A

Project Report

on

Fake News Detection Based on Machine Learning Techniques

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY DEGREE

SESSION 2022-23

in

Computer Science and Engineering

By

Lakshya Prakash Yadav (1900290100083)

Vidit Upreti (1900290100190)

Shwetank Pratap Tiwari (1900290100157)

Under the supervision of

Dr. Sanjiv Sharma

KIET Group of Institutions, Ghaziabad

Affiliated to

Dr. A.P.J. Abdul Kalam Technical University, Lucknow

(Formerly UPTU)

May, 2023

DECLARATION

We hereby declare that this submission is our own work and that, to the best of our knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Signature:	Signature:
Name: Shwetank Pratap Tiwari	Name: Vidit Upreti
Roll No.:1900290100157	Roll No.: 1900290100190
Date: 28 May 2023	Date: 28 May 2023

Signature:

Name: Lakshya Prakash Yadav

Roll No.: 1900290100083

Date: 28 May 2023

CERTIFICATE

This is to certify that Project Report entitled "**Fake News Detection Based On Machine Learning Techniques**" which is submitted by Lakshya Prakash Yadav, Vidit Upreti and Shwetank Pratap Tiwari in partial fulfillment of the requirement for the award of degree B. Tech. in Department of Computer Science & Engineering of Dr. A.P.J. Abdul Kalam Technical University, Lucknow is a record of the candidates own work carried out by them under my supervision. The matter embodied in this report is original and has not been submitted for the award of any other degree.

Date: **Supervisor Name: Dr. Sanjiv Sharma**

(Assistant professor)

ACKNOWLEDGEMENT

It gives us a great sense of pleasure to present the report on the B. Tech Project undertaken during B. Tech. Final Year. We owe a special debt of gratitude to Dr. Sanjiv Sharma, Department of Computer Science & Engineering, KIET, Ghaziabad, for his constant support and guidance throughout the course of our work. His sincerity, thoroughness and perseverance have been a constant source of inspiration for us. It is only his cognizant efforts that our endeavors have seen the light of the day.

We also take the opportunity to acknowledge the contribution of Dr. Vineet Sharma, Head of the Department of Computer Science & Engineering, KIET, Ghaziabad, for his full support and assistance during the development of the project. We also do not want to miss the opportunity to acknowledge the contribution of all the faculty members of the department for their kind assistance and cooperation during the development of our project.

We also do not like to miss the opportunity to acknowledge the contribution of all faculty members, especially faculty/industry person/any person, of the department for their kind assistance and cooperation during the development of our project. Finally, we acknowledge our friends for their contribution in the completion of the project.

Date: 28 May 2023	Date: 28 May 2023
Signature:	Signature:
Name: Shwetank Pratap tiwari	Name: Vidit Upreti
Roll No.: 1900290100157	Roll No.: 1900290100190

Date: 28 May 2023

Signature:

Name: Lakshya Prakash Yadav

Roll No.: 1900290100083

ABSTRACT

Fake news has become a problem of great impact in our information driven society because of the continuous and intense fakesters content distribution. Information quality in news feeds is under questionable veracity calling for automated tools to detect fake news articles. Due to the many faces of fakesters, creating such a tool is a challenging problem. In this work, we propose a model for fake news detection using content-based features and Machine Learning (ML) algorithms.

The advent of the World Wide Web and the rapid adoption of social media platforms (such as Facebook and Twitter) paved the way for information dissemination that has never been witnessed in human history before. With the current usage of social media platforms, consumers are creating and sharing more information than ever before, some of which are misleading with no relevance to reality. Automated classification of a text article as misinformation or disinformation is a challenging task. Even an expert in a particular domain must explore multiple aspects before giving a verdict on the truthfulness of an article. In this work, we propose to use machine learning ensemble approach for automated classification of news articles. Our study explores different textual properties that can be used to distinguish fake contents from real. By using those properties, we train a combination of different machine learning algorithms using various ensemble methods and evaluate their performance on 4 real world datasets. Experimental evaluation confirms the superior performance of our proposed ensemble learner approach in comparison to individual learners.

Keywords: Fake news, Social media, Authenticity, Artificial Intelligence, Logistic regression, Support vector machine, Naïve Bayes algorithm, Random Forest algorithm

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Sno.	Abbreviations
1	ISOT – Information Security and Object Technology
2	KNN – K - Nearest Nieghbour
3	SVM – Support Vector Machine
4	IBM – International Business Machines
5	AI – Artificial Intelligence
6	RBF – Radial Basis Function
7	NN – Neural Networks
8	CART - Classification and Regression Trees
9	ID3 - Iterative Dichotomiser 3
10	XGBoost – Extreme Gradient Boosting
11	CPU – Central Processing Unit
12	NLP – Natural Language Processing
13	ADABoost – Adaptive Boosting
14	NLTK – Natural Language Toolkit
15	TF-IDF Term Frequency – Inverse Document Frequency
16	DS – Dataset
17	RF – Random Forest

CHAPTER 1
INTRODUCTION

1.1 Introduction

False information spreading among the masses is nothing new and has been around since ancient times, long before the advent of the Internet. Misleading and false information is disseminated for financial gain using fake news. A couple of years ago the news used to disseminate from one person to another by word of mouth, but in the current scenario, the dissemination of information has come a long way since then. Because of the development of the World Wide Web and the rapid proliferation of online social media platforms (such as Facebook, Twitter, and WhatsApp), it is now possible for the information to be transmitted in a manner that has never been witnessed before in the annals of human history. As a result of increased connectivity and intelligent automation, now more than a million users get their news and other information primarily from social networking sites and other social platforms. Contrary to other internet applications, news organizations benefited from the extensive usage of social media platforms by updating their subscribers' news in almost real-time. Newspapers, tabloids, and magazines are the prime sources of online news, and blogs, social media feeds, and other digital media formats are the latest news media. With the help of social media, any idea can quickly spread to millions of people all over the world. These social media platforms, in their present phase, are very successful and helpful in providing users with the opportunity to debate, exchange, and discuss themes such as democracy, education, and health. However, some people or organizations also utilize these platforms negatively, frequently to obtain financial advantage, occasionally to influence public opinion, and people's attitudes, or to propagate satire or outrageousness. In the past ten years, false news has spread quickly.

