A Major Project Report on

Fake News Detection Using Python

A Dissertation Submitted in partial fulfillment of the requirement for the award of degree of

BACHELOR OF TECHNOLOGY

In

Electronics and Communication Engineering

By

TALLADA SAI SHARAN	**(Roll No: 20D41A04K5)**
SHAIK MOHAMMED ALEEMUDDIN	**(Roll No: 20D41A04J4)**
ALLAMLA SRIKANTH	**(Roll No: 18D41A0409)**
KATTA SRI LAKSHMI PRASANNA	**(Roll No: 21D45A0420)**

Under the esteemed guidance of
Ms. B. Deepika Rathod
Assoc. Prof

Department of Electronics and Communication Engineering

SRI INDU COLLEGE OF ENGINEERING & TECHNOLOGY
(An Autonomous Institution under UGC, New Delhi)
Recognized under 2(f) and 12(B) of UGC Act. 1956
Sheriguda village, Ibrahimpatnam, RR District – 501 510, T.S, INDIA

2023 - 2024

CERTIFICATE

This is to certify that the project report entitled **"FAKE NEWS DETECTION USING PYTHON"** submitted by

Mr. TALLADA SAI SHARAN	**(Roll No: 20D41A04K5)**
Mr. SHAIK MOHAMMED ALEEMUDDIN	**(Roll No: 20D41A04J4)**
Mr. ALLAMLA SRIKANTH	**(Roll No: 18D41A0409)**
Ms. KATTA SRI LAKSHMI PRASANNA	**(Roll No: 21D45A0420)**

In partial fulfillment of the requirement for the award of the degree of **Bachelor of Technology** in **Electronics and Communication Engineering** to the Jawaharlal Nehru Technological University, Hyderabad is a record of bonafide work carried out by them under our guidance and supervision

The results presented in this thesis have been verified and are found to be satisfactory. The results embodied in this thesis have not been submitted to any other University for the award of any other degree or diploma.

Ms. B. DEEPIKA RATHOD
Internal Guide

Dr. N.C. SENDHIL KUMAR
Head of the Department

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragement and guidance has been a source of inspiration throughout the course of the project**.**

It is my privilege and pleasure to express my profound sense of gratitude and indebtedness to my Project Guide **B. DEEPIKA RATHOD, Assoc. Prof.** of Electronics and Communication Engineering Department, Sri Indu College of Engineering & Technology, for her guidance, cogent discussion, constructive criticisms and encouragement throughout this dissertation work.

I take the opportunity to offer my humble thanks to **Dr. N.C. SENDHIL KUMAR, Prof. & Head of the Department,** Electronics & Communication Engineering, Sri Indu College of Engineering & Technology, for his encouragement and constant help.

I also thank **Dr. G.SURESH, Principal**, **SRI INDU COLLEGE OF ENGINEERING & TECHNOLOGY,** for his support in this Endeavour.

In addition I would like to thank all the **Faculty members** of Department of Electronics & Communication, & **Management,** who provided us with good lab facilities and helped us in carrying out the project successfully.

I finally thank my family members and friends for giving moral strength and support to complete this dissertation.

Mr. TALLADA SAI SHARAN	**(Roll No: 20D41A04K5)**
Mr. SHAIK MOHAMMED ALEEMUDDIN	**(Roll No: 20D41A04J4)**
Mr. ALLAMLA SRIKANTH	**(Roll No: 18D41A0409)**
Ms. KATTA SRI LAKSHMI PRASANNA	**(Roll No: 21D45A0420)**

ABSTRACT

This project is solely based on the purpose of creating a fake news detector for a given data set. This is a project associated with data analysis that was created using python programming language along with which machine learning classification algorithms namely Linear aggression, Decision tree classification, Gradient boost classification and Random Forest classification model were used. We live in the era where people blindly believe rumors and do not think twice before turning it into a gossip session without thoroughly checking the facts. Any kind of news no matter what gets spread quickly irrespective of time and distance.

Fake news, alternative facts are associated to each other since the time news was transmitted using newspapers or radio. There have been several hoax stories where citizens, governments as well all other social elements are all affected by these kinds of fake stories. Several social media organizations have been subjected to controversies by the media houses for targeting the audiences and showing them posts to their support. This project mainly focuses on detecting fake news with the help of various python libraries in association with counting feature such as TfidfVectorizer. The system will be taking input from the user and then compare them with an existing data-set. I have compared various algorithms to find out the best working model that will fit our project and give a proper prediction for fake news. The main objective is to detect the fake news, which is a classic text classification problem with a straightforward proposition. It is needed to build a model that can differentiate between “Real” news and “Fake” news thus helping in getting the facts out right.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

DFD	Data Flow Diagram
GPL	General Public License
GUI	Graphical User Interface
RAM	Random Access Memory
LSR	Least Square Regression
ML	Machine Learning
MTBF	Mean Time Between Failures
SNS	Social network sites
WWW	World Wide Web
WSGI	Web Server Gateway Interface

CHAPTER 1

INTRODUCTION

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. Obviously, a purposely misleading story is “fake news “but lately blathering social media’s discourse is changing its definition. Some of them now use the term to dismiss the facts counter to their preferred viewpoints.

The importance of disinformation within American political discourse was the subject of weighty attention, particularly following the American president election. 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, it is seemed to produce a model that can accurately predict the likelihood that a given article is fake news. Facebook has been at the epicenter 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 to distinguish these articles *in* an automated way. Certainly, it is not an easy task. 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.

1.1 MOTIVATION

We will be training and testing the data, when we use supervised learning, it means we are labeling the data. By getting the testing and training data and labels we can perform different machine learning algorithms but before performing the predictions and accuracies, the data is need to be preprocessing i.e. the null values which are not readable are required to be removed from the data set and the data is required to be converted into vectors by normalizing and tokening the data so that it could be understood by the machine. Next step is by using this data, getting the visual reports, which we will get by using the Mat Plot Library of Python and Sickit Learn. This library helps us in getting the results in the form of histograms, pie charts or bar charts.

1.2 OBJECTIVE

The objective of this project is to examine the problems and possible significances related with the spread of fake news. We will be working on different fake news data set in which we will apply different machine learning algorithms to train the data and test it to find which news is the real news or which one is the fake news. As the fake news is a problem that is heavily affecting society and our perception of not only the media but also facts and opinions themselves. By using the artificial intelligence and the machine learning, the problem can be solved as we will be able to mine the patterns from the data to maximize well defined objectives. So, our focus is to find which machine learning algorithm is best suitable for what kind of text dataset. Also, which dataset is better for finding the accuracies as the accuracies directly depends on the type of data and the amount of data. The more the data, more are your chances of getting correct accuracy as you can test and train more data to find out your results.

1.3 OVERVIEW OF PROJECT

With the advancement of technology, digital news is more widely exposed to users globally and contributes to the increment of spreading and disinformation online. Fake news can be found through popular platforms such as social media and the Internet. There have been multiple solutions and efforts in the detection of fake news where it even works with tools. However, fake news intends to convince the reader to believe false information which deems these articles difficult to perceive. The rate of producing digital news is large and quick, running daily at every second, thus it is challenging for machine learning to effectively detect fake news

CHAPTER 2

LITERATURE SURVEY

The available literature has described many automatic detection techniques of fake news and deception posts. Since there are multidimensional aspects of fake news detection ranging from using chatbots for spread of misinformation to use of clickbait's for the rumor spreading. There are many clickbait's available in social media networks including Facebook which enhance sharing and liking Proceedings of posts which in turn spreads falsified information. Lot of work has been done to detect falsified information.

2.1 MEDIA RICH FAKE NEWS DETECTION: A SURVEY

In general, the goal is profiting through clickbaits. Clickbaits lure users and entice curiosity with flashy headlines or designs to click links to increase advertisements revenues. This exposition analyzes the prevalence of fake news in light of the advances in communication made possible by the emergence of social networking sites. The purpose of the work is to come up with a solution that can be utilized by users to detect and filter out sites containing false and misleading information. We use simple and carefully selected features of the title and post to accurately identify fake posts. The experimental results show a 99.4% accuracy using logistic classifier.

2.1.1 WEAKLY SUPERVISED LEARNING FOR FAKE NEWS DETECTION ON TWITTER

The problem of automatic detection of fake news in social media, e.g., on Twitter, has recently drawn some attention. Although, from a technical perspective, it can be regarded as a straight-forward, binary classification problem, the major challenge is the collection of large enough training corpora, since manual annotation of tweets as fake or non-fake news is an expensive and tedious endeavor. In this paper, we discuss a weakly supervised approach, which automatically collects a large-scale, but very noisy training dataset comprising hundreds of thousands of tweets. During collection, we automatically label tweets by their source, i.e., trustworthy or untrustworthy source, and train a classifier on this dataset. We then use that classifier for a different classification target, i.e., the classification of fake and non- fake tweets. Although the labels are not accurate according to the new classification target (not all tweets by an untrustworthy source need to be fake news, and vice versa), we show that despite this unclean inaccurate dataset, it is possible to detect fake news with an F1 score of up to 0.9.

2.2 FAKE NEWS DETECTION IN SOCIAL MEDIA

Fake news and hoaxes have been there since before the advent of the Internet. The widely accepted definition of Internet fake news is: fictitious articles deliberately fabricated to deceive readers”. Social media and news outlets publish fake news to increase readership or as part of psychological warfare. In general, the goal is profiting through clickbaits. Clickbaits lure users and entice curiosity with flashy headlines or designs to click links to increase advertisements revenues. This exposition analyzes the prevalence of fake news in light of the advances in communication made possible by the emergence of social networking sites. The purpose of the work is to come up with a solution that can be utilized by users to detect and filter out sites containing false and misleading information. We use simple and carefully selected features of the title and post to accurately identify fake posts. The experimental results show a 99.4% accuracy using logistic classifier.

Automatic Online Fake News Detection Combining Content and Social Signals

The proliferation and rapid diffusion of fake news on the Internet highlight the need of automatic hoax detection systems. In the context of social networks, machine learning (ML) methods can be used for this purpose. Fake news detection strategies are traditionally either based on content analysis (i.e. analyzing the content of the news) or - more recently - on social context models, such as mapping the news‟ diffusion pattern. In this paper, we first propose a novel ML fake news detection method which, by combining news content and social context features, outperforms existing methods in the literature, increasing their already high accuracy by up to 4.8%. Second, we implement our method within a Facebook Messenger chatbot and validate it with a real-world application, obtaining a fake news detection accuracy of 81.7%.

In recent years, the reliability of information on the Internet has emerged as a crucial issue of modern society. Social network sites (SNSs) have revolutionized the way in which information is spread by allowing users to freely share content. As a consequence, SNSs are also increasingly used as vectors for the diffusion of misinformation and hoaxes. The amount of disseminated information and the rapidity of its diffusion make it practically impossible to assess reliability in a timely manner, highlighting the need for automatic hoax detection systems. As a contribution towards this objective, we show that Facebook posts can be classified with high accuracy as hoaxes or non-hoaxes on the basis of the users who "liked" them. We present two classification techniques, one based on logistic regression, the other on a novel adaptation of Boolean crowdsourcing algorithms. On a dataset consisting of 15,500 Facebook posts and 909,236 users, we obtain classification accuracies exceeding 99% even when the training set contains less than 1% of the posts. We further show that our techniques are robust: they work even when we restrict our attention to the users who like both hoax and non-hoax posts. These results suggest that mapping the diffusion pattern of information can be a useful component of automatic hoax detection systems.

2.3 THE SPREAD OF FAKE NEWS BY SOCIAL BOTS

The massive spread of fake news has been identified as a major global risk and has been alleged to influence elections and threaten democracies. Communication, cognitive, social, and computer scientists are engaged in efforts to study the complex causes for the viral diffusion of digital misinformation and to develop solutions, while search and social media platforms are beginning to deploy countermeasures. However, to date, these efforts have been mainly informed by anecdotal evidence rather than systematic data. Here we analyze 14 million messages spreading 400 thousand claims on Twitter during and following the 2016 U.S. presidential campaign and election. We find evidence that social bots play a key role in the spread of fake news. Accounts that actively spread misinformation are significantly more targeted.

Automated accounts are particularly active in the early spreading phases of viral claims, and tend to target influential users. Humans are vulnerable to this manipulation, retweeting bots who post false news. Successful sources of false and biased claims are heavily supported by social bots. These results suggests that curbing social bots may be an effective strategy for mitigating the spread of online misinformation.

2.4 MISLEADING ONLINE CONTENT

Tabloid journalism is often criticized for its propensity for exaggeration, sensationalization, scare-mongering, and otherwise producing misleading and low-quality news. As the news has moved online, a new form of tabloidization has emerged: click baiting. Clickbait refers to "content whose main purpose is to attract attention and encourage visitors to click on a link to a particular web page" [clickbait, n.d.] and has been implicated in the rapid spread of rumor and misinformation online. This paper examines potential methods for the automatic detection of clickbait as a form of deception. Methods for recognizing both textual and non-textual click baiting cues are surveyed, leading to the suggestion that ahybrid approach may yield best results.

Big Data Analytics and Deep Learning are two high-focus of data science. Big Data has become important as many organizations both public and private have been collecting massive amounts of domain-specific information, which can contain useful information about problems such as national intelligence, cyber security, fraud detection, marketing, and medical informatics. Companies such as Google and Microsoft are analyzing large volumes of data for business analysis and decisions, impacting existing and future technology. Deep Learning algorithms extract high- level, complex abstractions as data representations through a hierarchical learning process. Complex abstractions are learnt at a given level based on relatively simpler abstractions formulated in the preceding level in the hierarchy. A key benefit of Deep Learning is the analysis and learning of massive amounts of unsupervised data, making it a valuable tool for Big Data Analytics where raw data is largely unlabeled and un-categorized.

In the present study, we explore how Deep Learning can be utilized for addressing some important problems in Big Data Analytics, including extracting complex patterns from massive volumes of data, semantic indexing, data tagging, fast information retrieval, and simplifying discriminative tasks. We also investigate some aspects of Deep Learning research that need further exploration to incorporate specific challenges introduced by Big Data Analytics, including streaming data, high-dimensional data, scalability of models, and distributed computing. We conclude by presenting insights into relevant future works by posing some questions, including defining data sampling criteria, domain adaptation modeling, defining criteria for obtaining useful data abstractions, improving semantic indexing, semi - supervised learning, and active learning.

CHAPTER 3

METHODOLOGY

3.1 EXISTING SYSTEM

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. Particularly since late 2016 during the American Presidential election, the question of determining 'fake news' has also been the subject of particular attention within the literature. Conroy, Rubin, and Chen outline several approaches that seem promising towards the aim of perfectly classify the misleading articles. They note that simple content-related n-grams and shallow parts-of-speech tagging have proven insufficient for the classification task, often failing to account for important context information. Rather, these methodshave been shown useful only in tandem with more complex methods of analysis. Deep Syntax analysis using Probabilistic Context Free Grammars have been shown to be particularly valuable in combination with n-gram methods. Feng, Banerjee, and Choi are able to achieve 85%-91% accuracy in deception related classification tasks using online review corpora.

