DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING,
SHARDA SCHOOL OF ENGINEERING AND TECHNOLOGY,
SHARDA UNIVERSITY, GREATER NOIDA

Fake News Classifier Using NLP

A project submitted
in partial fulfillment of the requirements for the degree of
Bachelor of Technology in Computer Science and Engineering

by
AMAN MANDAL (2019004409)

SHAMSER ALAM (2019004281)

Supervised by:
DR. SAHIL KANSAL, ASST. PROF(CSE)

May, 2023

CERTIFICATE

This is to certify that the report entitled" **Fake News Classifier Using NLP** " submitted by "**AMAN MANDAL (2019004409) AND SHAMSER ALAM (2019004281)"** to Sharda University, towards the fulfillment of requirements of the degree of **Bachelor of Technology** is record of bonafide final year Project work carried out by him/her in the Department of Computer Science and Engineering, School of Engineering and Technology, Sharda University.

The results/findings contained in this Project have not been submitted in part or full to any other University/Institute for award of any other Degree/Diploma.

Signature of the Guide
Name: Dr Sahil kansal
Designation: Assistant professor

Signature of Head of Department
Name: Prof.(Dr.)Nitin Rakesh
Place: Sharda University
Date:

Signature of External Examiner
Date:

ACKNOWLEDGEMENT

A major project is a golden opportunity for learning and self-development. We consider our self very lucky and honored to have so many wonderful people lead us through in completion of this project.

First and foremost we would like to thank Dr. Nitin Rakesh, HOD, CSE who gave us an opportunity to undertake this project.

Our grateful thanks to Dr. Sahil Kansal for his guidance in my project work. Dr. Sahil Kansal who in spite of being extraordinarily busy with academics, took time out to hear, guide and keep us on the correct path. We do not know where we would have been without his help.

CSE department monitored our progress and arranged all facilities to make life easier. We choose this moment to acknowledge their contribution gratefully.

Name and signature of Students

AMAN MANDAL (2019004409)

SHAMSER ALAM (2019004281)

ABSTRACT

Fake news has emerged as a big issue in today's culture. Fake news has the power to affect people's beliefs and facts, and it is the most deadly weapon for influencing society.

The proposed research employs NLP algorithms to detect 'fake news,' or deceptive news reports obtained from untrustworthy sources. Fake news may be recognised by developing a model based on the Decision Tree Classifier technique. The data science community has reacted by taking measures to address the issue. It is impossible to correctly evaluate whether a piece of news is true or false. As a result, the proposed research employs datasets trained using the count vectorizer approach for the identification of fake news, and its accuracy will be evaluated using Natural Language Processing algorithms.

Keywords : Fake, Real,Tf-Idf Vectorizer, Test-Train split, and Decision Tree Classifier.

CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Problem Statement

Social media makes it simpler and more accessible to connect and communicate with others, improving the quality of interpersonal interactions. However, the standard of the interpersonal relationship is under jeopardy. People have made social media programmes important to their lives as the internet has become intertwined with daily life. The variety of social media usage continues to rise as technology advances, making the globe a domestics world where individuals may freely communicate with anybody from anywhere. Interaction with strangers all around the world has made it easier for hackers to obtain individuals' essential information and commit criminality. As a result, social media has a significant influence on interpersonal relationships when individuals rely heavily on it and allow it to dictate communication.

The spread of fake news and incorrect information can eventually lead to confusions and rumours circulating, and victims can be negatively influenced, with one of the worst consequences being suicide. Existing deep learning systems for deception false news detection have focused on online review and public publication through social media. It is challenging to spot false news because it might exist in a variety of patterns, and there has been a significant advancement in NLP frameworks.

1.2 Motivation

- Addressing the issue of misinformation: Fake news has become a major problem in today's society, and it can have serious consequences, such as influencing public opinion and elections, spreading hate and violence, and undermining trust in media and institutions.
- Improving news analysis and verification: With the abundance of news sources and

the speed at which news spreads on social media, it can be challenging for individuals and organizations to verify the accuracy and credibility of news articles. A fake news classifier using NLP can help to automate the process of news analysis and verification, improving the efficiency and accuracy of the task.

- Enhancing information dissemination: Fake news can also have a negative impact on the dissemination of accurate and useful information. By developing a reliable fake news classifier, it can help to increase the quality and reliability of news and information available to the public.
- Advancing NLP and machine learning: The development of a fake news classifier using NLP and machine learning techniques can contribute to the advancement of these fields, as well as help to develop new techniques and approaches for text classification and analysis.
- Providing a tool for researchers and organizations: A fake news classifier can be a useful tool for researchers, media organizations, and social media platforms to study and mitigate the effects of fake news, as well as to improve the quality and credibility of news reporting and analysis.
- Fostering critical thinking and media literacy: By developing a fake news classifier and raising awareness about the problem of fake news, it can help to promote critical thinking and media literacy among individuals and communities, enabling them to better evaluate and interpret news and information.
- Supporting democratic values: By combatting fake news and promoting the dissemination of accurate and reliable information, a fake news classifier can support democratic values such as transparency, accountability, and informed decision-making.
- Encouraging ethical use of technology: The development of a fake news classifier can

also encourage ethical use of technology, by highlighting the importance of using technology for social good and addressing the negative consequences of technology misuse.

- Providing a practical application of NLP: A fake news classifier using NLP techniques can provide a practical and real-world application of NLP, demonstrating its potential to address important societal issues and improve the quality of life.
- Addressing a pressing societal issue: The problem of fake news is a pressing societal issue that affects individuals, organizations, and communities worldwide. By developing a fake news classifier using NLP, it can contribute to addressing this issue and improving the quality and reliability of news and information.

1.3 Project Overview

False material presented as news is referred to as fake news. It is frequently used to harm a person's or entity's reputation or to make money through advertising income. Fake news, which was formerly ubiquitous in print, has grown in popularity

Fake News Classifier is the NLP model based on classification and prediction of the train dataset.

The project involves the following steps:

- Data collection: Collect a large dataset of news articles, both real and fake. This dataset should be diverse and cover a wide range of topics.
- Data preprocessing: Preprocess the data by cleaning and formatting the text. This step involves removing any unwanted characters, stop words, and stemming or lemmatizing the words.

- Feature extraction: Extract meaningful features from the preprocessed text using techniques such as Bag of Words (BoW), Term Frequency-Inverse Document Frequency (TF-IDF), and Word Embeddings.
- Model selection: Select an appropriate machine learning algorithm for the classification task. Possible options include Decision Trees, Random Forests, Support Vector Machines (SVM), and Neural Networks.
- Model training: Train the selected model using the extracted features and labeled data.
- Model evaluation: Evaluate the performance of the trained model using metrics such as accuracy, precision, recall, and F1-score.
- Model deployment: Deploy the trained model as an application or service that can classify news articles in real-time.

1.4 Expected Outcome

A "fake news classifier using NLP" is a system that uses natural language processing (NLP) techniques to identify and classify news articles as either real or fake. The expected outcome of such a system is to improve the accuracy and speed of detecting fake news, thereby reducing the spread of misinformation and improving overall media literacy.

By using machine learning algorithms and NLP techniques, the classifier can analyze the language and structure of a news article to determine its credibility. For example, it might identify patterns of bias or misinformation, analyze the sources cited, or cross-check factual claims with other reputable sources.

The expected benefits of a fake news classifier are numerous. It can help social media platforms, news outlets, and other organizations to quickly identify and flag misleading or false information. It can also be used by individuals to evaluate the credibility of news sources and articles, improving their ability to consume media critically.

However, it is important to note that building an effective fake news classifier is not a simple

task. There are challenges associated with accurately detecting subtle forms of misinformation, and the system must be regularly updated and trained to adapt to new types of fake news. Additionally, there is a risk of bias or censorship if the classifier is not designed and used carefully, so it is important to balance the benefits of the system with the need to protect free speech and diverse perspectives.

1.5 Hardware & Software Specifications

This project can run on commodity hardware. We ran entire project on an Intel I5 processor with 8 GB Ram, 2 GB Nvidia Graphic Processor, It also has 2 cores which runs at 1.7 GHz, 2.1 GHz respectively. First part of the is training phase which takes 10-15 mins of time and the second part is testing part which only takes few seconds to make predictions and calculate accuracy.

HARDWARE REQUIREMENTS:

• RAM: 4 GB

• Storage: 500 GB

• CPU: 2 GHz or faster

• Architecture: 32-bit or 64-bit

SOFTWARE REQUIREMENTS

• Python 3 in Jupyter is used for data pre-processing, model training and prediction.

• Streamlit for web interface used in vs code.

• Operating System: windows 7 and above or Linux based OS or MAC OS.

1.6 Report Outline

Chapter 2 contains a literature survey that provides a summary of a individual paper.

Chapter 3 contains a methodology that provides a functional and non – functional requirement

Chapter 4 contains the results and experimental analysis of the system.

Chapter 5 contains a conclusion about fake news classifier and future work about what you wanted to do in future.

CHAPTER 2

LITERATURE SURVEY

2.1 Introduction

In the world of rapidly increasing technology ,information sharing has become an easy task. There is no doubt that internet has made our lives easier and access to lots of information. This is an evolution in human history, but at the same time it unfocusses the line between true media and maliciously forged media. Today anyone can publish content – credible or not – that can be consumed by the world wide web. Sadly, fake news accumulates a great deal of attention over the internet, especially on social media. People get deceived and don't think twice before circulating such mis-informative pieces to the world. This kind of news vanishes but not without doing the harm it intended to cause. The social media sites like Facebook, Twitter, Whatsapp play a major role in supplying these false news. Many scientists believe that counterfeited news issue may be addressed by means of machine learning and artificial intelligence.

2.2 Existing Methods

[1]This work used machine learning, deep learning, and natural language processing to classify bogus news. To improve accuracy, NLP approaches such as Tokenize, TF-IDF, and Word2vec were used. Binomial logistic regression, naive Bayes classifier, support vector machines, and random forest are the four methods used in traditional ML models. Three algorithms were used in the neural network models: CNN with GlobalMaxpool, CNN with DeepNetwork, and LSTM. Previous works' novel empirical research were to classify labelled fake news, and NLP is combined with conventional machine learning to construct an AI model. We assess seven models on an accessible public dataset based on content-level features to find the best model between standard ML and neural network techniques. We can

observe from the trials that the random forest model achieves great accuracy.

[2] In this article, we explored the societal impact of false news and the issues that arise when it is used.fighting this occurrence, which may be tackled successfully by using automated detection methods We evaluated the available options, potential roadblocks, outcomes produced using various NLP algorithms, and data used in recent research. We also highlighted alternative approaches to the automatic identification of Romanian fake news items and the important topics at stake, drawing on existing international literature and, in some cases, our own relevant research findings.

[3] The goal of this study was to assess, synthesise, compare, and evaluate current studies on fake news.It covers quantitative and qualitative research of fake news as well as detection and intervention strategies As previously said, fake news detection is a machine learning approach to the problem of detecting false news, rumours, and disinformation. The composite classification system, in particular, consists of neural networks composed of traditional classification algorithms that primarily rely on lexical analysis of objects as the major feature for prediction and the usage of external background information.

