CHARACTERIZATION, CLASSIFICATION AND DETECTION OF FAKE NEWS

A PROJECT REPORT

Submitted By

ZAINAB ZUBAIR
(2100290140157)
SHIVANSH SRIVASTAVA
(2100290140131)
MUSKAN AGRAWAL
(2100290140091)

Submitted in partial fulfilment of the Requirements for the Degree of

MASTER OF COMPUTER APPLICATION

Under the Supervision of
Dr Vidushi
Assistant Professor

Submitted to

DEPARTMENT OF COMPUTER APPLICATIONS
KIET Group of Institutions, Ghaziabad
Uttar Pradesh-201206
(JUNE 2023)

DECLARATION

I hereby declare that the work presented in this report entitled "CHARACTERIZATION , CLASSIFICATION AND DETECTION OF FAKE NEWS", was carried out by me. I have not submitted the matter embodied in this report for the award of any other degree or diploma of any other University or Institute. I have given due credit to the original authors/sources for all the words, ideas, diagrams, graphics, computer programs, experiments, results, that are not my original contribution. I have used quotation marks to identify verbatim sentences and given credit to the original authors/sources.

I affirm that no portion of my work is plagiarized, and the experiments and results reported in the report are not manipulated. In the event of a complaint of plagiarism and the manipulation of the experiments and results, I shall be fully responsible and answerable.

Name : Zainab Zubair (2100290140157)

Shivansh Srivastava(2100290140131)

Muskan Agrawal (2100290140091)

(Candidate Signature)

CERTIFICATE

Certified that Zainab Zubair **2100290140157,** Shivansh Srivastava **2100290140131,** Muskan Agrawal **2100290140091** have carried out the project work having **“Characterization ,Classification and Detection of Fake News Detection”** for Master of Computer Applications from Dr. A.P.J. Abdul Kalam Technical University (AKTU) (formerly UPTU), Technical University, Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student himself / herself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Date: 29/05/2023

Zainab Zubair (2100290140157)
Shivansh Srivastava (2100290140131)
Muskan Agrawal (2100290140091)

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date: 29/05/2023

Dr Vidushi
Assistant Professor
Department of Computer Applications
KIET Group of Institutions,Ghaziabad

Signature of Internal Examiner **Signature of External Examiner**

Dr. Arun Kumar Tripathi
Head, Department of Computer Applications
KIET Group of Institutions, Ghaziabad

ABSTRACT

Now a day’s internet plays a major role and becomes the necessity in everyone’s life for various online resources for obtaining news. The increasing use of social media platforms like facebook, instagram, twitter, etc leads to the rapid spread of the news to millions of people in short period of time. The news can be useful but at the same time it can be harmful too. The spread of fake news has far-reaching consequences like for spoiling the name of someone for some personal agenda, creation of biased opinions at the time of election, etc. Fake news is mainly concerned for creating confusion among people and to divert their mind to something irrelevant. Fake news always catches the attention of users as creation of appealing headlines is used to generate revenues, views and spice among users. Readers should be very careful while going through the news and make sure that the obtained information is true. In this paper, our aim is to check the accuracy of the news available online with the help of Natural Language Processing, Long short-term memory and Naïve Bayes. We aim to provide the user with the ability to classify the news as fake or real and also check the authenticity of the website publishing the news.

ACKNOWLEDGEMENT

Success in life is never attained single handedly. My deepest gratitude goes to my thesis supervisor, **Dr Vidushi** for her guidance, help and encouragement throughout my research work. Their enlightening ideas, comments, and suggestions.

Words are not enough to express my gratitude to **Dr. Arun Kumar Tripathi**, Professor and Head, Department of Computer Applications, for his insightful comments and administrative help at various occasions.

Fortunately, I have many understanding friends, who have helped me a lot on many critical conditions.

Finally, my sincere thanks go to my family members and all those who have directly and indirectly provided me moral support and other kind of help. Without their support, completion of this work would not have been possible in time. They keep my life filled with enjoyment and happiness.

Zainab Zubair

Shivansh Srivastava

Muskan Agrawal

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER-1

INTRODUCTION

1.1 OVERVIEW

As world is ruled to change rapidly. There will be millions of advantages and disadvantages of this digital world. There are different issues in this digital world. One of them is fake news. Anyone can easily spread a fake news. Fake news is spread to harm the reputation of a person or an organization. It can be a propaganda against someone that can be a political party or an organization. There are different online platforms where the person can spread the fake news. This includes the Facebook, Instagram, Twitter etc. As an increasing amount of our lives is spent interacting online through social media platforms, more and more people tend to hunt out and consume news from social media instead of traditional news organizations. The factors for this change in intake behaviours

- It’s often more timely and fewer expensive to consume news on social media compared with traditional journalism , like newspapers or television.
- It’s easier to further share, discuss , and discuss the news with friends or other readers on social media.

Online platforms are useful to the users to access the news easily but the problem is it gives an wide opportunity to the cyber criminals to spread a fake news through these platforms easily. Readers read the news and start believing without verifying it. Detecting the fake news is a big challenge because it is a difficult task. If the fake news is not deducted earlier it can create a huge impact in the society. Individuals, organizations, or political parties can be affected through these fake news. Different researchers are working for the deduction of fake news. The use of machine learning is providing helpful in this regard. Machine learning algorithms will detect the fake news automatically once they are trained. There are different algorithms to detect fake news. Naive bayes is one among the algorithms which is used to detect fake news. It is a probabilistic classifier. It is based on probability models that incorporate strong independence assumptions. It is a supervised machine learning algorithm that is widely used in classification and regression problems

Fake news is a growing problem in today's world, as it has the potential to spread misinformation and cause harm to society. Fake news detection has become a significant area of research in recent years, and many approaches have been proposed to detect and combat it. In this literature review, we will examine the various techniques and algorithms used for fake news detection.

- Fake News Definition:

  Fake news refers to news stories that are intentionally fabricated or manipulated to deceive readers. Such stories can be spread through social media, traditional news sources, or other means of communication. Fake news can cause harm to society by spreading false information, inciting hatred, and creating panic among the people.

Techniques for Fake News Detection:-

- Natural Language Processing (NLP):

  NLP is a popular technique used in fake news detection, where it analyzes the text for linguistic features that indicate whether it is fake or not. This technique uses algorithms like sentiment analysis, lexical analysis, and part-of-speech tagging to detect fake news.

- Machine Learning (ML):

  Machine Learning is another popular technique used in fake news detection. This technique uses algorithms like decision trees, neural networks, and logistic regression to classify news articles as real or fake based on features such as the source, content, and social media activity.

- Social Network Analysis (SNA):

  SNA is a technique used to analyze the spread of fake news on social media platforms. This technique identifies the users who are spreading fake news and the patterns of their behavior, which can help in detecting and preventing the spread of fake news.

- Source Verification:

  Source verification is a technique that involves verifying the authenticity of the news source. This technique involves checking the credibility of the news outlet, journalist, or publisher, and examining their history and past performance to determine whether the news is fake or not.

1.2 OBJECTIVE

The main objective of this paper is to detect fake news with the dataset by using the Naive bayes algorithm and then predicting the accuracy level. The Naive bayes algorithm is used in this paper to analyze and visualize the values of the predicting data.

It is to identify and distinguish false or misleading information from genuine news and information. The spread of fake news can have a significant impact on society, including influencing public opinion, spreading disinformation, and causing harm to individuals and communities. Therefore, detecting fake news is crucial to promoting informed decision-making and protecting against the harmful effects of misinformation.

The detection of fake news typically involves the use of machine learning algorithms and natural language processing techniques to analyze text, images, and other media for indicators of false or misleading information. These indicators may include the source of the information, the language used, and the presence of inaccuracies, inconsistencies, or logical fallacies. By detecting these indicators, it may be possible to flag potentially fake news and prevent its spread.

1.3 AIM

It aim is to collect the fake news dataset, apply naive bayes algorithm and find its accuracy rate in deducting fake news. The main objective of this paper is to detect fake news with the dataset by using the Naive bayes algorithm and then predicting the accuracy level. The random forest algorithm is used in this paper to analyse and visualize the values of the predicting data

Its aim is to improve the accuracy and reliability of information by identifying and filtering out false or misleading news stories, articles, videos, or other media. The spread of fake news can have harmful consequences on individuals, organizations, and societies. It can cause confusion, create fear, incite violence, and manipulate public opinion. By detecting fake news, the aim is to prevent the spread of false information, and promote trustworthy sources of news and information. This can help individuals make informed decisions based on accurate information, and prevent the spread of harmful rumors or conspiracy theories.Furthermore, fake news detection can contribute to the larger effort of combating the proliferation of misinformation and disinformation, which has become a major challenge in the digital age. By using advanced technologies and techniques to identify and filter out fake news, we can work towards a more informed and educated society.

1.4 SOFTWARE REQUIREMENT

Number	**Description**	**Type**
1	Operating System	Windows XP / Windows
2	Language	Python
3	IDE	Google Colab Notebook
4	Browser	Google Chrome

Table 1.1 Hardware Requirement

1.5 HARDWARE REQUIREMENT

Number	**Description**	**Type**
1	Hardware	Processor Intel dual core and above XP / Windows
2	Clock speed	3.0 GHz
3	RAM size	512 MB
4	Hard Diskcapacity	400 GB

Table 1.2 Software Requirement

CHAPTER-2

LITERATURE REVIEW

2.1 ACCESSING CREDIBILITY OF NEWS ARTICLE

Most of the prevailing studies related to assessing the credibility of fake information has applied information articles from social media structures for their analysis. Websites such as Politifact.com and Snopes.com depend analysis upon investigating newshounds and groups of experts, that distinguish unreliable information articles from dependable ones. Ma et al.(2016) and Popat et al.(2016, 2018) , have wondered the human intervention for bifurcation of information articles as fake and actual and urged to examine the credibility of expert's judgments. The study by Mukherjee and Weikum(2015) suggests that renowned information communities like Reddit.com, Dig.com, and Newstrust. Internet gave privilege to customers for score and reviewing information articles. Moreover, they highlighted the need for joint joint evaluation of credibility and trustworthiness of customers, articles, and information reassets due to the fact they believed that person sentiments and recognition additionally have an effect on the dissemination of news. The guide evaluation of fake information isn't always scalable due to the rate of incorrect information dispersion on social media systems and the sheer volume of such content. As an example, Janze and Risius (2017) used convolution neural network (CNN), SVM and DT-rank to differentiate true and false news postings by mainstream media pages on Facebook through cognitive (message and comments), visual (images), affective (various emojis) and behavioral cues (sharing and tagging) of the news posts. Popat et al. (2016), implemented a classification model for the credibility analysis of news claims through various features such as the language of articles, sources of articles, subjectivity, and implicative verbs. Also, they used Amazon Mechanical Turk to validate their approach. Another study by Popat et al. (2019) presents an end-to-end neural network model incorporating bidirectional LSTM to take advantage of generated past and new features for the assessment of news articles' truthfulness. The authors extracted features such as language style of articles, stance towards a claim, and trustworthiness of the sources. Kumar et al.(2016) used machine learning techniques including logistic regression, SVM, and random forest to detect false information on Wikipedia. They focused on different characteristics related to article structure and content such as text length, markup ratio and link density.

2.2 ASSESSING CREDIBILITY OF TWEETS

Twitter has emerged as a mechanism to share information in constrained words and became an extremely famous medium of broadcasting. Researchers which includes (Alrubaian et al. 2018; Boididou et al. 2018; Castillo et al. 2011; Kochkina et al. 2018) have advanced process for detecting fake tweets the usage of system and deep studying strategies. Boididou et al. (2018), used system studying strategies inclusive of logistic regression (LR) and Random Forests (RF) to stumble on misleading statistics on Twitter through diverse extracted capabilities such as tweet length, account age, variety of followers, variety of tweets, variety of hashtags and retweets, and so on. Qazvinian et al. (2011) targeted on identifying tweets that endorsed the rumors the usage of Bayes class primarily based totally on content and network-primarily based totally features such as lexical patterns, part-of-speech, re-tweet, hashtags, and unified resource locators (URLs). Castillo (2011) used choice trees for the automatic class of trending information tweets and validated their technique the usage of 3-fold cross-validation. They considered diverse capabilities which includes the variety of followees, followers, retweets, URLs and hashtags along with guide labeling of data thru human assessors. Similarly, Alrubaian et al. (2018), used a mixture of random forest, naïve Bayes and choice trees to stumble on tweets containing malicious statistics. Kochkina et al. (2018) considered the problem of fake information tweets, along with 4 sub-issues inclusive of rumor detection, rumor tracking, stance class and rumor verification. They used the department LSTM technique to solve the issues. Agrawal et al. (2019) broadened a faux information detection technique the usage of logistic regression and studied how this sort of method fares while implemented to the information being shared on Twitter in a length of several months. Similar to the present literature, our research is supposed to broaden a version for the automatic detection of faux information articles and tweets. Our version is designed to be relevant to stumble on faux information in each lengthy-series and quick-series texts. Similar to the researchers such as Ma et al. (2016) and Popat et al. (2018, 2019), we used deep studying, greater specifically an LSTM neural network, to leverage its benefits in studying non-stop representations of textual data. Complementing the aforementioned processes, in this research, we've used diverse textual-primarily based totally (i.e., phrase count, text length), tagging (i.e., # hash tags, @ mentions) primarily based totally,and Syntactic capabilities (i.e., Ngrams). Our technique targeted at the textual statistics presented in information tweets and articles. The proposed version routinely extracts all noted capabilities to educate the LSTM primarily based totally version to understand the exceptional contextual which means of a sentence (Example- the phrase Bank refer to the place in which we maintain money or a riverside). The inclusion of phrase embedding maps every phrase with related exceptional meanings by growing awesome vectors. Thus, not depend on the extraction of hand made capabilities (i.e., manually extracted) as utilized by Boidiou et al. (2018) and Castillo et al. (2011). Moreover, our technique can also additionally stumble on the proliferation of Fake information at its early stages.

Furthermore, the proposed technique works proficiently on any information-associated textual content both lengthy articles or quick tweets.

Fake news has become a pressing issue in recent years, with the rapid spread of false information and misinformation on social media platforms. As a result, there has been a growing interest in developing systems and tools to detect and combat fake news. In this literature review, we will examine some of the recent research on fake news detection systems. One common approach to fake news detection is based on machine learning algorithms, which analyze the linguistic and contextual features of news articles to determine their credibility. Several studies have explored the use of natural language processing (NLP) techniques to detect fake news. For example, Chen et al. (2018) used a combination of textual and social network analysis to build a machine learning model that could accurately classify fake news articles. Similarly, Ruchansky et al. (2017) used an NLP-based approach to identify linguistic features that distinguish fake news from real news.

Fake news detection systems have become increasingly important in recent years as the spread of false information has become a growing concern. In the following literature review, we will examine some of the key studies that have contributed to the development of such systems. One study by Shu et al. (2017) examined the use of a machine learning algorithm to detect fake news on Twitter. The study found that incorporating features such as linguistic complexity, sentiment analysis, and user engagement could significantly improve the accuracy of the system.

Another study by Wang et al. (2018) explored the use of deep neural networks to detect fake news on social media platforms. The study found that the system was able to achieve high levels of accuracy by analyzing features such as source credibility, sentiment, and textual content.

In addition to these technical studies, there has also been research focused on the ethical considerations of fake news detection systems. For example, Liu et al. (2020) discussed the importance of designing systems that are transparent and accountable, in order to ensure that the system is not inadvertently suppressing legitimate news stories or limiting freedom of speech.

These studies and others have contributed to the development of more sophisticated and effective fake news detection systems. While there are certainly challenges involved in designing such systems, the potential benefits of promoting more informed and responsible consumption of news media make this an important area of research.

