

FAKE NEWS DETECTION WITH REAL NEWS GENERATION

A PROJECT REPORT

Submitted by

**SRUTHI VS [Reg No: RA1711008010235]
SNEHAL NAIR [Reg No: RA1711008010245]
MERWIN ROY [Reg No: RA1711008010283]**

Under the guidance of

Ms. P. Nithyakani

(Assistant Professor, Department of Information Technology)

In partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY



**DEPARTMENT OF INFORMATION TECHNOLOGY
FACULTY OF ENGINEERING AND TECHNOLOGY
SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
KATTANKULATHUR – 603203**

MAY 2021

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
KATTANKULATHUR-603203

BONAFIDE CERTIFICATE

Certified that this project report titled “**FAKE NEWS DETECTION WITH REAL NEWS GENERATION**” is the bonafide work of **SRUTHI VS [RA1711008010235]**, **SNEHAL NAIR [RA1711008010245]** and **MERWIN ROY [RA1711008010283]** who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion for this or any other candidate.



MS.P. NITHYAKANI
GUIDE
Assistant Professor
Dept. of Information Technology



Dr. G. VADIVU
Head
Department of Information Technology
Faculty of Engineering & Technology
SRM Institute of Science and Technology
SRM Nagar, Kattankulathur - 603 203.
Kancheepuram Dist., Tamil Nadu, India.

Dr. G. VADIVU
HEAD OF THE DEPARTMENT
Dept. of Information Technology

Dr. Kayalvizhi Jayavel
Assistant Professor / IT
SRMIST

Dr. Angeline Geetha
Prof. & HOD, School of Computing Science
Hindustan Institute of Technology & Science

Signature of Internal Examiner

Signature of External Examiner

ABSTRACT

With increasing popularity in the use of social media for news consumption, the substantial widespread dissemination of fake news has the potential to adversely affect individuals as well as the society as a whole. Even in the midst of the current covid-19 pandemic, false information shared on websites such as WhatsApp, Twitter, and Facebook have the potential to cause panic and shock a large number of people in various parts of the world. These misconceptions obscure healthier habits and encourage incorrect procedures, which aid in the transmission of the virus and, as a result, result in poor physical and psychological health results for individuals. Therefore, it is a research challenge to validate the source, content and publisher of a news article for classifying it as genuine or fake. The existing systems and techniques are not efficient enough to accurately classify a given news based on its statistical rating. Machine learning plays an imperative part in categorizing news data and information, despite some limitations.

Our project not only aims on fake news detection but also on generation of real news once the fake news is detected. We propose a user-friendly webpage on which the user enters the news article statement. It is then tested by our machine learning algorithm which then classifies it as genuine or fake, after which the important words are extracted from the statement which helps to get the corresponding genuine news by scraping it from trusted sources and show it to the user. We have compared two machine learning algorithms in this which are- Passive Aggressive Classifier and Naïve Bayes algorithm. We got an accuracy of about 93.5% from Passive Aggressive Classifier and about 83.5% from Naïve Bayes algorithm.

ACKNOWLEDGEMENT

The success and the final outcome of this project required guidance and assistance from different sources and we feel extremely fortunate to have got this all along the completion of our project. Whatever we have done is largely due to such guidance and assistance and we would not forget to thank them.

We express our sincere thanks to the Head of the Department, Department of Information Technology, Dr. G. Vadivu, for all the help and infrastructure provided to us to complete this project successfully and her valuable guidance.

We owe our profound gratitude to our project guide **Ms. P. Nithyakani**, who took keen interest in our project work and guided us all along, till the completion of our project work by providing all the necessary information for developing a good system.

We are thankful to and fortunate enough to get constant encouragement, support and guidance from all the Teaching staff of the Department of Information Technology who helped us in successfully completing our project work. Also, we would like to extend our sincere regards to all the non-teaching staff of the department of Information Technology for their timely support.

SRUTHI VS
[Reg No: RA1711008010235]

SNEHAL NAIR
[Reg No: RA1711008010245]

MERWIN ROY
[Reg No: RA1711008010283]

TABLE OF CONTENTS

ABSTRACT	iii
LIST OF FIGURES	ix
LIST OF FLOWCHARTS	x
LIST OF GRAPHS	xi
ABBREVIATIONS	xii
1. INTRODUCTION	13
1.1. Overview	13
1.2. Motivation and Problem statement	14
1.3. Learning Techniques	15
1.4. Flow chart	15
1.5. Python libraries and Algorithms	16
1.6. Addressing ethical and social issues and responsibilities	18
2. LITERATURE SURVEY	20
3. MODULE DESCRIPTION	24
3.1 FUNCTIONAL REQUIREMENTS	24
3.1.1 WEB APPLICATION	24
3.2 NON-FUNCTIONAL REQUIREMENTS	25
3.2.1 DATASET	25
3.3 SOFTWARE REQUIREMENTS	26
3.3.1 PROGRAMMING LANGUAGE	26
3.3.2 LIBRARIES	27
3.3.3 PLATFORM	28
3.4 HARDWARE REQUIREMENTS	29

3.5	COST BENEFIT ANALYSIS	29
4.	SYSTEM DESIGN AND ARCHITECTURE	30
4.1	SYSTEM OVERVIEW	30
4.2	FAKE NEWS DETECTION	
4.2.1	DATA COLLECTION	30
4.2.2	PRE-PROCESSING	31
4.2.3	FEATURE EXTRACTION	32
4.2.4	MACHINE LEARNING ALGORITHMS	33
4.2.5	OUTPUT PREDICTION	35
4.3	REAL NEWS GENERATION	35
4.3.1	DATA EXTRACTION	35
4.3.2.	WEB SCRAPING	36
4.3.3.	OUTPUT	37
4.4	USE-CASE DIAGRAM	38
4.5	EFFICIENCY OF CODE	38
5.	TESTING	39
6.	RESULTS AND ANALYSIS	42
7.	CONCLUSION	46
8.	FUTURE ENHANCEMENT	47
9.	REFERENCES	48
	APPENDIX	51
	PAPER PUBLICATION	63
	PLAGIARISM REPORT	64

LIST OF FIGURES

1. Flow chart diagram	15
2. Naïve bayes algorithm	17
3. Dataset used	24
4. Architecture Diagram	30
5. Naïve Bayes algorithm	33
6. Use-case Diagram	36
7. Fake news detected on website	41
8. Real news detected on website	42
9. Confusion matrix of Passive aggressive Classifier	42
10. Confusion matrix of Naïve Bayes Algorithm	43
11. Classification report (Passive-aggressive)	44

LIST OF FLOW-CHARTS

- | | |
|--|----|
| 1. Flowchart of Fake news detection and real news generation | 15 |
|--|----|

LIST OF GRAPHS

1.	Confusion matrix of Passive aggressive classifier	Co 42
2.	Confusion matrix of Naïve Bayes algorithm	Co 43

ABBREVIATIONS

ML -> Machine Learning

NLP -> Natural Language Processing

TFIDF -> Term Frequency–Inverse Document Frequency

NLTK -> Natural language toolkit

HTML -> Hypertext Markup Language

RAKE -> Rapid Automatic Keyword Extraction algorithm

HTTP -> Hypertext Transfer Protocol

CHAPTER 1

INTRODUCTION

In today's society, most of the news consumption by people is through different social media platforms, since it is the most easy and convenient way of sharing news to each other. But with this comes the risk of widespread dissemination of fake news. These fake news not just adversely affect an individual but it also affects the society as a whole. Today our world is fighting against covid19. This pandemic not just destroyed the livelihood of many people but also destroyed many families. Amidst these problems, fake news just acts as a fuel to the fire. These misinformation conceal healthy behavior and encourage erroneous activities which aid in the spread of virus and lead to poor mental and physical health outcomes in people. Thus, it is very important to stop the chain of fake news from the root itself. This can be done only if we have the proof whether the given news is real or fake and also the source of real news. This is where our project will be beneficial.

1.1 OVERVIEW

With the rapid rise of social media and the protean technological advancements in recent years, we have progressed from accessing news from traditional, conventional means such as radio, newspapers and TV news to a more ubiquitous, dynamic sources which can be credited due to the evolution of the internet. Thus, we are living in a period of time where there is an easy access to information that is growing exponentially. However, such conveniences that have been brought on by whole host of social media networks have also added multiple layers of intricacies and complexities which have made it difficult for a news consumer to differentiate between genuine and fake news, and such dissemination of news followed by sharing and forwarding of such news articles without cross-verification have contributed to rise in prevalence of falsification of news that can not only have grave consequences in the events of the real world but also risks the credibility of social media.

While the existence of fake news itself is not new as different civilizations, organizations, nations have been manipulating the news media to sway public opinion in their favor or for propaganda, the prominence of social media have augmented the power that the fake news can have on an individual and in a society.

Taking into consideration the impact that the consumption of fake news can have on the fragility of the ways society's function, we have proposed a system which can not only detect fake news by cross-verifying a news article with various trustworthy news sources but also generate real news for the users to consume.

1.2 MOTIVATION AND PROBLEM STATEMENT

Social media have enhanced the experience of news consumption due to its cost effective, easily accessible and widely distributable characteristic. However, it has made an average internet user easily vulnerable to consuming news that is intentionally or unintentionally distorted which can have drastic consequences and puts an individual and society at risk.

Therefore, detecting fake news especially on social media poses a relatively new and unique problem because of which it provides a wide range of research opportunities to tackle such challenges. One such challenge is the different ways in which a news is falsified. Fake news can vary greatly from satirical, inflated news articles that are misinterpreted as genuine to articles that make use of sensationalist, clickbait headlines to grasp the attention of users. News articles can even be fabricated and manipulated with intention to deceive, harm or influence public opinion that may result in confirmation bias or political polarization. Since fake news also usually emerge out of developing critical real time events, it is difficult to properly check and verify the quality of data itself.

Since fake news is riddled with factual inaccuracies, it can mitigate the influence of real news by competing with it. In this project, we propose a system that makes use of machine learning algorithms and various feature extraction methods to detect fake news by cross verifying from various other trusted news sites while also generating and displaying real news from trusted sources in the form of a website. Through this project, we aim to obtain maximum accuracy in fake news detection and real news generation to obtain a perfect result.

- A model is proposed to check whether a given stance of information or news article is true or false.
- Basically, the title content and domain name are checked.
- The new model can be constructed from algorithms like Passive Aggressive Classifier, Naïve Bayes algorithm and keyword search algorithm.
- Once we know that a piece of information is not real, it will give genuine news from trusted sites so the spread of fake news can be stopped.