[4] In this study, we test a false news detector called Fakebox against adversarial techniques such as fact-distortion, subject-object exchange, and cause confusing. Experiments reveal that our assault greatly subverts the model. Similar models based simply on linguistic traits, we predict, will perform far less well in the actual world and are particularly sensitive to tampering assaults. This type of assault is significantly more subtle, as it does not alter the general writing style of news stories, allowing it to elude similarity detection. To genuinely identify disinformation, we believe that multi-source fact comparison and checking must be included into false news detection algorithms.

Simultaneously, the false positive rate increases when it comes to either under-written actual articles or specific themes. The possibility of misclassifying under-written yet actual news

will dampen the excitement of amateur news writers. As a result, we propose that fact-checking be used as a complement to mitigate the harmful impact of false positive judgements.

One method for gathering information about news events is to utilise a crowdsourced knowledge graph, which is dynamically updated by local and knowledgeable individuals. The timely information gathered may then be compared to that retrieved from news items to assist establish a truthfulness label.

Our future work will include creating a visualised interface for news knowledge graph crowdsourcing in order to make work as simple as possible for non-experts and to detect fact-tampering fake news early on. We also intend to investigate the issue of forgery.

Fake news is a growing problem in the age of social media, where it can be easily spread and amplified. Various approaches have been proposed to tackle this issue, and natural language processing (NLP) techniques have gained attention for their ability to automatically identify and classify fake news.

One study conducted by Wang et al. (2018) used a machine learning approach to build a fake news detection system. They used a dataset of real and fake news articles and extracted a set of features including the length of the article, the frequency of punctuation marks, and the sentiment of the text. They then trained various machine learning algorithms including Random Forest, SVM, and Logistic Regression, and evaluated the performance using precision, recall, and F1-score metrics. Their results showed that Random Forest performed the best, achieving an F1-score of 0.93.

Another study by Saha and Saha (2020) used a similar approach but focused on identifying fake news related to the COVID-19 pandemic. They collected a dataset of news articles related to COVID-19 from various sources and used NLP techniques to extract features including word frequency, sentiment, and readability. They then used a Random Forest

algorithm to classify the articles as real or fake, achieving an accuracy of 88.3%.

A more recent study by Hu et al. (2021) used a deep learning approach to build a fake news detection system. They used a dataset of news articles and social media posts and applied a pre-trained language model called BERT (Bidirectional Encoder Representations from Transformers) to extract features. They then trained a deep neural network to classify the articles/posts as real or fake. Their results showed that the BERT-based model outperformed traditional machine learning algorithms, achieving an accuracy of 93.6%.

2.3 Proposed System

The proposed system when subjected to a scenario of a set of news articles , the new articles are categorized as true or fake by the existing data available . This prediction is done by using the relationship between the words used in the article with one another. The proposed system contains a Word2Vec model for finding the relationship between the words and with the obtained information of the existing relations,the new articles are categorized into fake and real news.

The proposed system of a fake news classifier involves using natural language processing (NLP) techniques to automatically identify and classify news articles as either real or fake. The system consists of several steps, including:

- Data collection: Collecting a dataset of news articles that are labeled as real or fake. This dataset will be used to train and test the machine learning model.
- Preprocessing: Cleaning and preprocessing the text data, which may involve removing stop words, stemming, and normalizing the text.
- Feature extraction: Extracting relevant features from the text data using techniques such as bag-of-words, TF-IDF, or word embeddings.
- Model training: Training a machine learning model, such as a decision tree or a

random forest, using the labeled dataset and the extracted features.

- Model evaluation: Evaluating the performance of the model using metrics such as accuracy, precision, recall, and F1-score.
- Deployment: Deploying the trained model in a production environment, where it can be used to automatically classify news articles as real or fake.

2.4 Methods

This section goes through the theories that were utilised in this work. The detailed related works are given in the subsection that follows.

- **Natural Language Processing (NLP)**

Natural Language Processing (NLP) is a subfield of computer science, artificial intelligence, and linguistics that focuses on enabling computers to understand, interpret, and generate human language. It involves programming computers to process and analyze large amounts of natural language data, such as text or speech, and derive meaning from it. NLP uses a combination of techniques from computer science, linguistics, and statistics to develop algorithms and models that can automatically process, analyze, and understand natural language text. Applications of NLP include machine translation, sentiment analysis, speech recognition, text summarization, and many others.

- **Tokenization**

Tokenization is the process of breaking down a text document into individual units, which are usually words or phrases, referred to as tokens. In natural language processing, tokenization is an essential step in text preprocessing, where it is used to transform the raw text into a format that can be further processed. The process involves splitting the text into

individual words, removing punctuation marks and converting everything to lower or upper case, depending on the requirements of the problem at hand. Tokenization is an important step in various NLP tasks, such as text classification, sentiment analysis, machine translation, and information retrieval, among others.

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

Term Frequency-Inverse Document Frequency (TF-IDF) is a commonly used technique in natural language processing for representing text documents as numerical vectors that can be used in machine learning algorithms.

The term frequency (TF) component of TF-IDF measures the frequency of a term (word) in a document and assigns a weight to it based on how often it appears in that document. The idea is that words that appear frequently in a document are more important to that document's content than words that appear infrequently.

The inverse document frequency (IDF) component of TF-IDF measures how rare or common a term is across all documents in a corpus. Words that appear frequently in many documents are considered less important, while words that appear in only a few documents are considered more important.

TF-IDF combines these two measures to produce a weight for each term in each document, which is used to create a numerical representation of the document. This representation can then be used in machine learning algorithms to classify, cluster, or otherwise analyze text documents.

- **Word2Vec**

Word2Vec is a neural network-based technique used in Natural Language Processing (NLP) for generating word embeddings. It is a shallow neural network that takes as input a large

corpus of text and learns to generate dense vector representations, or embeddings, of words in the corpus. The resulting embeddings capture the meaning of the words and their relationships with other words in the corpus. Word2Vec has two main approaches: Continuous Bag of Words (CBOW) and Skip-gram. CBOW predicts the target word based on its surrounding context words, while Skip-gram predicts the context words based on a given target word. The Word2Vec technique has proven to be effective in a wide range of NLP tasks such as text classification, sentiment analysis, and language modeling.

- **Word Cloud**

A word cloud is a visual representation of a group of words, where the size of each word is proportional to its frequency or importance in a given text. It is a way of summarizing text data and highlighting the most frequently used words in a visually appealing and intuitive manner. Typically, the words are arranged randomly or in a way that suggests their relationship to one another, with the most prominent words appearing in larger font sizes and bolder colors. Word clouds can be used for a variety of purposes, such as identifying key themes in a large body of text, visualizing social media conversations, or generating word art for creative purposes.

- **Decision Tree Classifier**

A Decision Tree Classifier is a machine learning algorithm that uses a tree-like model to classify data. It builds a decision tree by recursively splitting the dataset into smaller subsets, with each split based on the value of a chosen attribute or feature. The decision tree is constructed in a top-down manner, starting from the root node and working its way down to the leaf nodes. Each leaf node represents a class label, and the path from the root node to a leaf node represents a decision rule or condition. Decision trees are often used for classification problems because they can handle both categorical and numerical data and are easy to interpret and visualize.

CHAPTER 3

SYSTEM DESIGN AND ANALYSIS

3.1 System Architecture

The proposed system when subjected to a scenario of a set of news articles , the new articles are categorized as true or fake by the existing data available . This prediction is done by using the relationship between the words used in the article with one another. The proposed system contains a Word2Vec model for finding the relationship between the words and with the obtained information of the existing relations , the new articles are categorized into fake and real news.

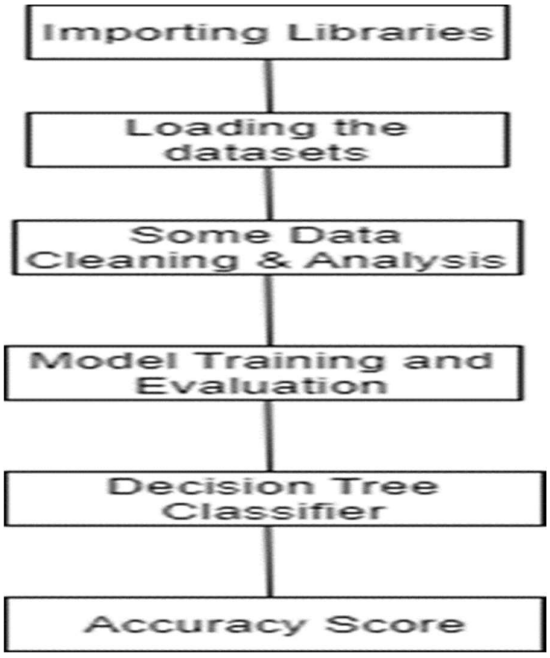


Fig 1:- Workflow of the project for classification

3.1.1 Importing Libraries

To make use of the functions in a module, you'll need to import the module with an import statement.An import statement is made up of the import keyword along with the name of the

module.In a Python file, this will be declared at the top of the code, under any shebang lines or general comments.So, in the Python program file my_rand_int.py we would import the random module to generate random numbers in this manner:

3.1.2 Loading the Datasets

Manually loading a file: This is the first, most popular, and least recommended way to load data as it requires many code parts to read one tuple from the DataFrame. This way comes into the picture when the dataset doesn't have any particular pattern to identify or a specific pattern.

I am using np. load txt: One of the NumPy methods for loading different types of data though it is only supported when we have data in a specific format, i.e., pattern recognizable, unlike the manual way of reading the dataset.

Using np.GenFromTxt: This is another NumPy way to read the data, but this time it is much better than the np. load txt() method recognizes the column header's presence on its own, which the previous one cannot follow. Along with that, it can also detect the right data type for each column.

Using PD.read_csv: Here is the most recommended and widely used method for reading, writing, and manipulating the dataset. It only deals with CSV formatted data, but the support of various parameters makes it a gold mine for data analysts to work with different sorts of data (they should have a specific format).

Using pickle: Last but not least, we will also use a pickle to read the dataset present in the binary format. So far, we chose pickle only to save the model, but the capabilities of this module are much more than that which we will explore in the future article.

3.1.3 Some Data Cleaning And Analysis

Data cleaning is the process of fixing or removing incorrect, corrupted, incorrectly formatted, duplicate, or incomplete data within a dataset. When combining multiple data sources, there are many opportunities for data to be duplicated or mislabeled. If data is incorrect, outcomes and algorithms are unreliable, even though they may look correct. There is no one absolute way to prescribe the exact steps in the data cleaning process because the processes will vary from dataset to dataset. But it is crucial to establish a template for your data cleaning process so you know you are doing it the right way every time.

