

IDENTIFYING FAKE NEWS ARTICLE USING MACHINE LERNING

A PROJECT - REPORT

*Submitted in partial fulfillment of the requirements for the award of the
degree of*

BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING

Submitted by

B. LALITHA	(20KH1A0512)
K. BINDU LAVANYA	(20KH1A0548)
G.L.N. SIVA KUMAR	(20KH1A0537)
K. PAVAN KUMAR	(20KH1A0554)

**Under the Esteemed Guidance of
Mr.Munnangi Suresh, M.Tech.
Assistant Professor**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
NARASARAOPETA INSTITUTE OF TECHNOLOGY**

**(Approved by A.I.C.T.E & Affiliated to J.N.T.U Kakinada)
(An ISO Certified Institution & Accredited by NBA)**

**NARASARAOPETA-522601
(2020-2024)**

NARASARAOPET INSTITUTE OF TECHNOLOGY

NARASARAOPET

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



CERTIFICATE

This is to certify that the project entitled “IDENTIFYING FAKE NEWS ARTICLE USING MACHINE LERNIG” is a bonafide project work carried out by the students **B. LALITHA (20KH1A0512), K. BINDU LAVANYA (20KH1A0548), G.L.N. SIVA KUMAR (20KH1A0537), K. PAVAN KUMAR (20KH1A0554),** under the Guidance and submitted in partial fulfillment for the award of **BACHELOR OF TECHNOLOGY** in **COMPUTER SCIENCE AND ENGINEERING** from JNTUK KAKINADA, during the year 2023-2024.

Project Guide

Mr.MUNNANGI SURESH., M.Tech.

Assistant Professor

Department of Computer Science and Engg.

Narasaraopeta Institute of Technology

Head of the department

Dr.R.SATHEESKUMAR., M.Tech.,Ph.D

Professor & Head

Department of Computer Science and Engg.

Narasaraopeta Institute of Technology

Submitted for the Project Viva – Voice examination held on _____

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

We wish to express our thanks to various personalities who are responsible for the completion of the Project-II. We are extremely thankful to our beloved Chairman **Sri. M.V. Koteswara Rao** who took keen interest in using every effort throughout this course. We owe out our gratitude to our principal **Dr. P. Hari Krishna Prasad** for his kind attention and valuable guidance throughout the course.

We would like to express my sincere thanks to our Head of the Department **Dr. R. Satheeskumar**, Professor, Department of Computer Science and Engineering, for his valuable guidance and support in completing my Project.

We express our thanks to our Guide **Mr.Munnangi Suresh**, Assistant Professor, Department of Computer Science and Engineering for his willingness and valuable guidance for successful completion of the Project.

We extend our sincere thanks to all other teaching and non-teaching staff of the Department for their cooperation and encouragement during our B.Tech course. We have no words to acknowledge the warm affection, constant inspiration and encouragement that we received from our parents.

SUBMITTED BY

B. LALITHA	(20KH1A0512)
K. BINDU LAVANYA	(20KH1A0548)
G.L.N. SIVA KUMAR	(20KH1A0537)
K. PAVAN KUMAR	(20KH1A0554)

DECLARATION

We (B.LALITHA, K.BINDU LAVANYA, G.L.N.SIVA KUMAR, K.PAVAN KUMAR), the students of the NARASARAOPETA INSTITUTE OF TECHNOLOGY here by this project titled “IDENTIFYING FAKE NEWS ARTICLE USING MACHINE LERNIG” being submitted to the Department of Computer Science and Engineering, NARASARAOPETA INSTITUTE OF TECHNOLOGY, Kotappakonda road, Yellamanda, Narasaraopeta, Guntur district. This project has not been submitted to any other University or Institute for the award of any degree.

Project Associates

B. LALITHA	(20KH1A0512)
K. BINDU LAVANYA	(20KH1A0548)
G.L.N. SIVA KUMAR	(20KH1A0537)
K. PAVAN KUMAR	(20KH1A0554)

ABSTRACT

This project helps us to detect the accuracy of the fake news using different classification techniques. Fake news is significantly affecting our social life, in fact in every field mainly in politics, education. In this project, we have presented the solution for Fake news problem by implementing fake news detection model by using different classification techniques. Fake News Detection becomes complicated when it comes to resources. Resources like datasets are limited. In this model, we have used classification techniques like Support Vector Machine(SVM), Naïve Bayes, Passive Aggressive Classifier. Output of our model using feature extraction techniques as Term Frequency-Inverted Document Frequency (TF-IDF) and Support Vector Machine (SVM) as classifier, has High accuracy.

TABLE OF CONTENT

CHAPTER NO	TITLE	PAGE NO
	Abstract	v
	Table of Content	vi
	List of Figures	viii
	List of Tables	xi
	List of Abbreviations	x
1	INTRODUCTION	1
	1.1 PROJECT OVERVIEW	1
	1.2 ML LAYER ANALYSIS ARCHITECTURE	1
	1.3. PROBLEM DEFINITION	2
	1.4 OBJECTIVES	3
2	LITERATURE SURVEY	4
3	SYSTEM ANALYSIS	9
	3.1. EXISTING SYSTEM	9
	3.1.1 Disadvantages of Existing System	9
	3.2. PROPOSED SYSTEM	9
	3.2.1 Advantages of Proposed System	9
4	SYSTEM SPECIFICATION	10
	4.1. SOFTWARE SPECIFICATION	10
	4.2. HARDWARE SPECIFICATION	10
	4.3 SOFTWARE DESCRIPTION	10
5	SYSTEM DESIGN	16
	5.1 SYSTEM ARCHITECTURE	16
	5.2 USE CASE DIAGRAM	18

	5.3 CLASS DIAGRAM	18
	5.4 SEQUENCE DIAGRAM	19
	5.5 ACTIVE DIAGRAM	20
6	SYSTEM IMPLEMENTATION	21
	6.1. MODULES	21
	6.1.1 Data Collection	21
	6.1.2 Text Prepossessing	21
	6.1.3 Model Selection	21
	6.1.4 Training the model	24
	6.1.5 Validation &Hyper parameter Tuning	24
	6.1.6 Testing & Evaluation	24
7	RESULTS AND DISCUSSION	25
	7.1 INTRODUCTION	25
	7.2 EVALUATION METRICS	25
	7.3 PERFORMANCE ANALYSIS	25
	7.4 CHALLENGES IN FAKE NEWS DETECTION	27
8	SYSTEM TESTING	33
	8.1.TESTING OF PRODUCT	33
	8.2 SOFTWARE IMPLEMENTATION	34
	8.3 SOFTWARE TESTING	35
9	9.1. CONCLUSION	36
	9.2. FUTURE ENHANCEMENT	36
	APPENDICES	37
	1. SAMPLE CODING	37
	REFERENCES	43

LIST OF FIGURES

FIGURE NO	TITLE	PAGE NO
1.1	Fake News image analysis Architecture	2
5.1	System Architecture	17
6.1.3.1	Classify of Two different category using hyperplane	22
6.1.3.2	Single straight line classifies two classes of linear data set	22
6.1.3.3	Non linear data set	23

LIST OF TABLES

FIGURE NO	TITLE	PAGE NO
7.3.1	Confusion Matrix	26
7.3.2	Passive Aggressive	26
7.3.3	Navie Bayes - TF-IDF	26
7.3.4	SVM confusion matrix	27
7.3.5	Performs Measure	27

LIST OF ABBREVIATIONS

NLP	Natural Language Processing
SVM	Support Vector Machine
NER	Named Entity Recognition
RNN	Recurrent Neural Networks
PDS	Project Developed System
TF-IDF	Term Frequency-Inverse Document Frequency

CHAPTER - 1

INTRODUCTION

1.1 PROJECT OVERVIEW

In the recent years, Social Media has been dominant in everyone's life. Fake news spreads mostly through social media. Fake news is threat to the politics, finance, education, democracy, business. Although fake news is not a new problem but today humans believe more in social media which leads to believe in fake news and then spread of the same fake news. It is becoming tough nowadays to distinguish between true and false news which creates problems, misunderstanding. It is difficult to manually identify fake news, its only possible when the person identifying the news has a vast knowledge on the topic of news. Due to recent advancements in computer science, it is easier nowadays to create and spread fake news but it is considerably hard to distinguish the information as true or false. This fake news can affect some products, business if fake news is spread about the products. In politics too, fake news can affect someone's career. "2019 has been a unique year where fact checkers continuously kept moving from one event to the other, and this has been the busiest year for us so far," said Jency Jacob, managing director at Mumbai based fact-checking website BOOM, which works with Facebook to check stories and tags specific posts spreading misinformation on the platform. We have compare three different supervised classification technique, Naive Bayes Classifier, Support Vector Machine (SVM), Passive Aggressive Classifier. We have used a dataset which contains real and fake news and it yields best results.

1.2 ARCHITECTURE OF LAYER IMAGE ANALYSIS

Fake news detection involves a multifaceted approach encompassing various stages. In the initial phase of data collection, emphasis is placed on sourcing information from diverse platforms, with particular attention given to reputable outlets. Subsequently, features are extracted from the collected data, including linguistic patterns, user behavior, and metadata. In the modeling stage, a range of techniques is utilized, encompassing supervised learning with algorithms like SVM and Random Forest, unsupervised learning employing K-means or topic modeling, and ensemble models to enhance overall accuracy. Natural

Language Processing (NLP) techniques, such as Named Entity Recognition (NER) and sentiment analysis, are integrated to discern entities within the text and gauge its overall tone. Deep learning approaches, including Recurrent Neural Networks (RNN) and transformer models like BERT and GPT, are leveraged for their ability to analyze sequential patterns and extract features from pre-trained language models.

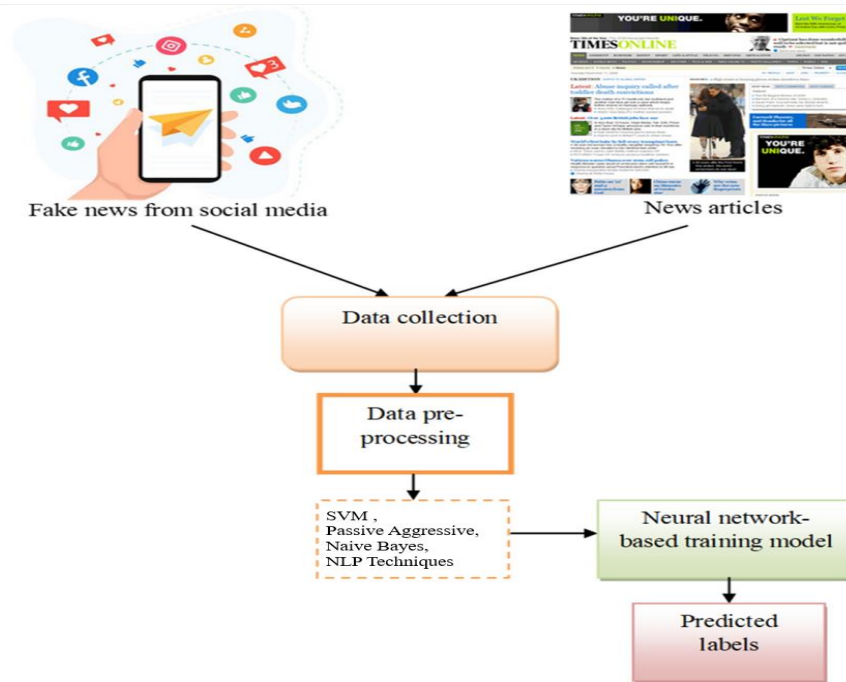


Figure 1.1 Fake News image analysis Architecture

1.3 PROBLEM DEFINITION

The fake news detection problem involves the development of methods and systems to automatically identify and distinguish between accurate, trustworthy information and intentionally deceptive or misleading content disseminated through various media channels. This challenge is rooted in the increasing prevalence of misinformation, rumors, and fabricated news stories in digital and traditional media platforms. The goal is to design and implement effective algorithms, models, and techniques that can analyze textual, visual, or audio content to assess its authenticity, credibility, and accuracy. Addressing the fake news detection problem requires a multidisciplinary approach, incorporating natural language processing, machine learning, data mining, and ethical considerations to build robust systems capable of mitigating the impact of false information on individuals, communities, and society at large.