3.2 PROPOSED SYSTEM

In this paper a model is build based on the count vectorizer or a tfidf matrix (i.e.) word tallies relatives to how often they are used in other articles in your dataset) can help . Since this problem is a kind of text classification, implementing a Naive Bayes classifier will be best as this is standard for text-based processing. The actual goal is in developing a model which was the text transformation (count vectorizer vs tfidf vectorizer) and choosing which type of text to use (headlines vs full text).

Now the next step is to extract the most optimal features for count vectorizer or tfidf-vectorizer, this is done by using a n-number of the most used words, and/or phrases, lower casing or not, mainly removing the stop words which are common words such as “the”, “when”, and “there” and only using those words that appear at least a given number of times in a given text dataset.

SYSTEM ARCHITECTURE

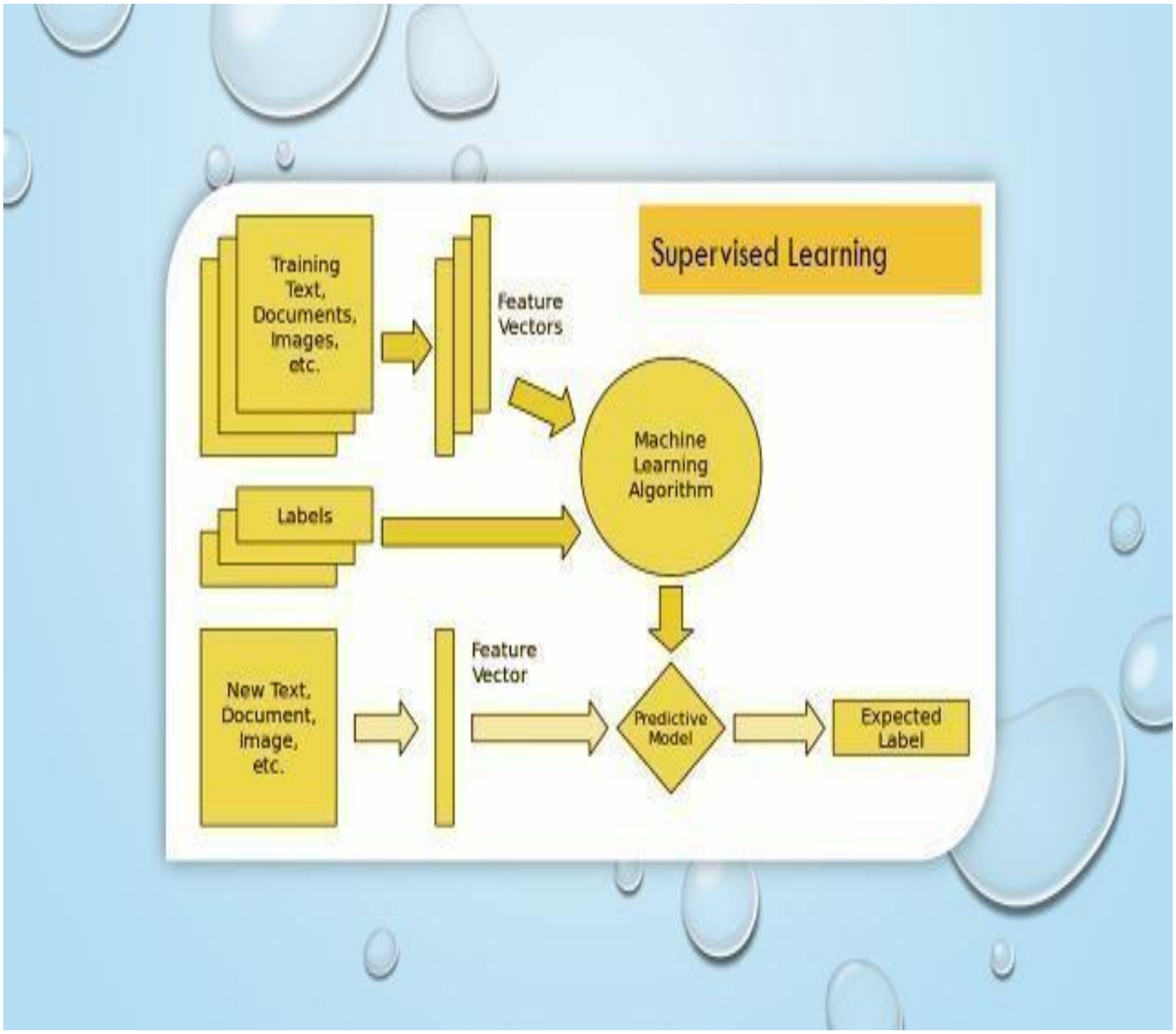

Supervised Learning
Training Text, Documents, Images, etc.
Feature Vectors
Machine Learning Algorithm
Labels
New Text, Document, Image, etc.
Feature Vector
Predictive Model
Expected Label

Fig:3.1 Architecture diagram

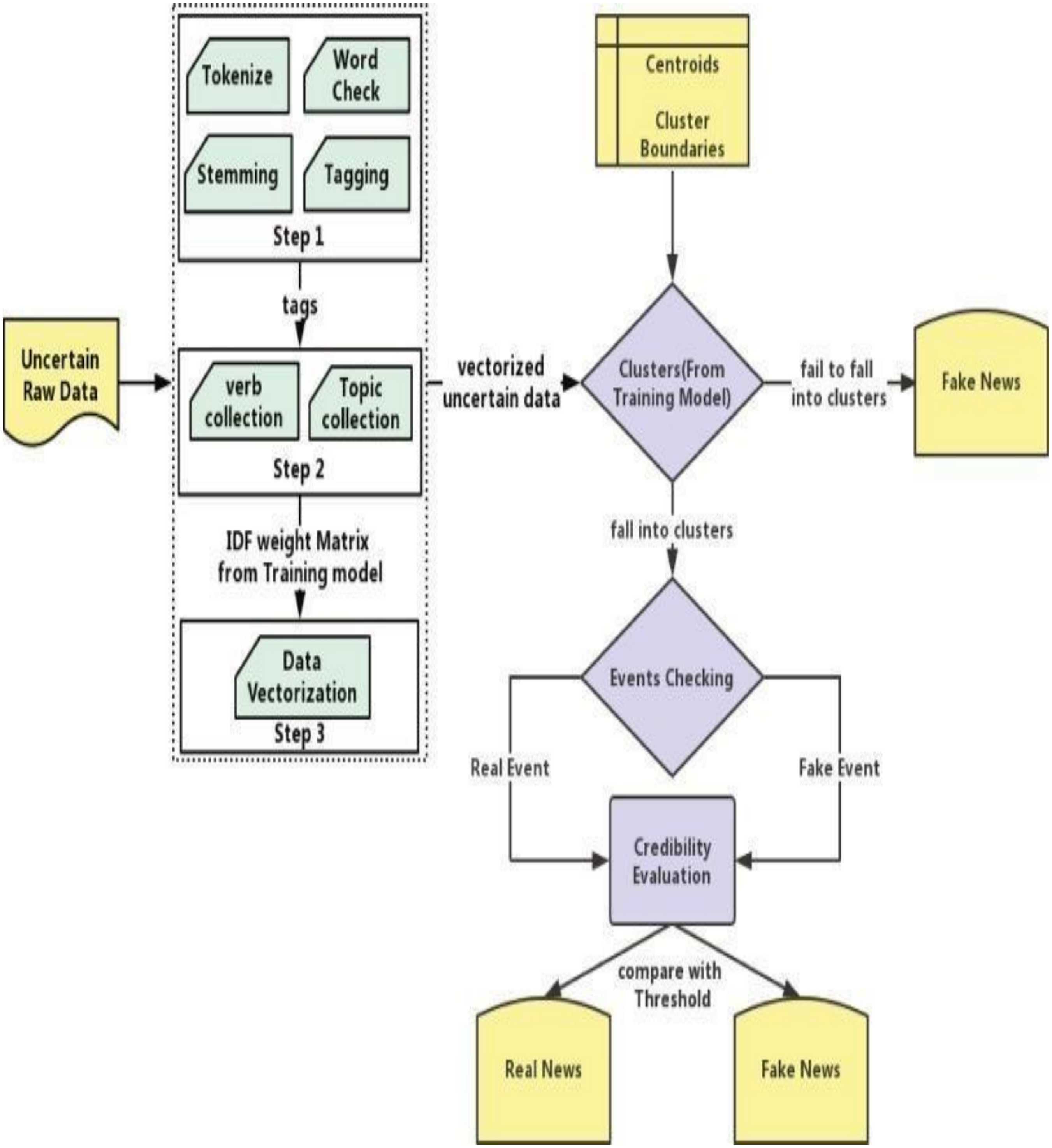


Fig.3.1.1 Diagrammatic Representation of classification of news into fake news and real news after undergoing certain process

In this project the first task that we do is to set a dataset containing enough amount of fake news and true news. Then we will do the coding according to the location of the directory at which the dataset has been stored. Then we use the basic required libraries such as pandas, sklearn where we import them to begin with our coding. Amidst this a TdidfVectorizer will be used to detect the frequency of the words provided in the data. Later onto the project Machine learning algorithms will be implemented to the codes so as to verify the accuracy of the provided datasets. And hence the user is now allowed to give in the input with which they will be able to extract the data with respect to its authenticity as to whether its Fake or Real.