3.1.3 Model Trainig and Evaluation

Whenever we build Machine Learning models, we need some form of metric to measure the goodness of the model. Bear in mind that the “goodness” of the model could have multiple interpretations, but generally when we speak of it in a Machine Learning context we are talking of the measure of a model's performance on new instances that weren’t a part of the training data.

Determining whether the model being used for a specific task is successful depends on 2 key factors:

Whether the evaluation metric we have selected is the correct one for our problem

If we are following the correct evaluation process

3.1.4 Decision Tree Classifier

Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a

tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.

In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.

The decisions or the test are performed on the basis of features of the given dataset.

It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions.

It is called a decision tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure.

In order to build a tree, we use the CART algorithm, which stands for Classification and Regression Tree algorithm.

A decision tree simply asks a question, and based on the answer (Yes/No), it further split the tree into subtrees.

Data Pre-processing step

Fitting a Decision-Tree algorithm to the Training set

Predicting the test result

Test accuracy of the result(Creation of Confusion matrix)

Visualizing the test set result.

3.1.5 Accuracy Score

Model accuracy is a machine learning classification model performance metric that is defined as the ratio of true positives and true negatives to all positive and negative observations. In other words, accuracy tells us how often we can expect our machine learning model will correctly predict an outcome out of the total number of times it made predictions. For example: Let's assume that you were testing your machine learning model with a dataset of 100 records and that your machine learning model predicted all 90 of those instances correctly. The accuracy metric, in this case, would be: (90/100) = 90%. The accuracy rate is great but it doesn't tell us anything about the errors our machine learning models make on new data we haven't seen before.

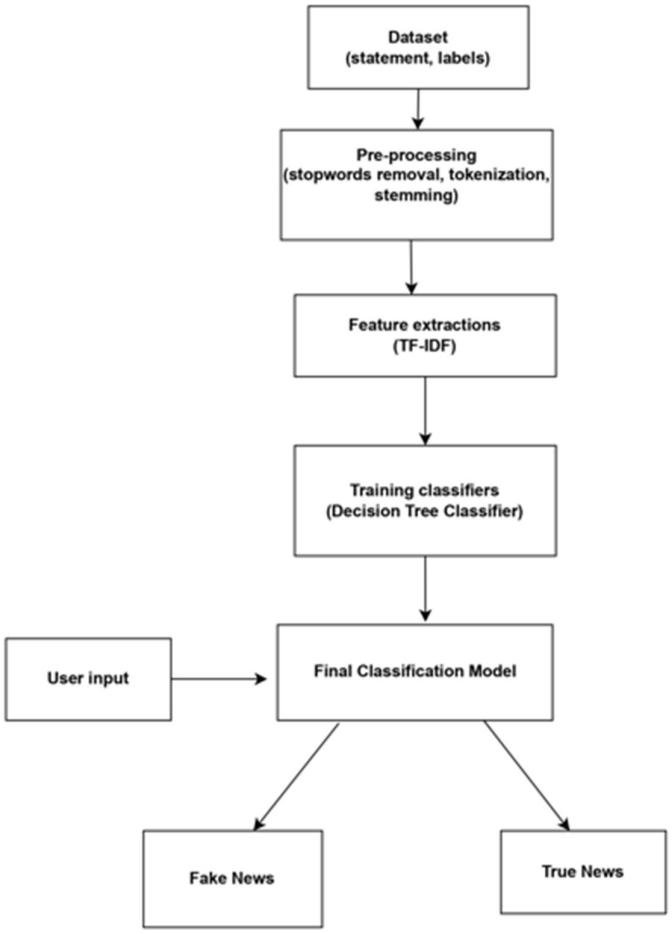


Fig 2:- Workflow for the prediction

3.1.6 Stemming

Stemming is a natural language processing technique used to reduce a word to its base or root form, known as the stem. The stem is obtained by removing the affixes (prefixes, suffixes, and inflections) from the word, so that words with the same stem can be treated as the same word.

For example, the word "running" has the stem "run", and the word "walked" has the stem "walk". By reducing these words to their stems, we can simplify the processing of text data

and improve the efficiency and accuracy of natural language processing tasks such as text classification, information retrieval, and information extraction.

There are several algorithms for stemming, such as the Porter stemming algorithm and the Snowball stemming algorithm. These algorithms use a set of rules and heuristics to identify and remove affixes from words and obtain their stems. However, stemming can sometimes lead to overstemming or understemming, where words are either too aggressively reduced to their stems, or not reduced enough, leading to errors in analysis.

Overall, stemming is a useful technique in natural language processing for simplifying text data and improving processing efficiency. However, it is important to carefully evaluate and tune the stemming algorithm to avoid errors and ensure accuracy in downstream tasks.

3.1.7 Stopwords Removal

Stopwords removal is a common technique used in natural language processing to remove words that are considered to be common, insignificant, or contextually irrelevant from a text corpus. These words are referred to as "stopwords." Examples of stopwords include "a," "an," "the," "in," "of," "is," and "and."

The purpose of removing stopwords is to reduce the size of the text corpus and to improve the efficiency and accuracy of natural language processing tasks such as sentiment analysis, text classification, and topic modeling. Removing stopwords can help to focus on the more meaningful and informative words in a text corpus, which are more likely to be relevant to the task at hand.

There are different approaches to stopwords removal, such as using pre-built lists of stopwords or dynamically generating stopwords lists based on the text corpus. Stopwords can be removed before or after stemming or other preprocessing techniques.

However, it is important to note that removing stopwords may not always be necessary or desirable. In some cases, stopwords may be important for conveying meaning or context in a text, especially in cases where the task requires a more nuanced understanding of language. Therefore, it is important to carefully evaluate the use of stopwords removal in a specific context and task, and to adjust the approach as needed to achieve the desired outcome.

3.1.8 Feature Extraction

Feature extraction is a technique used in natural language processing (NLP) to convert raw text data into a set of numerical features that can be used for machine learning models. The process of feature extraction involves selecting a set of relevant features from a text corpus and converting them into a structured format that can be used for machine learning algorithms.

The choice of features depends on the specific task at hand. For example, in sentiment analysis, the features might include word frequencies, n-grams (sequences of adjacent words), and part-of-speech (POS) tags. In text classification, features might include bag-of-words representations, which count the frequency of individual words in a text corpus, or word embeddings, which represent each word as a dense vector in a high-dimensional space.