1.4 OBJECTIVES

The main objective of fake news article detection in machine learning is to develop algorithms that can accurately identify and distinguish between genuine and misleading information.

CHAPTER - 2

LITERATURE SURVEY

1) When Fake News Becomes Real: Combined Exposure to Multiple News Sources and Political Attitudes of Inefficacy, Alienation, and Cynicism

AUTHORS: M. Balmas

This research assesses possible associations between viewing fake news (i.e., political satire) and attitudes of inefficacy, alienation, and cynicism toward political candidates. Using survey data collected during the 2006 Israeli election campaign, the study provides evidence for an indirect positive effect of fake news viewing in fostering the feelings of inefficacy, alienation, and cynicism, through the mediator variable of perceived realism of fake news. Within this process, hard news viewing serves as a moderator of the association between viewing fake news and their perceived realism. It was also demonstrated that perceived realism of fake news is stronger among individuals with high exposure to fake news and low exposure to hard news than among those with high exposure to both fake and hard news. Overall, this study contributes to the scientific knowledge regarding the influence of the interaction between various types of media use on political effects.

2) Miley, CNN and The Onion

AUTHORS: D. Berkowitz and D. A. Schwartz

Following a twerk-heavy performance by Miley Cyrus on the Video Music Awards program, CNN featured the story on the top of its website. The Onion— a fake-news organization—then ran a satirical column purporting to be by CNN's Web editor explaining this decision. Through textual analysis, this paper demonstrates how a Fifth Estate comprised of bloggers, columnists and fake- news organizations worked to relocate mainstream journalism back to within its professional boundaries.

3) The Impact of Real News about “Fake News”: Intertextual Processes and Political Satire

AUTHORS: P. R. Brewer, D. G. Young, and M. Morreale

This study builds on research about political humor, press Meta coverage, and

intertextuality to examine the effects of news coverage about political satire on audience members. The analysis uses experimental data to test whether news coverage of Stephen Colbert's Super PAC influenced knowledge and opinion regarding Citizens United, as well as political trust and internal political efficacy. It also tests whether such effects depended on previous exposure to The Colbert Report (Colbert's satirical television show) and traditional news. Results indicate that exposure to news coverage of satire can influence knowledge, opinion, and political trust. Additionally, regular satire viewers may experience stronger effects on opinion, as well as increased internal efficacy, when consuming news coverage about issues previously highlighted in satire programming.

4) Stopping Fake News

AUTHORS: M. Haigh, T. Haigh, and N. I. Kozak

Social media is acting as a double-edged sword for universe in a way of consuming news. On one side, its ease of access, popularity and low cost distribution channel lead people to gain news from social media. On other side, it is also acting as a source of spread of 'fake news'. The extensive spread of fake news on social media, websites are impacting society negatively. This makes extremely important to combat the spread of fake news and to aware the society. In this paper, we offer a review which lists out the sources of fake news, its types, generation, motivation and examples. Also, some approaches are suggested to spot and stop fake news spread.

5) With Facebook, Blogs, and Fake News, Teens Reject Journalistic "Objectivity"

AUTHORS: R. Marchi

This article examines the news behaviors and attitudes of teenagers, an understudied demographic in the research on youth and news media. Based on interviews with 61 racially diverse high school students, it discusses how adolescents become informed about current events and why they prefer certain news formats to others. The results reveal changing ways news information is being accessed, new attitudes about what it means to be informed, and a youth preference for opinionated rather than objective news. This does not indicate that young people disregard the basic ideals of professional journalism but, rather, that they desire more authentic renderings of them.

6) Social Media and Fake News in the 2016 Election

AUTHORS: H. Allcott and M. Gentzkow

Following the 2016 US presidential election, many have expressed concern about the effects of false stories ("fake news"), circulated largely through social media. We discuss the economics of fake news and present new data on its consumption prior to the election. Drawing on web browsing data, archives of fact-checking websites, and results from a new online survey, we find: 1) social media was an important but not dominant source of election news, with 14 percent of Americans calling social media their "most important" source; 2) of the known false news stories that appeared in the three months before the election, those favoring Trump were shared a total of 30 million times on Facebook, while those favoring Clinton were shared 8 million times; 3) the average American adult saw on the order of one or perhaps several fake news stories in the months around the election, with just over half of those who recalled seeing them believing them; and 4) people are much more likely to believe stories that favor their preferred candidate, especially if they have ideologically segregated social media networks.

7) The spread of fake news by social bots.

AUTHORS: C. Shao, G. L. Ciampaglia, O. Varol, A. Flammini, and F. Menczer

The massive spread of fake news has been identified as a major global risk and has been alleged to influence elections and threaten democracies. Communication, cognitive, social, and computer scientists are engaged in efforts to study the complex causes for the viral diffusion of digital misinformation and to develop solutions, while search and social media platforms are beginning to deploy countermeasures. However, to date, these efforts have been mainly informed by anecdotal evidence rather than systematic data. Here we analyze 14 million messages spreading 400 thousand claims on Twitter during and following the 2016 U.S. presidential campaign and election. We find evidence that social bots play a key role in the spread of fake news. Accounts that actively spread misinformation are significantly more likely to be bots. Automated accounts are particularly active in the early spreading phases of viral claims, and tend to target influential users. Humans are vulnerable to this manipulation, retweeting bots who post false news. Successful sources of false and biased claims are heavily supported by social bots. These results suggests that curbing social bots may be an effective strategy for mitigating.

8) Faking Sandy: Characterizing and Identifying Fake Images on Twitter during Hurricane Sandy

AUTHORS: A. Gupta, H. Lamba, P. Kumaraguru, and A. Joshi

In today's world, online social media plays a vital role during real world events, especially crisis events. There are both positive and negative effects of social media coverage of events, it can be used by authorities for effective disaster management or by malicious entities to spread rumors and fake news. The aim of this paper, is to highlight the role of Twitter, during Hurricane Sandy (2012) to spread fake images about the disaster. We identified 10,350 unique tweets containing fake images that were circulated on Twitter, during Hurricane Sandy. We performed a characterization analysis, to understand the temporal, social reputation and influence patterns for the spread of fake images. Eighty six percent of tweets spreading the fake images were retweets, hence very few were original tweets. Our results showed that top thirty users out of 10,215 users (0.3%) resulted in 90% of the retweets of fake images; also network links such as follower relationships of Twitter, contributed very less (only 11%) to the spread of these fake photos URLs. Next, we used classification models, to distinguish fake images from real images of Hurricane Sandy. Best results were obtained from Decision Tree classifier, we got 97% accuracy in predicting fake images from real. Also, tweet based features were very effective in distinguishing fake images tweets from real, while the performance of user based features was very poor. Our results, showed that, automated techniques can be used in identifying real images from fake images posted on Twitter.

9) The Fake News Spreading Plague: Was it Preventable

AUTHORS: E. Mustafaraj and P. T. Metaxas

In 2010, a paper entitled "From Obscurity to Prominence in Minutes: Political Speech and Real-time search" won the Best Paper Prize of the Web Science 2010 Conference. Among its findings were the discovery and documentation of what was termed a "Twitter-bomb", an organized effort to spread misinformation about the democratic candidate Martha Coakley through anonymous Twitter accounts. In this paper, after summarizing the details of that event, we outline the recipe of how social networks are used to spread misinformation. One of the most important steps in such a recipe is the "infiltration" of a community of users who are already engaged in conversations about a topic, to use them as organic spreaders of

misinformation in their extended subnetworks. Then, we take this misinformation spreading recipe and indicate how it was successfully used to spread fake news during the 2016 U.S. Presidential Election. The main differences between the scenarios are the use of Facebook instead of Twitter, and the respective motivations (in 2010: political influence; in 2016: financial benefit through online advertising). After situating these events in the broader context of exploiting the Web, we seize this opportunity to address limitations of the reach of research findings and to start a conversation about how communities of researchers can increase their impact on real-world societal issues.

10) Fake News Mitigation via Point Process Based Intervention.

AUTHORS: M. Farajtabar et al.

We propose the first multistage intervention framework that tackles fake news in social networks by combining reinforcement learning with a point process network activity model. The spread of fake news and mitigation events within the network is modeled by a multivariate Hawkes process with additional exogenous control terms. By choosing a feature representation of states, defining mitigation actions and constructing reward functions to measure the effectiveness of mitigation activities, we map the problem of fake news mitigation into the reinforcement learning framework. We develop a policy iteration method unique to the multivariate networked point process, with the goal of optimizing the actions for maximal total reward under budget constraints. Our method shows promising performance in real-time intervention experiments on a Twitter network to mitigate a surrogate fake news campaign, and outperforms alternatives on synthetic datasets

CHAPTER - 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

Up to now, most of the research on PDS has focused on how to enforce user privacy preferences and how to secure data when stored into the PDS. In contrast, the key issue of helping users to specify their privacy preferences on PDS data has not been so far deeply investigated. It can be classify into the KNN,Decision Tree.

3.1.1 Disadvantages of Existing System

- Limited Context Understanding
- Bias and Generalization Issues
- Adversarial Attacks
- Multimodal Content Challenges

3.2. PROPOSED SYSTEM

The proposed system combines these innovations to create a comprehensive and adaptive solution for fake news detection, addressing the evolving challenges in the information landscape.It can be used in the SVM,Passive Aggressive,Naive Bayes Classification techniques are used.

A proposed system for fake news detection aims to overcome the limitations of existing approaches by integrating advanced technologies and methodologies.

3.2.1 Advantages of Proposed System

- Enhanced Context Awareness
- Bias Mitigation and Fairness
- Adversarial Robustness
- Comprehensive Multimodal Analysis
- Balanced Dataset Construction

CHAPTER - 4

SYSTEM SPECIFICATION

4.1. HARDWARE REQUIREMENTS:

- ❖ **System** : Intel i3 2.2Ghz
- ❖ **Hard Disk** : 320 GB.
- ❖ **Ram** : 4 GB.

4.2. SOFTWARE REQUIREMENTS:

- ❖ **Operating system** : Windows 7 Ultimate.
- ❖ **Coding Language** : Python.
- ❖ **Front-End** : Python.
- ❖ **IDE** : Pycharm 2020 Community.

4.3 SOFTWARE DESCRIPTION

Python:

Python is a high-level, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation via the off-side rule. Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly procedural), object oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library. Guido van Rossum began working on Python in the late 1980s as a successor to the ABC programming language and first released it in 1991 as Python 0.9.0. Python 2.0 was released in 2000. Python 3.0, released in 2008, was a major revision not completely backward-compatible with earlier versions. Python 2.7.18, released in 2020, was the last

release of Python 2. Python consistently ranks as one of the most popular programming languages.

Machine Learning (ML) is a subset of artificial intelligence (AI) that focuses on creating algorithms and models capable of learning from data to make predictions or decisions without being explicitly programmed. The core idea behind machine learning is to enable systems to improve their performance over time through experience.

Key Components of Machine Learning:

Data:

Training Data: ML algorithms require large datasets to learn patterns and relationships. Training data consists of input features and corresponding output labels

Algorithms:

Supervised Learning: Involves training a model on a labeled dataset, where the algorithm learns to map input data to the correct output by generalizing from examples.

