A Comparative Study of Majority and Weighted Voting Ensemble Techniques for Improving the Accuracy of False Information Detection

A DISSERTATION

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE
OF

MASTER OF TECHNOLOGY
IN
INFORMATION SYSTEMS

Submitted by

MAYUR

2K21/ISY/15

Under the supervision of

Prof. DINESH K VISHWAKARMA

DEPARTMENT OF INFORMATION TECHNOLOGY
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi 110042

JUNE, 2023

DEPARTMENT OF INFORMATION TECHNOLOGY
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

CANDIDATE'S DECLARATION

I, MAYUR, Roll No – 2K21/ISY/15 students of M.Tech (Department of Information Technology), hereby declare that the project Dissertation titled "A Comparative Study of Majority and Weighted Voting Ensemble Techniques for Improving the Accuracy of False Information Detection" which is submitted by me to the Department of Information Technology, Delhi Technological University, Delhi in partial fulfilment of the requirement for the award of degree of Master of Technology, is original and not copied from any source without proper citation. This work has not previously formed the basis for the award of any Degree, Diploma Associateship, Fellowship or other similar title or recognition.

Place: Delhi MAYUR

Date: 01.06.2023

DEPARTMENT OF INFORMATION TECHNOLOGY
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

CERTIFICATE

I hereby certify that the Project Dissertation titled "A Comparative Study of Majority and Weighted Voting Ensemble Techniques for Improving the Accuracy of False Information Detection" which is submitted by MAYUR, Roll No – 2K21/ISY/15, Department of Information Technology, Delhi Technological University, Delhi in partial fulfilment of the requirement for the award of the degree of Master of Technology, is a record of the project work carried out by the students under my supervision. To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Place: Delhi

Prof. DINESH K VISHWAKARMA

Date: 01.06.2023

SUPERVISOR

DEPARTMENT OF INFORMATION TECHNOLOGY
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

ACKNOWLEDGEMENT

I wish to express my sincerest gratitude to Prof. DINESH K VISHWAKARMA for his continuous guidance and mentorship that he provided me during the project. He showed me the path to achieve my targets by explaining all the tasks to be done and explained to me the importance of this project as well as its industrial relevance. He was always ready to help me and clear my doubts regarding any hurdles in this project. Without his constant support and motivation, this project would not have been successful.

Place: Delhi MAYUR

Date: 01.06.2021

Abstract

A method of information dissemination that had never been seen before the digital era has emerged as a result of the development of the World Wide Web and the use of social media platforms like Facebook, Twitter & Instagram. Through these social media sites, a significant quantity of information may be fraudulent. Therefore, it is necessary to monitor this data. By feeding a new article or fact to the model, we may use a technique called machine learning-based fake news identification to determine its veracity. We will preprocess the data (in this example, text) before training the dataset. Preprocessing mostly entails eliminating redundant data. The dataset is then divided into two sections for testing and training. The data will next be vectorized using a variety of vectorization methods, including Countvectorizer, TF-IDF vectorizer, and n-grams. The various classifiers (such as random forest, decision tree, logistic regression etc.) are then trained using the vectorized data. The accuracy of cutting-edge false news detection is then increased by incorporating these results into ensemble models. The timeline of a dataset has an impact on how accurate the model is since newer information is not included in older datasets, making it impossible for the model to effectively forecast the veracity of newer information. After training, the model may be quickly tested using the testing dataset, and it will then be ready for usage.

Contents

List of Tables

List of Figures

List of Abbreviations

LR	Logistic Regression
DT	Decision Tree
RF	Random Forest
NBC	Naive Bayes Classifier
AUC	Area Under Curve
KNN	K Nearest Neighbors
XGB	XG Boost
CNN	Convolutional Neural Network
BERT	Bidirectional Encoder Representations from Transformers
LSTM	Long short-term memory
GBC	Gradient Boosting Classifier
VGG	Visual Geometry Group

Chapter 1

INTRODUCTION

From the start of the 21st century when social media came into existence, it has been used as a medium of communication and a variety of other things by the young as well as the old generation. They serve as a marketing medium and also compete with different well known newspapers as well as media agencies. For a record there are roughly 2.95 billion active monthly users on Facebook in the third quarter of 2021. During the first quarter of 2021 Meta stated that the company has 3.51 billion people who are using at least one of its platforms. Which as compared to 2008 (where there were only 100 million users) is a very high number. Social media networks provide a medium to both news agencies as well as the general public to post anything they want, which makes it an easy medium to spread the fake news and false content among the people.[1] The main problem is that there are no such accurate tools which can give us a warning about the truthfulness of information. Almost all the time fake news influences the people in a wrong manner and makes them do things which they are not supposed to do, like in the 2016 and 2008 US Presidential elections many people were influenced with the fake news all over the social media networks. Fake news changes the way an individual thinks about something and it sometimes can be very harmful for society.[2]

1.1 Classifications of Fake News

News can be either true or fake but there is a classification for Fake news which further classifies it into some new categories as follows:

- Clickbait: These are the stories which can be exaggerated or can be completely false. These stories are designed for a purpose to increase the ad revenue and make money.

- Propaganda: These types of articles are basically to promote the author's agenda and can be fake or deceptive. Such articles are written almost all the time for political purposes to promote the agenda of the party to which the author belongs or the party which he supports.

- Opinion/Commentary: These are some influential articles through which the author basically tries to influence the reader in the understanding of recent events.
- Satire/Humor: These stories contain some exaggerated things written only for the entertainment purpose but can make an individual think differently about something.

Fake news can also be classified into two different types on the basis of the reason of its distribution as follows:

- Misinformation: Here the spreader believes that the news is true but in actual fact it is fake.
- Disinformation: Here the spreader knows that the news is false and spreads it intentionally to deceive the audience.[3]

1.2 Fake News Detection

The technique of recognising and categorizing incorrect or misleading material in order to prevent its spread is known as fake news detection. The spread of false news has become a big concern with the expansion of internet platforms and social media. Detecting and combating false news is critical for guaranteeing the trustworthiness of information sources and a well-informed society.

Identifying real news articles, photographs, and videos from fake ones is the goal of fake news detection. It entails analyzing numerous elements of the content, context, and sources in order to determine its authenticity. Here are some typical tactics and methods for detecting fake news:

Natural Language Processing: To analyze the linguistic components of news items, natural language processing (NLP) methods are used. The content and context of the news may be understood using sentiment analysis, named entity identification, and topic modeling, which can also assist to spot any potential biases or discrepancies.

Source Verification: Verifying the integrity and reputation of the news source is essential to spotting false information. The chance that the news is factual may be determined by evaluating the source's credibility, authority, and fact-checking procedures.[4]

Fact-checking: Fact-checking entails cross-referencing the claims and assertions made in the news piece with reliable, trustworthy sources. Organizations that do fact-checking use both manual and automatic technologies to validate the data.

Social media analysis: On social media, false information spreads quickly. Potentially inaccurate information may be found by looking at user interaction, social network dynamics, and the reputation of the source within the social media community.[5]

Machine Learning Algorithms: Machine learning (ML) techniques are frequently utilised to identify false information. In order to categorize articles & content as authentic or false based on their attributes, supervised learning methods such as, support vector machine, decision trees & logistic regression can be trained on labelled datasets.[6]

Deep Learning Models: Recurrent neural networks (RNNs) and convolutional neural networks (CNNs) are examples of deep learning (DL) models that can analyze the textual and visual content of news articles, photos, and videos. These models can recognise false information by capturing intricate relationships and patterns.[7][8]

Ensemble Approaches: To increase the precision of false news detection, ensemble approaches integrate many models or procedures. The predictions of various models can be combined using strategies such as majority voting, weighted voting, or stacking.

It is significant to highlight that the area of detecting fake news is active and constantly changing as a result of the strategies used by those disseminating false information. In order to keep up with the rapidly evolving field of fake news, researchers and practitioners are continually investigating novel methods and strategies.