3.2 DataSet Preparation

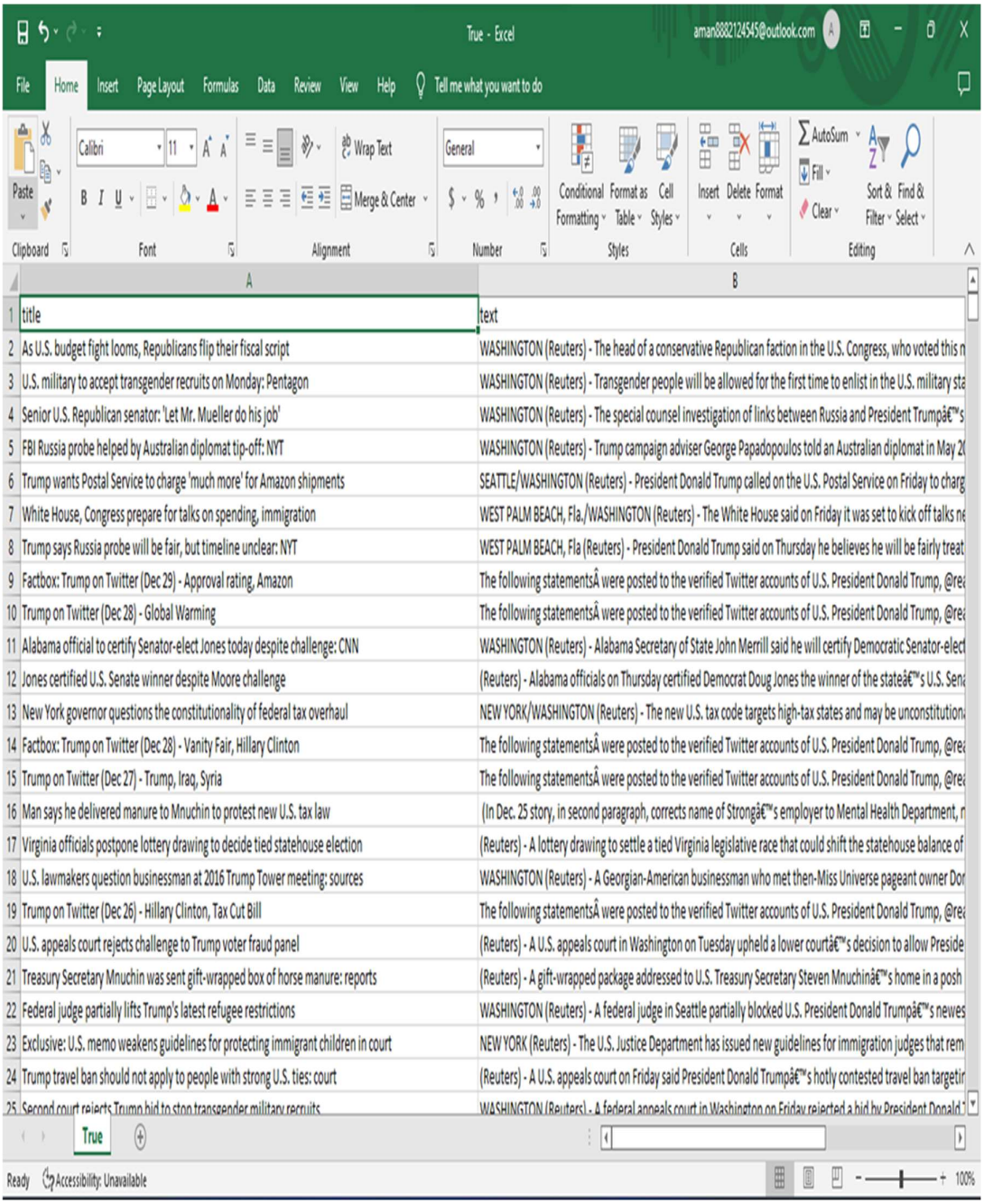


Fig 3:- Dataset for true.csv

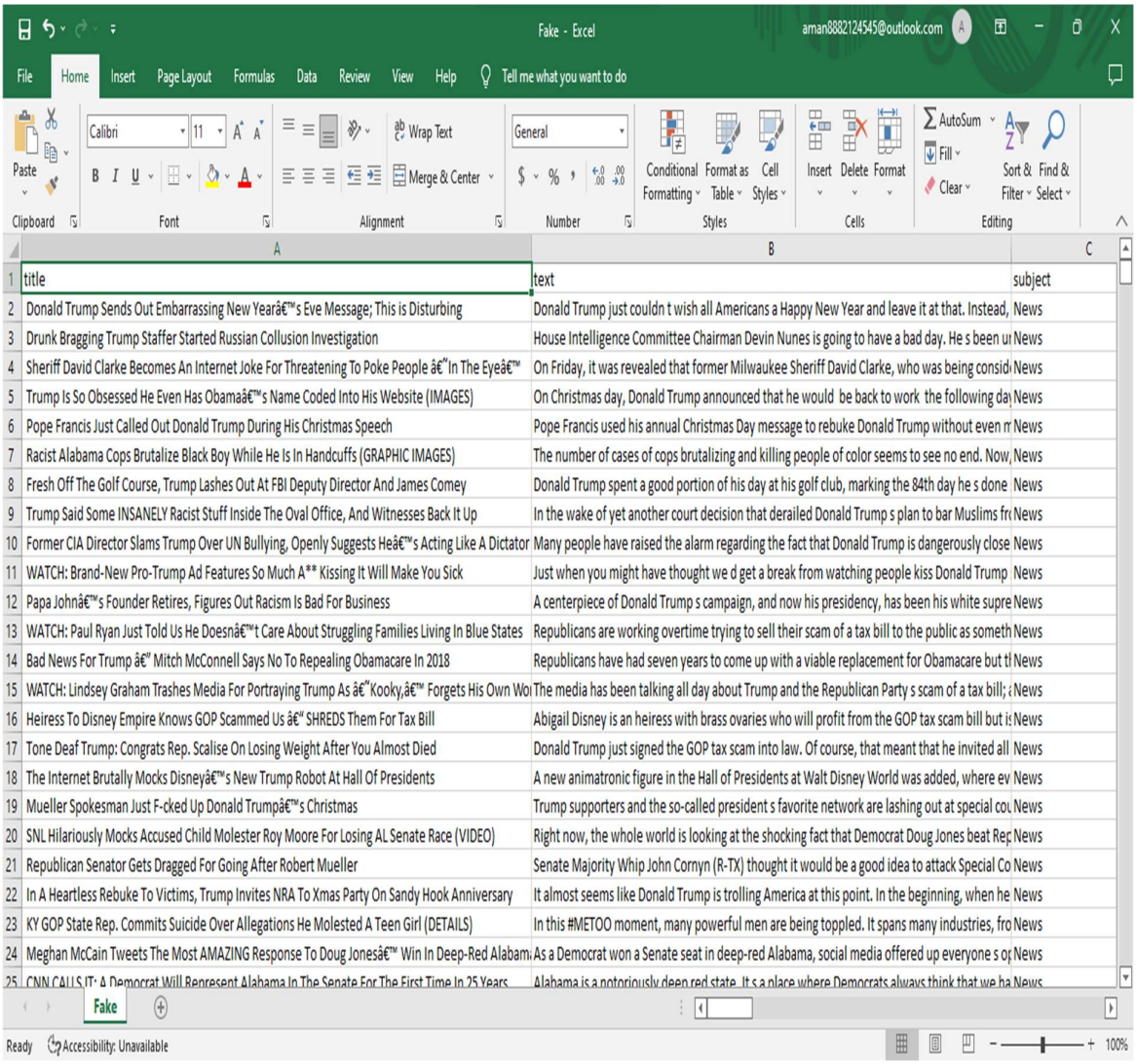
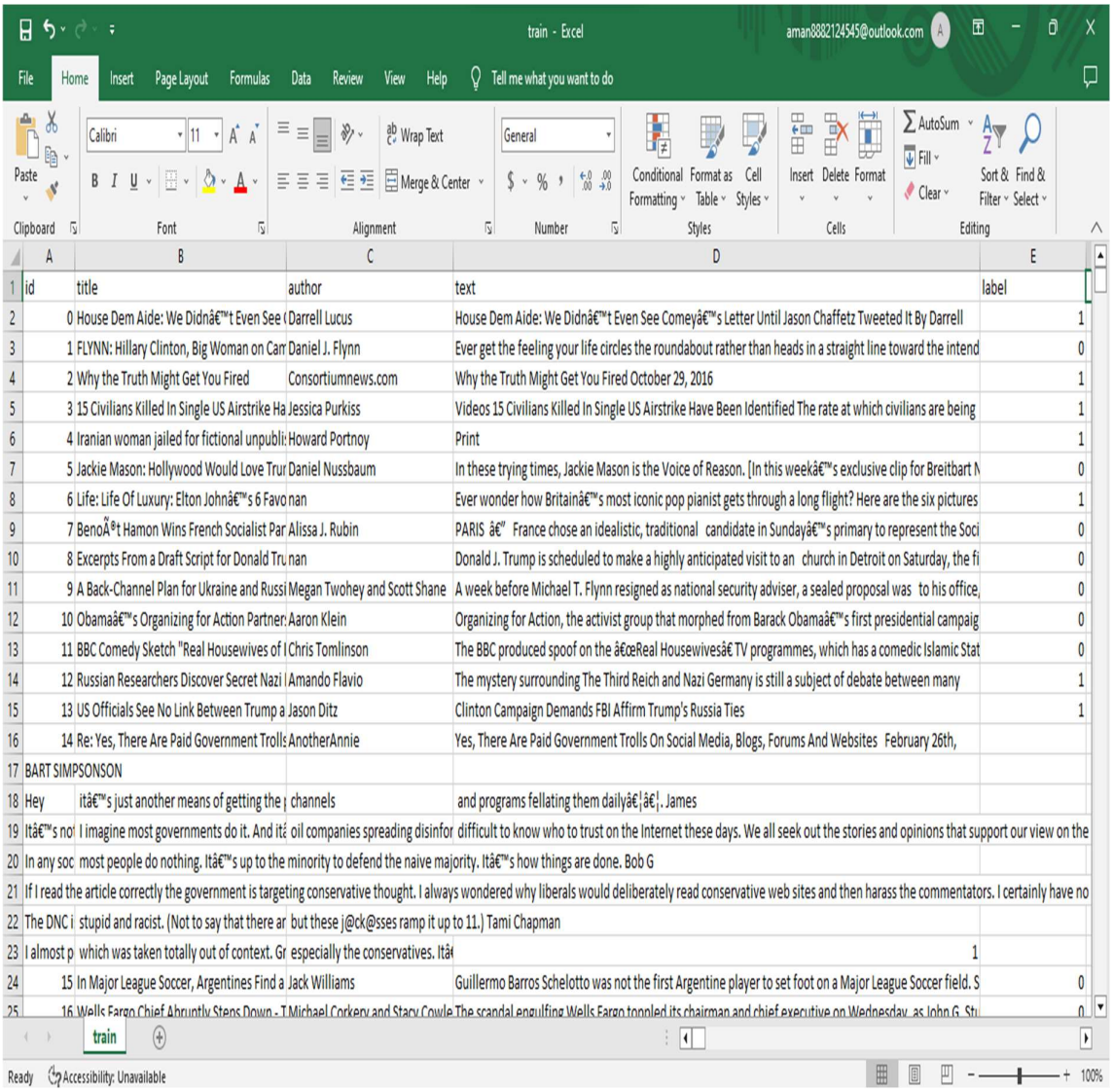


Fig 4: Dataset for fake.csv

train.csv: A full training dataset with the following attributes:

- **id**: unique id for a news article
- **title**: the title of a news article
- **author**: author of the news article
- **text**: the text of the article; could be incomplete
- **label**: a label that marks the article as potentially unreliable
    - 1: unreliable

- 0: reliable



	id	title	author	text	label
2	0	House Dem Aide: We Didnâ€™t Even See (	Darrell Lucus	House Dem Aide: We Didnâ€™t Even See Comeyâ€™s Letter Until Jason Chaffetz Tweeted It By Darrell	1
3	1	FLYNN: Hillary Clinton, Big Woman on Can	Daniel J. Flynn	Ever get the feeling your life circles the roundabout rather than heads in a straight line toward the intend	0
4	2	Why the Truth Might Get You Fired	Consortiumnews.com	Why the Truth Might Get You Fired October 29, 2016	1
5	3	15 Civilians Killed In Single US Airstrike Ha	Jessica Purkiss	Videos 15 Civilians Killed In Single US Airstrike Have Been Identified The rate at which civilians are being	1
6	4	Iranian woman jailed for fictional unpubli	Howard Portnoy	Print	1
7	5	Jackie Mason: Hollywood Would Love Trur	Daniel Nussbaum	In these trying times, Jackie Mason is the Voice of Reason. [In this weekâ€™s exclusive clip for Breitbart N	0
8	6	Life: Life Of Luxury: Elton Johnâ€™s 6 Favo	nan	Ever wonder how Britainâ€™s most iconic pop pianist gets through a long flight? Here are the six pictures	1
9	7	BenoÃ®t Hamon Wins French Socialist Par	Alissa J. Rubin	PARIS â€" France chose an idealistic, traditional candidate in Sundayâ€™s primary to represent the Soci	0
10	8	Excerpts From a Draft Script for Donald Tru	nan	Donald J. Trump is scheduled to make a highly anticipated visit to an church in Detroit on Saturday, the fi	0
11	9	A Back-Channel Plan for Ukraine and Russi	Megan Twohey and Scott Shane	A week before Michael T. Flynn resigned as national security adviser, a sealed proposal was to his office,	0
12	10	Obamaâ€™s Organizing for Action Partner	Aaron Klein	Organizing for Action, the activist group that morphed from Barack Obamaâ€™s first presidential campaig	0
13	11	BBC Comedy Sketch "Real Housewives of I	Chris Tomlinson	The BBC produced spoof on the â€œReal Housewivesâ€ TV programmes, which has a comedic Islamic Stat	0
14	12	Russian Researchers Discover Secret Nazi I	Amando Flavio	The mystery surrounding The Third Reich and Nazi Germany is still a subject of debate between many	1
15	13	US Officials See No Link Between Trump a	Jason Ditz	Clinton Campaign Demands FBI Affirm Trump's Russia Ties	1
16	14	Re: Yes, There Are Paid Government Trolls	AnotherAnnie	Yes, There Are Paid Government Trolls On Social Media, Blogs, Forums And Websites February 26th,	
17	BART SIMPSONSON				
18	Hey	itâ€™s just another means of getting the p	channels	and programs fellating them dailyâ€¦â€¦. James	
19	Itâ€™s not	I imagine most governments do it. And itâ	oil companies spreading disinfor	difficult to know who to trust on the Internet these days. We all seek out the stories and opinions that support our view on the	
20	In any soc	most people do nothing. Itâ€™s up to the	minority to defend the naive majority. Itâ€™s how things are done. Bob G		
21	If I read the	article correctly the government is targeting	conservative thought. I always wondered why liberals would deliberately read conservative web sites and then harass the commentators. I certainly have no		
22	The DNC i	stupid and racist. (Not to say that there ar	but these j@ck@sses ramp it up to 11.) Tami Chapman		
23	I almost p	which was taken totally out of context. Gr	especially the conservatives. Itâ€	1	
24	15	In Major League Soccer, Argentines Find a	Jack Williams	Guillermo Barros Schelotto was not the first Argentine player to set foot on a Major League Soccer field. S	0
25	16	Wells Fargo Chief Abruptly Steps Down - T	Michael Corkery and Stacy Cowle	The scandal engulfing Wells Fargo toppled its chairman and chief executive on Wednesday, as John G. Stu	0

Fig 5:- Dataset for train.csv

3.3 Algorithm For The Proposed System

For Classifiaction

Step 1: Start

Step 2: Import the libraries

Step 3: Input is collected from various sources and prepare a dataset (kaggle)

Step 4: Preprocessing of data is done and dataset is divided into 2 parts training and testing data.