Unsupervised Learning: In this type, the algorithm works with unlabeled data, discovering patterns, structures, or relationships within the data without explicit guidance.

Reinforcement Learning: Focuses on training models to make sequences of decisions by receiving feedback in the form of rewards or penalties based on the actions taken.

Models:

A model is a mathematical representation created by a machine learning algorithm after training on data. It is used to make predictions or decisions when presented with new, unseen data.

Features and Labels:

Features: These are the input variables or characteristics used by the algorithm to make predictions. Labels: In supervised learning, labels are the output variables that the algorithm aims to predict.

Training and Testing:

Training Phase: During this phase, the model is exposed to the training data to learn patterns and relationships.

Testing Phase: The model is evaluated on a separate set of data that it has not seen before to assess its generalization and predictive performance.

Overfitting and Underfitting:

Overfitting: Occurs when a model learns the training data too well, capturing noise and outliers, but fails to generalize to new, unseen data.

Underfitting: Occurs when a model is too simple to capture the underlying patterns in the training data.

Feature Engineering:

The process of selecting, transforming, or creating new features from the raw data to improve the model's performance.

Hyperparameters:

Parameters that are not learned from the data but are set prior to training. Tuning these hyperparameters can significantly impact a model's performance.

Deployment:

The process of integrating a trained model into a real-world system or application to make predictions on new data.

Overview of Machine learning

Machine Learning (ML) is a subset of artificial intelligence (AI) that focuses on creating algorithms and models capable of learning from data to make predictions or decisions without being explicitly programmed.. Machine learning is a rapidly evolving field with continuous advancements, driven by research, data availability, and improvements in computing power. Its versatility and ability to handle complex tasks make it a cornerstone of modern AI applications.

NLP:

Natural Language Processing (NLP) is a multifaceted field within artificial intelligence that enables computers to interact with and understand human language. In its operational workflow, NLP starts with data collection, where diverse textual datasets are gathered, spanning articles, social media posts, and various forms of written communication. The subsequent text pre-processing involves cleaning and normalization through tasks like tokenization and stemming. Feature extraction techniques, including Bag-of-Words, TF-IDF, and word embeddings, transform raw text into numerical representations, making it machine-readable. NLP incorporates a range of models, from traditional machine learning algorithms like SVM and Naive Bayes to advanced deep learning architectures such as Recurrent Neural Networks (RNNs) and transformer models like BERT.

Feature Extraction Techniques:

Feature extraction is a critical step in NLP, involving the transformation of raw text into numerical representations that machine learning models can understand. Some key feature extraction techniques include:

Bag-of-Words (BoW):

Represents text as an unordered set of words, disregarding grammar and word order. Each word is assigned a unique identifier, and the frequency of each word is used as a feature.

TF-IDF (Term Frequency-Inverse Document Frequency):

Measures the importance of a word in a document relative to its frequency across the entire dataset. It helps identify words that are distinctive to a specific document.

N-grams:

O-Considers sequences of adjacent words as features, capturing local context. Bigrams (two-word sequences) and trigrams (three-word sequences) are common in NLP applications.

Real-world Applications

NLP has found widespread applications in various domains:

Fake News Detection:

Using NLP to discern patterns and linguistic features that distinguish between genuine and deceptive content, contributing to combating misinformation.

Named Entity Recognition (NER):

Identifying and classifying entities such as names, locations, and organizations in text. This is crucial for information extraction.

Chatbots and Virtual Assistants:

NLP powers conversational agents, allowing users to interact with machines in natural language for tasks such as customer support or information retrieval.

Characteristics of NLP

NLP implementation the following features

- Ambiguity Handling.
- Multimodal Processing.
- Semantics Understanding.
- Machine Translation.
- Ethical Considerations.

Overview of NLP

Natural Language Processing (NLP) is a multifaceted field within artificial intelligence that aims to facilitate communication between computers and human language. The operational workflow of NLP encompasses several interconnected stages. It commences with the collection of diverse textual datasets, ranging from articles and social media posts to more extensive documents.

FAKE NEWS DETECTION USING NATURAL LANGUAGE PROCESSING:

NLP plays an important role in detecting fake news by using different techniques to analyze the text data. Different techniques used by NLP for fake news detection:

1.Text Classification: NLP models are trained to classify news articles or social media posts as either real or fake based on patterns like word structure, sentence structure, or other features.

2. Sentiment Analysis: NLP analyzes sentiments expressed in news articles or social media to detect any misleading content by examining the emotional tone and language used in the text. It can flag any deceptive or manipulative information.

3.Named Entity Recognition (NER): NLP techniques can identify any named entities mentioned in the text like people's names, organizations, locations, and dates. If any false information is found with the organization or the person's name across different sources, it would flag them.

4.Semantic Analysis: NLP models can also understand the semantic meaning of text by understanding the relationship between different words and phrases. It tries to get information from the data in real time and flags any contextual clues that indicate misinformation.

5.Topic Modeling: NLP techniques have the ability to tell us about the underlying topics or themes present in a collection of articles. It tells us about patterns of content manipulation or agenda-driven objectives used in fake news.

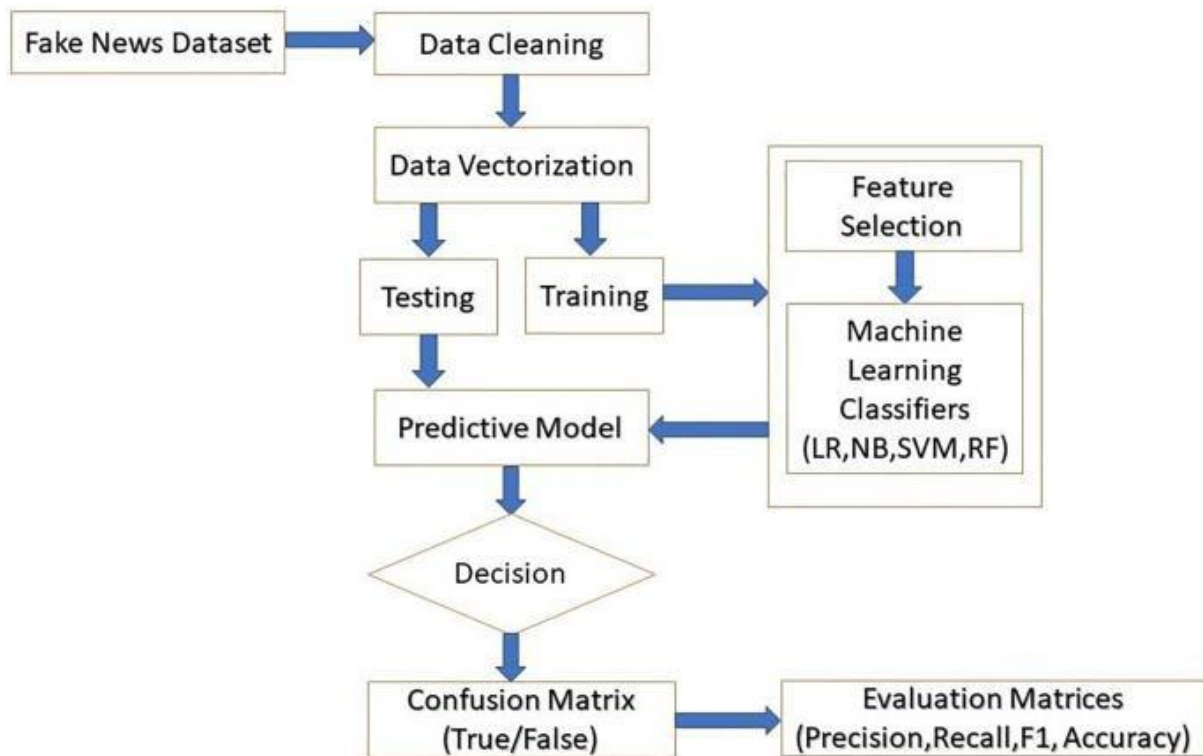
6.Fact-Checking: NLP systems can automatically verify the accuracy of claims made in news articles by cross-referencing them with reliable sources. Fact-checking algorithms can easily identify inconsistencies and contradictions in new stories.

7.Network Analysis: NLP can analyze the network structure of social media to identify accounts or bots spreading fake news. It would examine the patterns of communication, and information network analysis can detect fake news campaigns.

CHAPTER - 5

SYSTEM DESIGN

5.1. SYSTEM ARCHITECTURE



5.1. SYSTEM ARCHITECTURE

The system architecture for fake news detection integrates Natural Language Processing (NLP) and machine learning to effectively analyze and classify news articles. Beginning with the collection of a diverse dataset containing labeled instances of genuine and fake news, the system undergoes a comprehensive process. NLP techniques extract pertinent features such as word frequencies, sentiment scores, and named entities, capturing linguistic nuances. A carefully chosen machine learning model, ranging from traditional approaches like Support Vector Machines, is trained using the labeled dataset. The model learns to identify patterns associated with authentic and deceptive news during this phase. Validation and testing on separate datasets ensure the model's accuracy and generalization. Ensemble methods may be employed for enhanced robustness. The trained model is then integrated into a real-time analysis system, capable of processing and classifying news articles as they emerge. Overall, this architecture highlights the synergy between NLP and machine learning in combating misinformation by thoroughly examining linguistic features within news content.

UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

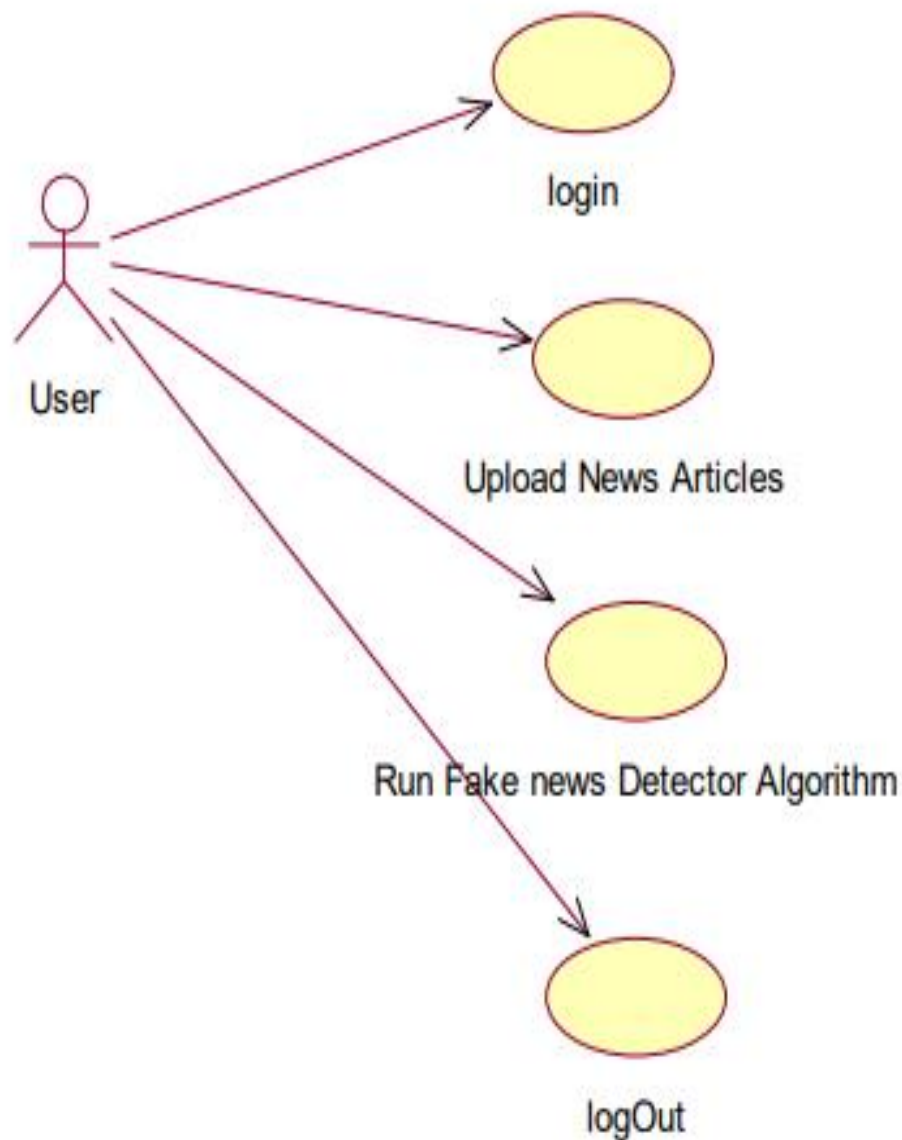
The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