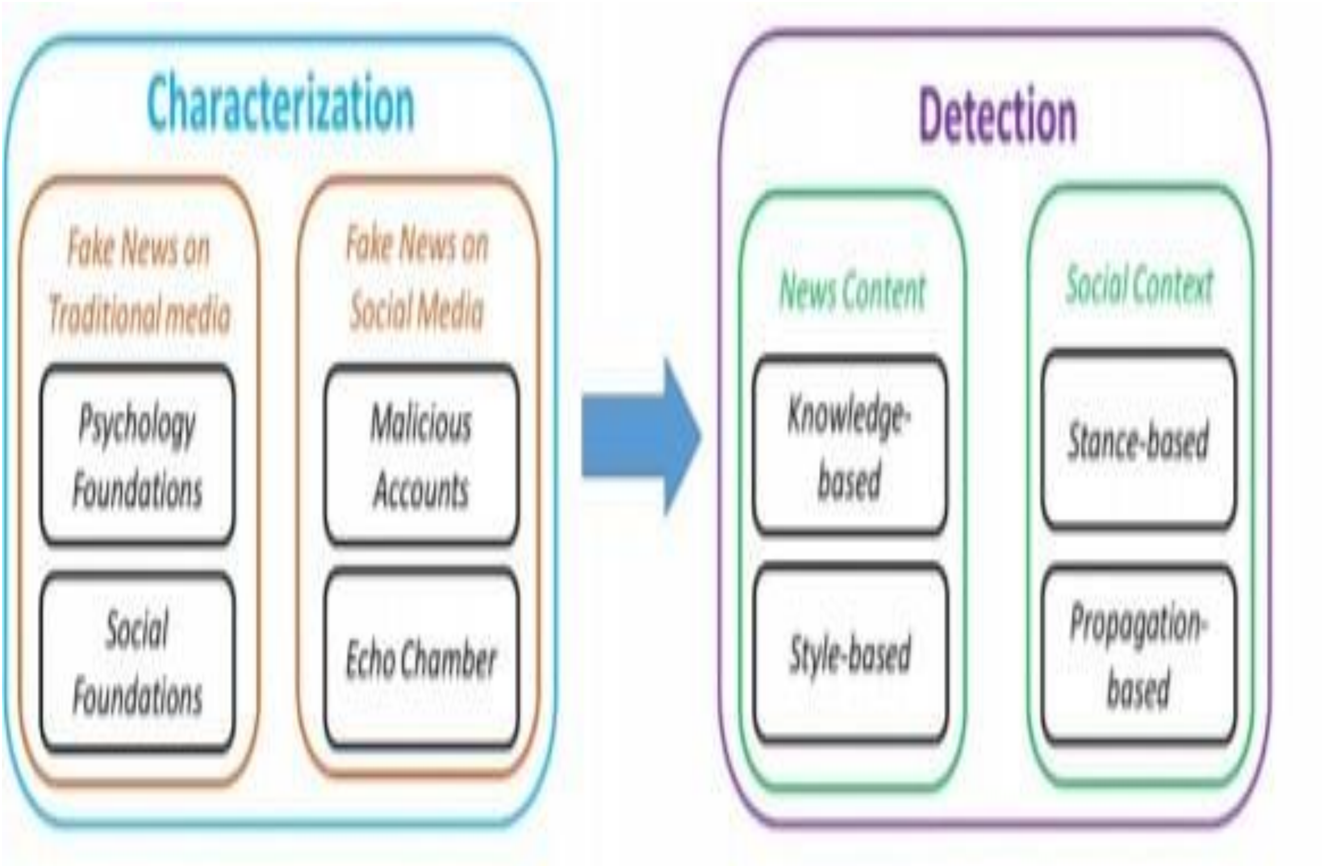


Fig.3.1.2 Characterization and detection of Fake news detection

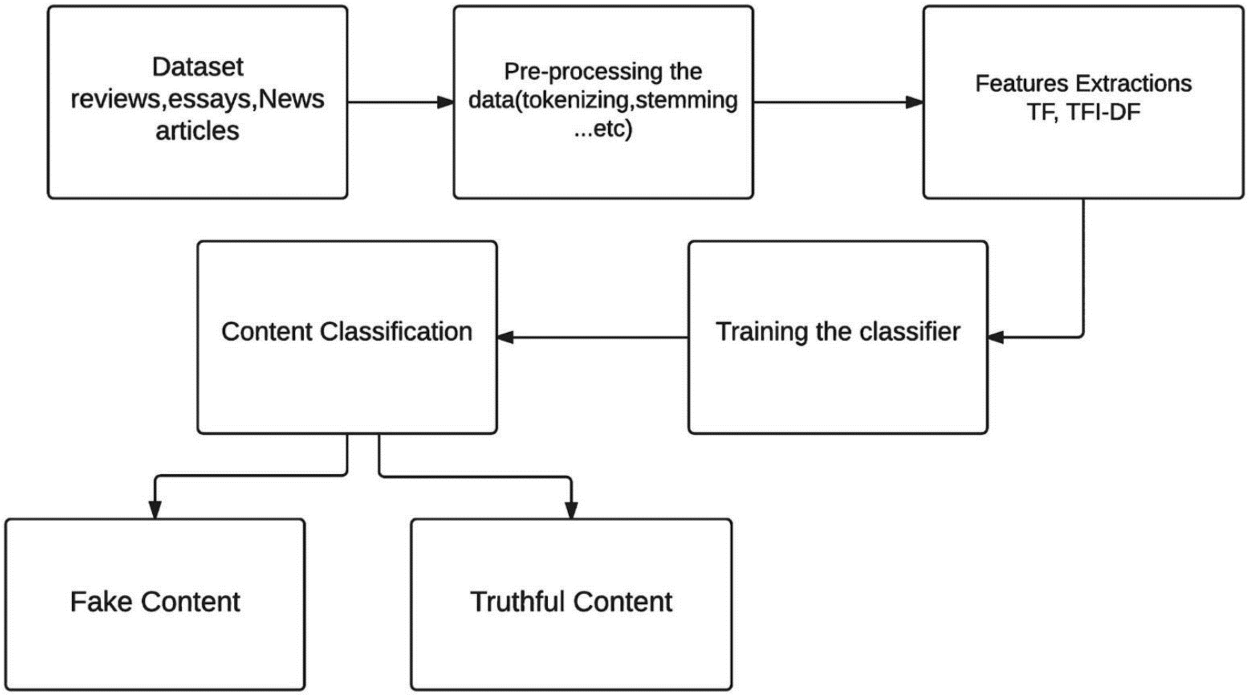


Fig.3.1.3 Architecture of fake news detection

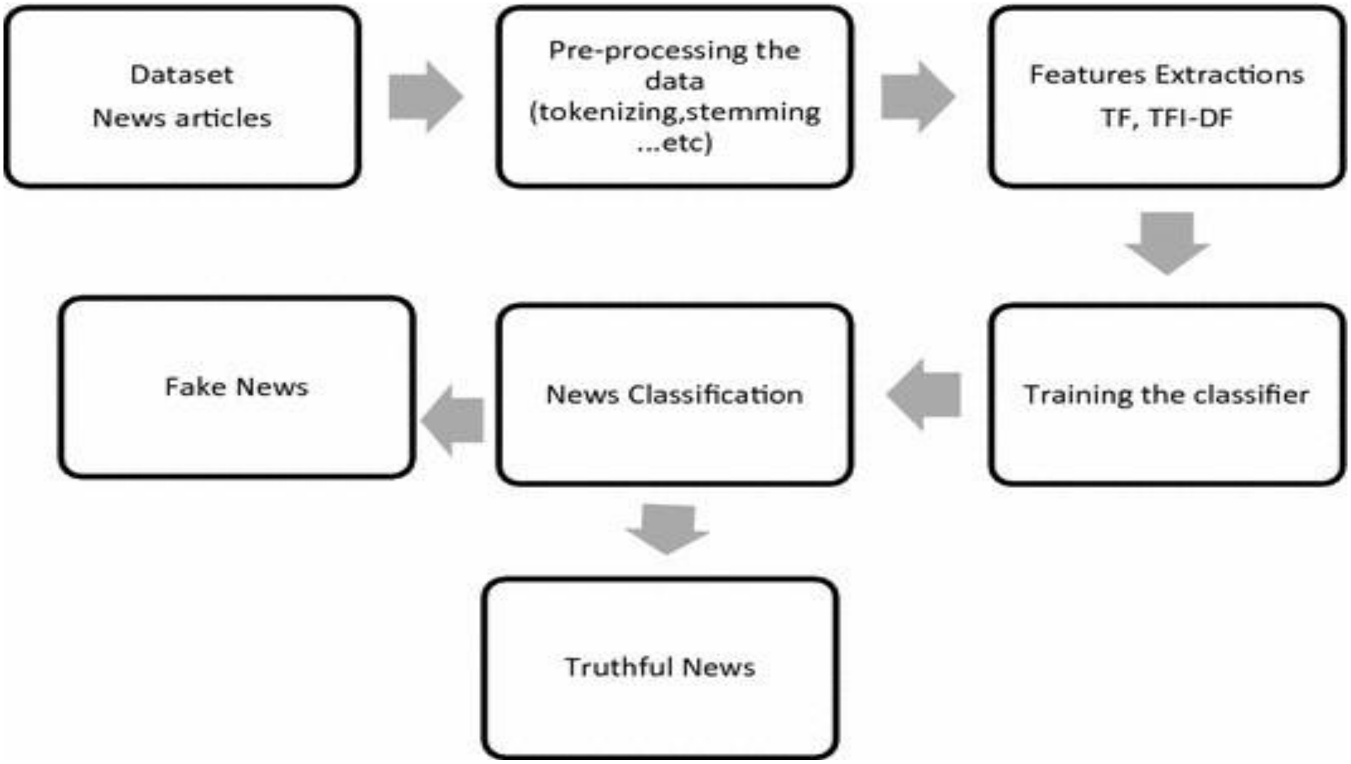


Fig.3.1.4 Flowchart to represent how the processing of fake news works

SYSTEM REQUIREMENTS

HARDWARE

REQUIREMENTS:

- System - Pentium-IV
- Speed - 2.4GHZ
- Hard disk - 40GB
- Monitor - 15VGA color
- RAM - 512MB

SOFTWARE REQUIREMENTS:

- Operating System - Windows XP
- Coding language - PYTHON

3.3 SOFTWARE ENVIRONMENT

3.3.1 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. 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 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.

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 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.

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.

3.3.2 PYTHON FEATURES

Python's features include −

- **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.

Python is available on a wide variety of platforms including Linux and Mac OS X.Let's understand how to set up our Python environment.

Getting Python

The most up-to-date and current source code, binaries, documentation, news, etc., is available on the official website of Python https://www.python.org.

Windows Installation

Here are the steps to install Python on Windows machine.

[1] Open a Web browser and go to https://www.python.org/downloads/.

[2] Follow the link for the Windows installer python-XYZ.msifile where XYZ is the version you need to install.

[3] To use this installer python-XYZ.msi, the Windows system must support Microsoft Installer 2.0. Save the installer file to your local machine and then run it to find out if your machine supports MSI.

[4] Run the downloaded file. This brings up the Python install wizard, which is really easy to use. Just accept the default settings, wait until the install is finished, and you are done.

The Python language has many similarities to Perl, C, and Java. However, there are some definite differences between the languages.

3.3.3 INTERACTIVE MODE PROGRAMMING

Invoking the interpreter without passing a script file as a parameter brings up the following prompt −

```
$ python

Python2.4.3(#1,Nov112010,13:34:43)
```

```
[GCC 4.1.220080704(RedHat4.1.2-48)] on linux2

Type"help","copyright","credits"or"license"for more information.

>>>
```

Type the following text at the Python prompt and press the Enter –

```
>>>print"Hello, Python!"
```

If you are running new version of Python, then you would need to use print statement with parenthesis as in **print ("Hello, Python!");**. However in Python version 2.4.3, this produces the following result –

```
Hello, Python!
```

3.3.4 SCRIPT MODE PROGRAMMING

Invoking the interpreter with a script parameter begins execution of the script and continues until the script is finished. When the script is finished, the interpreter is no longer active.

Let us write a simple Python program in a script. Python files have extension **.py**. Type the following source code in a test.py file –

```
print"Hello, Python!"
```

We assume that you have Python interpreter set in PATH variable. Now, try to run this program as follows –

```
$ python test.py
```

This produces the following result –

```
Hello, Python!
```

3.4 FLASK FRAMEWORK

Flask is a web application framework written in Python. Armin Ronacher, who leads an international group of Python enthusiasts named Pocco, develops it. Flask is based on Werkzeug WSGI toolkit and Jinja2 template engine. Both are Pocco projects.

Http protocol is the foundation of data communication in world wide web. Different methods of data retrieval from specified URL are defined in this protocol.

The following table summarizes different http methods −

Sl. No	Methods & Description
1	**GET** Sends data in unencrypted form to the server. Most common method.
2	**HEAD** Same as GET, but without response body
3	**POST** Used to send HTML form data to server. Data received by POST method is not cached by server.
4	**PUT** Replaces all current representations of the target resource with theuploaded content.

5	**DELETE** Removes all current representations of the target resource given by a URL

By default, the Flask route responds to the **GET** requests. However, this preference can be altered by providing methods argument to **route()** decorator.

In order to demonstrate the use of **POST** method in URL routing, first let us create an HTML form and use the **POST** method to send form data to a URL.

Save the following script as login.html

```
<html>

<body>

<formaction="http://localhost:5000/login"method="post">

<p>Enter Name:</p>

<p><inputtype="text"name="nm"/></p>

<p><inputtype="submit" value="submit"/></p>

</form>

</body>

</html>
```

Now enter the following script in Python shell.

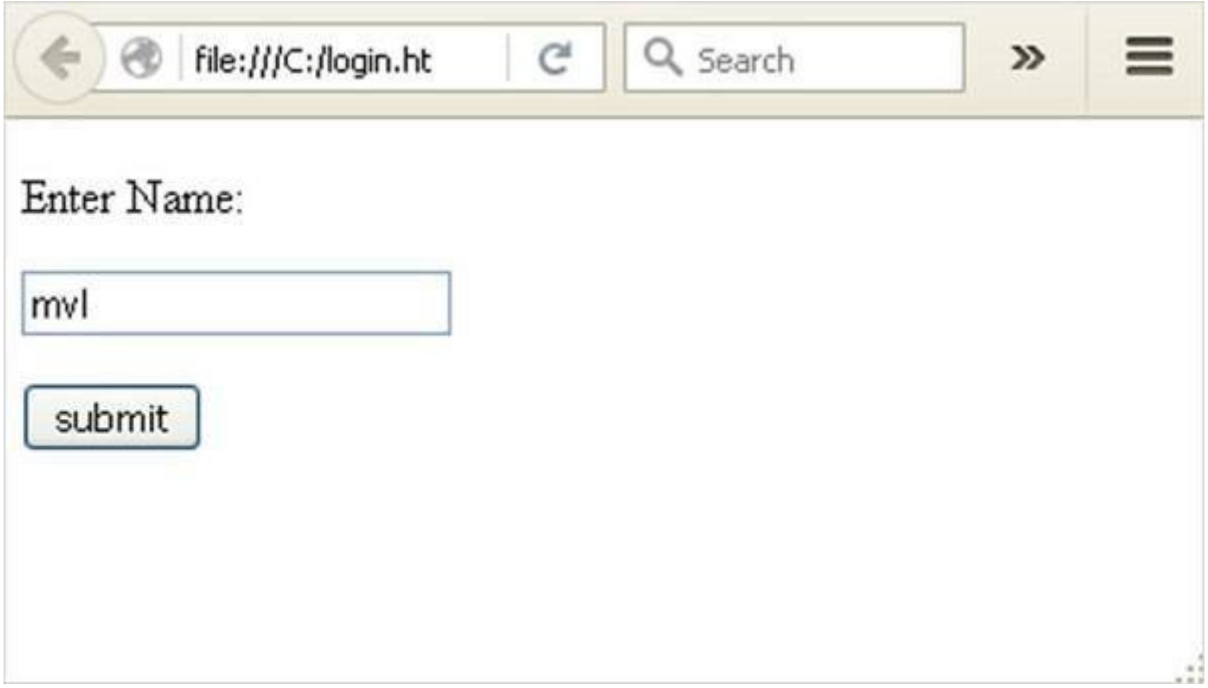
```
from flask importFlask, redirect,url_for, request

app=Flask(__name__)
```

```
@app.route('/success/<name>')

def success(name):

return'welcome  %s'% name

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

def login():

ifrequest.method=='POST':

user=request.form['nm']

return redirect(url_for('success',name= user))

else:

user=request.args.get('nm')

return redirect(url_for('success',name= user))

if__name__=='__main_':

app.run(debug =True)
```

After the development server starts running, open **login.html** in the browser, enter name in the text

When the input data to an algorithm is too large to be handled and i ts supposed to be redundant then the input data will be transformed into a reduced illustration set of features also named feature vectors. Altering the input data to perform the desired task using this reduced representation instead of the full-size input.

And in the next page click **Submit.**



Enter Name:

mvl

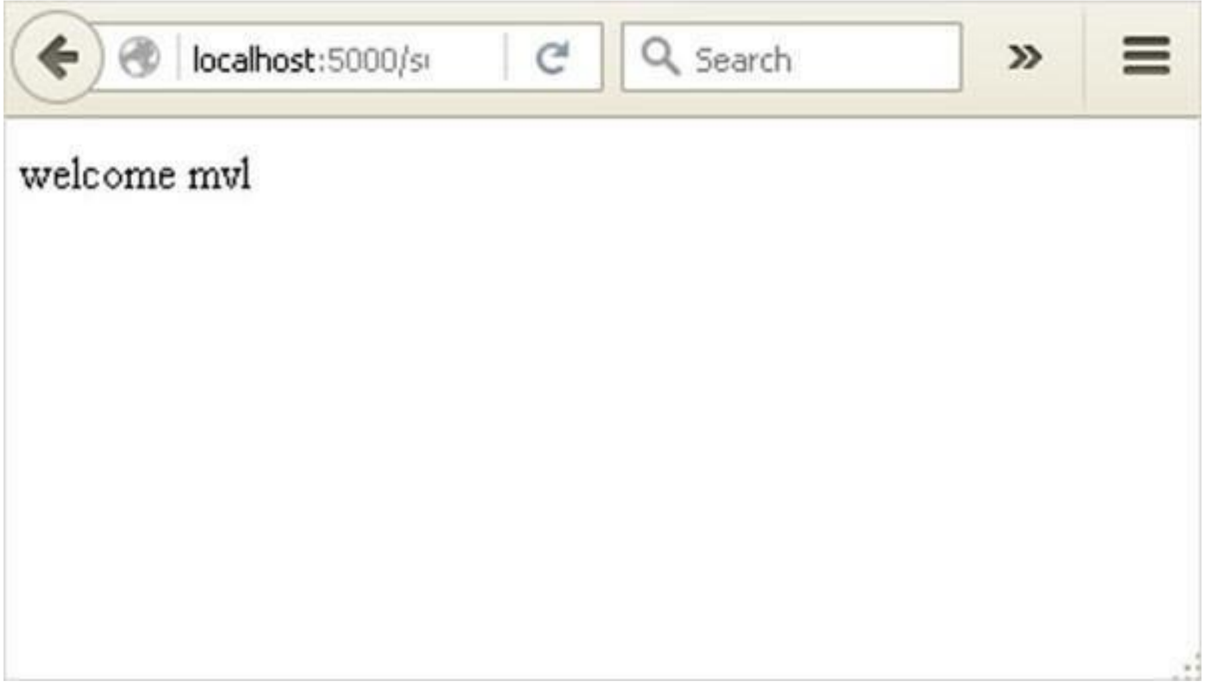
Fig:3.2 Login page

Form data is Posted to the URL in action clause of form tag.

http://localhost/login mapped to the **login()** function. Since the server has received data by **POST** method, value of „nm‟ parameter obtained from the form data is obtained by – We will be training and testing the data, when we use supervised learning it means we are labeling the data. By getting the testing and training data and labels we can perform different machine learning algorithms but before performing the predictions and accuracies, the data is need to be preprocessing i.e. the null values which are not readable are required to be removed from the data set and the data is required to be converted into vectors by normalizing and tokening the data so that it could be understood by the machine.

```
user = request.form['nm']
```

It is passed to **„/success‟** URL as variable part. The browser displays a **welcome** message in the window.



welcome mvl

Fig:3.3 Dashboard

Change the method parameter to **„GET‟** in **login.html in** fig 3.2 and open it again in the browser. The data received on server is by the **GET** method. The value of „nm‟ parameter is now obtained by –

```
User = request.args.get(„nm‟)
```

Here, **args** is dictionary object containing a list of pairs of form parameter and its corresponding value. The value corresponding to „nm‟ parameter is passed on to „/success‟.