1.3 LEARNING TECHNIQUES

We have used various algorithms and techniques to urge the specified results. Machine learning algorithms such as passive aggressive classifier and Naïve Bayes algorithms are used to predict the output and with a good accuracy. Firstly, data pre-processing is done with stemming and stopwords. This process helps in cleaning up the data. After the pre-processing, feature extraction takes place. This is achieved by TFIDF vectorizer. The TFIDF will check how significant a word is in the whole document. Thus, after the machine learning algorithms the news is predicted to be real or fake. If the news is found out to be fake, the data is extracted and we have used Keyword search algorithm which is achieved by using RAKE and thus we will be able to get the output i.e., related news from some of the trusted sites.

1.4 FLOW CHART

- a) Dat
a Pre-processing-In this model, we take input and preprocess the data using stemming and stopwords. Thus, the data is cleaned in this model.
- b) Fea
ture Extraction- Using TFIDF, how significant a word is in the whole document is checked.

- c) Cla
 ssifier Training- After training and testing the data, ML algorithms like passive aggressive classifier and Naïve Bayes algorithm is used to predict the value.
- d) Dat
 a extraction- After the predicted output, Keyword extraction algorithm is used, which is implemented using rake.

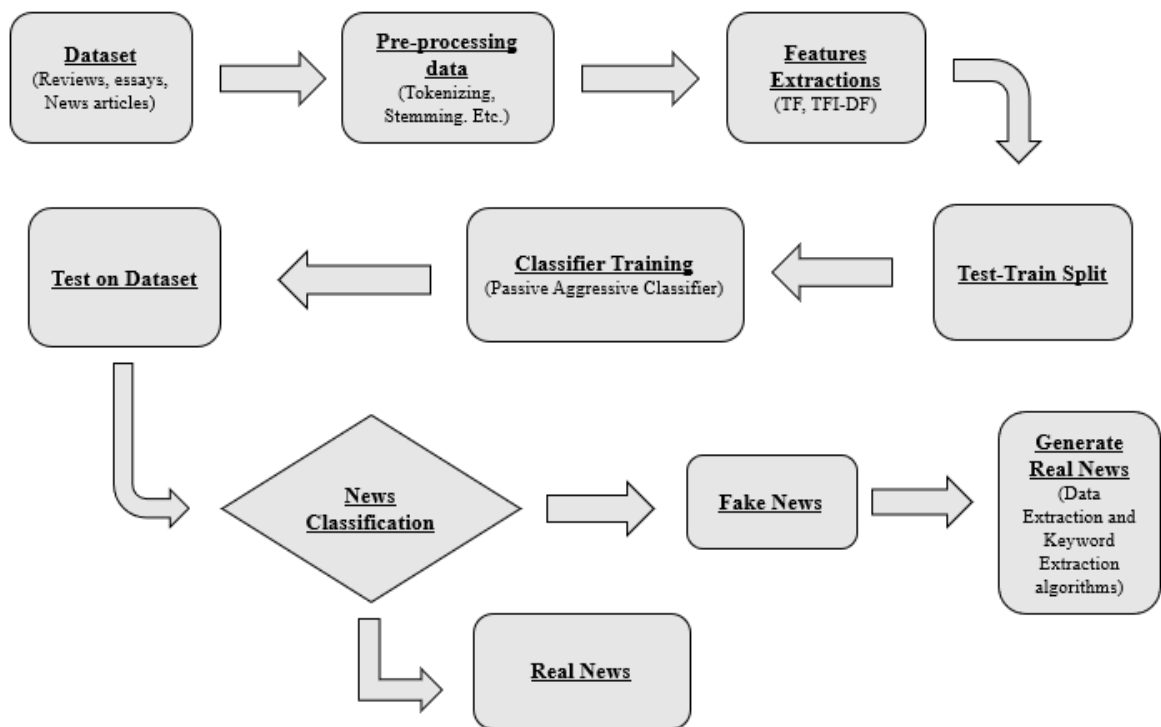


Fig 1. Flow chart

1.5 PYTHON LIBRARIES AND ALGORITHMS

The various ML algorithms that we used are deployed in python programming language which utilize a variety of open-source python libraries.

- a) Lib
- raries and algorithms
- Nu
mpy (numerical python, core library for SC, integrates C/C++)- NumPy is the elementary package of the Python language for scientific computing.
 - Pan
das (opensource library, data manipulation)- Pandas library is made upon Numpy, that means Pandas wants Numpy to control. Pandas offers a technique to produce, manipulate and altercate the data.
 - Sklearn
(opensource ML library, supervised and unsupervised learning algo)-. It offers numerous algorithms like the ones which can support random forests, vector machine, and k-neighbours, and with all this it also helps to support the Python numerical and scientific libraries like SciPy and NumPy.
 - Sci
kit (free ML library, features SVM, classification and regression)- This library is absorbed on modeling of data. Scikit does not emphasis on loading the data, manipulation and summarization of data.
 - **Passive Aggressive Classifier:** - Passive Aggressive Classifiers are an online learning algorithm family, that functions in the same way as a perceptron since they do not need a learning rate. Such a classifier remains passive when the classification outcome is correct, however it turns aggressive as soon as it comes across an incorrect outcome in the event of a miscalculation, after which it updates and modifies the unwanted outcome. In this project, such a classifier can help detect fake news and then fetch and generate relevant, genuine news to the user in the process from trusted news sources, thus fulfilling its purpose of making the

much-needed modifications that corrects the loss. Due to its simplicity in terms of implementation as well as its quality to be used for incremental large-scale learning, it plays an imperative role in classifier training stage after a dataset has been through a test-train split procedure in order to estimate and enhance the performance of the machine learning model used in this project.

- Naïve Bayes Algorithm:** - Naïve Bayes Algorithm is a family of classification algorithms which works on the principle of Bayes Theorem. Therefore, it is also known as a collection of probabilistic classifiers and can be implemented in various classification tasks. In such an algorithm, all pairs of features which are classified are independent of each other. Some of its applications include filtering spam, sentiment prediction and classification of documents. Naïve Bayes holds great significance in this project when it comes to classifying a news article as real news or fake news since it is highly scalable, efficient and can be used to produce real-time predictions while handling continuous as well as discrete data.

The diagram illustrates the Naïve Bayes algorithm. It features the formula for Posterior Probability: $P(c | x) = \frac{P(x | c)P(c)}{P(x)}$. Arrows point from the terms in the formula to their respective labels: $P(c | x)$ is labeled 'Posterior Probability', $P(x | c)$ is labeled 'Likelihood', $P(c)$ is labeled 'Class Prior Probability', and $P(x)$ is labeled 'Predictor Prior Probability'. Below the main formula, the joint probability formula is shown: $P(c | X) = P(x_1 | c) \times P(x_2 | c) \times \dots \times P(x_n | c) \times P(c)$.

Fig 2. Naïve bayes algorithm

- Keyword Search Algorithm:** - Keyword Search Algorithms is a text analysis technique which can be used to determine key phrases in a text in order to simplify information extraction. In this python project, feature extraction methods such as TF-IDF (Term Frequency – Inverse Document Frequency) method has been implemented which makes use of numerical statistic in order to assign a weighting factor based on the frequency of a word in a collection of documents which can determine its importance for information retrieval.

While during the real news generation stage of the model, RAKE (Rapid Automatic Keyword Extraction) which is a keyword extraction algorithm has been implemented to analyze frequency of a word as well as its co-occurrence with other words in a text, based on which genuine relevant news information can be presented to the user.

1.6 ADDRESSING ETHICAL AND SOCIAL ISSUES AND RESPONSIBILITIES

Our ethical and social responsibilities include: -

- Verification of the genuineness of the trusted-sources from where the real news will be generated.
- The generalized models proposed are not going to be in favour of any political, social or economic organization.
- We respect the copyrights, acknowledge the contributions to our research.
- We are always open to fresh thoughts and critique.
- As social media users, encouraging everyone to play their part of personal responsibility of double-checking the information they consume instead of demanding social media companies/journalists to play the role.

CHAPTER 2

LITERATURE SURVEY

In this paper, Shuo yang et al [1] inspect the matter of Unsupervised discovery of fake news on social media by utilizing the users' reckless social media engagement details. They used current event truths and users' integrity as dormant random factors, and they used users' social media engagements to recognise their views on the validity of current events. They suggest a method for unsupervised learning. This system employs a probabilistic graphical paradigm to model current case truths and, as a result, the users' reputation. To solve the inference dilemma, an effective Gibbs sampling technique is proposed. Their experiment results show that their proposed algorithm outperforms the unsupervised standards.

Kai Shu et al [2] examines two facets of the issue of false news identification: -

a) Characterization- This aspect introduces the fundamental concepts of fake news in both traditional and social media.

b) Detection- The current detection methods, including feature extraction and model construction, are examined from a data mining perspective.

They described fake news and characterized it by evaluating various theories and properties in both traditional and social media. They continue to systematically describe the issue of detecting fake news and summarize the strategies of doing so. They discussed about the datasets and measurement criteria that are currently used in existing methods.

Yuta Yanagi et al [3] proposes a fake news detector that can create fake social contexts (comments), with the aim of detecting fake news early on in its spread when few social contexts are available. It's been trained on a series of news articles and their social situations. They also trained a classify model using news posts, real-posted comments, and generated comments. They compared the quality of produced comments for articles with actual comments and those generated by the classifying model to determine the detector's effectiveness.

Limitation- According to their study, the words "!", "?", "false," "breaking," and other similar phrases are essential signals of fake news.

J Zhang et al [4] in this paper, the false news identification problem has been formulated as a legitimacy inference problem, in which genuine news has a higher reputation than fake news, which has a lower credibility.

A deep diffusive network model is proposed based on the interrelationship between various news stories, publishers, and their topics. They also implement a new diffusive unit model called GDU, which acquires multiple inputs from various sources simultaneously and then functionally combines the inputs to generate the necessary output using material "forget" and "change" gates.

Substantial deployment of this model on a real-world fake news repository, such as PolitiFact, has yielded remarkable results when it comes to identifying fake news stories,

publishers, and material in the network, demonstrating the proposed model's impressive efficiency and ability.

A Thota et al [5] proposes Dissemination and consumption of fake news has become a matter of major concern due to its potential to destabilize governments, which poses a grave threat to society and its individuals.

An alternative way in which fake news can be detected is through stance detection which functions by automatically detecting the interrelation between different news articles and its contents. This study thus surveys different ways to predict this relationship, with the help of the news article and headline pair provided. Based on the similarity between the news article and the headlines, the stances can be categorized as ‘unrelated’, ‘discuss’, ‘agree’ or ‘disagree’. Such an approach has been implemented with various traditional machine learning models to set a standard in order to identify contrasts with respect to the modern, sophisticated Deep neural networks are used to define the relationship between the news story and the headline.

