Student Declaration of Authorship

Course code and name:	F20PA Research Methods Requirements Engineering F20PB Design and Implementation F20PC Project testing and Presentation
Type of assessment:	**Individual**
Coursework Title:	Final Year Dissertation
Student Name:	Varun Senthil Kumar
Student ID Number:	H00332328

Declaration of authorship. By signing this form:

- **I declare** that the work I have submitted for individual assessment OR the work I have contributed to a group assessment, is entirely my own. I have NOT taken the ideas, writings or inventions of another person and used these as if they were my own. My submission or my contribution to a group submission is expressed in my own words. Any uses made within this work of the ideas, writings or inventions of others, or of any existing sources of information (books, journals, websites, etc.) are properly acknowledged and listed in the references and/or acknowledgements section.

- I confirm that I have read, understood and followed the University's Regulations on plagiarism as published on the University's website, and that I am aware of the penalties that I will face should I not adhere to the University Regulations.

- I confirm that I have read, understood and avoided the different types of plagiarism explained in the University guidance on Academic Integrity and Plagiarism

Student Signature *(type your name):* *Varun Senthil Kumar*

Date: *26/04/2023*

Copy this page and insert it into your coursework file in front of your title page.
For group assessment each group member must sign a separate form and all forms must be included with the group submission.

Your work will not be marked if a signed copy of this form is not included with your submission.

Final Year Dissertation

Fake News Detection Using Machine Learning

Author:

Varun Senthil Kumar

Supervisor:

Ubaid Abbasi

Heriot Watt University

School of Mathematical and Computer Sciences

BSc. Computer Science (Artificial Intelligence)(Hons)

April 2023

26th April 2023

Declaration of Authorship

I, **Varun Senthil Kumar,** confirm that this work submitted for assessment is my own and is expressed in my own words. Any uses made within it of the works of other authors in any form (e.g., ideas, equations, figures, text, tables, programs) are properly acknowledged at any point of their use.

A list of the references employed is included.

Signed: Varun Senthil Kumar

Date: 26/04/2023

Abstract

With the spread of social media and the internet, the spread of fake news has increased. The public have more access to information, and there is no way for them to know the authenticity of the news. This calls for an automated solution to detect fake news. Given the versatility of these fake news authors, a tool to detect fake news is a challenging issue. This project aims to propose an ensemble machine learning algorithm to combat this issue and detect fake news accurately. To find out the best working algorithm, we compare the performance metrics of the proposed ensemble method against the performance metrics of individual classifiers. A publicly available dataset has been employed for experimentation and evaluation.

Keywords: Machine Learning, Fake News, Naïve Bayes, Decision Trees

Acknowledgement

I would like to thank my supervisor, Prof. Ubaid Abbasi, for his guidance and encouragement, which supported me throughout this project. His feedback and advice helped me shape my work and accomplish this project. I would also like to thank my family and friends for their constant support and motivation.

Table of Contents

List of Figures

List of Tables

Abbreviations

ML - Machine Learning

LR - Logistic Regression

DT - Decision Trees

MNB - Multinomial Decision Trees

CNN - Convolutional Neural Networks

RNN - Recurrent Neural Networks

SVM - Support Vector Machine

LSVM - Linear Support Vector Machine

RF - Random Forests

CART - Classification and Regression Tree

KNN - K Nearest Neighbor

BERT - Bidirectional Encoders Representations from Transformers

CSV - Comma Separated Values

ROC - Receiver Operating Characteristic

AUC - Area Under Curve

1. Introduction

The phenomenon ‘Fake News’ refers to the deception of readers by falsifying information. Fake news is not new to civilization. It has been in existence since the Roman times. In the modern world, the term has seen a usage rise on social media. The usage of the word has increased by 365% since 2016, showing the impact it has had on the society. [3] [4]

In the past decade, social media has seen a surge in users which has increased the spread of fake news. Popularity of smart gadgets and low-cost internet have also played a role in fake news spreading to the far corners of the world. Social media platforms are being used to promote false propaganda, provoke riots and mislead the youth with unverified news. Individuals share information with no verification which in turn leads to serious public issues. Numerous sites deliberately publish fake news to cause unrest among the society. [5] [4] [6]

Major organizations exist that differentiate fake news from true news. Websites like PolitiFact help to verify news articles. But these organizations approach is not automated, which is time consuming. [3] Journalists also point out unverified sources of news and filter them out. But the main concern is that these processes are slow and needs a lot of human involvement. [5]

On social media, common intervention methods to curb fake news include the removal of social bots that spread this content. Researchers use AI and NLP to build systems that detect fake news. But fake news detection is a complex task as the intention or propaganda cannot be measured. The task of comparing the fake article with the original article is complex as well, as different people have different opinions in classifying news as fake. [7] [8] [9]

1.1 Aim

This project aims to develop an ensemble model, consisting of the top 3 Machine Learning classifiers to detect fake news which are Logistic Regression, Decision Trees and Naïve Bayes to improve the detection of fake news. We also aim to compare the performance of the ensemble model against the performance of the classifiers when individually applied and similar existing ensemble models.

1.2 Objective

The objectives of this dissertation are:

- To research and develop an ensemble model consisting of the top 3 classifiers to detect fake news which are Logistic Regression, Decision Trees and Naïve Bayes.
- To compare the performance of the ensemble model and the performance of the classifiers individually.
- To evaluate the ensemble model and compare its performance with similar ensemble models that already exist.

2. **Background**

2.1 **Fake News**

After the 2016 US Presidential elections, the usage of the word "Fake News" has seen an upward trend. Popular news outlets have extensive coverage of false articles, and many organizations are working to tackle the spread of fake news. Usage of bots has increased the similarity of fake news to appear like real articles. This in turn has reduced the credibility of legitimate news articles and journalism. [10]

Previous studies show that fake news can be classified into five types: Satire, Parody, Hoax, Manipulation and Propaganda.

2.1.1 Satirical News

Satirical news is a genre of journalism that mimics the style of authentic journalism. These news gain inspiration from the authentic article and cover the same domains (Sports, Politics etc.). Satirical news is usually published to create a humorous effect on the news but depending on the perception of the reader and the subtlety of the satire used, these satirical news articles can have effects that were unintended. Studies have shown that 60% of the people don't understand the satire intended, and end up believing the article is real. [11]

In the 19th century, many news outlets in the US, in order to gain reader attention, started printing out intentionally deceptive articles but instead ended up creating waves about the community[12]. *The Daily Show with Trevor Noah* and *The Colbert Show* are examples of sources for satirical news articles. They turn real world events into a comical version to capture the attention of the younger audience using wry humor. Satirical news articles are usually not meant to mislead the audience, rather to keep the people informed about world events in a funny way.

2.1.2 News Parody

Parody news articles are another form of fake news. These types of articles share similar characteristics to satirical articles as they use humor. Their main difference is that parody articles use false information to express humor. The content of a parody news article is fabricated and usually aimed at political issues. Parody news articles are published with the assumption that the reader would understand the joke. When the parody is too subtle, the readers don’t understand the joke, fall for the deception and start sharing the article, believing the article to be true. [10]

2.1.3 Hoax

Hoaxing is a method of intentional fabrication of factual information, with the motive of misleading people. Hoax articles usually include deceptions that can cause political and civil unrest, material loss or in serious cases, harm to certain individuals. The *Yellow Press* and *Tabloids* are some main sources of hoax articles. They use unverified headlines, also known as clickbait, to increase reader attention. Hoax articles usually focus on sections like crime, gossip, celebrities etc.[1]

2.1.4 Manipulative News

These types of articles are news articles that contain modified images or videos to deceive people. Images and videos can be modified to show that events that did not occur originally actually happened. Politicians alter articles that are to be published to either favor themselves or to slander their rivals. Many organizations and political parties fund their own news outlets, so that the news published can be manipulated to their favor. This misleads the audience into believing the deceptions. Herman and Chomsky argue that before a news article is published, money and power allows the government and private entities to filter the content.[13]

In 2012, the aftermath of Hurricane Sandy, there were a lot of morphed photos and videos that circulated on social media., with events that did not occur reported to have occurred.

2.1.5 Propaganda

The definition of propaganda is to express an action or opinion to intentionally alter the opinions or perceptions of the public. In terms of journalism, news outlets lacking neutrality can alter information by shedding more light on negative or positive aspects purposely. Propaganda articles use psychological methods to make the readers believe the articles are true. It has been proven that propaganda articles achieved large scale impacts. Increased usage of the internet, freedom to speech and social media presence have amplified the number of sources that promote propaganda.[2] Political events like Brexit and the 2016 USA Presidential Campaign have witnessed a lot of propaganda news.

2.2 **Machine Learning**

Machine Learning can be defined as the capability of a machine to use data and algorithms in order to mimic human behavior. Machine Learning is considered as a field of study where computers gain the ability to learn without the need to be programmed.[14]

Machine Learning can be classified into three subcategories:

- Supervised Learning : The models are trained using fully labelled datasets. This allows the model to learn while training and increases accuracy over multiple trainings.
- Unsupervised Learning : The models are trained using unlabeled datasets and tries to analyze correlations between features in the dataset.
- Reinforcement Learning : The models are trained using a trial-and-error system. It analyzes the errors and learns from it to make the right decisions. [14]

2.2.1 Logistic Regression

Logistic regression is a classification model that is efficient for both linear and binary classification problems. Logistic Regression is used for predictive analysis. The model computes the input features and calculates the logistic result. It is a supervised machine learning model. The range of logistic regression is always 0 or 1 and doesn’t need a linear relationship between input and output. [15]

2.2.2 Decision Trees

One of the most popular and powerful Machine Learning classifiers is the Decision Tree. A decision tree goes through a list of sub-decisions, and then based on the findings, settles on a conclusion. The conclusion at the end of a subprocess leads to either the conclusion to the tree or another decision process based on the previous process. Generally, a decision tree consists of one root node and multiple leaf and internal nodes. Decision outcomes correspond to the leaf node. The structure of a decision tree is like an inverted tree. Decision trees follow the *divide-and-conquer* strategy. Fig 2.2.2 shows the structure of a decision tree. [16]

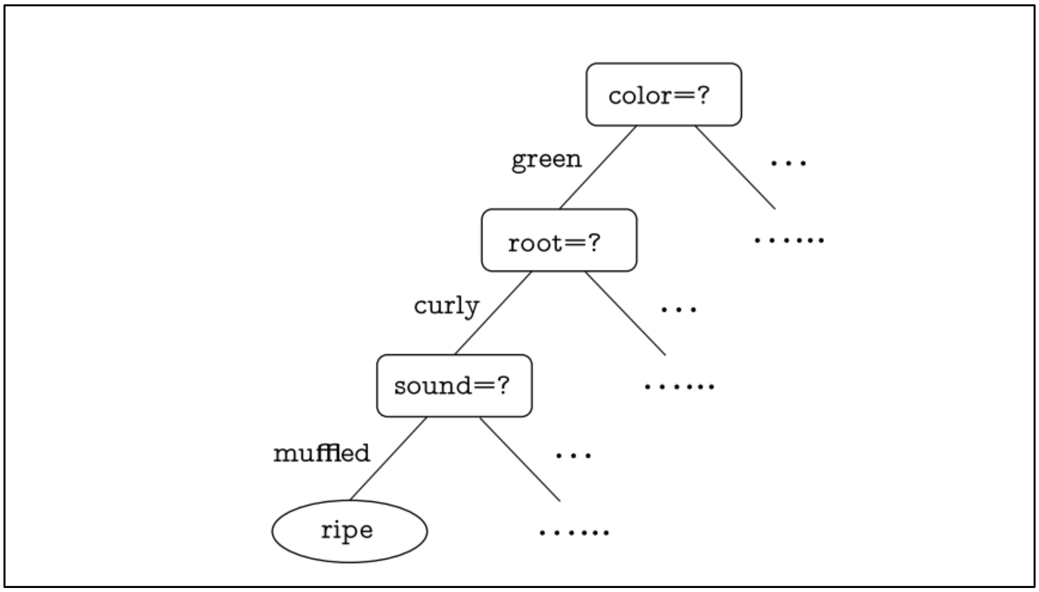


Figure 1: A example decision tree [16]

2.2.3 Naïve Bayes

Naïve Bayes is a machine learning model that is based on probability. The classifier is based on the Bayes Theorem and the assumption that the features are conditionally independent to the result. The Naïve Bayes classifier ignores dependencies, correlations and reduces a multivariate problem to a cluster of univariate problems.[17]

There are three types of Naïve Bayes classifiers:

- Multinomial Naïve Bayes: This model is popular for classification with discrete features and requires multinomial distribution (E.g.: Frequency of a word). [18]
- Gaussian Naïve Bayes: This model is used when the features are continuous. The assumption is that all the variables are normally distributed.
- Bernoulli Naïve Bayes: This model follows the principle of Bernoulli distribution and will be used if the features are binary(0 or 1). [19]

2.2.4 Ensemble models

A model consisting of multiple models and algorithms to predict an outcome is called an Ensemble Model. Ensemble models usually aggregate the predictions of each base model and provide a prediction for the combined model. Ensemble models show better performance than traditional models as they integrate the benefits of both deep learning models and ensemble models, thereby resulting in a model that performs greater in generalization. Even if an ensemble model consists of multiple individual models, it works as a single model. [20]

2.3 **Related Work**

Linguistic cues have been used to detect fake news since the 2000s. Linguist researchers have discovered language patterns that are used to deceive people. Phrases, certain tenses, sentences etc. are patterns that are commonly used in news articles that tend to deceive people. [3]