Step 5: Count vectorization technique is used to convert the train data into numericals.

Step 6: Decision Tree Classifier algorithm is used to build the predictive model using the train data .

Step 7: Accuracy is calculated.

For Prediction

Step1: Start

Step2: We will import the libraries

Step3: Load the dataset

Step4: Data prepocessing with the help of stemming

Step5: Converting the textual data to numerical data

Step6: Training the model using Decision tree classifier

Step 7: Integrate it with the streamlit to provide web interface

Step8: Generate results as reliable & unreliable news

CHAPTER 4

RESULTS AND OUTPUT

4.1 Results

4.1.1 For Classification

Fake News Classifier using NLP

Importing libraries

```
In [1]: import pandas as pd
        import numpy as np
        import seaborn as sns
        import matplotlib.pyplot as plt
        %matplotlib inline
        import sklearn
        import string
        import nltk
        import warnings
        warnings.filterwarnings('ignore')
```

Loading the datasets

```
In [2]: fake = pd.read_csv(r'C:\Users\HP\Desktop\majorProject\Fake.csv')

In [3]: true = pd.read_csv(r"C:\Users\HP\Desktop\majorProject\True.csv")
```

Fig 6:- Importing libraries and loading dataset

```
In [4]: fake.head()
```

Out[4]:

	title	text	subject	date
0	Donald Trump Sends Out Embarrassing New Year'...	Donald Trump just couldn t wish all Americans ...	News	December 31, 2017
1	Drunk Bragging Trump Staffer Started Russian ...	House Intelligence Committee Chairman Devin Nu...	News	December 31, 2017
2	Sheriff David Clarke Becomes An Internet Joke...	On Friday, it was revealed that former Milwauk...	News	December 30, 2017
3	Trump Is So Obsessed He Even Has Obama's Name...	On Christmas day, Donald Trump announced that ...	News	December 29, 2017
4	Pope Francis Just Called Out Donald Trump Dur...	Pope Francis used his annual Christmas Day mes...	News	December 25, 2017

```
In [5]: true.head()
```

Out[5]:

	title	text	subject	date
0	As U.S. budget fight looms, Republicans flip t...	WASHINGTON (Reuters) - The head of a conservat...	politicsNews	31-Dec-17
1	U.S. military to accept transgender recruits o...	WASHINGTON (Reuters) - Transgender people will...	politicsNews	29-Dec-17
2	Senior U.S. Republican senator: 'Let Mr. Muell...	WASHINGTON (Reuters) - The special counsel inv...	politicsNews	31-Dec-17
3	FBI Russia probe helped by Australian diplomat...	WASHINGTON (Reuters) - Trump campaign adviser ...	politicsNews	30-Dec-17
4	Trump wants Postal Service to charge 'much mor...	SEATTLE/WASHINGTON (Reuters) - President Donal...	politicsNews	29-Dec-17

```
In [6]: fake['label'] = 'fake'
        true['label'] = 'true'
```

Fig 7:- Head of fake and true dataset

```
In [7]: combined = pd.concat([fake, true], axis=0).reset_index(drop=True)

In [8]: combined.head()
```

Some data cleaning and analysis

```
In [9]: from sklearn.utils import shuffle
        combined = shuffle(combined)

In [10]: combined.drop(['title', 'date'], axis=1, inplace=True)

In [11]: #converting all strings to lower case
         combined['text'] = combined['text'].apply(lambda x : x.lower())
         combined.head()
```

Out[11]:

	text	subject	label
29925	sacramento, calif. (reuters) - a california re...	politicsNews	true
1701	landowners in nebraska are understandably upse...	News	fake
37016	berlin (reuters) - nato should defend sweden a...	worldnews	true
39940	(this october 25 story has been corrected to ...	worldnews	true

```
In [13]: combined['text'] = combined['text'].apply(characters)

In [14]: combined
```

Out[14]:

	text	subject	label
29925	sacramento calif reuters a california republi...	politicsNews	true
1701	landowners in nebraska are understandably upse...	News	fake
37016	berlin reuters nato should defend sweden and ...	worldnews	true
39940	this october 25 story has been corrected to c...	worldnews	true
23845	moscow reuters the kremlin said on monday rus...	politicsNews	true
...	...	...	...
43070	dubai reuters saudi arabia said on wednesday ...	worldnews	true
20494	it s really quite ironic that the guy who has ...	left-news	fake
15575	does anyone care if hillary personally gassed ...	politics	fake
29677	washington reuters the united states sent fou...	politicsNews	true
38884	beijing reuters the white house on wednesday ...	worldnews	true

