer and the tester are independent of each other.
- The tester does not need knowledge of any specific programming languages.
- The test is done from the point of view of the user, not the designer.
- Test cases can be designed as soon as the specifications are complete.

8.1.2.2 Disadvantages of Black Box Testing

- The test inputs need to be from large sample space. That is, from a huge set of data this will take time.
- Also it is difficult to identify all possible inputs in limited testing time. So writing test cases is slow and difficult.
- Chances are more that there will be unidentified paths during this testing.

8.2 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. 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.

8.3 System Testing

This information contributes towards reducing the ambiguity about the system. For example, when deciding whether to release a product, the decision makers would need to know the state of the product including aspects such as the conformance of the product to requirements, the usability of the product, any known risks, the product's compliance to any applicable regulations,

Software testing enables making objective assessments regarding the degree of conformance of the system to stated requirements and specifications. System testing checks complete end-end scenarios, as a user would exercise the system. The system has to be tested for correctness of the functionality by setting it up in a controlled environment. System testing includes testing of functional and non-functional requirements. It helps to verify and validate the system. All components of system should have been successfully unit tested and then checked for any errors after integration.

8.4 Quality Assurance

Quality assurance consists of the auditing and reporting functions of management. The goal of quality assurance is to provide management with the data necessary to be informed about product quality, thereby gaining insight and confident that the product quality is meeting its goals. This is an “umbrella activity” that is applied throughout the engineering process. Software quality assurance encompasses: -

- Analysis, design, coding and testing methods and tools
- Formal technical reviews that are applied during each software engineering
- Multi-tiered testing strategy
- Control of software documentation and the change made to it.
- A procedure to ensure compliance with software development standards.
- Measurement and reporting mechanisms.

8.4.1 Quality Factors

An important objective of quality assurance is to track the software quality and assess the impact of methodological and procedural changes on improved software quality. The factors that affect the quality can be categorized into two broad groups:

- Factors that can be directly measured.
- Factors that can be indirectly measured

These factors focus on three important aspects of a software product

- Its operational characteristics
- Its ability to undergo changes
- Its adaptability to a new environment.
- Effectiveness or efficiency in performing its mission
- Duration of its use by its customer.

8.5 Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

- Valid Input: identified classes of valid input must be accepted.
- Invalid Input: identified classes of invalid input must be rejected.
- Functions: identified functions must be exercised.
- Output: identified classes of application outputs must be exercised.
- Systems/Procedures: Interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