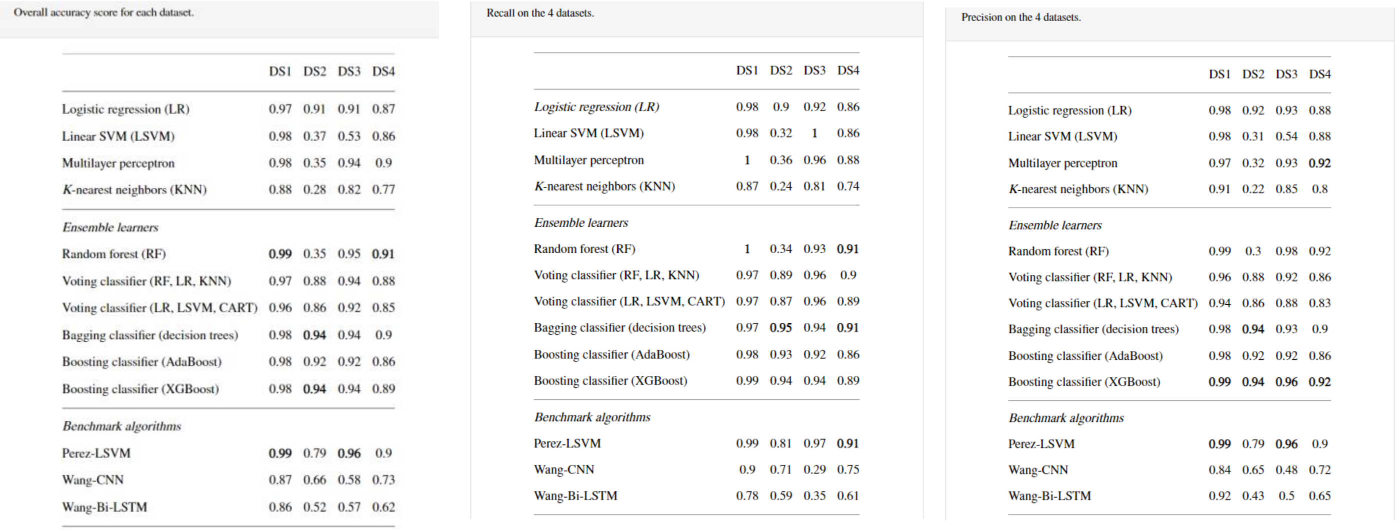
A study conducted by ***Burgoon et al***.[21] have filtered out a set of linguistic features that commonly tend to deceive people. The authors recruited a set of students and assigned them a task that involved theft and the students were interviewed after the task was completed. Based on the answers given by the students, the interviewer had to find out if the student was lying or not. This experiment was conducted to demonstrate the efficiency of using linguistic cues to detect deception. Using a C4.5 decision tree, the authors were able to filter out words, sentences or tenses that were used by the students to cheat the interviewer. Using this experiment, Burgoon et al. were able to filter out words, phrases or sentences that tend to fool people. This study has been a benchmark for future studies involving the detection of fake news using linguistic cues.[21]

Anjali et al [6] proposes and designs an ensemble model using Support Vector Machine(SVM) and Naïve Bayes classifiers. The authors build the ensemble model and compare their model's performance with the performance of the classifiers individually. Their ensemble model achieves an accuracy of 93.% on the dataset they used while existing work done on the same dataset achieved marginally lower. Using the result they achieved, they conclude that their ensemble model performs substantially better than the classifiers individually on the dataset. Their model was trained on a small dataset that contained articles from only one domain, and therefore their model couldn't achieve optimal results.[6]

Iftikhar et al. [7] proposes that ensemble models achieve higher performance than individual classifiers. The authors propose to build two different voting classifiers consisting of 3 classifiers each. One voting classifier consists of Logistic Regression, Random Forest and K Nearest Neighbor (KNN) while the other voting classifier consists of Logistic Regression, Linear SVM and CART. The authors compare the performance of their voting classifiers with the performance of existing boosting algorithms, XGBoost and AdaBoost.

The authors have used four different datasets and evaluated the performances of each classifier on these datasets. Using a feature extraction method called Linguistic Inquiry and Word Count(LIWC), they converted stop words, punctuations, etc. to a numerical value, which is then used to train the classifiers. All classifiers were fine-tuned using hyperparameters to achieve optimal results. Fig. 2 shows the performance comparison.[7]

Based on the results obtained, Iftikhar et al[7] conclude that the existing algorithm XGBoost achieved higher performance than the rest of the classifiers.



Overall accuracy score for each dataset.

	DS1	DS2	DS3	DS4
Logistic regression (LR)	0.97	0.91	0.91	0.87
Linear SVM (LSVM)	0.98	0.37	0.53	0.86
Multilayer perceptron	0.98	0.35	0.94	0.9
K-nearest neighbors (KNN)	0.88	0.28	0.82	0.77
Ensemble learners				
Random forest (RF)	**0.99**	0.35	0.95	**0.91**
Voting classifier (RF, LR, KNN)	0.97	0.88	0.94	0.88
Voting classifier (LR, LSVM, CART)	0.96	0.86	0.92	0.85
Bagging classifier (decision trees)	0.98	**0.94**	0.94	0.9
Boosting classifier (AdaBoost)	0.98	0.92	0.92	0.86
Boosting classifier (XGBoost)	0.98	**0.94**	0.94	0.89
Benchmark algorithms				
Perez-LSVM	**0.99**	0.79	**0.96**	0.9
Wang-CNN	0.87	0.66	0.58	0.73
Wang-Bi-LSTM	0.86	0.52	0.57	0.62

Recall on the 4 datasets.

	DS1	DS2	DS3	DS4
Logistic regression (LR)	0.98	0.9	0.92	0.86
Linear SVM (LSVM)	0.98	0.32	1	0.86
Multilayer perceptron	1	0.36	0.96	0.88
K-nearest neighbors (KNN)	0.87	0.24	0.81	0.74
Ensemble learners				
Random forest (RF)	1	0.34	0.93	**0.91**
Voting classifier (RF, LR, KNN)	0.97	0.89	0.96	0.9
Voting classifier (LR, LSVM, CART)	0.97	0.87	0.96	0.89
Bagging classifier (decision trees)	0.97	**0.95**	0.94	**0.91**
Boosting classifier (AdaBoost)	0.98	0.93	0.92	0.86
Boosting classifier (XGBoost)	0.99	0.94	0.94	0.89
Benchmark algorithms				
Perez-LSVM	0.99	0.81	0.97	**0.91**
Wang-CNN	0.9	0.71	0.29	0.75
Wang-Bi-LSTM	0.78	0.59	0.35	0.61

Precision on the 4 datasets.

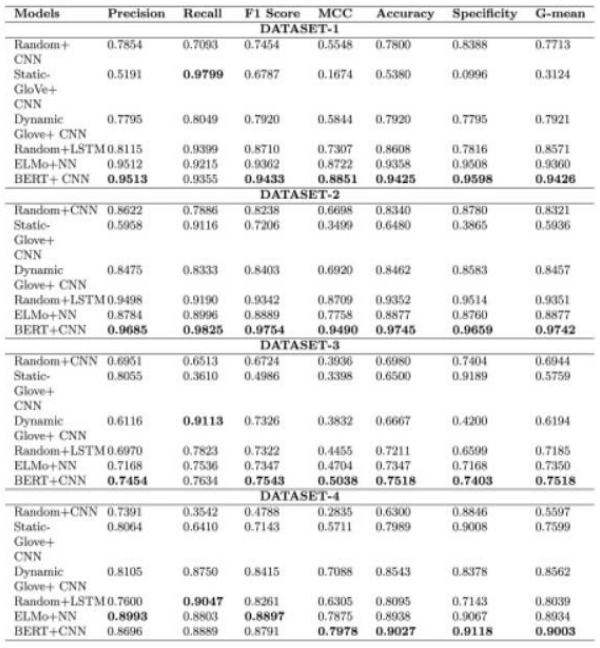
	DS1	DS2	DS3	DS4
Logistic regression (LR)	0.98	0.92	0.93	0.88
Linear SVM (LSVM)	0.98	0.31	0.54	0.88
Multilayer perceptron	0.97	0.32	0.93	**0.92**
K-nearest neighbors (KNN)	0.91	0.22	0.85	0.8
Ensemble learners				
Random forest (RF)	0.99	0.3	0.98	0.92
Voting classifier (RF, LR, KNN)	0.96	0.88	0.92	0.86
Voting classifier (LR, LSVM, CART)	0.94	0.86	0.88	0.83
Bagging classifier (decision trees)	0.98	**0.94**	0.93	0.9
Boosting classifier (AdaBoost)	0.98	0.92	0.92	0.86
Boosting classifier (XGBoost)	**0.99**	**0.94**	**0.96**	**0.92**
Benchmark algorithms				
Perez-LSVM	**0.99**	0.79	**0.96**	0.9
Wang-CNN	0.84	0.65	0.48	0.72
Wang-Bi-LSTM	0.92	0.43	0.5	0.65

Figure 2: Performance comparison between classifiers [7]

Monika et al.[22] proposes a hybrid framework, called *BerConvoNet*, which is a framework that combines Bidirectional Encoder Representations from Transformers(BERT) and Convolutional Neural Networks (CNN). BERT is a pretrained, unsupervised language model developed by Google and trained on Wikipedia.

The author has BERT used to generate word embeddings and detect homographs in news articles. Using BERT, the author transforms homographs, small phrases etc. into vector representations. CNN has been integrated to detect media-based fake news. To validate their model's performance, the authors use four different datasets and compare the results. Fig. 3 shows the performance comparison of *BerConvoNet* with existing algorithms. [22]

The *BerConvoNet* achieves slightly low performance, due to the authors using just small phrases to determine whether the article is true or fake. BERT could not transform long sentences into vector representations, which could have led to the model achieving optimal performance. [22]



Models	Precision	Recall	F1 Score	MCC	Accuracy	Specificity	G-mean
DATASET-1							
Random+ CNN	0.7854	0.7093	0.7454	0.5548	0.7800	0.8388	0.7713
Static-GloVe+ CNN	0.5191	**0.9799**	0.6787	0.1674	0.5380	0.0996	0.3124
Dynamic Glove+ CNN	0.7795	0.8049	0.7920	0.5844	0.7920	0.7795	0.7921
Random+LSTM	0.8115	0.9399	0.8710	0.7307	0.8608	0.7816	0.8571
ELMo+NN	0.9512	0.9215	0.9362	0.8722	0.9358	0.9508	0.9360
BERT+ CNN	**0.9513**	0.9355	**0.9433**	**0.8851**	**0.9425**	**0.9598**	**0.9426**
DATASET-2							
Random+CNN	0.8622	0.7886	0.8238	0.6698	0.8340	0.8780	0.8321
Static-Glove+ CNN	0.5958	0.9116	0.7206	0.3499	0.6480	0.3865	0.5936
Dynamic Glove+ CNN	0.8475	0.8333	0.8403	0.6920	0.8462	0.8583	0.8457
Random+LSTM	0.9498	0.9190	0.9342	0.8709	0.9352	0.9514	0.9351
ELMo+NN	0.8784	0.8996	0.8889	0.7758	0.8877	0.8760	0.8877
BERT+CNN	**0.9685**	**0.9825**	**0.9754**	**0.9490**	**0.9745**	**0.9659**	**0.9742**
DATASET-3							
Random+CNN	0.6951	0.6513	0.6724	0.3936	0.6980	0.7404	0.6944
Static-Glove+ CNN	0.8055	0.3610	0.4986	0.3398	0.6500	0.9189	0.5759
Dynamic Glove+ CNN	0.6116	**0.9113**	0.7326	0.3832	0.6667	0.4200	0.6194
Random+LSTM	0.6970	0.7823	0.7322	0.4455	0.7211	0.6599	0.7185
ELMo+NN	0.7168	0.7536	0.7347	0.4704	0.7347	0.7168	0.7350
BERT+CNN	**0.7454**	0.7634	**0.7543**	**0.5038**	**0.7518**	**0.7403**	**0.7518**
DATASET-4							
Random+CNN	0.7391	0.3542	0.4788	0.2835	0.6300	0.8846	0.5597
Static-Glove+ CNN	0.8064	0.6410	0.7143	0.5711	0.7989	0.9008	0.7599
Dynamic Glove+ CNN	0.8105	0.8750	0.8415	0.7088	0.8543	0.8378	0.8562
Random+LSTM	0.7600	**0.9047**	0.8261	0.6305	0.8095	0.7143	0.8039
ELMo+NN	**0.8993**	0.8803	**0.8897**	0.7875	0.8938	0.9067	0.8934
BERT+CNN	0.8696	0.8889	0.8791	**0.7978**	**0.9027**	**0.9118**	**0.9003**

Figure 3: Performance comparison for BerConvoNet [22]

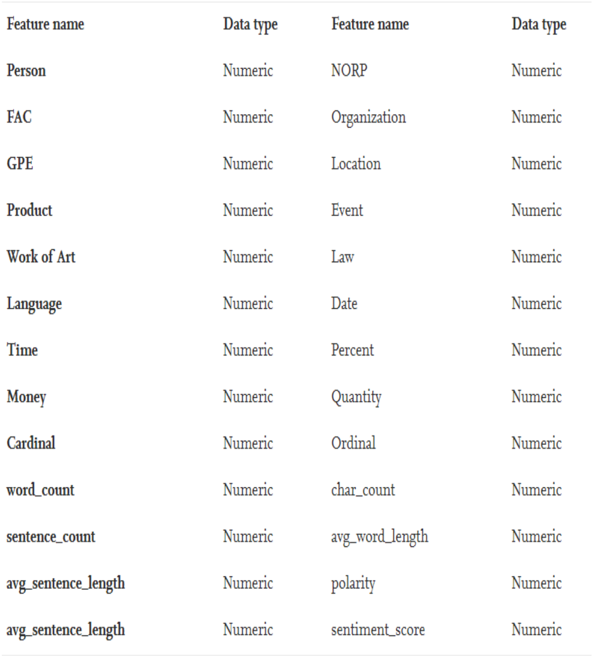
An ensemble model consisting of Convolutional Neural Networks(CNN) and Recurrent Neural Networks(RNN) has been proposed by ***Jamal et al.***[23] Previous studies show that the usage of Neural Networks has shown successful results in fake news detection and classification tasks.

