A Major Project Report on

CLASSIFICATION OF RUMOR AND NON – RUMOR USING SENTIMENT ANALYSIS

Submitted to the

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD

In partial fulfillment of the requirement for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE & ENGINEERING

BY

CH. NAVEEN	**(16WJ1A0551)**
B. SAI SHYAM	**(16WJ1A0537)**
A. ARAVIND	**(16WJ1A0526)**

Under the Esteemed Guidance of

Mr. S. SHREEKANTH

Associate Professor, CSE Department.

Department of Computer Science & Technology

GURU NANAK INSTITUTIONS TECHNICAL CAMPUS
School of Engineering & Technology
Ibrahimpatnam, R.R District 501506

2020

GNI www.gniindia.org

GURU NANAK INSTITUTIONS TECHNICAL CAMPUS

Approved by AICTE - New Delhi | Affiliated to JNTU - Hyderabad | Accredited by National Assessment and Accreditation Council | Accredited by National Board of Accreditation

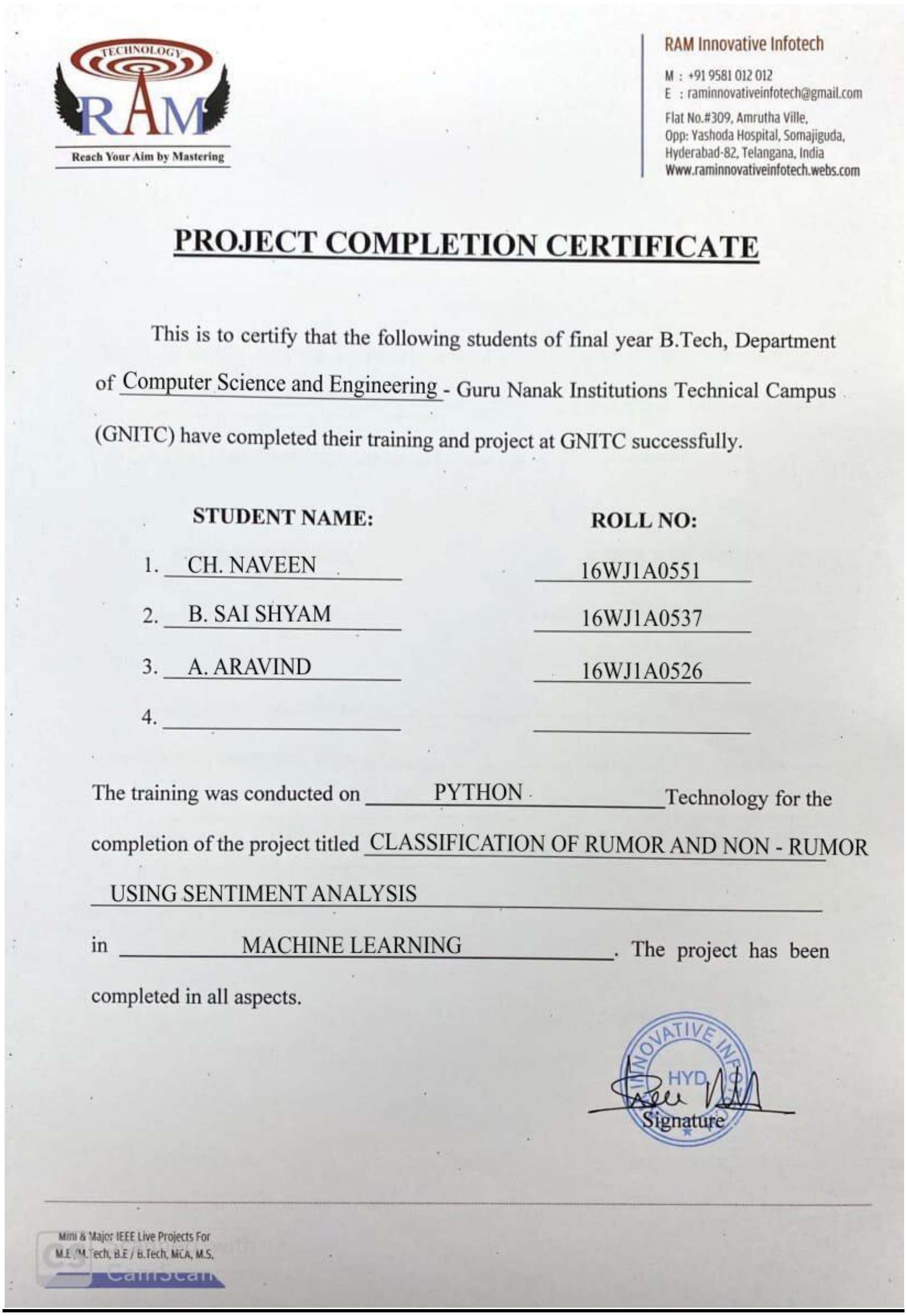
AUTONOMOUS under Section 2 (f) & 12 (b) of University Grants Commission Act

CERTIFICATE

This is to certify that this project report entitled **“CLASSIFICATION OF RUMOR AND NON – RUMOR USING SENTIMENT ANALYSIS”** being submitted by **CH. NAVEEN (16WJ1A0551), B. SAI SHYAM (16WJ1A0537), A. ARAVIND (16WJ1A0526)** in partial fulfillment of the requirements for the degree of **Bachelor of Technology** in **Computer Science & Engineering** of the **Jawaharlal Nehru Technological University Hyderabad** during the academic year 2019-20, is a bonafide record of work carried out under our guidance and supervision at Guru Nanak Institutions Technical Campus.

INTERNAL GUIDE	**PROJECT CO-ORDINATOR**	**HOD CSE**
Mr. S. Shreekanth	**Mr. A. Ravi**	**Dr. J. Rajeshwar**

EXTERNAL EXAMINER



Reach Your Aim by Mastering

RAM Innovative Infotech

M : +91 9581 012 012
E : raminnovativeinfotech@gmail.com
Flat No.#309, Amrutha Ville,
Opp: Yashoda Hospital, Somajiguda,
Hyderabad-82, Telangana, India
Www.raminnovativeinfotech.webs.com

PROJECT COMPLETION CERTIFICATE

This is to certify that the following students of final year B.Tech, Department of Computer Science and Engineering - Guru Nanak Institutions Technical Campus (GNITC) have completed their training and project at GNITC successfully.

STUDENT NAME:	ROLL NO:
1. CH. NAVEEN	16WJ1A0551
2. B. SAI SHYAM	16WJ1A0537
3. A. ARAVIND	16WJ1A0526
4.	

The training was conducted on PYTHON Technology for the completion of the project titled CLASSIFICATION OF RUMOR AND NON - RUMOR USING SENTIMENT ANALYSIS in MACHINE LEARNING. The project has been completed in all aspects.

INNOVATIVE INFO
HYD

[illegible]

Signature

Mini & Major IEEE Live Projects For M.E./M.Tech, B.E / B.Tech, MCA, M.S.

ACKNOWLEDGEMENT

We express a whole hearted gratitude to our chairman, **Sardar T. S. Kohli** and Vice Chairman **Sardar G.S. Kohli** of Guru Nanak Institutions for having well qualified staff and labs furnished with necessary equipment's and computers.

We express a whole hearted gratitude to **Dr. H.S. Saini**, Managing Director, GNITC for providing strong vision in engineering education through accreditation policies under regular training in upgrading education for the staff members

We wish to express our sincere thanks to **Dr. Rishi Sayal,** Associate Director, GNITC for providing us the conductive environment for carrying through our academic schedules and Project with ease.

We have been truly blessed to have a wonderful adviser **Dr. J. Rajeshwar,** Professor & HOD CSE, GNITC for guiding us to explore the development of our work and we express our sincere gratitude towards him for leading us through the completion of Project.

We would like to express our sincere thanks to **Mr. A. Ravi,** Associate Professor, Department of CSE, Mini Project Coordinator, for providing seamless support and right suggestions during the development of the project.

We specially thank our internal guide **Mr. S. Shreekanth,** Associate Professor, Department of CSE, for his suggestions and constant guidance in every stage of the project.

Finally, we would like to thank our family members and friends for their moral support and encouragement to achieve goals.

CH. NAVEEN	(16WJ1A0551)
B. SAI SHYAM	(16WJ1A0537)
A. ARAVIND	(16WJ1A0526)

TABLE OF CONTENTS

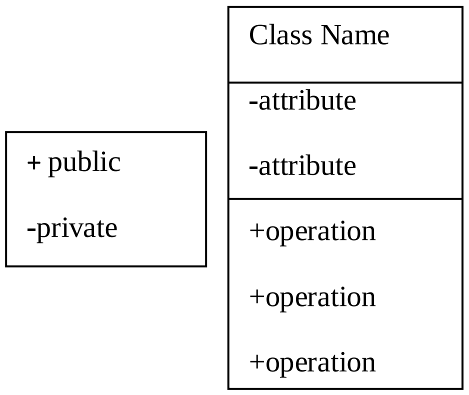
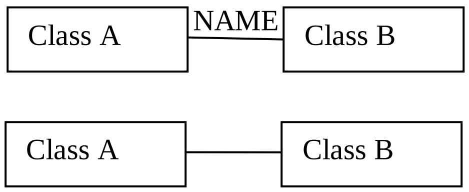
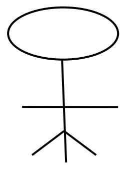
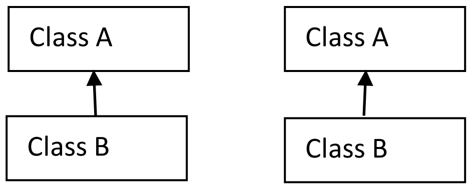
Contents **Page No.**

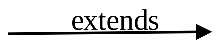
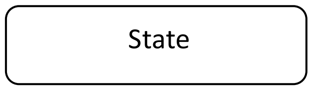
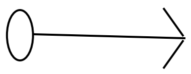
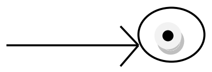
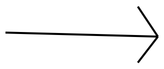
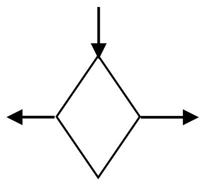
ABSTRACT

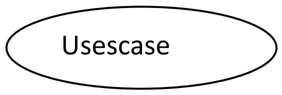
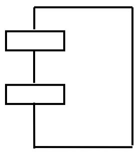
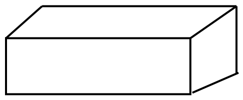
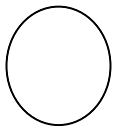
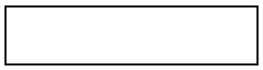
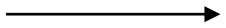
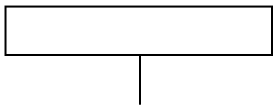
With the intrusion of social media into personal, social and political space of today's world, the iniquities associated with it have also found a place to thrive. One such malevolent act is the spread of rumors through social networks. This survey analyses the various domains where social media finds its relevance and has a considerable impact. It also explores various studies done on the nature of rumors, the characteristics of rumors and the studies identifying the important features for the social media-based rumors. Further, the latest developments in the field of rumor detection through the approaches of Machine learning and Deep learning have also been discussed.