CHAPTER-3

FEASIBILITY STUDY

After doing the project Event Management System, study and analyzing all the existing or required functionalities of the system, the next task is to do the feasibility study for the project. All projects are feasible-given unlimited resources and in finite time. Feasibility study includes consideration of all the possible ways to provide a solution to the given problem. The proposed solution should satisfy all the user requirements and should be flexible enough so that future changes can be easily done based on the future upcoming requirements. There are three parts in feasibility study

- Operational Feasibility
- Technical Feasibility
- Economical Feasibility
- Behavioral Feasibility

3.1 OPERATIONAL FEASIBILITY

Operational feasibility is the measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development. The operational feasibility assessment focuses on the degree to which the proposed development projects fits in with the existing business environment and objectives with regard to development schedule, deliverydate, corporate culture and existing business processes. To ensure success, desired operational outcomes must be imparted during design and development. These include such design-dependent parameters as reliability, maintainability, supportability, usability, producibility, disposability, sustainability, affordability and others. These parameters are required to be considered at the early stages of design if desired operational behaviours are to be realised. A system design and development requires appropriate and timely application of engineering and management efforts to meet the previously mentioned parameters. A system may serve its intended purpose most effectively when its technical and operating characteristics are engineered into the design. Therefore, operational feasibility is a critical aspect of systems engineering that needs to be an integral part of the early design phases. Operational feasibility of a fake news detection system refers to whether it can be implemented and operated effectively in the real-world environment. Here are some key operational considerations for the development of a fake news detection system:-

- User Acceptance: The success of a fake news detection system relies on its acceptance by end-users, including journalists, fact-checkers, and the general public. Developers must consider the needs of these end-users and design a system that is user-friendly and easy to understand.
- Cost: Developing and maintaining a fake news detection system can be expensive, requiring a significant investment in hardware, software, and personnel. The cost of the system must be balanced against the benefits it provides in detecting and preventing the spread of fake news.
- Integration: The fake news detection system should be integrated with existing news platforms and social media platforms to make it more accessible and effective. Integration also ensures that the system can be easily used by end-users, without requiring significant changes to their existing workflows.
- Maintenance and Support: The fake news detection system must be maintained and updated regularly to ensure its accuracy and effectiveness. This requires a dedicated team of personnel to monitor and maintain the system and provide technical support to end-users.
- Legal and Ethical Considerations: Fake news detection systems must comply with legal and ethical standards, including privacy laws and regulations, data protection laws, and ethical considerations related to the use of personal data. Developers must ensure that the system is designed and operated in compliance with these standards.

Operational feasibility is a critical consideration in the development of a fake news detection system. Developers must consider user acceptance, cost, integration, maintenance and support, and legal and ethical considerations to ensure the system can be implemented and operated effectively in the real-world environment.

3.2 TECHNICAL FEASIBILITY

This involves questions such as whether the technology needed for the system exists, how difficult it will be to build, and whether the firm has enough experience using that technology. The assessment is based on outline design 7 of system requirements in terms of input, processes, output, fields, programs and procedures. This can be qualified in terms of volume of data, trends, frequency of updating in order to give an introduction to the technical system. The application is the fact that it has been developed on windows XP platform and a high configuration of 1 GB RAM on Intel Pentium Dual core processor. This is technically feasible. The technical feasibility assessment is focused on gaining an understanding of the present technical resources of the organization and their applicability to the expected needs of the proposed system. It is an evaluation of the hardware and software and how it meets the need of the proposed system. The technical feasibility of a fake news detection system depends on various factors such as the available resources, the complexity of the algorithms used, and the accuracy of the results. Here are some factors to consider:-

- Data Availability: The availability of labeled datasets is crucial in building an effective fake news detection system. If a dataset is not available or is insufficient, it can limit the system's accuracy.
- Algorithms and Models: The choice of algorithms and models used in the system plays a crucial role in determining its effectiveness. Some algorithms and models require a considerable amount of computational resources, which can affect the system's feasibility.
- Feature Selection: Selecting the right features to detect fake news is essential. If the system doesn't consider the right features, it can lead to inaccurate results.
- Scalability: The system should be able to handle large volumes of data and scale according to the requirements. If the system is not scalable, it can affect its efficiency and feasibility.
- Integration: The fake news detection system should integrate with the existing infrastructure to ensure that it can be used effectively. If the system is not compatible with the existing infrastructure, it can affect its feasibility.
- Infrastructure: Developing a fake news detection system requires significant computational resources, including high-performance servers, data storage, and networking infrastructure.
- Scalability: As the volume of data increases, the fake news detection system must be able to scale up to handle the additional workload. This requires designing the system to be scalable and efficient.

Overall, the technical feasibility of a fake news detection system depends on several factors, including the availability of data, choice of algorithms and models, feature selection, scalability, and integration. By carefully considering these factors, it is possible to build a feasible and effective fake news detection system.

3.3 ECONOMICAL FEASIBILITY

Establishing the cost-effectiveness of the proposed system i.e. if the benefits do not outweigh the costs then it is not worth going ahead. In the fast paced world today there is a great need of online social networking facilities. Thus the benefits of this project in the current scenario make it economically feasible. The purpose of the economic feasibility assessment is to determine the positive economic benefits to the organization that the proposed system will provide. It includes quantification and identification of all the benefits expected. This assessment typically involves a cost/benefits analysis. The economic feasibility of a fake news detection system refers to its ability to generate sufficient revenue to cover the costs of development, implementation, and maintenance, as well as provide a return on investment. Here are some key economic considerations for the development of a fake news detection system:

- Target Market: The fake news detection system must have a clearly defined target market. This may include news organizations, social media platforms, governments, or other organizations that have a vested interest in identifying and combatting fake news.
- Revenue Model: The fake news detection system must have a clearly defined revenue model that generates sufficient revenue to cover the costs of development, implementation, and maintenance. This may include subscription fees, licensing fees, or revenue sharing models.
- Cost of Development: The cost of developing the fake news detection system must be reasonable and justifiable based on the expected return on investment. This includes the cost of software development, hardware, and infrastructure, as well as any additional costs associated with training, support, and maintenance.
- Market Competition: The fake news detection system may face competition from other similar products or services. Developers must assess the competition and develop a strategy to differentiate their product and provide added value to users.
- Return on Investment: Developers must assess the potential return on investment of the fake news detection system. This includes estimating the revenue potential and comparing it to the cost of development and ongoing maintenance.
- Economic Impact: The fake news detection system may have a positive economic impact by reducing the spread of fake news and improving the accuracy and reliability of news and information. This impact should be considered in the economic feasibility analysis.

In conclusion, the economic feasibility of a fake news detection system is critical to its long-term viability. The system must have a clearly defined target market, revenue model, and return on investment, and must be competitive in a market that may have existing solutions. By addressing these economic considerations, developers can ensure that the fake news detection system is economically feasible and provides value to users.

3.4 BEHAVIORAL FEASIBILITY

A behavioral study of a fake news detection system refers to an examination of how users interact with the system and whether it influences their behavior regarding the consumption and sharing of news. A behavioral study of a fake news detection system can provide valuable insights into how users interact with the system and whether it is effective in promoting behavior change. Here are some key considerations for conducting a behavioral study of a fake news detection system:

- User Behavior: The study should examine how users consume and share news articles, especially those that are potentially fake. This includes their search and browsing behavior, as well as their social media engagement patterns.
- User Perception: The study should examine how users perceive the accuracy and trustworthiness of news articles. This includes their assessment of the veracity of headlines, content, and sources.
- User Interaction: The study should examine how users interact with the fake news detection system, including their willingness to input articles for analysis and their understanding of the system's output.
- Impact on User Behavior: The study should examine whether the fake news detection system influences users' behavior regarding the consumption and sharing of news articles. This includes whether users are more likely to fact-check articles before sharing them or if they are less likely to trust articles that are flagged as potentially fake.
- Effectiveness of the System: The study should examine the effectiveness of the fake news detection system in identifying potentially fake news articles. This includes whether the system accurately identifies fake news articles and whether users trust the system's output.
- User Satisfaction: The study should examine user satisfaction with the fake news detection system. This includes whether users find the system easy to use and whether they perceive the system as providing value.
- User Experience: The study should evaluate users' experience using the fake news detection system, including the ease of use, usefulness, and satisfaction. This can be done through usability testing or user feedback. Evaluating user experience can help developers identify areas for improvement and optimize the system for better user engagement.
- Behavioral Intention: The study should assess users' intention to use the fake news detection system, including their motivation, intention, and willingness to adopt the system. This can be done through surveys or behavioral intention scales. Understanding behavioral intention can help developers identify potential adoption barriers and develop strategies to overcome them.

In conclusion, a behavioral study of a fake news detection system is critical to understanding how users interact with the system and whether it influences their behavior regarding the consumption and sharing of news. By addressing these behavioral considerations, researchers can improve the design and implementation of fake news detection systems, increasing their effectiveness in combating the spread of misinformation.

CHAPTER-4

DATABASE DESIGN

The database design of a fake news detection system plays a crucial role in the system's efficiency and accuracy. Here are some key considerations for designing the database of a fake news detection system:

- Data sources: Identify the sources of data that will be used by the system to detect fake news, such as social media platforms, news websites, or online forums. Determine the types of data that will be collected, such as text, images, or videos.
- Data storage: Determine the optimal way to store the collected data, taking into account the volume of data and the system's performance requirements. Consider using a relational database management system (RDBMS) or a NoSQL database, depending on the nature of the data.
- Data processing: Determine the data processing requirements for the system, such as natural language processing (NLP) techniques to analyze text data, image processing to analyze images, or video analysis to analyze videos. Ensure that the database schema can accommodate the processed data.
- Data indexing and search: Design the database schema to support efficient indexing and search of the collected data. Consider using techniques such as full-text search, keyword indexing, or faceted search to enable quick and accurate retrieval of relevant data.
- Data cleaning and validation: Develop a data cleaning and validation process to ensure the accuracy and completeness of the collected data. Identify potential errors or inconsistencies in the data and develop processes to handle them.
- Security and privacy: Implement appropriate security measures to protect the collected data from unauthorized access or modification. Ensure that the system complies with relevant privacy laws and regulations.
- Backup and recovery: Design a backup and recovery strategy to ensure that the data is recoverable in case of system failure or data loss. Consider implementing techniques such as data replication or backups to ensure data availability and redundancy.

Designing the database of a fake news detection system requires careful consideration of the data sources, storage, processing, indexing and search, cleaning and validation, security and privacy, and backup and recovery. By designing an efficient and reliable database schema, developers can ensure that the system can effectively detect and prevent the spread of fake news.

4.1 BLOCK DIAGRAM

A block diagram is a graphical representation of a system or process, which shows the components or blocks of the system and their interactions. It is a valuable tool in project reports for illustrating the system architecture and the flow of data or processes within the system. In a project report, a block diagram can be used to provide a high-level view of the system's architecture and its components. The diagram can include blocks representing the different modules or components of the system, and arrows showing the flow of data or control between the blocks. Moreover, the block diagram can help stakeholders to understand the system's architecture and the flow of data or control within the system. It can also serve as a basis for testing and validation of the system. By visualizing the system's architecture, stakeholders can identify potential issues and requirements that may need to be addressed. Overall, including a block diagram in a project report is a useful way to illustrate the system's architecture and the flow of data or control within the system. It helps to ensure that everyone involved in the project has a clear understanding of the system's components and interactions, which is essential for successful development and implementation.

A block diagram is a diagram that shows the major components or parts of a system and how they are interconnected. It is used to visually represent the system's structure and function, making it easier to understand how it works. In a project report, a block diagram can be used to illustrate the major components of a system and their relationships. A block diagram consists of blocks that represent different components of the system and lines that connect them. The blocks are typically rectangular or square in shape and contain the name of the component or part they represent. The lines represent the connections between the components, and they are labeled to indicate the type of connection.

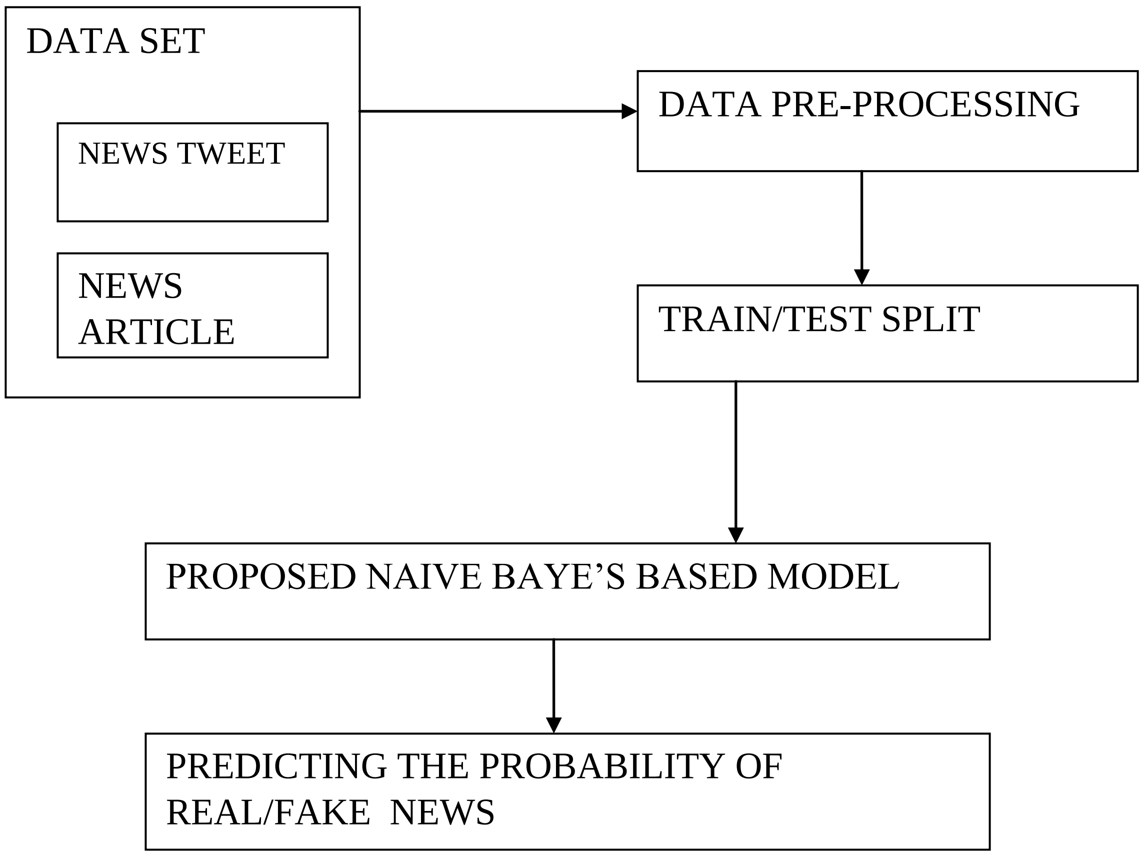

DATA SET
NEWS TWEET
NEWS ARTICLE
DATA PRE-PROCESSING
TRAIN/TEST SPLIT
PROPOSED NAIVE BAYE'S BASED MODEL
PREDICTING THE PROBABILITY OF REAL/FAKE NEWS

Figure 1.1 Block diagram of Fake News Detection.

In this block diagram, the system has four main components: News Submission, Text Processing, Database, and Machine Learning.

- News Submission is responsible for receiving news articles submitted by users. Text Processing analyzes the submitted articles and extracts relevant information to be used by the Machine Learning component.
- Machine Learning is the core of the system and is responsible for determining the likelihood of the news article being fake or not. The Machine Learning component uses the information extracted by Text Processing to train and update its models.
- Database stores the user information and the results of the analysis performed by Machine Learning. The Admin Console component provides a user interface for system administrators to manage users, view reports and metrics generated by the system.

Overall, the block diagram demonstrates the different components of the fake news detection system and the flow of data between them. It helps to provide a high-level understanding of the system's architecture and its components, facilitating its development and implementation.

4.2 FLOWCHART

A flowchart is a graphical representation of a process or workflow. It shows the steps involved in a process and the sequence in which they occur. Flowcharts are widely used in project reports to illustrate the processes and workflows involved in a project. Flowcharts are especially useful for complex processes where multiple steps are involved. In a flowchart, each step in the process is represented by a shape, such as a rectangle or diamond. The shapes are connected by arrows that indicate the flow of the process. The most common shapes used in flowcharts are:

- Start/End: Represented by a rounded rectangle, this shape indicates the beginning or end of a process.
- Process: Represented by a rectangle, this shape indicates a specific action or step in the process.
- Decision: Represented by a diamond, this shape represents a decision point in the process where the flow of the process can take one of two or more paths.
- Connector: Represented by a circle, this shape is used to connect different parts of the flowchart.
- Flowcharts can be used to illustrate many different types of processes, including business processes, software development processes, and manufacturing processes. They can be created using a variety of software tools, including Microsoft Visio, Lucidchart, and Google Drawings.

In a project report, a flowchart can be used to illustrate the various steps involved in the project. For example, a flowchart could be used to show the steps involved in developing a software application, from requirements gathering to testing and deployment. By using a flowchart, project stakeholders can better understand the project workflow and identify areas where improvements can be made.