The author uses the famous FA-KE and ISOT datasets to train their ensemble model. After splitting the datasets into training and test sets, the test data is split into sentences and stopwords, punctuation etc. are removed. Then, the training and test sets are tokenized using the Keras library. The author uses CNN that has been tuned with hyperparameters like 'kernel size' and 'number of filters' to extract the local features from the datasets. The RNN layer uses the local features extracted by CNN to classify them as either true or fake news.

The authors conclude that their proposed ensemble model combining CNN and RNN have performed marginally better than other classifiers and have achieved almost perfect scores for all performance metrics.[23] Although near-perfect scores are achieved, the model is not trained on datasets containing images or videos, which could have been integrated as the model makes use of CNN.

Hakak et al.[24] proposes an ensemble model consisting of a combination of Decision Trees, Random Forest and Extra Trees. The ensemble model was built using a bagging approach, as it reduces overfitting and provides stability to the model. The authors tune the ensemble model using the *random_search* hyperparameter.

Hakak et al. makes use of the ISOT and Liar datasets to train the model. The authors identify 26 linguistic features and extracted them from the dataset. Fig.4 shows the 26 linguistic features identified. After the extraction of the linguistic features, the dataset is preprocessed using the NLTK library to remove stopwords, punctuation etc. After training their ensemble model on their datasets, they achieve perfect scores for all performance metrics for the ISOT dataset.[24]



Feature name	Data type	Feature name	Data type
Person	Numeric	NORP	Numeric
FAC	Numeric	Organization	Numeric
GPE	Numeric	Location	Numeric
Product	Numeric	Event	Numeric
Work of Art	Numeric	Law	Numeric
Language	Numeric	Date	Numeric
Time	Numeric	Percent	Numeric
Money	Numeric	Quantity	Numeric
Cardinal	Numeric	Ordinal	Numeric
word_count	Numeric	char_count	Numeric
sentence_count	Numeric	avg_word_length	Numeric
avg_sentence_length	Numeric	polarity	Numeric
avg_sentence_length	Numeric	sentiment_score	Numeric

Figure 4: The linguistic features extracted. [24]

2.4 Conclusion

The spreading of fake news and misinformation has been on the rise, since the COVID-19 impact, and has had dire consequences on the general public. In the above section, we have discussed the impact of fake news and the importance of detecting fake news. Critically analyzing the different approaches taken towards the detection of fake news using machine learning algorithms and how different algorithms provide different results in detecting fake news has been understood. A clear knowledge about the usage of ensemble methods to detect fake news has been achieved due to the undertaking of this study. Machine Learning is crucial for the development of fake news detection, especially in the field of journalism. Deep learning models can be designed and used to curb the spread of fake news and misinformation to the public.

3. Implementation

3.1 Requirement Analysis

The functional and non-functional requirements of the project are given below. Requirements are divided into three categories based on their priority.

- Must Have (M) – Essential requirements for the project.
- Should Have (S) – Essential but can be replaced if they are not achieved.
- Could Have (C) – Not essential to the project and could be considered if time is available.

3.1.1 Functional Requirements

FR No.	Description	Priority	Status
FR-1	Find a suitable dataset that contains both true and fake news articles	Must	Completed
FR-2	Preprocess data to remove unwanted features and convert text to numerical values	Must	Completed
FR-3	Apply appropriate feature extraction methods to obtain optimal performance	Must	Completed
FR-4	Build the ensemble model of Logistic Regression, Decision Trees and Naïve Bayes	Must	Completed
FR-5	Model must detect fake and true news using the ensemble model	Must	Completed
FR-6	Ensemble model has to be evaluated using the test set and perform evaluation metrics	Must	Completed
FR-7	Compare ensemble model's performance with the performance of the individual classifiers and similar existing ensemble models	Should	Completed

Table 1: Functional requirements table

3.1.2 Non-Functional Requirements

NFR No.	Description	Priority
NFR-1	The code must be well documented and contain comments for better readability.	Must
NFR-2	The code must be reusable	Must
NFR-3	The code must be backed up on GitHub	Must
NFR-4	The code must be flexible to accommodate any future improvements	Must

Table 2: Non-Functional requirements table

3.2 **Project Workflow**

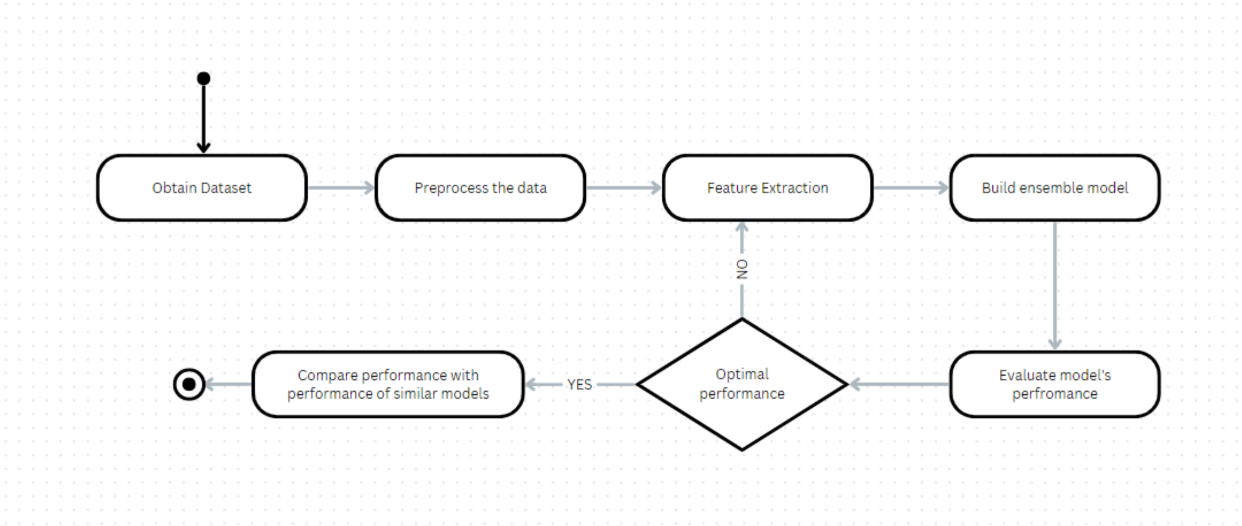


Figure 5: Project Workflow

Fig 1 describes the workflow that has been followed during the development of the project. The optimal performance of the ensemble model depends on accurate feature extraction. This requires frequent testing of the ensemble model using different features from the dataset.

3.3 **Source Code**

The ensemble model has been developed and evaluated on Jupyter Notebook, using Python code. The code for data preprocessing, feature extraction, ensemble model implementation and individual classifier implementation have all been separated for readability. Comments have been added where necessary for better understanding of the code. The datasets and Jupyter notebooks that have been used in this project have been stored on GitHub.

https://github.com/VarunPro23/Fake-News-Detection

3.4 **Data Collection and Preparation**

3.4.1 Datasets

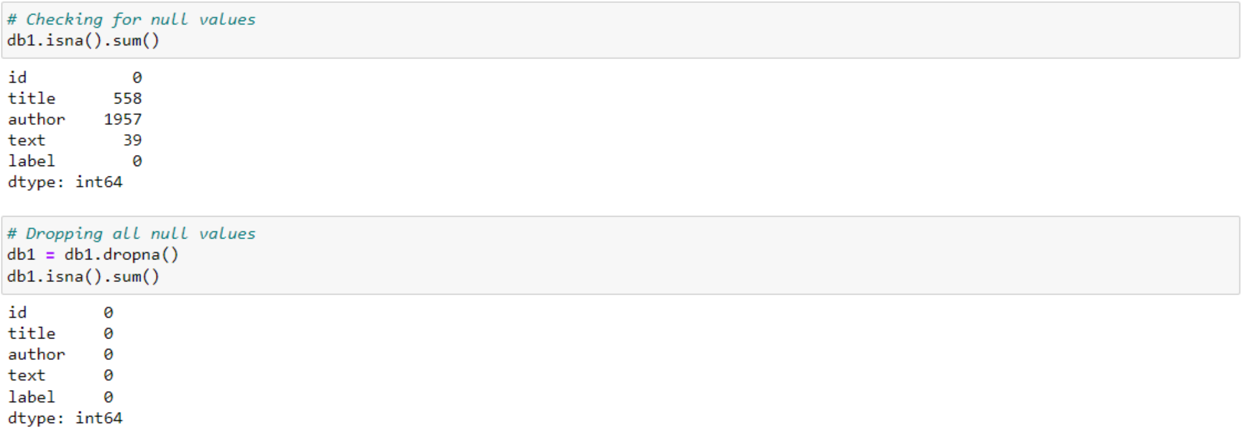
The datasets that have been used in the development of this project are open source and publicly available online. The datasets contain true and fake news from multiple sources. The truthful articles are published and contain real world events, while the fake news articles contain false claims. The authenticity of the articles in the dataset have been verified using the website *Politifact.com.* Two different datasets have been used in this project and are described briefly below.

The first dataset has been taken from Kaggle [25](referred to as DB1) and contains 20836 articles. The dataset has been built using articles from multiple sources. The articles cover multiple domains like politics, sports, business etc. The dataset has a comma-separated values(CSV) file with the features *id, title, author, text* and *label.*

The second dataset has also been taken from Kaggle [26] (referred to as DB2) and contains 3352 articles, both fake and true. The truthful news articles have been taken from sources like BBC, New York Times, CNN etc. The articles cover multiple sectors like sports, entertainment etc. The dataset has a comma-separated values(CSV) file with the features *URL, Headline, Body* and *Label.*

3.4.2 Data Preprocessing

We loaded the DB1 dataset as a pandas data frame. While checking for null values in the dataset, we found there were few records with null values. Using the dropna() function, we dropped these records. Next, we dropped the *text* column using the drop(), as each record contains more than 80 characters, and it will be hard to train the model.



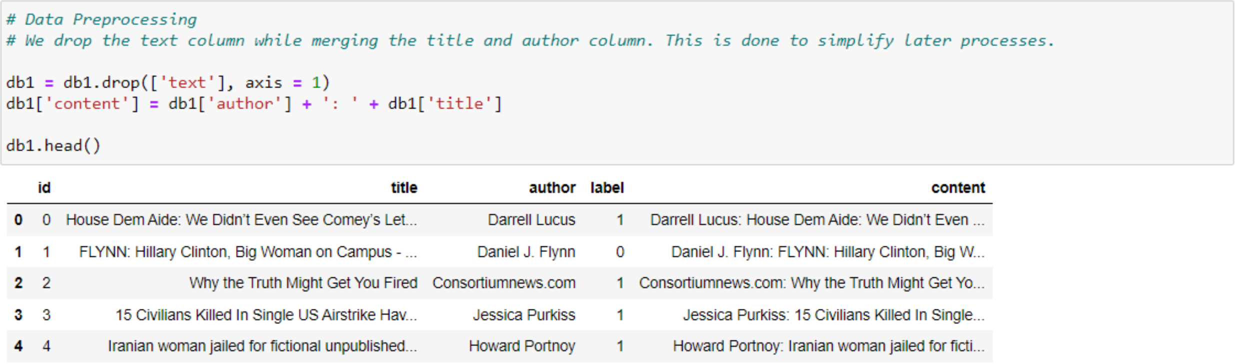
```
# Checking for null values
db1.isna().sum()
```

```
id           0
title      558
author    1957
text        39
label        0
dtype: int64
```

```
# Dropping all null values
db1 = db1.dropna()
db1.isna().sum()
```

```
id        0
title     0
author    0
text      0
label     0
dtype: int64
```

Figure 6: Checking for and dropping null values in DB1.

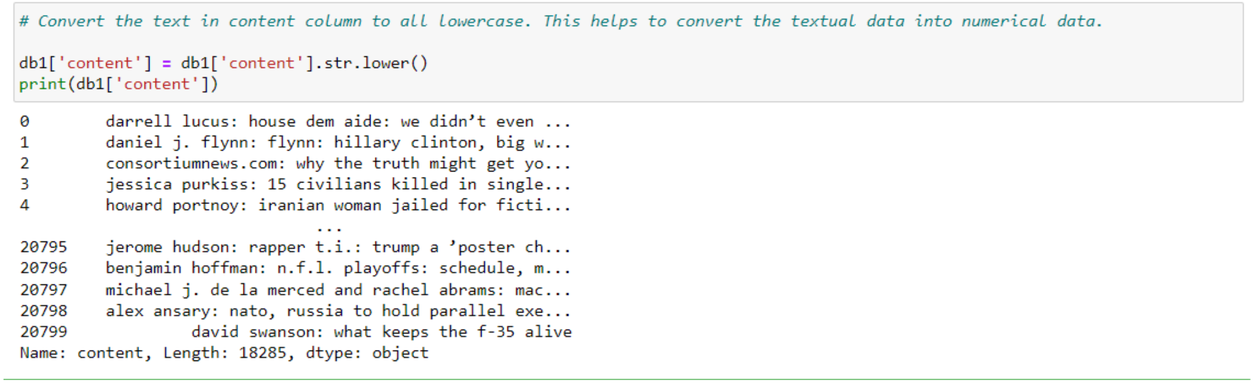


```
# Data Preprocessing
# We drop the text column while merging the title and author column. This is done to simplify later processes.

db1 = db1.drop(['text'], axis = 1)
db1['content'] = db1['author'] + ': ' + db1['title']

db1.head()
```

	id	title	author	label	content
0	0	House Dem Aide: We Didn't Even See Comey's Let...	Darrell Lucus	1	Darrell Lucus: House Dem Aide: We Didn't Even ...
1	1	FLYNN: Hillary Clinton, Big Woman on Campus - ...	Daniel J. Flynn	0	Daniel J. Flynn: FLYNN: Hillary Clinton, Big W...
2	2	Why the Truth Might Get You Fired	Consortiumnews.com	1	Consortiumnews.com: Why the Truth Might Get Yo...
3	3	15 Civilians Killed In Single US Airstrike Hav...	Jessica Purkiss	1	Jessica Purkiss: 15 Civilians Killed In Single...
4	4	Iranian woman jailed for fictional unpublished...	Howard Portnoy	1	Howard Portnoy: Iranian woman jailed for ficti...