LIST OF FIGURES

LIST OF SYMBOLS

S.NO	NOTATION NAME	NOTATION	DESCRIPTION
1.	Class	 + public -private Class Name -attribute -attribute +operation +operation +operation	Represents a collection of similar entities grouped together.
2.	Association		Associations represents static relationships between classes. Roles represents the way the two classes see each other.
3.	Actor		It aggregates several classes into a single classes.
4.	Aggregation		Interaction between the system and external environment

5.	Relation (uses)	uses	Used for additional process communication.
6.	Relation (extends)		Extends relationship is used when one use case is similar to another use case but does a bit more.
7.	Communication		Communication between various use cases.
8.	State		State of the processs.
9.	Initial State		Initial state of the object
10.	Final state		Final state of the object
11.	Control flow		Represents various control flow between the states.
12.	Decision box		Represents decision making process from a constraint

13.	Usecase		Interact ion between the system and external environment.
14.	Component		Represents physical modules which are a collection of components.
15.	Node		Represents physical modules which are a collection of components.
16.	Data Process/State		A circle in DFD represents a state or process which has been triggered due to some event or acion.
17.	External entity		Represents external entities such as keyboard,sensors,etc.
18.	Transition		Represents communication that occurs between processes.
19.	Object Lifeline		Represents the vertical dimensions that the object communications.

LIST OF ABBREVATION

S.NO	ABBREVATION	EXPANSION
1.	DB	DataBase
2.	JVM	Java Virtual Machine
3.	JSP	Java Server Page
4.	PWS	Personalised Web Search
5.	UPS	User Personalised Search
6.	JRE	Java Runtime Environment

CHAPTER 1

INTRODUCTION

1.1 GENERAL

Social media has become an integral part of the digital existence of the people nowadays. A person has a number of social media accounts on different platforms identifying his digital existence. A person uses these different types of accounts (e.g. Instagram, Facebook, Twitter, Reddit etc.) to enjoy the different services provided by them. In between all the intentional use, social media is being leveraged in different ways, for purposes apart from the intended ones.

1.2 OBJECTIVE

Social media has been used to study the public perception and opinion in various studies. The effect of social media in shaping perception or opinion of a society about an array of issues and material things has also been studied and it has been found to be useful in measuring public opinion about an organization towards welfare schemes and most importantly, elections.

1.3 EXISTING SYSTEM

Due to the enormous influence of social media on different spheres of life and different strata of society, a rumor on social media can exist in different contexts and can affect different shades of human society. Rumors are often seen in political context but the effect of rumors on economy, human life, change in social norm and their role in flaring up of cultural aggression, and fanning of communal tensions cannot be ruled out.

1.4 LITERATURE SURVEY

Title: Breaking News Detection and Tracking in Twitter

Author: SwitPhuvipadawat ; Tsuyoshi Murata

Year: 2010

Description: Twitter has been used as one of the communication channels for spreading breaking news. We propose a method to collect, group, rank and track breaking news in Twitter. Since short length messages make similarity comparison difficult, we boost scores on proper nouns to improve the grouping results. Each group is ranked based on popularity and reliability factors. Current detection method is limited to facts part of messages. We developed an application called "Hotstream" based on the proposed method. Users can discover breaking news from the Twitter timeline. Each story is provided with the information of message originator, story development and activity chart. This provides a convenient way for people to follow breaking news and stay informed with real-time updates.

Title: Reading the riots on Twitter: Methodological innovation for the analysis of big data.

Author: Rob N Procter, Farida Vis.

Year: 2013

Description: For social scientists, the widespread adoption of social media presents both an opportunity and a challenge. Data that can shed light on people's habits, opinions and behaviour is available now on a scale never seen before, but this also means that it is impossible to analyse using conventional methodologies and tools. This article represents an experiment in applying a computationally assisted methodology to the analysis of a large corpus of tweets sent during the August 2011 riots in England.

Title: Finding and assessing social media information sources in the context of journalism

Author: Nicholas Diakopoulos, Munmun De Choudhury, Mor Naaman

Year: 2012

Description: Social media is already a fixture for reporting for many journalists, especially around breaking news events where non-professionals may already be on the scene to share an eyewitness report, photo, or video of the event. At the same time, the huge amount of content posted in conjunction with such events serves as a challenge to finding interesting and trustworthy sources in the din of the stream. In this paper we develop and investigate new methods for filtering and assessing the verity of sources found through social media by journalists. We take a human centered design approach to developing a system, SRSR ("Seriously Rapid Source Review"), informed by journalistic practices and knowledge of information production in events. We then used the system, together with a realistic reporting scenario, to evaluate the filtering and visual cue features that we developed. Our evaluation offers insights into social media information sourcing practices and challenges, and highlights the role technology can play in the solution.

Title: Supporting the use of user generated content in journalistic practice

Author: Peter Tolmie, Rob Procter, David William Randall.

Year: 2017

Description: Social media and user-generated content (UGC) are increasingly important features of journalistic work in a number of different ways. However, their use presents major challenges, not least because information posted on social media is not always reliable and therefore its veracity needs to be checked before it can be considered as fit for use in the reporting of news. We report on the results of a series of in-depth ethnographic studies of journalist work practices undertaken as part of the requirements gathering for a prototype of a social media verification 'dashboard' and its subsequent evaluation. We conclude with some reflections upon the broader implications of our findings for the design of tools to support journalistic work.

Title: Learning from the crowd: Collaborative filtering techniques for identifying on-the-ground Twitterers during mass disruptions

Author: Kate Starbird, Grace Muzny, LeysiaPalen.

Year: 2012

Description: Social media tools, including the micro blogging platform Twitter, have been appropriated during mass disruption events by those affected as well as the digitally-convergent crowd. Though tweets sent by those local to an event could be a resource both for responders and those affected, most Twitter activity during mass disruption events is generated by the remote crowd. Tweets from the remote crowd can be seen as noise that must be filtered, but another perspective considers crowd activity as a filtering and recommendation mechanism. This paper tests the hypothesis that crowd behavior can serve as a collaborative filter for identifying people tweeting from the ground during a mass disruption event. We test two models for classifying on-the-ground Twitterers, finding that machine learning techniques using a Support Vector Machine with asymmetric soft margins can be effective in identifying those likely to be on the ground during a mass disruption event.

Title: Real-Time Crisis Mapping of Natural Disasters Using Social Media

Author: Stuart E. Middleton ; Lee Middleton.

Year: 2013

Description: The proposed social media crisis mapping platform for natural disasters uses locations from gazetteer, street map, and volunteered geographic information (VGI) sources for areas at risk of disaster and matches them to geoparsed real-time tweet data streams. The authors use statistical analysis to generate real-time crisis maps. Geoparsing results are benchmarked against existing published work and evaluated across multilingual datasets. Two case studies compare five-day tweet crisis maps to official post-event impact assessment from the US National Geospatial Agency (NGA), compiled from verified satellite and aerial imagery sources.

1.5 PROPOSED SYSTEM

There have been various efforts in the field of rumor detection and mitigation. Many authors have used simple cue-based, network based, Psycho and social theory based approaches whereas many other have used machine learning approaches. Many other studies have incorporated different aspects and their methodology is an amalgamation of various techniques. There has also been a debate around which features are most important in detecting a rumor. This has led to a new approach of deep learning where feature selection is not required for the efficient performance of the framework. Here, we discuss various supervised, unsupervised and other machine learning approaches, as well as the deep learning based approaches in the field of rumor detection.

CHAPTER 2

PROJECT DESCRIPTION

2.1 GENERAL:

Social media is being used to disseminate news, create awareness, and track information about various societal, cultural and social issues Social media, as a tool of information dissemination, has found significance. Many people use it to disseminate their private information while as many others use it in a more professional way to convey ideas, opinions and information. The ease of sharing information, rapidity in disseminating it and real-time updates have made social media a powerful medium used by masses to convey information.

2.2 METHODOLOGIES

MODULE:

1. INPUT (KEYWORD)
2. PRE-PROCESSING
3. CLASSIFICATION

INPUT (KEYWORD):

Data in the form of raw tweets is acquired by using the Python library “tweepy” which provides a package for simple twitter streaming API . This API allows two modes of accessing tweets: SampleStream and FilterStream. SampleStream simply delivers a small, random sample of all the tweets streaming at a real time. FilterStream delivers tweet which match a certain criteria. It can filter the delivered tweets according to three criteria:

• Specific keyword to track/search for in the tweets

• Specific Twitter user according to their name

• Tweets originating from specific location(s)

PRE-PROCESSING

Twitter is a micro-blog where people generally write in a conversational style. Tweets are known to be very noisy for any text mining task as they contain a number of symbols that do not have any useful information and make further processing ineffective. Therefore this model includes effective pre-processing phase which removes meaningless symbols from tweets and hence, effective keywords can be extracted. The steps for pre-processing are as follows:

Remove username and re-tweet symbol:

Tweets often contain usernames beginning with the symbol '@'. Sometimes a tweet is also re-tweeted, which means a tweet by any user is shared again by other users and it contains the symbol RT. These user- names and retweet symbol do not contribute any significance to keyword extraction and act as noise. So, usernames and retweet symbols are removed.

Remove URLs:

Any URL links appearing in the tweets are re- moved as the model focuses only on the textual part of the tweet and URLs act as unnecessary noise while keywords are extracted.

Remove hash tags:

The Hash tag i.e. # before a word such as #KarnatakaWithCongress is removed to get KarnatakaWithCongress '.

Tokenization:

Each term in a tweet is treated as a token. To- kens are the basic constituents of a tweet/text. Let T be the set of tweets which is represented as T = { T 1 , T 2 , T 3 ,…, T i | i is the number of tweets}. Then each tweet in T is pre-processed and its terms are treated as tokens. Let t be the set of tokens rep- resented as t = { t 1 , t 2 , t 3 ,…, t k }. t includes tokens from all the tweets of T where the number of tokens in the set T is k .

Stop word removal:

A standard list of stop words is created and these stop words are then removed from the set.

Removal of unimportant tokens:

There are many tokens in the set which are comparatively less important and cannot be key- words. A mechanism is established to identify and remove these tokens so that they do not compete in the keyword ex- traction phase, making it more efficient. The tokens which oc- cur less than the Average Occurrence Frequency (AOF) are re- moved.

Part-of-speech Tagging:

POS-Tagging is the process of assigning a tag to each word in the sentence as to which grammatical part of speech that word belongs to, i.e. noun, verb, adjective, adverb, coordinating conjunction etc. For each tweet, we have features for counts of the number of verbs, adverbs, adjectives, nouns, and any other parts of speech.

CLASSIFICATION

Let's build a sentiment analysis of Twitter data to show how you might integrate an algorithm like this into your applications. We'll first start by choosing a topic, then we will gather tweets with that keyword and perform sentiment analyis on those tweets. We'll end up with an overall impression of whether people view the topic positively or not.