Fake news detection strives to give people and communities accurate and trustworthy information by utilizing the power of cutting-edge technology and multidisciplinary approaches, promoting a better informed society.[3][9]

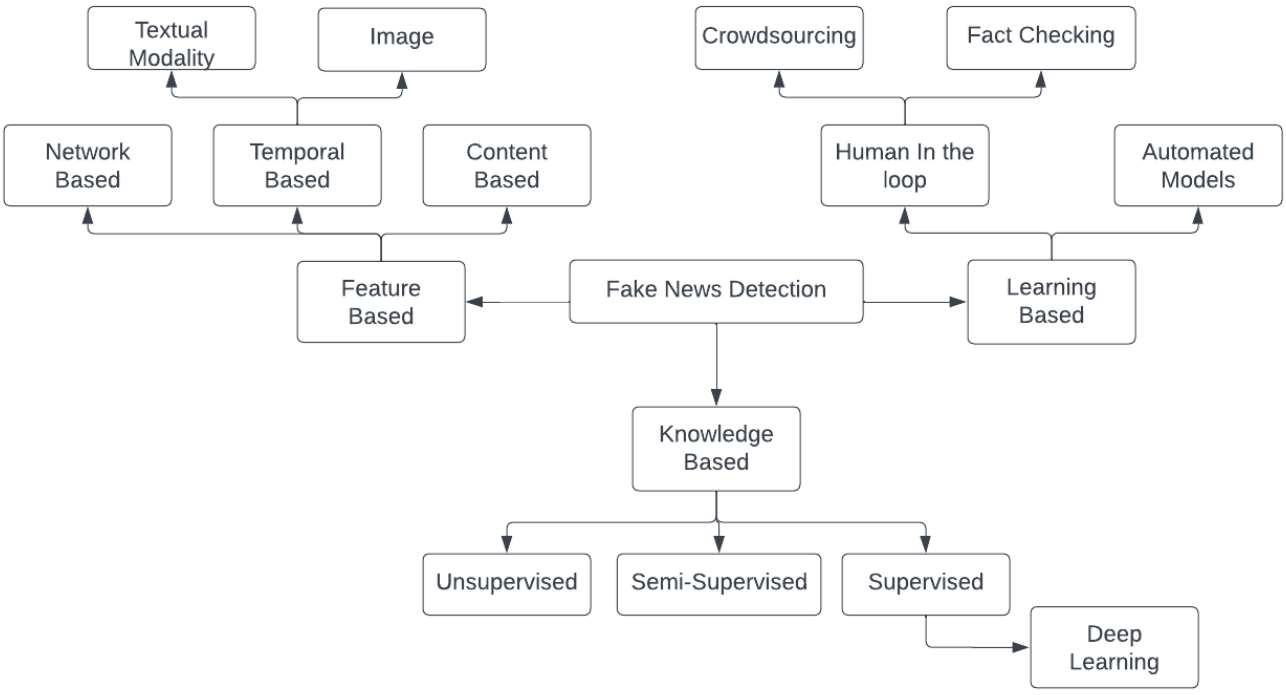

Figure 1.1: Various Approaches for Fake News Detection

Additional strategies, like textual modality, crowdsourcing, and fact-checking, have a great deal of promise for improving the precision and dependability of false information detection systems. Textual modality analysis makes use of cutting-edge natural language processing methods to identify misleading material by capturing linguistic clues and contextual discrepancies. By utilising the group brainpower of volunteers, crowdsourcing creates a scalable and quick verification procedure for textual and visual material. The crowdsourced evaluations and judgements are included into ensemble models to boost the detection process. The experience and information provided by fact-checking procedures, which involve in-depth research and cross-referencing with reliable sources, can also be incorporated into ensemble models. Researchers may improve false information detection systems' precision, robustness, and practical application by adding these strategies, which will result in a landscape of information that is more dependable and trustworthy.

Chapter 2

LITERATURE REVIEW

After reviewing different research papers, we observed different types of approaches and we can classify them into the following categories based on the characteristics that they have used for detection:

2.1 Naïve Bayes Classification

This approach tries to classify the fake news with the help of Naïve Bayes classifier. The dataset used in this approach is collected from github and contains nearly 11000 news articles in form of rows and four columns named index, title, text and label (either fake or true). The vectorization techniques used to create word embeddings are bag of words and n-grams. The author has done the classification by first taking the only the title column in consideration and then the text column is also taken into consideration. The AUC score for both the columns using both the classifiers is than compared and it is found that the number of words in the text provided to the classifier improves the AUC score. This paper also talks about the use of web scrapping to keep our datasets updated.[10]

2.2 Supervised Learning for Fake News Detection

In this approach the author focused on three different types of features of a news article for classification: feature extracted from news articles, feature extracted from news source, features extracted from news environment. The features extracted from news content consists of: Language features (obtained using POS tagging), Lexical Features (consists of number of unique words and their frequency in the text), Psycholinguistic features (obtained through Linguistic enquiry and word count (LIWC)), Subjectivity (Obtained using Textblob's API). The features extracted from news source contains: Bias, Credibility and trustworthiness (obtained using Facebook's and Alexa's API by collecting the rankings of different newspapers and websites), Domain Location (obtained using ipstack API). The environment features contains: Engagements (likes and comments), temporal patterns

(obtained by computing the rate at which the comments are posted). The dataset used in this paper consists of 2282 Buzzfeed news articles related to 2016 US elections. All the articles under the category of "mostly false" and "the mixture of true and false" are merged into a single class and that class is named as the fake news class, "non-factual content" stories are all removed from the dataset and rest all articles are labeled as real news. The classification is done using five classifiers which are: k-nearest neighbors (KNN), Random Forest (RF), Naïve Bayes (NB), Support Vector Machine With RBF kernel (SVM) and XGBoost (XGB). Also the performance is calculated using AUC and F1 score.[11]

2.3 TI-CNN

In this approach the author has tried to classify the news using convolution neural considering both text as well as the image aspects of the news article. This strategy collects the dataset from Kaggle which has 20,015 news articles scrapped from nearly 240 websites. It contains nearly 8,000 true news articles and 12,000 false news articles. For the textual data this paper considers some linguistic features like number of words and sentences in a news article (generally less in case of fake news), punctuation marks which tells us how confident the writer is while writing that article (usually found to be more in case of fake news), cognitive perspective which includes the use of negative words in the article (used less by the fake news creators to avoid contradictions), lexical diversity (more diverse the use of words, more likely it is a real news) and sentiments analysis (usually negative in case of fake news due to the mindset of the creator). The image analysis was also done where it was found that there are more number of faces in case of real news in an image as compared to the fake news. Also the fake articles contain more irrelevant images like sceneries and animals which have nothing to do with the article. The approach for classification includes two parallel CNNs, one for the textual data and the other for the image analysis. The text branch is utilizing two features: textual latent and textual explicit features. The textual explicit features are the linguistic features which are explained above and the textual latent features are creating by CNN by creating the word embeddings and each word embedding can be concatenated together to form a feature vector for the news article. The image branch is also utilizing two features which include visual latent and visual explicit features. The visual explicit feature is used to extract the resolution and the number of faces in the image and the visual latent features are used to learn from raw images and derive some more powerful features.[12]

2.4 Topic Agnostic Approach

In this approach the classification is done by considering both linguistic as well as the web mark features of the news article. The approach is based on a baseline paper called

FNDetector which also considers these key aspects of the news article. The linguistic features includes: Morphological features (obtained using part-of-speech tagging assigning each word to a category based on its context), psychological features (obtained using Linguistic Enquiry and word count (LIWC)), Readability features (obtained using Textstat which is an inbuilt python library that gives us an ease score for readability of an article). The web markup features are extracted using python libraries called Beautiful Soup and The Newspaper. The datasets used in this approach are Celebrity, US-Elections2016 and PoliticalNews. The classification is done using three classifiers: Random Forest (RF), K-Nearest Neighbors (KNN) and Support Vector Machine (SVM). Every combination of features is used separately to observe the difference in accuracy of the model. Also headline and content of the news article are given separately as well as together to the model to observe the accuracy difference. This paper significantly increase the accuracy of its base paper which is FNDetector.[13]

2.5 Machine Learning with Knowledge Engineering Approach

This is a proposed integrated approach consisting three main steps: Classification, User Stance Detection and fact checking using knowledge engineering. This strategy collects the dataset from Kaggle which contains 17,946 news articles out of which 12,460 are biased, 572 are fake articles, 870 are conspiracy and 2,059 are non-fake articles. The Support Vector Machine (SVM), a highly well-liked classifier in machine learning, is used for the classification. The author also discussed about other techniques which can be used for classification (like neural networks, Bayesian Classifiers etc.) along with their limitations. Stance detection can be done by simply checking the views of the users on the article. They can be categorized in two categories: Explicit (where user gives a direct impression), implicit (can be extracted from social media). The final step which is Fact checking can be done using three methods: Expert Based (human expertise is required to check facts in the article), Crowd-Sourcing Based (reader can read the article and on the basis of the reading experience and after understanding he/she can flag the article to be real or fake), Computational Oriented (knowledge engineering is used here where several rules are given to a machine so that it can imitate the thought process of a human expert).[14]