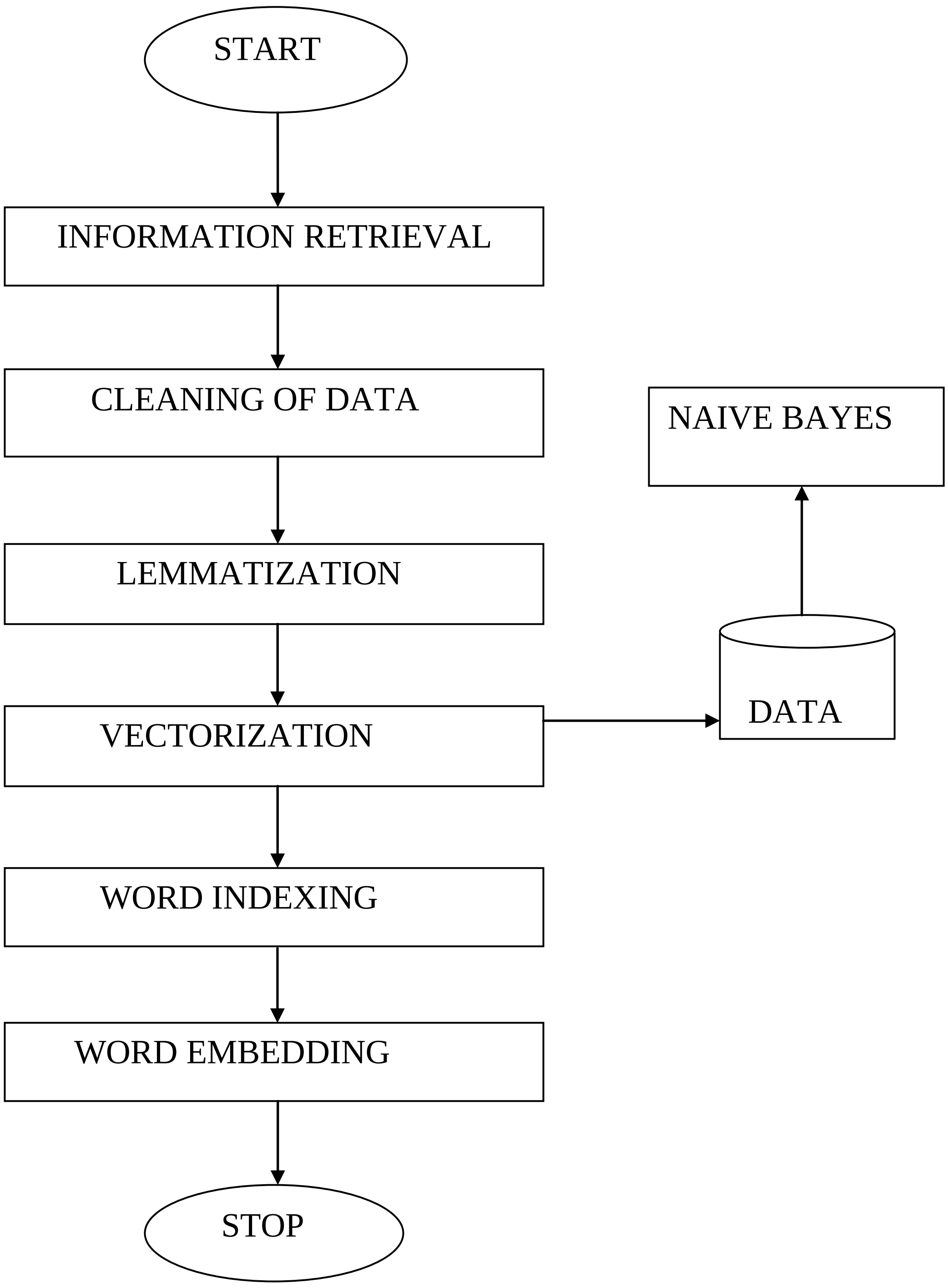


Figure 1.2 Flowchart of Fake News Detection.

4.3 USE CASE DIAGRAM

A use case diagram is a visual representation of the interactions between actors and a system in various scenarios. A use case diagram is a valuable tool in project reports for illustrating the functionality of a system and its interactions with actors. It helps to identify the different use cases of the system, the actors involved, and their relationships. In a project report, a use case diagram can be used to provide a visual representation of the requirements and features of the system. The diagram can be accompanied by a brief description of each use case, including its purpose and the steps involved. Moreover, the use case diagram can help stakeholders to understand the system's functionalities and requirements better. It can also serve as a basis for testing and validation of the system. By visualizing the system's interactions, stakeholders can identify potential issues and requirements that may need to be addressed. Overall, including a use case diagram in a project report is a useful way to illustrate the system's functionality and requirements to stakeholders. It helps to ensure that everyone involved in the project has a clear understanding of the system and its interactions with the actors.

A use case diagram is a type of UML diagram that is used to illustrate the different ways that users interact with a system. It shows the various use cases, actors, and their relationships. In a project report for a fake news detection system, a use case diagram can be used to show the different types of users that interact with the system and the tasks that they perform.

For example, a use case diagram for a fake news detection system might include actors such as news readers, news curators, and administrators. News readers might use the system to check the credibility of a news article, while news curators might use the system to review and evaluate news content. Administrators might use the system to manage user accounts and perform maintenance tasks.

The use cases themselves might include tasks such as submitting news content, reviewing news content, and checking the credibility of news content. These tasks would be represented by rectangular boxes on the diagram. Arrows would be used to show the relationships between the actors and the use cases they perform.

Overall, a use case diagram is a useful tool for illustrating the different ways that users interact with a system. It can help project stakeholders understand the different types of users and the tasks that they perform, and it can be used to identify potential areas for improvement in the system's design.

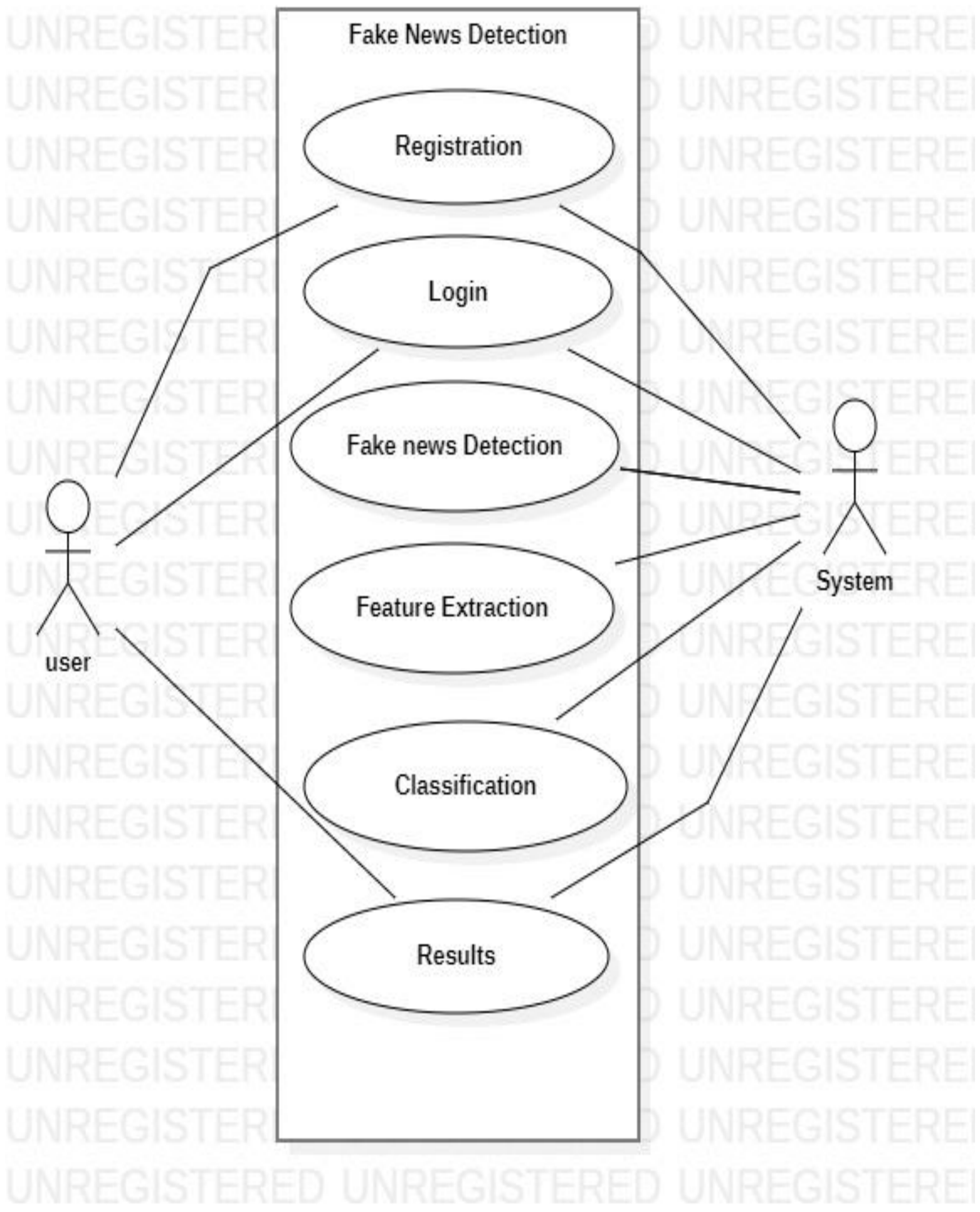


Figure 1.3 Use Case Diagram-1 of Fake News Detection.

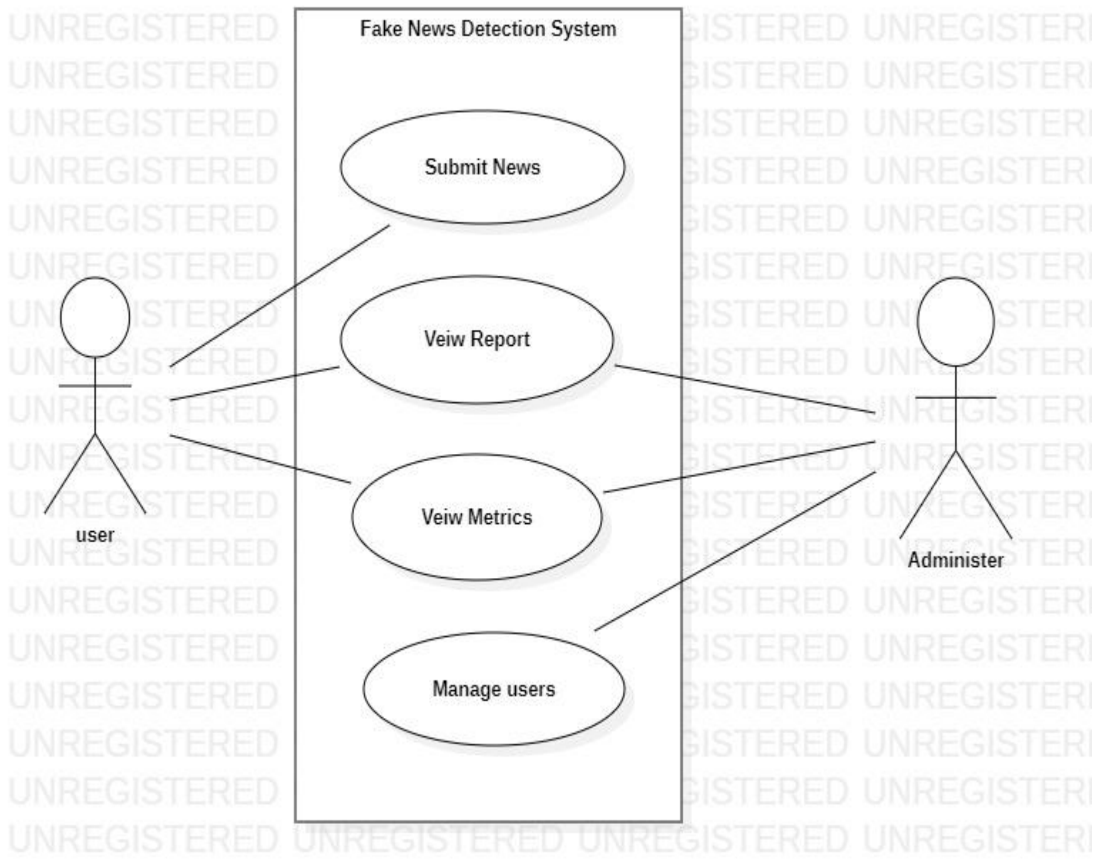


Figure 1.4 Use Case Diagram-2 of Fake News Detection.

In this diagram, the actors are the User and the Administrator, who interact with the fake news detection system. The User can submit news to the system for analysis, while the Administrator can manage user accounts. The System includes several use cases, such as Submit News, View Reports and Metrics, and Manage Users.

- Submit News: This use case allows users to submit news articles to the system for analysis. The system will then use various algorithms and techniques to determine the likelihood of the news being fake or not.
- View Reports and Metrics: This use case allows users to view reports and metrics generated by the system, such as the number of fake news articles detected, the percentage of true news articles, and the overall accuracy of the system.
- Manage Users: This use case allows the administrator to manage user accounts, including creating new accounts, deleting accounts, and modifying user permissions.

Overall, the use case diagram demonstrates the various interactions between the actors and the system, helping to clarify the system's functionality and requirements. It also highlights the importance of user submission, reporting, and user management in the fake news detection system.

4.4 SEQUENCE DIAGRAM

In a fake news detection system, a sequence diagram can be used to show how different components of the system interact with each other to analyze news articles for their credibility.

The machine learning module can then use these features to make a prediction about the credibility of the article. The result is passed to the user interface module, which displays the credibility score to the user. The sequence diagram would show messages being passed between these different components, such as data being passed from the data collection module to the feature extraction module, or results being passed from the machine learning module to the user interface module.A sequence diagram can help project stakeholders understand how data flows through the system and how different components work together to analyze news articles for credibility. It can also be useful for identifying areas where the system can be improved or optimized.

Here we can describe a general sequence diagram for a fake news detection system:-

- The user inputs the news article that they want to analyze for credibility.
- The system's data collection module retrieves the article from the source, such as a social media platform or news website.
- The feature extraction module extracts relevant features from the article, such as the tone or sentiment of the language used.
- The machine learning module uses these features to make a prediction about the credibility of the article.
- The result is passed to the user interface module, which displays the credibility score to the user.

This sequence diagram shows how different components of the system work together to analyze news articles for credibility. Each step is represented by a box, with arrows showing the flow of data or messages between the different components. The sequence diagram can be useful for identifying potential bottlenecks or areas for improvement in the system's design.