3.5 MODULES

These are the different modules:

A. Data Use
B. Preprocessing
C. Feature Extraction
D. Training the Classifier

MODULES DESCRIPTION

A. Data Use

So, in this project we are using different packages and to load and read the data set we are using pandas. By using pandas, we can read the .csv file and then we can display the shape of the dataset with that we can also display the dataset in the correct form. We will be training and testing the data, when we use supervised learning, it means we are labeling the data. By getting the testing and training data and labels we can perform different machine learning algorithms but before performing the predictions and accuracies, the data is need to be preprocessing i.e. the null values which are not readable are required to be removed from the data set and the data is required to be converted into vectors by normalizing and tokening the data so that it could be understood by the machine. Next step is by using this data, getting the visual reports, which we will get by using the Mat Plot Library of Python and Sickit Learn. This library helps us in getting the results in the form of histograms, pie charts or bar charts.

B. Preprocessing

The data set used is split into a training set and a testing set containing in Dataset 1
-3256 training data and 814 testing data and in Dataset II- 1882 training data and 471 testing data respectively. Cleaning the data is always the first step. In this, those words are removed from the dataset. That helps in mining the useful information. Whenever we collect data online, it sometimes contains the undesirable characters like stop words, digits etc. which creates hindrance while spam detection. It helps in removing the texts which are language independent entities and integrate the logic which can improve the accuracy of the identification task. This just needs importing the packages and you can compile the command as soon as you write it. If the command doesn’t run, we can get the error at the same time. I am using 4 different algorithms and I have trained these 4 models i.e. Naïve Bayes, Support Vector Machine, K Nearest Neighbors and Logistic Regression win are very popular methods for document classification problem.

C. Feature Extraction

Feature extraction s the process of selecting a subset of relevant features for use in model construction. Feature extraction methods helps in to create an accurate predictive model. They help in selecting features that will give better accuracy. When the input data to an algorithm is too large to be handled and i ts supposed to be redundant then the input data will be transformed into a reduced illustration set of features also named feature vectors. Altering the input data to perform the desired task using this reduced representation instead of the full-size input. Feature extraction is performed on raw data prior to applying any machine learning algorithm, on the transformed data in feature space.

D. Training the Classifier

As In this project I am using Scikit-Learn Machine learning library for implementing the architecture. Scikit Learn is an open-source python Machine Learning library which comes bundled in 3rd distribution anaconda. This just needs importing the packages and you can compile the command as soon as you write it. If the command doesn’t run, we can get the error at the same time. I am using 4 different algorithms and I have trained these 4 models i.e. Naïve Bayes, Support Vector Machine, K Nearest Neighbors and Logistic Regression win are very popular methods for document classification problem. Once the classifiers are trained, we can c heck the performance of the models on test-set. We can extract the word count vector for each mail in test-set and predict it class with the trained models. More specifically, that y can be calculated from a linear combination of the input variables (x). When there is a single input variable (x), the method is referred to as simple linear regression. When there are multiple input variables, literature from statistics often refers to the method as multiple linear regression. It is common to therefore refer to a model prepared this way as Ordinary Least Squares Linear Regression or just Least Squares Regression. Linear Regression is an attractive model because the representation is so simple.

3.6 Algorithms

In this project we have 4 Machine Learning classification algorithms and they are:

1. LINEAR REGRESSION

Machine learning, more specifically the field of predictive modelling is primarily concerned with minimizing the error of a model or making the most accurate predictions possible, at the expense of an explanation. In applied machine learning we will borrow, reuse and steal algorithms from many different fields, including statistics and use them towards these ends. As such, linear regression was developed in the field of statistics and is studied as a model for understanding the relationship between input and output numerical variables, but has been borrowed by machine learning. It is both a statistical algorithm and a machine learning algorithm. Linear regression is a linear model, e.g. a model that assumes a linear relationship between the input variables (x) and the single output variable (y). More specifically, that y can be calculated from a linear combination of the input variables (x). When there is a single input variable (x), the method is referred to as simple linear regression. When there are multiple input variables, literature from statistics often refers to the method as multiple linear regression. Different techniques can be used to prepare or train the linear regression equation from data, the most common of which is called Ordinary Least Squares. It is common to therefore refer to a model prepared this way as Ordinary Least Squares Linear Regression or just Least Squares Regression. Linear Regression is an attractive model because the representation is so simple. The representation is a linear equation that combines a specific set of input values (x) the solution to which is the predicted output for that set of input values (y). As such, both the input values (x) and the output value are numeric. The linear equation assigns one scale factor to each input value or column, called a coefficient and represented by the capital Greek letter Beta (B). One additional coefficient is also added, giving the line an additional degree of freedom (e.g. moving up and down on a two-dimensional plot) and is often called the intercept or the bias coefficient. In applied machine learning we will borrow, reuse and steal algorithms from many different fields, including statistics and use them towards these ends.

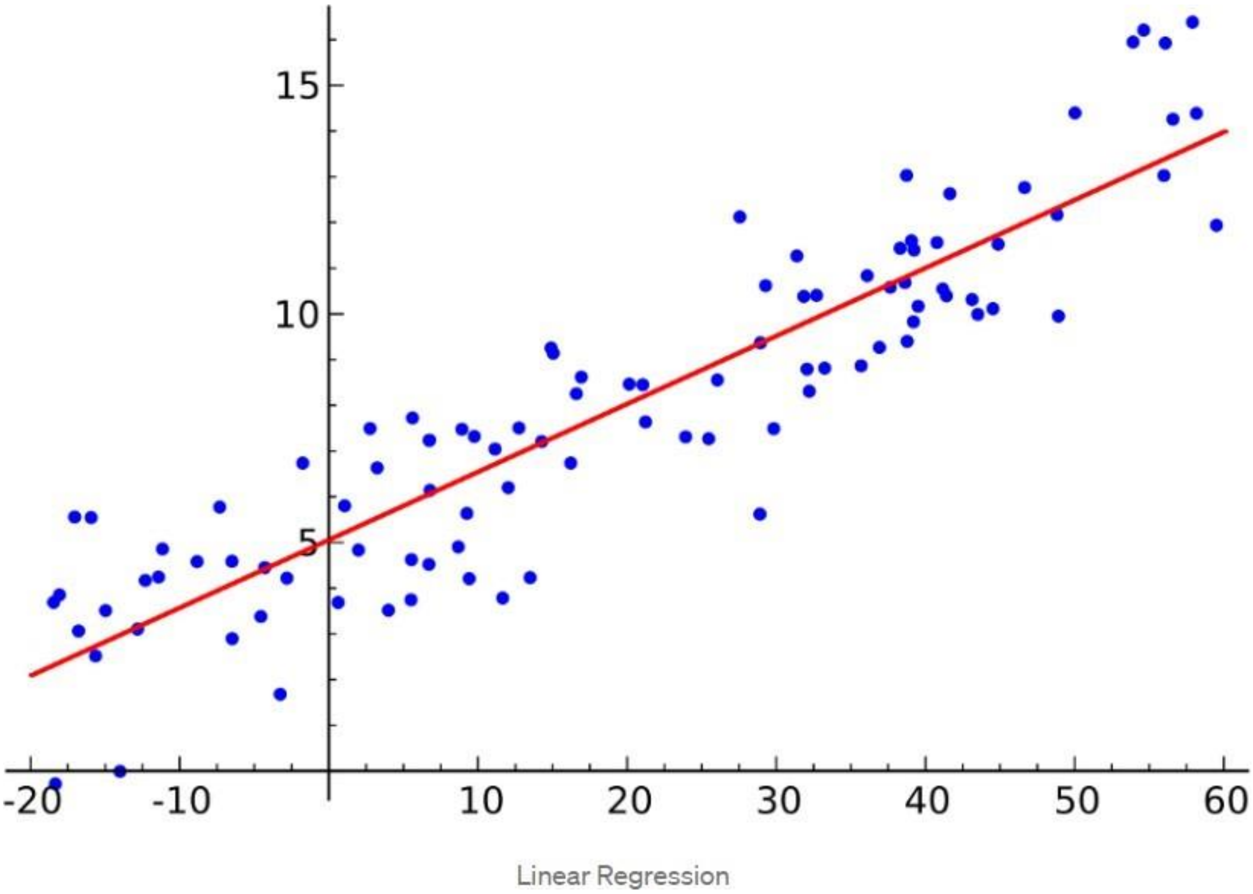


Fig 3.4 Linear Regression

2.DECISION TREE CLASSIFICATION

Decision tree is the most powerful and popular tool for classification and prediction. A Decision tree is a flowchart like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label. A tree can be learned by splitting the source set into subsets based on an attribute value test. This process is repeated on each derived subset in a recursive manner called recursive partitioning. The recursion is completed when the subset at a node all has the same value of the target variable, or when splitting no longer adds value to the predictions. The construction of decision tree classifier does not require any domain knowledge or parameter setting, and therefore is appropriate for exploratory knowledge discovery. Decision trees can handle high dimensional data. In general decision tree classifier has good accuracy.

Decision tree induction is a typical inductive approach to learn knowledge on classification. Below are some assumptions that we might make while using decision tree:

- At the beginning, we consider the whole training set as the root.
- Feature values are preferred to be categorical. If the values are continuous then they are discretized prior to building the model.
- On the basis of attribute values records are distributed recursively.
- We use statistical methods for ordering attributes as root or the internal node.

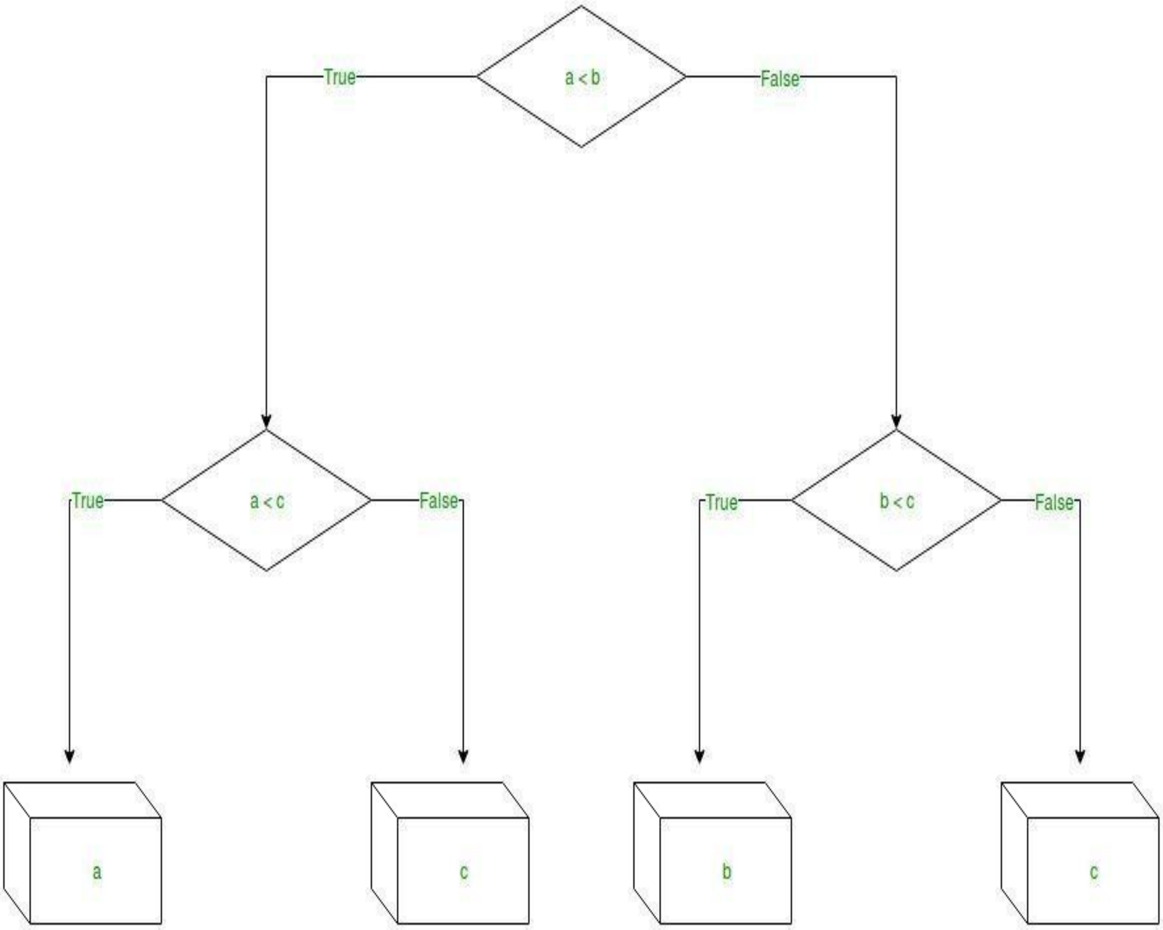


Fig 3.5 Decision Tree Classification

3.GRADIENT BOOST CLASSIFICATION

Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees.

Gradient boosting re-defines boosting as a numerical optimization problem where the objective is to minimize the loss function of the model by adding weak learners using gradient descent. Gradient Boosting is a first-order iterative optimization algorithm for finding a local minimum of a differentiable function. As gradient boosting is based on minimizing a loss function, different types of loss functions can be used resulting in a flexible technique that can be applied to regression, multi-class classification, etc. Intuitively, gradient boosting is a stage-wise additive model that generates learners during the learning process (i.e., trees are added one at a time, and existing trees in the model are not changed). The contribution of the weak learner to the ensemble is based on the gradient descent optimization process. The calculated contribution of each tree is based on minimizing the overall error of the strong learner. Gradient boosting does not modify the sample distribution as weak learners train on the remaining residual errors of a strong learner (i.e, pseudo-residuals). By training on the residuals of the model, this is an

alternative means to give more importance to misclassified observations. Intuitively, new weak learners are being added to concentrate on the areas where the existing learners are performing poorly. The contribution of each weak learner to the final prediction is based on a gradient optimization process to minimize the overall error of the strong learner.