Kai Shu et al [6], In this study, proposes a FakeNewsNet which is a comprehensive repository that contains data from a variety of features which were otherwise scarce, such as spatiotemporal information, social context and news material. The repository implements an approach to fetch relevant information from eclectic sources. Furthermore, a preliminary exploration analysis has also been conducted on FakeNewsNet through a variety of features to demonstrate its efficiency and utility in fake news detection tasks.

Limitations: -Apart from being time consuming, another limitation of this study is the presence of significant amount of noise in the selection strategy that is implemented for web search results during the process of fetching data.

Adrian Groza [7], in this paper, proposes that although the consumption and sharing of false, unverified news and pieces of information pertaining to the health and medical domain has been an old practice, there are still a plethora of challenges that still exists and is needed to be tackled in order to save people from falling prey to medical myths. The study aims to identify fake news related to the Covid-19 by integrating natural language processing with ontology reasoning. They look into the way in which reasoning in Description Logics (DLs) can

identify inconsistencies between information from trusted medical sources and information that is not verified and is presented in natural language.

Limitation: -System assessments and verbalizing explanations for each conflicting information are some limitations of this paper.

Subhadra Gurav et al [8] proposes that the current techniques and systems are ineffective in providing an accurate statistical rating for any news. Moreover, the limitations in terms of news categorization and feedback makes the systems less diverse.

In this study, an innovative system for detecting false news using machine learning algorithms is proposed. Based on Twitter feedback and the application of classification algorithms to identify such news events, this model takes news events as input and calculates the percentage of news that is real or false.

Limitations: - Some of the major limitations of this paper is the accuracy of the model as well as the limited information that the model can fetch from different sources.

Kai-Chou Yang et al [9] in this paper treats the problem as a natural language inference (NLI) task where the sentences can be classified as “premise” (P) and “hypothesis” (H). For such a task, NLI models tend to be more reliable and accurate. The collective utility of gradient boosting and fine-tuning with noisy labels demonstrated its significance in the model.

Limitations: - The performance of the model was not satisfactory and the research was adversely affected by time constraints.

Limeng Cui et al [10], In this paper, proposes a robust COVID-19 misinformation dataset known as CoAID, which includes news articles, posts on social media platforms as well as the user interaction that pertains to such misinformation. In addition to the description of the datasets fetched for this study, data analysis has also been conducted to illustrate the distinctive characteristics between fake and factual information, as well as to demonstrate the potential future research opportunities that can be addressed through such methods with the implementation of modern techniques.

Limitations: - A major limitation of this paper is the difficulty regarding authenticity of news or a piece of information as the study addresses a fairly recent and ongoing issue which adds to the complexity of the problem, as well as the process of fetching datasets since it is dynamic and frequently changing.

CHAPTER 3

MODULE DESCRIPTION

In this paper, a system is proposed that not only detects the news as real or fake using machine learning algorithms but it also fetches and presents real news to the user using keyword search algorithms.

3.1 FUNCTIONAL REQUIREMENTS

Functional Requirements consist of:

3.1.1 Web Application

In this project, a web application is created using languages such as HTML, CSS and JavaScript along with Heroku which is a cloud application platform to create an interface that enables a user to enter a news article that is needed to be validated as genuine news or fake news. When the user enters the news article in the text box, with the help of Flask framework which integrates the web application with the machine learning model that is the machinery behind the projects functionality, the news article is utilized as input for the model of machine learning to identify and classify news as genuine or false. In case of detection of real news, the application simply prompts the news as genuine, else the application detects fake news, prompts the user about the detection and then generates real news that may be relevant to the news article.

To access this web application, web browsers such as Chrome, Mozilla etc can be used by the user.

3.2 NON-FUNCTIONAL REQUIREMENTS

Non-functional requirements include:

3.2.1 DATASET

title	text	label					
You Can St	Daniel	FAKE					
Watch The	Google	FAKE					
Kerry to g	U.S.	REAL					
Bernie sup	â€”	FAKE					
The Battle	It's	REAL					
Tehran, U		FAKE					
Girl Horrif	Share	FAKE					
â€” Britain	A Czech st	REAL					
Fact check	Hillary	REAL					
Iran repor	Iranian	REAL					
With all th	CEDAR	REAL					
Donald Tr	Donald	REAL					
Strong Sol	Click	FAKE					
10 Ways A	October	FAKE					
Trump tak	Killing Ob	REAL					
How wom	As more	REAL					
Shocking!	Shocking	FAKE					
Hillary Cli	0	FAKE					
What's in	Washingt	REAL					
The 1 char	While	REAL					
The slippe	With	REAL					
Episode #	Novemb	FAKE					
news_datasets							

Fig. 3 Dataset used

In this study, we have utilized the dataset for both fake news and real news with over 8000 records. The datasets consist of features such as title of the news, text or news content, and label. The data once fetched from the datasets are then pre-processed with the help of processes such as stemming and stopwords which filters or cleans the unnecessary words and only keeps pieces of text or information that can be used as key words to simplify the search process. Then with feature extraction methods such as TFIDF vectorizer, the frequency of words or texts are identified in the collection of documents based on which the relevant topic of the data as well as its authenticity can be checked.

The datasets are then exposed to test-train split, which divides the datasets into training and testing subsets to assess the efficiency of the machine learning models used in this analysis which are Naïve Bayes Classifier and Passive-Aggressive Classifier. While the train subset is implemented to fit the model, the test subset is utilized to make predictions and comparisons between the model's outcome which is generated once an input element of the dataset is provided and the expected result. Such a procedure is fitting especially for large datasets which is expected in such a study.

Upon implementing the two machine learning models on the dataset, the accuracy for both Passive aggressive classifier and Naïve Bayes algorithm is obtained.

Table 3.1: Accuracy comparison

Algorithm	Accuracy
Naïve Bayes	83%
Passive Aggressive Classifier	93.55%

3.3 SOFTWARE REQUIREMENTS

Software requirements consist of:

3.3.1 PROGRAMMING LANGUAGE

In this project, Python version 3.5 has been implemented.

Python programming language is an open-source programming language and since it is free, its use is extensive and has an active community development and support.

Python programming language offers creation of solutions to machine learning problems with code that is readable and intuitive, its simplicity also enables developers to develop robust, reliable projects.

Python is also platform independent which enables the developers to deploy and utilize the code or frameworks on different systems with little to no changes. Python is also supported by a variety of platforms, some of which includes Windows, macOS and Linux.

One of the major reasons for implementing Python programming language is its extensive collection of libraries and frameworks. In this project, Pandas, NumPy, Seaborn are a handful of examples of libraries that have enabled developers to create the system quickly and effectively.

3.3.2 LIBRARIES

The libraries that have been implemented in this project are as follows:

- Sci kit-learn: It is a Python library and it plays an imperative role for implementing machine learning concepts using Python programming language. It contains functions and tools for machine learning as well as for statistical modelling which includes clustering, regression, classification and dimensionality reduction.
- NumPy: It is a Python library used to enable computational power to a python program. It contains N-dimensional arrays, matrix data structures and functions to work with arrays. It is a vital component to integrate variety of datasets into the project.
- Pandas: It is a package that provides developers with efficient, high-speed data analysis tools used to work with structured data which can be n-dimensional or tabular.
- Matplotlib: It is a Python library that consists of a set of functions that can be implemented to visualize and plot data.
- Seaborn: is a visualization library that is based on Matplotlib which is used to implement an interface to create interactive visualization and graphics.
- NLTK: Natural Language Toolkit is a python suite that contains functions and text processing packages such as stemming and tokenization in order to enable a python program to utilize natural language data. In this project, tokenizers such as RegexpTokenizer and

WordpunctTokenizer are implemented to extract tokens or key pieces of text by using regular expressions and by separating punctuation from string of words or sentences. Porter's stemmer algorithm has been implemented for the process of stemming used to reduce words into its root form to filter any unnecessary piece of text. This algorithm implements data mining and information retrieval techniques. When a news article entered by the user is detected and classified as fake, then RAKE which stands for Rapid Automatic Keyword Extraction is implemented which is a keyword search algorithm that determines key words or terms that occurs concurrently in different collection of documents based on which, relevant genuine news can be fetched and displayed to the user.

- Beautiful Soup: This is a Python library that is used to extract data from HTML and XML formats. In this project, such a library can help extract relevant news content from websites of trusted news sources once a news article input inserted by the user is classified as fake by the machine learning model.
- Pickle: It is a python module that is used to convert an object structure into bytestream so that it can be stored in a file and then it can be converted back into bytestream. In other words, it can serialize and deserialize an object structure.

3.3.3 PLATFORM (IDE)

In this project, PyCharm IDE which is developed by JetBrains has been implemented. PyCharm also supports various libraries and consists of an interactive console. PyCharm even supports Anaconda IDE.

Google Colab which is a free IDE has also been implemented in this project. makes it easy for developers to share code through their google drive account, it also comes preinstalled with plenty of frequently used modules and has an user-friendly interface.

3.4 Hardware Requirement

One of the major advantages of this project is its minimal hardware requirement for the user as any electronic device such as mobile phone or laptop from where web browsers access is possible can be used.

3.5 Cost-Benefit Analysis

The cost-benefit analysis of the project could be:

- Basic cost of setting up a device such as laptop or mobile device.
- Almost no user cost
- Cost for service provider includes the cost to fulfill software requirements as well as the cost of training large amount of data for real news generation from genuine sites.
- Users can insert news article and analyze the authenticity of news which will help mitigate the effects of fake news.

CHAPTER 4

SYSTEM DESIGN AND ARCHITECTURE

4.1 SYSTEM OVERVIEW

We have moved from receiving news from old, traditional means such as radio, newspapers, and TV news to a more widespread, dynamic outlets which can be attributed to the growth of the internet, thanks to the rapid rise of social media and the protean technological advances in recent years. Thus, we are living in a time when knowledge is readily available and increasing exponentially. However, such conveniences that have been brought on by whole host of social media networks have also added multiple layers of intricacies and complexities which have made it more complicated for a news consumer to differentiate between genuine and fake news, and such dissemination of news followed by sharing and forwarding of such news articles without cross-verification have contributed to rise in prevalence of falsification of news that can not only have grave consequences in the events of the real world but also risks the credibility of social media.

We suggest a model in this project that makes use of machine learning algorithms and various feature extraction methods to identify fake news by cross-referencing it with other reliable news sources, as well as producing and displaying real news from reliable sources in the form of a website. To achieve a perfect result, we strive to achieve maximum accuracy in fake news detection and real news generation in this project.

These are the steps followed:

- A model is proposed to check whether a given stance of information or news article is true or false.
- Basically, the title content and domain name are checked.
- The new model can be constructed from algorithms like Passive Aggressive Classifier, Naïve Bayes algorithm and keyword search algorithm.
- Once we know that a piece of information is not real, it will give genuine news from trusted sites so the dissemination of false information can be stopped.