Figure 7: Merging the author and title column into one column in DB1.

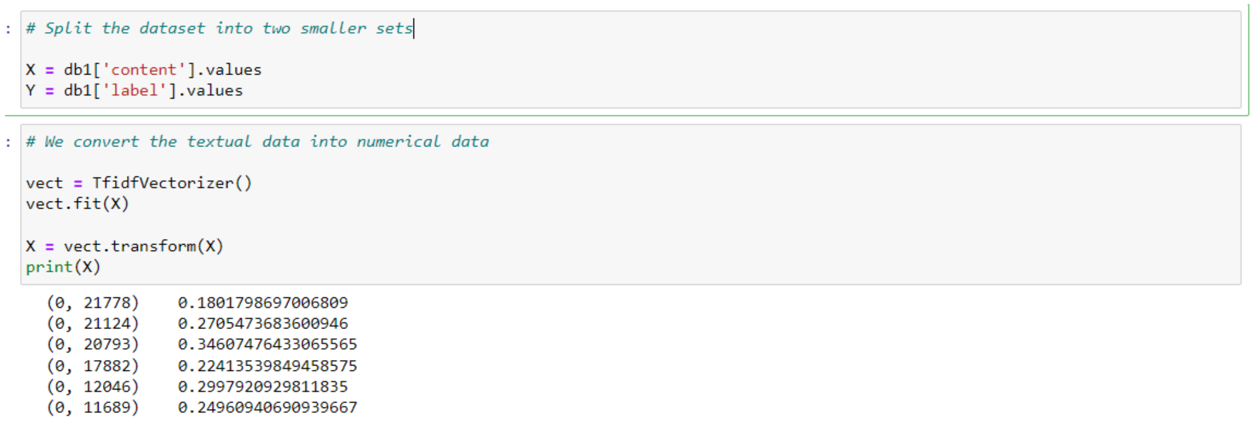
As shown in Fig. 7, we merged the *author* and *title* column into a new column called *content*, as it helps reduce training time. Using the function str.lower(), we converted all text in the content column into lowercase, as it makes it easier to convert textual content into numerical values, which is shown in Fig. 8. We split the dataset into smaller sets of the columns *content* and *label* as *X* and *Y*, respectively. We converted all data in *X* to numerical values using the TfidfVectorizer() function.



```
# Convert the text in content column to all lowercase. This helps to convert the textual data into numerical data.

db1['content'] = db1['content'].str.lower()
print(db1['content'])
```

```
0        darrell lucus: house dem aide: we didn’t even ...
1        daniel j. flynn: flynn: hillary clinton, big w...
2        consortiumnews.com: why the truth might get yo...
3        jessica purkiss: 15 civilians killed in single...
4        howard portnoy: iranian woman jailed for ficti...
                               ...
20795    jerome hudson: rapper t.i.: trump a ’poster ch...
20796    benjamin hoffman: n.f.l. playoffs: schedule, m...
20797    michael j. de la merced and rachel abrams: mac...
20798    alex ansary: nato, russia to hold parallel exe...
20799             david swanson: what keeps the f-35 alive
Name: content, Length: 18285, dtype: object
```

Figure 8: Converting the content in Body column into lowercase in DB1.



```
# Split the dataset into two smaller sets

X = db1['content'].values
Y = db1['label'].values
```

```
# We convert the textual data into numerical data

vect = TfidfVectorizer()
vect.fit(X)

X = vect.transform(X)
print(X)
```

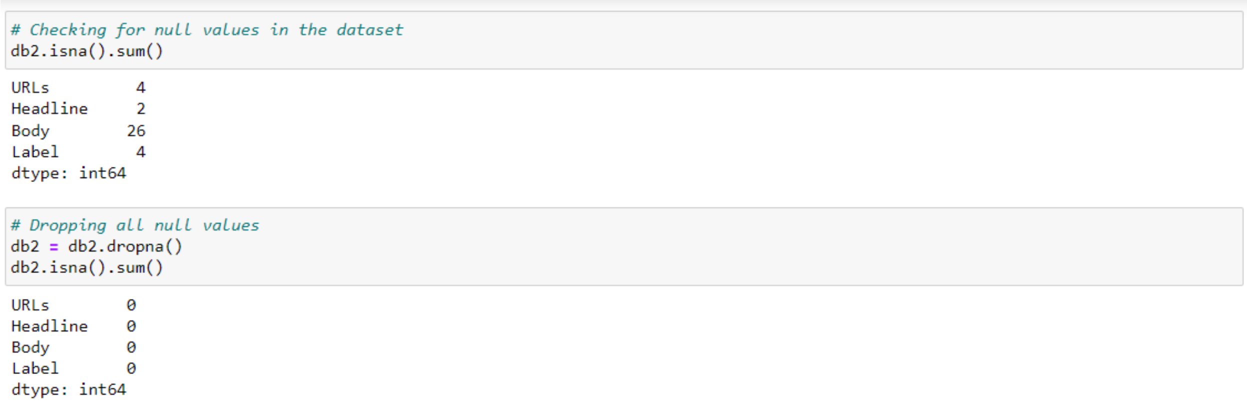
```
  (0, 21778)	0.1801798697006809
  (0, 21124)	0.2705473683600946
  (0, 20793)	0.34607476433065565
  (0, 17882)	0.22413539849458575
  (0, 12046)	0.2997920929811835
  (0, 11689)	0.24960940690939667
```

Figure 9: Convert textual value in content column into vector format in DB1.

We loaded the DB2 dataset as a pandas data frame. Similar to the process for DB1, we checked for null values in DB2 and dropped any null values using the dropna() function. Next, we dropped the *URLs* and *Headline* columns using the drop() as these columns are the least useful in detecting fake news. We then converted the text in the Body column to all lowercase using str.lower() function to simplify their conversion to vector format.

The dataset is then split into two smaller sets containing the column *Body* and *Label* as *x* and *y*, respectively. Using the TfidfVectorizer() function, we converted all the data in dataset *x* to its vector format.

Fig. 10 to 13 show the code snippets used to preprocess the data for DB2.



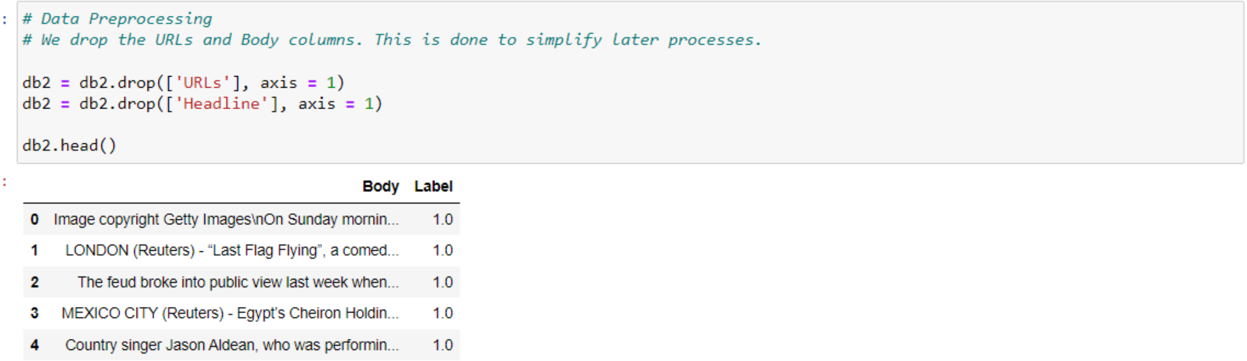
```
# Checking for null values in the dataset
db2.isna().sum()
```

```
URLs         4
Headline     2
Body        26
Label        4
dtype: int64
```

```
# Dropping all null values
db2 = db2.dropna()
db2.isna().sum()
```

```
URLs        0
Headline    0
Body        0
Label       0
dtype: int64
```

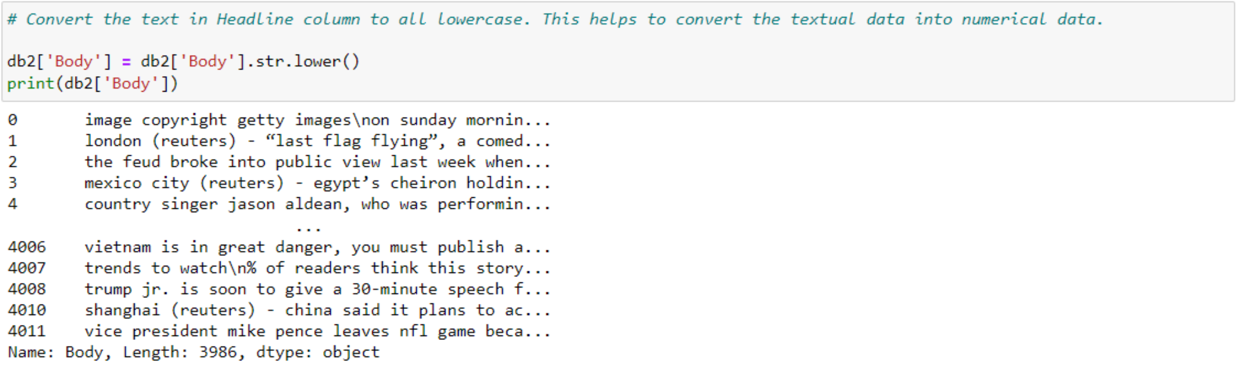
Figure 10: Checking for and dropping null values in DB2.



```
# Data Preprocessing
# We drop the URLs and Body columns. This is done to simplify later processes.

db2 = db2.drop(['URLs'], axis = 1)
db2 = db2.drop(['Headline'], axis = 1)

db2.head()
```

	Body	Label
0	Image copyright Getty Images\nOn Sunday mornin...	1.0
1	LONDON (Reuters) - “Last Flag Flying”, a comed...	1.0
2	The feud broke into public view last week when...	1.0
3	MEXICO CITY (Reuters) - Egypt’s Cheiron Holdin...	1.0
4	Country singer Jason Aldean, who was performin...	1.0

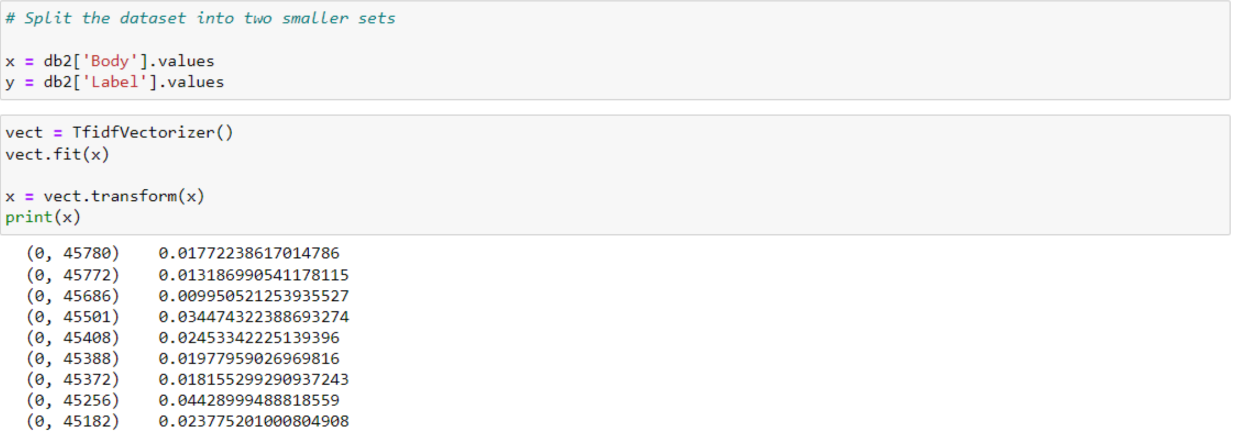
Figure 11: Dropping the URLs and Body Columns from DB2.



```
# Convert the text in Headline column to all lowercase. This helps to convert the textual data into numerical data.

db2['Body'] = db2['Body'].str.lower()
print(db2['Body'])
```

```
0       image copyright getty images\non sunday mornin...
1       london (reuters) - “last flag flying”, a comed...
2       the feud broke into public view last week when...
3       mexico city (reuters) - egypt’s cheiron holdin...
4       country singer jason aldean, who was performin...
                              ...
4006    vietnam is in great danger, you must publish a...
4007    trends to watch\n% of readers think this story...
4008    trump jr. is soon to give a 30-minute speech f...
4010    shanghai (reuters) - china said it plans to ac...
4011    vice president mike pence leaves nfl game beca...
Name: Body, Length: 3986, dtype: object
```

Figure 12: Converting the content in Body column into lowercase in DB2.



```
# Split the dataset into two smaller sets

x = db2['Body'].values
y = db2['Label'].values
```

```
vect = TfidfVectorizer()
vect.fit(x)

x = vect.transform(x)
print(x)
```

```
  (0, 45780)	0.01772238617014786
  (0, 45772)	0.013186990541178115
  (0, 45686)	0.009950521253935527
  (0, 45501)	0.034474322388693274
  (0, 45408)	0.02453342225139396
  (0, 45388)	0.01977959026969816
  (0, 45372)	0.018155299290937243
  (0, 45256)	0.04428999488818559
  (0, 45182)	0.023775201000804908
```

Figure 13: Converting all textual data in x dataset into vector format.

3.5 Model Development

3.5.1 Stacking

In Section 2.2, we have talked about what an ensemble model is, how they work and what are the advantages of using an ensemble model. To improve the detection of fake news, we are building an ensemble model using the top three classifiers used for fake news detection, using the method *Stacking*.

Stacking involves combining predictions made by different machine learning algorithms on the same dataset. Stacking is designed in a way to improve the model's performance, but the improvement is not guaranteed. Performance improvement depends on the problem's complexity, training data's efficiency and the choice of base models.

The architecture of stacking involves two kinds of models, base models and meta model. Two or more base models are fitted on the training data and their predictions are combined. One meta model learns to combine the predictions made by the base models.