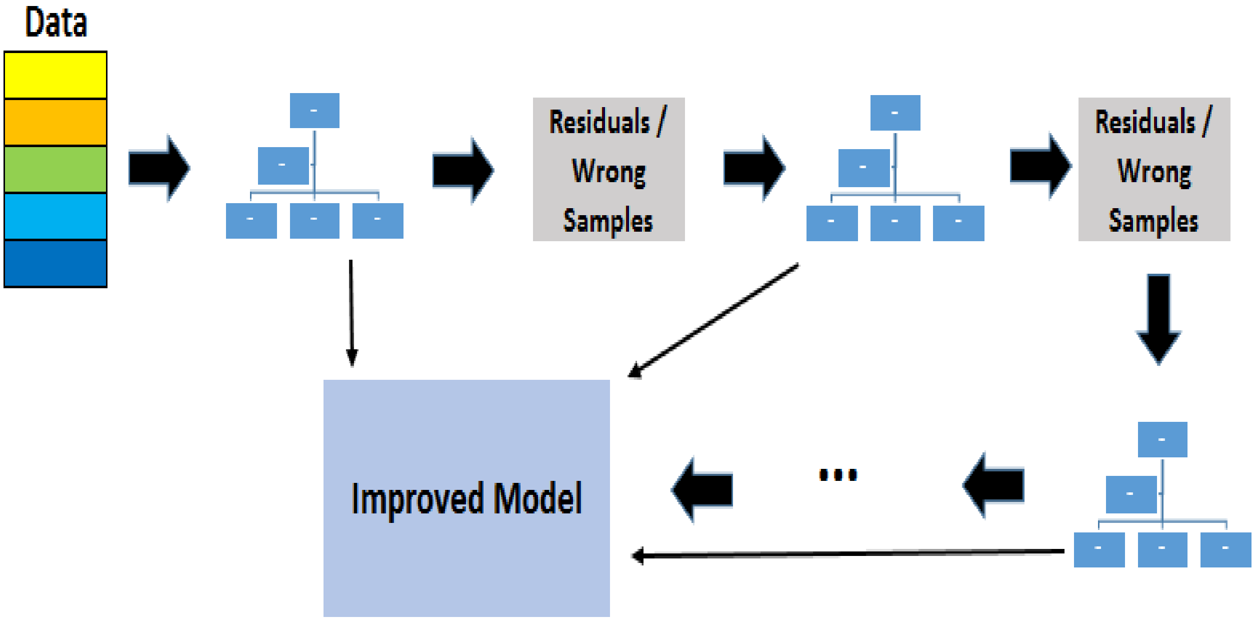


Fig 3.6 Gradient Boost Classification

4.RANDOM FOREST CLASSIFICATION MODEL

Random forest is a flexible, easy to use machine learning algorithm that produces, even without hyper-parameter tuning, a great result most of the time. It is also one of the most used algorithms, because of its simplicity and diversity (it can be used for both classification and regression tasks).

Random forest is a supervised learning algorithm. The "forest" it builds, is an ensemble of decision trees, usually trained with the “bagging” method. The general idea of the bagging method is that a combination of learning models increases the overall result. Random forest has nearly the same hyperparameters as a decision tree or a bagging classifier. Fortunately, there's no need to combine a decision tree with a bagging classifier because you can easily use the classifier-class of random forest. With random forest, you can also deal with regression tasks by using the algorithm's regressor. Random forest adds additional randomness to the model, while growing the trees. Instead of searching for the most important feature while splitting a node, it searches for the best feature among a random subset of features. This results in a wide diversity that generally results in a better model. Therefore, in random forest, only a random subset of the features is taken into consideration by the algorithm for splitting a node. You can even make trees more random by additionally using random thresholds for each feature rather than searching for the best possible thresholds (like a normal decision tree does).

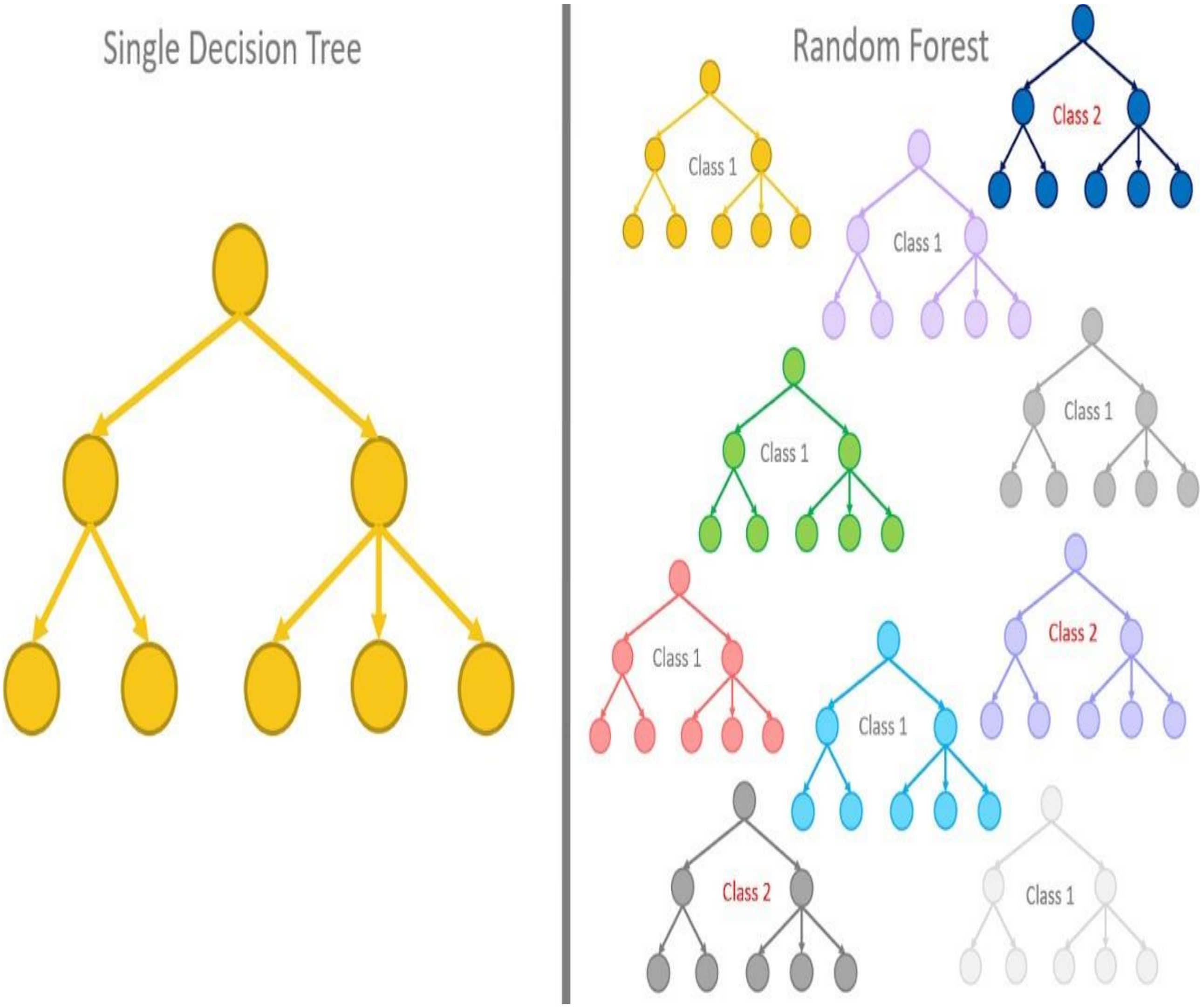


Fig 3.7 Random Forest Classification Model

Table 3.1 Difference Between Random Forest Classifier and Linear Regression

Feature	**Random Forest Classifier**	**Linear Regression**
Type of algorithm	Ensemble learning method	Supervised learning method
Type of problem	Classification	Regression
Model complexity	High	Low
Interpretability	Low	High
Handling non-linearity	Yes	No
Handling multicollinearity	Yes	No
Overfitting	Less likely	More likely
Performance on small datasets	Less effective	Effective
Performance on large datasets	Effective	May encounter computational issues
Sensitivity to outliers	Less sensitive	Sensitive
Training time	Longer	Shorter
Prediction time	Faster	Slower

Table 3.2 Difference Between Gradient Boosting Classifier and Decision Tree Classifier

Feature	**Gradient Boosting Classifier**	**Decision Tree Classifier**
Type of algorithm	Ensemble learning method	Single decision tree
Type of problem	Classification	Classification
Model complexity	High	Low
Interpretability	Low	Moderate
Handling non-linearity	Yes	Yes
Handling multicollinearity	Yes	No
Overfitting	Less likely	More likely
Performance on small datasets	Less effective	Effective
Performance on large datasets	Effective	May encounter overfitting if not pruned
Sensitivity to outliers	Less sensitive	Sensitive
Training time	Longer	Shorter
Prediction time	Faster	Faster

USED PYTHON PACKAGES:

1. **sklearn :**

    a. In python, sklearn is a machine learning package which include a lot of ML algorithms.
    b. Here, we are using some of its modules like train_test_split, Decision Tree Classifier and accuracy score.

2. NumPy :

a. It is a numeric python module which provides fast math functions for calculations.
b. It is used to read data in numpy arrays and for manipulation purpose.

3. Pandas :

a. Used to read and write different files.
b. Data manipulation can be done easily with datasets.

Implementation Details

(1) Collect the data set and save it in the required directory.

(2) Launch jupyterlab through anaconda prompt.

(3) Import all the modules required for extracting details for detection of fake news.

(4) Analyze the data and use the TfidfVectorizer to detect the frequency of words.

(5) Machine learning algorithms such as Linear Regression, Decision Tree classification, Gradient boost classification and Random Forest classification are used to detect the accuracy of

the dataset.

(6) Lastly the user can check the authenticity of the provided information in the data set as to whether it is real or fake.

(7) Hence, the project is created.

TOOLS USED

(1) Python
(2) Conda
(3) jupyterLab
(4) sklearn
(5) NumPy
(6) Pandas

CHAPTER 4

TESTING AND DISCUSSION

- Algorithm’s accuracy depends on the type and size of your dataset. More the data, more chances of getting correct accuracy.
- Machine learning depends on the variations and relations
- Understanding what is predictable is as important as trying to predict it.
- While making algorithm choice, speed should be a consideration factor.

4.1 REQUIREMENT ANALYSIS

Requirement analysis, also called requirement engineering, is the process of determining user expectations for a new modified product. It encompasses the tasks that determine the need for analyzing, documenting, validating and managing software or system requirements. The requirements should be documentable, actionable, measurable, testable and traceable related to identified business needs or opportunities and define to a level of detail, sufficient for system design.

4.2 FUNCTIONAL REQUIREMENTS

It is a technical specification requirement for the software products. It is the first step in the requirement analysis process which lists the requirements of particular software systems including functional, performance and security requirements.

4.2.1 SYSTEM TESTING

Usability

It specifies how easy the system must be use. It is easy to ask queries in any format which is short or long, porter stemming algorithm stimulates the desired response for user. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple.

Robustness

It refers to a program that performs well not only under ordinary conditions but also under unusual conditions. It is the ability of the user to cope with errors for irrelevant queries during execution.

Security

The state of providing protected access to resource is security. The system provides good security and unauthorized users cannot access the system there by providing high security.

Reliability

It is the probability of how often the software fails. The measurement is often expressed in MTBF (Mean Time Between Failures). The requirement is needed in order to ensure that the processes work correctly and completely without being aborted. It can handle any load and survive and survive and even capable of working around any failure.

Compatibility

It is supported by version above all web browsers. Using any web servers like localhost makes the system real-time experience.

Flexibility

The flexibility of the project is provided in such a way that is has the ability to run on different environments being executed by different users.

Safety

Safety is a measure taken to prevent trouble. Every query is processed in a secured manner without letting others to know one’s personal information. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user.

4.3 NON- FUNCTIONAL REQUIREMENTS

4.3.1 Portability

It is the usability of the same software in different environments. The project can be run in any operating system.

4.3.2 Performance

These requirements determine the resources required, time interval, throughput and everything that deals with the performance of the system.

4.3.3 Accuracy

The result of the requesting query is very accurate and high speed of retrieving information. The degree of security provided by the system is high and effective.

4.3.4 Maintainability

Project is simple as further updates can be easily done without affecting its stability. Maintainability basically defines that how easy it is to maintain the system. It means that how easy it is to maintain the system, analyze, change and test the application. Maintainability of this project is simple as further updates can be easily done without affecting its stability.

4.4 SYSTEM DESIGN AND TESTING PLAN

4.4.1 INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

The output form of an information system should accomplish one or more of the following objectives.

- Convey information about past activities, current status or projections of the
- Future.
- Signal important events, opportunities, problems, or warnings.
- Trigger an action.
- Confirm an action.

SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

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 input data to the system, various processing carried out on this data, and the output data is generated by this system.
- The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
- DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
- DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.
- It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration.

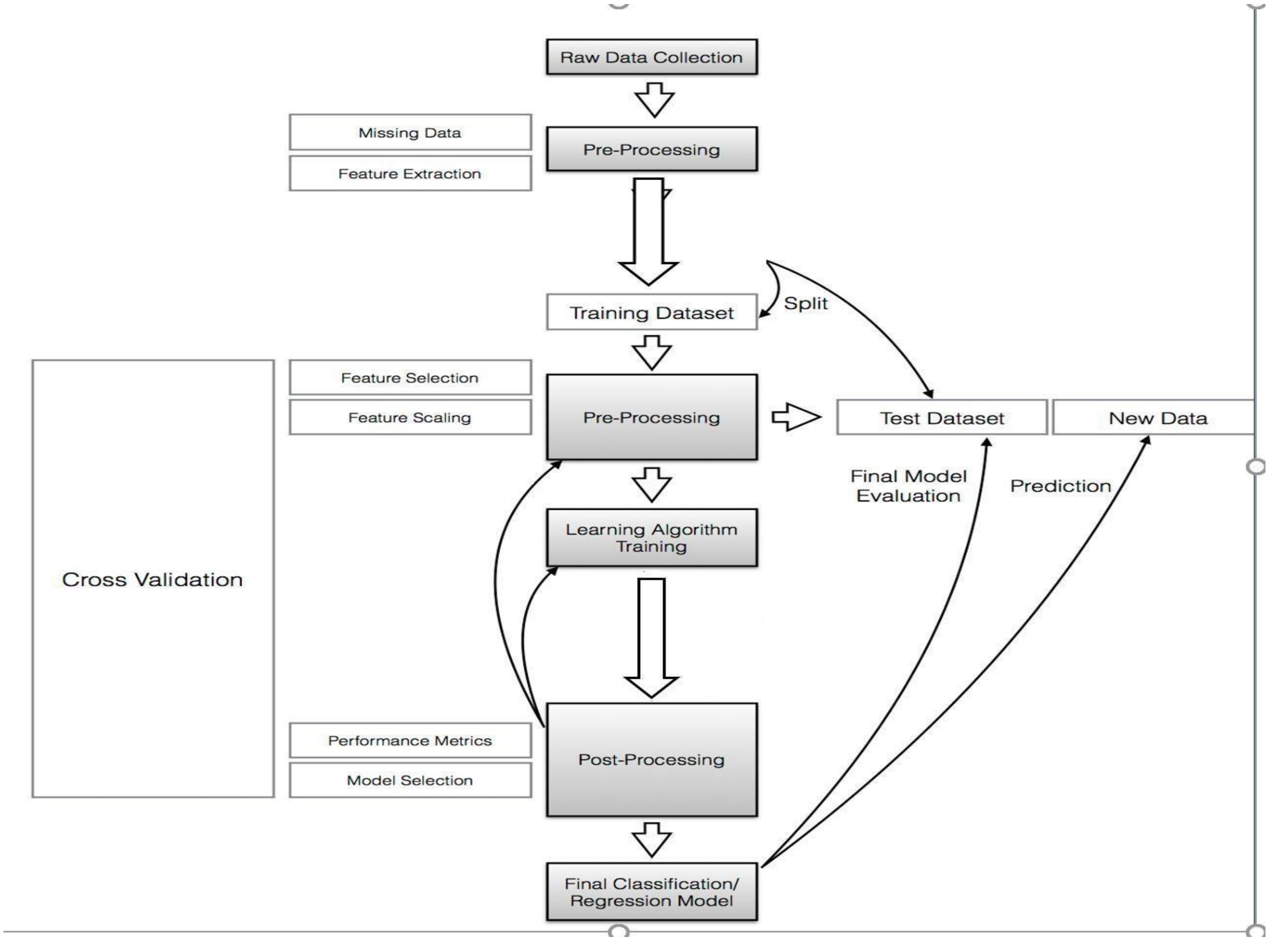


Fig:4.1 Data Flow Diagram

4.5 TEST PROCEDURE

4.5.1 SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying todiscover 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.