44898 rows × 3 columns

Fig 8:- Datacleaning and analysis

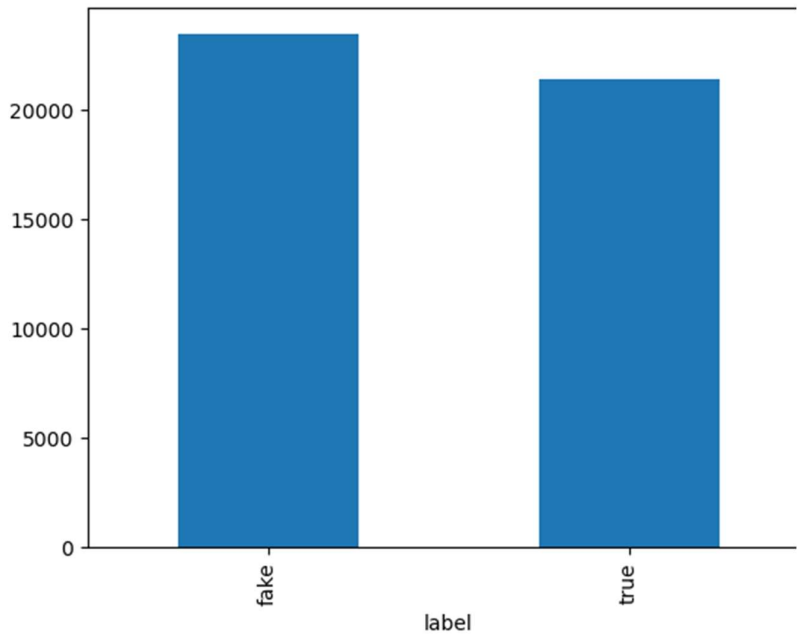


Fig 9:- Bar representation of label true and false

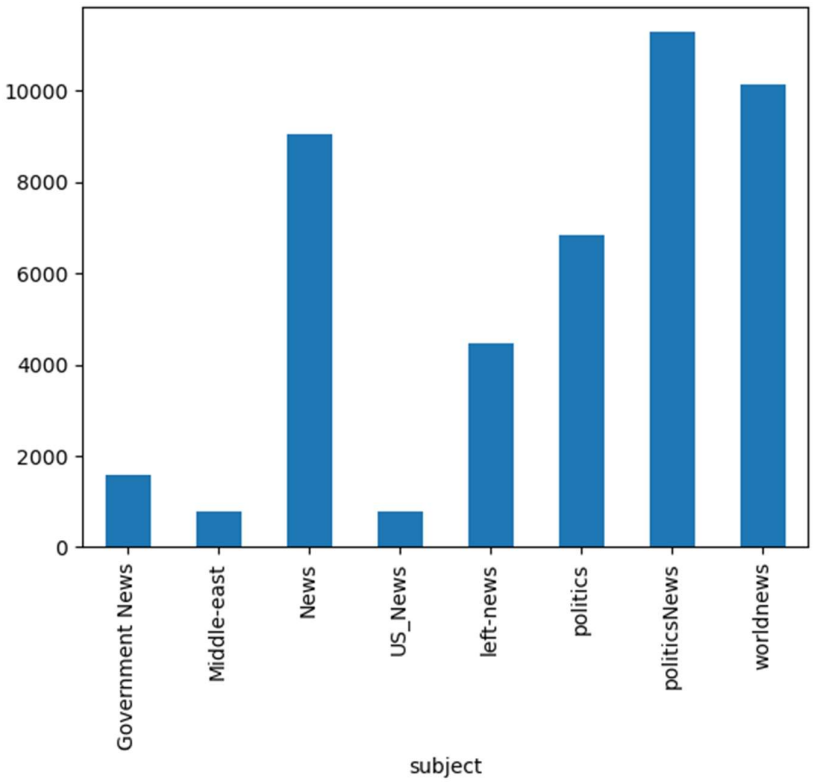


Fig 10:- Bar representation of news subject wise

Fig 11:- Word Cloud for true news

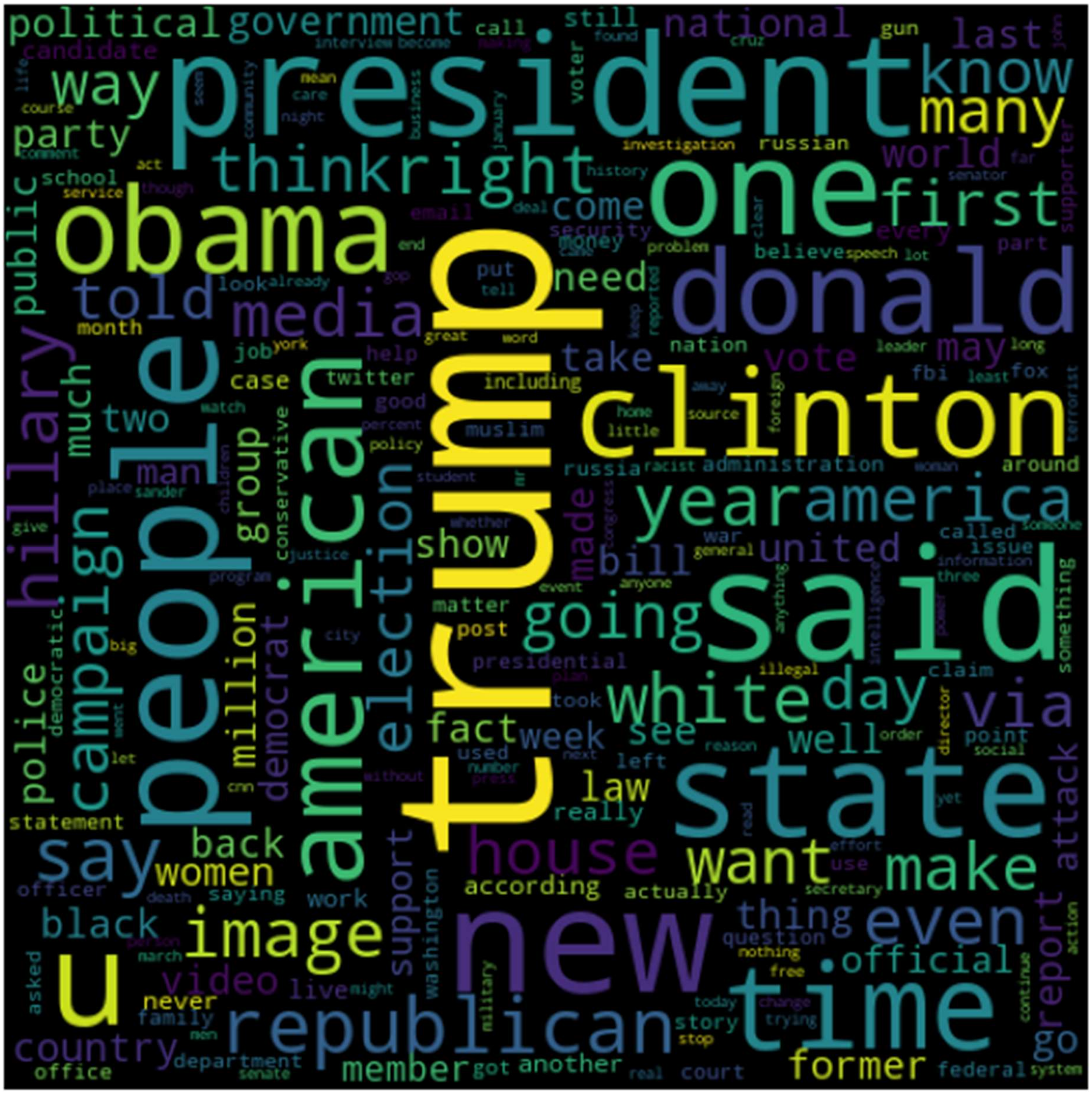


Fig 12:- Wordcloud for true news

Model training and evaluation

```
In [26]: from sklearn.model_selection import train_test_split

In [27]: X_train, X_test, y_train, y_test = train_test_split(combined['text'], combined.label, test_size = 0.2, random_state = 101)

In [28]: X_train.head()

Out[28]: 418      somebody must put truth serum little donnie ch...
         13610    desire push hillary clinton across finish line...
         17372    stats lie one obama hoping reach low informati...
         3168     tuesday donald trump amped stupidity latest at...
         38069    berlin reuters chancellor angela merkel effort...
         Name: text, dtype: object

In [29]: y_train.head()

Out[29]: 418      fake
         13610    fake
         17372    fake
         3168     fake
         38069    true
         Name: label, dtype: object
```

Fig 13:- Model training and Evaluation

We will use Decision Tree Classifier

```
In [31]: from sklearn.tree import DecisionTreeClassifier
         from sklearn.feature_extraction.text import CountVectorizer
         from sklearn.feature_extraction.text import TfidfTransformer
         from sklearn.metrics import classification_report
         from sklearn.metrics import accuracy_score
         from sklearn.metrics import roc_auc_score
         from sklearn.metrics import confusion_matrix

In [32]: from sklearn.pipeline import Pipeline
         pipe = Pipeline([('vect', CountVectorizer()), ('tfidf', TfidfTransformer()),
                          ('model', DecisionTreeClassifier(criterion= 'entropy', max_depth = 20,
                                                           splitter='best', random_state=101))])

In [33]: model = pipe.fit(X_train, y_train)

In [34]: pred = model.predict(X_test)

In [35]: print("Classifier Accuracy Score: {}%".format(round(accuracy_score(y_test, pred)*100,2)))

         Classifier Accuracy Score: 99.64%
```

Fig 14:- Classifier Accuracy Score

4.1.2 For Prediction

```
In [1]: import pandas as pd
```

```
In [2]: df=pd.read_csv("train.csv")
```

```
In [3]: df.head()
```

Out[3]:

	id	title	author	text	label
0	0	House Dem Aide: We Didn't Even See Comey's Let...	Darrell Lucus	House Dem Aide: We Didn't Even See Comey's Let...	1
1	1	FLYNN: Hillary Clinton, Big Woman on Campus - ...	Daniel J. Flynn	Ever get the feeling your life circles the rou...	0
2	2	Why the Truth Might Get You Fired	Consortiumnews.com	Why the Truth Might Get You Fired October 29, ...	1
3	3	15 Civilians Killed In Single US Airstrike Hav...	Jessica Purkiss	Videos 15 Civilians Killed In Single US Airstr...	1
4	4	Iranian woman jailed for fictional unpublished...	Howard Portnoy	Print \nAn Iranian woman has been sentenced to...	1

```
In [4]: df.describe()
```

Out[4]:

	id	label
count	20800.000000	20800.000000
mean	10399.500000	0.500625
std	6004.587135	0.500012
min	0.000000	0.000000
25%	5199.750000	0.000000
50%	10399.500000	1.000000
75%	15599.250000	1.000000
max	20799.000000	1.000000

In [5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20800 entries, 0 to 20799
Data columns (total 5 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   id      20800 non-null  int64 
 1   title   20242 non-null  object
 2   author  18843 non-null  object
 3   text    20761 non-null  object
 4   label   20800 non-null  int64 
dtypes: int64(2), object(3)
memory usage: 812.6+ KB
```

In [6]:
```
df.isnull().sum()
```

Out[6]:
```
id          0
title     558
author   1957
text       39
label       0
dtype: int64
```

In [7]:
```
df=df.fillna('')
```

In [8]:
```
df.isnull().sum()
```

Out[8]:
```
id        0
title     0
author    0
text      0
label     0
dtype: int64
```

In [11]:
```
df.head()
```

Out[11]:

	text	label
0	House Dem Aide: We Didn't Even See Comey's Let...	1
1	Ever get the feeling your life circles the rou...	0
2	Why the Truth Might Get You Fired October 29, ...	1
3	Videos 15 Civilians Killed In Single US Airstr...	1
4	Print \nAn Iranian woman has been sentenced to...	1

In [12]:
```
from nltk.corpus import stopwords
```

In [13]:
```
from nltk.stem.porter import PorterStemmer
```

In [14]:
```
import re
```

In [15]:
```
port_stem=PorterStemmer()
```

In [16]:
```
port_stem
```

Out[16]:
```
<PorterStemmer>
```

In [17]:
```
port_stem.stem("Hi thIs is world * % %@@@")
```

Out[17]:
```
'hi this is world * % %@@@'
```

In [18]:
```
def stemming(content):
    con=re.sub('[^a-zA-Z]', ' ', content)
    con=con.lower()
    con=con.split()
    con=[port_stem.stem(word) for word in con if not word in stopwords.words('english')]
    con=' '.join(con)
    return con
```

In [19]:
```
stemming('Hi this is world')
```

Out[19]:
```
'hi world'
```

In [ ]:
```
df['text']= df['text'].apply(stemming)
```

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

```
df['text']= df['text'].apply(stemming)
```

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

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

```
y.shape
```

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

```
x_train , x_test , y_train, y_test = train_test_split(x, y, test_size=0.20)
```

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

```
vect=TfidfVectorizer()
```

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

```
model=DecisionTreeClassifier()
```

```
model.fit(x_train, y_train)
```

```
prediction=model.predict(x_test)
```

```
prediction
```

```
model.score(x_test, y_test)
```

```
import pickle
```

```
pickle.dump(vect, open('vector.pkl', 'wb'))
```

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

```
vector_form=pickle.load(open('vector.pkl', 'rb'))
```

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

```
def fake_news(news):
    news=stemming(news)
    input_data=[news]
    vector_form1=vector_form.transform(input_data)
    prediction = load_model.predict(vector_form1)
    return prediction
```

```
val=fake_news("""In these trying times, Jackie Mason is the Voice of Reason. [In this week's exclusive clip for Breitbart News, Jackie discusses the looming threat of Nort
```