It is better to use a range of complex base models, as combining models with diverse assumptions increases the performance of the stacked model. Linear models are used for meta models, like Linear Regression for regression tasks and Logistic Regression for classification tasks. [27]

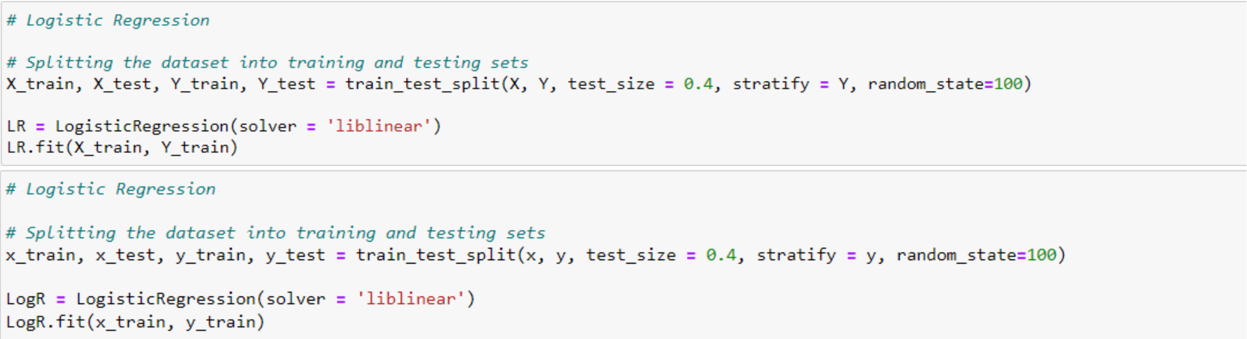
3.5.2 Models Developed

Logistic Regression

The logistic regression classifier is stored as *LR* for DB1 and *LogR* for DB2. We use the hyperparameter '*liblinear'* solver for Logistic Regression. '*Liblinear'* is an open-source library that is efficient for binary classifiers when using large datasets.[28]

The smaller sets X,Y for DB1 and x,y for DB2 are split into testing and training sets using the train_test_split(). The test size is set to 40% and is stratified to the size of Y and y respectively for DB1 and DB2.

LR and LogR are then fit with X_train, Y_train and x_train, y_train, respectively. Fig. 14 shows the code snippet for the implementation of Logistic Regression classifier.



```
# Logistic Regression

# Splitting the dataset into training and testing sets
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.4, stratify = Y, random_state=100)

LR = LogisticRegression(solver = 'liblinear')
LR.fit(X_train, Y_train)
```

```
# Logistic Regression

# Splitting the dataset into training and testing sets
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.4, stratify = y, random_state=100)

LogR = LogisticRegression(solver = 'liblinear')
LogR.fit(x_train, y_train)
```

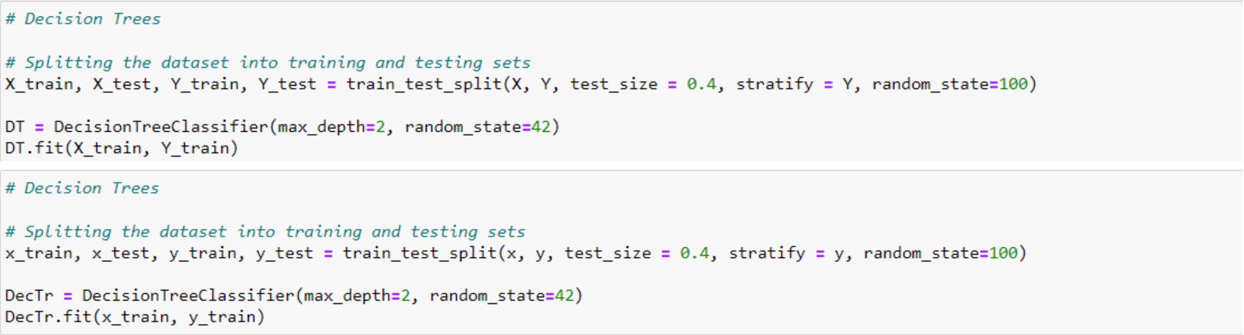
Figure 14: Logistic Regression code snippet

Decision Trees

The decision trees classifier is stored as *DT* and *DecTr* for DB1 and DB2, respectively. We use the *max_depth* and *random_state* hyperparameters for the decision tree classifier. *max_depth* is used to define the depth of the decision tree. The depth of the tree defines the complexity of the tree and creates a balance between generalization and fitting [29]. We have defined the *max_depth* parameter as 2. The *random_state* hyperparameter represents the number generated that is used to shuffle the data.

The smaller sets X,Y for DB1 and x,y for DB2 are split into testing and training sets using the train_test_split(). The test size is set to 40% and is stratified to the size of Y and y respectively for DB1 and DB2.

DT and DecTr are then fit with X_train, Y_train and x_train, y_train, respectively. Fig. 15 shows the code snippet for the implementation of Decision Trees classifier.



```
# Decision Trees

# Splitting the dataset into training and testing sets
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.4, stratify = Y, random_state=100)

DT = DecisionTreeClassifier(max_depth=2, random_state=42)
DT.fit(X_train, Y_train)
```

```
# Decision Trees

# Splitting the dataset into training and testing sets
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.4, stratify = y, random_state=100)

DecTr = DecisionTreeClassifier(max_depth=2, random_state=42)
DecTr.fit(x_train, y_train)
```

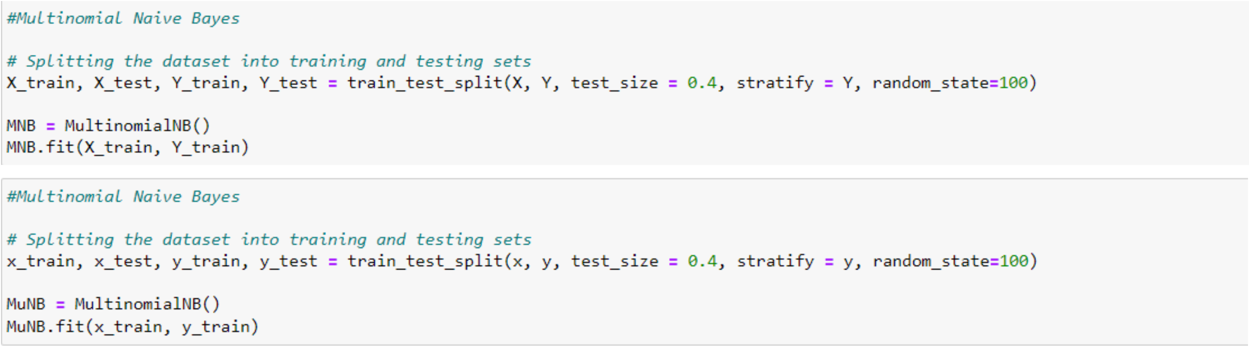
Figure 15: Decision Trees code snippet

Multinomial Naïve Bayes

The multinomial naïve bayes classifier is stored as *MNB* and *MuNB* for DB1 and DB2, respectively. There are no hyperparameters used for the Multinomial Naïve Byes classifier.

The smaller sets X,Y for DB1 and x,y for DB2 are split into testing and training sets using the train_test_split(). The test size is set to 40% and is stratified to the size of Y and y respectively for DB1 and DB2.

MNB and MuNB are then fit with X_train, Y_train and x_train, y_train, respectively. Fig. 16 shows the code snippet for the implementation of Multinomial Naïve Bayes classifier.



```
#Multinomial Naive Bayes

# Splitting the dataset into training and testing sets
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.4, stratify = Y, random_state=100)

MNB = MultinomialNB()
MNB.fit(X_train, Y_train)
```

```
#Multinomial Naive Bayes

# Splitting the dataset into training and testing sets
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.4, stratify = y, random_state=100)

MuNB = MultinomialNB()
MuNB.fit(x_train, y_train)
```

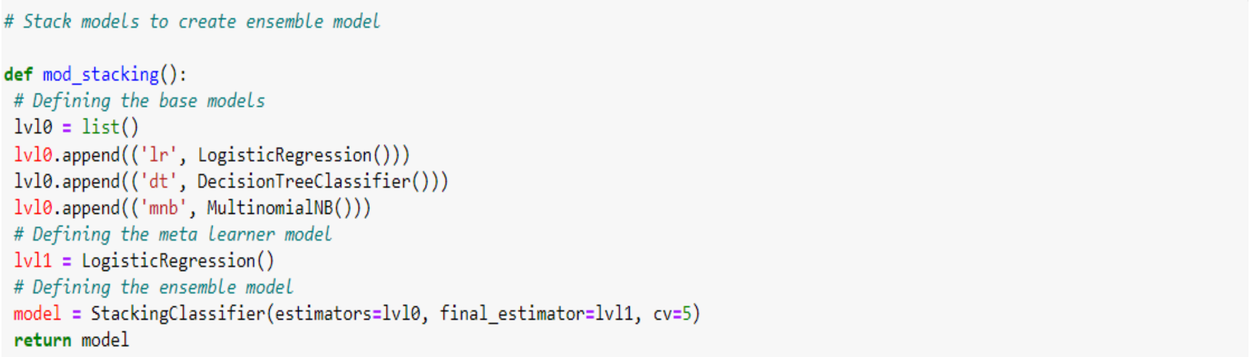
Figure 16: Multinomial Naive Bayes code snippet

Ensemble Model

After reviewing existing work done on fake news detection, it was concluded that the top three performing classifiers were Logistic Regression, Decision Trees, and Multinomial Naïve Bayes. This was also proven by experimenting with different classifiers using our datasets.

To develop our ensemble model, we are using the method of stacking. The base models for our model are Logistic Regression, Decision Trees and Multinomial Naïve Bayes classifiers, while Logistic Regression is used as the meta model. We define a function to stack the models. First, we create a list called *lvl0* to hold all the base models and append our base models to this list. We then define the meta model as *lvl1*. Using *lvl0* and *lvl1*, we build our ensemble model, and the function returns the model built. [30]

Fig. 17 shows the code snippet of how we have stacked the base models.



```
# Stack models to create ensemble model

def mod_stacking():
 # Defining the base models
 lvl0 = list()
 lvl0.append(('lr', LogisticRegression()))
 lvl0.append(('dt', DecisionTreeClassifier()))
 lvl0.append(('mnb', MultinomialNB()))
 # Defining the meta learner model
 lvl1 = LogisticRegression()
 # Defining the ensemble model
 model = StackingClassifier(estimators=lvl0, final_estimator=lvl1, cv=5)
 return model
```

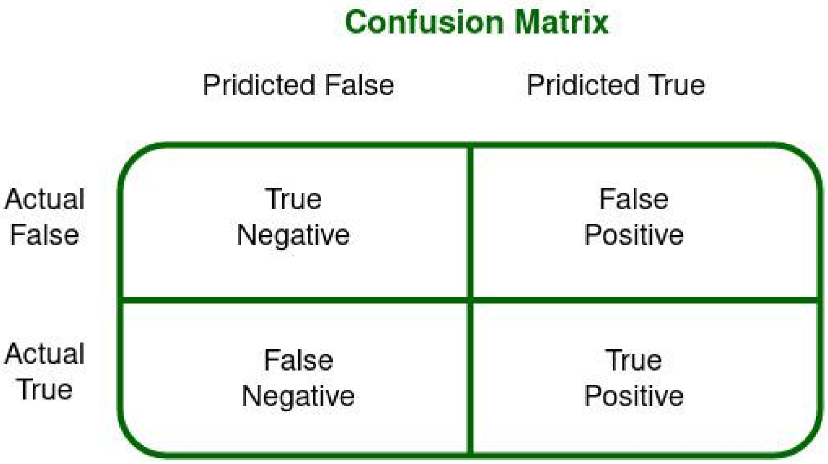
Figure 17: Code snippet of function to build a stacking ensemble model.

4. Testing and Performance Analysis

4.1 Evaluation Criteria

4.1.2 Confusion Matrix

It is a performance measurement metric that is used to summarize the performance of a machine learning algorithm. When the datasets are imbalanced and skewed, accuracy is not the ideal performance metric. A confusion matrix is the preferred metric in this situation[31]. The total number of accurate and inaccurate predictions are listed with each class of expected and actual values. It provides information on the errors and the different kinds of errors that are made.[32]



	Pridicted False	Pridicted True
Actual False	True Negative	False Positive
Actual True	False Negative	True Positive

Figure 18: Confusion Matrix [32]

4.1.1 Performance Metrics

Using the values achieved by plotting a confusion matrix, we can evaluate the model’s performance by calculating the following values:

Accuracy, Precision, Recall and F1 Score

Accuracy is defined as “The ratio of the total number of correct predictions to the total number of predictions.” But the number of false predictions made is not included in this ratio and therefore is not enough to determine the model’s performance, which is why we compare our model’s performance using other performance metrics as well.[33]

Precision is an indicator to a model’s performance , that is, the quality of the positive predictions made by the model. Precision is the ratio of true positives to the total number of positive predictions (the sum of true positives and false positives) made by the model.[34] A high precision value means that the classifier is extremely strict in classifying positive records, and very few false positives are recorded.[35]

$$\boldsymbol{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Figure 20: Accuracy formula [33]

$$\boldsymbol{Precision} = \frac{\boldsymbol{TP}}{\boldsymbol{TP + FP}}$$

Figure 19: Precision formula [33]

Recall is defined as “The ratio of true positives to the total number of positives predicted by the model.” Recall helps in reducing the number of false negatives and provides a measure of how accurately the model identifies the data. It is also known as ‘Sensitivity.’ [36] A high recall value shows that the classifier has identified very few false negative values.[35]

The harmonic mean achieved by combining precision and recall is known as F1 Score. It is usually used to compare the performance of two classifiers. The F1 score provides a clearer evaluation of a model’s performance as it uses both precision and recall performance of the model[37]. A higher F1 score signifies the quality of the classifier.

$$\frac{2(P * R)}{P + R}$$

P = the precision

R = the recall of the classification model

Figure 21: F1 Score formula. [37]

Specificity and Sensitivity

Specificity is the proportion of true negatives to the total number of negative predictions made.