4.5.2 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 .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.

4.5.3 INTEGRATION TESTING

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

4.5.4 FUNCTIONAL TESTING

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.

Function: identified functions must be exercised.

Output: identified classes of application outputs must be exercised.

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.

4.5.5 WHITE BOX TESTING

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level. Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box. you cannot "see" into it. The test provides inputs and responds to outputs without considering how the software works

4.5.6 ACCEPTANCE TESTING

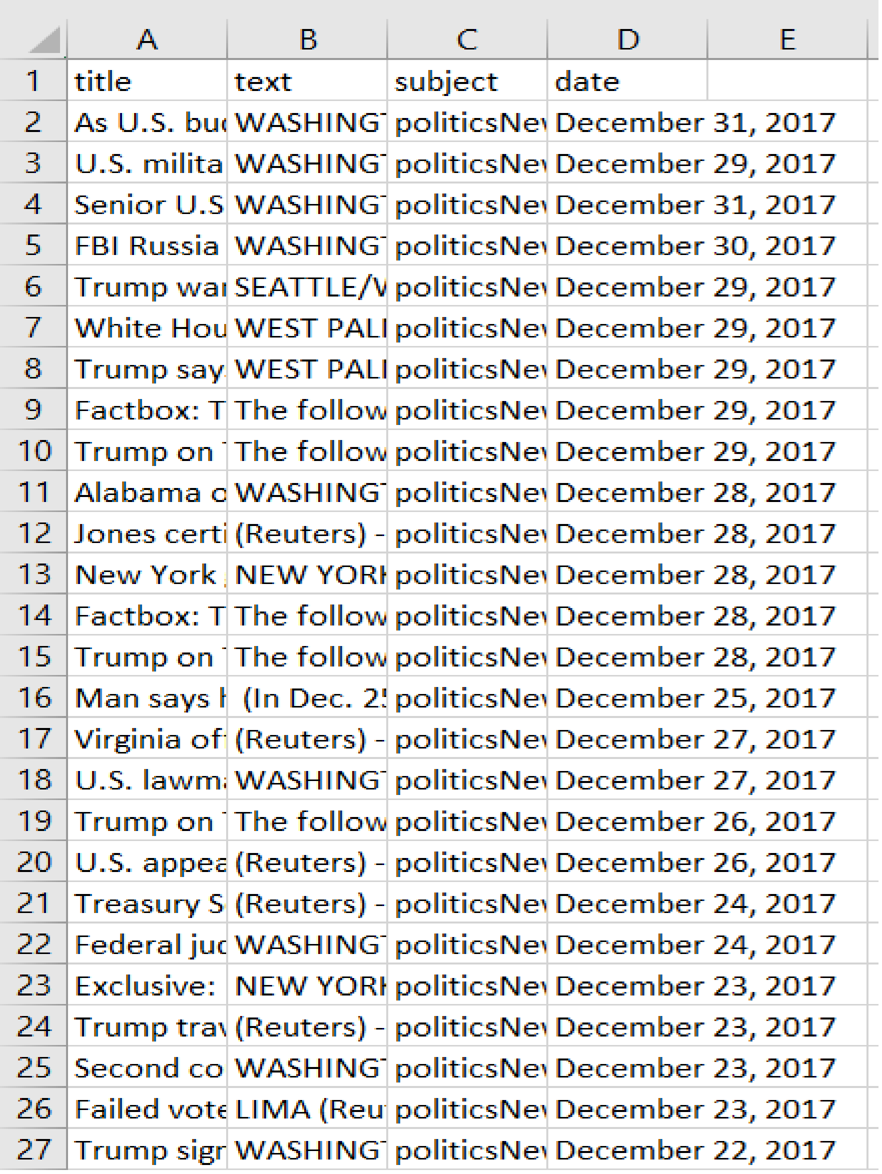
User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

CHAPTER 5

RESULTS AND VERIFICATION

Two datasets containing fake news and true news will be saved in the directory during the initial part of doing the project.

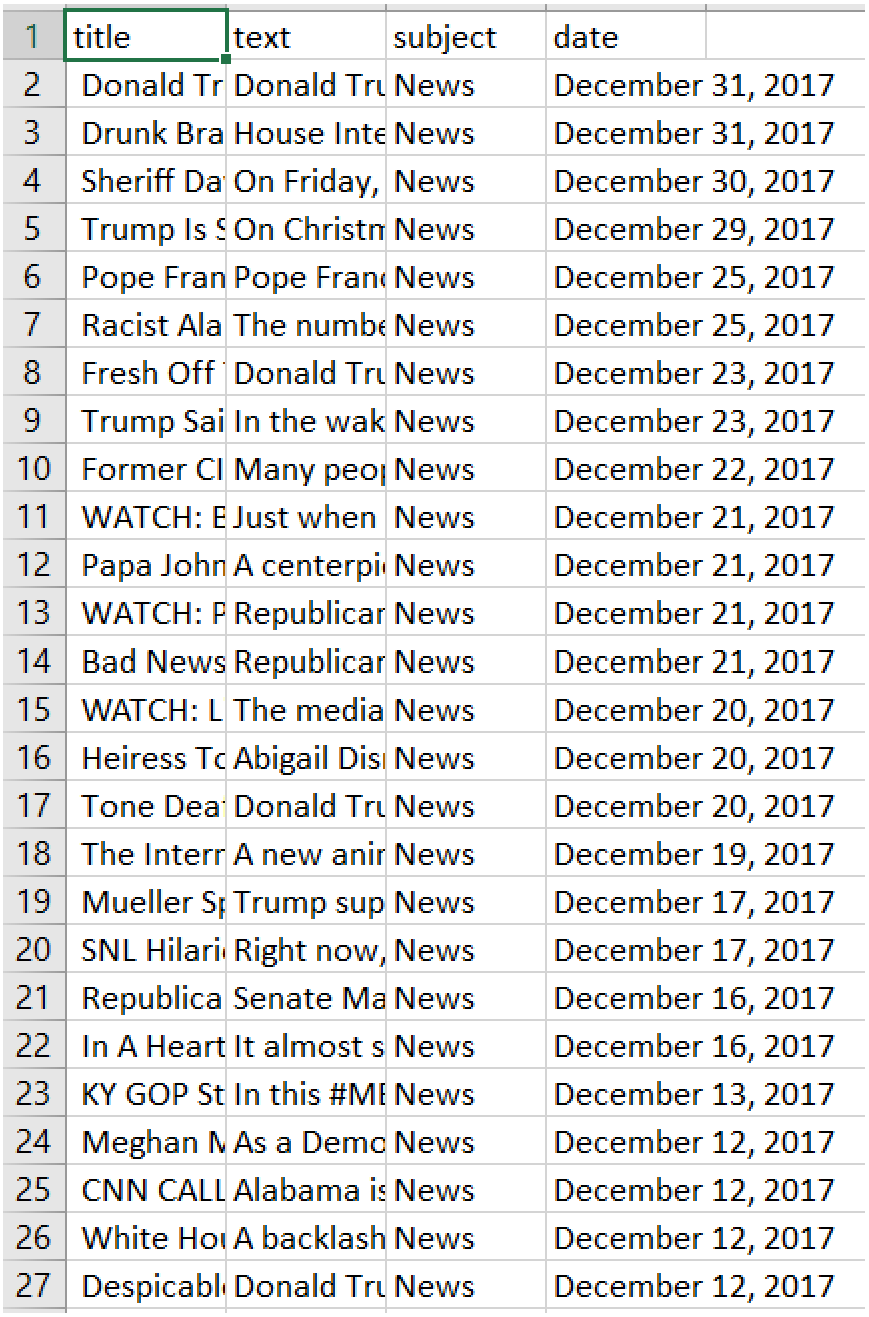
(A) True news



	A	B	C	D	E
1	title	text	subject	date	
2	As U.S. bu	WASHING	politicsNe	December 31, 2017	
3	U.S. milita	WASHING	politicsNe	December 29, 2017	
4	Senior U.S	WASHING	politicsNe	December 31, 2017	
5	FBI Russia	WASHING	politicsNe	December 30, 2017	
6	Trump wa	SEATTLE/V	politicsNe	December 29, 2017	
7	White Hou	WEST PAL	politicsNe	December 29, 2017	
8	Trump say	WEST PAL	politicsNe	December 29, 2017	
9	Factbox: T	The follow	politicsNe	December 29, 2017	
10	Trump on	The follow	politicsNe	December 29, 2017	
11	Alabama c	WASHING	politicsNe	December 28, 2017	
12	Jones certi	(Reuters) -	politicsNe	December 28, 2017	
13	New York	NEW YOR	politicsNe	December 28, 2017	
14	Factbox: T	The follow	politicsNe	December 28, 2017	
15	Trump on	The follow	politicsNe	December 28, 2017	
16	Man says	(In Dec. 2	politicsNe	December 25, 2017	
17	Virginia of	(Reuters) -	politicsNe	December 27, 2017	
18	U.S. lawm	WASHING	politicsNe	December 27, 2017	
19	Trump on	The follow	politicsNe	December 26, 2017	
20	U.S. appea	(Reuters) -	politicsNe	December 26, 2017	
21	Treasury S	(Reuters) -	politicsNe	December 24, 2017	
22	Federal ju	WASHING	politicsNe	December 24, 2017	
23	Exclusive:	NEW YOR	politicsNe	December 23, 2017	
24	Trump tra	(Reuters) -	politicsNe	December 23, 2017	
25	Second co	WASHING	politicsNe	December 23, 2017	
26	Failed vote	LIMA (Reu	politicsNe	December 23, 2017	
27	Trump sigr	WASHING	politicsNe	December 22, 2017	

Fig.5.1 True news dataset

(B) Fake News



1	title	text	subject	date
2	Donald Tr	Donald Tru	News	December 31, 2017
3	Drunk Bra	House Inte	News	December 31, 2017
4	Sheriff Da	On Friday,	News	December 30, 2017
5	Trump Is S	On Christm	News	December 29, 2017
6	Pope Fran	Pope Franc	News	December 25, 2017
7	Racist Ala	The numbe	News	December 25, 2017
8	Fresh Off T	Donald Tru	News	December 23, 2017
9	Trump Sai	In the wak	News	December 23, 2017
10	Former CI	Many peop	News	December 22, 2017
11	WATCH: E	Just when	News	December 21, 2017
12	Papa John	A centerpi	News	December 21, 2017
13	WATCH: P	Republicar	News	December 21, 2017
14	Bad News	Republicar	News	December 21, 2017
15	WATCH: L	The media	News	December 20, 2017
16	Heiress Tc	Abigail Dis	News	December 20, 2017
17	Tone Dea	Donald Tru	News	December 20, 2017
18	The Interr	A new anir	News	December 19, 2017
19	Mueller Sp	Trump sup	News	December 17, 2017
20	SNL Hilari	Right now,	News	December 17, 2017
21	Republica	Senate Ma	News	December 16, 2017
22	In A Heart	It almost s	News	December 16, 2017
23	KY GOP St	In this #ME	News	December 13, 2017
24	Meghan M	As a Demc	News	December 12, 2017
25	CNN CALL	Alabama is	News	December 12, 2017
26	White Hou	A backlash	News	December 12, 2017
27	Despicabl	Donald Tru	News	December 12, 2017

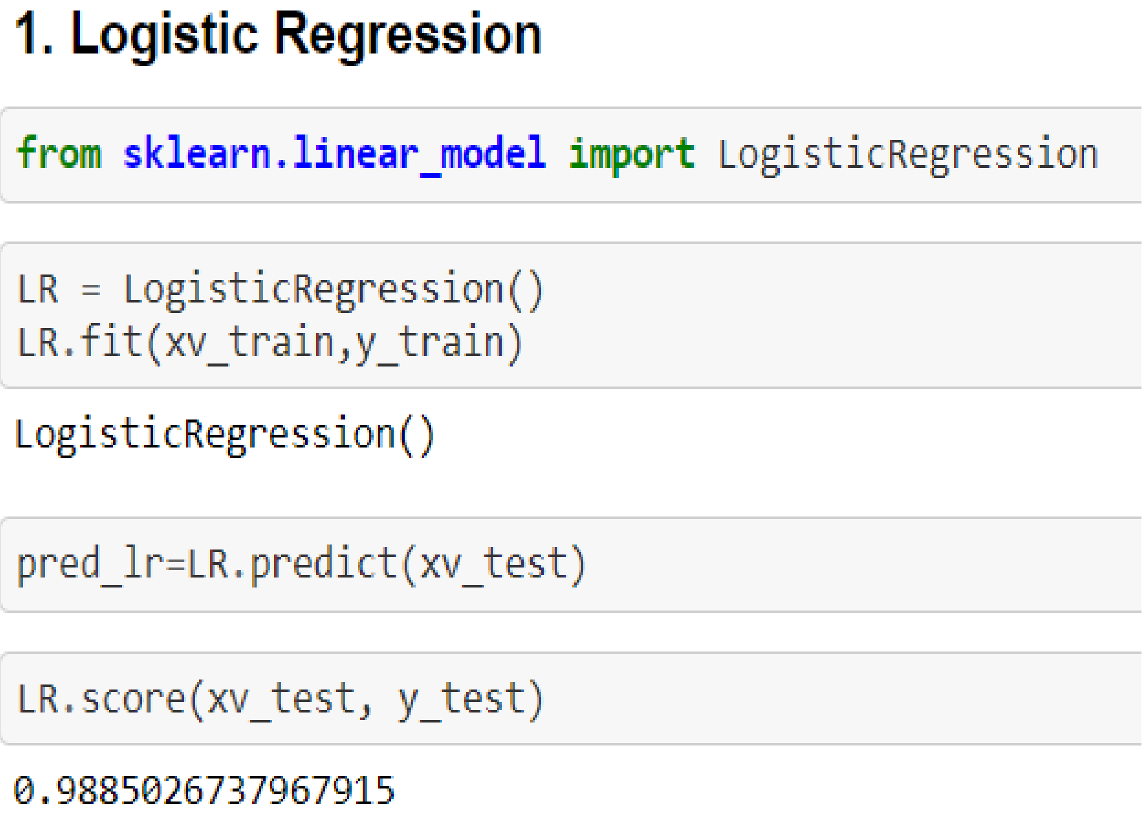
Fig.5.2 Fake news dataset

The next procedure to be followed is to launch jupyterlab using anaconda prompt. The first step in jupyterlab is to import all the necessary libraries.