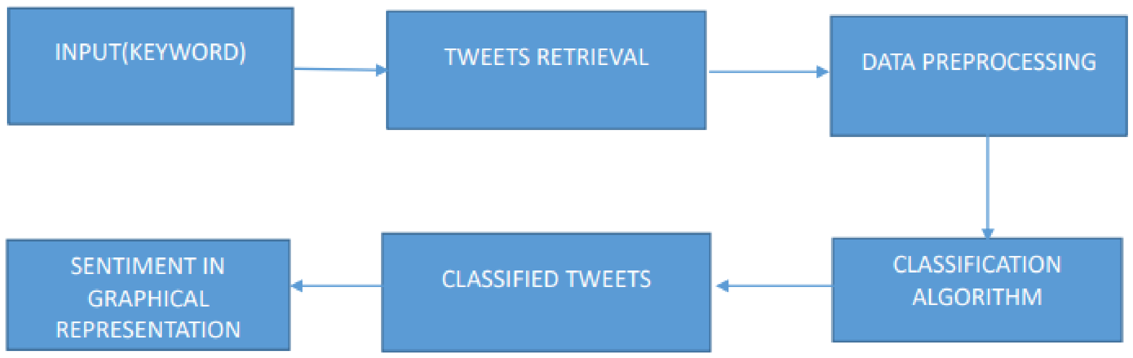


Fig 2.1: Module Diagram

2.3 TECHNIQUE OR ALGORITHMS

Random Forest

Introduction

Random forest is a supervised learning algorithm which is used for both classification as well as regression. But however, it is mainly used for classification problems. As we know that a forest is made up of trees and more trees means more robust forest. Similarly, random forest algorithm creates decision trees on data samples and then gets the prediction from each of them and finally selects the best solution by means of voting. It is an ensemble method which is better than a single decision tree because it reduces the over-fitting by averaging the result.

Working of Random Forest Algorithm

We can understand the working of Random Forest algorithm with the help of following steps −

- **Step 1** − First, start with the selection of random samples from a given dataset.
- **Step 2** − Next, this algorithm will construct a decision tree for every sample. Then it will get the prediction result from every decision tree.
- **Step 3** − In this step, voting will be performed for every predicted result.

- **Step 4** − At last, select the most voted prediction result as the final prediction result.

The following diagram will illustrate its working −

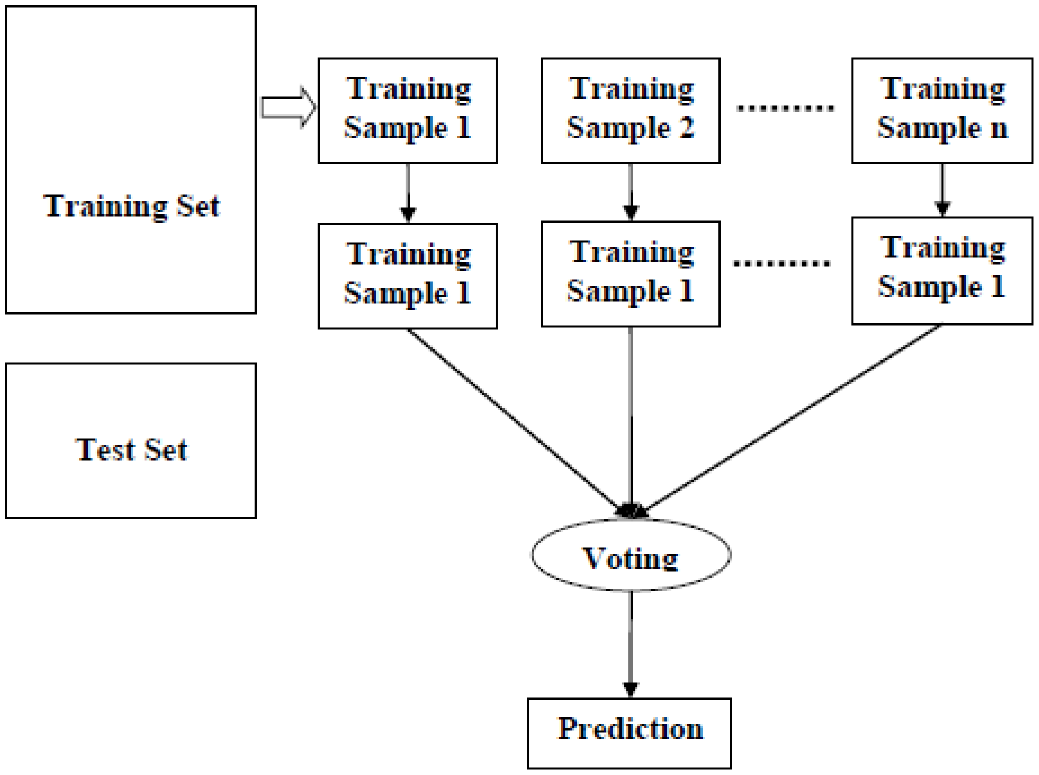


Fig 2.2: Working of Random Forest Algorithm

CHAPTER 3

REQUIREMENTS ENGINEERING

3.1 GENERAL

As Social media finds use as an information source by both public and professionals, it finds applications in various domains. The use of social media in various domain make it a potent carrier of the various ills along with it being a very useful and productive space. The use of social media in news gathering has been studied by various researchers and the regulative mechanisms have been suggested by them so that the goodness of social media can be leveraged to the best.

3.2 HARDWARE REQUIREMENTS

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point for the system design. It shouls what the system do and not how it should be implemented.

HARDWARE REQUIREMENTS

• PROCESSOR	:	DUAL CORE 2 DUOS.
• RAM	:	4GB DD RAM
• HARD DISK	:	250 GB

3.3 SOFTWARE REQUIREMENTS

The software requirements document is the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks and tracking the teams and tracking the team's progress throughout the development activity.

SOFTWARE REQUIREMENTS:

• FRONT END	:	PYTHON
• OPERATING SYSTEM	:	WINDOWS 7
• IDE	:	Spyder3

3.4 FUNCTIONAL REQUIREMENTS

A functional requirement defines a function of a software-system or its component. A function is described as a set of inputs, the behavior,Firstly, the system is the first that achieves the standard notion of semantic security for data confidentiality in attribute-based deduplication systems by resorting to the hybrid cloud architecture.

3.5 NON-FUNCTIONAL REQUIREMENTS

EFFICIENCY

Our multi-modal event tracking and evolution framework is suitable for multimedia documents from various social media platforms, which can not only effectively capture their multi-modal topics, but also obtain the evolutionary trends of social events and generate effective event summary details over time. Our proposed mmETM model can exploit the multi-modal property of social event, which can effectively model social media documents including long text with related images and learn the correlations between textual and visual modalities to separate the visual-representative topics and non-visual-representative topics.

CHAPTER 4

DESIGN ENGINEERING

4.1 GENERAL

Design Engineering deals with the various UML [Unified Modelling language] diagrams for the implementation of project. Design is a meaningful engineering representation of a thing that is to be built. Software design is a process through which the requirements are translated into representation of the software. Design is the place where quality is rendered in software engineering. Design is the means to accurately translate customer requirements into finished product.

UML Diagrams

Use Case Diagram

To model a system, the most important aspect is to capture the dynamic behavior. To clarify a bit in details, dynamic behavior means the behavior of the system when it is running /operating.

So only static behavior is not sufficient to model a system rather dynamic behavior is more important than static behavior. In UML there are five diagrams available to model dynamic nature and use case diagram is one of them. Now as we have to discuss that the use case diagram is dynamic in nature there should be some internal or external factors for making the interaction.

These internal and external agents are known as actors. So use case diagrams are consists of actors, use cases and their relationships. The diagram is used to model the system/subsystem of an application. A single use case diagram captures a particular functionality of a system. So to model the entire system numbers of use case diagrams are used.

Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. So when a system is analyzed to gather its functionalities use cases are prepared and actors are identified.

In brief, the purposes of use case diagrams can be as follows:

a. Used to gather requirements of a system.

b. Used to get an outside view of a system.

c. Identify external and internal factors influencing the system.

d. Show the interacting among the requirements are actors.

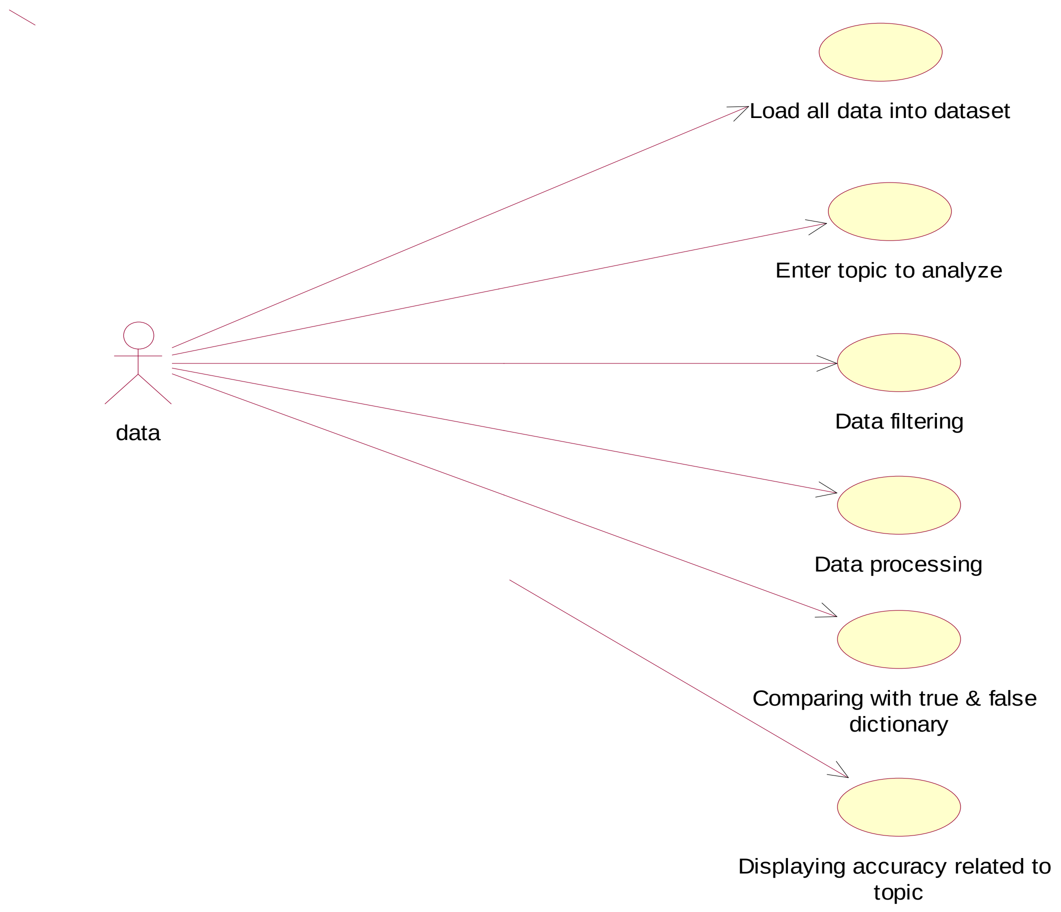

Load all data into dataset
Enter topic to analyze
data
Data filtering
Data processing
Comparing with true & false dictionary
Displaying accuracy related to topic

Figure 4.1 : Use case Diagram for entire application functionality

CLASS DIAGRAM FOR CALCULATE ACCURACY

A class diagram in the UML is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, and the relationships between the classes. Private visibility hides information from anything outside the class partition. Public visibility allows all other classes to view the marked information. Protected visibility allows child classes to access information they inherited from a parent class.