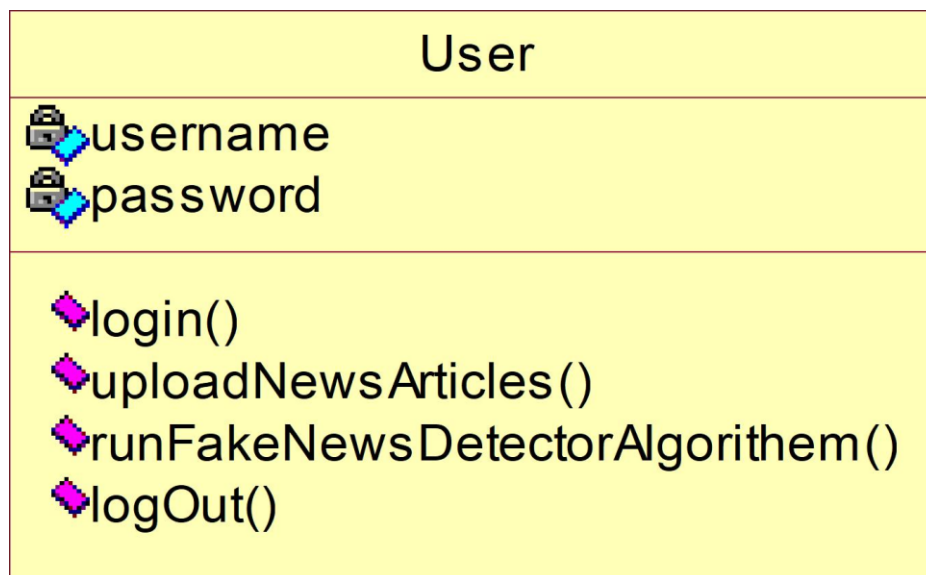
5.2. USE CASE DIAGRAM

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.



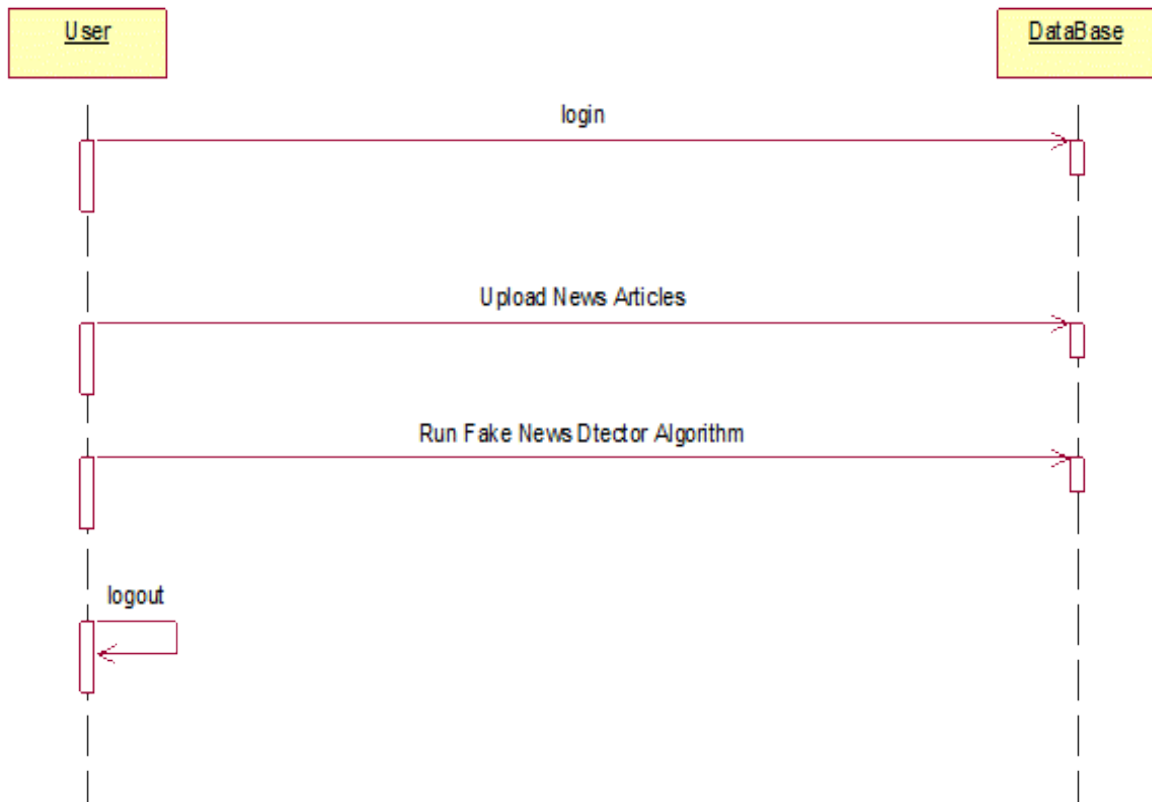
5.3. CLASS DIAGRAM

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.



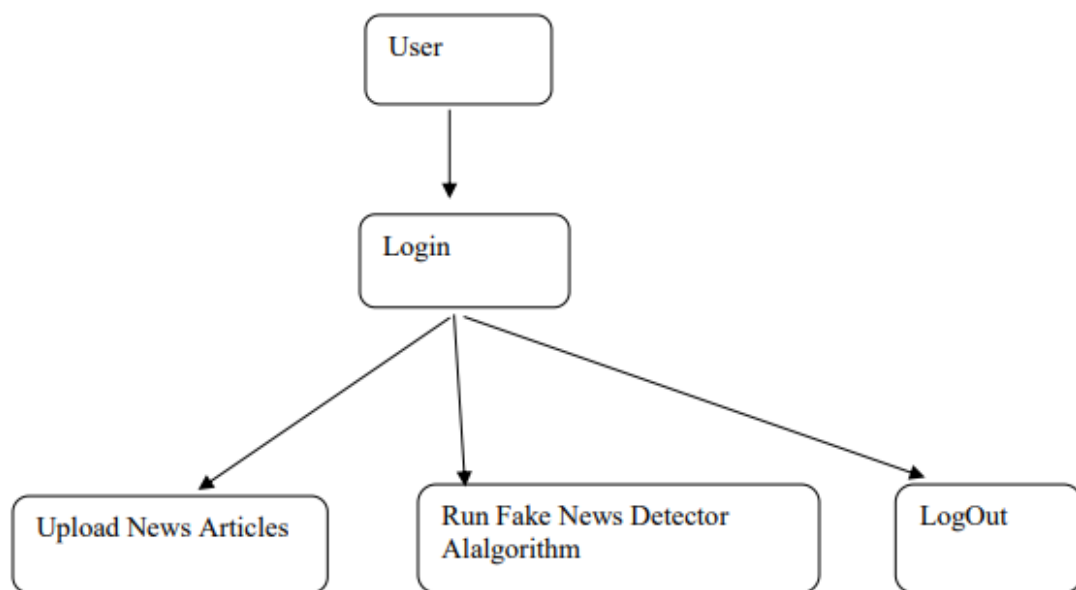
5.4. SEQUENCE DIAGRAM

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.



5.5. ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.



CHAPTER - 6

SYSTEM IMPLEMENTATION

6.1. MODULES

1. Data Collection
2. Text Pre-processing
3. Model Selection
4. Training the Model
5. Validation and Hyperparameter Tuning
6. Testing and Evaluation

6.1.1 Data Collection:

Gather diverse datasets containing textual information. This data could include documents, articles, social media posts, or any other form of written communication it contain the thous eds of data.

6.1.2 Text Pre-processing:

Clean and preprocess the raw text to remove noise and irrelevant information. This step involves tasks such as tokenization (breaking text into words or phrases), stemming (reducing words to their base form), and lemmatization (reducing words to their dictionary form).

Extract relevant features from the text. This could include information about word frequency, n-grams (sequential word combinations), sentiment analysis scores, and named entities (identifiable elements such as names, locations, organization.

6.1.3 Model Selection:

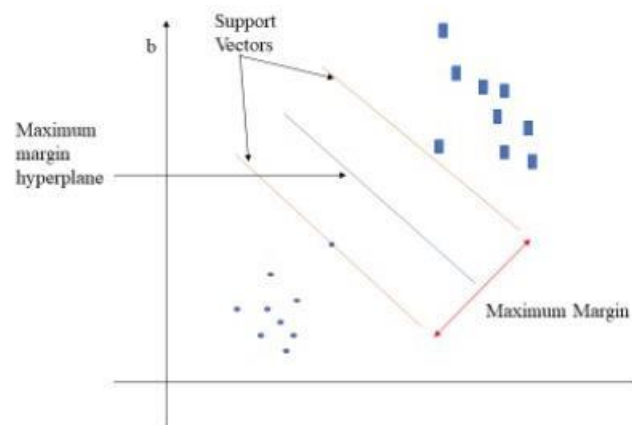
Choose a suitable machine learning or deep learning model for the specific NLP task. Common models include Support Vector Machines (SVM), Naive Bayes, Passive Aggressive

SVM Classification Algorithm:

Support Vector Machine (SVM)

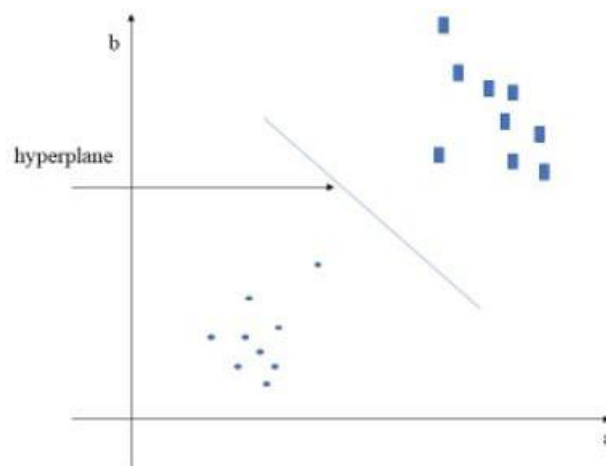
A support vector machine (SVM), is a managed learning calculation. Hence, the model is constructed after it has already been trained. The main motive of SVM is to categorize new data that comes under. There is a decision boundary or hyperplane that splits dataset into two

class. For the considered class, a point is chosen such that it is close to the opponent class. A line is drawn touching the point parallel to hyperplane. Hyperplane is drawn considering maximum margin. SVM are more accurate on smaller dataset. The disadvantage of using SVM on large dataset is training time is high.