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report
import re
import string
```

After the above process we insert the fake and true dataset.

Then after the insertion of these datasets Machine learning concepts starts following for testing the accuracy of datasets. The testing process goes by the following manner:



1. Logistic Regression

```
from sklearn.linear_model import LogisticRegression
```

```
LR = LogisticRegression()
LR.fit(xv_train,y_train)
```

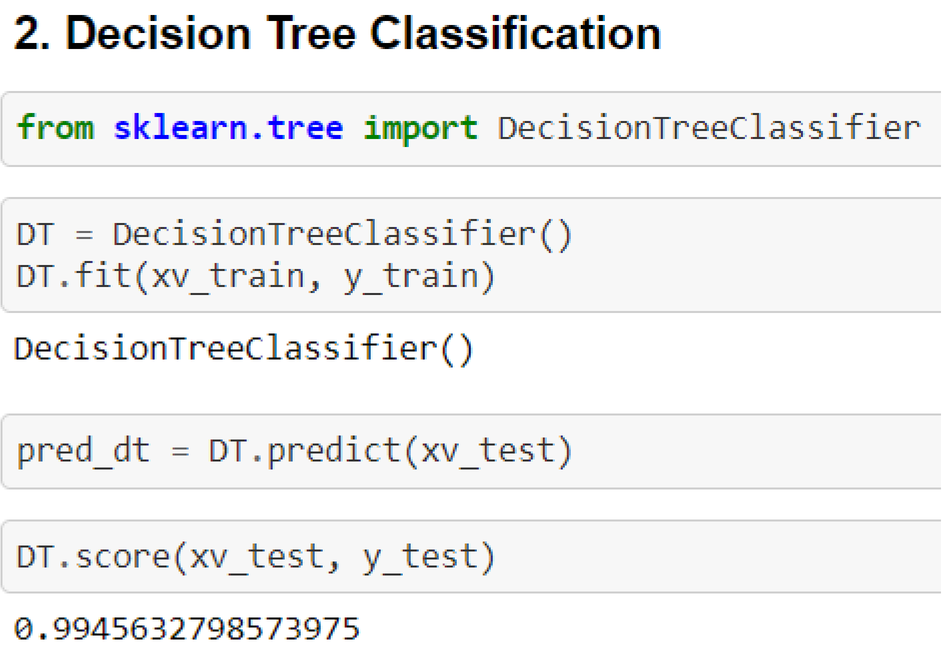
```
LogisticRegression()
```

```
pred_lr=LR.predict(xv_test)
```

```
LR.score(xv_test, y_test)
```

```
0.9885026737967915
```

Fig 5.3 Logistic Regression



2. Decision Tree Classification

```
from sklearn.tree import DecisionTreeClassifier
```

```
DT = DecisionTreeClassifier()
DT.fit(xv_train, y_train)
```

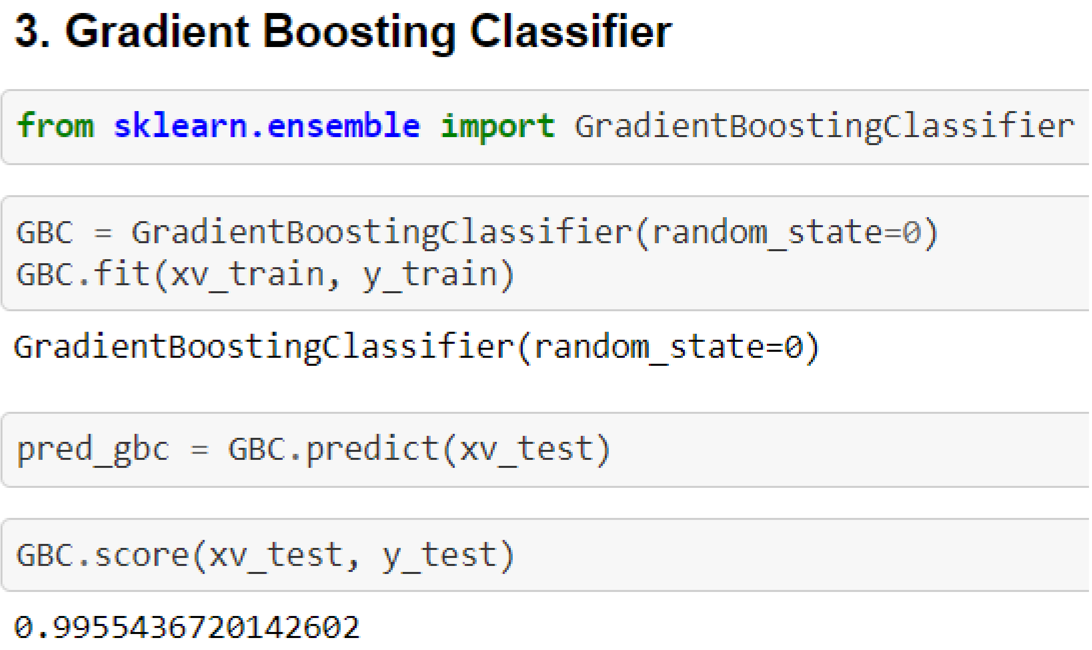
```
DecisionTreeClassifier()
```

```
pred_dt = DT.predict(xv_test)
```

```
DT.score(xv_test, y_test)
```

```
0.9945632798573975
```

Fig 5.4 Decision Tree Classification



3. Gradient Boosting Classifier

```
from sklearn.ensemble import GradientBoostingClassifier
```

```
GBC = GradientBoostingClassifier(random_state=0)
GBC.fit(xv_train, y_train)
```

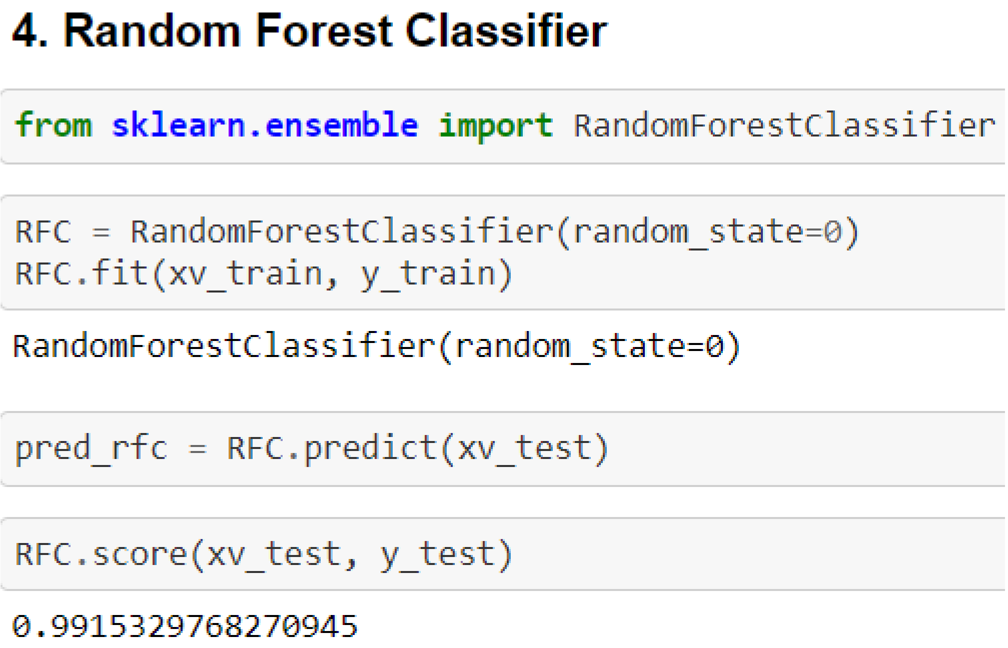
```
GradientBoostingClassifier(random_state=0)
```

```
pred_gbc = GBC.predict(xv_test)
```

```
GBC.score(xv_test, y_test)
```

```
0.9955436720142602
```

Fig 5.5 Gradient Boosting Classifier



4. Random Forest Classifier

```
from sklearn.ensemble import RandomForestClassifier
```

```
RFC = RandomForestClassifier(random_state=0)
RFC.fit(xv_train, y_train)
```

```
RandomForestClassifier(random_state=0)
```

```
pred_rfc = RFC.predict(xv_test)
```

```
RFC.score(xv_test, y_test)
```

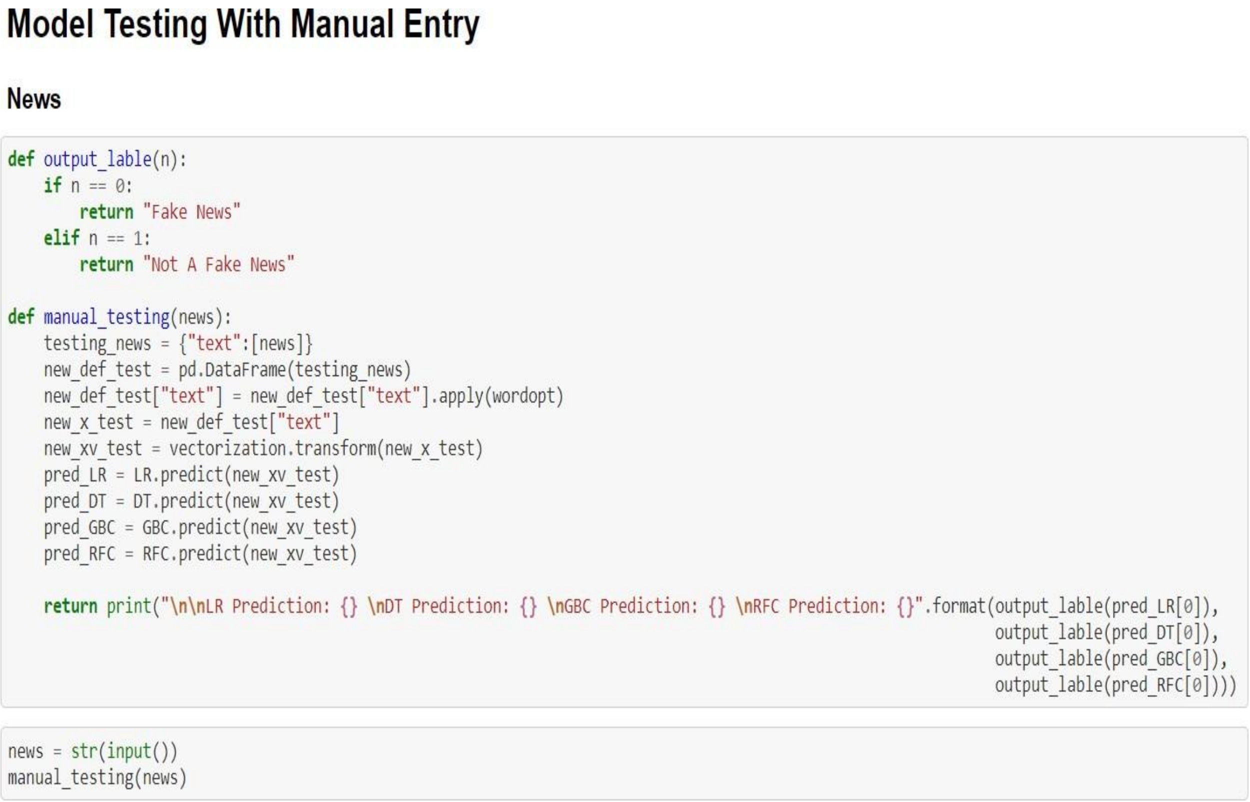
```
0.9915329768270945
```

Fig 5.6 Random Forest Classifier

The accuracy level of the datasets has been successfully verified by the above-mentioned concepts and the accuracy score is pretty much good.

5.1 VERIFICATION

After the above-mentioned testing is done we proceed further for the manual testing wherein the user can input the details and know whether that particular news is fake or not.



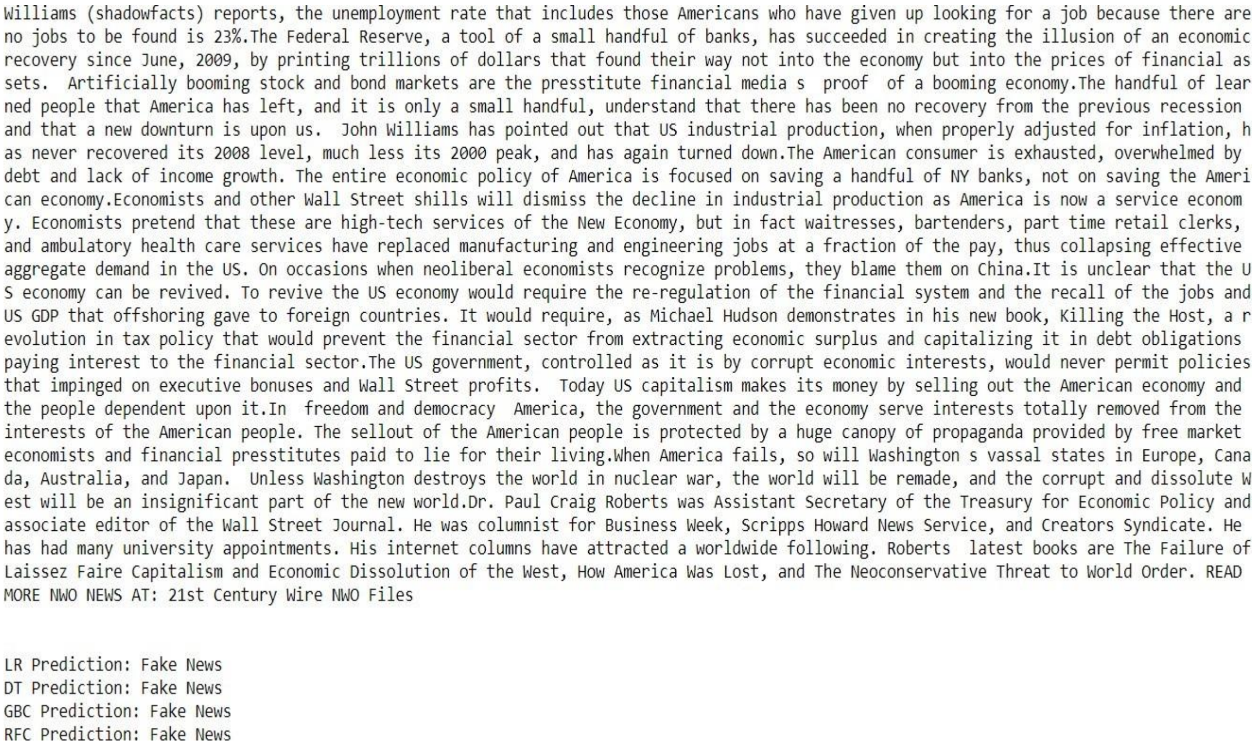
Model Testing With Manual Entry

News