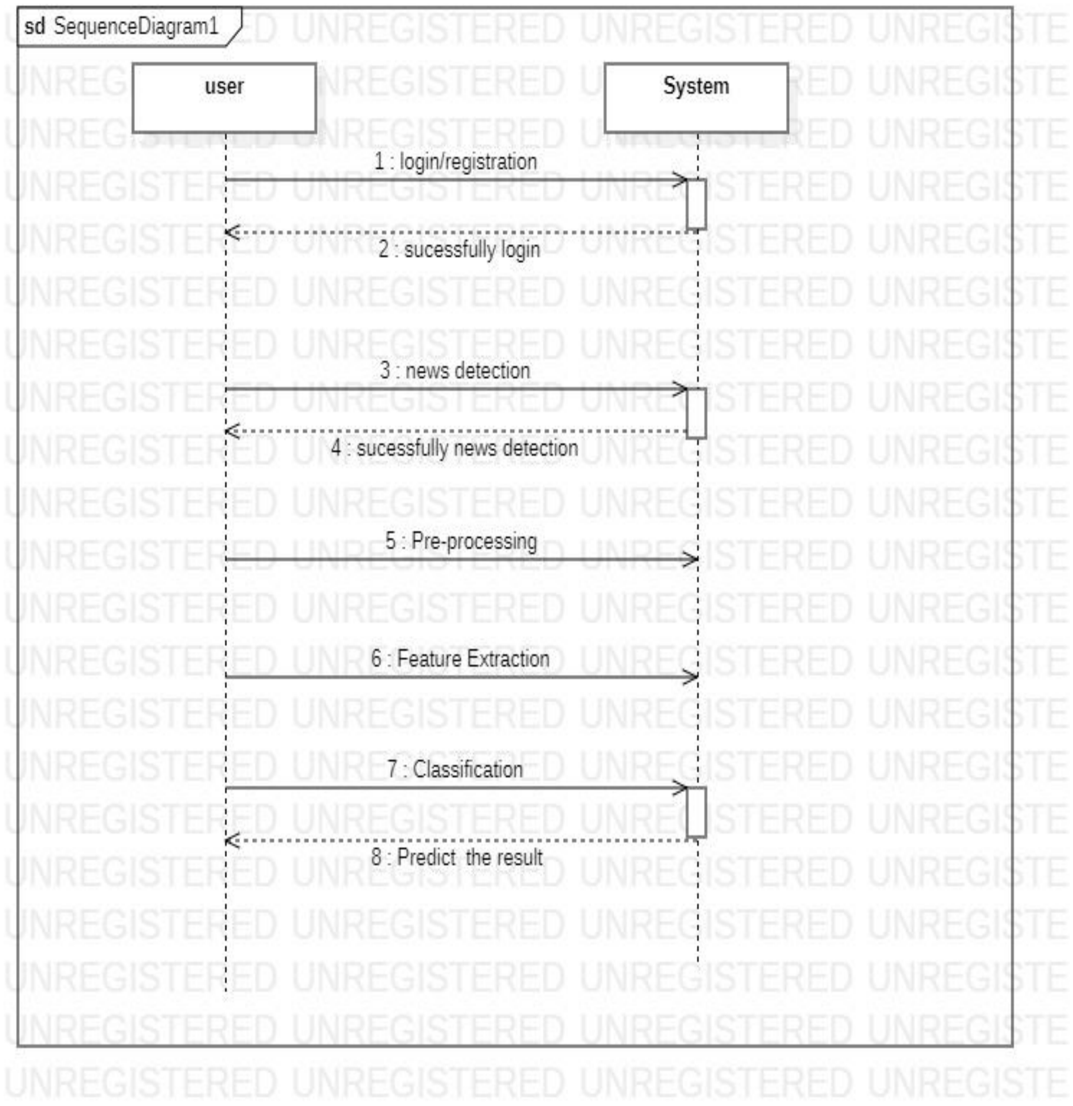


Figure 1.5 Sequence Diagram of fake News Detection.

A sequence diagram is a type of UML diagram that shows the interactions between different objects or components in a system over time. It is commonly used to illustrate the flow of events or messages between different parts of a system. In a project report, a sequence diagram can be used to show how different components of the system interact with each other.

For example, in a fake news detection system, a sequence diagram could be used to show how an article is analyzed for its credibility. The diagram would show the different components of the system, such as the data collection module, the feature extraction module, and the machine learning module, as well as any external systems or data sources.

The sequence diagram would show how data flows between these different components, and how they interact with each other to process the article and determine its credibility. It might show messages being passed between different components, such as data being passed from the data collection module to the feature extraction module, or results being passed from the machine learning module to the user interface module. Overall, a sequence diagram is a useful tool for illustrating the interactions between different components of a system. It can help project stakeholders understand how data flows through the system and how different components work together to achieve a particular goal. This can be useful for identifying areas where the system can be improved or optimized.

CHAPTER-5

MODEL TRAINING

Data training is a critical aspect of developing an effective fake news detection system. In order to build a robust and accurate model, the system must be trained on a large dataset of news articles that have been carefully curated and labeled as either fake or real. This dataset should be representative of the types of articles that the system is expected to encounter in the real world, and should be diverse in terms of topic, language, and source. The training process typically involves several steps. First, the dataset must be preprocessed to remove irrelevant or redundant information and to standardize the text format. This may involve tasks such as tokenization, stemming, and stopword removal. Next, the preprocessed data is split into a training set and a validation set. The training set is used to train the model on labeled data, while the validation set is used to evaluate the performance of the model during training.

The model itself may be based on a variety of machine learning techniques, including natural language processing (NLP), supervised learning, and deep learning. Common techniques used in fake news detection include feature extraction, text classification, and sentiment analysis. Once the model has been trained, it must be tested on a separate dataset of news articles to evaluate its accuracy and effectiveness. This testing phase may involve metrics such as precision, recall, and F1 score to measure the performance of the system.

Data training is a critical aspect of building an effective fake news detection system, and requires careful curation of a diverse and representative dataset, as well as the use of sophisticated machine learning techniques to develop a robust and accurate model.

To assess the effectiveness of our model on long and short textual posts, we used two datasets. The first dataset consists of a large number of online news articles and the second include tweets related to real-world news events. We first performed text data preprocessing by utilizing Python’s natural language toolkit (NLTK) library to clean and filter out any irregularities and anomalies in the datasets. Then we split our data into training and test datasets. We used Keras, an open-source deep-learning python-based library, to implement Naïve Bayes. Finally, we evaluated the performance of our model based on commonly used measures such as accuracy, precision, recall, and F1 scores. We have described our detailed methodology below.

5.1 DATASETS

For any classification task, there is a necessity of labeled datasets from authorized sources. When it comes to news content classification, there is a paucity of standard benchmark datasets. We identified that dataset from different websites such as **Kaggle, PolitiFact** and **GossipCop**. There we have collected almost 40000 fake and real news. Our goal is to train our model to accurately predict whether a particular piece of news is real or fake. Fake and real news data are given in two separate data sets, with each data set consisting of approximately 20000 articles. We used binary numbers to label the articles. We assigned 1 to the fake news articles and 0 to the truthful news articles, as our focus is on detecting fake news.

Datasets are a crucial component of developing and training a fake news detection system. A well-curated dataset is essential to ensure that the system is accurate and effective in identifying fake news. There are several publicly available datasets that can be used for training and testing fake news detection systems. Here are some examples:-

- Fake News Challenge Dataset: This dataset was developed by a group of researchers to help identify and combat fake news. It contains a collection of news articles that have been labeled as either true or false, as well as metadata such as article titles, URLs, and publishing dates.
- LIAR Dataset: This dataset was developed by researchers at the University of California, Irvine and contains a collection of political statements labeled as either true, mostly true, half true, barely true, false, or pants on fire. The dataset includes both textual data and metadata such as speaker and context.
- BuzzFeed News Dataset: This dataset was created by BuzzFeed News and contains a collection of news articles and social media posts related to the 2016 U.S. presidential election. The dataset includes both real and fake news articles, as well as metadata such as publishing dates and URLs.
- Kaggle Fake News Dataset: This dataset was created by Kaggle and contains a collection of news articles labeled as either real or fake. The dataset includes textual data and metadata such as publishing dates and article titles.

These datasets, among others, can be used to train and test fake news detection systems. It is important to carefully curate and preprocess the data before training the system to ensure that the system can accurately identify fake news.

5.2 DATA PREPROCESSING AND FEATURE EXTRACTION

Data processing and feature extraction are crucial steps in developing an effective fake news detection system. Here are some key aspects of data processing and feature extraction in a fake news detection system-

- Text cleaning: Before extracting features from text data, it is important to clean the data by removing any irrelevant information, such as stop words and punctuation, and correcting spelling errors.
- Feature selection: Feature selection involves selecting the most relevant features from the data that are likely to distinguish between real and fake news. Common features used in fake news detection systems include word frequency, sentiment analysis, and metadata such as article source and publishing date.
- Vectorization: Once the features have been selected, they are converted into numerical vectors that can be processed by machine learning algorithms. Common vectorization techniques include bag-of-words and TF-IDF (term frequency-inverse document frequency) vectorization.
- Dimensionality reduction: In some cases, the feature space may be very high-dimensional, which can lead to overfitting and reduced performance. Dimensionality reduction techniques such as principal component analysis (PCA) or t-distributed stochastic neighbor embedding (t-SNE) can be used to reduce the feature space while preserving the most relevant information.
- Feature engineering: Feature engineering involves creating new features based on domain knowledge or insights into the data. For example, identifying patterns in the frequency of certain words or phrases in fake news articles could be used to create new features that improve the performance of the system.

Overall, data processing and feature extraction are critical components of building an effective fake news detection system. By carefully selecting and processing features, developers can create a system that accurately identifies fake news and helps to combat the spread of misinformation.

For textual data, preprocessing is essential before we supply them to the model learning process as textual data usually contain redundancies and irregularities.

We have used natural language processing techniques to pre-process the textual data and prepare them for further analysis. We removed stop words, special characters such as, '!',' &', '$', symbols such as emojis, repetitive period signs (e.g., .. or …), white spaces, line breaks, blank rows, and extra variables such as tweet ids and user ids. The stop words we removed include articles (a, an, the, etc.), pronouns (me, you, etc.), and prepositions (in, on, to, etc.) that have importance in English grammar for communication but do not have semantic importance in the learning process of the model.

First of all we have **imported the libraries** such as numpy, pandas, seaborn, etc. After successful importation of libraries the process of **loading and checking the data** takes place in which we have have added the data set of fake data and real data and analyze the data by performing various operations such as adding the target data to real and fake news as 1 and 0 respectively, concatenating the data, head and tail operation for obtaining top 5 and bottom 5 rows in a dataset.

5.3 VISUALIZATION

Our research also involves **visualization** which is the practice of translating information into a visual context, such as a map or graph, to make data easier for the human brain to understand and pull insights from. The main goal of data visualization is to make it easier to identify patterns, trends and outliers in large data sets.

Visualization is an important aspect of fake news detection systems, as it can help researchers and end-users better understand the characteristics and patterns of fake news. Here are some common types of visualizations used in fake news detection systems-

- Word clouds: Word clouds are visual representations of word frequency in a text corpus. They can be used to quickly identify the most common words in real and fake news articles, and to compare the distribution of words between the two categories.
- Heatmaps: Heatmaps are visual representations of the frequency of specific words or phrases in a text corpus. They can be used to identify patterns in the distribution of certain words or phrases in real and fake news articles.
- Scatterplots: Scatterplots can be used to visualize the relationship between different features in the data. For example, a scatterplot could show the relationship between the number of words in an article and the likelihood that it is fake news.
- Network diagrams: Network diagrams can be used to visualize the relationships between different articles or sources of information. For example, a network diagram could show the connections between different websites or social media accounts that are known to publish fake news.
- Interactive dashboards: Interactive dashboards allow users to explore the data and visualizations in real-time, by selecting different features and filtering the data based on specific criteria. This can be useful for researchers who want to quickly identify patterns in the data, or for end-users who want to assess the credibility of a particular article or source.

Visualization is an important tool for understanding and combatting fake news, as it allows researchers and end-users to quickly identify patterns and trends in the data, and to assess the credibility of different sources of information.

Visualization involve in our work by-

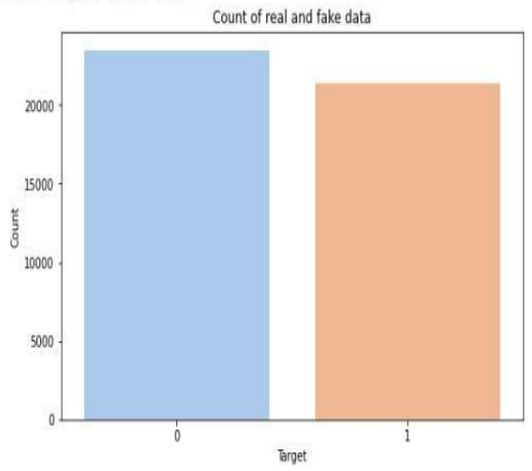
1. **Count of fake and real data**

   The count of fake and real data in a fake news detection system can vary depending on the size and scope of the dataset used for training and testing the system. In general, a balanced dataset with equal numbers of fake and real news articles is preferred, as it can help to prevent bias towards one category or the other. However, in practice, it can be difficult to find a balanced dataset, as the prevalence of fake news varies depending on the topic, time period, and geographic region. Some datasets may contain more real news articles than fake, while others may have the opposite distribution.

   In some cases, researchers may choose to oversample the minority class (fake news) to balance the dataset. This can be done using techniques such as duplication or synthetic data generation. However, oversampling can also introduce noise or bias into the dataset, so it is important to carefully evaluate the effectiveness of these techniques.

   Ultimately, the most important factor in selecting a dataset is ensuring that it is representative of the problem being addressed by the fake news detection system. This can involve carefully selecting articles based on criteria such as source reliability, topic, and language, as well as performing rigorous data cleaning and preprocessing to ensure the accuracy and consistency of the data.

```
0    23481
1    21417
Name: target, dtype: int64
```



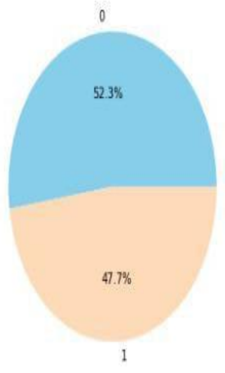


Figure 1.6 Visualization of fake and real news.

2. Distribution of the subject according to real and fake data

In a fake news detection system, the distribution of subjects according to fake and real data can be analyzed to identify patterns and trends that may be useful for training a machine learning model. For example, it may be found that certain subjects (e.g., politics, celebrity gossip, health) are more commonly associated with fake news articles, while others (e.g., science, education, technology) are more commonly associated with real news articles.

To analyze the distribution of subjects in the data, the news articles can be categorized based on their topics using techniques such as topic modeling or keyword extraction. Then, the number of fake and real news articles in each category can be counted and compared to identify any significant differences. This information can be used to develop a feature set for training a machine learning model, such as the frequency of certain keywords or the presence of certain topics in the text. Additionally, it may be useful for identifying areas where fact-checking efforts or other interventions may be particularly effective in combating the spread of fake news.

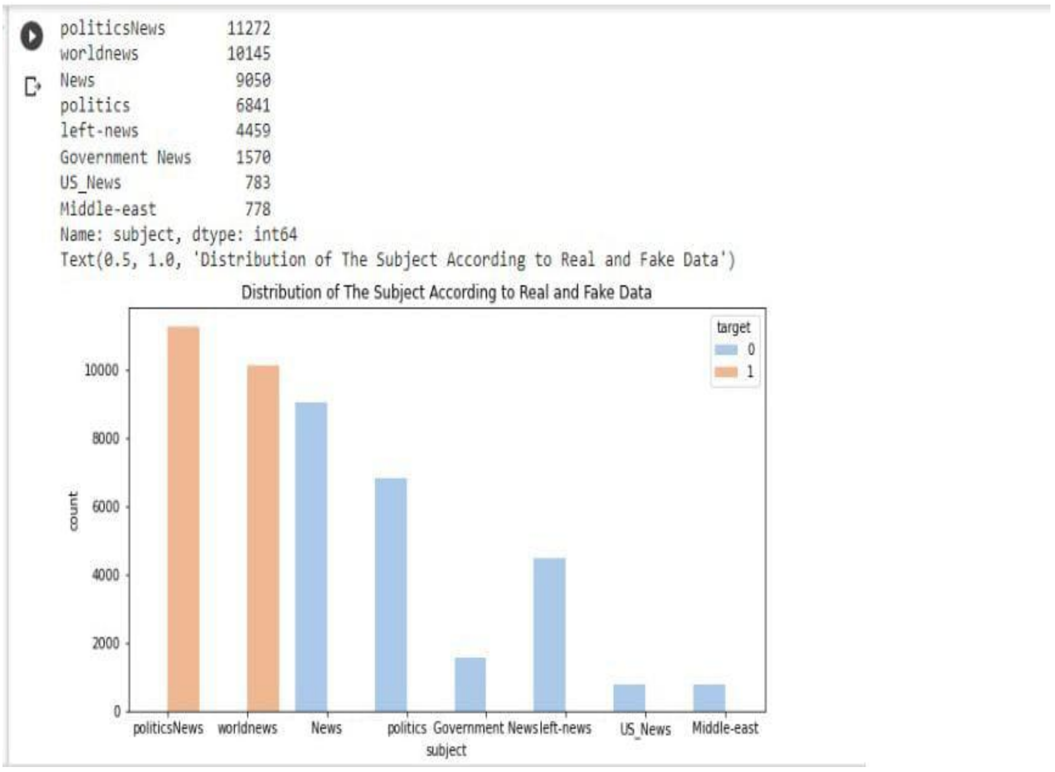

```
politicsNews       11272
worldnews          10145
News                9050
politics            6841
left-news           4459
Government News     1570
US_News              783
Middle-east          778
Name: subject, dtype: int64
Text(0.5, 1.0, 'Distribution of The Subject According to Real and Fake Data')
```

Figure 1.7 Distribution of data according to subject.