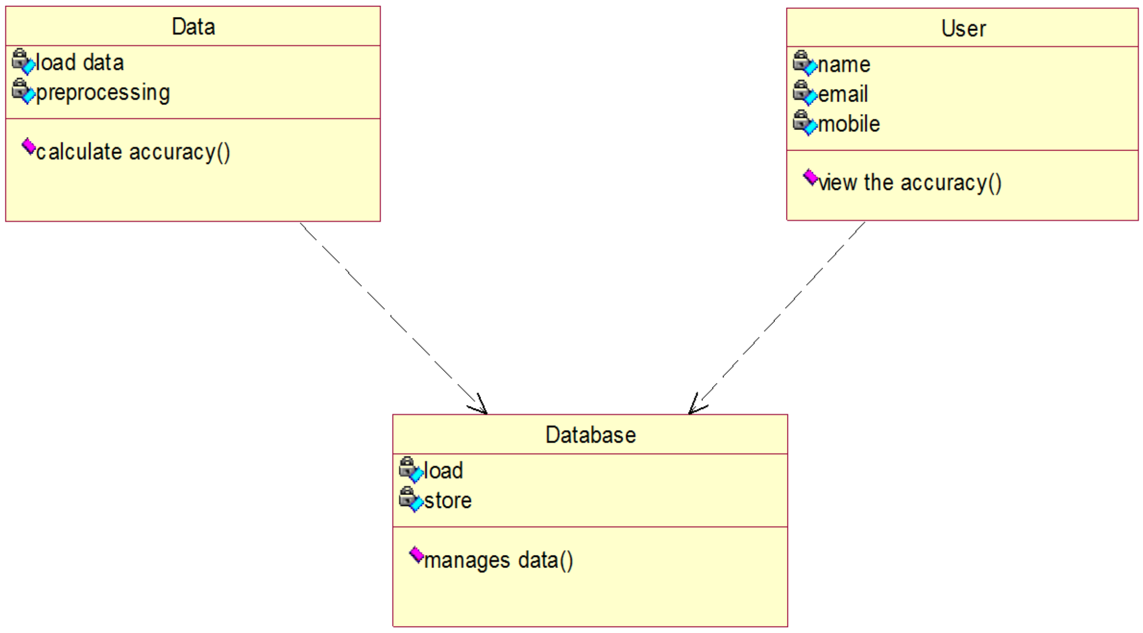


Fig 4.2 : Class Diagram

COMPONENT DIAGRAM FOR TWITTER DATA

Components are wired together by using an assembly connector to connect the required interface of one component with the provided interface of another component. This illustrates the service consumer - service provider relationship between the two components.

An assembly connector is a "connector between two components that defines that one component provides the services that another component requires. An assembly connector is a connector that is defined from a required interface or port to a provided interface or port."

When using a component diagram to show the internal structure of a component, the provided and required interfaces of the encompassing component can delegate to the corresponding interfaces of the contained components.

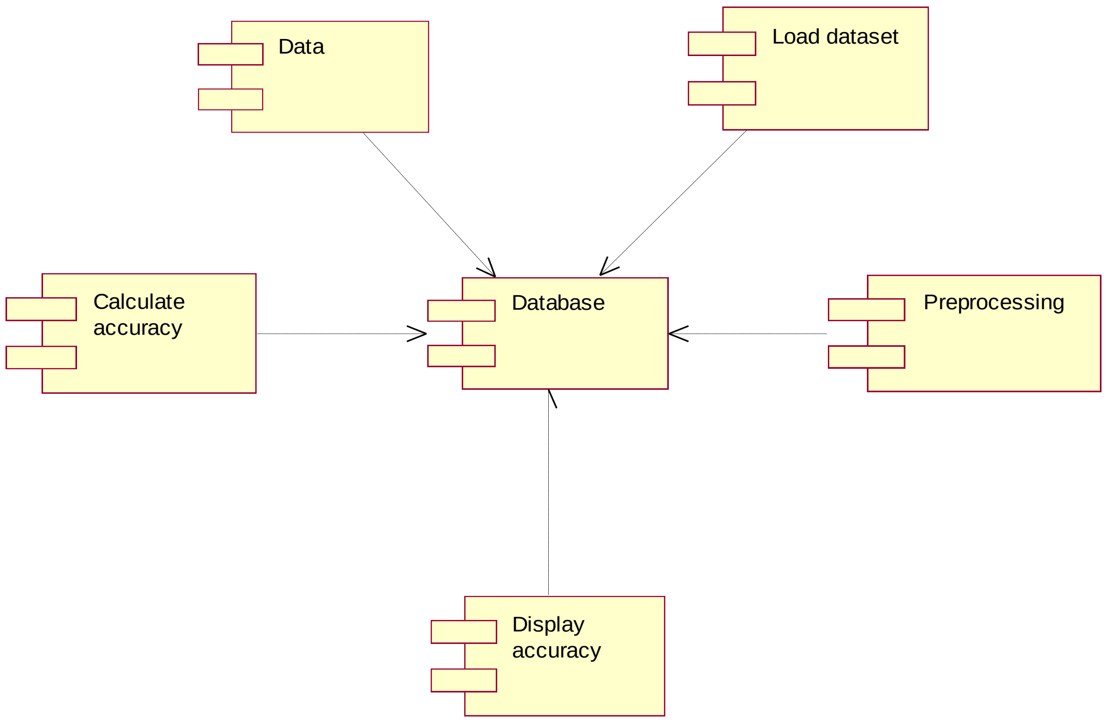


Fig 4.3 : Component Diagram

ACTIVITY DIAGRAM FOR DISPLAY ACCURACY

Activity diagram are a loosely defined diagram to show workflows of stepwise activities and actions, with support for choice, iteration and concurrency. UML, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. UML activity diagrams could potentially model the internal logic of a complex operation. In many ways UML activity diagrams are the object-oriented equivalent of flow charts and data flow diagrams (DFDs) from structural development.

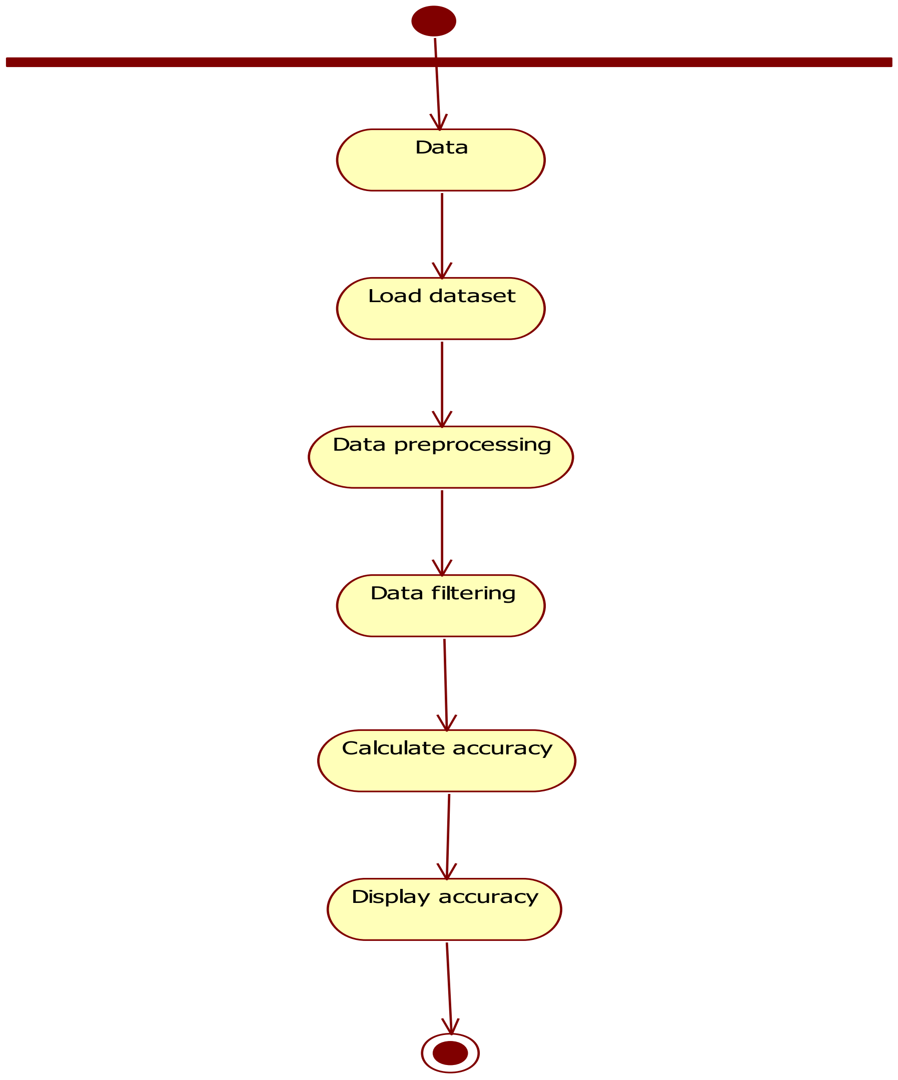


Fig 4.4: Activity Diagram

DEPLOYMENT DIAGRAM FOR DATASET

Deployment diagram is a structure diagram which shows architecture of the system as deployment (distribution) of software artifacts to deployment targets. Artifacts represent concrete elements in the physical world that are the result of a development process.

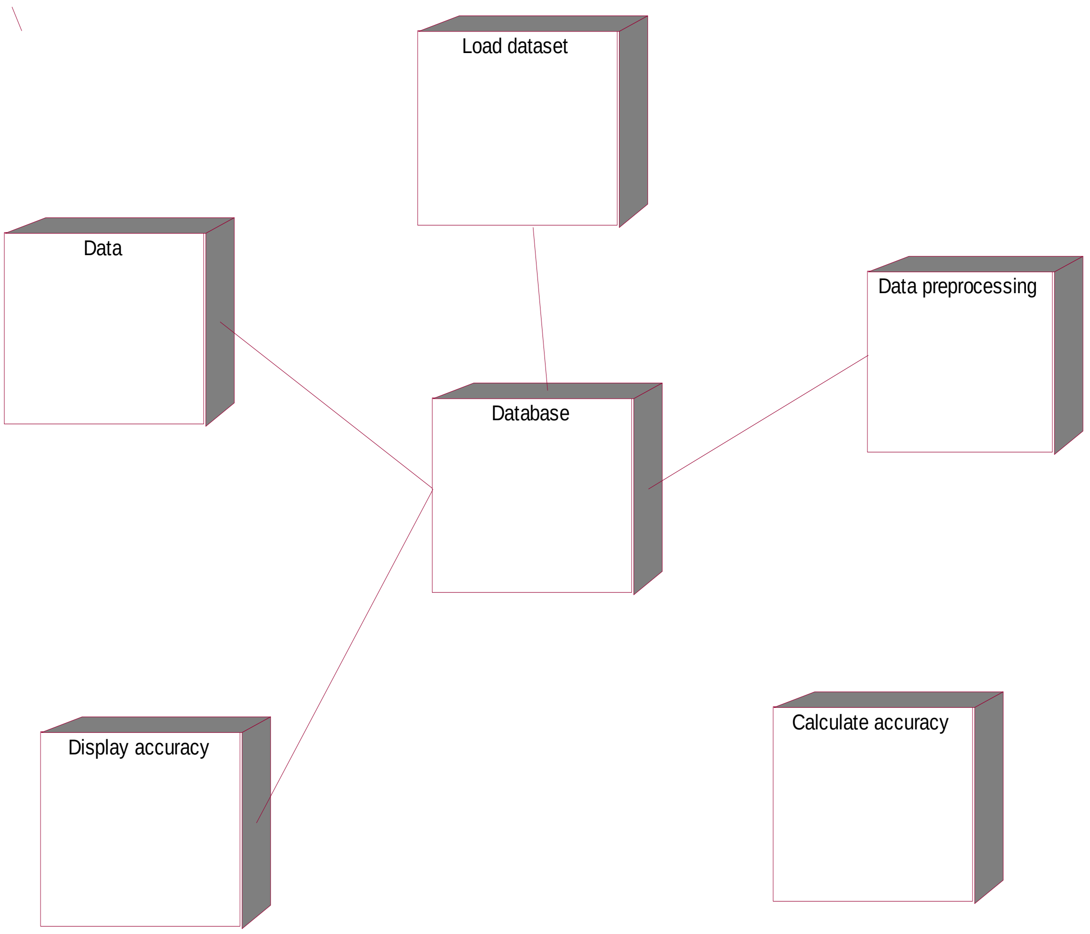


Fig 4.5: Deployment Diagram

SYSTEM ARCHITECTURE:

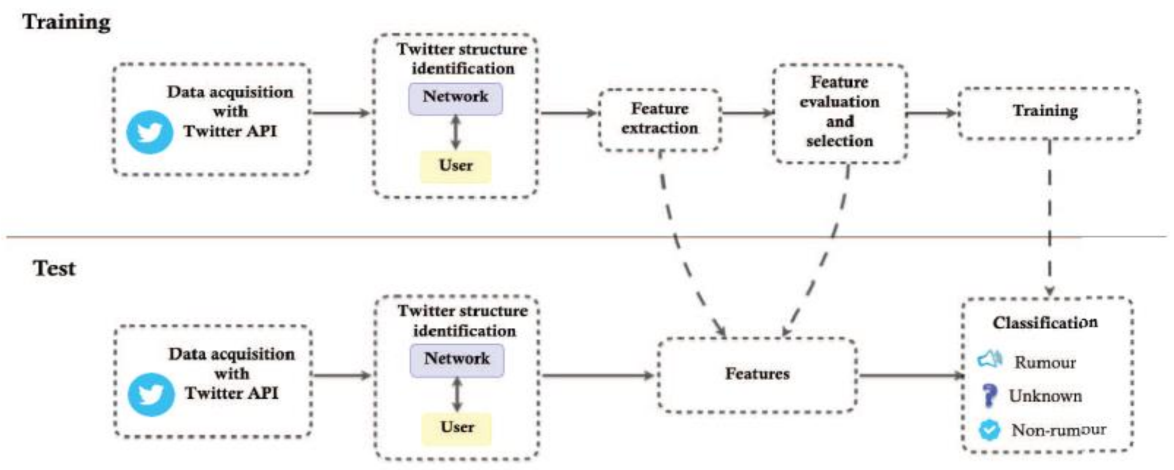

Training
Data acquisition with Twitter API
Twitter structure identification
Network
User
Feature extraction
Feature evaluation and selection
Training
Test
Data acquisition with Twitter API
Twitter structure identification
Network
User
Features
Classification
Rumour
Unknown
Non-rumour

Fig 4.6: Architecture Diagram

EXPLANATION:

A system architecture or systems architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system. A system architecture can consist of system components and the sub-systems developed, that will work together to implement the overall system. There have been efforts to formalize languages to describe system architecture, collectively these are called architecture description languages (ADLs).

CHAPTER 5

DEVELOPMENT TOOLS

Python

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

History of Python

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

Importance of Python

- **Python is Interpreted** − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- **Python is Interactive** − You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- **Python is Object-Oriented** − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.

- **Python is a Beginner's Language** − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

Features of Python

- **Easy-to-learn** − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- **Easy-to-read** − Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain** − Python's source code is fairly easy-to-maintain.
- **A broad standard library** − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode** − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- **Portable** − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- **Extendable** − You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- **Databases** − Python provides interfaces to all major commercial databases.
- **GUI Programming** − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- **Scalable** − Python provides a better structure and support for large programs than shell scripting.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below −

- It supports functional and structured programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to byte-code for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking.
- IT supports automatic garbage collection.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

Libraries used in python:

- numpy - mainly useful for its N-dimensional array objects.
- pandas - Python data analysis library, including structures such as dataframes.
- matplotlib - 2D plotting library producing publication quality figures.
- scikit-learn - the machine learning algorithms used for data analysis and data mining tasks.

Figure : NumPy, Pandas, Matplotlib, Scikit-learn

CHAPTER 6

IMPLEMENTATION

6.1 GENERAL

Coding:

```
import pandas as pd

import numpy as np



# Pre-processing / feature extraction

import nltk

from datetime import datetime as dt

import string

from nltk.tokenize import TweetTokenizer

from nltk.corpus import stopwords

from nltk.stem.porter import PorterStemmer

from sklearn.feature_extraction.text import CountVectorizer



# Models

from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier

from sklearn.neighbors import KNeighborsClassifier



# Testing

from sklearn.model_selection import cross_val_score



import warnings

warnings.simplefilter('ignore')



tokenizer = TweetTokenizer()

stemmer = PorterStemmer()

stopws = set(stopwords.words('english'))

vectorizer = CountVectorizer(analyzer = 'word')



def clean_data(data):

data.creationDate = pd.to_datetime(data.creationDate)

    data['creationDay'] = data.creationDate.apply(lambda x: x.weekday())

    data['creationMonth'] = data.creationDate.apply(lambda x: x.month)

data.text = data.text.apply(tokenizer.tokenize)
data.text = data.text.apply(lambda x: [w for w in x if w not in stopws and w.lower() not in
string.punctuation])

data.text = data.text.apply(lambda x: [stemmer.stem(w) for w in x])

    data['text_j'] = data.text.apply(lambda x: ' '.join(x))

    data['response'] = data.text_j.apply(lambda x: x.startswith(('@', ' @')))

    return data



tweets = pd.read_csv('tweets_unfiltered.csv', delimiter=';')

tweets = clean_data(tweets)



attributes = tweets[['retweetCount', 'favoriteCount', 'creationDay', 'creationMonth',
'response']]

attributes = pd.concat((attributes,
pd.DataFrame(vectorizer.fit_transform(tweets.text_j).toarray())), axis=1)

classes = tweets.rumor

print(classes)

rf = RandomForestClassifier(n_estimators = 100)

dt = DecisionTreeClassifier()



# Train the classifiers
```

```
rf.fit(attributes, classes)

dt.fit(attributes, classes)



def predict_rf(text, rt, fav, date):

    data = pd.DataFrame([[text, date, rt, fav]], columns=['text','creationDate',
'retweetCount', 'favoriteCount'])

    data = clean_data(data)

    attributes = data[['retweetCount', 'favoriteCount', 'creationDay', 'creationMonth',
'response']]

    attributes = pd.concat((attributes,
pd.DataFrame(vectorizer.transform(data.text_j).toarray())), axis=1)

    print(attributes)

    return str(rf.predict(attributes)[0])



def predict_dt(text, rt, fav, date):

    data = pd.DataFrame([[text, date, rt, fav]], columns=['text','creationDate',
'retweetCount', 'favoriteCount'])

    data = clean_data(data)

    attributes = data[['retweetCount', 'favoriteCount', 'creationDay', 'creationMonth',
'response']]
```

```
    attributes = pd.concat((attributes,
pd.DataFrame(vectorizer.transform(data.text_j).toarray())), axis=1)

    return str(rf.predict(attributes)[0])




def test(X, y, f, k=10):

    """

    Runs cross-validation for multiple models.



:param X: input data

:param y: input data

:param f: scoring function

:param k: number of folds

:return: data frame with scoring mean and std for each model

    """



    models = {

        'Random Forest': RandomForestClassifier(n_estimators = 100),

        'Decision Tree': DecisionTreeClassifier(),
        'k-NN': KNeighborsClassifier()
    }


    df = pd.DataFrame(columns=('mean', 'std'))
    for n, m in models.items():
np.random.seed(121)
        fs = cross_val_score(m, X, y, scoring=f, cv=k)
print("%s\%s: %0.2f (+/- %0.2f)" % (n, f, fs.mean(), fs.std() * 2))
df.loc[n] = fs.mean(), fs.std()
    return df


res = test(attributes, classes, 'accuracy')
print(res)


import os
from flask import Flask, request
from ml import predict_rf, predict_dt
app = Flask(__name__)
```

```
@app.route('/dt/', methods=['POST'])

def predict_tweet_dt():

        text = request.form['text']

        rt = request.form['rt']

        fav = request.form['fav']

        date = request.form['date']

        return predict_dt(text, rt, fav, date)



@app.route('/rf/', methods=['POST'])

def predict_tweet_rf():

        text = request.form['text']

        rt = request.form['rt']

        fav = request.form['fav']

        date = request.form['date']



        return predict_rf(text, rt, fav, date)



if __name__ == "__main__":

        app.run()
```

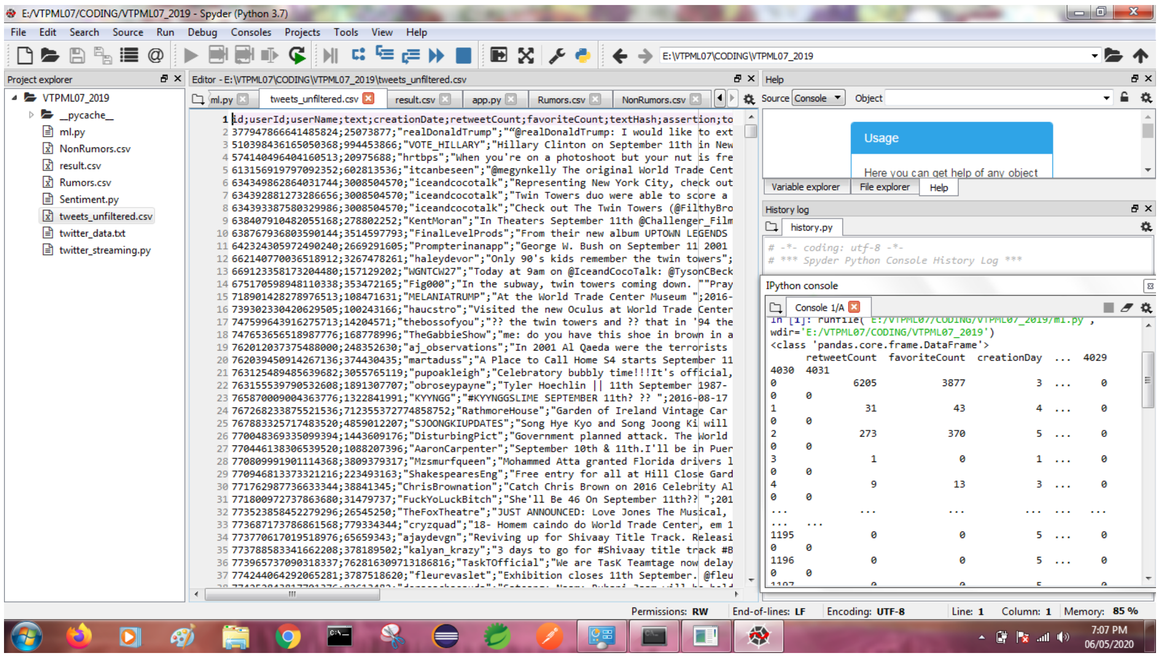
CHAPTER 7

SNAPSHOTS

General:

This project is implements like application using python and the Server process is maintained using the SOCKET & SERVERSOCKET and the Design part is played by Cascading Style Sheet.