CHAPTER 9

RESULT AND PERFORMANCE ANALYSIS

9.1 Snapshot of Input

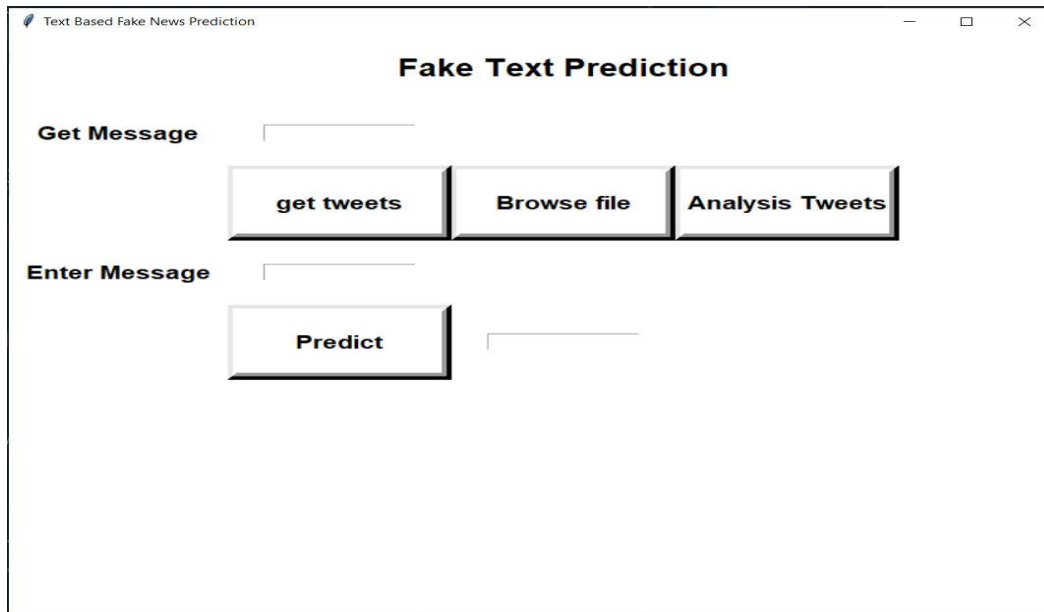
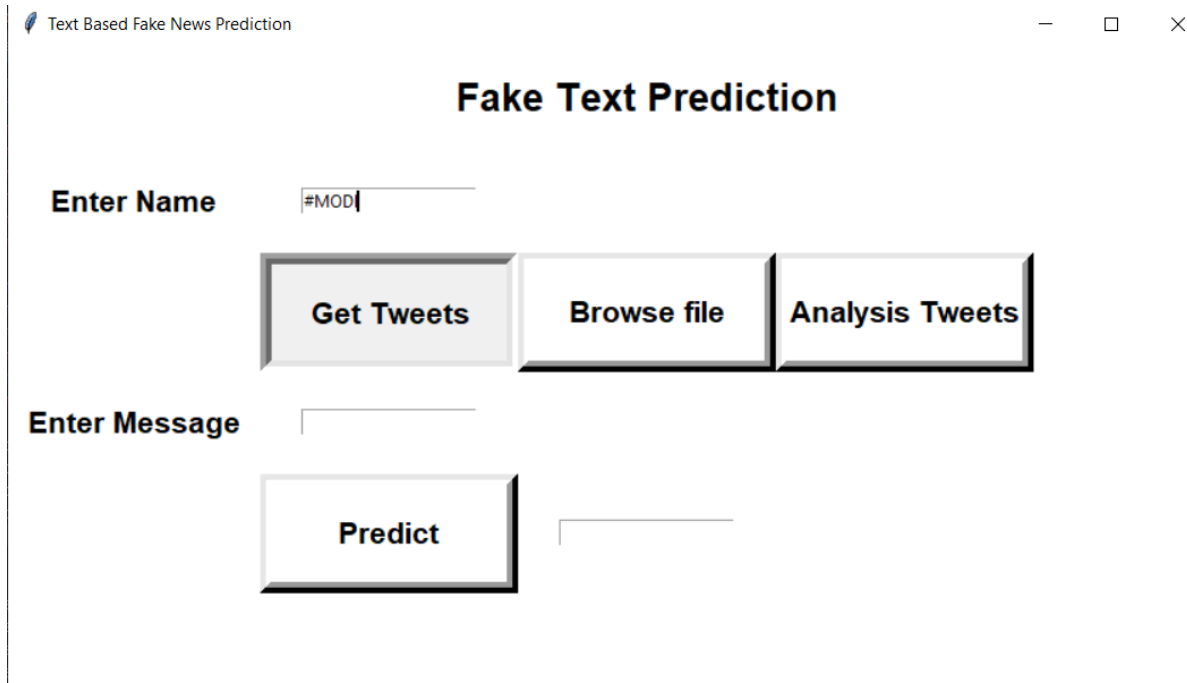


Fig - 9.1 : input window

9.2 SNAPSHOTS OF OUTPUT

In this section snapshots showing the performance of the final code that is generated is shown. Also different scenarios where the user enters invalid input is shown also the output of each module is shown.