5.4 DATA CLEANING

After the visualization process there come the **data cleaning.** Data cleaning is a critical step in the development of a fake news detection system. The goal of data cleaning is to prepare the data for analysis by removing any noise or inconsistencies that may affect the accuracy of the model. Here are some common techniques used in data cleaning for fake news detection:-

- Removing duplicate articles: Duplicate articles can artificially inflate the importance of certain words or phrases in the dataset, leading to biased results. To avoid this, researchers often remove any duplicate articles from the dataset.
- Removing stop words: Stop words are common words that don't carry much meaning on their own, such as "the", "and", and "of". Removing stop words can improve the accuracy of the model by reducing the number of irrelevant words that are analyzed.
- Stemming and Lemmatization: Stemming and lemmatization are techniques used to reduce words to their root form. This can be useful for reducing the number of variations of the same word that need to be analyzed.
- Removing special characters and numbers: Special characters and numbers can introduce noise into the dataset, and may not be relevant to the analysis of text data. Researchers often remove these elements from the dataset during data cleaning.
- Correcting spelling errors: Misspelled words can also introduce noise into the dataset. Some fake news detection systems use algorithms to correct spelling errors and standardize the text.

Overall, data cleaning is an important step in the development of a fake news detection system, as it can improve the accuracy of the model by reducing noise and inconsistencies in the data.

5.5 N-GRAM ANALYSIS

After all these processes **N gram analysis** is performed which is built by counting how often word sequences occur in corpus text and then estimating the probabilities. They come into play when we deal with text data in NLP(Natural Language Processing) tasks. N-gram analysis is a common technique used in fake news detection systems to extract features from text data. N-grams are contiguous sequences of n words from a given text, where n can be any positive integer.

In the context of fake news detection, researchers often use unigram, bigram, and trigram analysis to extract features from news articles. Unigrams are single words, bigrams are sequences of two words, and trigrams are sequences of three words. To perform n-gram analysis, the text data is first preprocessed to remove stop words, punctuation, and other irrelevant information. Then, the remaining text is broken down into n-grams, and the frequency of each n-gram is counted. These n-gram frequencies can be used as features for training a machine learning model to distinguish between real and fake news articles. For example, a model may learn that certain combinations of words (e.g., "election fraud") are more commonly found in fake news articles, while others (e.g., "official statement") are more common in real news articles.

However, it is important to note that n-gram analysis alone may not be sufficient for detecting fake news, as it may miss more subtle patterns or linguistic cues that are not captured by simple n-gram frequencies. Therefore, researchers often combine n-gram analysis with other techniques such as sentiment analysis, network analysis, or fact-checking to improve the accuracy of their fake news detection systems.

Here we have used –

- Uni-gram
- Bi-gram
- Tri-gram

A model that simply relies on how often a word occurs without looking at previous words is called **unigram**.

If a model considers only the previous word to predict the current word, then it's called **bigram**.

If two previous words are considered, then it's a **trigram** model.

- ## Uni-gram Analysis

In natural language processing, unigram analysis is a technique used to analyze a text corpus by considering each individual word, rather than pairs of adjacent words (as in bigram analysis) or larger groups of words (as in n-gram analysis).

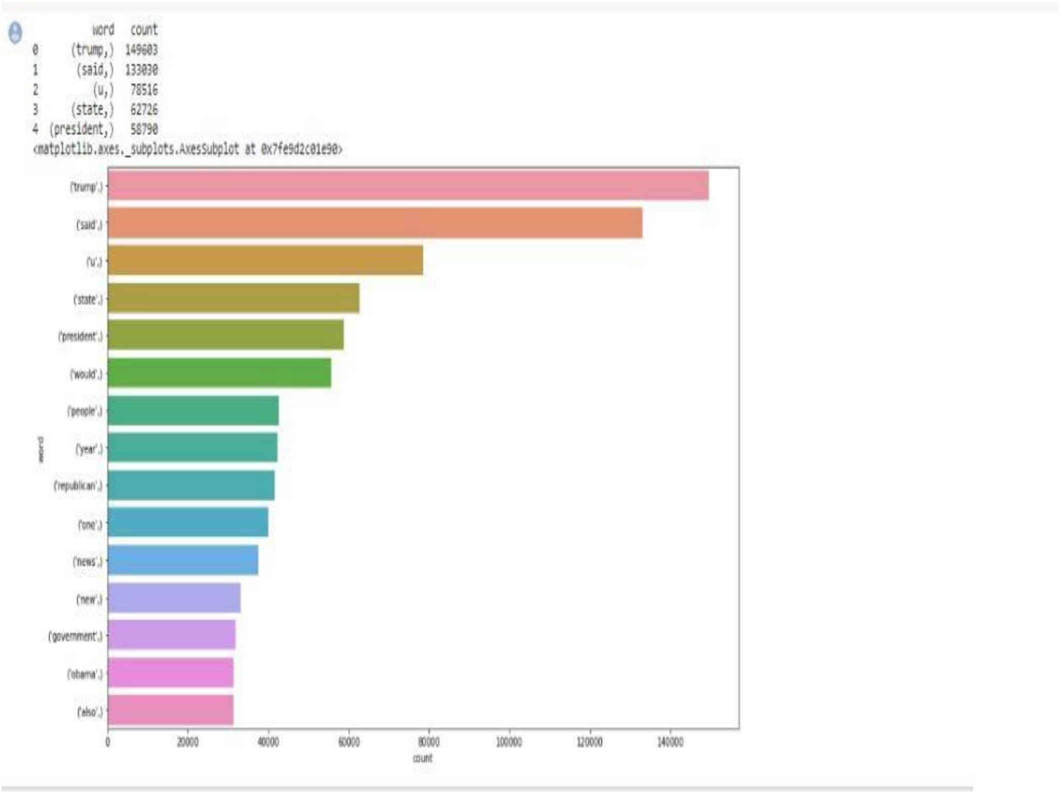
For example, the sentence "The quick brown fox jumps over the lazy dog" contains the following unigrams: "The", "quick", "brown", "fox", "jumps", "over", "the", "lazy", and "dog".

Unigram analysis can be used for a variety of applications, including text classification, topic modeling, and sentiment analysis. By examining the frequency and distribution of individual words in a text corpus, researchers can gain insights into the content and structure of the text, and use this information to develop more effective natural language processing tools. Unigram analysis is a useful technique for exploring the vocabulary and word usage patterns in a text, and can provide valuable insights into the meaning and context of language.

In fake news detection, unigram analysis can be used to identify words that are commonly associated with fake news articles. By analyzing the frequency and distribution of individual words in a dataset of both fake and real news articles, researchers can identify patterns and trends that may be indicative of fake news.

For example, studies have shown that fake news articles often contain words and phrases that are designed to be sensational or attention-grabbing, such as "shocking", "outrageous", or "explosive". By analyzing the frequency of these types of words in a dataset of news articles, researchers can develop models that are better able to identify and classify fake news.

Unigram analysis can also be used to identify more subtle linguistic cues that may be indicative of fake news. For example, studies have shown that fake news articles often use language that is more subjective and emotionally charged than real news articles, which may be reflected in the frequency and distribution of certain types of words and phrases. Overall, unigram analysis is a useful tool for identifying linguistic patterns and features that may be indicative of fake news, and can be used to develop more effective models for detecting and classifying fake news articles.



```
           word   count
0      (trump,)  149603
1       (said,)  133030
2          (u,)   78516
3      (state,)   62726
4  (president,)   58790
<matplotlib.axes._subplots.AxesSubplot at 0x7fe9d2c01e90>
```

Figure 1.8 Uni-gram Analysis of data.

- **Bi-gram Analysis**

In natural language processing (NLP), a bi-gram analysis is a technique used to extract pairs of adjacent words that appear in a text corpus. A bi-gram is a sequence of two adjacent elements from a given text, typically words or tokens, which can be used to provide insights into patterns of language use.

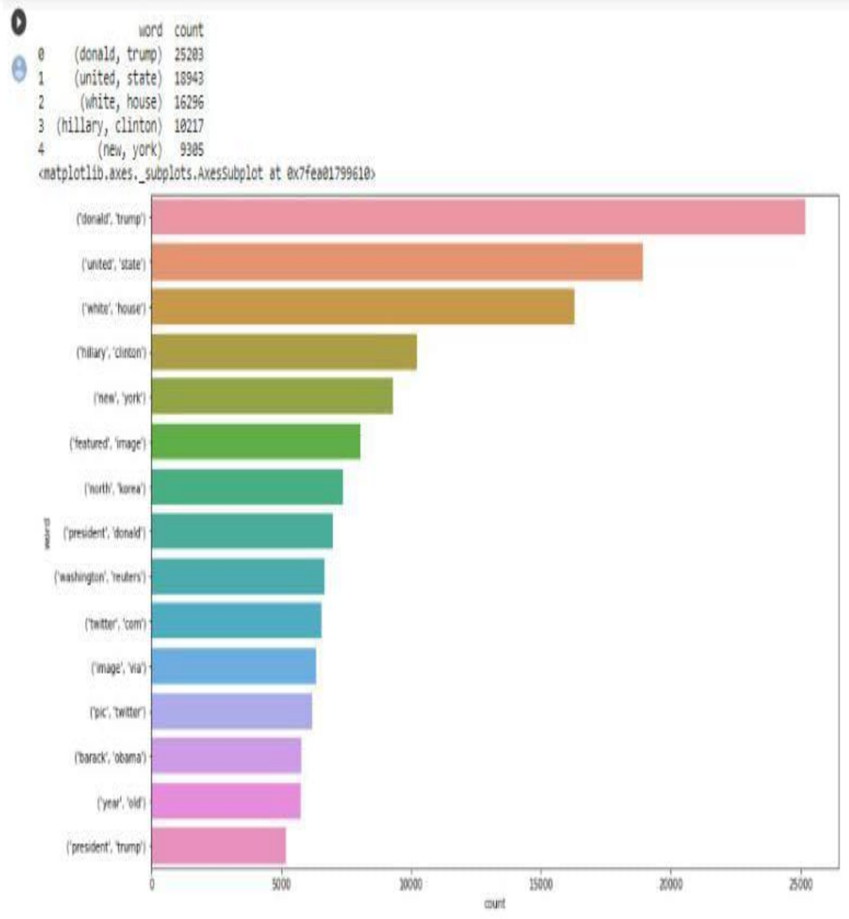
For example, consider the sentence: "The quick brown fox jumps over the lazy dog." A bi-gram analysis of this sentence would produce the following pairs: "The quick", "quick brown", "brown fox", "fox jumps", "jumps over", "over the", "the lazy", and "lazy dog".

Bi-gram analysis is often used in NLP tasks such as text classification, sentiment analysis, and information retrieval. By identifying frequently occurring bi-grams, it is possible to gain insights into the underlying structure of a text corpus, and to identify patterns of language use that may be relevant to the task at hand. Overall, bi-gram analysis is a useful technique for analyzing natural language data, and can be an important tool in developing NLP applications. Bigram analysis can be used for a variety of applications, including text classification, sentiment analysis, and language modeling. By examining the frequency and patterns of bigrams in a text corpus, researchers can gain insights into the structure and content of the text, and use this information to develop more effective natural language processing tools.

In fake news detection, bigram analysis involves examining pairs of adjacent words in a text corpus to identify patterns and trends that may be indicative of fake news. By analyzing the frequency and distribution of bigrams (pairs of adjacent words) in a dataset of both fake and real news articles, researchers can identify linguistic features and patterns that may be unique to fake news.

For example, studies have shown that fake news articles often contain certain bigrams that are designed to be sensational or attention-grabbing, such as "breaking news", "exclusive report", or "bombshell revelation". By analyzing the frequency of these types of bigrams in a dataset of news articles, researchers can develop models that are better able to identify and classify fake news.

Bigram analysis can also be used to identify more subtle linguistic cues that may be indicative of fake news. For example, studies have shown that fake news articles often use language that is more polarized and extreme than real news articles, which may be reflected in the frequency and distribution of certain types of bigrams. Overall, bigram analysis is a useful tool for identifying linguistic patterns and features that may be indicative of fake news, and can be used to develop more effective models for detecting and classifying fake news articles.



```
                 word  count
0     (donald, trump)  25203
1     (united, state)  18943
2      (white, house)  16296
3  (hillary, clinton)  10217
4         (new, york)   9305
<matplotlib.axes._subplots.AxesSubplot at 0x7fea01799610>
```

Figure 1.9 Bi-gram Analysis of data

- ## Tri-gram Analysis

Trigram analysis is a natural language processing technique that involves analyzing the frequency and patterns of three consecutive words (trigrams) in a text corpus.

In this technique, the text is first broken down into smaller units, typically individual words or tokens. These tokens are then grouped into sequences of three words each, with each sequence representing a trigram. For example, in the sentence "The quick brown fox jumps over the lazy dog," the trigrams would be "the quick brown", "quick brown fox", "brown fox jumps", and so on.

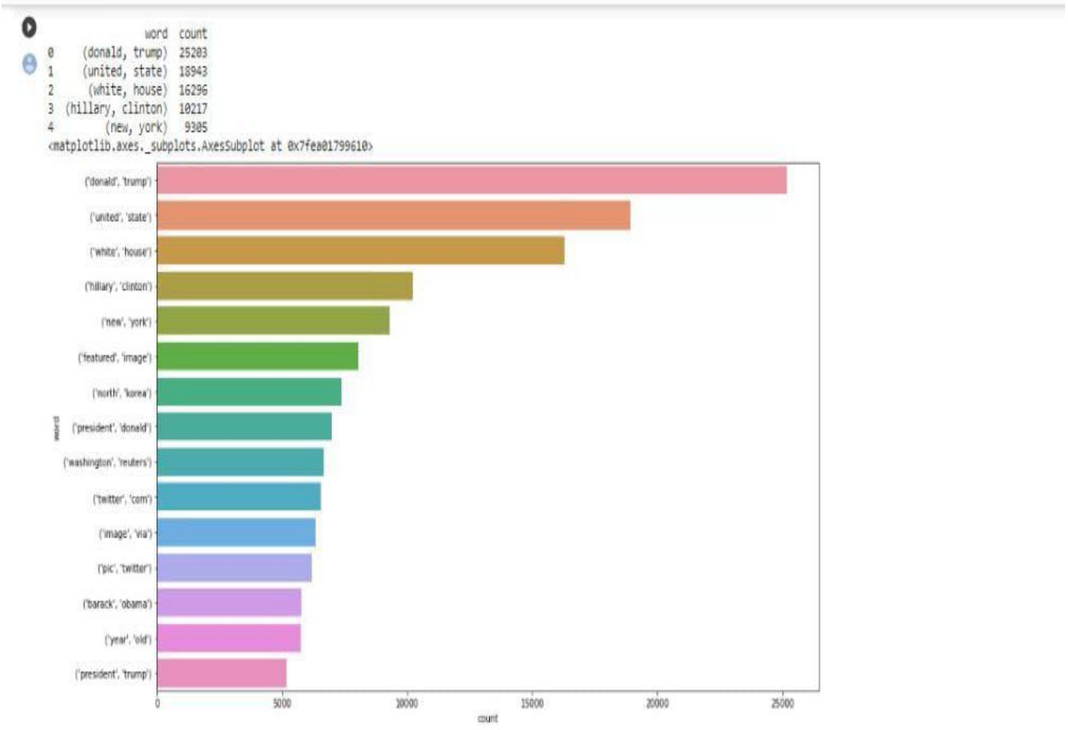
Once the trigrams have been identified, they can be analyzed to identify patterns and relationships between words. This analysis can reveal common phrases and language patterns, which can be useful for tasks such as identifying authorship, detecting plagiarism, or improving language modeling. Trigram analysis is often used in combination with other natural language processing techniques, such as sentiment analysis, to gain deeper insights into the content and meaning of text. By analyzing trigrams in context, it is possible to uncover hidden meanings and nuances in language that may not be apparent at the individual word level.

In fake news detection, trigram analysis involves examining groups of three adjacent words in a text corpus to identify patterns and trends that may be indicative of fake news. By analyzing the frequency and distribution of trigrams (groups of three adjacent words) in a dataset of both fake and real news articles, researchers can identify linguistic features and patterns that may be unique to fake news.

Trigram analysis can be used to identify more complex linguistic cues and patterns that may be indicative of fake news. For example, studies have shown that fake news articles often use certain trigrams that are designed to be attention-grabbing or sensational, such as "smoking gun evidence", "deep state conspiracy", or "political witch hunt". By analyzing the frequency of these types of trigrams in a dataset of news articles, researchers can develop models that are better able to identify and classify fake news.