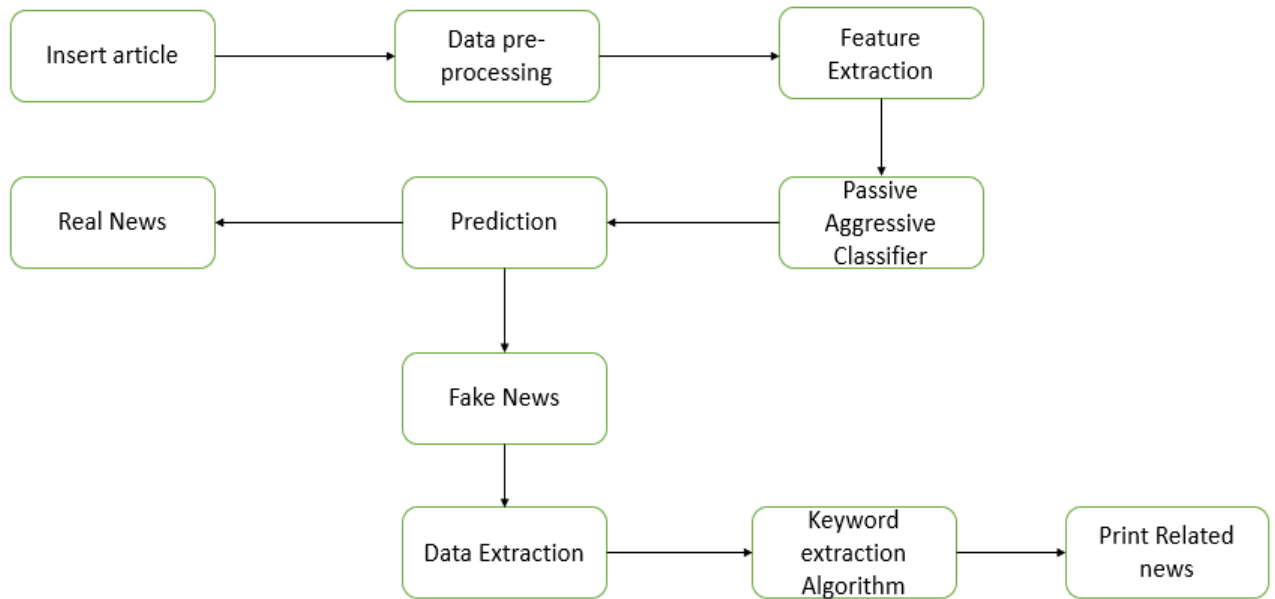


Fig 4. Architecture Diagram

4.2 FAKE NEWS DETECTION

4.2.1. DATA COLLECTION

In the proposed system, the data is collected keeping in mind the current covid situation. So, we have collected the dataset which were publicly available on Kaggle. We went through various datasets and at last came up with dataset with maximum number of records.

4.2.2. PRE-PROCESSING

In the pre-processing step, the data is cleaned such that the unwanted and unnecessary information can be removed and only the relevant details will be kept. In this project we have used Stemming and stopwords. There are different methods used in pre-processing. Some of the methods are mentioned below-

- **Stemming:** - The method of minimizing various words to their root or basic word is known as stemming. For example: If we have words like 'retrieval', 'retrieves', 'retrieved' etc., these words will be reduced to its root form which is retrieve.

Stemming is an important part of Natural language processing and is widely used. In a domain analysis, the stemming is used to evaluate the main vocabularies.

- **Stopwords:** - Stopwords are the common words present in a text such as 'a', 'an', 'the' etc. In the pre-processing, these are the steps which will be filtered out and are not necessary. These are the words which add very little meaning to a sentence in any language. They can be easily overlooked without jeopardizing the sentence's purpose. When we remove the stopwords, the dataset size also decreases which helps in faster processing of data and it also enhances the performance.
- **Tokenization:** - Tokenization refers to splitting of text or words into small tokens. For example, in a paragraph, a line is a token. Similarly, in a line a word is a token. Tokenization is important because, by studying the words in a document, the meaning of the text can be easily deduced. There are different types of tokenization present such as word tokenization, line tokenization, regular expression tokenization etc.

4.2.3. FEATURE EXTRACTION

In Feature extraction, after identifying the key feature from the document, the data is reduced so that it can be cleaned and further be tested on various machine learning algorithms. There are various feature extraction methods. In this project, we have used the TFIDF vectorizer.

TFIDF vectorizer-

TFIDF vectorizer is an abbreviation for Term Frequency and Inverse Document Frequency. It checks that how significant a word is in the whole document.

The term frequency function determines how often a term appears in the text.

The inverse document frequency determines whether a word is uncommon or common across a document.

The TFIDF will thus check the authenticity. So, if a word occurs frequently in many documents like 'what', 'if' etc., they have the chances that they are fake, while the words that appear often in one text but not in all others have a good chance of being true.

4.2.4 MACHINE LEARNING ALGORITHMS

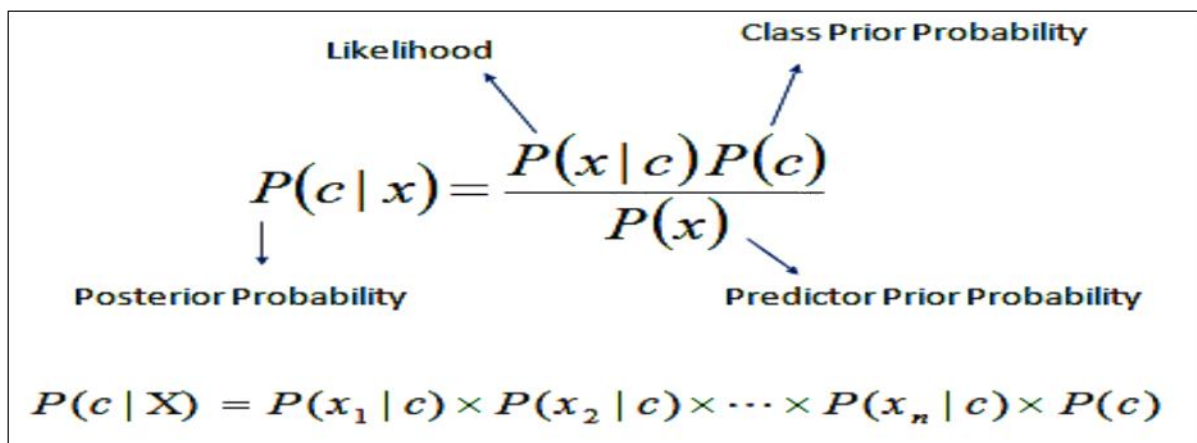
- **Passive Aggressive Classifier:**

Passive Aggressive Classifiers are an online learning algorithm family that functions in the same way as a perceptron since they do not need a learning rate. Such a classifier remains passive when the classification outcome is correct, however it turns aggressive as soon as it comes across an incorrect outcome in the event of a miscalculation, after which it updates and modifies the unwanted outcome. In this project, such a classifier can help detect fake news and then fetch and generate relevant, genuine news to the user in the process from trusted news sources, thus fulfilling its purpose of making the much-needed modifications that corrects the loss. Due to its simplicity in terms of implementation as well as its quality to be used for incremental large-scale learning, it plays an imperative role in classifier

training stage after a dataset has been through a test-train split procedure to approximate and improve the efficiency of the machine learning model used in this project.

- **Naïve Bayes Algorithm:**

Naïve Bayes Algorithm is a family of classification algorithms which works on the principle of Bayes Theorem. Therefore, it is also known as a collection of probabilistic classifiers and can be implemented in various classification tasks. In such an algorithm, all pairs of features which are classified are independent of each other. Some of its applications include filtering spam, sentiment prediction and classification of documents. Naïve Bayes holds great significance in this project when it comes to classifying a news article as real news or fake news since it is highly scalable, efficient and can be used to produce real-time predictions while handling continuous as well as discrete data.



The diagram shows the Naïve Bayes formula with labels for its components:

$$P(c | x) = \frac{P(x | c) P(c)}{P(x)}$$

- Likelihood** points to $P(x | c)$
- Class Prior Probability** points to $P(c)$
- Posterior Probability** points to $P(c | x)$
- Predictor Prior Probability** points to $P(x)$

$$P(c | X) = P(x_1 | c) \times P(x_2 | c) \times \dots \times P(x_n | c) \times P(c)$$

Fig 5. Naïve Bayes algorithm

4.2.5 OUTPUT PREDICTION

After the machine learning algorithms are done, the output will be predicted i.e., whether the news is real or fake. If the news is real the user will know the result and can know about the fact and be

aware, but if the news turned out to be fake, the user will need to check the facts and stop the spread of fake news. For this, our next module real news generation comes into play.

4.3 REAL NEWS GENERATION

4.3.1 DATA EXTRACTION

- **Keyword extraction algorithm**

Keyword extraction algorithm is a method which extracts the most frequently used and relevant words and phrases from a text automatically. It aids in the summarization of text content and identification of the key topics addressed. There are a variety of techniques available for keyword extraction. From basic statistical approach which counts the frequency of words to more sophisticated approach which learns from previous examples to generate even more advanced models. In this project, we have done the implementation using RAKE.

RAKE-

It is Rapid Automatic Keyword Extraction. It is a highly efficient algorithm which works on different files to facilitate application to diverse array. It can easily and effectively handle various documents especially those with unique grammar norms. Its input parameters include a list stopwords, phrase and word delimiters. It partitions the document into candidate keywords, these keywords are primarily the words that aids a developer in extracting the exact keyword which will be helpful in extracting the data from the document.

4.3.2 WEB SCRAPING

Web scraping is the method by which we can extract some data from different websites on internet. The web scraping extracts the underlying Html code. This scraped data can then be replicated to somewhere else. We have used Beautiful soup method in this project.

Beautiful soup-

Beautiful Soup is a Python library for extracting information from markup languages like XML, HTML, and others. Assume you've discovered some websites that show data important to your study, such as dates or addresses, but don't allow you to download the data directly. Beautiful Soup lets you extract basic content from a website, remove the HTML markup, and save the files. It is a method that assists us in cleaning the data that we obtained from the internet.

4.3.3 OUTPUT

Once the Keyword extraction algorithm is applied and then the web scraping is done, we will get the related news from the web. With this we can check the news from some of the trusted sites. This will help the user to clear the facts and stop the spread of fake news.