2.6 SpotFake

This strategy is centred on the news content's textual and graphic elements. Two datasets are used in this approach: Twitter and Weibo. The Twitter dataset contains 17000

different tweets related to various events. Each tweet contains the text data of the tweet and the images associated with it. This dataset contains nearly 10000 fake tweets and 7000 real tweets. ON the other hand Weibo is a dataset collected from authoritative news sources of China. The fake news in this dataset is collected from 2012-2016. The textual features in this approach are extracted through Bidirectional Encoder Representations from Transformers (BERT). To represent the contextual features in form of vectors BERT contains 12 encoding layers. Through the pre-trained VGG-19 convolutional network on the ImageNet dataset, the visual features are retrieved. The feature vector is finally shrunk down to 32 dimensions. The two feature vectors obtained through both the feature extractors are than fused together using concatenation technique to obtain an integrated vector representation of both image and text from the article. In the pre-processing step of the data, the text data length is fixed by trimming anything above the fixed length and by padding zeros to anything which is below the fixed length. For image components of the data every image is resized to 224x224x3. Hyperparameter tuning is also done to improve the accuracy of the model.[15]

2.7 Fake News Detection: A Deep Learning Approach

This strategy is based on a textual examination of the news article's data. This method simply shows how closely an article's body and title link to one another and focuses on identifying the viewpoint of a news story. The FNC-1 dataset was utilised in this method. It includes the news article's substance, title, and designation of the relationship (stance) between the two. The dataset includes 49,973 distinct pairings of news stories and headlines that fall under one of the four viewpoint categories: disagree, agree, discuss, or unrelated.

In this technique various pre-processing steps are used to make the data ready for modeling few of which are stop words removal (most common words used in a language), punctuation removal (punctuation marks like: , ? ! . etc) and stemming (removing prefixes and suffixes from a word). The vectorization techniques which are used in this approach are: Tf-idf, bag-of-words, word2vec and GloVe. These vectorization techniques are used with different types of neural networks which are: Deep Neural Networks (DNN), Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN). The input given to the neural networks contains the vector for the headline, vector for the body and the cosine similarity between both the vectors (all three things in a concatenated form). Also the activation functions contains ReLU, Tanh and Softmax. When employing Tf-idf on bigrams and unigrams with cosine similarity fed to a dense neural network, the best results are obtained.[16][17]

2.8 Credibility Based Fake News Detection

This strategy is focused only on a news article's textual elements. Datasets for this strategy were gathered from Politifact and Buzzfeed News. These databases not only contain information on the news' labels and content, but also about the users' social networks. The dataset from politifact contains 240 news articles out of which 120 are fake and 120 are real. The dataset collected from Buzzfeed News contains 182 news articles out of which 91 are fake and 91 are real. The features extracted from the data are mainly divided into two categories: Source Credibility and Content Credibility. The source credibility features contains: Authors (usually articles containing more than one authors are likely to be real), Co-authorship (whether the author is associated with Fake News articles, Real News articles or both). The content credibility features contains Sentiments, Readability, Argumentation (build by providing data and references), Number of characters, words and sentences, Typos etc. These extracted features are given to seven machine learning classifiers which are: SVM (RBF kernel),Logistic Rgression, Linear SVM, Adaboost, Random Forest, Gradient Boosting Decision Tree and Naive Bayes. The performance metrics used to calculate the performance are F1-macro, F1-weighted and F1-micro. These scores are calculated separately for the source credibility and Content credibility features and also separately for both the datasets.[18]

2.9 Multimodal Multi-Image Fake News Detection

This approach is developed using SpotFake as a baseline paper. It considers both textual and image data for classification purpose. The dataset is collected from FakeNews Net collection. In particular GossipCop (news about celebrities and entertainment) posts are used. In total 5459 news articles were collected which contains at least one image, out of which 2745 are false articles and 2714 are true articles. The three components of this technique are linguistic, visual and text and picture similarity. BERT (Bidirectional Encoder Representation from Transformers) is utilised in the textual component to collect contextual information. An encoder and a decoder are two distinct systems that are present. The decoder produces the task prediction after reading the input from the encoder. The pre-trained BERT is provided the padding text in order for it to receive the 768-word vector. For the visual component, the pre-trained VGG-16 on the visual dataset ImageNet is utilised for the picture content. Additionally, LSTM is used to determine the temporal picture order using VGG-16 activations. Finally, mean pooling is applied to the LSTM output to create a single temporal component. The top ten picture tags from the pre-trained VGG-16 model are extracted to calculate the third component, or text and image similarity. Than the word embeddings are created using word2vec and a 300 dimension vector is created by averaging the embeddings. The output layer uses the

Softmax function to calculate a probability representation for each feature, after which the concatenated features are multiplied by a Soft Mask of values between 0 and 1. The best results are obtained when a 3-image VGG-16 is used with LSTM and BERT is used to extract contextual features, also the similarity is calculated as explained above and all these features are fused using attention mechanism.[19]

Table 2.1: Comparison Table for Various Approaches.

Year	Technique	Performance Metric	Performance Score
2018[10]	NBC	AUC Score	Title (0.806 with countvectorizer) Title (0.807 with N-grams) Text (0.912 with countvectorizer) Text(0.931 with N-grams)
2017[11]	KNN, NB, RF, SVM, XGB	AUC, F1 Score	KNN (0.80, 0.75), NB (0.72, 0.75), RF (0.85, 0.81), SVM (0.79, 0.76), XGB (0.86, 0.81)
2018[12]	CNN for both textual and visual data	Precision, Recall, F1-score	0.9220, 0.9277, 0.9210
2017[20]	Linear SVM	Accuracy	Collected Dataset 0.74
2019[13]	SVM, RF, KNN	Accuracy	Celebrity News 0.73 Celebrity-0.78 US-Elections2016-0.86 PoliticalNews-0.83
2019[14]	SVM	Proposed Approach	-
2019[15]	BERT (for textual features) VGG-19 (for visual features)	Accuracy	77.77 (Twitter), 89.23 (Weibo)
2018[16]	DNN, CNN , RNN	Accuracy	94.31
2020[18]	SVM (RBF Kernel), Linear SVM, LR, RF, Adaboost, NB, GBC, DT	F1-score	0.80
2020[19]	VGG-16 (for visual content), BERT (for textual content)	F1-score	0.7955

Chapter 3

METHODOLOGY

In the Methodology part we will be discussing about the dataset used, process flow and other theoretical aspects like vectorizers and classifiers used in the coding part. We can start by discussing about the dataset used.

3.1 Dataset Used

The dataset we used during the coding part is named as WELFake Dataset. It contains 72134 entries of news articles, out of which 35028 are real and 37106 are fake. This dataset is made by merging four most popular news datasets which are: kaggle, McIntire, Reuters & Buzzfeed Political. The main purpose of merging these datasets is to provide more data for training and also to prevent the classifiers from over-fitting.[21]

This dataset contains four columns named Serial number, title, text and label. The serial number column is starting from index 0. The title column contains the heading of the news whereas the text column contains the news content. There are two types of labels present in the label column i.e, 0 and 1. Label 0 is for the fake news and 1 is for the real news. There are five classifiers that we have used to classify during coding part.

3.2 Classifiers Used

3.2.1 Logistic Regression

- Logistic regression is one of the most well-known Machine Learning algorithms used in the Supervised Learning method. A categorical dependent variable can be predicted using this technique using a collection of independent variables.

- The output of a dependent categorical variable is predicted via logistic regression. The outcome therefore has to be a discrete or categorical value. It can be True or

False, Yes or No, 0 or 1, and so on, but rather than providing precise values like 0 and 1, it provides probabilistic values that are in the range of 0 and 1.

- In logistic regression, we construct a "S"-shaped logistic function that predicts two upper and lower bounds (0 or 1) rather than a regression line.
- The sigmoid function is employed in logistic regression to forecast probability. It is beneficial to convert a real value to a number between 0 and 1.
- In python, we first need to import the logistic classifier from sklearn library and then we need to feed it with the vectorized data and the output label.