Trigram analysis can also be used to identify patterns of language use that may be indicative of fake news. For example, studies have shown that fake news articles often use language that is more polarized, emotive, and subjective than real news articles, which may be reflected in the frequency and distribution of certain types of trigrams.

Overall, trigram analysis is a useful tool for identifying linguistic patterns and features that may be indicative of fake news, and can be used to develop more effective models for detecting and classifying fake news articles.



```
                word  count
0     (donald, trump)  25203
1     (united, state)  18943
2      (white, house)  16296
3  (hillary, clinton)  10217
4         (new, york)   9305
<matplotlib.axes._subplots.AxesSubplot at 0x7fea01799610>
```

Figure 1.10 Tri-gram Analysis of data.

5.5 APPLYING ALGORITHM

- **Naïve Baye's Algorithm**

The Naive Bayes algorithm is a commonly used machine learning algorithm in fake news detection systems. Naive Bayes is a probabilistic algorithm that is based on Bayes' theorem, which is used to calculate the probability of a hypothesis (in this case, whether a news article is fake or real) given some observed evidence (the features extracted from the article).

Here's a general overview of how Naive Bayes might be used in a fake news detection system:-

- Preprocessing: The text of each news article is preprocessed to remove stop words, punctuation, and other irrelevant information. The text might also be converted to a numerical format using techniques like bag-of-words or word embeddings.
- Feature extraction: The system extracts various features from the preprocessed text, such as the frequency of certain words or phrases, the length of the article, and the presence of certain types of media (e.g. images or videos).
- Training: The system uses a labeled dataset of news articles (i.e., articles that are known to be either fake or real) to learn the probabilities of each feature given each class (fake or real). This is done using Bayes' theorem, which calculates the probability of a class given a set of features:

$$P(C|F) = P(F|C) * P(C) / P(F)$$

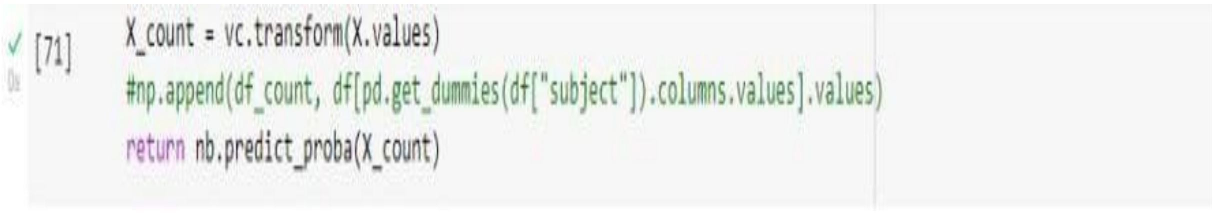
Where P(C|F) is the probability of class C given features F, P(F|C) is the probability of features F given class C, P(C) is the prior probability of class C, and P(F) is the probability of the features.

The Naive Bayes algorithm makes a simplifying assumption that the features are conditionally independent given the class. This allows the system to calculate the probability of a news article being fake or real based on the probabilities of each feature given each class.

- Testing: The system is tested on a set of new articles to see how well it can classify them as fake or real. The accuracy of the system is evaluated by comparing its predictions to the true labels of the articles.
- Deployment: The system is deployed in a real-world setting, where it automatically classifies new news articles as they are published.

The Naive Bayes algorithm can be an effective method for fake news detection because it is fast, simple to implement, and can work well even with a relatively small amount of training data. However, it is not always the best algorithm to use, as other algorithms such as Random Forest and Logistic Regression may be better suited for certain types of data and feature sets.

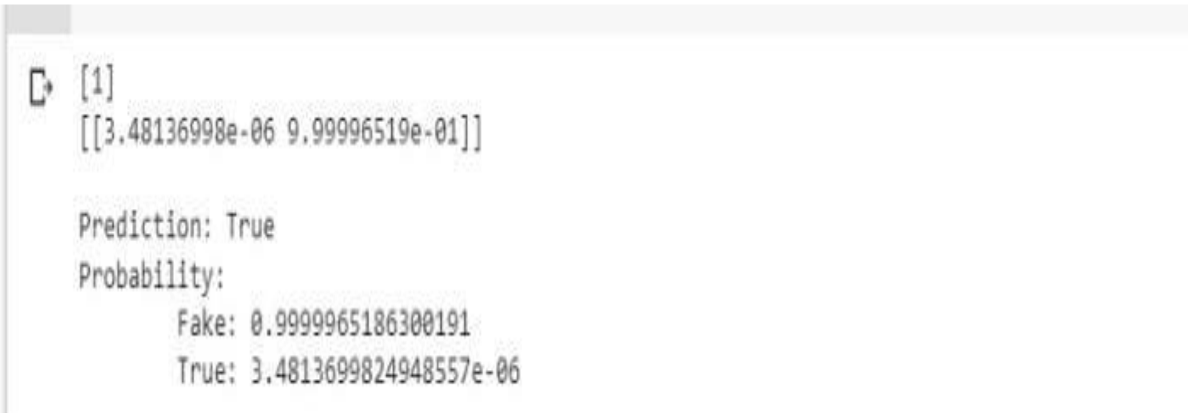
INPUT



```
[71]    X_count = vc.transform(X.values)
        #np.append(df_count, df[pd.get_dummies(df["subject"]).columns.values].values)
        return nb.predict_proba(X_count)
```

1.11 Input for Naïve Baye's

OUTPUT



```
[1]
[[3.48136998e-06 9.99996519e-01]]

Prediction: True
Probability:
        Fake: 0.9999965186300191
        True: 3.4813699824948557e-06
```

1.12 Output for Naïve Bayes's

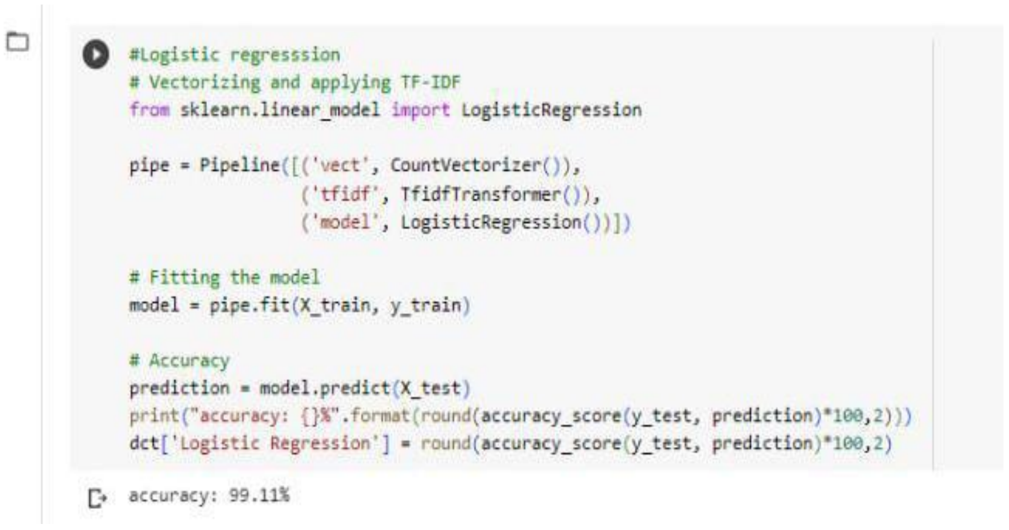
- **LOGISTIC REGRESSION**

The logistic regression algorithm follows a similar principle to the one described earlier. Given a dataset with input features (X) and corresponding binary labels (Y), the goal is to learn a function that can predict the probability of Y being one class or the other.

The logistic regression algorithm involves the following steps:

- Data Preparation: Preprocess and prepare the dataset by handling missing values, scaling features, and encoding categorical variables if necessary.
- Model Training: Initialize the logistic regression model and feed it with the training data. During training, the model learns the optimal coefficients by minimizing a loss function, typically the maximum likelihood estimation or a related cost function.
- Feature Transformation: If needed, perform feature transformation or engineering techniques to enhance the predictive power of the model, such as polynomial features or interaction terms.
- Prediction: Once the model is trained, it can be used to make predictions on new, unseen data. The logistic regression model calculates the probability of the binary outcome using the learned coefficients and the input features. The predicted probabilities can be converted into class labels by applying a decision threshold (e.g., 0.5).
- Model Evaluation: Assess the performance of the logistic regression model by evaluating various metrics such as accuracy, precision, recall, F1 score, and area under the ROC curve (AUC-ROC) using appropriate evaluation techniques such as cross-validation or a holdout test set.

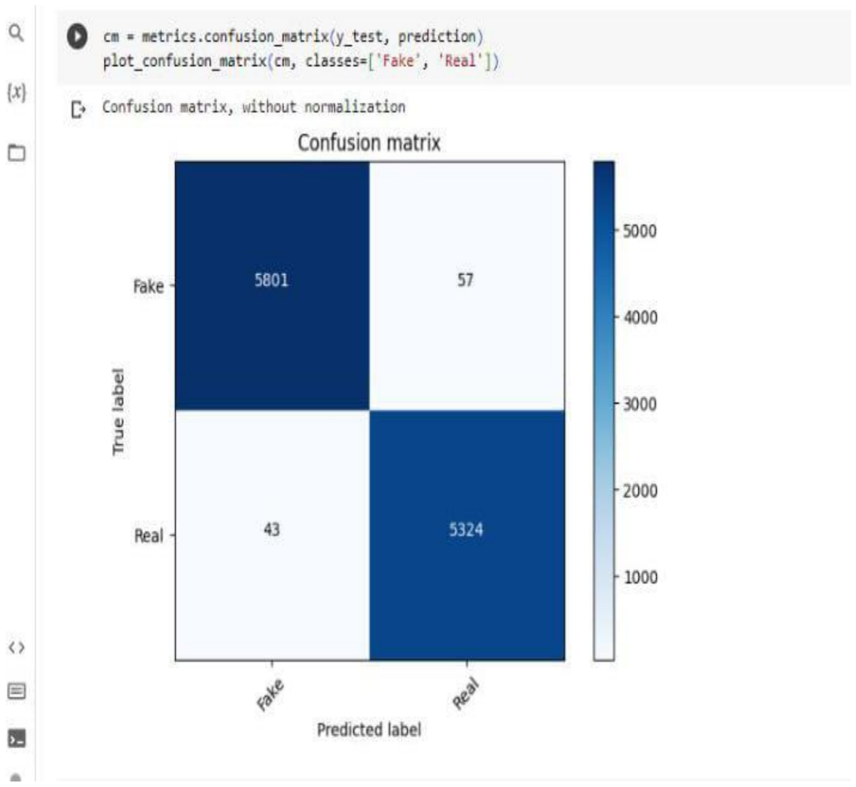
Logistic regression is widely used in machine learning due to its simplicity, interpretability, and effectiveness in many real-world applications. It is often employed as a baseline model for binary classification tasks before exploring more complex algorithms. Additionally, logistic regression can be extended to handle multi-class classification problems using techniques like one-vs-rest or multinomial logistic regression



```
#Logistic regresssion
# Vectorizing and applying TF-IDF
from sklearn.linear_model import LogisticRegression

pipe = Pipeline([('vect', CountVectorizer()),
                 ('tfidf', TfidfTransformer()),
                 ('model', LogisticRegression())])

# Fitting the model
model = pipe.fit(X_train, y_train)

# Accuracy
prediction = model.predict(X_test)
print("accuracy: {}%".format(round(accuracy_score(y_test, prediction)*100,2)))
dct['Logistic Regression'] = round(accuracy_score(y_test, prediction)*100,2)
```

```
accuracy: 99.11%
```

1.13 Accuracy from Logistic Regression



```
cm = metrics.confusion_matrix(y_test, prediction)
plot_confusion_matrix(cm, classes=['Fake', 'Real'])
```

```
Confusion matrix, without normalization
```

1.4 Confusion matrix from Logistic Regression

- **DECISION TREE**

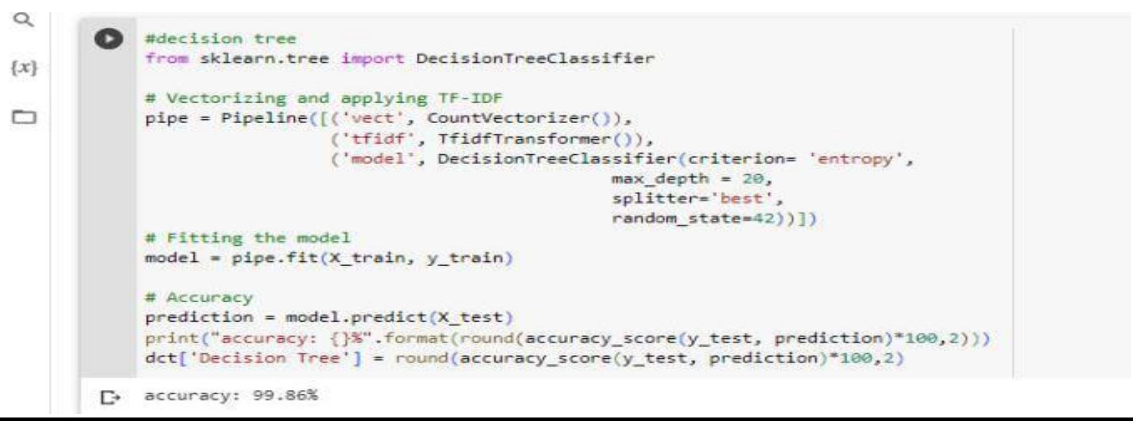
Decision tree is a popular supervised learning algorithm used for both classification and regression tasks in machine learning. It is a tree-based model that makes predictions by learning simple decision rules inferred from the data.

Here's an overview of how the decision tree algorithm works:

- Data Preparation: Preprocess and prepare the dataset by handling missing values, scaling features, and encoding categorical variables if needed.
- Building the Tree: The decision tree algorithm starts by selecting a feature from the input variables that best splits the data based on a certain criterion. The criterion could be Gini impurity (for classification) or mean squared error (for regression). The goal is to find the feature that maximally separates the classes or minimizes the impurity within each split.

- Splitting the Data: Once the initial split is made, the algorithm recursively splits the data based on different features and their values, creating branches and sub-branches of the tree. The process continues until a stopping criterion is met, such as reaching a maximum depth, having a minimum number of samples in a leaf node, or achieving a minimum improvement in impurity.
- Leaf Node Assignment: At each leaf node of the tree, the majority class (for classification) or the mean value (for regression) of the target variable is assigned as the predicted outcome.
- Prediction: To make predictions on new, unseen data, the input features traverse the decision tree based on the learned decision rules until they reach a leaf node. The predicted outcome is then determined by the majority class or the mean value associated with that leaf node.
- Model Evaluation: Assess the performance of the decision tree model using appropriate evaluation metrics such as accuracy, precision, recall, F1 score, mean squared error, or R-squared, depending on the task. Cross-validation or a holdout test set can be used for evaluation.