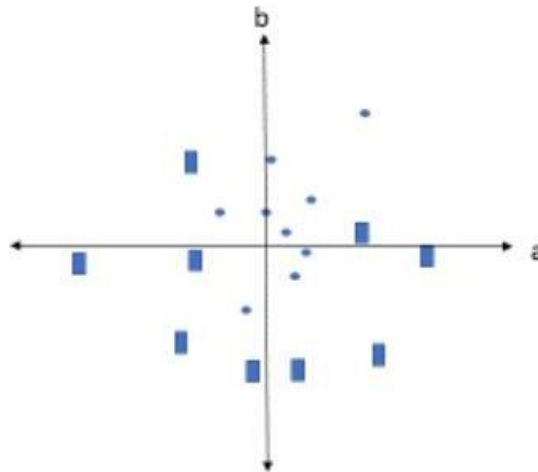
6.1.3.1 Classification of two different categories using hyperplane

There are two types of SVM model. Linear SVM is used for linearly separable data. When Single straight line classify two classes of a dataset, such data is called as linearly separable data and the classifier used for this type of data is Linear SVM.



6.1.3.2 Single Straight line classifies two classes of linear dataset

In figure 6.1.5, as it is 2-D space, a single straight line can easily classify two classes. However, there can be multiple lines possible that can separate data. SVM chooses best hyperplane (straight line) considering maximum margin between support vectors for nonlinear data, a single straight line cannot separate data into two class. We have to consider 3-D space for it. For linear SVM, 2-D space was used.



6.1.3.3 Nonlinear data

Passive Aggressive Classification Algorithm:

Passive Aggressive algorithms are online learning algorithms and used for both regression as well as classification. It is easy to use and work fast as compared to SVM but does not provide high accuracy like SVM. It is mainly used to classify massive data. The algorithm remains passive for a correct classification outcome, and it turns aggressive if it is an incorrect classification, updating and adjusting.

Naive Bayes Classification Algorithm:

Naïve Bayes is used for calculating conditional probability. It is derived from Bayes theorem. It states that “probability that something will happen, given that something else has already occurred” (Saxena, 2017). In Naive Bayes, occurrence of one feature is independent of other feature. Naïve Bayes is a type of classifier and it’s a supervised learning algorithm. The prediction occurs on the basis of probability. It makes quick predictions for the machine learning models. It works best with text classification. Naive Bayes Classifier is used for

multiclass and binary classifications. But the disadvantage of using it is, classifier fails to learn the relationship between features as it treats all features independent of each other.

$$P(A/B) = P(B/A)P(A) / P(B)$$

$P(A|B)$: Probability of event A such that event B has already occurred.

6.1.4 Training the Model:

Train the selected model using labeled data. In supervised learning, the model learns to recognize patterns and relationships between the extracted features and the target labels.

6.1.5 Validation and Hyperparameter Tuning:

Validate the model's performance on a separate dataset not used during training. Adjust hyperparameters (configurable settings) to optimize the model's performance, avoiding overfitting or underfitting.

6.1.6 Testing and Evaluation:

Evaluate the trained model on a held-out testing dataset to assess its generalization performance. Common evaluation metrics include accuracy, precision, recall, and F1-score.

CHAPTER - 7

RESULTS AND DISCUSSION

7.1 INTRODUCTION

The primary purpose of this paper is to review numerous publications in the field of deep learning applications in medical images. Classification, detection, and segmentation are essential tasks in medical image processing. For specific deep learning tasks in medical applications, the training of deep neural networks needs a lot of labeled data. But in the medical field, at least thousands of labeled data is not available. This issue is alleviated by a technique called transfer learning. Two transfer learning approaches are popular and widely applied that are fixed feature extractors and fine-tuning a pre-trained network. In the classification process, the deep learning models are used to classify images into two or more classes. In the detection process, Deep learning models have the function of identifying tumors and organs in medical images.

7.2 PERFORMANCE MEASURES

The proposed algorithm has been assessed through various performance evaluation metrics that include True Positive, True Negative the former one that designates how many times does the proposed algorithm is able to correctly recognize the damaged region as damaged region and the later one designates how many times does the proposed algorithm correctly identified non-damaged region as non-damaged region. And the False Positive (FP) and False Negative (FN) the former one designates how many times does the proposed algorithm fails to recognize the damaged region correctly, and the later represents how many times does the proposed algorithm fails to identify the non-tumors region as non-tumors regions. Basing on values of TP, TN, FP, and FN, the values of Accuracy, Specificity and sensitivity are calculated of the proposed algorithm.

7.3 PERFORMANCE EVALUTION

Implementation was done using the above algorithms with Vector features- Count Vectors and Tf-Idf vectors at Word level and Ngram

level. Accuracy was noted for all models. We used K-fold cross validation technique to improve the effectiveness of the models.

A. Dataset split using K-fold cross validation

This cross-validation technique was used for splitting the dataset randomly into k-folds. (k-1) folds were used for building the model while kth fold was used to check the effectiveness of the model. This was repeated until each of the k-folds served as the test set. I used 3- fold cross validation for this experiment where 67% of the data is used for training the model and remaining 33% for testing.

B. Confusion Matrices for Static System

After applying various extracted features (Bag-of-words, Tf-Idf, N-grams) on three different classifiers (Naïve bayes, Logistic Regression and Random Forest), their confusion matrix showing actual set and predicted sets are mentioned below

Table 7.3.1 confusion matrix

N	PREDICTED POSITIVE	PREDICTED NEGATIVE
ACTUAL POSITIVE	TRUE POSITIVE(TP)	FLASE NEGATIVE(FN)
ACTUAL NEGATIVE	FLASE POSITIVE (FP)	TRUE NEGATIVE(TN)

Table 7.3.2. confusion matrix using passive aggressive classifier

N = 1267	Predicted Yes	Predicted No
Actual Yes(638)	TP(587)	FN(51)
Actual No(629)	FP(40)	TN(589)

Table 7.3.3. confusion matrix using Naive Bayes classification

N=1267	Predicted Yes	Pridicted No
Actual Yes (638)	TP(450)	FN(188)
Actual No(629)	FP(14)	TN(615)

Table 7.3.4. confusion matrix using SVM

N=1267	Predicted Yes	Predicted No
Actual Yes(638)	TP(598)	FN(40)
Actual No(629)	FP(48)	TN(581)

Table 7.3.5. Performance Parameters of over all models

CLASSIFIER	Accurency	Recall	Precision	Fakenews measure
NAIVE BAYES CLASSIFIER	84.056%	70.53%	96.98%	81.666%
PASSIVEAGGRESSIVE CLASSIFIER	92.2%	92%	93.62%	92.80%
SUPPORT VECTOR MACHINE	95.05%	93.73%	92.56+%	93.141%

7.4 CHALLENGES IN FAKE NEWS DETECTION

The primary challenge in fake news detection stems from the intricate and dynamic nature of misinformation. The diversity of deceptive tactics, ranging from subtle manipulations to outright fabrications across various topics and contexts, poses a significant hurdle. Misinformation evolves rapidly, with creators adapting techniques such as deepfakes, altered headlines, and context manipulation, necessitating detection systems to constantly evolve. Contextual understanding presents a formidable challenge, as fake news often exploits nuances of language, sarcasm, and cultural references that are complex for automated systems to grasp.

Document examples

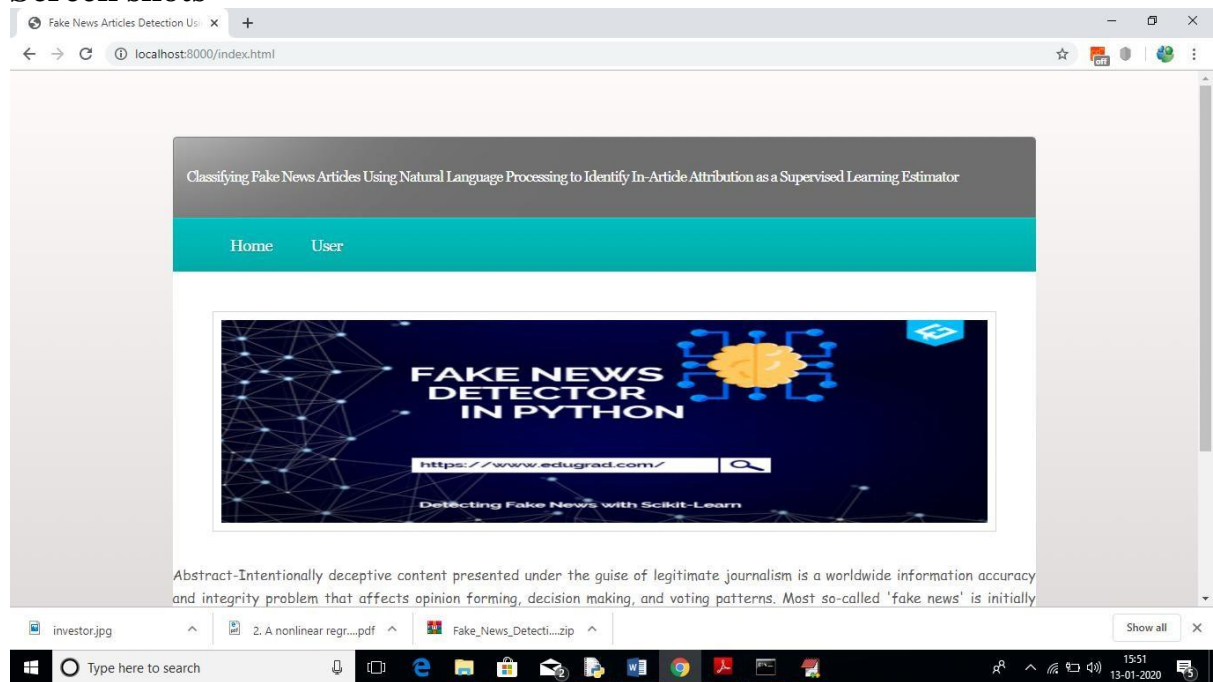
When “Mitt Romney” was governor of Massachusetts, we didnt just slow the rate of growth of our government, we actually cut it."

In above sentence quotes are there and it's talking about 'Mitt Romney' and it's contains some verbs such as 'was, didn't, slow, cut'. By analyzing above 3 features from articles we can come to the conclusion whether news is FAKE or REAL.

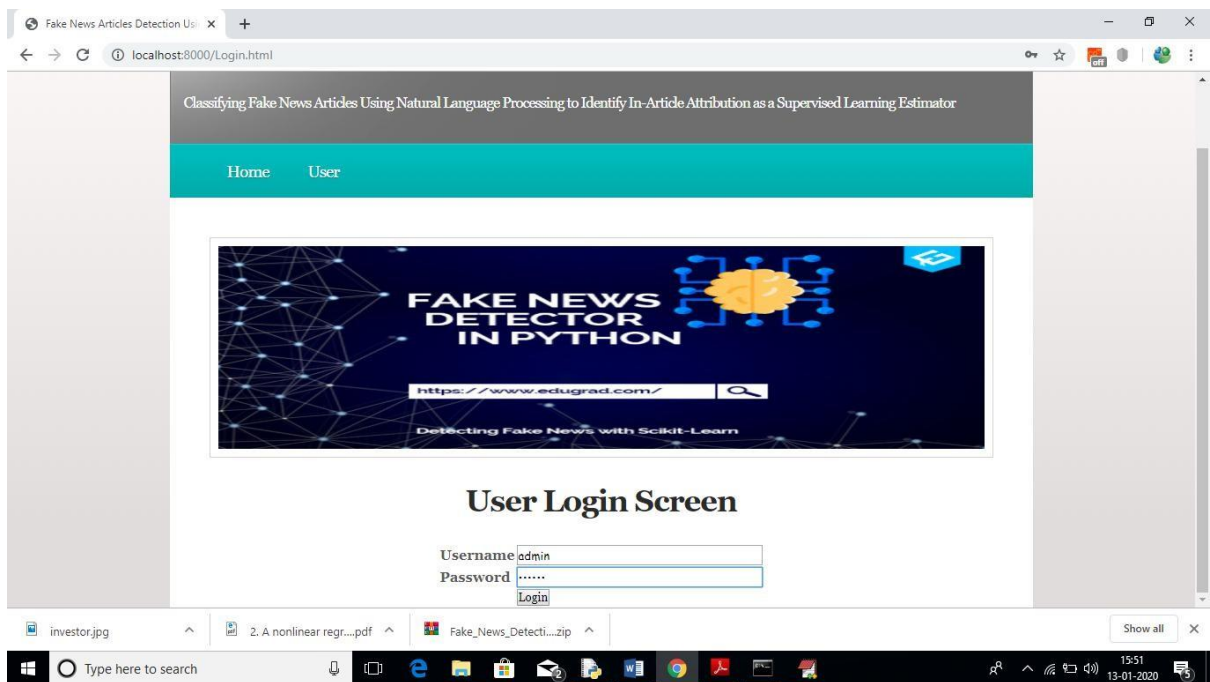
All FAKE peoples will not write such statements in their articles so we can detect by applying this techniques.