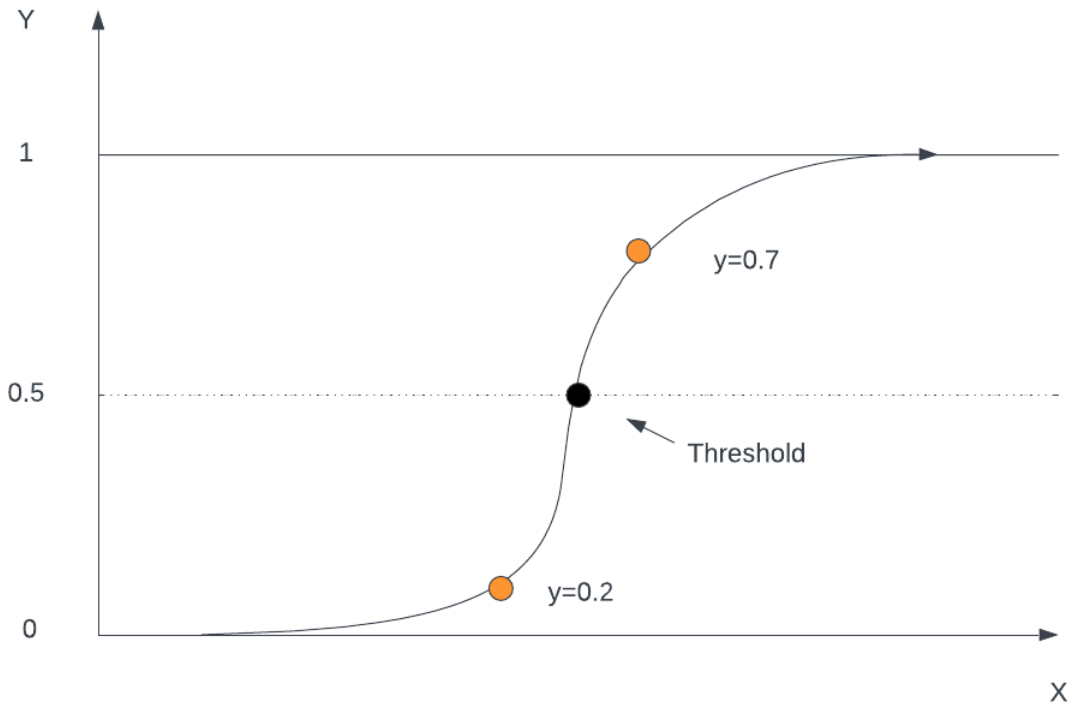

Figure 3.1: Logistic Regression Curve

3.2.2 Decision Tree

- Decision trees are the most well-known and effective categorization and prediction techniques. Each internal node in a decision tree represents a test of an attribute, each branch indicates the test's result, and each leaf node represents the class label. The layout of a decision tree is similar to a flowchart.

- The decision is performed on the basis of attributes of the given dataset.
- CART, ID3, and C4.5 are the three primary methods used to construct a decision tree. The most popular of these three algorithms is CART.
- At each internal node a simple question is asked like yes/no and based on the answer we further split the tree into sub-trees.
- In python we first need to import the decision tree classifier from sklearn.tree library and then we need to feed it with the vectorized data and the output label.

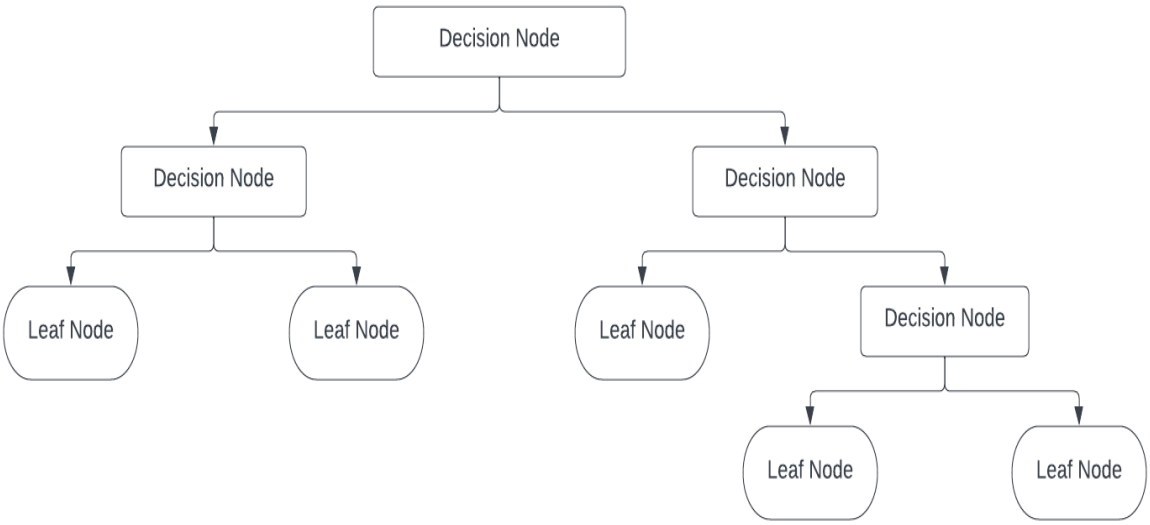

Figure 3.2: Decision Tree Approach

3.2.3 Random Forest

- Random forest is a supervised learning technique which can be used for both classification as well as the regression problem.
- Random forest classifier is a type of ensemble learning technique in which we combine multiple classifiers to improve the performance of model and to solve complex problems.
- In random forest classifier we take various subsets of a dataset and make different decision trees on these subsets. Based on majority votes from different trees random forest predicts the final output.
- It also improves the predictive accuracy of the decision tree classifier.

- In python we first need to import random tree classifier from sklearn.ensemble library and then we need to feed it with the vectorized data and the output label.

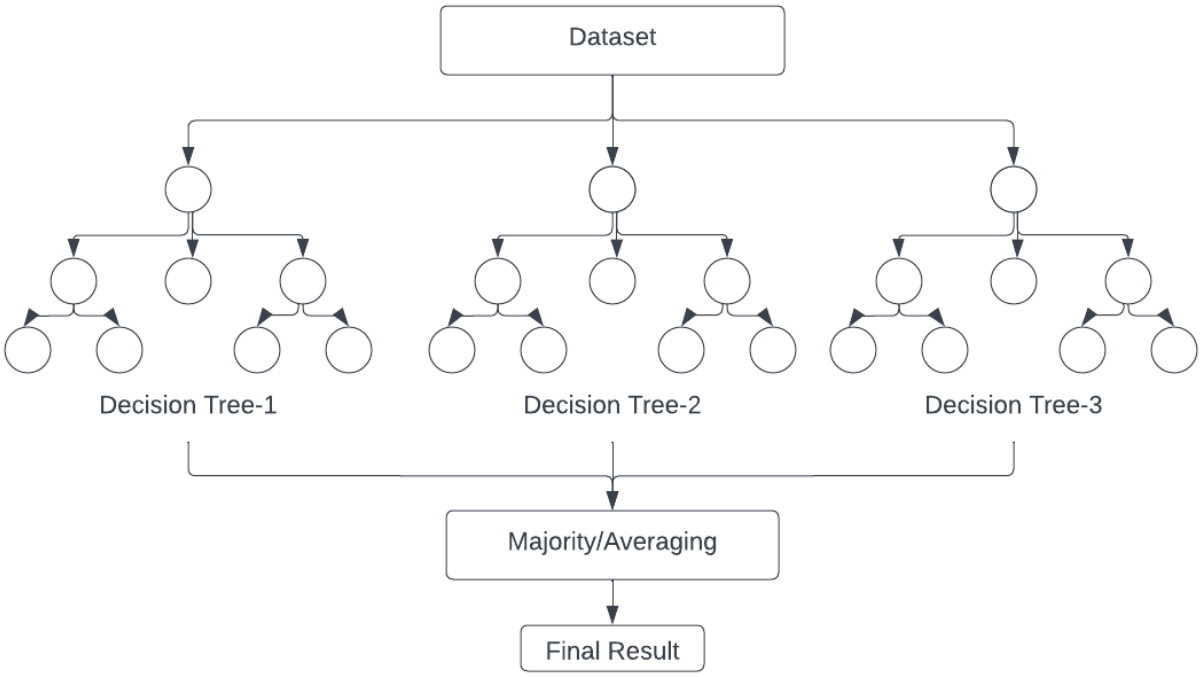

Figure 3.3: Random Forest Approach

3.2.4 Gradient Boosting Classifier

- Gradient Boosting classifier is the expansion of boosting procedure.
- Gradient Boosting can be seen as a sum of Gradient Descent Algorithm and Boosting.
- Differential loss function can be optimized using the gradient descent function.
- A group of trees are then constructed individually, each tree tries to restore the value of loss by the previous one.
- In python we first need to import Gradient Boosting Classifier from sklearn.ensemble library and then we need to feed it with the vectorized data and the output label.