4.4 USE-CASE DIAGRAM

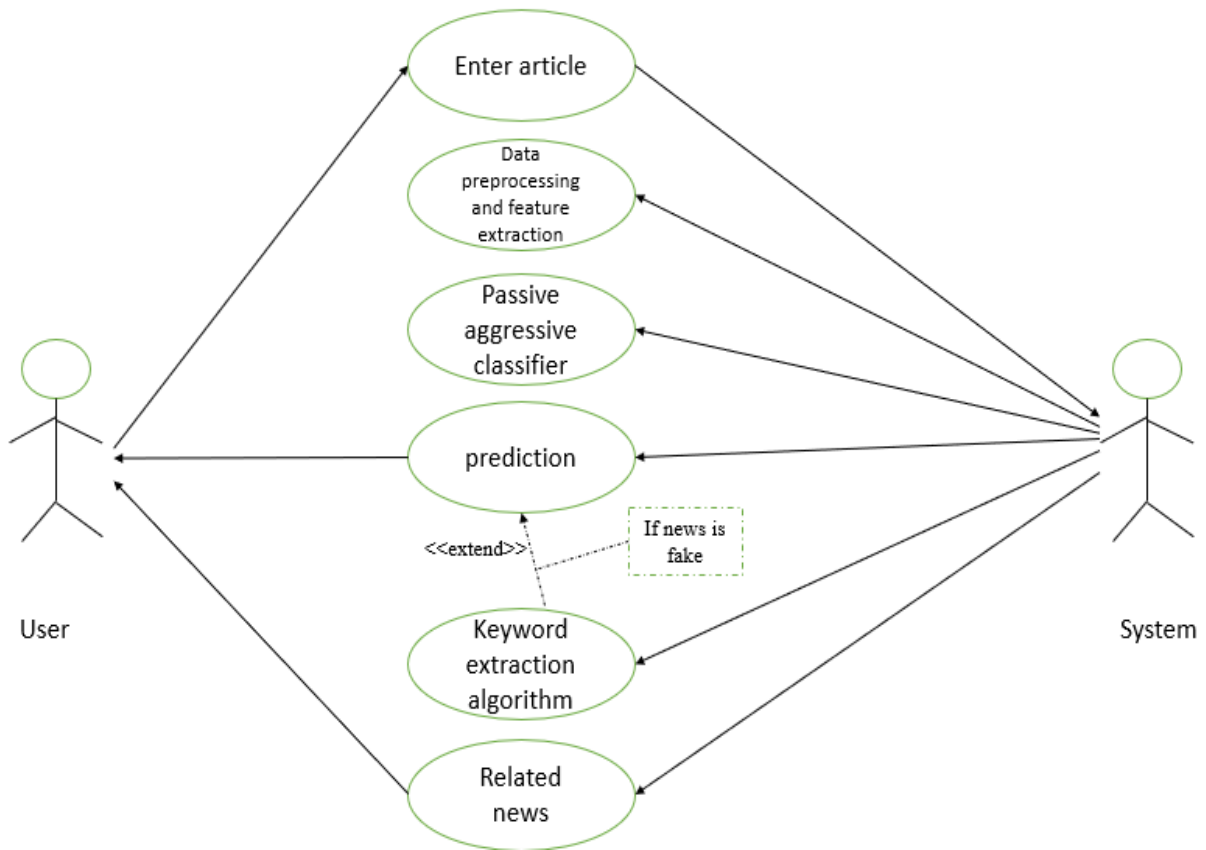


Fig 6. Use case diagram

4.5 EFFICIENCY OF CODE

- Memory Utilization
 - Private heap in Python involves memory management which contains all objects of python and data structures.
 - When you make an object, the Python Virtual Machine manages the memory requirements and determines where the object will be located in the memory structure.

- To approximate huge collections of results, use generators.
- We also used libraries like Numpy for big number/data crunching, which gracefully manages memory management.
- Speed-Up Performance
 - Using the latest version of Python
 - Use built-in functions wherever possible
 - PyPy is yet another Python implementation that includes a JIT (just-in-time) compiler, which speeds up code execution.
- Analyzing the code
 - Code analysis for coverage, accuracy, and efficiency.
 - Python includes the cProfile module, which can be used to test efficiency. It not only totals the operating time, but it also decreases the time of each function separately.



It

also shows you how many times each feature was named, making it easier to decide where to optimize.

CHAPTER 5

TESTING

1) TFIDF vectorizer and Passive Aggressive classifier

```
[ ] tfvect = TfidfVectorizer(stop_words='english',max_df=0.7)
tfidf_x_train = tfvect.fit_transform(x_train)
tfidf_x_test = tfvect.transform(x_test)
```

- max_df = 0.50 means "ignore terms that appear in more than 50% of the documents".
- max_df = 25 means "ignore terms that appear in more than 25 documents".

```
[ ] classifier = PassiveAggressiveClassifier(max_iter=50)
classifier.fit(tfidf_x_train,y_train)
```

```
PassiveAggressiveClassifier(C=1.0, average=False, class_weight=None,
early_stopping=False, fit_intercept=True,
loss='hinge', max_iter=50, n_iter_no_change=5,
n_jobs=None, random_state=None, shuffle=True,
tol=0.001, validation_fraction=0.1, verbose=0,
warm_start=False)
```

```
[ ] y_pred = classifier.predict(tfidf_x_test)
score = accuracy_score(y_test,y_pred)
print(f'Accuracy: {round(score*100,2)}%')
```

```
Accuracy: 93.21%
```

```
[ ] cmatrix = confusion_matrix(y_test,y_pred, labels=['FAKE','REAL'])
print(cmatrix)
```

```
[[570 45]
 [ 41 611]]
```

```
[ ] import matplotlib.pyplot as plt
import sklearn.metrics as metrics
import numpy as np
import itertools

def plot_confusionMatrix(cm, classes,
```

2) Output

```
[ ] def fake_real(news):
    input_text = [news]
    vectorized_input_text = tfvect.transform(input_text)
    prediction = classifier.predict(vectorized_input_text)
    print(prediction)
```

```
[ ] fake_real('Washington (CNN) Both of the remaining Democratic candidates for president easily top Republican front-runner Donald Trump in hypothetical general election match-ups, according to a new CNN/ORC Poll')

['REAL']
```

```
[ ] fake_real('U.S. Republican presidential candidate Donald Trump delivers a campaign speech about national security in Manchester, New Hampshire, U.S. June 13, 2016.')

['FAKE']
```

```
[ ] import pickle
pickle.dump(classifier,open('model.pkl', 'wb'))
```

```
[ ] model_loaded = pickle.load(open('model.pkl', 'rb'))
```

```
[ ] def detector(news):
    input_data = [news]
    vectorized_input_data = tfvect.transform(input_data)
    prediction = model_loaded.predict(vectorized_input_data)
    print(prediction)
```

```
[ ] detector('President Barack Obama has been campaigning hard for the woman who is supposedly going to extend his legacy four more years. The only problem with stumping for Hillary Clinton, however, is she's not exactly a

['FAKE']
```

```
[ ] detector('U.S. Secretary of State John F. Kerry said Monday that he will stop in Paris later this week, amid criticism that no top American officials attended Sunday's unity march against terrorism.')
```

3)Related news

```

from google.colab import drive
drive.mount('/gdrive')

Mounted at /gdrive

[ ] input_news = """
    Wuhan's coronavirus can be cured by one bowl of
    freshly boiled garlic water
    """

[ ] import requests
from bs4 import BeautifulSoup as soup
!pip install rake_nltk
from rake_nltk import Rake

collecting rake_nltk
  Downloading https://files.pythonhosted.org/packages/8e/c4/b4ff57e541ac5624ad4b20b89c2baf4e98f29fd83139f3a81858bdb3815/rake_nltk-1.0.4.tar.gz
Requirement already satisfied: nltk in /usr/local/lib/python3.7/dist-packages (from rake_nltk) (3.2.5)
Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages (from nltk->rake_nltk) (1.15.0)
Building wheels for collected packages: rake-nltk
  Building wheel for rake-nltk (setup.py) ... done
  Created wheel for rake-nltk: filename=rake_nltk-1.0.4-py2.py3-none-any.whl size=7819 sha256=289f772bf5bb247702564e7fd3dc92dd9d2e401c775b8783fc54a5c0707b157
  Stored in directory: /root/.cache/pip/wheels/ef/92/fc/271b3709e71a96ffe934b27818946b795ac6b9b6ff8682483f
Successfully built rake-nltk
Installing collected packages: rake-nltk
Successfully installed rake-nltk-1.0.4

[ ] r = Rake()
r.extract_keywords_from_text(input_news)
keywords = r.get_ranked_phrases()[:5]
query = " "
query = query.join(keywords)

[ ] print(query)

freshly boiled garlic water one bowl wuhan cured coronavirus

[ ] url = "https://news.google.com/search?q=" + query
res = requests.get(url)

[ ] soup = soup(res.content)
all_news = soup.findAll('div', {'class': 'NiLAwe y6IFtc R7GTQ keNKEd j7Vlaaf nID9nc'})

[ ] related_news = []
for news in all_news:
    temp = {}
    temp['title'] = news.find('h3').find('a').text
    temp['bilinear'] = news.find('span', {'class': 'xBbh9'}).text
    temp['img'] = news.find('a').find('img')['src']
    temp['link'] = "https://news.google.com" + news.find('h3').find('a')['href'][1:]
    temp['source'] = news.find('a', {'class': 'wEwYrc AMW2gc uQIVzc Sksgp'}).text
    related_news.append(temp)

[ ] ## Output related_news
print(related_news)

[{'title': 'No, boiled garlic water cannot cure coronavirus. The claim is false', 'bilinear': 'In case you have come across messages which claim that patient infected with coronavirus can be cured with one bowl of freshly

```


CHAPTER 6

RESULTS AND ANALYSIS

1)