- To implement this project we are using 'News' dataset and then by applying above technique we can detect whether this news are fake or real. This dataset I kept inside dataset folder. Upload this dataset when you are running application.
- To run this project deploy 'FakeNews' folder on 'django' python web server and then start server and run in any web browser. After running code in web browser will get below page.

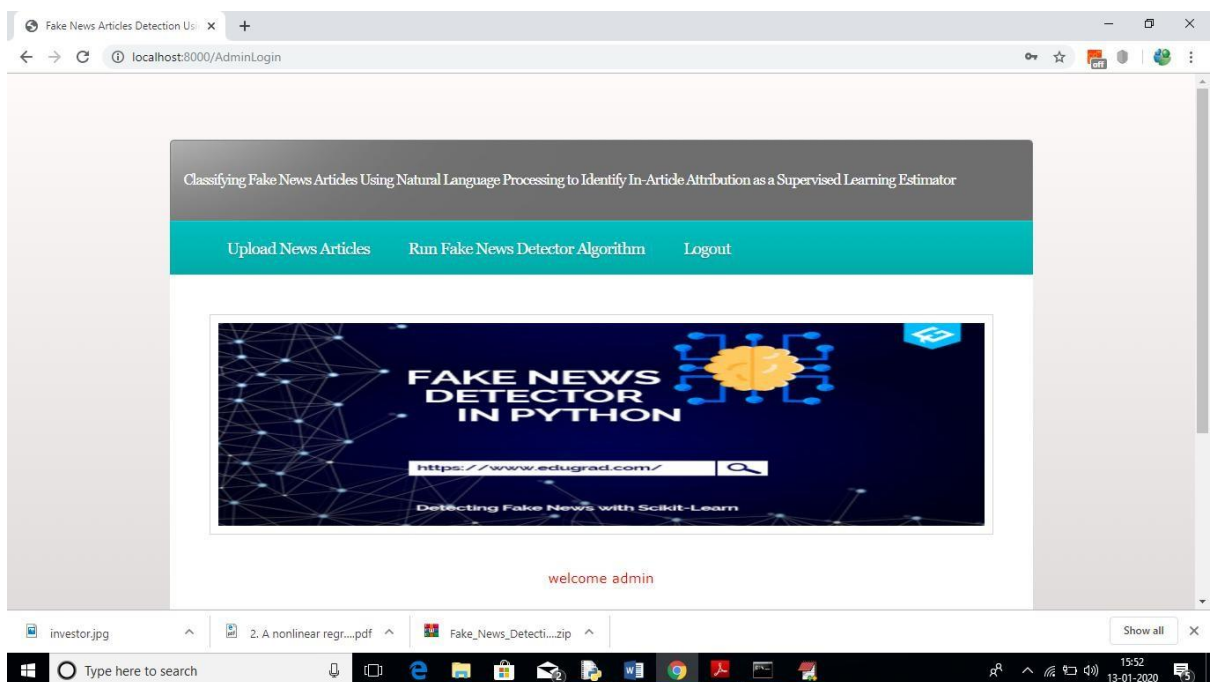
Screen shots



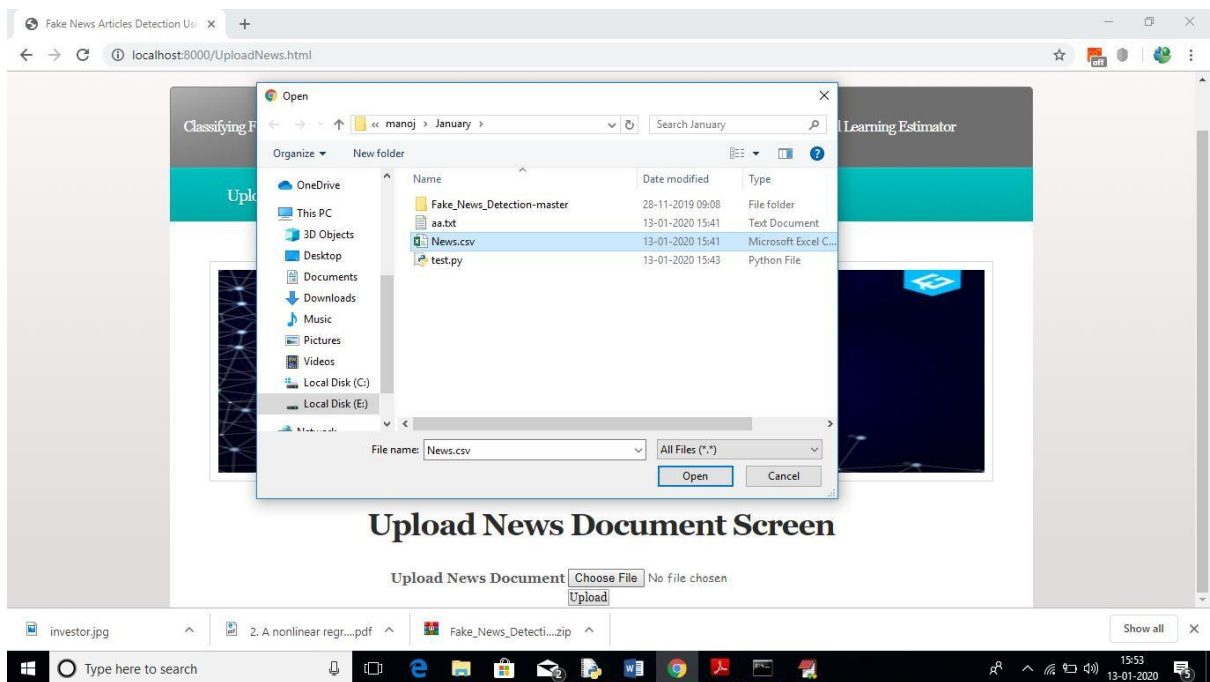
In above screen click on 'User' link to get below screen



In above screen enter username and password as ‘admin’ and then click on ‘Login’ button to get below screen

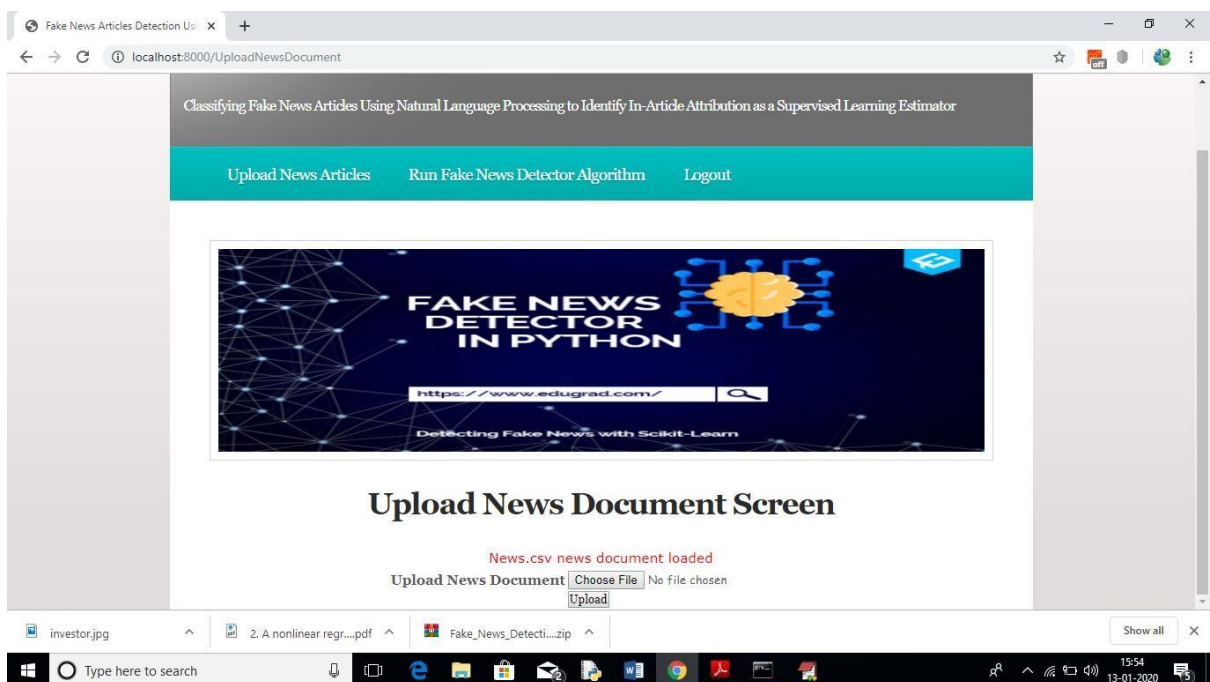


In above screen click on ‘Upload News Articles’ link to upload news document



Upload News Document Screen

In above screen I am uploading 'News.csv' file which contains 150 news paragraphs. After uploading news will get below screen



Upload News Document Screen

In above screen news file uploaded successfully, now click on 'Run Fake News Detector Algorithm' link to calculate Fake News Detection algorithm score and based on score and naïve bayes algorithm we will get result.

News Text	Detection Result	Fake Rank Score
Says the Annies List political group supports third-trimester abortions on demand.	Fake News	0.8333333333333333
When did the decline of coal start? It started when natural gas took off that started to begin in (President George W.) Bushs administration.	Real News	2.142857142857143
"Hillary Clinton agrees with John McCain ""by voting to give George Bush the benefit of the doubt on Iran.""	Real News	3.076923076923077
Health care reform legislation is likely to mandate free sex change surgeries.	Fake News	0.7692307692307693
The economic turnaround started at the end of my term.	Real News	0.9090909090909092
The Chicago Bears have had more starting quarterbacks in the last 10 years than the total number of tenured (UW) faculty fired during the last two decades.	Real News	1.3333333333333333
Jim Dunnam has not lived in the district he represents for years now.	Real News	2.142857142857143
"I'm the only person on this stage who has worked actively just last year passing, along with Russ Feingold, some of the toughest ethics reform since Watergate."	Real News	1.5151515151515151
"However, it took \$19.5 million in Oregon Lottery funds for the Port of Newport to eventually land the new NOAA Marine Operations Center-Pacific."	Real News	2.142857142857143
Says GOP primary opponents Glenn Grothman and Joe Leibham cast a compromise vote that cost \$788 million in higher electricity costs.	Real News	2.1739130434782608
"For the first time in history, the share of the national popular vote margin is smaller than the Latino vote margin."	Fake News	0.8
"Since 2000, nearly 12 million Americans have slipped out of the middle class and into poverty."	Real News	1.5
"When Mitt Romney was governor of Massachusetts, we didnt just slow the rate of growth of our government, we actually cut it."	Real News	2.2222222222222223
The economy bled \$24 billion due to the government shutdown.	Fake News	0.8333333333333333
Most of the (Affordable Care Act) has already in some sense been waived or otherwise suspended.	Real News	2.1052631578947367
"In this last election in November, ... 63 percent of the American people chose not to vote, ... 80 percent of young people, (and) 75 percent of low-income workers chose not to vote."	Real News	0.975609756097561