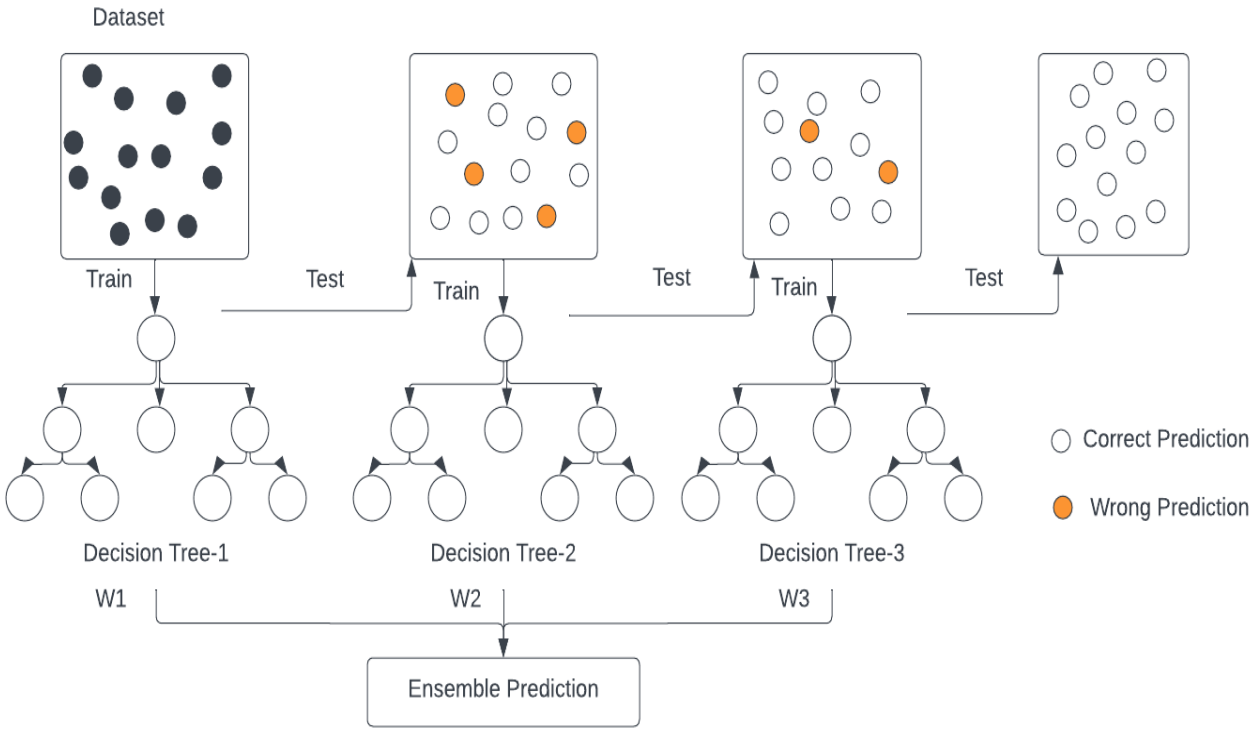

Figure 3.4: Gradient Boosting Approach

3.2.5 XGBoost Classifier

- XGBoost classifier stands for Extreme Gradient Boosting classifier.
- XGBoost classifier is a decision tree based ensemble machine learning algorithm, it uses a gradient boosting framework.
- XGBoost uses parallel processing, tree pruning and handles the missing values too.
- In python we first need to import XGBoost Classifier from XGBoost library and then we need to feed it with the vectorized data and the output label.

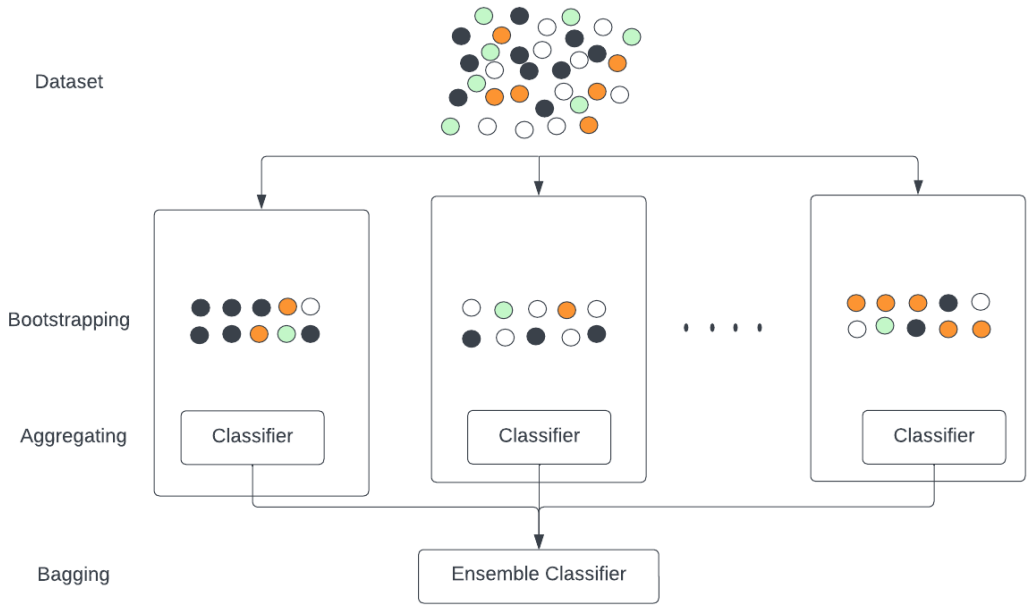

Figure 3.5: XGBoost Approach

3.3 Process Flow

First we are selecting the algorithms with best accuracy using simple approach containing following steps:

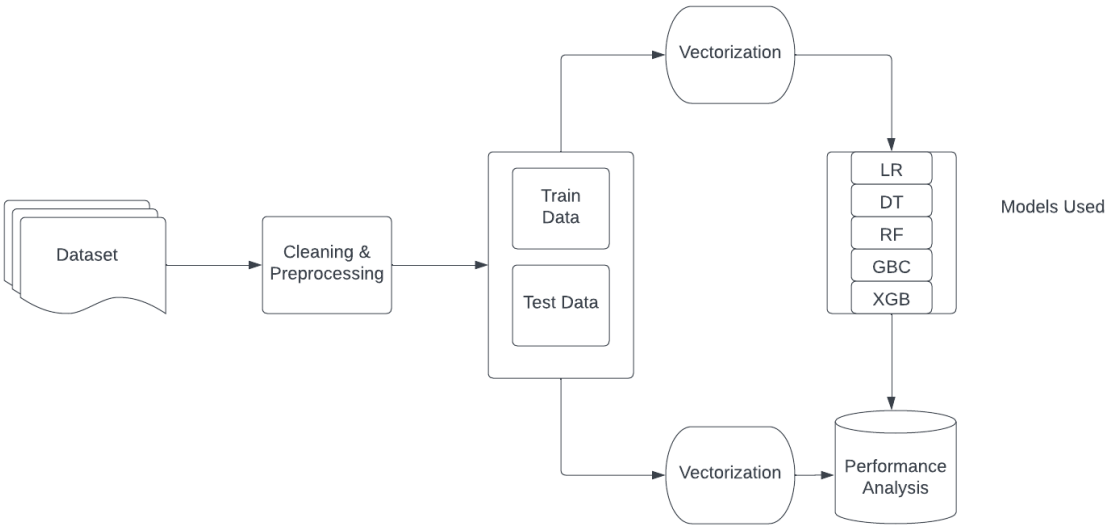

Figure 3.6: Approach for Selecting Best Performing Algorithm

3.3.1 Data Cleaning & Preprocessing

- First we are removing extra columns from the dataset named 'Unnamed'. In some cases we are combining the title of the news article with the text but in other cases we are simply removing it.
- There is also a need to shuffle the data before splitting it into training and testing set to remove any type of imbalance caused by the amount of data present for both the labels.
- We tried plotting the amount of data present for both the labels and these are the bar graphs we got:

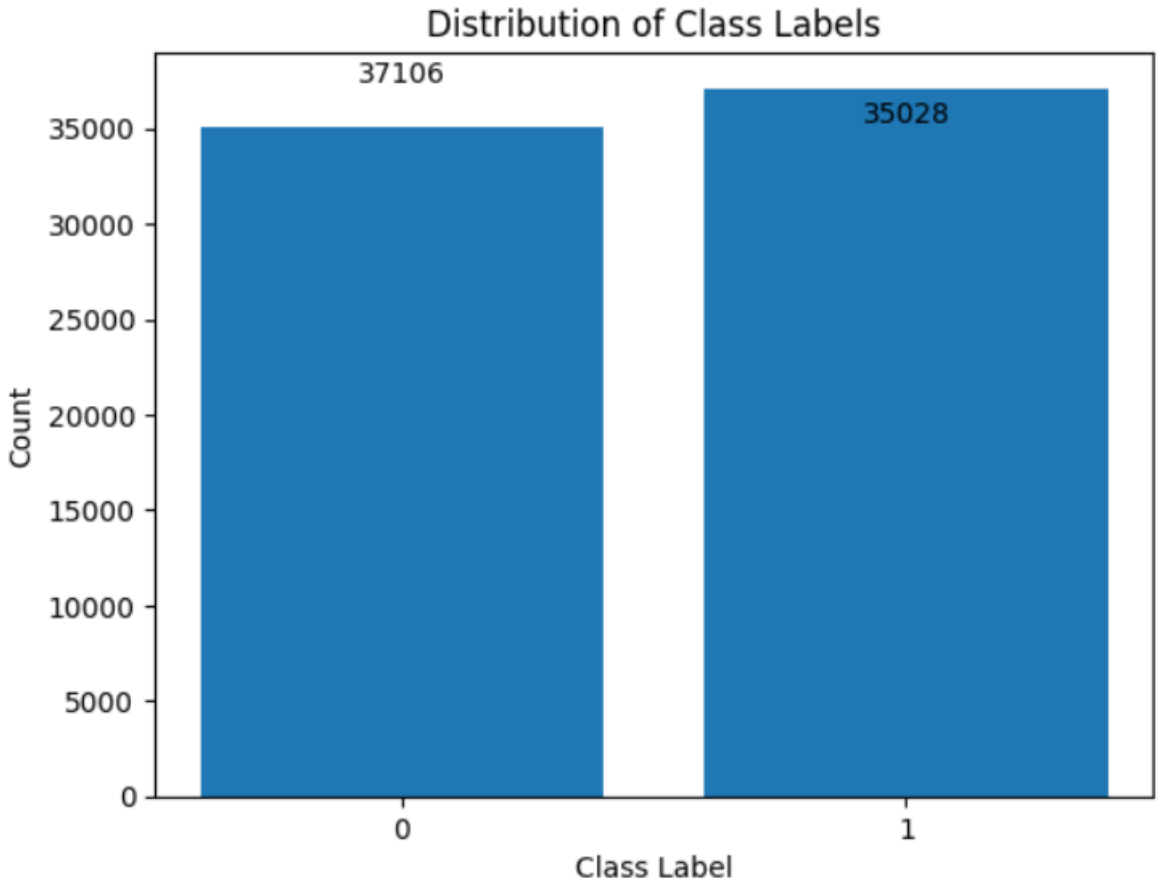

Figure 3.7: Distribution of Class Labels

- After that we are applying regular expression functions to remove things like: links, punctuation marks, brackets etc. which can deprive the performance of different classifiers.
- After applying regular expression functions we have to make the text data ready to feed it to the corresponding machine learning algorithms and for that we have used three to four vectorization techniques.

3.3.2 Vectorization Techniques Used

We are using three different vectorization techniques: Countvectorizer, TF-IDF vectorizer and n-gram (mainly unigrams & bigrams) for selecting the best algorithms out of the five that we have discussed earlier. The explanation of these vectorizers is as follows:

- **Countvectorizer:** By simply counting the number of times each word appears in the full text, this utility in the Python scikit-learn module can turn text into vectors. This method is useful when we wish to convert each document in a corpus into a distinct vector.

  It simply builds a vector for each word by counting that word in each document and just writing its frequency. For example let us suppose we have three documents and a word "love" and it occur one time in the first document, two times in second and five times in third document, than the vector for this word can simply be written as [1 2 5]. To use it on text first we need to import it from the scikit-learn library and simply applying it to the training and testing data.

- **TF – IDF Vectorizer:** It stand for Term Frequency – Inverse Document Frequency. It simply tells us the relevance of a word in a particular text or the corpus.
  Term frequency of a word is calculated by simply dividing the frequency of that word in the document by total number of words in that document.
  Inverse Document Frequency can be calculated by dividing the total number of documents with the number of documents which contains that particular word.
  In the end the tf-idf term is calculated by multiplying the term frequency with the log of inverse document frequency. To form the vector we simply write the tf-idf value of the word for each document in the vector as we did in the countvectorizer. To use it on text first we need to import it from the scikit-learn library and simply applying it to the training and testing data.

- **N-grams Vectorizer:** This vectorization technique can be used with both countvectorizer and tf-idf vectorizer. Here we have used it with tf-idf vectorizer. It simply takes a parameter n_gram in which we can give a range. For example if we are giving a range (1,2), then it will consider unigrams and bigrams. Suppose we have a sentence "No one is here". Then it will take unigrams as: "No", "one", "is", "here" and bigrams as "No one", "one is", "is here". It will consider both unigrams and bigrams as a single word and will apply the tf-idf approach to them as we discussed in the tf-idf vectorizer. To use it on text first we need to import it from the scikit-learn library and simply applying it to the training and testing data. After vectorization we will simply give the vectorized data to different classifiers, along with the output labels.

3.3.3 Results for Basic Approach

We have achieved the following accuracies for different vectorization techniques applied with different classifiers:

Table 3.1: Results for Basic Approach.

Calssifiers/Vectorizers	TF-IDF	Count	N-grams
Logistic Regression	94.12%	94.22%	94.16%
Decision Tree	91.61%	92.06%	92.38%
Random Forest	93.29%	93.04%	93.36%
Gradient Boosting	92.27%	91.89%	92.40%
XG Boost	92.22%	91.90%	92.34%

As we see that the algorithms performing best with all three vectorizers are Logistic Regression, Decision Tree and Random Forest. So we'll be using these three algorithms as an input to the ensemble models. It is because fusing the models with highest accuracy using ensemble models increases the accuracy of the resulting model.

3.3.4 Ensemble Learning

A more accurate and reliable prediction model is produced by combining several separate models, also referred to as base models or weak learners, using the effective technique of ensemble learning. Every base model provides its predictions, which are then combined by the ensemble model to get a conclusion. By utilising the diversity and collective intelligence of the ensemble members, ensemble learning can frequently do better than a single model.[22]

There are various ensemble learning approaches, each with unique features and benefits. Let's examine these strategies in greater detail:

Bagging:
Bootstrap aggregating, sometimes known as "bagging," is a common ensemble technique in which various base models are trained independently using various subsets of the training data. Through a procedure known as bootstrapping, random samples are taken with replacement from the initial training set to construct the subgroups. To add variety to the training process, each base model is trained using a different bootstrap sample. Usually, the forecasts of different base models are combined into one final prediction through voting or averaging. The capacity of bagging algorithms to lower variance and increase generalisation by lowering overfitting is well recognised. Examples include Random Forest and Extra Trees.

Boosting:
In the iterative ensemble technique known as "boosting," base models are trained successively, with each succeeding model aiming to fix the errors caused by the models that came before it. The samples that the prior models incorrectly identified are given more attention during training, enabling the ensemble to learn from its errors and progressively enhance its performance. By integrating the predictions of various weak models, boosting algorithms like AdaBoost, Gradient Boosting Machines (GBM), and XGBoost attempt to produce a strong final model. Boosting works especially well for managing complicated relationships and identifying subtle trends in the data.

Stacking:
When training numerous base models on the same dataset and integrating their predictions using a meta-model, this process is referred to as stacking or stacked generalisation. The meta-model's input features are the predictions of the basic models. The meta-model learns to produce the final prediction by taking into account the predictions of the underlying models rather than making predictions directly. Stacking enables the fusion of various models that each capture a particular aspect of the data and can reveal intricate connections between the base models. The meta-model of choice can range from a straightforward linear regression to more complex models like neural networks.[23]

Voting:
Voting is a simple ensemble technique that integrates the predictions of various base models to produce the final forecast. Voting is also known as majority voting or ensemble voting. Each base model is given an equal number of votes, and the final prediction is made using the class label that obtains the majority of the votes. Hard voting and soft voting are two different methods of casting a ballot. While soft voting takes into consideration the anticipated probability or confidence scores that the base models awarded to each class, hard voting solely takes into account each base model's final judgement. As soft voting takes into account the certainty or uncertainty of the predictions made by the base models, it frequently produces forecasts that are more accurate.

Weighted Ensemble:
In a weighted ensemble, the predictions from each base model are given a distinct weight. The relative importance or effectiveness of each model is represented by these weights. The weighted predictions of the base models are added together to create the final forecast. When specific models are predicted to perform better or have a greater impact on the outcome, weighted ensembles can be helpful. The ensemble can capitalise on the advantages of various models and raise overall prediction accuracy by applying the proper weights.[24]

We are using majority voting and weighted ensemble to fuse the results of the best performing machine learning algorithms, which are: Logistic Regression, Decision Tree and Random Forest. Also we are trying both the approaches using four feature extraction techniques, from which three are same as in the approach in which we tried to select the best performing algorithms i.e Count Vectorizer, TF-IDF Vectorizer and N-Grams Vectorizer. One more feature extraction technique is Word2Vec.