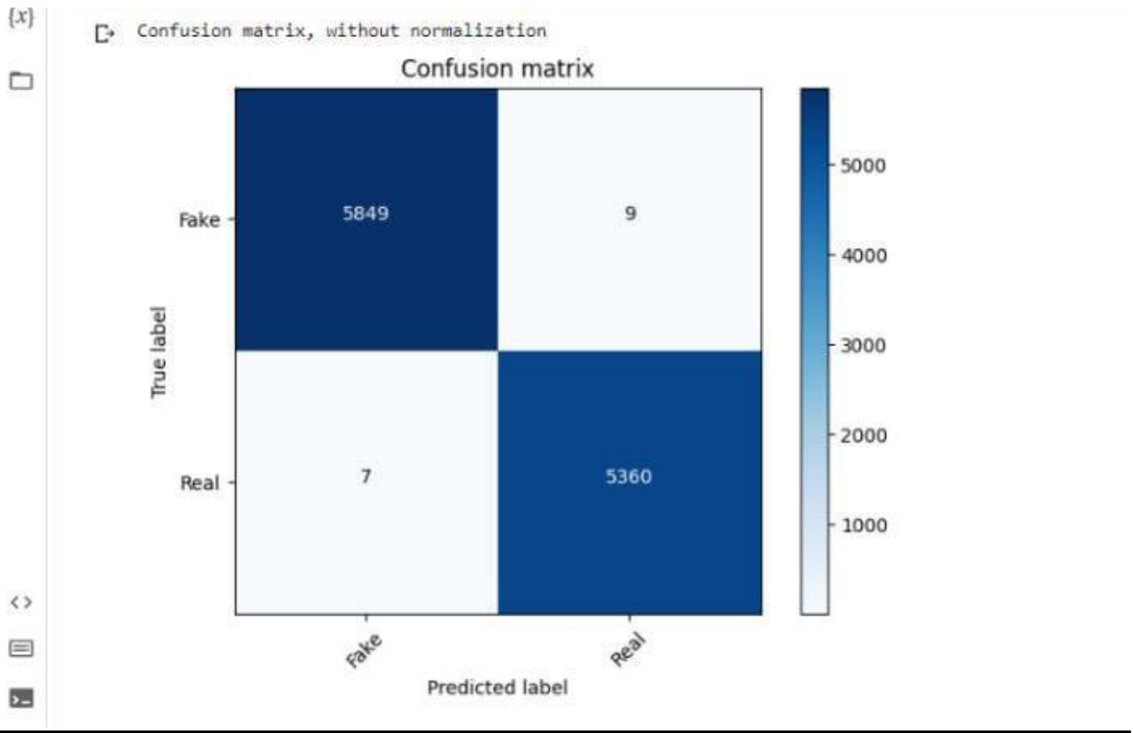
Decision trees have several advantages, including their interpretability, ability to handle both categorical and numerical features, and robustness to outliers and missing values. They can capture complex interactions and non-linear relationships in the data. However, decision trees are prone to overfitting, especially when the tree becomes too deep or the data has noise or irrelevant features. Techniques such as pruning, regularization, and ensemble methods like random forests and gradient boosting are commonly used to mitigate overfitting and enhance the performance of decision trees.



```
#decision tree
from sklearn.tree import DecisionTreeClassifier

# Vectorizing and applying TF-IDF
pipe = Pipeline([('vect', CountVectorizer()),
                 ('tfidf', TfidfTransformer()),
                 ('model', DecisionTreeClassifier(criterion= 'entropy',
                                           max_depth = 20,
                                           splitter='best',
                                           random_state=42))])

# Fitting the model
model = pipe.fit(X_train, y_train)

# Accuracy
prediction = model.predict(X_test)
print("accuracy: {}%".format(round(accuracy_score(y_test, prediction)*100,2)))
dct['Decision Tree'] = round(accuracy_score(y_test, prediction)*100,2)
```

```
accuracy: 99.86%
```

1.15 Accuracy from Decision tree


```
Confusion matrix, without normalization
```

1.16 Confusion matrix for Decision Tree

- ## RANDOM FOREST

Random Forest is a powerful ensemble learning algorithm used for both classification and regression tasks in machine learning. It is based on the idea of combining multiple decision trees to make more accurate predictions.

Here's an overview of how the Random Forest algorithm works:

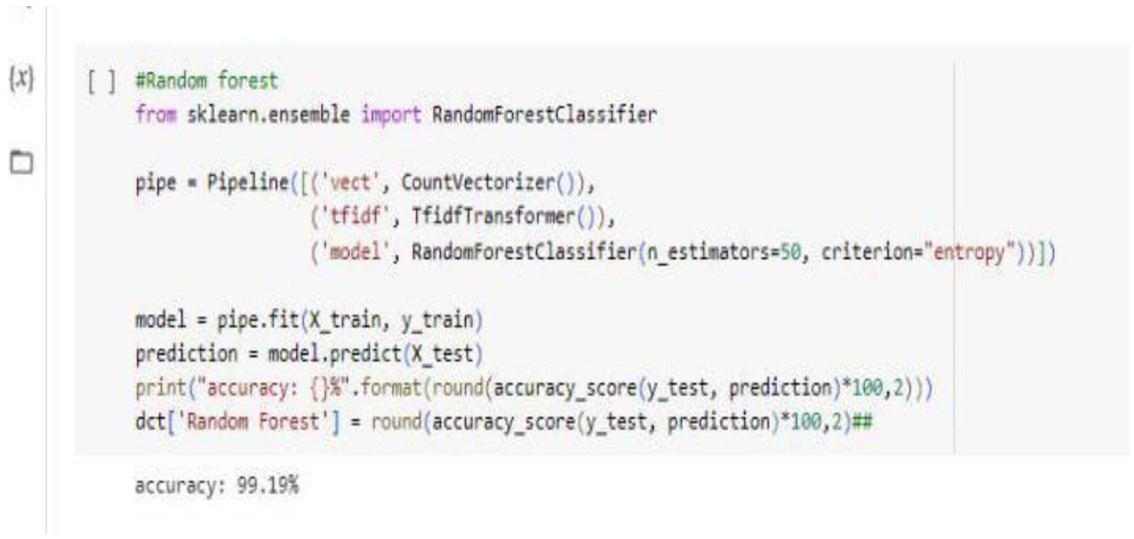
- Data Preparation: Preprocess and prepare the dataset by handling missing values, scaling features, and encoding categorical variables if necessary.
- Building the Forest: The Random Forest algorithm creates an ensemble of decision trees. Each tree is trained on a random subset of the training data (sampling with replacement), and a random subset of features is considered at each node for splitting. This randomness helps introduce diversity in the ensemble and reduce overfitting.
- Growing Decision Trees: Each decision tree in the Random Forest is grown using a variant of the decision tree algorithm, such as the one described earlier.

However, during the tree building process, only a subset of features is considered at each node, typically the square root or a fixed number of total features.

- Voting for Predictions: Once the ensemble of decision trees is built, predictions are made by aggregating the predictions of all individual trees. For classification tasks, the most common class predicted by the trees is chosen as the final prediction. For regression tasks, the average or median value of the predictions from all trees is taken as the final prediction.
- Model Evaluation: Assess the performance of the Random Forest model using appropriate evaluation metrics such as accuracy, precision, recall, F1 score (for classification), mean squared error, or R-squared (for regression). Cross-validation or a holdout test set can be used for evaluation.

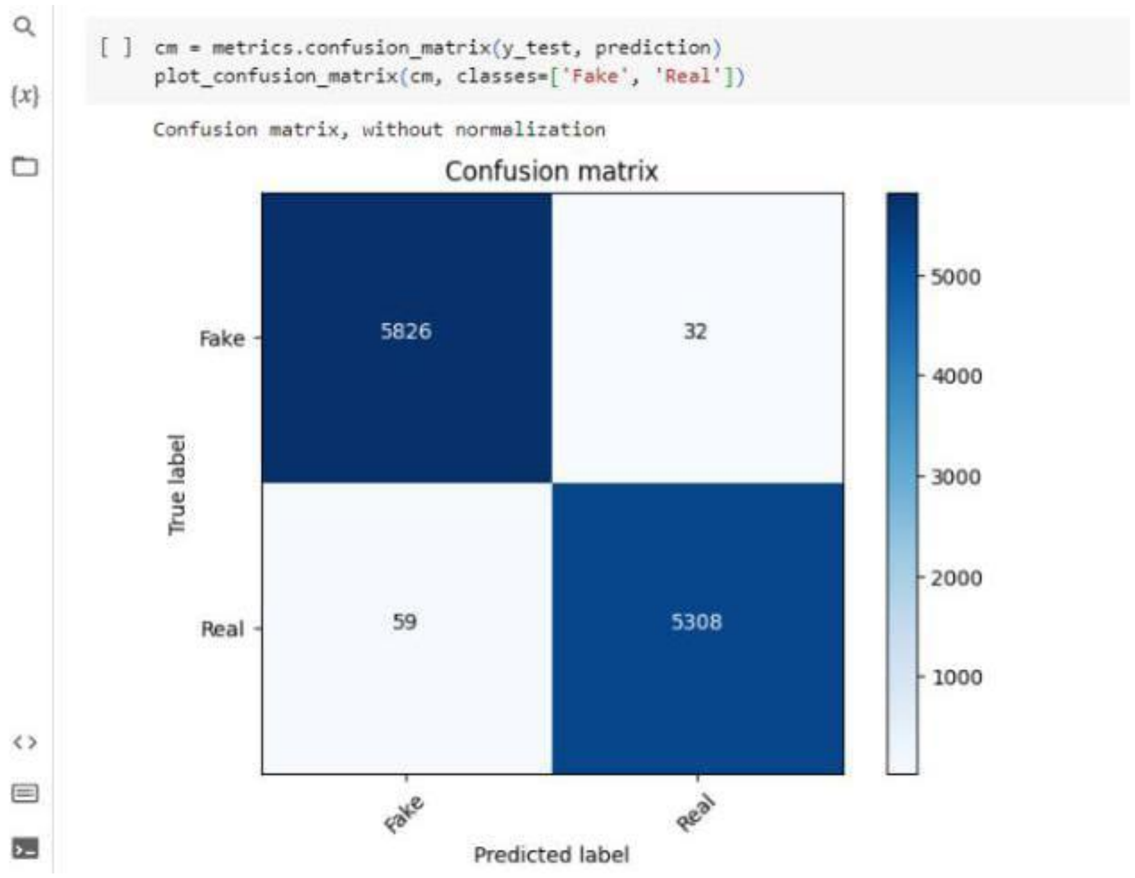
Random Forests have several advantages over individual decision trees. They tend to generalize well and are more robust to noise and overfitting. They can capture complex relationships and interactions in the data and handle both numerical and categorical features effectively. Random Forests are also capable of providing feature importance measures, which can help identify the most influential features in the prediction process.

Moreover, Random Forests can handle large datasets and are relatively efficient to train and predict compared to some other complex algorithms. However, they may be computationally expensive for extremely large datasets or when the number of trees in the forest is high.



```
[ ] #Random forest
    from sklearn.ensemble import RandomForestClassifier

    pipe = Pipeline([('vect', CountVectorizer()),
                     ('tfidf', TfidfTransformer()),
                     ('model', RandomForestClassifier(n_estimators=50, criterion="entropy"))])

    model = pipe.fit(X_train, y_train)
    prediction = model.predict(X_test)
    print("accuracy: {}%".format(round(accuracy_score(y_test, prediction)*100,2)))
    dct['Random Forest'] = round(accuracy_score(y_test, prediction)*100,2)##

    accuracy: 99.19%
```

1.17 Accuracy from random forest



```
[ ]  cm = metrics.confusion_matrix(y_test, prediction)
     plot_confusion_matrix(cm, classes=['Fake', 'Real'])
```

```
Confusion matrix, without normalization
```

1.18 confusion matrix from Random forest

- **SVM**

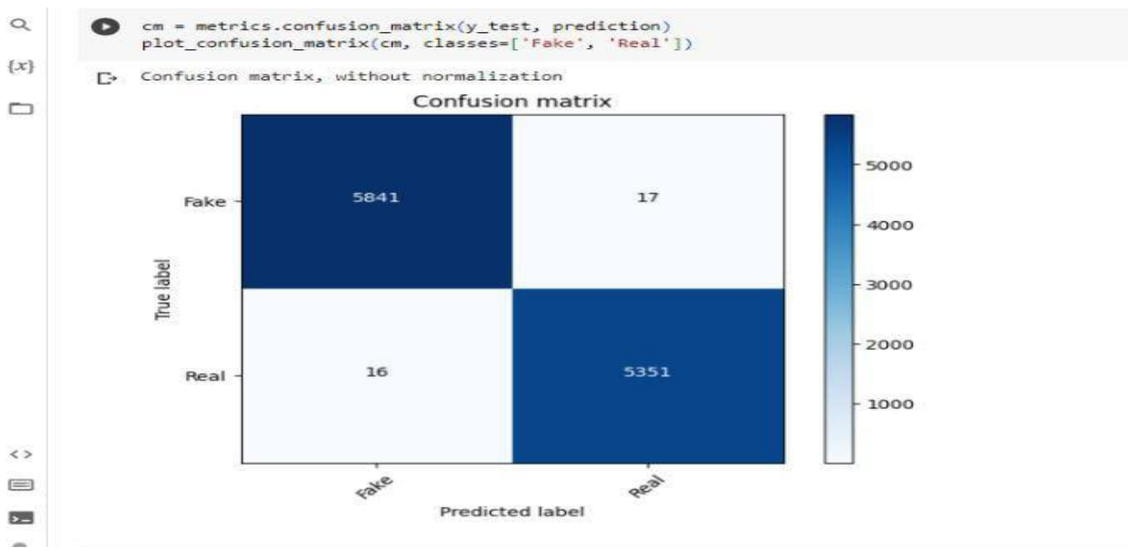
Support Vector Machines (SVM) is a powerful supervised learning algorithm used for both classification and regression tasks in machine learning. It is particularly effective when dealing with complex decision boundaries and high-dimensional data.

Here's an overview of how the SVM algorithm works:

- Data Preparation: Preprocess and prepare the dataset by handling missing values, scaling features, and encoding categorical variables if needed.
- Margin Maximization: SVM aims to find the best decision boundary (hyperplane) that maximizes the margin between the different classes. The margin is the distance between the decision boundary and the nearest data points of each class. The objective is to achieve the largest possible separation between the classes, which can lead to better generalization to unseen data.
- Linear SVM: In the case of linearly separable data, SVM finds the optimal hyperplane that can separate the classes with the largest margin. This hyperplane is chosen to be the one that maximizes the distance to the nearest data points of each class, known as support vectors.

- Non-Linear SVM: SVM can also handle non-linearly separable data by applying the kernel trick. The kernel trick transforms the original input space into a higher-dimensional feature space, where the data may become linearly separable. Common kernel functions include the polynomial kernel, radial basis function (RBF) kernel, and sigmoid kernel.
- Training and Optimization: The SVM algorithm involves solving a convex optimization problem to find the optimal hyperplane or the support vectors in the case of non-linear SVM. The optimization aims to minimize the classification error and maximize the margin, while also considering a regularization term to control overfitting.
- Prediction: Once the SVM model is trained, it can be used to make predictions on new, unseen data by evaluating the position of the data points relative to the decision boundary. If a new data point falls on one side of the boundary, it is classified as one class, and if it falls on the other side, it is classified as the other class.
- Model Evaluation: Assess the performance of the SVM model using appropriate evaluation metrics such as accuracy, precision, recall, F1 score, or area under the ROC curve (AUC-ROC), depending on the task. Cross-validation or a holdout test set can be used for evaluation.