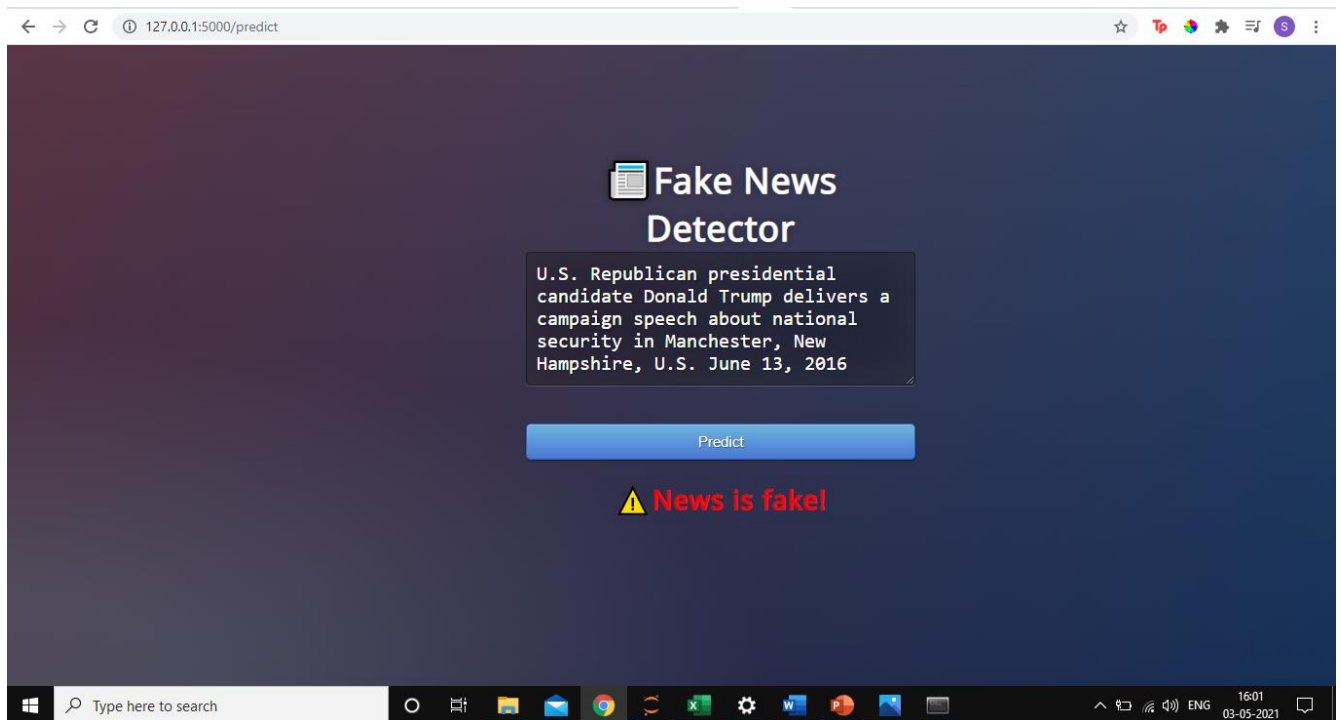


Fig 7. Fake news detected on website

2)

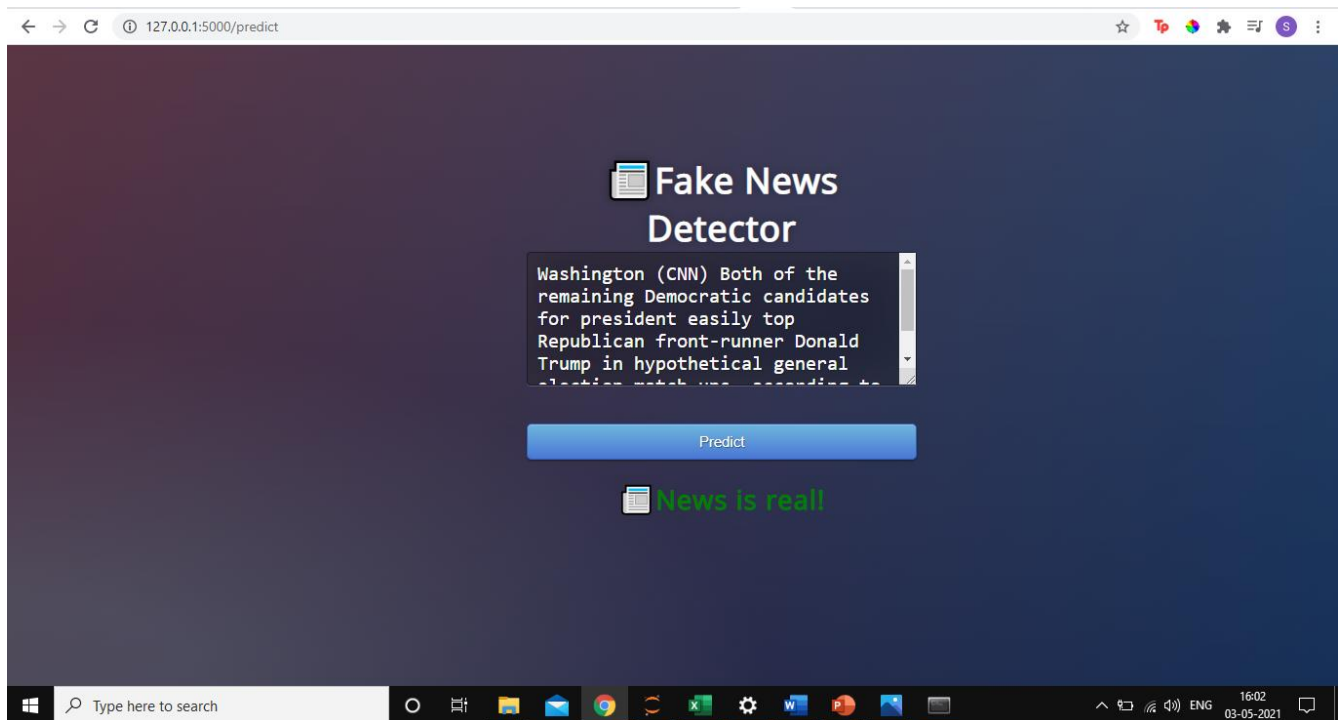


Fig 8. Real news detected on website

3)

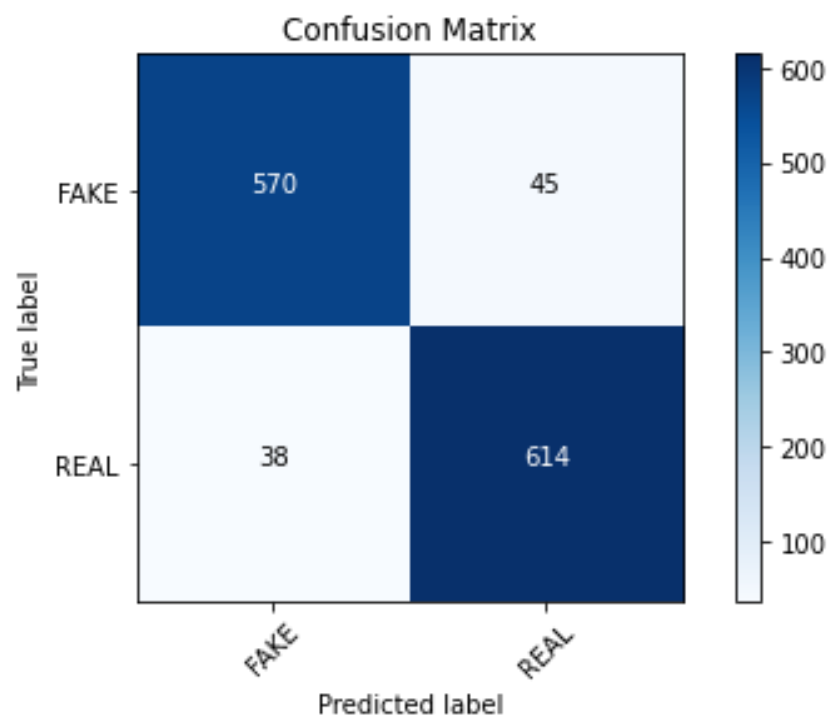


Fig 9. Confusion matrix of Passive aggressive Classifier

From here we can see-

True negative (TN) =572,

False positive (FP)=43,

False negative (FN)=39,

True positive (TP)=613

so, the Accuracy= (TP+TN)/Total.

i.e., Accuracy= 1185/1267=0.9352

And Precision= TP/ Predicted real= 613/656=0.9345

2)

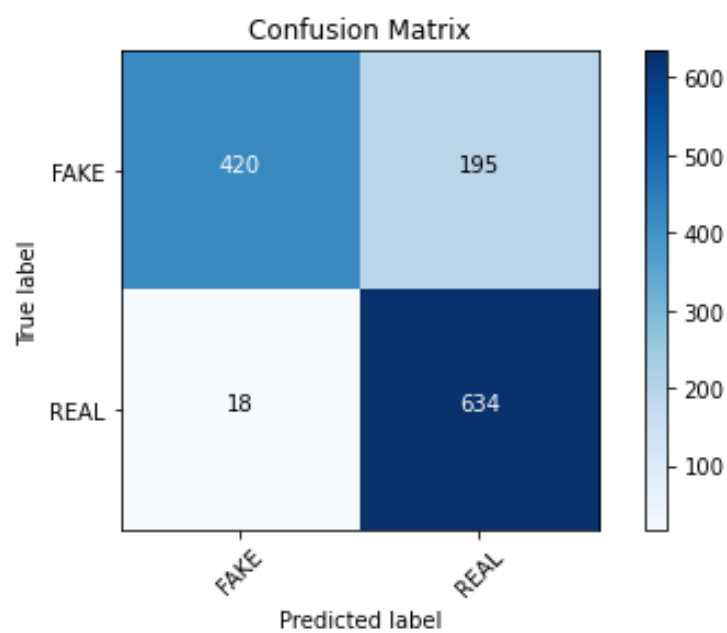


Fig 10. Confusion matrix of Naïve Bayes Algorithm

From here we can see-

True negative (TN) =420,

False positive (FP)=195,

False negative (FN)=18,

True positive (TP)=634

so, the Accuracy= (TP+TN)/Total.

i.e., Accuracy= (634+420)/1267=0.835

And Precision= TP/ Predicted real= 634/829=0.7645

```
[27] print(f"Classification Report : \n\n{classification_report(y_test, y_pred)}")
```

Classification Report :

	precision	recall	f1-score	support
FAKE	0.94	0.93	0.94	615
REAL	0.93	0.95	0.94	652
accuracy			0.94	1267
macro avg	0.94	0.94	0.94	1267
weighted avg	0.94	0.94	0.94	1267

Fig 11. Classification report(Passive-aggressive)

CHAPTER 7

CONCLUSION

With the increased use of social media for news consumption and in prevalence, the widespread distribution of false news has the potential to harm both individuals and society as a whole. Even in the midst of the current covid-19 pandemic, false information on platforms like WhatsApp, Twitter and Facebook can cause panic and have a shocking impact not just on an individual but to a society as a whole. The objective is to detect the fake news through latest technologies and algorithms like- Passive aggressive classifier. We used fake news detection where the user will enter the text and this text will go through our various models and at last give a prediction whether it is true or false. Further, our real news generation will check and validate the news and give us some news from trusted sites.

Our proposed model consists of two components, one where the detection takes place and the other where its correction takes place, if the news is found out to be false corresponding correct news is given as output. We determine the accuracy of these models and discuss about their limitations. In our project, the user can enter the text. Various machine learning algorithms are performed and we found out that Passive aggressive classifier gives a better accuracy as compared to Naïve Bayes. Further, the data is extracted and then real news generation is done using the keyword extraction algorithm. On the basis of our analysis, we can successfully remove the fake news if any.