3.3.5 Word2Vec

In natural language processing (NLP), Word2Vec is a well-liked word embedding method that aims to capture the semantic links between words. It is predicated on the notion that words with related meanings frequently occur in related settings.

Learning the distributed representations of words in a continuous vector space is the basic idea behind Word2Vec. The semantic and grammatical characteristics of words are encoded in a dense numerical form by these representations, sometimes referred to as word embeddings. The "distributional hypothesis," which states that words appearing in comparable settings are likely to have similar meanings, is the central tenet of Word2Vec.

Word2Vec uses the Continuous Bag of Words (CBOW) and Skip-gram designs as its primary building blocks. Using the context words around the target word, the CBOW architecture model predicts the term. It attempts to anticipate the target word in the middle of a window of context words as input. On the other hand, the Skip-gram architecture uses the target word as input and tries to foretell the context words within a specified frame. To learn the word embeddings, both architectures employ a neural network with a hidden layer.

Iteratively changing the neural network's weights throughout Word2Vec's training process will increase the neural network's capacity to predict the target word given the context or vice versa. The goal is to increase the likelihood that a target word or a target word for a particular context will be predicted accurately. This is accomplished through a method known as "negative sampling," in which the model is trained to differentiate between the appropriate context terms and randomly selected negative phrases.

The Word2Vec model gains the ability to assign comparable vector representations to words that frequently occur in comparable contexts during training. As a result, in the embedding space, words with comparable meanings or semantic links frequently have comparable vector representations. As an illustration, the vectors of "king" and "queen" would be closer together than the vectors of "king" and "cat."

The learnt word embeddings from the Word2Vec model can be applied to a variety of NLP applications after it has been trained. These embeddings are capable of capturing semantic relationships and analogies between words, enabling operations like the deter-

mination of word similarity and the completion of word analogies, as well as the recording of contextual data for later operations like sentiment analysis or text classification.

While Word2Vec is an effective tool for learning word embeddings, it has some drawbacks that must be taken into consideration. It operates under the presumption that a word's meaning is solely influenced by its immediate context, ignoring more extensive linguistic and semantic structures. Additionally, because their representations are not explicitly provided in the training data, Word2Vec may have trouble with uncommon words or terminology that are not commonly used.

As a result of capturing the contextual links between words, the commonly used word embedding method Word2Vec develops distributed representations of words. Word2Vec uses semantic and syntactic similarities between words to represent words in a continuous vector space, enabling a variety of NLP applications and tasks.

3.4 Final Methodology

As we discussed earlier we are combining three best models out of five that we have used in the basic approach to improve its accuracy. The flow diagram for the methodology is:

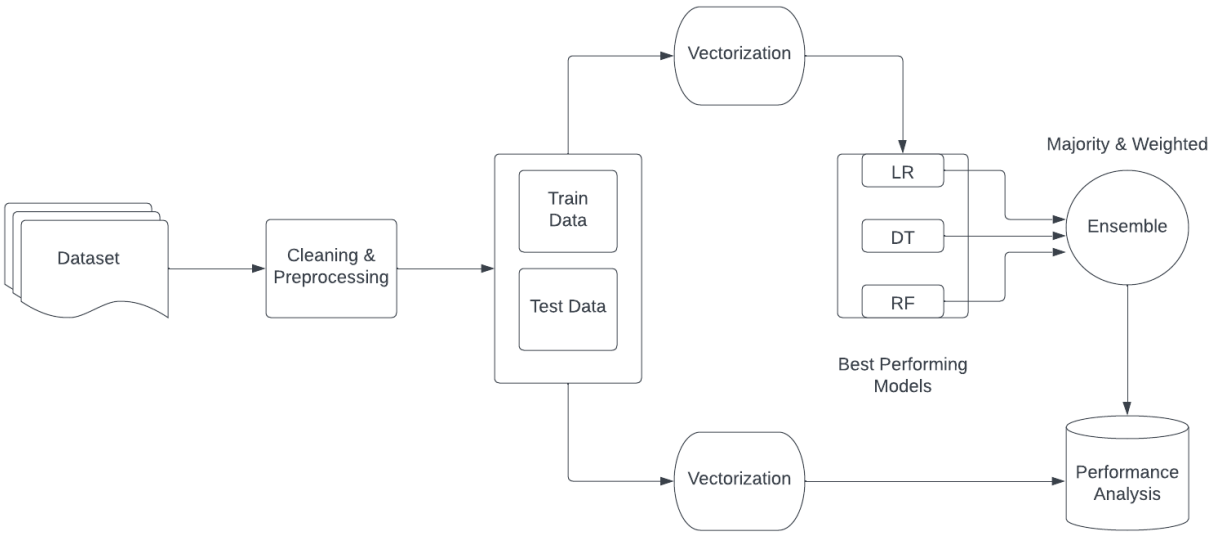

Figure 3.8: Final Methodology using Ensemble Methods

The cleaning and preprocessing as well as the vectorization parts are same here except the fact that we are using one more vectorization technique here which is Word2Vec. Also in case of majority voting ensemble we are using both soft and hard voting. In case of weighted ensemble it makes no sense to use hard voting because it kind of gives preference to the model having more weight.

To obtain the best weights in case of weighted ensemble we are using GridSearchCV. In this Method we can define a grid of weights for the ensemble technique to work on,

and in turn it gives us the weights which are showing highest accuracy.

We are using a grid containing weights which sums to 1. The weights are given in such away to avoid preferring any model disproportionately. Additionally, rounding the weights up to 1 can make it easier to analyse and comprehend the ensemble. It makes it simpler to evaluate each model's contributions to the final prediction and gives a clear indication of the relative relevance of each one.

Although it's standard practice to add the weights up to 1, it's not technically necessary. Sometimes, you may need to give distinct weights that don't add up to 1 for certain reasons or requirements. However, for a balanced and understandable ensemble in the context of grid search, it is typically advised to choose weights that add up to 1.

Chapter 4

RESULTS and DISCUSSION

We have evaluated the impact of Ensemble models on the accuracy of state-of-the-art machine learning algorithms. The accuracy of the formerly employed algorithms significantly improved as a result of the ensemble learning. Both soft and hard voting methods are used to calculate the effect of majority voting ensemble. Also to use the weighted ensemble we used the GridSerachCV method with a parameter grid containing weights for the state-of-the-art machine learning algorithms. In the parameter grid that we used each set of parameters adds up to 1. This way we have avoided preferring one algorithm over the other. The parameter grid that we have used is:

[(0.1, 0.5, 0.4), (0.2, 0.4, 0.4), (0.3, 0.3, 0.4), (0.4, 0.3, 0.3), (0.4, 0.4, 0.2), (0.5, 0.4, 0.1)]

For almost all the vectorizers, we are experiencing the same weights to perform well and give us the best accuracy. The weights are (0.4,0.3,0.3). In the estimator parameter of the GridSearchCV we placed LR as first member, then DT and then RF. Also LR was showing the best accuracy among all five algorithms. This is the reason that the weights (0.4, 0.3, 0.3) are performing well. Because these weights are giving a very little preference to LR (which is performing well iin terms of accuracy).

Table 4.1: Accuracy Obtained For Majority Voting Ensemble(in %).

Vectorizers/Voting	**Soft Voting**	**Hard Voting**
TF-IDF	95.83	95.38
Count	95.32	95.60
Count N-grams	97.42	97.55
TF-IDF N-grams	95.40	95.61
Word2Vec	93.77	93.26

The results for weighted ensemble are obtained on a specific parameter grid and the final accuracy is obtained for the weights (0.4, 0.3, 0.3).

Table 4.2: Accuracy Obtained For Weighted Ensemble(in %).

Vectorizers/Weighted Ensemble	**Accuracy**
TF-IDF	94.60
Count	95.46
Count N-grams	96.87
TF-IDF N-grams	95.24
Word2Vec	94.62

Our approach shows slightly higher accuracy than the baseline model WELFake[21]. The WELFake paper shows the accuracy using various state-of-the-art machine learning techniques with various features. It also uses Voting Classifier fusing the results of a single classifier using various features separately, that too for various machine learning algorithms.

Table 4.3: Comparison With Baseline Model

Model	**Accuracy**
WELFake	96.73
Count N-Grams With Soft Voting	97.42
Count N-Grams With Hard Voting	97.55
Count N-Grams With Weighted Ensemble	96.87

Chapter 5

CONCLUSION AND FUTURE SCOPE

In order to increase the precision of false information detection in text data, we conducted a comparative analysis of the majority and weighted voting ensemble methodologies in this work. Our results show that these ensemble strategies perform better than fundamental machine learning algorithms in terms of accuracy, illustrating their potency in dealing with the problems presented by false data.