```
def output_lable(n):
    if n == 0:
        return "Fake News"
    elif n == 1:
        return "Not A Fake News"

def manual_testing(news):
    testing_news = {"text":[news]}
    new_def_test = pd.DataFrame(testing_news)
    new_def_test["text"] = new_def_test["text"].apply(wordopt)
    new_x_test = new_def_test["text"]
    new_xv_test = vectorization.transform(new_x_test)
    pred_LR = LR.predict(new_xv_test)
    pred_DT = DT.predict(new_xv_test)
    pred_GBC = GBC.predict(new_xv_test)
    pred_RFC = RFC.predict(new_xv_test)

    return print("\n\nLR Prediction: {} \nDT Prediction: {} \nGBC Prediction: {} \nRFC Prediction: {}".format(output_lable(pred_LR[0]),
                                                                                                              output_lable(pred_DT[0]),
                                                                                                              output_lable(pred_GBC[0]),
                                                                                                              output_lable(pred_RFC[0])))
```

```
news = str(input())
manual_testing(news)
```

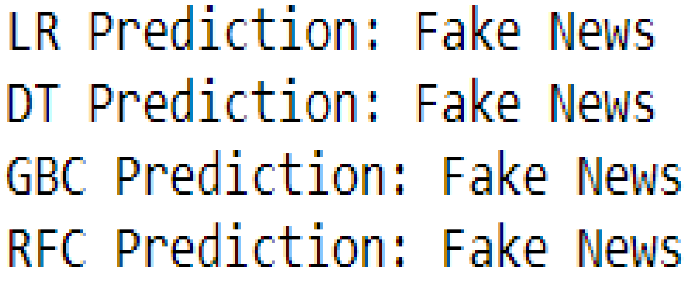
Fig 5.7 Verification (Input)



```
Williams (shadowfacts) reports, the unemployment rate that includes those Americans who have given up looking for a job because there are
no jobs to be found is 23%.The Federal Reserve, a tool of a small handful of banks, has succeeded in creating the illusion of an economic
recovery since June, 2009, by printing trillions of dollars that found their way not into the economy but into the prices of financial as
sets.  Artificially booming stock and bond markets are the presstitute financial media s  proof  of a booming economy.The handful of lear
ned people that America has left, and it is only a small handful, understand that there has been no recovery from the previous recession
and that a new downturn is upon us.  John Williams has pointed out that US industrial production, when properly adjusted for inflation, h
as never recovered its 2008 level, much less its 2000 peak, and has again turned down.The American consumer is exhausted, overwhelmed by
debt and lack of income growth. The entire economic policy of America is focused on saving a handful of NY banks, not on saving the Ameri
can economy.Economists and other Wall Street shills will dismiss the decline in industrial production as America is now a service econom
y. Economists pretend that these are high-tech services of the New Economy, but in fact waitresses, bartenders, part time retail clerks,
and ambulatory health care services have replaced manufacturing and engineering jobs at a fraction of the pay, thus collapsing effective
aggregate demand in the US. On occasions when neoliberal economists recognize problems, they blame them on China.It is unclear that the U
S economy can be revived. To revive the US economy would require the re-regulation of the financial system and the recall of the jobs and
US GDP that offshoring gave to foreign countries. It would require, as Michael Hudson demonstrates in his new book, Killing the Host, a r
evolution in tax policy that would prevent the financial sector from extracting economic surplus and capitalizing it in debt obligations
paying interest to the financial sector.The US government, controlled as it is by corrupt economic interests, would never permit policies
that impinged on executive bonuses and Wall Street profits.  Today US capitalism makes its money by selling out the American economy and
the people dependent upon it.In  freedom and democracy  America, the government and the economy serve interests totally removed from the
interests of the American people. The sellout of the American people is protected by a huge canopy of propaganda provided by free market
economists and financial presstitutes paid to lie for their living.When America fails, so will Washington s vassal states in Europe, Cana
da, Australia, and Japan.  Unless Washington destroys the world in nuclear war, the world will be remade, and the corrupt and dissolute W
est will be an insignificant part of the new world.Dr. Paul Craig Roberts was Assistant Secretary of the Treasury for Economic Policy and
associate editor of the Wall Street Journal. He was columnist for Business Week, Scripps Howard News Service, and Creators Syndicate. He
has had many university appointments. His internet columns have attracted a worldwide following. Roberts  latest books are The Failure of
Laissez Faire Capitalism and Economic Dissolution of the West, How America Was Lost, and The Neoconservative Threat to World Order. READ
MORE NWO NEWS AT: 21st Century Wire NWO Files


LR Prediction: Fake News
DT Prediction: Fake News
GBC Prediction: Fake News
RFC Prediction: Fake News
```

Fig 5.8 Verification (Output)

So here in the above provided image you can see the output for which the input was provided by the user that is me. So now the machine learning concepts have verifies the data thoroughly and has undergone so many algorithmic verifications wherein the result provided was that the data is absolutely fake which means that it is a fake news.



```
LR Prediction: Fake News
DT Prediction: Fake News
GBC Prediction: Fake News
RFC Prediction: Fake News
```

Fig 5.9 Results Showing Fake News

5.2 RESULT

The project on Detection of Fake news has been executed successfully using python programming language.

Machine learning algorithms have been used to verify the data accuracy and hence the results were the proven fact for it to have an excellent accuracy rate.

After the implementation of the above-mentioned processes a manual testing was done wherein the user was supposed to input the data and hence the result proved to provide the accurate information about its authenticity as to whether it is a fake news or real news. Then upon checking the news seemed to be absolutely fake.

Hence the project Fake News Detection has been implemented, executed and verified successfully.

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

CONCLUSION:

Many people consume news from social media instead of traditional news media. However, social media has also been used to spread fake news, which has negative impacts on individual people and society. In this paper, an innovative model for fake news detection using machine learning algorithms has been presented. This model takes news events as an input and based on twitter reviews and classification algorithms it predicts the percentage of news being fake or real.

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential. This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus, the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

FUTURE SCOPE:

Create a separate model for each genre of news that is spread worldwide.

Create new features from various other complicated version of tools.

Furthermore, data sets can also be added which will become a massive dataset and fake news can be detected through it.

This kind of project can also be used in certain organizations to detect the spread of false information that will help in not misleading the clients.

APPENDIX

SOURCE CODE

Fake News Detection

Importing Libraries

```
import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split

from sklearn.metrics import accuracy_score

from sklearn.metrics import classification_report

import re

import string
```

Importing Dataset

```
df_fake = pd.read_csv("Fake.csv")

df_true = pd.read_csv("True.csv")

df_fake.head()

df_true.head(5)
```

Inserting a column "class" as target feature

```
df_fake["class"] = 0

df_true["class"] = 1

df_fake.shape, df_true.shape

# Removing last 10 rows for manual testing
```

```
df_fake_manual_testing = df_fake.tail(10)

for i in range(23480,23470,-1):

df_fake.drop([i], axis = 0, inplace = True)

df_true_manual_testing = df_true.tail(10)

for i in range(21416,21406,-1):

df_true.drop([i], axis = 0, inplace = True)

df_fake.shape, df_true.shape

df_fake_manual_testing["class"] = 0

df_true_manual_testing["class"] = 1

df_fake_manual_testing.head(10)

df_true_manual_testing.head(10)

df_manual_testing = pd.concat([df_fake_manual_testing,df_true_manual_testing], axis = 0)

df_manual_testing.to_csv("manual_testing.csv")
```

Merging True and Fake Dataframes

```
df_merge = pd.concat([df_fake, df_true], axis =0 )

df_merge.head(10)

df_merge.columns
```

Removing columns which are not required

```
df = df_merge.drop(["title", "subject","date"], axis = 1)

df.isnull().sum()
```

Random Shuffling the dataframe

```
df = df.sample(frac = 1)

df.head()
df.reset_index(inplace = True)

df.drop(["index"], axis = 1, inplace = True)

df.columns

df.head()
```

Creating a function to process the texts

```
def wordopt(text):

text = text.lower()

text = re.sub('\[.*?\]', '', text)

text = re.sub("\\W"," ",text)

text = re.sub('https?://\S+|www\.\S+', '', text)

text = re.sub('<.*?>+', '', text)

text = re.sub('[%s]' % re.escape(string.punctuation), '', text)

text = re.sub('\n', '', text)

text = re.sub('\w*\d\w*', '', text)

return text

df["text"] = df["text"].apply(wordopt)
```

Defining dependent and independent variables

```
x = df["text"]

y = df["class"]
```

Splitting Training and Testing

```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.25)
```

Convert text to vectors

```
from sklearn.feature_extraction.text import TfidfVectorizer
```

```
vectorization = TfidfVectorizer()
xv_train = vectorization.fit_transform(x_train)
xv_test = vectorization.transform(x_test)
```

Logistic Regression

```
from sklearn.linear_model import LogisticRegression


LR = LogisticRegression()
LR.fit(xv_train,y_train)
pred_lr=LR.predict(xv_test)
LR.score(xv_test, y_test)
print(classification_report(y_test, pred_lr))
```

Decision Tree Classification

```
from sklearn.tree import DecisionTreeClassifier
D T = DecisionTreeClassifier()
DT.fit(xv_train, y_train)
pred_dt = DT.predict(xv_test)
DT.score(xv_test, y_test)
print(classification_report(y_test, pred_dt))
```

Gradient Boosting Classifier

```
from sklearn.ensemble import GradientBoostingClassifier


GBC = GradientBoostingClassifier(random_state=0)
```

```
GBC.fit(xv_train, y_train)
pred_gbc = GBC.predict(xv_test)
GBC.score(xv_test, y_test)
print(classification_report(y_test, pred_gbc))
```

Random Forest Classifier

```
from sklearn.ensemble import RandomForestClassifier


RFC = RandomForestClassifier(random_state=0)
RFC.fit(xv_train, y_train)
pred_rfc = RFC.predict(xv_test)
RFC.score(xv_test, y_test)
print(classification_report(y_test, pred_rfc))
```

Model Testing

```
def output_lable(n):
if n == 0:
return "Fake News"
elif n == 1:
return "Not A Fake News"
def manual_testing(news):
testing_news = {"text":[news]}
new_def_test = pd.DataFrame(testing_news)
new_def_test["text"] = new_def_test["text"].apply(wordopt)
```

```
new_x_test = new_def_test["text"]
new_xv_test = vectorization.transform(new_x_test)
pred_LR = LR.predict(new_xv_test)
pred_DT = DT.predict(new_xv_test)
pred_GBC = GBC.predict(new_xv_test)
pred_RFC = RFC.predict(new_xv_test)

return print("\n\nLR Prediction: {} \nDT Prediction: {} \nGBC Prediction: {} \nRFC Prediction:
{}".format(output_lable(pred_LR[0]),
output_lable(pred_DT[0]),
output_lable(pred_GBC[0]),
output_lable(pred_RFC[0])))
news = "Vidya is America's first women president"#str(input())
manual_testing(news)
news = "Trump is one of America's President"
manual_testing(news)
news = "SEATTLE/WASHINGTON (Reuters) - President Donald Trump called on the U.S.
Postal Service "
manual_testing(news)
```

REFERENCES

[1]. Parikh, S. B., & Atrey, P. K. (2018, April). Media-Rich Fake News Detection: A Survey. In 2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR) (pp. 436-441). IEEE.

[2]. Conroy, N. J., Rubin, V. L., & Chen, Y. (2015, November). Automatic deception detection: Methods for finding fake news. In Proceedings of the 78th ASIS&T Annual Meeting: Information Science with Impact: Research in and for the Community (p. 82). American Society for Information Science.

[3]. Helmstetter, S., & Paulheim, H. (2018, August). Weakly supervised learning for fake news detection on Twitter. In 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM) (pp. 274-277). IEEE. [4]. Stahl, K. (2018). Fake News Detection in Social Media.

[5]. Della Vedova, M. L., Tacchini, E., Moret, S., Ballarin, G., DiPierro, M., & de Alfaro, L. (2018, May). Automatic Online Fake News Detection Combining Content and Social Signals. In 2018 22nd Conference of Open Innovations Association (FRUCT) (pp. 272-279). IEEE.

[6] Tacchini, E., Ballarin, G., Della Vedova, M. L., Moret, S., & de Alfaro, L. (2017). Some like it hoax: Automated fake news detection in social networks. arXiv preprint arXiv:1704.07506.

[7]. Shao, C., Ciampaglia, G. L., Varol, O., Flammini, A., & Menczer, F. (2017). The spread of fake news by social bots. arXiv preprint arXiv:1707.07592, 96-104.

[8]. Chen, Y., Conroy, N. J., & Rubin, V. L. (2015, November). Misleading online content: Recognizing clickbait as false news. In Proceedings of the 2015 ACM on Workshop on Multimodal Deception Detection (pp. 15-19). ACM.

[9]. Najafabadi, M. M., Villanustre, F., Khoshgoftaar, T. M., Seliya, N., Wald, R., & Muharemagic, E. (2015). Deep learning applications and challenges in big data analytics. Journal of Big Data, 2(1), 1.

[10]. Haiden, L., & Althuis, J. (2018). The Definitional Challenges of Fake News.

Project Title: **Fake News Detection Using Python**

Guide(s): **Ms. B. Deepika Rathod** (Associate Prof)

Student Name(s): **Tallada Sai Sharan**

Shaik Mohammed Aleemuddin

Allamla Srikanth

Katta Sri Lakshmi Prasanna

Academic Year: 2023-24

Name of Course from which Principles are applied in this project	**Related Course Outcome Number**	**Description of the application**	**Page Number**	**Attained PO**
Algorithm Design & Analysis	-	Statistical data analysis, sorting & processing		PO1, PO2, PO3, PO4, PO5, PO9, PO10, PSO1, PSO2, PS03
Machine Learning	-	Decision Making		PO1, PO2, PO3, PO4, PO5, PO6, PO7, PSO1, PSO2, PSO3
Python	-	Robust, Extensive Libraries		PO1, PO2, PO3, PO5, PO6, PO7, PSO1, PSO2, PSO3, PO9, PO10
Prediction Analytics	-	Data cleaning & Imputations, classifications, tagging		PO1, PO2, PO3, PO4, PO5, PO9, PO10, PSO1, PSO2, PSO3
Data Visualization Techniques	-	Pattern recognition, data quality. comparative analysis, data distribution, decision support		POI, PO2, PO3, PO4, PO5, PO9, PO10, PSO1, PSO2, PS03

Guide Signature