CHAPTER 8

FUTURE ENHANCEMENT

Because of its low cost, easy accessibility, and broad distribution, social media has improved the news consumption experience. However, it has rendered the average internet consumer susceptible to consuming news that has been skewed deliberately or inadvertently, which can have serious implications and put a person and society at risk. Thus, we focused on eliminating the problem of fake news from the root itself and also providing people with a consequent genuine news which will help them to gain knowledge and be aware of the facts.

Based on our obtained results the following are the future directions for continuing the research:

- a) We could test, optimize and cross validate various machine learning models so that we get good results across different types of news as well as news related to covid-19.
- b) We could compare various other machine learning algorithms.
- c) Testing the proposed method in this paper on a larger dataset to check for accuracy and problems associated.
- d) Can improve the webpage by using animations and wallpapers and making it more attractive.
- e) After the successful implementation and removing all problems, can try for making this in the form of mobile app.

CHAPTER 9

REFERENCES

- [1] Yang, S., Shu, K., Wang, S., Gu, R., Wu, F. and Liu, H., 2019, July. Unsupervised fake news detection on social media: A generative approach. In Proceedings of the AAAI conference on artificial intelligence (Vol. 33, No. 01, pp. 5644-5651).
- [2] Kai Shu , Amy Sliva , Suhang Wang , Jiliang Tang , and Huan Liu, 2017 september. Fake News Detection on Social Media: A Data Mining Perspective
- [3] Yanagi, Y., Orihara, R., Sei, Y., Tahara, Y. and Ohsuga, A., 2020, July. Fake News Detection with Generated Comments for News Articles. In 2020 IEEE 24th International Conference on Intelligent Engineering Systems (INES) (pp. 85-90). IEEE.
- [4] Zhang, J., Dong, B. and Philip, S.Y., 2020, April. Fakedetector: Effective fake news detection with deep diffusive neural network. In 2020 IEEE 36th International Conference on Data Engineering (ICDE) (pp. 1826-1829). IEEE.
- [5] Thota, A., Tilak, P., Ahluwalia, S. and Lohia, N., 2018. Fake news detection: A deep learning approach. SMU Data Science Review, 1(3), p.10.
- [6] Shu, K., Mahudeswaran, D., Wang, S., Lee, D. and Liu, H., 2020. FakeNewsNet: A Data Repository with News Content, Social Context, and Spatiotemporal Information for Studying Fake News on Social Media. Big Data, 8(3), pp.171-188.
- [7] Groza, A., 2020. Detecting fake news for the new coronavirus by reasoning on the Covid-19 ontology. arXiv preprint arXiv:2004.12330.

- [8] Gurav, S., Sase, S., Shinde, S., Wabale, P. and Hirve, S., 2019. Survey on Automated System for Fake News Detection using NLP & Machine Learning Approach. *International Research Journal of Engineering and Technology (IRJET)*, 6(01), pp.308-309.
- [9] Yang, K.C., Niven, T. and Kao, H.Y., 2019. Fake news detection as natural language inference. *arXiv preprint arXiv:1907.07347*.
- [10] Cui, L. and Lee, D., 2020. Coaid: Covid-19 healthcare misinformation dataset. *arXiv preprint arXiv:2006.00885*.
- [11] Qi, P., Cao, J., Yang, T., Guo, J. and Li, J., 2019, November. Exploiting multi-domain visual information for fake news detection. In *2019 IEEE International Conference on Data Mining (ICDM)* (pp. 518-527). IEEE.
- [12] Srivastava, A., Kannan, R., Chelmiss, C. and Prasanna, V.K., 2019, December. RecANt: Network-based Recruitment for Active Fake News Correction. In *2019 IEEE International Conference on Big Data (Big Data)* (pp. 940-949). IEEE
- [13] Long, Y., 2017. Fake news detection through multi-perspective speaker profiles. *Association for Computational Linguistics*.
- [14] Wang, W. Y. 2017. "liar, liar pants on fire": A new benchmark dataset for fake news detection. *arXiv preprint arXiv:1705.00648*
- [15] Jin, Z.; Cao, J.; Zhang, Y.; and Luo, J. 2016. News verification by exploiting conflicting social viewpoints in microblogs. In *AAAI*, 2972–2978.
- [16] N. Grinberg, K. Joseph, L. Friedland, B. Swire-Thompson, and D. Lazer. Fake news on twitter during the 2016 us presidential election. *Science*, 363(6425):374–378, 2019.
- [17] Magdy, A., and Wanas, N. 2010. Web-based statistical fact checking of textual documents. In *Proceedings of the 2nd international workshop on Search and mining user-generated contents*, 103–110. ACM.
- [18] Ajao, O., Bhowmik, D. and Zargari, S., 2018, July. Fake news identification on twitter with hybrid cnn and rnn models. In *Proceedings of the 9th international conference on social*

media and society (pp. 226-230).

- [19] De Sarkar, S., Yang, F. and Mukherjee, A., 2018, August. Attending sentences to detect satirical fake news. In Proceedings of the 27th International Conference on Computational Linguistics (pp. 3371-3380).
- [20] Shu, K., Sliva, A., Wang, S., Tang, J. and Liu, H., 2017. Fake news detection on social media: A data mining perspective. ACM SIGKDD explorations newsletter, 19(1), pp.22-36.
- [21] Zhou, X. and Zafarani, R., 2018. A Survey of Fake News: Fundamental Theories, Detection Methods, and Opportunities. arXiv preprint arXiv:1812.00315.
- [22] Roy, A., Basak, K., Ekbal, A. and Bhattacharyya, P., 2018. A deep ensemble framework for fake news detection and classification. arXiv preprint arXiv:1811.04670.
- [23] CUȘMALIUC, C.G., COCA, L.G. and IFTENE, A., 1843. Identifying Fake News on Twitter using Naive Bayes, SVM and Random Forest Distributed Algorithms. In Proceedings of The 13th Edition of the International Conference on Linguistic Resources and Tools for Processing Romanian Language (ConsILR-2018). ISSN (pp. 177-188).
- [24] Liu, Y. and Wu, Y.F.B., 2018, April. Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks. In Thirty-Second AAAI Conference on Artificial Intelligence.
- [25] Oshikawa, R., Qian, J. and Wang, W.Y., 2018. A survey on natural language processing for fake news detection. arXiv preprint arXiv:1811.00770.
- [26] Yang, Y., Zheng, L., Zhang, J., Cui, Q., Li, Z. and Yu, P.S., 2018. TI-CNN: Convolutional neural networks for fake news detection. arXiv preprint arXiv:1806.00749.

APPENDIX

i.

Fa

ke news detection

```
import pandas as pd
```

```
dataframe = pd.read_csv('news.csv')
```

```
dataframe.head()
```

```
x = dataframe['text']
```

```
y = dataframe['label']
```

```
x
```

```
0    Daniel Greenfield, a Shillman Journalism Fello...
```

```
1    Google Pinterest Digg Linkedin Reddit Stumbleu...
```

```
2    U.S. Secretary of State John F. Kerry said Mon...
```

```
3    — Kaydee King (@KaydeeKing) November 9, 2016 T...
```

```
4    It's primary day in New York and front-runners...
```

```
...
```

```
6330    The State Department told the Republican Natio...
```

6331 The 'P' in PBS Should Stand for 'Plutocratic' ...

6332 Anti-Trump Protesters Are Tools of the Oligar...

6333 ADDIS ABABA, Ethiopia —President Obama convene...

6334 Jeb Bush Is Suddenly Attacking Trump. Here's W...

Name: text, Length: 6335, dtype: object

y

0 FAKE

1 FAKE

2 REAL

3 FAKE

4 REAL

...

6330 REAL

6331 FAKE

6332 FAKE

6333 REAL

6334 REAL

Name: label, Length: 6335, dtype: object

```
from sklearn.model_selection import train_test_split
```

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

```
from sklearn.linear_model import PassiveAggressiveClassifier
```

```
from sklearn.metrics import accuracy_score, confusion_matrix
```

```
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2,random_state=0)
```

```
y_train
```

```
2402  REAL
```

```
1922  REAL
```

```
3475  FAKE
```

```
6197  REAL
```

```
4748  FAKE
```

```
...
```

```
4931  REAL
```

```
3264  REAL
```

```
1653  FAKE
```

```
2607  FAKE
```

```
2732  REAL
```

```
Name: label, Length: 5068, dtype: object
```

```
tfvect = TfidfVectorizer(stop_words='english',max_df=0.7)
```

```
tfidf_x_train = tfvect.fit_transform(x_train)
```

```
tfidf_x_test = tfvect.transform(x_test)
```

- `max_df = 0.50` means "ignore terms that appear in more than 50% of the documents".
- `max_df = 25` means "ignore terms that appear in more than 25 documents".

```
classifier = PassiveAggressiveClassifier(max_iter=50)
```

```
classifier.fit(tfidf_x_train,y_train)
```

```
PassiveAggressiveClassifier(C=1.0, average=False, class_weight=None,  
                             early_stopping=False, fit_intercept=True,  
                             loss='hinge', max_iter=50, n_iter_no_change=5,  
                             n_jobs=None, random_state=None, shuffle=True,  
                             tol=0.001, validation_fraction=0.1, verbose=0,  
                             warm_start=False)
```

```
y_pred = classifier.predict(tfidf_x_test)
```

```
score = accuracy_score(y_test,y_pred)
```

```
print(f'Accuracy: {round(score*100,2)}%')
```

Accuracy: 93.45%

```
cfmatrix = confusion_matrix(y_test,y_pred, labels=['FAKE','REAL'])
```

```
print(cfmatrix)
```

```
[[570  45]
```

```
 [ 38 614]]
```

```
import matplotlib.pyplot as plt
```

```
import sklearn.metrics as metrics
```

```
import numpy as np
```

```
import itertools
```

```
def plot_ConfusionMatrix(cm, classes,  
                           normalize=False,  
                           title='Confusion Matrix',  
                           cmap=plt.cm.Blues):
```

```
plt.imshow(cm, interpolation='nearest', cmap=cmap)
```

```
plt.title(title)
```

```
plt.colorbar()
```

```
tick_marks = np.arange(len(classes))
```

```
plt.xticks(tick_marks, classes, rotation=45)
```

```
plt.yticks(tick_marks, classes)
```

```
if normalize:
```

```
    cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
```

```
    print("Normalized Confusion Matrix")
```

```
else:
```

```
    print('Confusion Matrix, without normalization')
```

```
thresh = cm.max() / 2.
```

```
for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):
```

```
    plt.text(j, i, cm[i, j],
```



```
horizontalalignment="center",  
color="white" if cm[i, j] > thresh else "black")
```

```
plt.tight_layout()
```

```
plt.ylabel('True label')
```

```
plt.xlabel('Predicted label')
```

```
cm = metrics.confusion_matrix(y_test, y_pred)
```

```
plot_ConfusionMatrix(cm, classes=['FAKE', 'REAL'])
```

Confusion Matrix, without normalization

```
def fake_real(news):
```

```
    input_text = [news]
```

```
    vectorized_input_text = tfvect.transform(input_text)
```

```
    prediction = classifier.predict(vectorized_input_text)
```

```
    print(prediction)
```

```
detector("""U.S. Secretary of State John F. Kerry said Monday that he will stop in Paris later this  
week, amid criticism that no top American officials attended Sunday's unity march against  
terrorism.""")
```

```
['REAL']
```

detector('President Barack Obama has been campaigning hard for the woman who is supposedly going to extend his legacy four more years. The only problem with stumping for Hillary Clinton, however, is she™s not exactly a candidate easy to get too enthused about.')