SNAPSHOTS

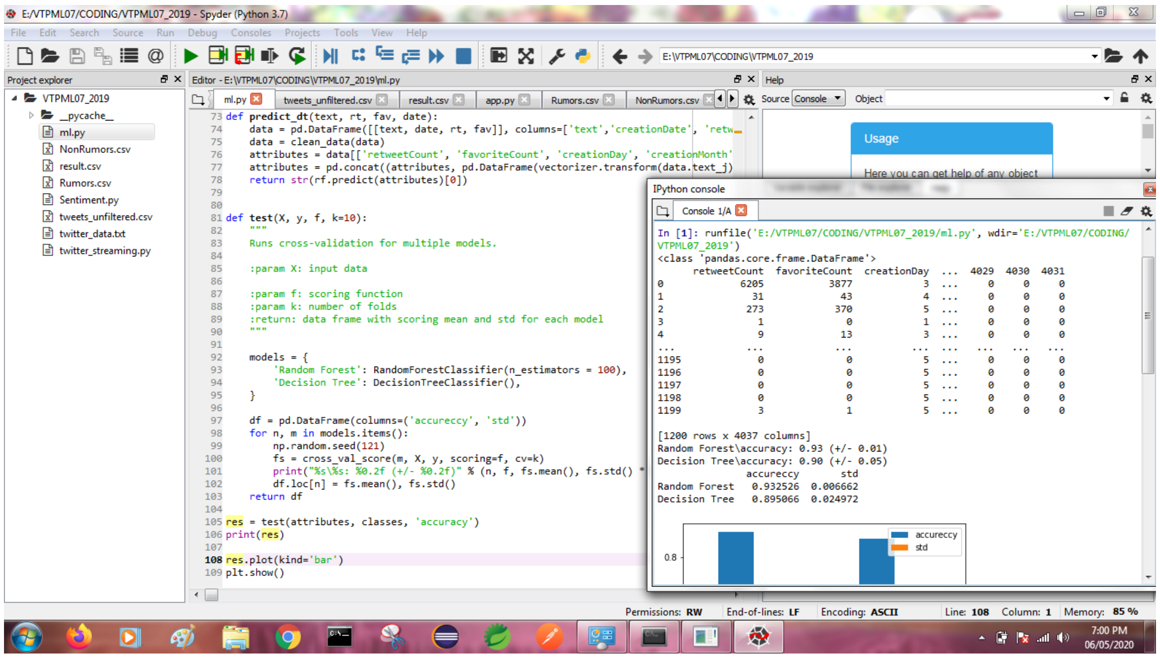


```
id;userId;userName;text;creationDate;retweetCount;favoriteCount;textHash;assertion;to
377947866641485824;25073877;"realDonaldTrump";"“@realDonaldTrump: I would like to ext
510398436165050368;994453866;"VOTE_HILLARY";"Hillary Clinton on September 11th in New
574140496404160513;20975688;"hrtbps";"When you're on a photoshoot but your nut is fre
613156919797092352;602813536;"itcanbeseen";"@megynkelly The original World Trade Cent
634349862864031744;3008504570;"iceandcocotalk";"Representing New York City, check out
634392881273286656;3008504570;"iceandcocotalk";"Twin Towers duo were able to score a
634393387580329986;3008504570;"iceandcocotalk";"Check out The Twin Towers (@FilthyBro
638407910482055168;278802252;"KentMoran";"In Theaters September 11th @Challenger_Film
638767936803590144;3514597793;"FinalLevelProds";"From their new album UPTOWN LEGENDS
642324305972490240;2669291605;"Prompterinanapp";"George W. Bush on September 11 2001
662140770036518912;3267478261;"haleydevor";"Only 90's kids remember the twin towers";
669123358173204480;157129202;"WGNTCW27";"Today at 9am on @IceandCocoTalk: @TysonCBeck
675170598948110338;353472165;"Fig000";"In the subway, twin towers coming down. ""Pray
718901428278976513;108471631;"MELANIATRUMP";"At the World Trade Center Museum ";2016-
739302330420629505;100243166;"haucstro";"Visited the new Oculus at World Trade Center
747599643916275713;14204571;"thebossofyou";"?? the twin towers and ?? that in '94 the
747653656518987776;168778996;"TheGabbieShow";"me: do you have this shoe in brown in a
762012037375488000;248352630;"aj_observations";"In 2001 Al Qaeda were the terrorists
762039450914267136;374430435;"martaduss";"A Place to Call Home S4 starts September 11
763125489485639682;3055765119;"pupoakleigh";"Celebratory bubbly time!!!It's official,
763155539790532608;1891307707;"obroseypayne";"Tyler Hoechlin || 11th September 1987-
765870009004363776;1322841991;"KYYNGG";"#KYYNGGSLIME SEPTEMBER 11th? ?? ";2016-08-17
767268233875521536;712355372774858752;"RathmoreHouse";"Garden of Ireland Vintage Car
767883325717483520;4859012207;"SJOONGKIUPDATES";"Song Hye Kyo and Song Joong Ki will
770048369335099394;1443609176;"DisturbingPict";"Government planned attack. The World
770446138306539520;1088207396;"AaronCarpenter";"September 10th & 11th.I'll be in Puer
770809991901114368;3809379317;"Mzsmurfqueen";"Mohammed Atta granted Florida drivers l
770946813373321216;223493163;"ShakespearesEng";"Free entry for all at Hill Close Gard
771762987736633344;38841345;"ChrisBrownation";"Catch Chris Brown on 2016 Celebrity Al
771800972737863680;31479737;"FuckYoLuckBitch";"She'll Be 46 On September 11th?? ";201
773523858452279296;26545250;"TheFoxTheatre";"JUST ANNOUNCED: Love Jones The Musical,
773687173786861568;779334344;"cryzquad";"18- Homem caindo do World Trade Center, em 1
773770617019518976;65659343;"ajaydevgn";"Reviving up for Shivaay Title Track. Releasi
773788583341662208;378189502;"kalyan_krazy";"3 days to go for #Shivaay title track #B
773965737090318337;762816309713186816;"TaskTOfficial";"We are TasK Teamtage now delay
774244064292065281;3787518620;"fleurevaslet";"Exhibition closes 11th September. @fleu
```

```
# -*- coding: utf-8 -*-
# *** Spyder Python Console History Log ***
```

```
In [1]: runfile('E:/VTPML07/CODING/VTPML07_2019/ml.py',
wdir='E:/VTPML07/CODING/VTPML07_2019')
<class 'pandas.core.frame.DataFrame'>
      retweetCount  favoriteCount  creationDay  ...  4029
4030  4031
0             6205           3877            3  ...     0
0     0
1               31             43            4  ...     0
0     0
2              273            370            5  ...     0
0     0
3                1              0            1  ...     0
0     0
4                9             13            3  ...     0
0     0
...            ...            ...          ...  ...   ...
...   ...
1195             0              0            5  ...     0
0     0
1196             0              0            5  ...     0
0     0
```

Fig: Unfiltered Twitter Data

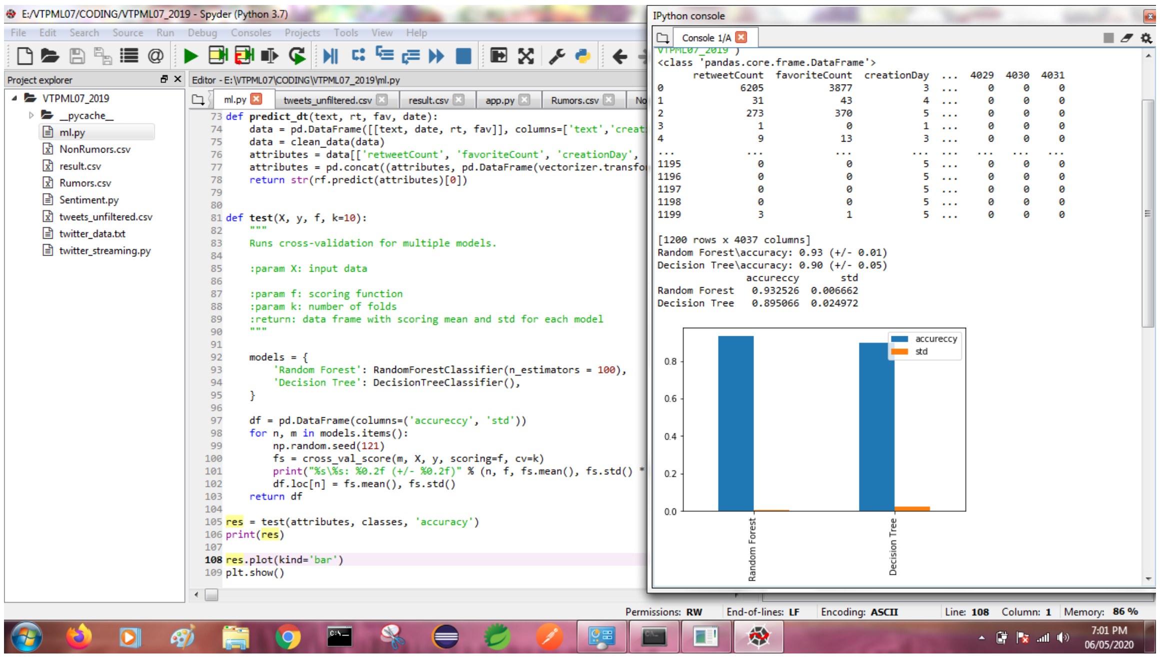


```
def predict_dt(text, rt, fav, date):
    data = pd.DataFrame([[text, date, rt, fav]], columns=['text','creationDate', 'retw
    data = clean_data(data)
    attributes = data[['retweetCount', 'favoriteCount', 'creationDay', 'creationMonth'
    attributes = pd.concat((attributes, pd.DataFrame(vectorizer.transform(data.text_j)
    return str(rf.predict(attributes)[0])


def test(X, y, f, k=10):
    """
    Runs cross-validation for multiple models.

    :param X: input data

    :param f: scoring function
    :param k: number of folds
    :return: data frame with scoring mean and std for each model
    """

    models = {
        'Random Forest': RandomForestClassifier(n_estimators = 100),
        'Decision Tree': DecisionTreeClassifier(),
    }

    df = pd.DataFrame(columns=('accureccy', 'std'))
    for n, m in models.items():
        np.random.seed(121)
        fs = cross_val_score(m, X, y, scoring=f, cv=k)
        print("%s\%s: %0.2f (+/- %0.2f)" % (n, f, fs.mean(), fs.std() *
        df.loc[n] = fs.mean(), fs.std()
    return df

res = test(attributes, classes, 'accuracy')
print(res)

res.plot(kind='bar')
plt.show()
```

```
In [1]: runfile('E:/VTPML07/CODING/VTPML07_2019/ml.py', wdir='E:/VTPML07/CODING/
VTPML07_2019')
<class 'pandas.core.frame.DataFrame'>
      retweetCount  favoriteCount  creationDay  ...  4029  4030  4031
0             6205           3877            3  ...     0     0     0
1               31             43            4  ...     0     0     0
2              273            370            5  ...     0     0     0
3                1              0            1  ...     0     0     0
4                9             13            3  ...     0     0     0
...            ...            ...          ...  ...   ...   ...   ...
1195             0              0            5  ...     0     0     0
1196             0              0            5  ...     0     0     0
1197             0              0            5  ...     0     0     0
1198             0              0            5  ...     0     0     0
1199             3              1            5  ...     0     0     0

[1200 rows x 4037 columns]
Random Forest\accuracy: 0.93 (+/- 0.01)
Decision Tree\accuracy: 0.90 (+/- 0.05)
               accureccy       std
Random Forest   0.932526  0.006662
Decision Tree   0.895066  0.024972
```