Sensitivity is the proportion of true positives to the total number of positive predictions made. Sensitivity is also known as ‘Recall.’

$$\boldsymbol{Specificity}\ (Proportion\ of\ True\ Negatives) = \frac{TN}{TN + FP}$$

$$\boldsymbol{Sensitivity}\ (Proportion\ of\ True\ Positives\ or\ Recall)\ = \frac{TP}{TP + FN}$$

Figure 22: Sensitivity and Specificity formulas. [2]

ROC Curve

The Receiver Operating Characteristic (ROC) Curve is a plot that displays the graph between the True Positives and the False Positives for a binary classifier. The Area Under Curve (AUC) summarizes the overall performance of the classifier. AUC values range from 0 to 1, with 0 meaning the model got all predictions wrong while 1 means the model got all predictions right. [38]

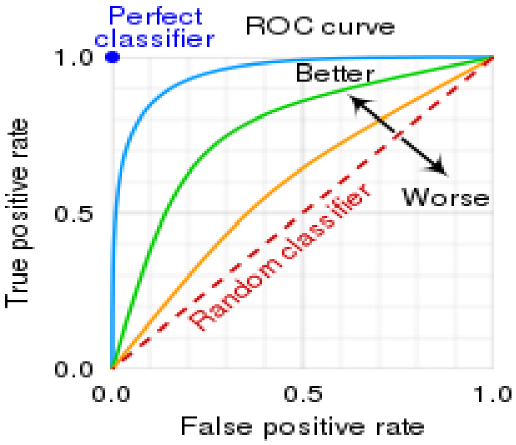
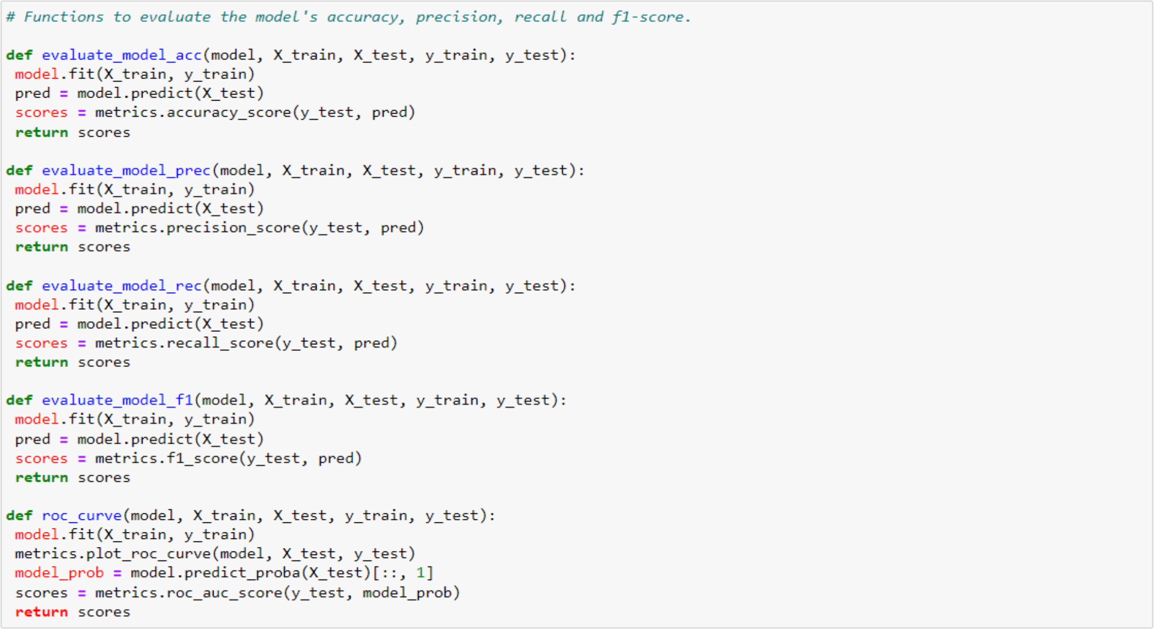


Figure 23: ROC Curve [38]

4.2 Model Testing

To evaluate the performance of the ensemble model, we define functions that calculate the evaluation metrics. The functions have the parameters *model*, *X_train*, *y_train*, *X_test and y_test*. The model is fit using the parameters *X_train* and *y_train*. Using the *predict()* function, the model's test target value is predicted. The performance metric value is then calculated and returned. Fig 24. shows how the performance metrics of the ensemble model have been calculated.

For each dataset DB1 and DB2, the smaller sets X,Y for DB1 and x,y for DB2 are split into testing and training sets using the train_test_split(). The test size is set to 40% and is stratified to the size of Y and y respectively for DB1 and DB2.



```
# Functions to evaluate the model's accuracy, precision, recall and f1-score.

def evaluate_model_acc(model, X_train, X_test, y_train, y_test):
 model.fit(X_train, y_train)
 pred = model.predict(X_test)
 scores = metrics.accuracy_score(y_test, pred)
 return scores

def evaluate_model_prec(model, X_train, X_test, y_train, y_test):
 model.fit(X_train, y_train)
 pred = model.predict(X_test)
 scores = metrics.precision_score(y_test, pred)
 return scores

def evaluate_model_rec(model, X_train, X_test, y_train, y_test):
 model.fit(X_train, y_train)
 pred = model.predict(X_test)
 scores = metrics.recall_score(y_test, pred)
 return scores

def evaluate_model_f1(model, X_train, X_test, y_train, y_test):
 model.fit(X_train, y_train)
 pred = model.predict(X_test)
 scores = metrics.f1_score(y_test, pred)
 return scores

def roc_curve(model, X_train, X_test, y_train, y_test):
 model.fit(X_train, y_train)
 metrics.plot_roc_curve(model, X_test, y_test)
 model_prob = model.predict_proba(X_test)[::, 1]
 scores = metrics.roc_auc_score(y_test, model_prob)
 return scores
```

Figure 24: Code snippet showing the calculation of performance metrics.

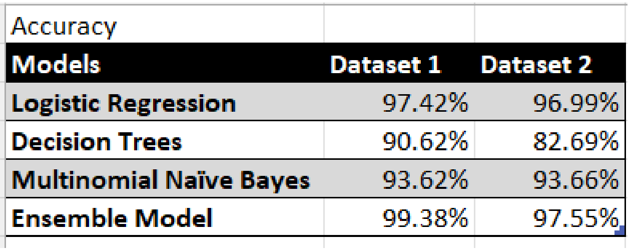
4.3 Model Evaluation

4.3.1 Accuracy

For DB1, the accuracy achieved by using Logistic Regression, Decision Trees and Multinomial Naïve Bayes are 97.42%, 90.62% and 93.62% respectively. Using the ensemble model on DB1, the accuracy is 99.38%.

For DB2, the accuracy achieved by using Logistic Regression, Decision Trees and Multinomial Naïve Bayes are 96.99%, 82.69% and 93.66% respectively. Using the ensemble model on DB2, the accuracy is 97.55%.

Clearly, the ensemble model has achieved higher accuracy on the datasets than the individual classifiers trained on the datasets. Table 3 shows the accuracy comparison between the models and datasets.



Accuracy		
Models	**Dataset 1**	**Dataset 2**
Logistic Regression	97.42%	96.99%
Decision Trees	90.62%	82.69%
Multinomial Naïve Bayes	93.62%	93.66%
Ensemble Model	99.38%	97.55%

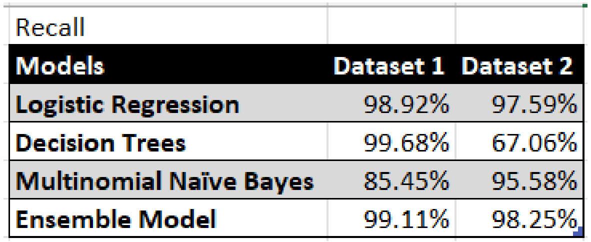
Table 3: Accuracy comparison

4.3.2 Recall

For DB1, the individual models achieved a recall of 98.92%, 99.68% and 85.45% for Logistic Regression, Decision Trees and Multinomial Naïve Bayes, respectively. The ensemble model achieved a recall score of 99.11%. In terms of recall on DB1, Decision Trees performed slightly better than the ensemble model.

For DB2, the individual models achieved a recall of 97.59%, 67.06% and 95.58% for Logistic Regression, Decision Trees and Multinomial Naïve Bayes, respectively. The ensemble model achieved a recall score of 98.25%. In DB2, the ensemble model performed better than the individual classifiers in terms of recall.

Table 4 shows the recall comparison between the models and datasets.



Recall

Models	Dataset 1	Dataset 2
Logistic Regression	98.92%	97.59%
Decision Trees	99.68%	67.06%
Multinomial Naïve Bayes	85.45%	95.58%
Ensemble Model	99.11%	98.25%

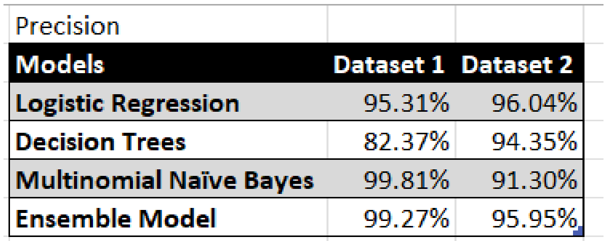
Table 4: Recall comparison

4.3.3 Precision

For DB1, the individual classifiers achieved a precision score of 95.31%, 82.37% and 99.81% for Logistic Regression, Decision Trees and Multinomial Naïve Bayes, respectively. The ensemble model achieved a precision score of 99.27% on DB1. The Multinomial Naïve Bayes classifier performed slightly better than the ensemble model in terms of precision on DB2.

For DB2, the individual classifiers achieved a precision score of 96.04%, 94.35% and 91.30% for Logistic Regression, Decision Trees and Multinomial Naïve Bayes, respectively. The ensemble model achieved a precision score of 95.95% on DB2. The Decision Trees classifier performed slightly better than the ensemble model in terms of precision on DB2.

Table 5 shows the precision comparison between the models and datasets.



Precision

Models	Dataset 1	Dataset 2
Logistic Regression	95.31%	96.04%
Decision Trees	82.37%	94.35%
Multinomial Naïve Bayes	99.81%	91.30%
Ensemble Model	99.27%	95.95%

Table 5: Precision comparison

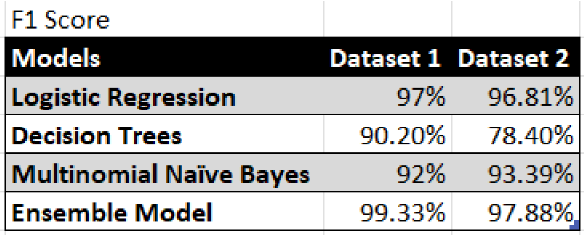
4.3.4 F1 Score

For DB1, the F1 scores were 97%, 90.2% and 92% for Logistic Regression, Decision Trees and Multinomial Naïve Bayes, respectively. The ensemble model achieved a F1 score of 99.33% on DB1.

For DB2, the F1 scores were 96.81%, 78.4% and 93.39% for Logistic Regression, Decision Trees and Multinomial Naïve Bayes, respectively. The ensemble model achieved a F1 score of 97.88% on DB2.

The ensemble model outperformed the individual classifiers on both the datasets in terms of F1 score.

Table 6 shows the F1 score comparison between the models and the datasets.



F1 Score

Models	Dataset 1	Dataset 2
Logistic Regression	97%	96.81%
Decision Trees	90.20%	78.40%
Multinomial Naïve Bayes	92%	93.39%
Ensemble Model	99.33%	97.88%

Table 6: F1 Score comparison.

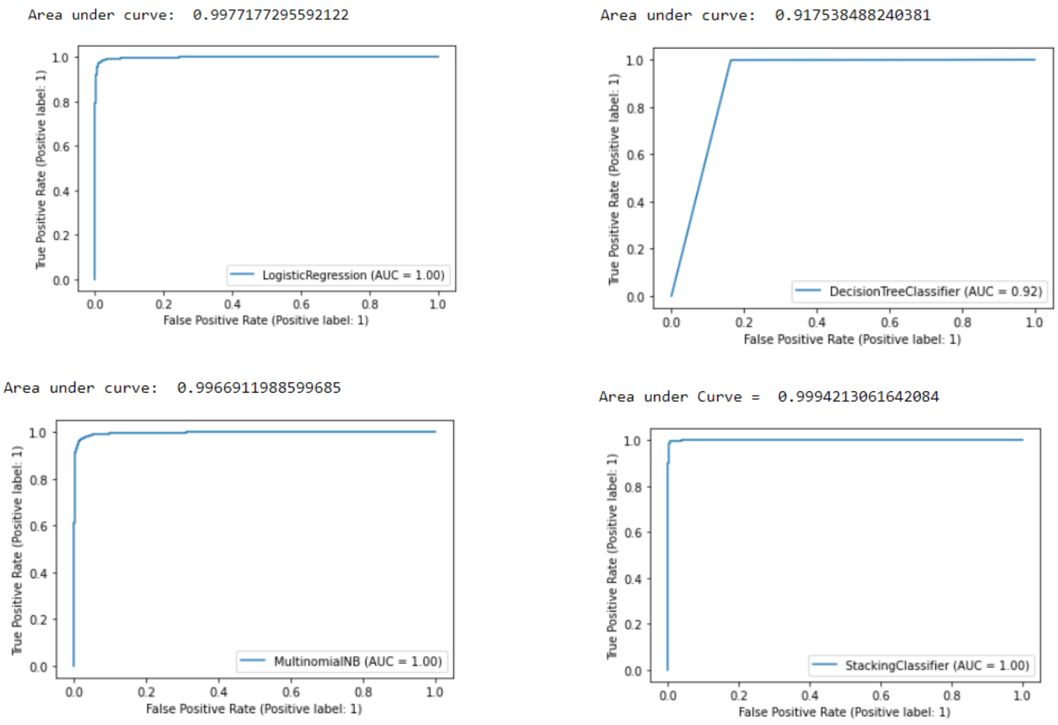
4.3.5 ROC Curve and AUC Score

For DB1, the AUC scores for the individual classifiers were 0.997, 0.917 and 0.996 for Logistic Regression, Decision Trees and Multinomial Naïve Bayes, respectively. The ensemble model had an AUC score of 0.999 for DB1.