```
if val==[0]:
    print('reliable')
else:
    print('unreliable')
```

Fig 15:- Prediction for fake news

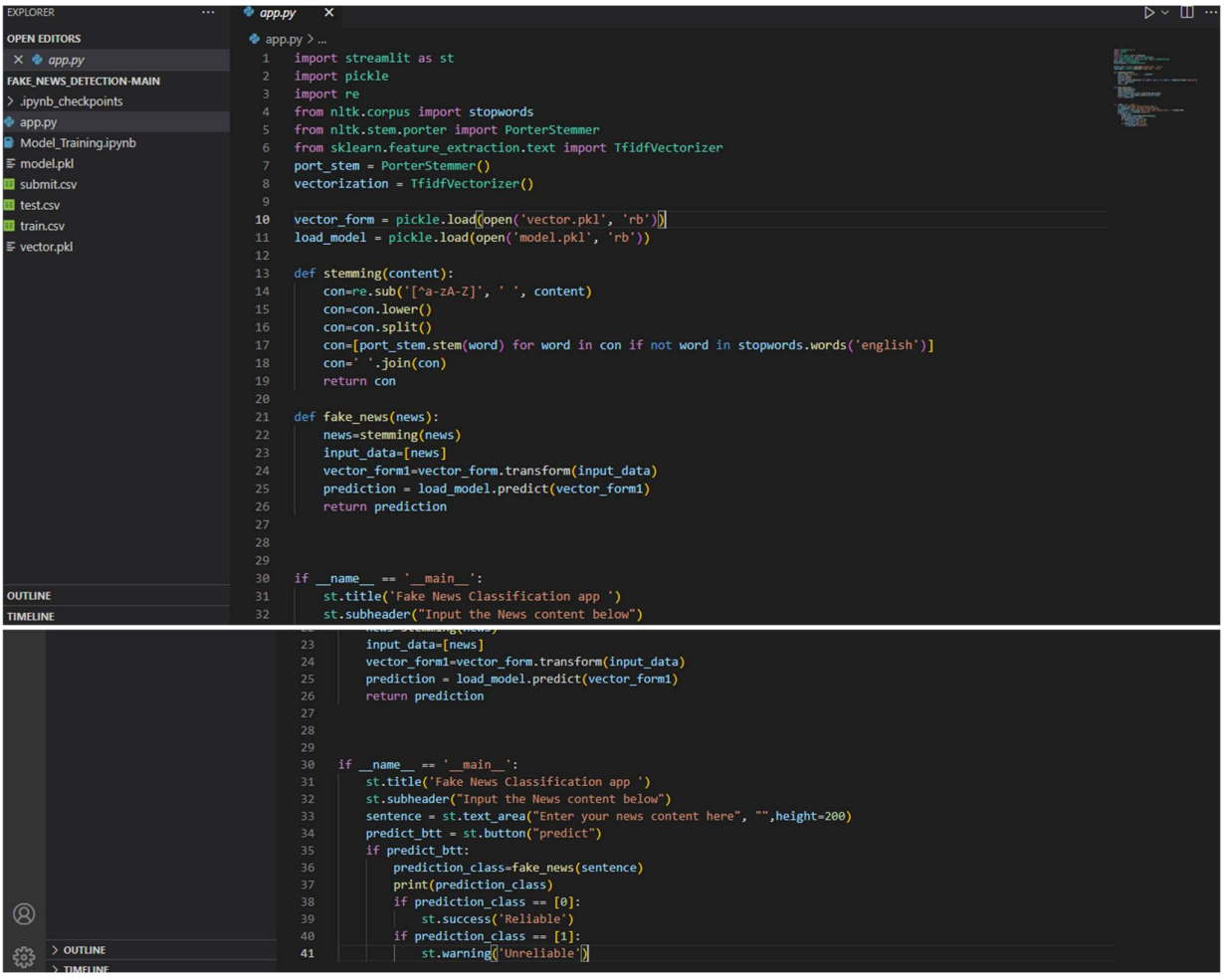


Fig 16:- streamlit webinterface

4.2 Output

Fig 17:- Local URL and Network URL

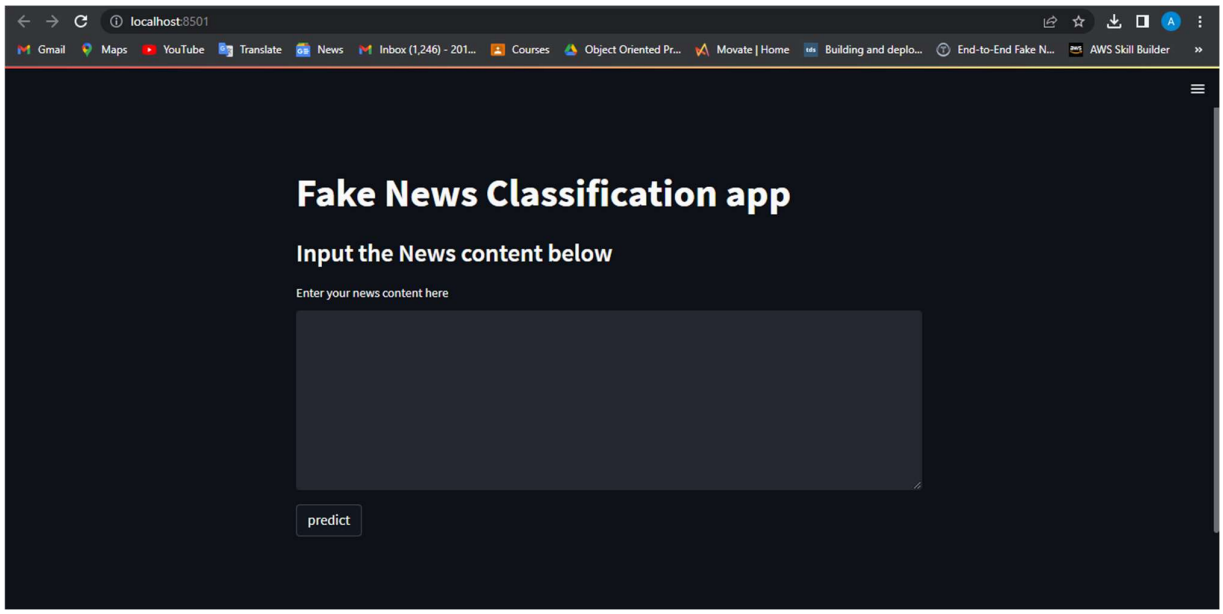


Fig 18:- Fake News Classification app

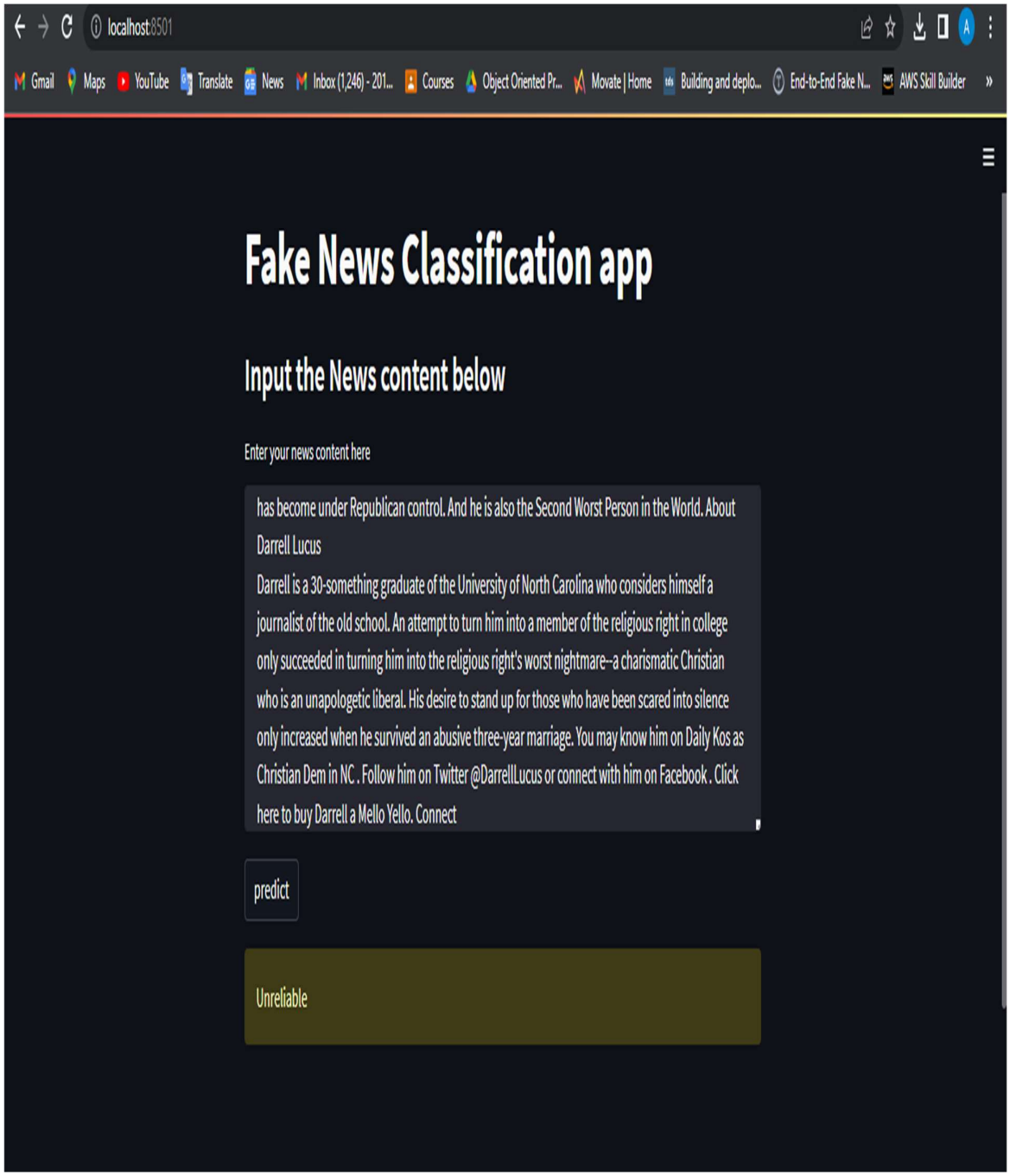


Fig 19:- Fake News Unreliable

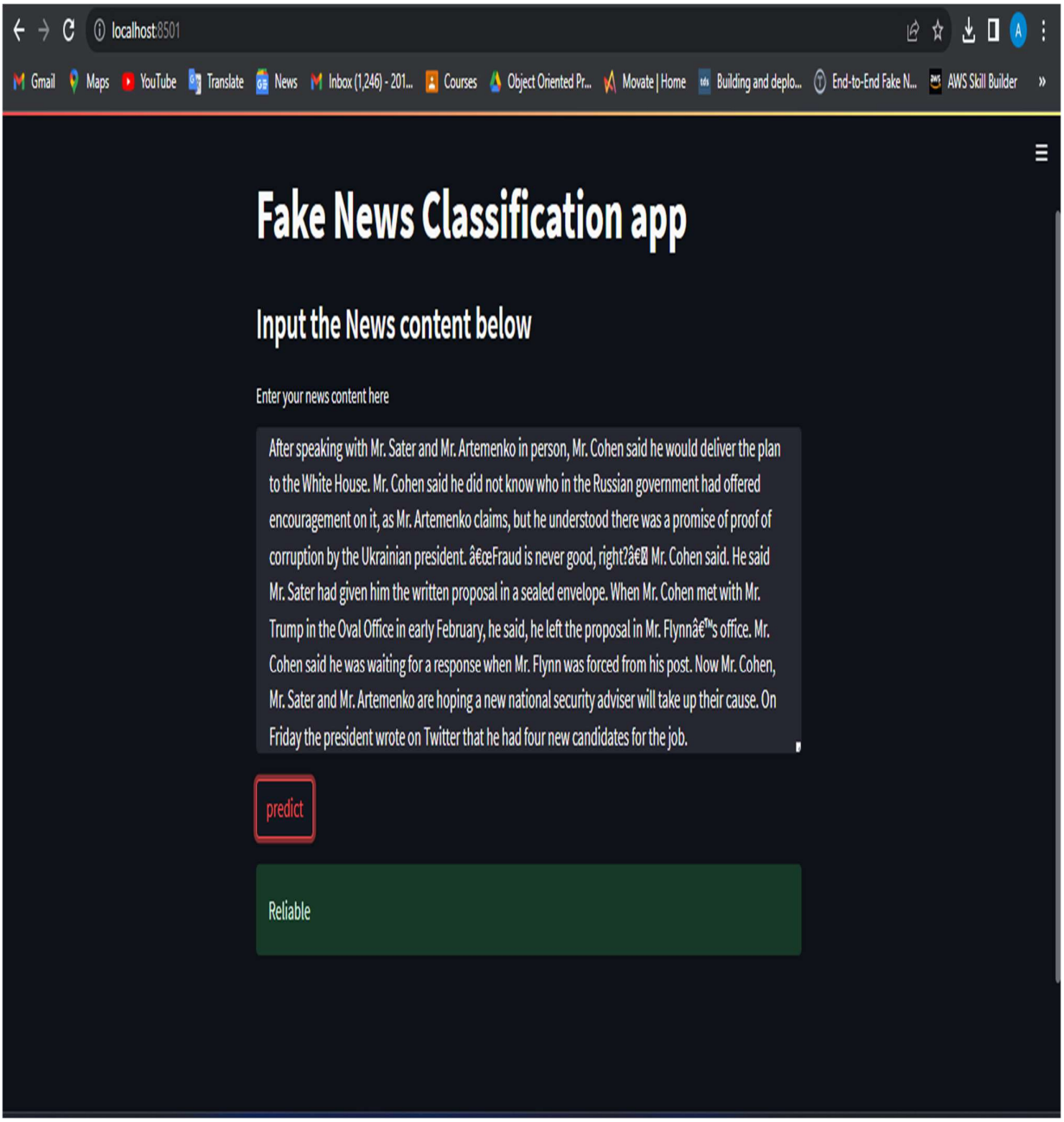


Fig 20:- Fake News Reliable

CHAPTER 5

CONCLUSION & FUTURE WORK

In this project, we forecast if an article is legitimate or fraudulent based on the relationship between the words. This system was built using datasets from the 2016 US presidential election. We achieved an accuracy of 99.64% by using the Word2Vec model for model building and the decision tree classifier for prediction.

In conclusion, the "fake news classifier using NLP" is an important application of natural language processing and machine learning techniques to address the problem of fake news. By training a classifier to distinguish between real and fake news articles, we can help prevent the spread of misinformation and improve the overall quality of information available to the public.

The proposed system architecture involves data collection, preprocessing, feature extraction, classification, and model evaluation, with each component serving a critical role in the overall performance of the system. The algorithm involves using a decision tree classifier or other classification algorithm to learn from the data and predict the label of new articles.

While the "fake news classifier using NLP" can be a powerful tool, it also has limitations, such as the need for high-quality labeled data, potential biases in the training data, and the difficulty of distinguishing between subtly misleading information and outright falsehoods. Nonetheless, with careful design and evaluation, the system can provide a useful tool for combating the spread of fake news and improving the accuracy and quality of information available to the public.

5.1 System Usability

The system usability for the "fake news classifier using NLP" depends on several factors, such as the accuracy of the classifier, the user interface, and the ease of use for non-technical

users.

The accuracy of the classifier is crucial for the usability of the system. If the classifier has a high false positive rate or false negative rate, it may misclassify articles and lead to misinformation. Therefore, the classifier needs to be carefully designed and evaluated to ensure high accuracy and reliability.

The user interface is another important factor in the usability of the system. The interface should be intuitive and easy to use, allowing non-technical users to input new articles and receive a prediction on whether the article is fake or real. The system should also provide clear explanations of how the classifier works and how the prediction was made, to increase the transparency and trustworthiness of the system.

Additionally, the ease of use of the system is critical for its adoption and use by the public. The system should be accessible and easy to use on a variety of devices, such as mobile phones and desktop computers. It should also be compatible with different web browsers and operating systems.

Overall, the system usability for the "fake news classifier using NLP" is critical for its effectiveness in combating the spread of fake news. The accuracy of the classifier, the user interface, and the ease of use are all important factors that must be carefully considered and designed to ensure high adoption and usefulness by the public.

5.2 Conclusion

In conclusion, a fake news classifier using NLP techniques and machine learning algorithms has the potential to address the growing problem of misinformation in today's society. By

automating the process of news analysis and verification, it can improve the efficiency and accuracy of identifying fake news articles, which can have a significant impact on public opinion and decision-making.

The proposed system involves several steps, including data collection, preprocessing, feature extraction, model training, and evaluation. By utilizing NLP techniques such as stemming, stopwords removal, and feature extraction methods like bag-of-words, TF-IDF, or word embeddings, the system can effectively extract relevant features from the text data and train a machine learning model to classify news articles as real or fake.

However, the system is not without its limitations and challenges. For instance, the accuracy of the model heavily relies on the quality and representativeness of the dataset used for training. Additionally, fake news articles can be deliberately crafted to appear as legitimate news, making them difficult to identify with automated tools.

Despite these challenges, the development of a fake news classifier using NLP techniques represents a significant step forward in addressing the problem of fake news and promoting media literacy and critical thinking. The proposed system has the potential to improve the quality and reliability of news and information available to the public, which is crucial for informed decision-making and the functioning of democratic societies.

5.3 Future Scope

In the future, we will work on the user interface or front end of this project, and we will strive to apply as many additional algorithms as possible to evaluate authentic and bogus news with more accuracy.

Multilingual Support: Currently, most fake news detection systems operate only in one language, mainly English. Expanding the system to other languages can help detect fake news in regions where English is not the primary language.

Fact-Checking: Integrating the system with fact-checking tools and databases can help validate the claims made in news articles and provide additional context for the users.