We discovered thorough testing and assessment that ensemble models frequently outperformed individual models in terms of accuracy rates. The weighted voting ensemble strategy allowed us to provide varying weights to classifiers based on their performance, significantly improving the detection accuracy. The majority voting ensemble approach produced robust findings by combining the predictions of various classifiers.

Furthermore, text data—a frequent medium for the spread of incorrect information—was the only focus of our study. The findings show that ensemble approaches, which make use of the collective intelligence of several classifiers to more accurately detect and differentiate between true and incorrect information, are particularly successful in this field.

While this study has shed light on the effectiveness of the majority and weighted voting ensemble procedures, there are still a number of opportunities for improvement. Future study might focus on a number of issues, including:

- Feature Engineering: Look at how various text representation methods—like word embeddings or deep learning-based methods—affect the effectiveness of ensemble models. The identification of fraudulent information may be improved by investigating other factors like language patterns or environmental clues.

- Ensemble Combination Strategies: To capitalize on the advantages of various ensemble approaches, investigate cutting-edge ensemble combination tactics like stacking or hybrid ensembles. These methods could perform better and be more resilient, especially in cases where misleading information is complicated and dynamic.

- Integration of Visual Data: Look into the use of visual data in false information detection algorithms, such as photographs and videos. False information is largely

conveyed through visual imagery, especially on social media platforms. The accuracy and resilience of false information detection systems may be improved by creating methods for extracting useful characteristics from visual input and incorporating them into ensemble models.[25]

- User Perception and Behavioral Analysis: Consider both textual and visual clues when examining the function of user perception and behavioural analysis in the identification of misleading information. Examine how human cognitive biases and cognitive processes affect how visual material is interpreted, and devise strategies for incorporating user input and behaviour into ensemble models for more precise decision-making.[26]

As a result, our study has shown how majority and weighted voting ensemble strategies may increase the accuracy of text data false information detection. We can continue to progress the field of false information detection and contribute to the creation of more dependable and resilient systems for recognising and reducing the impact of false information by further investigating the indicated topics for future study.

REFERENCES

[1] X. Zhang and A. A. Ghorbani, "An overview of online fake news: Characterization, detection, and discussion," *Information Processing & Management*, vol. 57, no. 2, p. 102025, 2020.

[2] R. Oshikawa, J. Qian, and W. Y. Wang, "A survey on natural language processing for fake news detection," *arXiv preprint arXiv:1811.00770*, 2018.

[3] A. Campan, A. Cuzzocrea, and T. M. Truta, "Fighting fake news spread in online social networks: Actual trends and future research directions," in *2017 IEEE International Conference on Big Data (Big Data)*. IEEE, 2017, pp. 4453–4457.

[4] K. Shu, X. Zhou, S. Wang, R. Zafarani, and H. Liu, "The role of user profiles for fake news detection," in *Proceedings of the 2019 IEEE/ACM international conference on advances in social networks analysis and mining*, 2019, pp. 436–439.

[5] X. Zhou and R. Zafarani, "Network-based fake news detection: A pattern-driven approach," *ACM SIGKDD explorations newsletter*, vol. 21, no. 2, pp. 48–60, 2019.

[6] A. Jain, A. Shakya, H. Khatter, and A. K. Gupta, "A smart system for fake news detection using machine learning," in *2019 International conference on issues and challenges in intelligent computing techniques (ICICT)*, vol. 1. IEEE, 2019, pp. 1–4.

[7] R. K. Kaliyar, A. Goswami, P. Narang, and S. Sinha, "Fndnet–a deep convolutional neural network for fake news detection," *Cognitive Systems Research*, vol. 61, pp. 32–44, 2020.

[8] 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, p. 100007, 2021.

[9] D. K. Vishwakarma and C. Jain, "Recent state-of-the-art of fake news detection: A review," in *2020 International Conference for Emerging Technology (INCET)*. IEEE, 2020, pp. 1–6.

[10] A. Jain and A. Kasbe, "Fake news detection," in *2018 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS)*. IEEE, 2018, pp. 1–5.

[11] J. C. Reis, A. Correia, F. Murai, A. Veloso, and F. Benevenuto, "Supervised learning for fake news detection," *IEEE Intelligent Systems*, vol. 34, no. 2, pp. 76–81, 2019.

[12] Y. Yang, L. Zheng, J. Zhang, Q. Cui, Z. Li, and P. S. Yu, "Ti-cnn: Convolutional neural networks for fake news detection," *arXiv preprint arXiv:1806.00749*, 2018.

[13] S. Castelo, T. Almeida, A. Elghafari, A. Santos, K. Pham, E. Nakamura, and J. Freire, "A topic-agnostic approach for identifying fake news pages," in *Companion proceedings of the 2019 World Wide Web conference*, 2019, pp. 975–980.

[14] S. Ahmed, K. Hinkelmann, and F. Corradini, "Combining machine learning with knowledge engineering to detect fake news in social networks-a survey," *arXiv preprint arXiv:2201.08032*, 2022.

[15] S. Singhal, R. R. Shah, T. Chakraborty, P. Kumaraguru, and S. Satoh, "Spotfake: A multi-modal framework for fake news detection," in *2019 IEEE fifth international conference on multimedia big data (BigMM)*. IEEE, 2019, pp. 39–47.

[16] A. Thota, P. Tilak, S. Ahluwalia, and N. Lohia, "Fake news detection: a deep learning approach," *SMU Data Science Review*, vol. 1, no. 3, p. 10, 2018.

[17] F. Monti, F. Frasca, D. Eynard, D. Mannion, and M. M. Bronstein, "Fake news detection on social media using geometric deep learning," *arXiv preprint arXiv:1902.06673*, 2019.

[18] N. Sitaula, C. K. Mohan, J. Grygiel, X. Zhou, and R. Zafarani, "Credibility-based fake news detection," *Disinformation, misinformation, and fake news in social media: Emerging research challenges and Opportunities*, pp. 163–182, 2020.

[19] A. Giachanou, G. Zhang, and P. Rosso, "Multimodal multi-image fake news detection," in *2020 IEEE 7th international conference on data science and advanced analytics (DSAA)*. IEEE, 2020, pp. 647–654.

[20] V. Pérez-Rosas, B. Kleinberg, A. Lefevre, and R. Mihalcea, "Automatic detection of fake news," *arXiv preprint arXiv:1708.07104*, 2017.

[21] P. K. Verma, P. Agrawal, I. Amorim, and R. Prodan, "Welfake: Word embedding over linguistic features for fake news detection," *IEEE Transactions on Computational Social Systems*, vol. 8, no. 4, pp. 881–893, 2021.

[22] I. Ahmad, M. Yousaf, S. Yousaf, and M. O. Ahmad, "Fake news detection using machine learning ensemble methods," *Complexity*, vol. 2020, pp. 1–11, 2020.

[23] J. Thorne, M. Chen, G. Myrianthous, J. Pu, X. Wang, and A. Vlachos, "Fake news detection using stacked ensemble of classifiers." Association for Computational Linguistics, 2017.

[24] H. Reddy, N. Raj, M. Gala, and A. Basava, "Text-mining-based fake news detection using ensemble methods," *International Journal of Automation and Computing*, vol. 17, no. 2, pp. 210–221, 2020.

[25] P. Qi, J. Cao, T. Yang, J. Guo, and J. Li, "Exploiting multi-domain visual information for fake news detection," in *2019 IEEE international conference on data mining (ICDM)*. IEEE, 2019, pp. 518–527.

[26] S. Mohseni, F. Yang, S. Pentyala, M. Du, Y. Liu, N. Lupfer, X. Hu, S. Ji, and E. Ragan, "Machine learning explanations to prevent overtrust in fake news detection," in *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 15, 2021, pp. 421–431.

turnitin

Similarity Report ID: oid:27535:36365320

PAPER NAME

Thesis_Content2.pdf

WORD COUNT	CHARACTER COUNT
7618 Words	**40284 Characters**
PAGE COUNT	FILE SIZE
29 Pages	**278.9KB**
SUBMISSION DATE	REPORT DATE
May 28, 2023 11:32 PM GMT+5:30	**May 28, 2023 11:32 PM GMT+5:30**

5% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

- 3% Internet database
- 3% Publications database
- Crossref database
- Crossref Posted Content database
- 4% Submitted Works database

Excluded from Similarity Report

- Small Matches (Less then 10 words)