Text Based Fake News Prediction

Fake Text Prediction

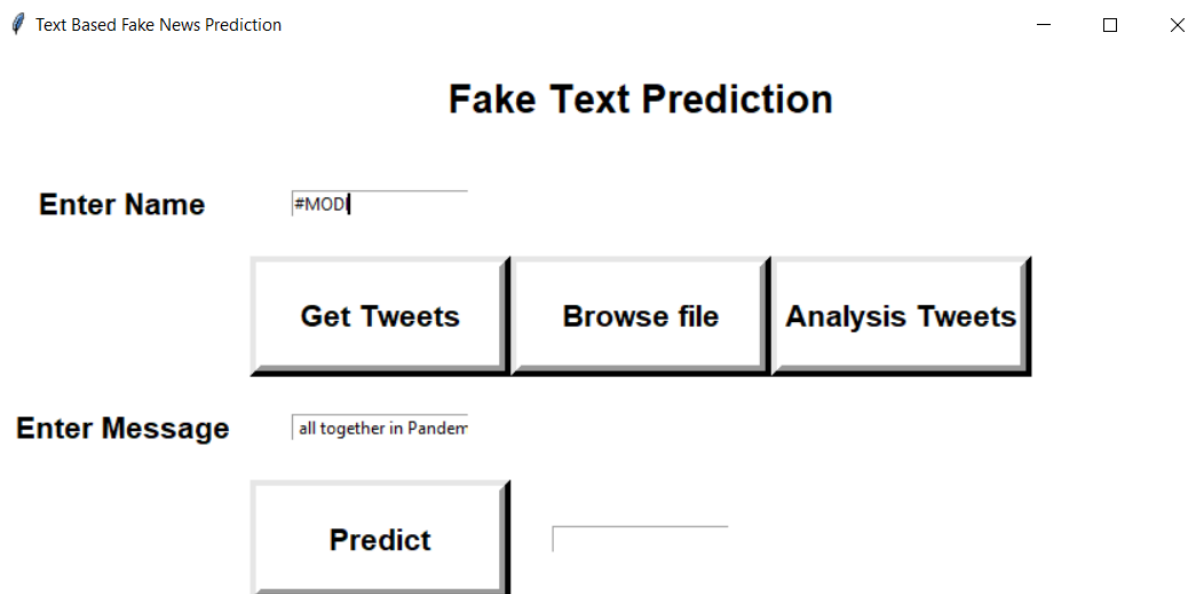
Enter Name

Get Tweets Browse file Analysis Tweets

Enter Message

Predict

Fig – 9.2 Enter the name or hashtag



Text Based Fake News Prediction

Fake Text Prediction

Enter Name

Get Tweets Browse file Analysis Tweets

Enter Message

Predict

Fig – 9.3 : Browse files and get tweet

Fake Text Prediction

Enter Name

Get Tweets

Browse file

Analysis Tweets

Enter Message

Predict

Not a Fake

Fig 9.3 : Predict

Text Based Fake News Prediction

Fake Text Prediction

Enter Name

Get Tweets

Browse file

Analysis Tweets

Enter Message

Predict

Not a Fake

Fig 9.4a : Enter a message to predict its outcome

Text Based Fake News Prediction

Fake Text Prediction

Enter Name

Get Tweets **Browse file** **Analysis Tweets**

Enter Message

Predict

Fig 9.4b : Enter a message to predict its outcome

CHAPTER 10

CONCLUSION AND FUTURE SCOPE

10.1 CONCLUSION

proposed a system to identify the fake text on Twitter. Fake news detection on social media requires a method that is able to find and capture distinctive characteristics, patterns and regularities of the news consumption on the online ecosystem. Existing works on fake news detection mechanism demonstrate the utilization of methods that mostly highlight on specific content-based or social context based approaches for the classification and verification tasks. Evidently, these detection mechanisms which are implemented in restricted domains reveal high accuracy result in predicting deception of the news content.

10.2 Limitation

The system we designed is used to determine the opinion of the people based on twitter data. We somehow completed our project and was able to determine only positivity and negativity of tweet. For neutral data we were unable to merge dataset. Also we are currently analyzing only 25 live tweets. This may not give proper value and results. The results are not much accurate.

10.2.1 FUTURE SCOPE

- Analyzing sentiments on emo/smiley.
- Determining neutrality.
- Potential improvement can be made to our data collection and analysis method.
- Future research can be done with possible improvement such as more refined data and more accurate algorithm.

REFERENCES

- Supanya Aphiwongsophon and Prabhas Chongstitvatana, "Detecting Fake News with Machine Learning Method", CP Journal, 2018.
- Shivam B. Parikh and Pradeep K. Atrey, "Media-Rich Fake News Detection: A Survey", IEEE Conference on Multimedia Information Processing and Retrieval, 2018.
- Mykhailo Granik and Volodymyr Mesyura, "Fake News Detection Using Naive Bayes Classifier", IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON), 2017.
- Shlok Gilda, "Evaluating Machine Learning Algorithms for Fake News Detection", IEEE 15th Student Conference on Research and Development (SCORED), 2017.
- Akshay Jain and Amey Kasbe, "Fake News Detection", IEEE International Students' Conference on Electrical, Electronics and Computer Sciences, 2018.