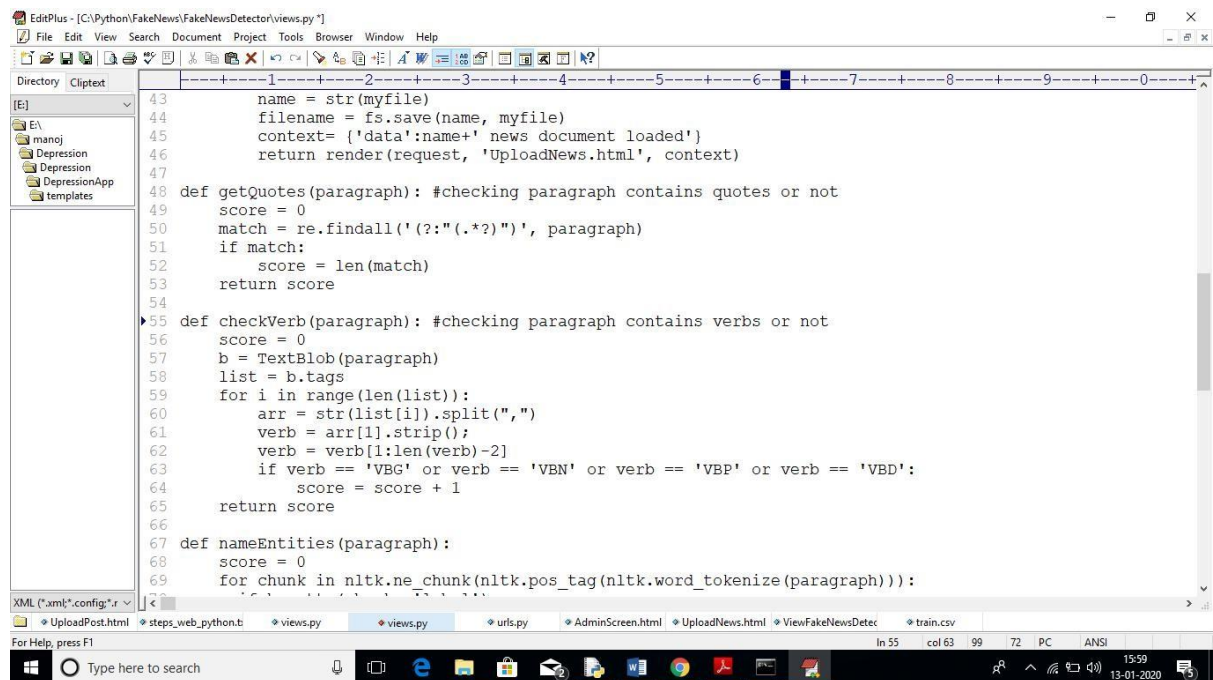
In above screen first column contains news text and second column is the result value as 'fake or real' and third column contains score. If score greater > 0.90 then I am considering news as REAL otherwise fake.

Some neighborhood schools are closing.	Real News	3.333333333333333
He told gay organizers in Massachusetts he would be a stronger advocate for special rights than even Ted Kennedy.	Real News	1.5
"The years that I was speaker, the Florida House consistently offered leaner budgets than the governor offered."	Real News	2.380952380952381
"We are already almost halfway to our 2010 goal of creating 700,000 new jobs in seven years."	Real News	1.5
Says the U.S. Supreme Court found that Social Security is not guaranteed.	Real News	3.8461538461538463
Says Michael Bennet wants to close Guantanamo Bay prison and bring terrorists right here to Colorado.	Real News	2.6666666666666665
Oregonians have an amazing no-cost way to fight abortion with free political donations	Fake News	0.7692307692307693
"The president said hes going to bring in 250,000 (Syrian and Iraqi) refugees into this country."	Real News	2.380952380952381
"Research shows that a vast majority of arriving immigrants today come here because they believe that government is the source of prosperity, and thats what they support."	Real News	1.6129032258064515
Newt Gingrichs immigration plan offers a new doorway to amnesty.	Real News	1.8181818181818183
Mr. Caprio is a career politician who has never worked in the private sector.	Real News	2.0
"In Rhode Island, 9 percent of workers use the states temporary disability insurance program each year while in New Jersey, the rate is only 3 percent."	Real News	1.2903225806451613
"In just 17 years, spending for Social Security, federal health care and interest on the debt will exceed ALL tax revenue!"	Fake News	0.7692307692307693
President Obama took more money from Wall Street in the 2008 campaign than anybody ever had.	Real News	2.3529411764705883
Donald Trump has said nuclear proliferation is OK.	Real News	3.333333333333333
"Hillary Clinton has taken over \$800,000 from lobbyists."	Real News	2.5
Barack Obama has never even worked in business.	Real News	3.333333333333333
Says the Arizona immigration law expressly bans racial profiling.	Real News	1.0
Says Gov. Rick Perry has been begging for the federal government to send the Coast Guard to patrol two lakes on the U.S.-Mexico border.	Real News	1.0230769230769231
"On the VA: Over 300,000 veterans have died waiting for care."	Real News	2.6666666666666665

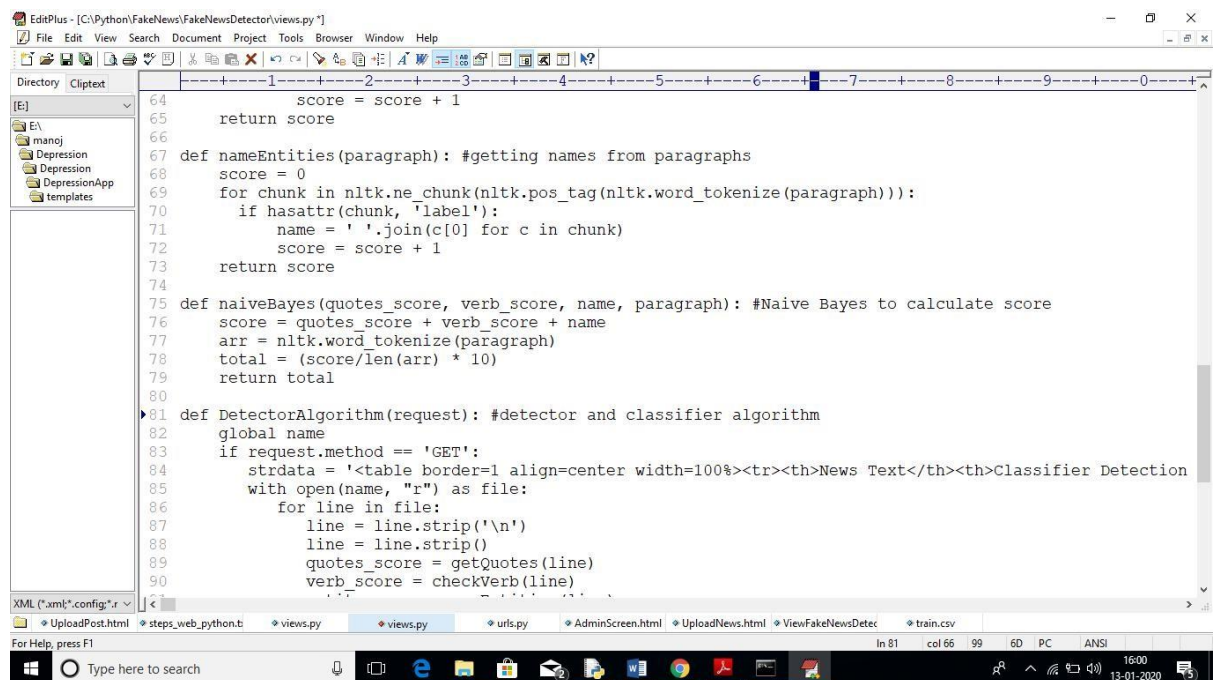
For all 150 news text articles we got result as fake or real.

See below screen shots of code calculating quotes, name entity and verbs from

news paragraphs



```
43     name = str(myfile)
44     filename = fs.save(name, myfile)
45     context= {'data':name+' news document loaded'}
46     return render(request, 'UploadNews.html', context)
47
48 def getQuotes(paragraph): #checking paragraph contains quotes or not
49     score = 0
50     match = re.findall('(?:".*?")', paragraph)
51     if match:
52         score = len(match)
53     return score
54
55 def checkVerb(paragraph): #checking paragraph contains verbs or not
56     score = 0
57     b = TextBlob(paragraph)
58     list = b.tags
59     for i in range(len(list)):
60         arr = str(list[i]).split(",")
61         verb = arr[1].strip()
62         verb = verb[1:len(verb)-2]
63         if verb == 'VBG' or verb == 'VBN' or verb == 'VBP' or verb == 'VBD':
64             score = score + 1
65     return score
66
67 def nameEntities(paragraph):
68     score = 0
69     for chunk in nltk.ne_chunk(nltk.pos_tag(nltk.word_tokenize(paragraph))):
```



```
64         score = score + 1
65     return score
66
67 def nameEntities(paragraph): #getting names from paragraphs
68     score = 0
69     for chunk in nltk.ne_chunk(nltk.pos_tag(nltk.word_tokenize(paragraph))):
70         if hasattr(chunk, 'label'):
71             name = ' '.join(c[0] for c in chunk)
72             score = score + 1
73     return score
74
75 def naiveBayes(quotes_score, verb_score, name, paragraph): #Naive Bayes to calculate score
76     score = quotes_score + verb_score + name
77     arr = nltk.word_tokenize(paragraph)
78     total = (score/len(arr) * 10)
79     return total
80
81 def DetectorAlgorithm(request): #detector and classifier algorithm
82     global name
83     if request.method == 'GET':
84         strdata = '<table border=1 align=center width=100%><tr><th>News Text</th><th>Classifier Detection'
85         with open(name, "r") as file:
86             for line in file:
87                 line = line.strip('\n')
88                 line = line.strip()
89                 quotes_score = getQuotes(line)
90                 verb_score = checkVerb(line)
```

Above code you can see inside ‘Views.py’ program

CHAPTER - 8

SYSTEM TESTING

8.1. TESTING OF PRODUCT

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the

Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

➤ **Unit testing**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. It is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

➤ **Integration testing**

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

➤ **Functional test**

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

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

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

8.2 SYSTEM TESTING

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

White Box Testing

White Box Testing is a testing in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements

document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

8.3. SOFTWARE TESTING

Unit Testing:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

Integration Testing

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

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

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

Acceptance Testing

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

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

CHAPTER- 9

CONCLUSION

9.1. CONCLUSION

This project presented the results of a study that produced a limited fake news detection system. The work presented herein is novel in this topic domain in that it demonstrates the results of a full-spectrum project that started with qualitative observations and resulted in a working quantitative model. The work presented in this project is also promising, because it demonstrates a relatively effective level of machine learning classification for large fake news documents with only one extraction feature. Finally, additional research and work to identify and build additional fake news classification grammars is ongoing and should yield a more refined classification scheme for both fake news and direct quotes.

9.2. FUTURE WORK

The future of fake news detection is expected to witness several promising developments as researchers and technologists strive to enhance the effectiveness of misinformation mitigation efforts. Advanced natural language processing (NLP) techniques will likely be at the forefront, incorporating deep learning models, such as transformer architectures, to improve the nuanced understanding of language and context. Additionally, the integration of cross-modal analysis, which involves examining both textual and multimedia content, could become more prevalent to address the growing sophistication of fake news, which often involves manipulated images and videos.