E:/VTPML07/CODING/VTPML07_2019 - Spyder (Python 3.7)
File Edit Search Source Run Debug Consoles Projects Tools View Help
Project explorer
VTPML07_2019
__pycache__
ml.py
NonRumors.csv
result.csv
Rumors.csv
Sentiment.py
tweets_unfiltered.csv
twitter_data.txt
twitter_streaming.py
Editor - E:\VTPML07\CODING\VTPML07_2019\ml.py
Help
Usage
Here you can get help of any object
IPython console
Console 1/A
accureccy
std
0.8
Permissions: RW End-of-lines: LF Encoding: ASCII Line: 108 Column: 1 Memory: 85 %
7:00 PM
06/05/2020

Fig: code

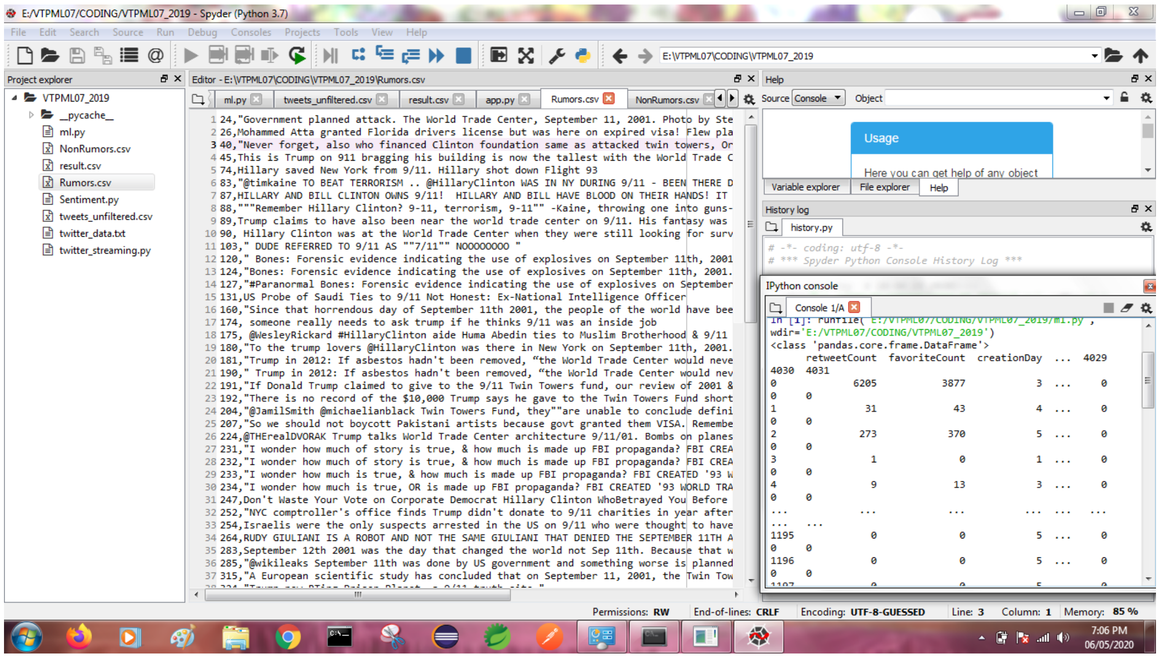


```
def predict_dt(text, rt, fav, date):
    data = pd.DataFrame([[text, date, rt, fav]], columns=['text','creat
    data = clean_data(data)
    attributes = data[['retweetCount', 'favoriteCount', 'creationDay',
    attributes = pd.concat((attributes, pd.DataFrame(vectorizer.transfo
    return str(rf.predict(attributes)[0])


def test(X, y, f, k=10):
    """
    Runs cross-validation for multiple models.

    :param X: input data

    :param f: scoring function
    :param k: number of folds
    :return: data frame with scoring mean and std for each model
    """

    models = {
        'Random Forest': RandomForestClassifier(n_estimators = 100),
        'Decision Tree': DecisionTreeClassifier(),
    }

    df = pd.DataFrame(columns=('accureccy', 'std'))
    for n, m in models.items():
        np.random.seed(121)
        fs = cross_val_score(m, X, y, scoring=f, cv=k)
        print("%s\%s: %0.2f (+/- %0.2f)" % (n, f, fs.mean(), fs.std() *
        df.loc[n] = fs.mean(), fs.std()
    return df

res = test(attributes, classes, 'accuracy')
print(res)

res.plot(kind='bar')
plt.show()
```

```
<class 'pandas.core.frame.DataFrame'>
      retweetCount  favoriteCount  creationDay  ...  4029  4030  4031
0             6205           3877            3  ...     0     0     0
1               31             43            4  ...     0     0     0
2              273            370            5  ...     0     0     0
3                1              0            1  ...     0     0     0
4                9             13            3  ...     0     0     0
...            ...            ...          ...  ...   ...   ...   ...
1195             0              0            5  ...     0     0     0
1196             0              0            5  ...     0     0     0
1197             0              0            5  ...     0     0     0
1198             0              0            5  ...     0     0     0
1199             3              1            5  ...     0     0     0

[1200 rows x 4037 columns]
Random Forest\accuracy: 0.93 (+/- 0.01)
Decision Tree\accuracy: 0.90 (+/- 0.05)
               accureccy       std
Random Forest   0.932526  0.006662
Decision Tree   0.895066  0.024972
```

accureccy
std
0.0
0.2
0.4
0.6
0.8
Random Forest
Decision Tree
Permissions: RW End-of-lines: LF Encoding: ASCII Line: 108 Column: 1 Memory: 86 %
7:01 PM
06/05/2020

Fig: OUTPUT

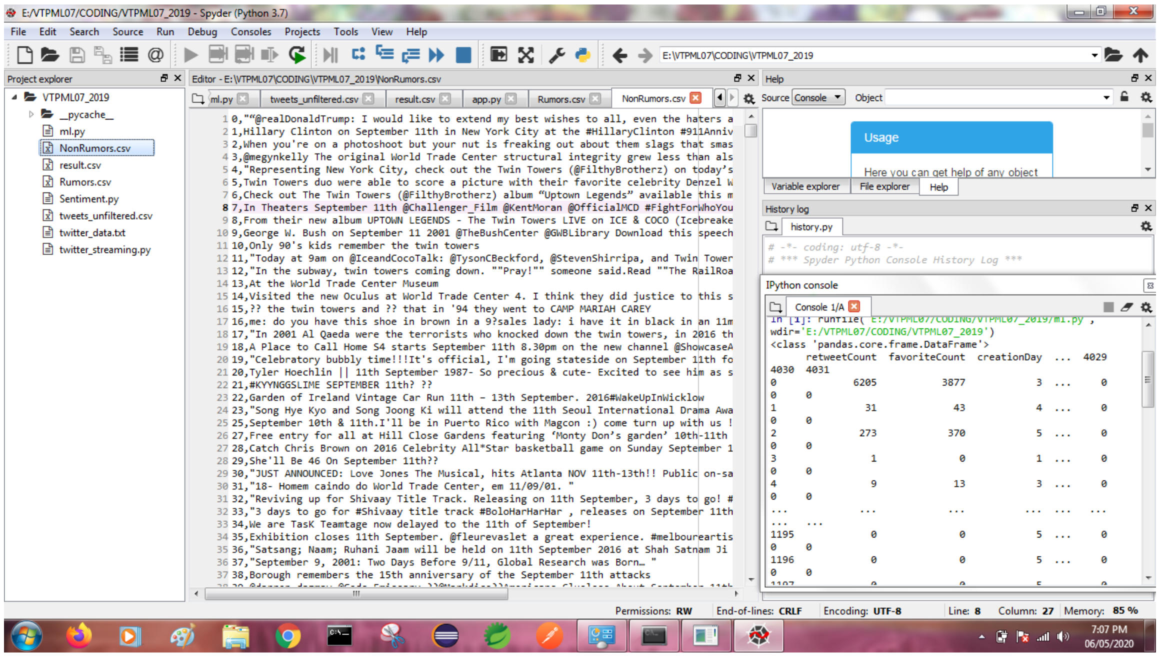

E:/VTPML07/CODING/VTPML07_2019 - Spyder (Python 3.7)
File Edit Search Source Run Debug Consoles Projects Tools View Help
Project explorer
VTPML07_2019
__pycache__
ml.py
NonRumors.csv
result.csv
Rumors.csv
Sentiment.py
tweets_unfiltered.csv
twitter_data.txt
twitter_streaming.py
Editor - E:\VTPML07\CODING\VTPML07_2019\Rumors.csv
ml.py
tweets_unfiltered.csv
result.csv
app.py
Rumors.csv
NonRumors.csv
Help
Source Console Object
Usage
Here you can get help of any object
Variable explorer File explorer Help
History log
history.py
IPython console
Console 1/A
Permissions: RW End-of-lines: CRLF Encoding: UTF-8-GUESSED Line: 3 Column: 1 Memory: 85 %
7:06 PM
06/05/2020

```
24,"Government planned attack. The World Trade Center, September 11, 2001. Photo by Ste
26,Mohammed Atta granted Florida drivers license but was here on expired visa! Flew pla
40,"Never forget, also who financed Clinton foundation same as attacked twin towers, Or
45,This is Trump on 911 bragging his building is now the tallest with the World Trade C
74,Hillary saved New York from 9/11. Hillary shot down Flight 93
83,"@timkaine TO BEAT TERRORISM .. @HillaryClinton WAS IN NY DURING 9/11 - BEEN THERE D
87,HILLARY AND BILL CLINTON OWNS 9/11!  HILLARY AND BILL HAVE BLOOD ON THEIR HANDS! IT
88,"""Remember Hillary Clinton? 9-11, terrorism, 9-11"" -Kaine, throwing one into guns-
89,Trump claims to have also been near the world trade center on 9/11. His fantasy was
90, Hillary Clinton was at the World Trade Center when they were still looking for surv
103," DUDE REFERRED TO 9/11 AS ""7/11"" NOOOOOOOO "
120," Bones: Forensic evidence indicating the use of explosives on September 11th, 2001
124,"Bones: Forensic evidence indicating the use of explosives on September 11th, 2001.
127,"#Paranormal Bones: Forensic evidence indicating the use of explosives on September
131,US Probe of Saudi Ties to 9/11 Not Honest: Ex-National Intelligence Officer
160,"Since that horrendous day of September 11th 2001, the people of the world have bee
174, someone really needs to ask trump if he thinks 9/11 was an inside job
175, @WesleyRickard #HillaryClinton aide Huma Abedin ties to Muslim Brotherhood & 9/11
180,"To the trump lovers @HillaryClinton was there in New York on September 11th, 2001.
181,"Trump in 2012: If asbestos hadn't been removed, “the World Trade Center would neve
190," Trump in 2012: If asbestos hadn't been removed, “the World Trade Center would nev
191,"If Donald Trump claimed to give to the 9/11 Twin Towers fund, our review of 2001 &
192,"There is no record of the $10,000 Trump says he gave to the Twin Towers Fund short
204,"@JamilSmith @michaelianblack Twin Towers Fund, they""are unable to conclude defini
207,"So we should not boycott Pakistani artists because govt granted them VISA. Remembe
224,@THErealDVORAK Trump talks World Trade Center architecture 9/11/01. Bombs on planes
231,"I wonder how much of story is true, & how much is made up FBI propaganda? FBI CREA
232,"I wonder how much of story is true, & how much is made up FBI propaganda? FBI CREA
233,"I wonder how much is true, & how much is made up FBI propaganda? FBI CREATED '93 W
234,"I wonder how much is true, OR is made up FBI propaganda? FBI CREATED '93 WORLD TRA
247,Don't Waste Your Vote on Corporate Democrat Hillary Clinton WhoBetrayed You Before
252,"NYC comptroller's office finds Trump didn't donate to 9/11 charities in year after
254,Israelis were the only suspects arrested in the US on 9/11 who were thought to have
264,RUDY GIULIANI IS A ROBOT AND NOT THE SAME GIULIANI THAT DENIED THE SEPTEMBER 11TH A
283,September 12th 2001 was the day that changed the world not Sep 11th. Because that w
285,"@wikileaks September 11th was done by US government and something worse is planned
315,"A European scientific study has concluded that on September 11, 2001, the Twin Tow
```

```
# -*- coding: utf-8 -*-
# *** Spyder Python Console History Log ***
```

```
In [1]: runfile('E:/VTPML07/CODING/VTPML07_2019/ml.py',
wdir='E:/VTPML07/CODING/VTPML07_2019')
<class 'pandas.core.frame.DataFrame'>
      retweetCount  favoriteCount  creationDay  ...  4029
4030  4031
0             6205           3877            3  ...     0
0     0
1               31             43            4  ...     0
0     0
2              273            370            5  ...     0
0     0
3                1              0            1  ...     0
0     0
4                9             13            3  ...     0
0     0
...            ...            ...          ...  ...   ...
...   ...
1195             0              0            5  ...     0
0     0
1196             0              0            5  ...     0
0     0
```