['FAKE']

import pickle

`pickle.dump(classifier,open('model.pkl', 'wb'))`

`model_loaded = pickle.load(open('model.pkl', 'rb'))`

from sklearn.naive_bayes import MultinomialNB

`tfidf_df = pd.DataFrame(tfidf_x_train.A, columns=tfvect.get_feature_names())`

`clf = MultinomialNB()`

`clf.fit(tfidf_x_train, y_train)`

`prediction = clf.predict(tfidf_x_test)`

`score = metrics.accuracy_score(y_test, prediction)`

`print("accuracy: %0.3f" % score)`

```
accuracy: 0.832
```

```
cm = metrics.confusion_matrix(y_test, prediction)
```

```
print(cm)
```

```
[[420 195]
```

```
 [ 18 634]]
```

```
plot_ConfusionMatrix(cm, classes=['FAKE', 'REAL'])
```

Confusion Matrix, without normalization

ii. Real news generation

```
input_news = ""
```

```
    Wuhan's coronavirus can be cured by one bowl of
```

```
    freshly boiled garlic water
```

```
""
```

```
import requests
```

```
from bs4 import BeautifulSoup as soup
```

```
!pip install rake_nltk
```

```
from rake_nltk import Rake
```

```
r = Rake()
```

```
r.extract_keywords_from_text(input_news)
```

```
keywords = r.get_ranked_phrases()[:5]
```

```
query = " "
```

```
query = query.join(keywords)
```

```
print(query)
```

```
url = "https://news.google.com/search?q=" + query
```

```
res = requests.get(url)
```

```
soup = soup(res.content)
```

```
all_news = soup.findAll('div', {'class' : 'NiLAwe y6IFtc R7GTQ keNKEd j7vNaf nID9nc'})
```

```
related_news = []
```

```
for news in all_news:
```

```

temp = {}

temp['title'] = news.find('h3').find('a').text

temp['biliner'] = news.find('span', {'class': 'xBbh9'}).text

temp['img'] = news.find('a').find('img')['src']

temp['link'] = "https://news.google.com" + news.find('h3').find('a')['href'][1:]

temp['source'] = news.find('a', {'class': 'wEwycr AVN2gc uQIVzc Sksgp'}).text

related_news.append(temp)

```

```

### Output related_news

```

```

print(related_news)

```

iii. Web deployment

```

from flask

```

```

import Flask,

```

```

render_template,

```

```

request

```

```

from sklearn.feature_extraction.text import TfidfVectorizer

```

```
from sklearn.linear_model import PassiveAggressiveClassifier
```

```
from sklearn.model_selection import train_test_split
```

```
import pickle
```

```
import pandas as pd
```

```
app = Flask(__name__)
```

```
tfvect = TfidfVectorizer(stop_words='english', max_df=0.7)
```

```
model_loaded = pickle.load(open('model.pkl', 'rb'))
```

```
df = pd.read_csv('news.csv')
```

```
x = df['text']
```

```
y = df['label']
```

```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2,  
random_state=0)
```

```
def fake_news_det(news):

    tfidf_x_train = tfvect.fit_transform(x_train)

    tfidf_x_test = tfvect.transform(x_test)

    input_text = [news]

    vectorized_input_text = tfvect.transform(input_text)

    result = model_loaded.predict(vectorized_input_text)

    return result


@app.route('/')

def home():

    return render_template('index.html')
```

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

def predict():

    if request.method == 'POST':

        message = request.form['message']

        pred = fake_news_det(message)

        print(pred)

        return render_template('index.html', prediction=pred)

    else:

        return render_template('index.html', prediction="Something went wrong")

if __name__ == '__main__':
```



```
app.run(debug=True)
```

PAPER PUBLICATION STATUS

Submitted to a conference waiting for approval

PLAGIARISM REPORT

ABSTRACT

ABSTRACT

ORIGINALITY REPORT

0%

SIMILARITY INDEX

0%

INTERNET SOURCES

0%

PUBLICATIONS

0%

STUDENT PAPERS

PRIMARY SOURCES

Exclude quotes

On

Exclude matches

< 10 words

Exclude bibliography

On

CHAPTER 1

INTRODUCTION

CHAPTER 1

ORIGINALITY REPORT

2%	1%	2%	0%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Pakala Prahasit Reddy, Yempati Prasheela, Avula Uday Kumar Reddy, Rajanikanth Aluvalu. "An Approach to detect fault text in articles", IOP Conference Series: Materials Science and Engineering, 2021 Publication	2%
2	kuid-rm-web.ofc.kobe-u.ac.jp Internet Source	1%

Exclude quotes On
Exclude bibliography On

Exclude matches < 10 words

CHAPTER 2

LITERATURE SURVEY

CHAPTER 2

ORIGINALITY REPORT

5%	0%	5%	3%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Yuta Yanagi, Ryohei Orihara, Yuichi Sei, Yasuyuki Tahara, Akihiko Ohsuga. "Fake News Detection with Generated Comments for News Articles", 2020 IEEE 24th International Conference on Intelligent Engineering Systems (INES), 2020 Publication	2%
2	Jiawei Zhang, Bowen Dong, Philip S. Yu. "Deep Diffusive Neural Network based Fake News Detection from Heterogeneous Social Networks", 2019 IEEE International Conference on Big Data (Big Data), 2019 Publication	2%

CHAPTER 3

MODULE DESCRIPTION

CHAPTER 3

ORIGINALITY REPORT

2%	2%	0%	0%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	nais.kcg.jp Internet Source	1%
2	bioone.org Internet Source	1%

CHAPTER 4

SYSTEM DESIGN AND ARCHITECTURE

CHAPTER 4

ORIGINALITY REPORT

3%	0%	0%	3%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to University of Greenwich Student Paper	1%
2	Submitted to Coventry University Student Paper	1%
3	Submitted to The University of Manchester Student Paper	1%

CHAPTER 6

RESULTS AND ANALYSIS

CHAPTER 6

ORIGINALITY REPORT

10%	0%	10%	0%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Jyoti Dabass, M. Hanmandlu, Rekha Vig. "Formulation of probability-based pervasive information set features and Hanman transform classifier for the categorization of mammograms", SN Applied Sciences, 2021 Publication	10%
---	---	-----

CHAPTER 7

CONCLUSION

CHAPTER 7

ORIGINALITY REPORT

0%

SIMILARITY INDEX

0%

INTERNET SOURCES

0%

PUBLICATIONS

0%

STUDENT PAPERS

PRIMARY SOURCES

CHAPTER 8

FUTURE ENHANCEMENT

CHAPTER 8

ORIGINALITY REPORT

0%

SIMILARITY INDEX

0%

INTERNET SOURCES

0%

PUBLICATIONS

0%

STUDENT PAPERS

PRIMARY SOURCES

PLAGIARISM REPORT

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
--

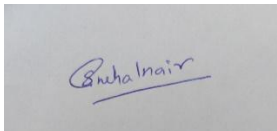
(Deemed to be University u/s 3 of UGC Act, 1956)
--

Office of Controller of Examinations

REPORT FOR PLAGIARISM CHECK ON THE DISSERTATION / PROJECT REPORTS FOR UG/ PG PROGRAMMES

1	Name of the candidate (IN BLOCK LETTERS)	SNEHAL NAIR		
2	Address of the candidate	73-Anurag Nagar, Bombay hospital road, Behind Shalimar township, Indore, Madhya Pradesh		
3	Registration number	RA1711008010245		
4	Date of Birth	18/11/1999		
5	Department	Information Technology		
6	Faculty	Engineering and Technology		
7	Title of the Dissertation / Project	FAKE NEWS DETECTION WITH REAL NEWS GENERATION		
8	Whether the above dissertation is done by	Individual/ Group a) If group, number of students: 3 b) Name and Register Numbers of other candidates: Sruthi VS (RA1711008010235) Merwin Roy (RA1711008010283)		
9	Name and address of the Supervisor/ Guide	Email ID: nithyakp@srmist.edu.in phone: 9790626371		
10	Name and address of the C0-Supervisor/ Co-guide (if any)	None		
11	Software used	Turnitin		
12	Date of Verification			
13	Plagiarism Details: (to attach the final report from the software)			
Chapter	Title of the Chapter	Percentage of similarity index (including self citations)	Percentage of similarity index (excluding self citations)	Percentage of plagiarism excluding Quotes, Bibliography, etc
	Abstract	0%	0%	0%
1.	Introduction	2%	0%	2%
2.	Literature survey	5%	0%	5%
3.	Module description	2%	0%	2%
4.	System design and architecture	3%	0%	3%
5.	Testing	0%	0%	0%
6.	Results and analysis	10%	0%	10%
7.	Conclusion	0%	0%	0%
8.	Future enhancement	0%	0%	0%
9.	References	0%	0%	0%
Appendices				

We declare that the above information has been verified and found true to the best of our knowledge.

A handwritten signature in blue ink, appearing to read "Suhainair", is written on a white rectangular background.

Signature of the Candidate

A handwritten signature in blue ink, appearing to read "M. Arjun", is written in blue ink.

Name and Signature of the Staff who uses the plagiarism software

A handwritten signature in blue ink, appearing to read "S. N. Arjun", is written in blue ink.

Name and Signature of Guide :

Name and Signature of Co - Guide :

A handwritten signature in black ink, appearing to read "Dr. G. VADIVU", is written in black ink.

Dr. G. VADIVU
Head
Department of Information Technology
Faculty of Engineering & Technology
SRM Institute of Science and Technology
SRM Nagar, Kattankulathur - 603 203.
Kancheepuram Dist., Tamil Nadu, India.

Name and signature of the Head of Department