APPENDICES

1.SAMPLE CODING

MAIN CODE

```
"""Django's command-line utility for administrative tasks."""
import os
import sys

def main():
    os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'FakeNews.settings')
    try:
        from django.core.management import execute_from_command_line
    except ImportError as exc:
        raise ImportError(
            "Couldn't import Django. Are you sure it's installed and "
            "available on your PYTHONPATH environment variable? Did you "
            "forget to activate a virtual environment?"
        ) from exc
    execute_from_command_line(sys.argv)

if __name__ == '__main__':
    main()
```

#VIWES OF THE CODE

```
from django.shortcuts import render
from django.shortcuts import render
from django.template import RequestContext
from django.contrib import messages
from django.http import HttpResponseRedirect
from django.conf import settings
from django.core.files.storage import FileSystemStorage
from textblob import TextBlob
import re
```



```

import nltk

global name

def index(request):
    if request.method == 'GET':
        return render(request, 'index.html', {})

def Login(request):
    if request.method == 'GET':
        return render(request, 'Login.html', {})

def UploadNews(request):
    if request.method == 'GET':
        return render(request, 'UploadNews.html', {})

def AdminLogin(request):
    if request.method == 'POST':
        username = request.POST.get('t1', False)
        password = request.POST.get('t2', False)
        if username == 'admin' and password == 'admin':
            context= {'data': 'welcome '+username}
            return render(request, 'AdminScreen.html', context)
        else:
            context= {'data': 'login failed'}
            return render(request, 'Login.html', context)

def UploadNewsDocument(request):
    global name
    if request.method == 'POST' and request.FILES['t1']:
        output = "
        myfile = request.FILES['t1']

```

```

    fs = FileSystemStorage()
    name = str(myfile)
    filename = fs.save(name, myfile)
    context= {'data':name+' news document loaded'}
    return render(request, 'UploadNews.html', context)

def getQuotes(paragraph): #checking paragraph contains quotes or not
    score = 0
    match = re.findall('(?:"(.*?)"')', paragraph)
    if match:
        score = len(match)
    return score

def checkVerb(paragraph): #checking paragraph contains verbs or not
    score = 0
    b = TextBlob(paragraph)
    list = b.tags
    for i in range(len(list)):
        arr = str(list[i]).split(",")
        verb = arr[1].strip();
        verb = verb[1:len(verb)-2]
        if verb == 'VBG' or verb == 'VBN' or verb == 'VBP' or verb == 'VBD':
            score = score + 1
    return score

def nameEntities(paragraph): #getting names from paragraphs
    score = 0
    for chunk in nltk.ne_chunk(nltk.pos_tag(nltk.word_tokenize(paragraph))):
        if hasattr(chunk, 'label'):
            name = ' '.join(c[0] for c in chunk)
            score = score + 1
    return score

```

```

def naiveBayes(quotes_score, verb_score, name, paragraph): #Naive Bayes to calculate score
    score = quotes_score + verb_score + name
    arr = nltk.word_tokenize(paragraph)
    total = (score/len(arr) * 10)
    return total

def DetectorAlgorithm(request): #detector and classifier algorithm
    global name
    if request.method == 'GET':
        strdata = '<table border=1 align=center width=100%><tr><th>News
Text</th><th>Classifier Detection Result</th><th>Fake Rank Score</th></tr><tr>'
        with open(name, "r") as file:
            for line in file:
                line = line.strip('\n')
                line = line.strip()
                quotes_score = getQuotes(line)
                verb_score = checkVerb(line)
                entity_name = nameEntities(line)
                score = naiveBayes(quotes_score, verb_score, entity_name, line)
                if score > 0.90:
                    strdata+='<td>'+line+'</td><td>Real News</td><td>'+str(score)+'</td></tr>'
                else:
                    strdata+='<td>'+line+'</td><td>Fake News</td><td>'+str(score)+'</td></tr>'

        context= {'data':strdata}
        return render(request, 'ViewFakeNewsDetector.html', context)
# URL PATH
from django.urls import path

from . import views

```

```

urlpatterns = [path("index.html", views.index, name="index"),
                path("Login.html", views.Login, name="Login"),
                path("AdminLogin", views.AdminLogin, name="AdminLogin"),
                path("UploadNews.html", views.UploadNews, name="UploadNews"),
                path("UploadNewsDocument", views.UploadNewsDocument,
name="UploadNewsDocument"),
                path("DetectorAlgorithm", views.DetectorAlgorithm,
name="DetectorAlgorithm"),
]

```

INDEX PAGE

```

{% load static %}

<html>
<head>
<title>Fake News Articles Detection Using NLP</title>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<link href="{% static 'style.css' %}" rel="stylesheet" type="text/css" media="screen" />
</head>
<body>
<div id="wrapper">
    <div id="header">
        <div id="logo">
            <h1>Classifying Fake News Articles Using Natural
Language Processing to Identify In-Article
Attribution as a Supervised Learning Estimator</h1>
        </div>
        <div id="slogan">

        </div>
    </div>
    <div id="menu">
        <ul>
<li><a href="{% url 'index' %}">Home</a></li>

```

```

<li><a href="{ % url 'Login' % }">User</a></li>

</ul>

<br class="clearfix" />

</div>

<div id="splash">

    </div>
<br/><p align="justify"><font size="3" style="font-family: Comic Sans MS">
Abstract-Intentionally deceptive content presented under the guise of legitimate journalism is
a worldwide information accuracy and integrity problem that affects opinion forming,
decision making, and voting patterns. Most so-called 'fake news' is initially distributed over
social media conduits like Facebook and Twitter and later finds its way onto mainstream
media platforms such as traditional television and radio news.</p>
<p>The fake news stories that are initially seeded over social media platforms share key
linguistic characteristics such as making excessive use of unsubstantiated hyperbole and non-
attributed quoted content. In this paper, the results of a fake news identification study that
documents the performance of a fake news classifier are presented. The Textblob,Natural
Language, and SciPy Toolkits were used to develop a novel fake news detector that uses
quoted attribution in a Bayesian machine learning system as a key feature to estimate the
likelihood that a news article is fake.</p>
</body>
</html

```

REFERENCES

- [1] H. Liu, T. Mei, J. Luo, H. Li, and S. Li, “Finding perfect rendezvous on the go: accurate mobile visual localization and its applications to routing,” in Proceedings of the 20th ACM international conference on Multimedia. ACM, 2012, pp. 9–18.
- [2] J. Li, X. Qian, Y. Y. Tang, L. Yang, and T. Mei, “Gps estimation for places of interest from social users’ uploaded photos,” IEEE Transactions on Multimedia, vol. 15, no. 8, pp. 2058–2071, 2013.
- [3] S. Jiang, X. Qian, J. Shen, Y. Fu, and T. Mei, “Author topic model based collaborative filtering for personalized poi recommendation,” IEEE Transactions on Multimedia, vol. 17, no. 6, pp. 907–918, 2015.
- [4] J. Sang, T. Mei, and C. Sun, J.T.and Xu, “Probabilistic sequential pois recommendation via check-in data,” in Proceedings of ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. ACM, 2012.
- [5] Y. Zheng, L. Zhang, Z. Ma, X. Xie, and W. Ma, “Recommending friends and locations based on individual location history,” ACM Transactions on the Web, vol. 5, no. 1, p. 5, 2011.
- [6] H. Gao, J. Tang, X. Hu, and H. Liu, “Content-aware point of interest recommendation on location-based social networks,” in Proceedings of 29th International Conference on AAAI. AAAI, 2015.
- [7] Q. Yuan, G. Cong, and A. Sun, “Graph-based point-of-interest recommendation with geographical and temporal influences,” in Proceedings of the 23rd ACM International Conference on Information and Knowledge Management. ACM, 2014, pp. 659–668.
- [8] H. Yin, C. Wang, N. Yu, and L. Zhang, “Trip mining and recommendation from geo-tagged photos,” in IEEE International Conference on Multimedia and Expo Workshops. IEEE, 2012, pp. 540–545.
- [9] Y. Gao, J. Tang, R. Hong, Q. Dai, T. Chua, and R. Jain, “W2go: a travel guidance system by automatic landmark ranking,” in Proceedings of the international conference on Multimedia. ACM, 2010, pp. 123–132.
- [10] X. Qian, Y. Zhao, and J. Han, “Image location estimation by salient region matching,” IEEE Transactions on Image Processing, vol. 24, no. 11, pp. 4348–4358, 2015.
- [11] H. Kori, S. Hattori, T. Tezuka, and K. Tanaka, “Automatic generation of multimedia

tour guide from local blogs,” *Advances in Multimedia Modeling*, pp. 690–699, 2006.

[12] T. Kurashima, T. Tezuka, and K. Tanaka, “Mining and visualizing local experiences from blog entries,” in *Database and Expert Systems Applications*. Springer, 2006, pp. 213–222.

[13] Y. Shi, P. Serdyukov, A. Hanjalic, and M. Larson, “Personalized landmark recommendation based on geo-tags from photo sharing sites,” *ICWSM*, vol. 11, pp. 622–625, 2011.

[14] M. Clements, P. Serdyukov, A. de Vries, and M. Reinders, “Personalised travel recommendation based on location co-occurrence,” *arXiv preprint arXiv:1106.5213*, 2011.

[15] X. Lu, C. Wang, J. Yang, Y. Pang, and L. Zhang, “Photo2trip: generating travel routes from geo-tagged photos for trip planning,” in *Proceedings of the international conference on Multimedia*. ACM, 2010, pp. 143–152.

[16] Y. Zheng, L. Zhang, X. Xie, and W. Ma, “Mining interesting locations and travel sequences from gps trajectories,” in *Proceedings of the 18th international conference on World wide web*. ACM, 2009, pp. 791–800.

[17] V. W. Zheng, Y. Zheng, X. Xie, and Q. Yang, “Collaborative location and activity recommendations with gps history data,” in *Proceedings of the 19th international conference on World wide web*. ACM, 2010, pp. 1029–1038.

[18] N. J. Yuan, Y. Zheng, X. Xie, Y. Wang, K. Zheng, and H. Xiong, “Discovering urban functional zones using latent activity trajectories,” *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 3, pp. 712–725, 2015. [Online]. Available: <http://dx.doi.org/10.1109/TKDE.2014.2345405>

[19] J. Liu, Z. Huang, L. Chen, H. T. Shen, and Z. Yan, “Discovering areas of interest with geo-tagged images and check-ins,” in *Proceedings of the 20th ACM international conference on Multimedia*. ACM, 2012, pp. 589–598.

[20] Y. Pang, Q. Hao, Y. Yuan, T. Hu, R. Cai, and L. Zhang, “Summarizing tourist destinations by mining user-generated travelogues and photos,” *Computer Vision and Image Understanding*, vol. 115, no. 3, pp. 352–363, 2011.

[21] L. Cao, J. Luo, A. Gallagher, X. Jin, J. Han, and T. Huang, “A worldwide tourism recommendation system based on geo-tagged web photos,” in *IEEE International Conference on Acoustics Speech and Signal Processing*. IEEE, 2010, pp. 2274–2277.

[22] H. Huang and G. Gartner, “Using trajectories for collaborative filtering-based poi

recommendation,” *International Journal of Data Mining, Modelling and Management*, vol. 6, no. 4, pp. 333–346, 2014.

[23] C. Zhang and K. Wang, “Poi recommendation through crossregion collaborative filtering,” *Knowledge and Information Systems*, pp. 1–19, 2015.

[24] A. Majid, L. Chen, G. Chen, H. Mirza, and I. Hussain, “Gothere: travel suggestions using geotagged photos,” in *Proceedings of the 21st international conference companion on World Wide Web*. ACM, 2012, pp. 577–578.

[25] C. Cheng, H. Yang, M. R. Lyu, and I. King, “Where you like to go next: Successive point-of-interest recommendation,” in *IJCAI*, 2013.