Combating Disinformation Campaigns: Combining the fake news classifier with social media monitoring tools can help detect disinformation campaigns and targeted propaganda by analyzing patterns of fake news distribution.

Continuous Learning: Implementing a continuous learning model that automatically updates the classifier as new data becomes available can help the system remain effective and relevant over time.

Hybrid Approaches: Combining different machine learning algorithms, such as ensemble models or deep learning models, can help improve the accuracy and reliability of the fake news classifier.

User Feedback Integration: Collecting feedback from users and integrating it into the system can help identify potential issues and improve the accuracy and usability of the classifier.

Overall, the future scope for the "fake news classifier using NLP" is vast, and there is a lot of potential for improving the system's accuracy, reliability, and usefulness. By continually adapting and improving the system, we can help combat the spread of fake news and improve the overall quality of information available to the public.

REFERENCES

[1] Lai C-M, Chen M-H, Kristiani E, Verma VK, Yang C-T. Fake News Classification Based on Content Level Features. Applied Sciences. 2022; 12(3):1116. https://doi.org/10.3390/app12031116

[2] Busioc, Costin, Stefan Ruseti, and Mihai Dascalu. "A Literature Review of NLP Approaches to Fake News Detection and Their Applicability to RomanianLanguage News Analysis." Revista Transilvania 10 (2020).

[3] Hirlekar, V. V., & Kumar, A. (2020). Natural Language Processing based Online Fake News Detection Challenges – A Detailed Review. 2020 5th International Conference on Communication and Electronics Systems (ICCES). doi:10.1109/icces48766.2020.9137915 10.1109/icces48766.2020.9137915

[4] Zhou, Z., Guan, H., Bhat, M. M., & Hsu, J. (2019). Fake news detection via NLP is vulnerable to adversarial attacks. arXiv preprint arXiv:1901.09657.

[5] Johnson, W.; Bouchard, T.J., Jr. Sex differences in mental abilities: G masks the dimensions on which they lie. Intelligence 2007, 35, 23–39.

[6] Newman, M.L.; Pennebaker, J.W.; Berry, D.S.; Richards, J.M. Lying words: Predicting deception from linguistic styles. Personal. Soc. Psychol. Bull. 2003, 29, 665–675.

[7] Dey, A.; Rafi, R.Z.; Parash, S.H.; Arko, S.K.; Chakrabarty, A. Fake News Pattern Recognition using Linguistic Analysis. In Proceedings of the 2018 Joint 7th International Conference on Informatics, Electronics & Vision (ICIEV) and 2018 2nd International Conference on Imaging, Vision & Pattern Recognition (icIVPR), Kitakyushu, Japan, 25–29 June 2018; pp. 305–309.

[8] Bondielli, A.; Marcelloni, F. A survey on fake news and rumour detection techniques. Inf. Sci. 2019, 497, 38–55.

[9] Marquardt, D. Linguistic Indicators in the Identification of Fake News. Mediat. Stud. 2019, 3, 95–114.

[10] Torabi Asr, F.; Taboada, M. Big Data and quality data for fake news and misinformation detection. Big Data Soc. 2019, 6, 2053951719843310.

ANNEXURE 1

Review Paper for the said project has been **published** in Gradiva Review Journal.

Paper Title:

Fake News Classifier Using NLP

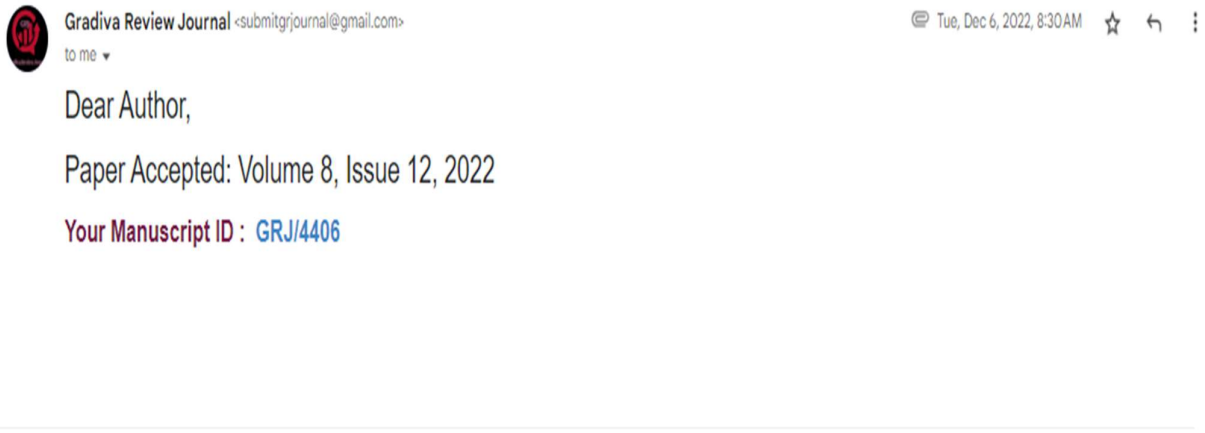
Abstract:

In today's culture, fake news has arisen as a major concern. Fake news has the ability to influence people's views and facts, making it the most lethal weapon for influencing society.

The proposed study applies natural language processing (NLP) algorithms to detect 'fake news,' or misleading news items received from untrustworthy sources. Fake news may be identified by creating a model using the Decision Tree Classifier approach. The data science community has responded by taking steps to solve the problem. It is difficult to know whether a piece of news is real or false. As a result, the proposed research uses datasets trained using the count vectorizer technique to detect false news, and its accuracy will be assessed using Natural Language Processing algorithms.

Authors:

Aman Mandal, Shamser Alam, Sahil Kansal



Gradiva Review Journal <submitgrjournal@gmail.com> Tue, Dec 6, 2022, 8:30 AM

to me

Dear Author,

Paper Accepted: Volume 8, Issue 12, 2022

Your Manuscript ID : GRJ/4406

Gradiva Review Journal Thu, Apr 27, 10:46 PM (7 days ago)

to me

Dear Author,

Please follow the Publication link below.

https://gradivareview.com/volume-8-issue-12-2022/

Please find the certificates attached below.

ANNEXURE 2

Review Paper for the said project has been **accepted** in Gradiva Review Journal.

Paper Title: Improving Fake News Detection Using Decision Tree Classifier

Abstract:

The rise of fake news has become a major problem in today's society, with serious consequences such as influencing public opinion and undermining trust in media and institutions. This has led to an increasing demand for automated tools to detect and classify fake news articles. In this paper, we propose a decision tree classifier for improving the detection of fake news using a combination of lexical and semantic features extracted from the news text. We use a publicly available dataset of news articles labeled as real or fake to train and evaluate the performance of the classifier. Our results show that the decision tree classifier outperforms other state-of-the-art machine learning classifiers, achieving an accuracy of 99.64% and a precision of 99.64% in detecting fake news articles. Additionally, we conduct a feature importance analysis to identify the most important features for classification. Our findings suggest that features related to the use of subjective language and emotional appeals are particularly informative for detecting fake news. Overall, our study demonstrates the effectiveness of decision tree classifiers for improving the detection of fake news and provides insights into the linguistic and semantic features that are most informative for classification.

Authors:

Aman Mandal, Shamser Alam, Sahil Kansal