Fig: Rumors generated in Rumors.csv file


E:/VTPML07/CODING/VTPML07_2019 - Spyder (Python 3.7)
File Edit Search Source Run Debug Consoles Projects Tools View Help
Project explorer
VTPML07_2019
__pycache__
ml.py
NonRumors.csv
result.csv
Rumors.csv
Sentiment.py
tweets_unfiltered.csv
twitter_data.txt
twitter_streaming.py
Editor - E:\VTPML07\CODING\VTPML07_2019\NonRumors.csv
ml.py
tweets_unfiltered.csv
result.csv
app.py
Rumors.csv
NonRumors.csv
Help
Source Console Object
Usage
Here you can get help of any object
Variable explorer File explorer Help
History log
history.py
IPython console
Console 1/A
Permissions: RW End-of-lines: CRLF Encoding: UTF-8 Line: 8 Column: 27 Memory: 85 %
7:07 PM
06/05/2020

```
0,"“@realDonaldTrump: I would like to extend my best wishes to all, even the haters a
1,Hillary Clinton on September 11th in New York City at the #HillaryClinton #911Anniv
2,When you're on a photoshoot but your nut is freaking out about them slags that smas
3,@megynkelly The original World Trade Center structural integrity grew less than als
4,"Representing New York City, check out the Twin Towers (@FilthyBrotherz) on today’s
5,Twin Towers duo were able to score a picture with their favorite celebrity Denzel W
6,Check out The Twin Towers (@FilthyBrotherz) album “Uptown Legends” available this m
7,In Theaters September 11th @Challenger_Film @KentMoran @OfficialMCD #FightForWhoYou
8,From their new album UPTOWN LEGENDS - The Twin Towers LIVE on ICE & COCO (Icebreake
9,George W. Bush on September 11 2001 @TheBushCenter @GWBLibrary Download this speech
10,Only 90's kids remember the twin towers
11,"Today at 9am on @IceandCocoTalk: @TysonCBeckford, @StevenShirripa, and Twin Tower
12,"In the subway, twin towers coming down. ""Pray!"" someone said.Read ""The RailRoa
13,At the World Trade Center Museum
14,Visited the new Oculus at World Trade Center 4. I think they did justice to this s
15,?? the twin towers and ?? that in '94 they went to CAMP MARIAH CAREY
16,me: do you have this shoe in brown in a 9?sales lady: i have it in black in an 11m
17,"In 2001 Al Qaeda were the terrorists who knocked down the twin towers, in 2016 th
18,A Place to Call Home S4 starts September 11th 8.30pm on the new channel @ShowcaseA
19,"Celebratory bubbly time!!!It's official, I'm going stateside on September 11th fo
20,Tyler Hoechlin || 11th September 1987- So precious & cute- Excited to see him as s
21,#KYYNGGSLIME SEPTEMBER 11th? ??
22,Garden of Ireland Vintage Car Run 11th – 13th September. 2016#WakeUpInWicklow
23,"Song Hye Kyo and Song Joong Ki will attend the 11th Seoul International Drama Awa
25,September 10th & 11th.I'll be in Puerto Rico with Magcon :) come turn up with us !
27,Free entry for all at Hill Close Gardens featuring ‘Monty Don’s garden’ 10th-11th
28,Catch Chris Brown on 2016 Celebrity All*Star basketball game on Sunday September 1
29,She'll Be 46 On September 11th??
30,"JUST ANNOUNCED: Love Jones The Musical, hits Atlanta NOV 11th-13th!! Public on-sa
31,"18- Homem caindo do World Trade Center, em 11/09/01. "
32,"Reviving up for Shivaay Title Track. Releasing on 11th September, 3 days to go! #
33,"3 days to go for #Shivaay title track #BoloHarHarHar , releases on September 11th
34,We are TasK Teamtage now delayed to the 11th of September!
35,Exhibition closes 11th September. @fleurevaslet a great experience. #melboureartis
36,"Satsang; Naam; Ruhani Jaam will be held on 11th September 2016 at Shah Satnam Ji
37,"September 9, 2001: Two Days Before 9/11, Global Research was Born… "
38,Borough remembers the 15th anniversary of the September 11th attacks
```

```
# -*- coding: utf-8 -*-
# *** Spyder Python Console History Log ***
```

```
In [1]: runfile('E:/VTPML07/CODING/VTPML07_2019/ml.py',
wdir='E:/VTPML07/CODING/VTPML07_2019')
<class 'pandas.core.frame.DataFrame'>
      retweetCount  favoriteCount  creationDay  ...  4029
4030  4031
0             6205           3877            3  ...     0
0     0
1               31             43            4  ...     0
0     0
2              273            370            5  ...     0
0     0
3                1              0            1  ...     0
0     0
4                9             13            3  ...     0
0     0
...            ...            ...          ...  ...   ...
...   ...
1195             0              0            5  ...     0
0     0
1196             0              0            5  ...     0
0     0
```

Fig: Non – Rumors generated in NonRumors.csv file

CHAPTER 8
SOFTWARE TESTING

8.1 GENERAL

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.

8.2 DEVELOPING METHODOLOGIES

The test process is initiated by developing a comprehensive plan to test the general functionality and special features on a variety of platform combinations. Strict quality control procedures are used. The process verifies that the application meets the requirements specified in the system requirements document and is bug free. The following are the considerations used to develop the framework from developing the testing methodologies.

8.3 Types of Tests

8.3.1 Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program input 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.

8.3.2 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.

8.3.3 System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

8.3.4 Performance Test

The Performance test ensures that the output be produced within the time limits, and the time taken by the system for compiling, giving response to the users and request being send to the system for to retrieve the results.

8.3.5 Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

8.3.6 Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Acceptance testing for Data Synchronization:

- The Acknowledgements will be received by the Sender Node after the Packets are received by the Destination Node
- The Route add operation is done only when there is a Route request in need
- The Status of Nodes information is done automatically in the Cache Updation process

8.3.7 Build the test plan

Any project can be divided into units that can be further performed for detailed processing. Then a testing strategy for each of this unit is carried out. Unit testing helps to identity the possible bugs in the individual component, so the component that has bugs can be identified and can be rectified from errors.

CHAPTER 9

FUTURE ENHANCEMENT

Two broad classes of networks with a similar general structure, where one is finite impulse and the other is infinite impulse. Both classes of networks exhibit temporal dynamic behavior. A finite impulse recurrent network is a directed acyclic graph that can be unrolled and replaced with a strictly feedforward neural network, while an infinite impulse recurrent network is a directed cyclic graph that cannot be unrolled. Both finite impulse and infinite impulse recurrent networks can have additional stored state, and the storage can be under direct control by the neural network.

CHAPTER 10

CONCLUSION & REFERENCE

10.1 CONCLUSION

There has been a lot of work on rumor detection on OSNs. But, much of the works have focused only on microblogs like twitter only. There are some limitations to work with Facebook on this front such data access limitations due to user data protection and as such a lack of quality data. In this study, we underline the need of different approaches to rumor detection which address rumors other than political rumors, by highlighting the reach of social media in different fields of life. We present a brief study of characteristics and patterns of rumors studied by various people and their findings. Lastly, we present a review of various empirical studies in the field of rumor detection through Machine learning and deep learning frameworks. This study provides an insight into the direction of work in rumor detection vis-à-vis latest technologies and the intuition for the further work.

10.2 REFERENCES

[1] D. Lazer, A. S. Pentland, L. Adamic, S. Aral, A. L. Barabasi, D. Brewer, N. Chirstakis, N. Contractor and J. Flower, "Life in the network: The coming age of computational social science," Science, vol. 323, no. 5915, pp. 721-723, 6 February 2009.

[2] A. Hermida, "Twittering the news: The emergence of ambient journalism," Journalism Practice, vol. 4, no. 3, pp. 297-308, 2010.

[3] K. E. Matsa and E. Shearer, "News Use Across Social Media Platforms 2018," Pew Research Center, Journalism and Media, 2018.

[4] J. Cokley, "The Reuters Institute’s Digital News Report 2012," Digital Journalism, vol. 1, no. 2, pp. 286-287, 2013.

[5] S. Phuvipadawat and T. Murata, "Breaking news detection and tracking in Twitter," in IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, 2010.

[6] H. Webb, P. Burnap, R. Procter, O. Rana, B. C. Stahl, M. Williams, W. Housley, A. Edwards and M. Jirotka, "Digital wildfires: Propagation, Verification, Regulation, and Responsible Innovation," ACM Transactions on Information Systems-Special issue on Trust and Veracity of Information in Social Media, vol. 34, no. 3, pp. 1-23, 3 May 2016.

[7] R. Procter, f. Vis and A. Voss, "Reading the riots on Twitter: Methodological innovation for the analysis of Big data," International Journal of Social Research Methodology, vol. 16, no. 3, pp. 197-214, 2013.

[8] D. Lazer, M. Baum at al. , "The Science of Fake News," Science, vol. 359, no. 6380, pp. 1094-1096, 2018.

[9] J. v. Dijck, The Culture of Connectivity: A Critical History of Social Media, Oxford University Press, 2013.

[10] C. Fuchs, Social Media: A Critical Introduction, SAGE Publications Ltd, 2014