For DB2, the AUC scores for the individual classifiers were 0.996, 0.898 and 0.985 for Logistic Regression, Decision Trees and Multinomial Naïve Bayes, respectively. The ensemble model had an AUC score of 0.997 for DB2.

The ensemble models outperformed the individual classifiers on both datasets in terms of AUC score.

Fig. 25 shows the AUC scores and the ROC curves for the classifiers on DB1 while Fig. 26 shows the AUC scores and the ROC curves for the classifiers on DB2.



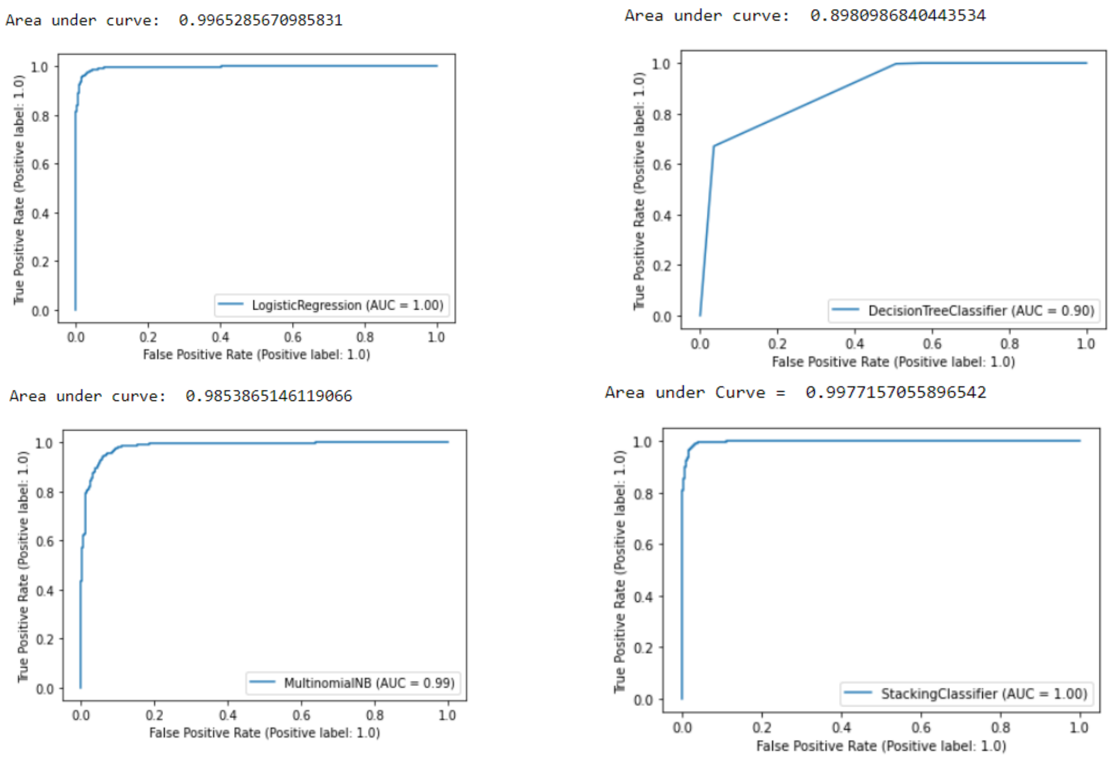
```
Area under curve:  0.9977177295592122
```

```
Area under curve:  0.917538488240381
```

```
Area under curve:  0.9966911988599685
```

```
Area under Curve =  0.9994213061642084
```

Figure 25: ROC Curves and AUC scores for DB1



```
Area under curve:  0.9965285670985831
```

```
Area under curve:  0.8980986840443534
```

```
Area under curve:  0.9853865146119066
```

```
Area under Curve =  0.9977157055896542
```

Figure 26: ROC Curves and AUC scores for DB2

4.4 Comparative Analysis

Numerous experiments and studies have been conducted towards the detection of fake news. In this section, we will be comparing our ensemble model's performance against the performance of existing ensemble models used to detect fake news, discussed in Section 2.3.

The performance metrics of our model such as accuracy, precision, recall and F1 scores achieved on DB1 will be compared to that of the existing models. Table 7 shows the comparison between our model and existing models.

Author	Accuracy	Precision	Recall	F1 Score
Our ensemble model (LR+DT+MNB)	**99.38%**	**99.27%**	**99.11%**	**99.33%**
Iftikhar et al. [7] (LR+RF+KNN)	88%	88%	89%	88%
Iftikhar et al. [7] (LR+LSVM+CART)	86%	86%	87%	86%
Monika et al. [22] (BerConvoNet)	97.45%	96.85%	98.25%	97.54%
Jamal et al. [23] (CNN+RNN)	99.02%	99%	99%	99%

Table 7: Performance comparison between our model and existing models

The main difference between our approach and the existing approaches is that the other authors, while preprocessing the data remove stopwords, punctuation, tense etc. while our approach does not remove any words. This is because, my personal opinion is that the stopwords, punctuation etc. add the element of "emotiveness" to the article, which is an important factor in deceiving people. Removing the emotion from the article might make the news look true, which means there is a low chance of the model detecting the fake article.

The models compared above use the same dataset as our project (DB1 [25]). Comparing the performances of the other ensemble models on the same dataset, our ensemble model has performed significantly better than the other models, except for the models using Neural Networks. ***Jamal et al.***[23] has the closest performance to our model, mainly because the authors have made use of Neural Networks. Finetuning the Neural Networks using hyperparameters has helped their model to perform optimally. The models developed by ***Iftikhar et al.*** [7] performs marginally lower than our ensemble model due to the data being preprocessed multiple times to remove stopwords.

The technique of not removing the stopwords, punctuation, tenses etc. and enough optimizing time to test out different hyperparameters has been crucial for our ensemble model to perform better than the existing ensemble models.

5. Conclusions

5.1 Achievements

This project was undertaken with the aim of designing an ensemble model that could detect fake news from a given set of news articles. The development stages included finding large datasets with a good mix of true and fake news articles and news articles from multiple domains, feature extraction and building an ensemble model that would provide high performance.

Using our ensemble model, in DB1 we have achieved scores of 99.38%, 99.27%, 99.11%, 99.33% and 99.9% for accuracy, precision, recall, F1 scores and AUC score respectively while in DB2, we have achieved scores of 97.55%, 95.95%, 98.25%, 97.88% and 99.7% for accuracy, precision, recall, F1 scores and AUC scores, respectively. We can conclude that our ensemble model has outperformed the individual classifiers in terms of overall performance. We have also observed that our ensemble model has higher performance than existing ensemble models that are used to detect fake news, and with further finetuning, the model can achieve maximum potential.

5.2 Limitations

The development of the project faced a set of challenges that caused hinderance to the project in the development stages. Due to time constraints in submitting other course's courseworks, the project development fell slightly behind the planned schedule. Due to this delay, much time could not be allocated to experimentation on the ensemble model using hyperparameters. Even though our ensemble model produced results of 99% and above for each performance metric, I believe that with more testing and finetuning, the ensemble model could have yielded better results.

The ensemble model currently only detects text-based fake news. With more time, the ensemble model could have been trained to detect media-based fake news as well.

5.3 Future Works

The project can be enhanced a lot more, by allocating more time and effort. The ensemble model can be fine-tuned and evaluated using different hyperparameters to achieve better results. Further, the datasets used had few features only. Using datasets with more features can help achieve better results.

The ensemble model currently detects text-based fake news only. Integrating Convolutional Neural Networks (CNN) with our ensemble model can help detect media-based fake news as well in the future.

Currently, the ensemble model is not accessible for users as there is no interface for them to interact with. In the future, the ensemble model can be integrated with webpages, where users can check if articles are true or fake.

References

[1] V. Rubin, Y. Chen, and N. Conroy, "Deception detection for news: Three types of fakes," *Proceedings of the Association for Information Science and Technology*, vol. 52, pp. 1-4, 2015.

[2] A. Barrón-Cedeño, G. Da San Martino, I. Jaradat, and P. Nakov, "Proppy: A System to Unmask Propaganda in Online News," *Proceedings of the AAAI Conference on Artificial Intelligence,* vol. 33, no. 01, pp. 9847-9848, 07/17 2019, doi: 10.1609/aaai.v33i01.33019847.

[3] G. Gravanis, A. Vakali, K. Diamantaras, and P. Karadais, "Behind the cues: A benchmarking study for fake news detection," *Expert Systems with Applications*, vol. 128, pp. 201-213, 2019.

[4] S. Ishfaq Manzoor, J. Singla, and Nikita, "Fake News Detection Using Machine Learning approaches: A systematic Review.," presented at the Third International Conference on Trends in Electronics and Informatics, Tirunelveli, India, 2019. [Online]. Available: https://www.researchgate.net/publication/336436870_Fake_News_Detection_Using_Machine_Learning_approaches_A_systematic_Review.

[5] V. Jayakumar, N. Kumar, and N. Himanshu, "Fake News Detection using ML," *Turkish Journal of Computer and Mathematics Education,* vol. Vol 12, no. Issue 11, pp. 2943-2947, 2021.

[6] A. Jain, A. Shakya, H. Khatter, and A. Gupta, "A smart System for Fake News Detection Using Machine Learning," presented at the International Conference on Issues and Challenges in Intelligent Computing Techniques, Ghaziabad, India, 2019. [Online]. Available: https://www.researchgate.net/publication/339022255_A_smart_System_for_Fake_News_Detection_Using_Machine_Learning.

[7] I. Ahmed, M. Yousaf, S. Yousaf, and M. Ovais Ahmad, "Fake News Detection Using Machine Learning Ensemble Methods," *Complexity* vol. 2020, p. 11, 2020.

[8] A. Thota, P. Tilka, S. Ahluwalia, and N. Lohia, "Fake News Detection: A Deep Learning Approach," *SMU Data Science Review,* vol. 1, no. 3, 2018.

[9] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, "Fake News Detection on Social Media: A Data Mining Approach," *ACM SIGKDD Explorations Newsletter,* vol. 19, no. 1, pp. 22-36, 2017, doi: https://doi.org/10.1145/3137597.3137600.

[10] E. Tandoc, Z. Wei Lim, and R. Ling, "Defining "Fake News": A typology of scholarly definitions," *Digital Journalism,* vol. 6, pp. 1-17, 2017, doi: 10.1080/21670811.2017.1360143.

[11] V. L. Rubin, N. Conroy, Y. Chen, and S. Cornwell, "Fake news or truth? using satirical cues to detect potentially misleading news," in *Proceedings of the second workshop on computational approaches to deception detection*, 2016, pp. 7-17.

[12] I. Ermida, *News satire in the press: Linguistic construction of humour*, C. Scholars, ed., 2012, pp. 185-210.

[13] S. E. Herman and N. Chomsky, *Manufacturing Consent: The political economy*

of the mass media. New York: Pantheon Books, 2002.

[14] S. Brown, "Machine Learning, Explained," M. Sloan, Ed., ed. Cambridge: MIT, 2021.

[15] IBM. "What is logistic regression?" https://www.ibm.com/topics/logistic-regression (accessed 30/11/2022).

[16] Z.-H. Zhou, "Decision trees," *Machine Learning*, pp. 79-102, 2021.

[17] M. J. Islam, Q. J. Wu, M. Ahmadi, and M. A. Sid-Ahmed, "Investigating the performance of naive-bayes classifiers and k-nearest neighbor classifiers," in *2007 international conference on convergence information technology (ICCIT 2007)*, 2007: IEEE, pp. 1541-1546.

[18] R. Gandhi. "Naive Bayes Classifier." https://towardsdatascience.com/naive-bayes-classifier-81d512f50a7c (accessed 30/11/2022).

[19] S. Saxena. "Introduction to Naive Bayes Algorithm." Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/03/introduction-to-naive-bayes-algorithm/ (accessed 20/04/2023).

[20] V. Kotu and B. Deshpande, "Chapter 2 - Data Mining Process," in *Predictive Analytics and Data Mining*, V. Kotu and B. Deshpande Eds. Boston: Morgan Kaufmann, 2015, pp. 17-36.

[21] J. K. Burgoon, J. P. Blair, T. Qin, and J. F. Nunamaker, "Detecting deception through linguistic analysis," in *International Conference on Intelligence and Security Informatics*, 2003: Springer, pp. 91-101. [Online]. Available: https://www.researchgate.net/publication/221247057_Detecting_Deception_through_Linguistic_Analysis

[22] M. Choudhary, S. S. Chouhan, E. Pilli, and S. K. Vipparthi, "*BerConvoNet:* A deep learning framework for fake news classification," *Applied Soft Computing*, vol. 110, 2021, doi: https://doi.org/10.1016/j.asoc.2021.107614.

[23] J. A. Nasir, O. S. Khan, and I. Varlamis, "Fake news detection: A hybrid CNN-RNN based deep learning approach," *International Journal of Information Management Data Insights*, vol. 1, no. 1, 2021, doi: https://doi.org/10.1016/j.jjimei.2020.100007.

[24] S. Hakak, M. Alazab, S. Khan, T. R. Gadekallu, P. K. Maddikunta, and W. Z. Khan, "An ensemble machine learning approach through effective feature extraction to classify fake news," *Future Generation Computer Systems*, vol. 117, pp. 47-58, 2021, doi: https://doi.org/10.1016/j.future.2020.11.022.

[25] Kaggle. *Fake News*,

[26] Kaggle. *Fake News Detection*,

[27] J. Brownlee. "Stacking Ensemble Machine Learning with Python." Machine Learning Mastery. https://machinelearningmastery.com/stacking-ensemble-machine-learning-with-python/ (accessed 26/03/2023).