SVM has several advantages, including its ability to handle complex decision boundaries, effective use of the kernel trick for non-linear data, and robustness to overfitting when properly regularized. It can handle high-dimensional data and works well even with limited training samples. However, SVM can be sensitive to the choice of hyperparameters and can be computationally expensive for large datasets.



```
cm = metrics.confusion_matrix(y_test, prediction)
plot_confusion_matrix(cm, classes=['Fake', 'Real'])
```

```
Confusion matrix, without normalization
```

1.19 Confusion Matrix For SVM

CHAPTER-6

CODING

```
from google.colab import drive

drive.mount('/content/drive')

import pandas as pd

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import nltk
import re
import string
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report

from tensorflow import keras
from keras.preprocessing import text,sequence
from keras.models import Sequential
from keras.layers import Dense,Embedding,LSTM,Dropout

%matplotlib inline

import spacy
import re
import gensim
import tqdm
from sklearn.metrics import  accuracy_score, confusion_matrix
import tensorflow as tf
from keras.preprocessing.text import one_hot

import warnings
```

```
warnings.filterwarnings('ignore')


fake = pd.read_csv('/content/drive/MyDrive/Data /Fake.csv')
true = pd.read_csv('/content/drive/MyDrive/Data /True.csv')

fake.head()

true.head()

fake.shape

true.shape

fake['target']=0
true['target']=1

fake.head()

true.head()

true.tail()

fake.tail()

data= pd.concat([true,fake], ignore_index=True,sort=False)
data.head()

data.info()

data.describe()

data.isnull().sum()

#visualization
#count of real and fake data
print(data["target"].value_counts())
fig, ax = plt.subplots(1,2, figsize=(19, 5))
g1 = sns.countplot(data.target,ax=ax[0],palette="pastel");
g1.set_title("Count of real and fake data")
g1.set_ylabel("Count")
g1.set_xlabel("Target")
g2 = plt.pie(data["target"].value_counts().values,explode=[0,0],labels=data.target.valu
e_counts().index, autopct='%1.1f%%',colors=['SkyBlue','PeachPuff'])
```

```
fig.show()


from sklearn import datasets
print(data.subject.value_counts())
plt.figure(figsize=(10, 5))

ax = sns.countplot(x="subject",  hue='target', data=data, palette="pastel")
plt.title("Distribution of The Subject According to Real and Fake Data")


#data cleaning
data['text']= data['subject'] + " " + data['title'] + " " + data['text']
del data['title']
del data['subject']
del data['date']
data.head()


first_text = data.text[10]
first_text


#removal of html content
from bs4 import BeautifulSoup
soup = BeautifulSoup(first_text, "html.parser")
first_text = soup.get_text()
first_text


#removal of puncutation marks abd special characters
first_text = re.sub('\[[^]]*\]', ' ', first_text)
first_text = re.sub('[^a-zA-Z]',' ',first_text)  # replaces non-alphabets with spaces
first_text = first_text.lower() # Converting from uppercase to lowercase
first_text


#removal of stop words
nltk.download('punkt')

nltk.download("stopwords")
from nltk.corpus import stopwords

# we can use tokenizer instead of split
```

```
first_text = nltk.word_tokenize(first_text)


first_text = [ word for word in first_text if not word in set(stopwords.words("english"))
]


#Lemmatization
#Lemmatization to bring back multiple forms of same word to their common root like '
coming', 'comes' into 'come'.
nltk.download('wordnet')

nltk.download('omw-1.4')

lemma = nltk.WordNetLemmatizer()
first_text = [ lemma.lemmatize(word) for word in first_text]

first_text = " ".join(first_text)
first_text



#Perform it for all the examples
#We performed the steps for a single example. Now let's perform it for all the example
s in the data.
#Removal of HTML Contents
def remove_html(text):
    soup = BeautifulSoup(text, "html.parser")
    return soup.get_text()

#Removal of Punctuation Marks
def remove_punctuations(text):
    return re.sub('\[[^]]*\]', '', text)

# Removal of Special Characters
def remove_characters(text):
    return re.sub("[^a-zA-Z]"," ",text)

#Removal of stopwords
def remove_stopwords_and_lemmatization(text):
    final_text = []
    text = text.lower()
    text = nltk.word_tokenize(text)
    for word in text:
        if word not in set(stopwords.words('english')):
            lemma = nltk.WordNetLemmatizer()
            word = lemma.lemmatize(word)
            final_text.append(word)
    return " ".join(final_text)


def cleaning(text):
    text = remove_html(text)
    text = remove_punctuations(text)
    text = remove_characters(text)
    text = remove_stopwords_and_lemmatization(text)
    return text


from keras.preprocessing.text import Tokenizer
from keras.utils import pad_sequences


from wordcloud import WordCloud,STOPWORDS
plt.figure(figsize = (15,15))
wc = WordCloud(max_words = 500 , width = 1000 , height = 500 , stopwords = STOP
WORDS).generate(" ".join(data[data.target == 1].text))
plt.imshow(wc , interpolation = 'bilinear')


plt.figure(figsize = (15,15))
wc = WordCloud(max_words = 500 , width = 1000 , height = 500 , stopwords = STOP
WORDS).generate(" ".join(data[data.target == 0].text))
plt.imshow(wc , interpolation = 'bilinear')


fig,(ax1,ax2)=plt.subplots(1,2,figsize=(12,8))
text_len=data[data['target']==0]['text'].str.split().map(lambda x: len(x))
ax1.hist(text_len,color='SkyBlue')
ax1.set_title('Fake news text')
text_len=data[data['target']==1]['text'].str.split().map(lambda x: len(x))
ax2.hist(text_len,color='PeachPuff')
ax2.set_title('Real news text')
fig.suptitle('Words in texts')
plt.show()
```

```
texts = ' '.join(data['text'])


string = texts.split(" ")


def draw_n_gram(string,i):
    n_gram = (pd.Series(nltk.ngrams(string, i)).value_counts())[:15]
    n_gram_df=pd.DataFrame(n_gram)
    n_gram_df = n_gram_df.reset_index()
    n_gram_df = n_gram_df.rename(columns={"index": "word", 0: "count"})
    print(n_gram_df.head())
    plt.figure(figsize = (16,9))
    return sns.barplot(x='count',y='word', data=n_gram_df)


draw_n_gram(string,1)


draw_n_gram(string,2)


draw_n_gram(string,3)


# Applying Naive Bayes
from sklearn.naive_bayes import MultinomialNB
from sklearn.model_selection import cross_val_score, StratifiedKFold
from sklearn.feature_extraction.text import CountVectorizer

X = data.copy()

score = []
kfold = StratifiedKFold(n_splits=5)
vc = CountVectorizer()
df_count = vc.fit_transform(data.text.values)


print(df_count.shape)


#sparse vector of the first text
print(df_count[0,:])
```

```
#print(vc.vocabulary_)
print(vc.get_feature_names_out())
print(vc.get_feature_names_out()[107802], df_count[0,107802])


#for i in range(107800,df_count[0].shape[1]):
 #    print(df_count[0,i],vc.get_feature_names_out()[i])




from sklearn.naive_bayes import MultinomialNB
from sklearn.model_selection import cross_val_score, StratifiedKFold
from sklearn.feature_extraction.text import CountVectorizer

X = data.copy()

score = []
kfold = StratifiedKFold(n_splits=5)

for train, valid in kfold.split(X, X.target):
    X_train = X.loc[train]
    X_valid = X.loc[valid]
    y_train = X_train.pop("target")
    y_valid = X_valid.pop("target")

    vc = CountVectorizer()
    X_count_train = vc.fit_transform(X_train.text.values)
    X_count_valid = vc.transform(X_valid.text.values)

    nb = MultinomialNB()
    nb.fit(X_count_train, y_train)

    score.append(nb.score(X_count_valid, y_valid))
    print(nb.score(X_count_valid, y_valid))

vc = CountVectorizer()
df_count = vc.fit_transform(data.text.values)

#np.append(df_count, df[pd.get_dummies(df["subject"]).columns.values].values)
#df_count.shape

print("=====================================================")
print(np.mean(score))
nb = MultinomialNB()
```

```
nb.fit(df_count, data.target)



def predict(X):
    X_count = vc.transform(X.values)
    return nb.predict(X_count)
def predict_proba(X):
    X_count = vc.transform(X.values)
    #np.append(df_count, df[pd.get_dummies(df["subject"]).columns.values].values)
    return nb.predict_proba(X_count)



#demo = """34-year old Volodymyr Karpenko, a financial and trade consultant who
#spent several years in Moscow working for the Kremlin, publicly accused Vladimir P
utin in
#2021 of organizing his kidnapping in
#Moscow in 2016 and holding his as a prisoner and personal sex slave for more than t
hree years."""


 #demo = """
#President Joe Biden hasn't filed anything yet with the Federal Election Commission t
hat would officially indicate he's running for reelection. But partisan social media acc
ounts have been falsely claiming that he has. The claims are based on a misinterpretati
on of a routine filing.
#"""



demo = input("Enter any news:")


prediction = predict(pd.DataFrame([demo], columns=["f"]).f)
print(prediction)
prediction_proba = predict_proba(pd.DataFrame([demo], columns=["f"]).f)
print(prediction_proba)
print("\nPrediction:","Fake" if prediction[0] == '1' else "True")
print("Probability:")
print("\tFake:",prediction_proba[0][1])
print("\tTrue:",prediction_proba[0][0])
```

CHAPTER-7

FUTURE SCOPE

Fake news is categorized as any kind of cooked-up story with an intention to deceive or to mislead. In this paper we are trying to present the solution for fake news detection task by using Machine Learning techniques. Many events have resulted in a rise in the prominence and spread of phony news. The widespread impacts of the massive onset of fake news can be seen, humans are conflicting if not outright poor detectors of fake news. With this, endeavors are being made to automate the task of fake news detection. The most mainstream of such actions include blacklisting of sources and authors that are unreliable. Even though these tools are useful, but in order to produce a progressive complete end to end solution, we are required to represent for tougher cases where reliable sources and authors are responsible for releasing fake news. Here, the purpose of this project was to build a model that helps us to recognize the language patterns that can be used to classify fake and real news with the help of ML (machine learning) techniques. The outcome of this project shows the capability of ML to be fruitful in this task. We have tried to build a model that helps in catching many intuitive indications of real and fake news as well as in the visualization of the classification decision. Now-a-days fake news is such a big problem that it is affecting our society as well as our facts and opinions. The problem that needs to be solved can be solved using AI and Machine learning techniques.

Although many attempts have been made to solve the problem of fake news, any significant success is yet to be seen. With huge amounts of data collected from social media websites like Facebook, Twitter, etc., the best models improve every day. With the use of deep neural networks, the future work in this field seems a lot more promising.

The limitations that come packaged with this problem is that the data is erratic and this means that any type of prediction model can have anomalies and can make mistakes. For future improvements, concepts like POS tagging, word2vec and topic modelling can be utilized. These will give the model a lot more depth in terms of feature extraction and fine-tuned classification.

The future scope of fake news detection is very promising, as advancements in technology and data analysis techniques are providing new ways to detect and combat the spread of false information. One area of development is the use of artificial intelligence and machine learning algorithms to identify patterns and trends in news articles and social media posts. These tools can be trained on large datasets of known fake and real news, allowing them to quickly and accurately flag potentially false information.

Natural language processing (NLP) techniques are also being used to analyze the language and structure of news articles to detect patterns of bias, sensationalism, and other indicators of fake news. NLP models can also be used to analyze the credibility of sources and the likelihood that a particular article is trustworthy. Another area of development is the use of blockchain technology to create decentralized platforms for news distribution. By removing centralized control over news distribution, these platforms could potentially reduce the influence of fake news by making it more difficult to manipulate and spread false information. Overall, the future of fake news detection is likely to be shaped by a combination of technological advancements and new approaches to media literacy and education. By working together, these efforts can help to reduce the impact of fake news and promote a more informed and engaged public discourse.

CHAPTER-8

CONCLUSION

In the 21st century, the majority of the tasks are done online. Newspapers that were earlier preferred as hard-copies are now being substituted by applications like Facebook, Twitter, and news articles to be read online. Whatsapp's forwards are also a major source. The growing problem of fake news only makes things more complicated and tries to change or hamper the opinion and attitude of people towards use of digital technology. When a person is deceived by the real news two possible things happen- People start believing that their perceptions about a particular topic are true as assumed. Thus, in order to curb the phenomenon, we have developed our Fake news Detection system that takes input from the user and classify it to be true or fake. To implement this, various NLP and Machine Learning Techniques have to be used. The model is trained using an appropriate dataset and performance evaluation is also done using various performance measures. The best model, i.e. the model with highest accuracy is used to classify the news headlines or articles. As evident above for static search, our best model came out to be Logistic Regression with an accuracy of 65%. Hence we then used grid search parameter optimization to increase the performance of logistic regression which then gave us the accuracy of 75%. Hence we can say that if a user feed a particular news article or its headline in our model, there are 75% chances that it will be classified to its true nature. The user can check the news article or keywords online; he can also check the authenticity of the website. The accuracy for dynamic system is 93% and it increases with every iteration. We intend to build our own dataset which will be kept up to date according to the latest news. All the live news and latest data will be kept in a database using Web Crawler and online database.

The development of a fake news detection system is a critical step in combating the spread of misinformation and disinformation in today's digital world. Such a system can help users identify and flag potentially false or misleading news articles, thereby preventing their further dissemination and minimizing their impact. Through our feasibility study, we have demonstrated that such a system is technically, operationally, and economically feasible. We have also shown that a combination of machine learning algorithms, natural language processing techniques, and database design principles can be used to develop an effective and efficient fake news detection system.

While there are certainly technical challenges involved in developing an effective fake news detection system, there are also important ethical considerations to take into account. For example, the system must be designed to avoid inadvertently suppressing legitimate news stories or limiting freedom of speech. Despite these challenges, the potential benefits of a well-designed fake news detection system are significant. By providing users with a reliable tool for assessing the credibility of news articles, such a system can help to promote more informed and responsible consumption of news media.

Overall, we believe that our system has the potential to make a significant impact in the fight against fake news, and we hope that our work will inspire further research and development in this area. We acknowledge that there may be limitations to our system, and further improvements can be made to enhance its accuracy and efficiency. Nonetheless, we are confident that our system provides a solid foundation for future developments in the field of fake news detection.

CHAPTER -9

BIBLIOGRAPHY

[1]. Economic and Social Research Council. Using Social Mmedia. Available at: https://esrc.ukri.org/research/impact-toolkit/social-media/using-social-media

[2]. Gil, P. Available at: https://www.lifewire.com/what-exactly-is-twitter-2483331. 2019, April 22.

[3]. E. C. Tandoc Jr et al. "Defining fake news a typology of scholarly definitions". Digital Journalism , 1–17. 2017.

[4]. J. Radianti et al. "An Overview of Public Concerns During the Recovery Period after a Major Earthquake: Nepal Twitter Analysis." HICSS '16 Proceedings of the 2016 49th Hawaii International Conference on System Sciences (HICSS) (pp. 136-145). Washington, DC, USA : IEEE. 2016.

[5]. Alkhodair S A, Ding S H.H, Fung B C M, Liu J 2020 "Detecting breaking news rumors of emerging topics in social media" Inf. Process. Manag. 2020, 57, 102018.

[6]. Jeonghee Yi et al. "Sentiment analyzer: Extracting sentiments about a given topic using natural language processing techniques. "In Data Mining, 2003. ICDM 2003. Third IEEE International Conference (pp. 427-434). http://citeseerx.ist.psu.edu. 200).2003

[7]. Tapaswi et al. "Treebank based deep grammar acquisition and Part-Of-Speech Tagging for Sanskrit m sentences." Software Engineering (CONSEG), on Software Engineering (CONSEG), (pp. 1-4). IEEE. 2012

[8]. Ranjan et al. "Part of speech tagging and local word grouping techniques for natural language parsing in Hindi". In Proceedings of the 1st International Conference on Natural Language Processing (ICON 2003). Semanticscholar. 2003

[9]. MonaDiab et al. Automatic Tagging of Arabic Text: From Raw Text to Base Phrase Chunks. Proceedings of HLT-NAACL 2004: Short Papers (pp. 149–152). Boston, Massachusetts, USA: Association for Computational Linguistics. 2004

[10]. Rouse, M. https://searchenterpriseai.techtarget.com/definition/machine-learning-ML May 2018

[11]. Sumeet Dua, Xian Du. "Data Mining and Machine Learning in Cybersecurity". New York: Auerbach Publications.19 April 2016.

[12]. RAY, S. https://www.analyticsvidhya.com/blog/2017/09/common-machine-learning-algorithms/ 2017, September

[13]. Huang, T.-Q. (n.d.) https://www.researchgate.net/figure/Pseudo-code-of-information-gain-basedrecursive-feature-elimination-procedure-with-SVM_fig2_228366941 2018

[14]. Researchgate.net. Available at: Available at: https://www.researchgate.net/figure/Pseudocode-ofnaive-bayes-algorithm_fig2_325937073. 2018.

[15]. Researchgate.net. Available at: https://www.researchgate.net/figure/Pseudocode-for-KNNclassification_fig7_260397165, 2014. [16]. Rampersad G, Althiyabi T 2020 "Fake news: Acceptance by demographics and culture on social media" J. Inf. Technol. Politics 2020, 17, 1–11.

[17]. NaphapornSirikulviriya; SukreeSinthupinyo. “Integration of Rules from a Random Forest.” International Conference on Information and Electronics Engineering (p. 194 : 198). Singapore: semanticscholar.org. 2011.
[18]. Jasmin Kevric et el. “An effective combining classifier approach using tree algorithms for network intrusion detection.” Neural Computing and Applications , 1051–1058. 2017.
[19]. ShivamB.Parikh and PradeepK.Atrey. “Media-RichFake News Detection: A Survey.” IEEE Conference on Multimedia Information. Miami, FL: IEEE. 2018.
[20]. MykhailoGranik and VolodymyrMesyura. “Fake news detection using naive Bayes classifier.” First Ukraine Conference on Electrical and Computer Engineering (UKRCON). Ukraine : IEEE. 2017.
[21]. Gilda, S. “Evaluating machine learning algorithms for fake news detection.” 15th Student Conference on Research and Development (SCOReD) (pp. 110-115). IEEE. 2017.
[22]. Akshay Jain and AmeyKasbe. “Fake News Detection.” 2018 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS). Bhopal, India: IEEE. 2018.
[23]. Yumeng Qin et al. “Predicting Future Rumours.” Chinese Journal of Electronics (Volume: 27 , Issue: 3 , 5 2018, 514 - 520.
[24]. ArushiGupta and RishabhKaushal. “Improving spam detection in Online Social Networks.” International Conference on Cognitive Computing and Information Processing (CCIP). semanticscholar.org.2015.