[28] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A Library for Large Linear Classification," *J. Mach. Learn. Res.*, vol. 9, pp. 1871–1874, 2008.

[29] I. Bernardo. "5 Decision Tree Hyperparameters to Enhance your Tree Algorithms." https://towardsdatascience.com/5-decision-tree-hyperparameters-to-enhance-your-tree-algorithms-aee2cebe92c8 (accessed 22/04/2023).

[30] J. Brownlee. "A Gentle Introduction to Ensemble Learning Algorithms." https://machinelearningmastery.com/tour-of-ensemble-learning-algorithms/ (accessed 30/11/2022).

[31] Nitin. "Confusion Matrix in Machine Learning." https://www.geeksforgeeks.org/confusion-matrix-machine-learning/ (accessed 30/11/2022).

[32] Geetansh. "How to calculate precision in R programming." https://www.geeksforgeeks.org/how-to-calculate-precision-in-r-programming/ (accessed 30/11/2022).

[33] J. Jordan. "Evaluating a machine learning model." https://www.jeremyjordan.me/evaluating-a-machine-learning-model/#:~:text=Accuracy%20is%20defined%20as%20the,the%20number%20of%20total%20predictions. (accessed 30/11/2022).

[34] C. AI. "Precision." https://c3.ai/glossary/machine-learning/precision/ (accessed 20/04/2023).

[35] J. Villalobos. "Precision vs Recall." https://www.brainstobytes.com/precision-vs-recall/ (accessed 20/04/2023).

[36] C. AI. "Recall." C3 AI. https://c3.ai/glossary/data-science/recall/ (accessed 20/04/2023).

[37] "What is the F1-Score?" https://www.educative.io/answers/what-is-the-f1-score (accessed 30/11/2022).

[38] C. AI. "Receiver Operating Characteristic Curve." C3 AI. https://c3.ai/glossary/data-science/receiver-operating-characteristic-roc-curve/ (accessed 20/04/2023).

Appendix

Project Management

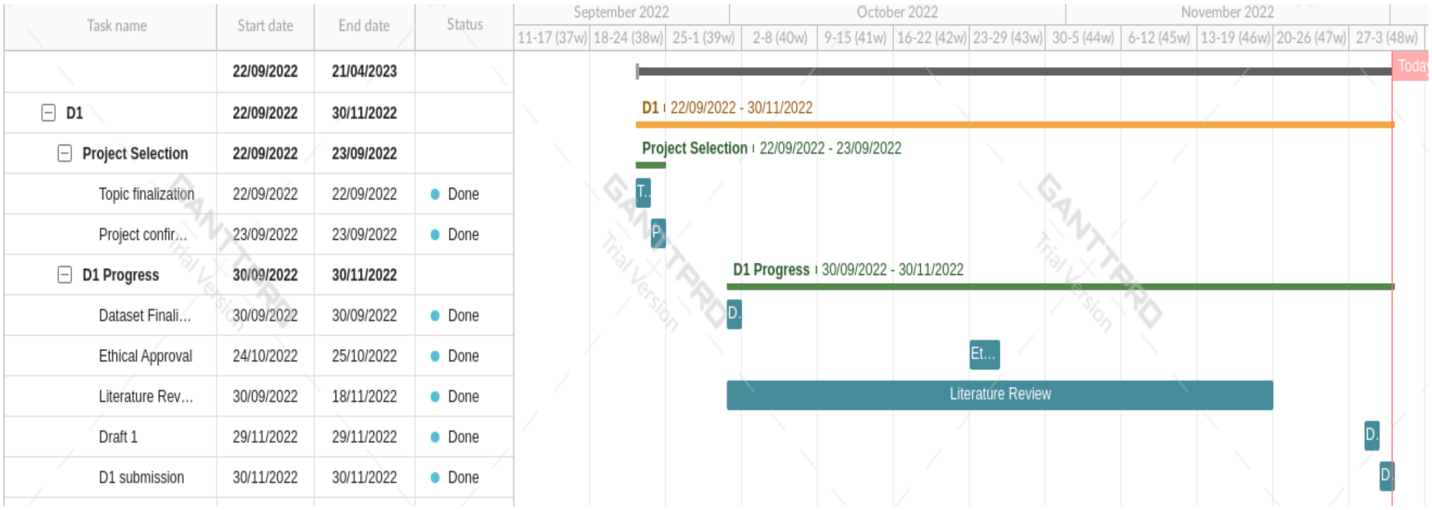
Project Methodology

In this project where we design an ensemble model, an industry-standard methodology is desirable. So we have chosen SCRUM, which is an Agile development methodology. SCRUM is an iterative process with frequent deployments. Weekly meetings will be organized with the SCRUM Master where the progress of the project, any issues faced during development and feedback on progress will be discussed.

Project Plan

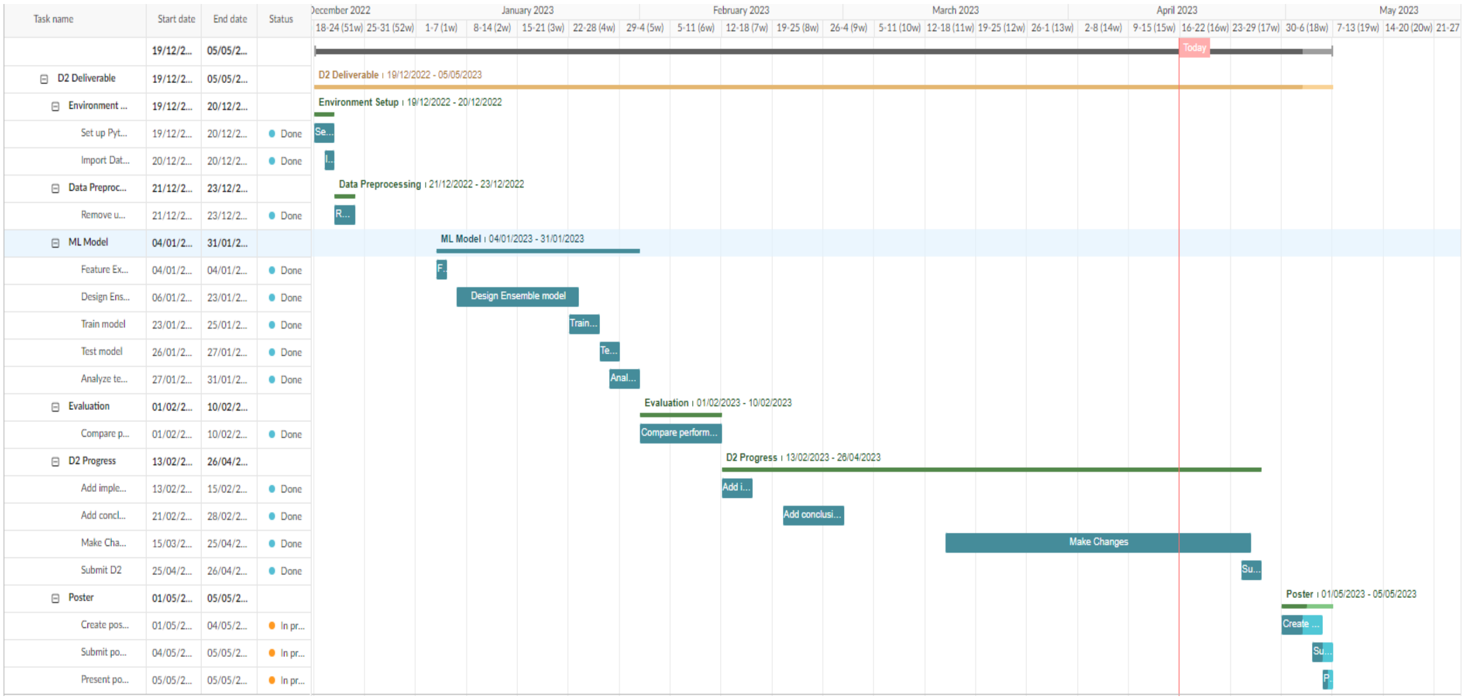
The project development plan has been described using a Gantt Chart. The Gantt chart below shows the timeline that has been followed till the submission of D1 and the timeline that will be followed till submission of D2 and poster. Weekly meetings will be held with the Project Supervisor to discuss progress and to ensure task completion. The chart displays the duration of each task. Extra time has been allotted for each task to accommodate any delays, problems and to complete other courseworks.

Figure displays the project plan for semester 1 while Figure displays the project plan for semester 2. The project plan has been divided broadly into – Environment establishment, Data Preprocessing, ML model development, Evaluation, Documentation and Demonstration.



Task name	Start date	End date	Status
	22/09/2022	**21/04/2023**	
⊟ **D1**	**22/09/2022**	**30/11/2022**	
⊟ **Project Selection**	**22/09/2022**	**23/09/2022**	
Topic finalization	22/09/2022	22/09/2022	● Done
Project confir...	23/09/2022	23/09/2022	● Done
⊟ **D1 Progress**	**30/09/2022**	**30/11/2022**	
Dataset Finali...	30/09/2022	30/09/2022	● Done
Ethical Approval	24/10/2022	25/10/2022	● Done
Literature Rev...	30/09/2022	18/11/2022	● Done
Draft 1	29/11/2022	29/11/2022	● Done
D1 submission	30/11/2022	30/11/2022	● Done

Figure 27 : Gantt Chart for Semester 1



Task name	Start date	End date	Status
	19/12/2...	05/05/2...	
⊟ D2 Deliverable	19/12/2...	05/05/2...	
⊟ Environment ...	19/12/2...	20/12/2...	
Set up Pyt...	19/12/2...	20/12/2...	● Done
Import Dat...	20/12/2...	20/12/2...	● Done
⊟ Data Preproc...	21/12/2...	23/12/2...	
Remove u...	21/12/2...	23/12/2...	● Done
⊟ ML Model	04/01/2...	31/01/2...	
Feature Ex...	04/01/2...	04/01/2...	● Done
Design Ens...	06/01/2...	23/01/2...	● Done
Train model	23/01/2...	25/01/2...	● Done
Test model	26/01/2...	27/01/2...	● Done
Analyze te...	27/01/2...	31/01/2...	● Done
⊟ Evaluation	01/02/2...	10/02/2...	
Compare p...	01/02/2...	10/02/2...	● Done
⊟ D2 Progress	13/02/2...	26/04/2...	
Add imple...	13/02/2...	15/02/2...	● Done
Add concl...	21/02/2...	28/02/2...	● Done
Make Cha...	15/03/2...	25/04/2...	● Done
Submit D2	25/04/2...	26/04/2...	● Done
⊟ Poster	01/05/2...	05/05/2...	
Create pos...	01/05/2...	04/05/2...	● In pr...
Submit po...	04/05/2...	05/05/2...	● In pr...
Present po...	05/05/2...	05/05/2...	● In pr...

December 2022
January 2023
February 2023
March 2023
April 2023
May 2023
18-24 (51w) 25-31 (52w) 1-7 (1w) 8-14 (2w) 15-21 (3w) 22-28 (4w) 29-4 (5w) 5-11 (6w) 12-18 (7w) 19-25 (8w) 26-4 (9w) 5-11 (10w) 12-18 (11w) 19-25 (12w) 26-1 (13w) 2-8 (14w) 9-15 (15w) 16-22 (16w) 23-29 (17w) 30-6 (18w) 7-13 (19w) 14-20 (20w) 21-27
Today
D2 Deliverable | 19/12/2022 - 05/05/2023
Environment Setup | 19/12/2022 - 20/12/2022
Se...
I...
Data Preprocessing | 21/12/2022 - 23/12/2022
R...
ML Model | 04/01/2023 - 31/01/2023
F
Design Ensemble model
Train...
Te...
Anal...
Evaluation | 01/02/2023 - 10/02/2023
Compare perform...
D2 Progress | 13/02/2023 - 26/04/2023
Add i...
Add conclusi...
Make Changes
Su...
Poster | 01/05/2023 - 05/05/2023
Create ...
Su...
P

Figure 28: Gantt Chart for Semester 2

Risk Analysis

During implementation of a project, certain issues can occur that can delay or derail the project development. It is important to understand these risks and have appropriate mitigation plans in place to overcome these risks with minimal damage to the project development.

1. **Risk Identification :** This step involves identifying and recognizing possible risks that can occur during project development.
2. **Risk Analysis :** This step involves analyzing the impact of the risk and the likelihood of the risk occurring during project development.
3. **Mitigation Plan :** Strategies to be followed if the project faces risk circumstances.
4. **Risk Monitoring :** Regular monitoring of the risks based on their likelihood to occur.

Table 8: Risk Analysis Classification

Likelihood	Impact
High	Catastrophic
Medium	Serious
Low	Tolerable

Table 9: Risk Planning

S. No.	Risk	Likelihood	Impact	Mitigation Plan
1	Loss of code	Low	Catastrophic	Have backup of code on GitHub
2	Low quality dataset	Low	Serious	Choose dataset with appropriate attributes and quality data
3	High processing time	Medium	Tolerable	Remove unwanted attributes from dataset
4	Difficulty in model implementation	Low	Catastrophic	Perform proper research and get feedback from supervisor to solve issues
5	Low performance of model	Medium	Serious	Test model frequently after tuning to achieve optimal results
6	Tasks incomplete	Medium	Serious	Revise project plan and dedicate time for project
7	Personal Issues (Health/Work)	Medium	Serious	Allocate new deadlines and revise project plan

Professional, Social, Ethical and Logical Issues

Professional and Legal Issues

All research papers, journals, articles, conference papers used for literature review have been referenced appropriately. Any code that is being utilized from open-source works will be acknowledged. The datasets used for the implementation of this model are open-sourced and publicly available.

Social and Ethical Issues

The datasets used are open-source and publicly available datasets that contain no personal or confidential information of any human subject. This project shall not be harmful to society or the environment. This project aims to improve the detection of fake news and assist the general public regarding the authenticity of a news article.