A lot of issues have arisen in politics, sports, health, and science, because of the widespread circulation of fake news online. The financial markets are one of these areas where fake news is a problem. A rumor can have terrible results or even stop the market altogether. Someone's ability to make decisions is significantly influenced by the information they take in. There are mounting pieces of evidence that people have behaved unwisely in response to news that afterward turned out to be false. One recent example is the propagation of the new coronavirus, in which false information regarding the virus's origin and behavior is propagated across the Internet, which further leads to chaos and unrest in society.

It would take a tremendous amount of time and resources for people to verify the veracity of every social media post or article. Therefore, more effort should be put into developing automated applications and techniques for determining whether news stories shared on social media are authentic. There is an irresistible need for a low-cost, high-performing model that can accurately categorize the proliferation of fake news or articles while requiring minimal manual effort.

1.2 Project Description

In this project we have used different machine learning models to detect fake news through articles . To evaluate the better result there are two main factors such as quality of data set and quantity of dataset. The three datasets are taken of which two are taken from Kaggle and third dataset is “ISOT Fake news Dataset”. 80% of the data set is training data set and rest 20% is testing data set. In the project the accuracy, f1-score are predicted. There are two types of news: reliable and unreliable.These are the following machine learning algorithms which are used during this project:

1. Naïve Bayes
2. Support Vector Machine
3. Decision Tree
4. Random Forest Algorithm
5. KNN
6. Gradient Boosting Machine
7. Bagging classifier

These are the steps used in Machine Learning for analysing the sentiment:

1. Cleaning
2. Stemming
3. Lemmatization
4. Tokenization and Vectorization

1.3 Report Structure

This report paper has five chapters that include a preface, followed by a literature review, exploration methodology, data findings, and analysis and the last bone is a conclusion with suitable recommendations.

The first chapter provides a transparent summary of this project called Fake News Detection using Machine learning techniques. The explicit chapter, therefore, works to highlight that how different machine learning approaches works. In addition to it, a summary of how the prediction of fake news will happen. It provide us a complete overview of fake news detection. This tells that data set should have good quality and quantity. A good result requires good data set for training.

The second chapter is the literature review that does work on finding the research variables with the use of definitions, an overview, and the past content. The section works on the various sentiment analysis prediction methodologies and allows a clear understanding of many algorithms used for it. Additionally, the second chapter aims to critically appraise the impact of analysis variables on each other. This chapter also tells about the step by step process of many machine learning algorithms used for sentiment analysis.

The third chapter is the methodology; during this chapter, the algorithms are implemented for the sentiment analysis. At this step all the algorithms works on the data set and provide their accuracy and f1-score. In this chapter all the nine ways are implemented. The analysis make use of surveys to gather information and analyse them through the use of Machine Learning.

The fourth chapter is s and discussion, the collected data from primary sources area units analysed through the comparative analysis of algorithms. The illustrate that which algorithm is better for the given data set and what is their accuracy and f1-score. The discussion of result and a comparative analysis of different methodologies is done.

The fifth chapter conclude that the outcome of the analysis paper is terminated and suggestion will be enforced to enhance this analysis shortly.

1.4 Problem Statement

The proposed framework aims at classifying news articles form different domains as a real or fake using statements and publishing author or source of news statements and aims at producing an accuracy of 95% or above using different machine learning algorithms.

1.5 Summary

This chapter introduces the topic of this report and has mentioned the reasons why we research this topic in detail. Background problems related to this research topic have been discussed in this chapter of the dissertation. On the other hand, the aim, objectives, and questions of this research are postulating the purpose of this research. This work is critical to review the problems associated with heart condition prediction and to seek out a far better solution by implementing an appropriate technique into the prediction model.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Machine Leaming Classification and Bayesian networks are some of the core concepts and terminologiescovered in this chapter. Related and existing works on driver drowsiness detection using various machinelearning techniques such as Naive Bayes. Logistic Regression. KNN. Decision Tree. SVM, and others are examined, with an emphasis on what was done, how it was done the classification technique used, the data set used for implementation, the tools used, and the system's result and accuracy.

2.2 Machine Learning

Since 1959, the field of Machine Learning has existed. While working for IBM. Arthur Samuel describedMachine Learning as a branch of research that allows computers to learn without having to be explicitlytaught Various strategies and procedures are used in all sectors to obtain relevant knowledge from unprocessed real-world data. We create a hybrid strategy in this thesis that combines classic statistical methods with model-based machine learning techniques to efficiently identify driver drowsiness detection systems with smaller feature sets and higher accuracy.

In this thesis, we propose a hybrid strategy that combines classic statistical methods with model-basedmachine learning techniques to identify driver drowsiness detection with smaller feature sets and higher accuracy. Machine Learning has been a popular field of Artificial Intelligence (AI) study and application for decades Machine Learning is being applied in self-driving cars, speech recognition, robotic controls, effective web search, and facial detection, to name a few applications.

Machine learning algorithms are divided into four categories:

1. Supervised Learning Algorithm
2. Unsupervised Learning Algorithm
3. Semi-Supervised Learning
4. Reinforcement Learning

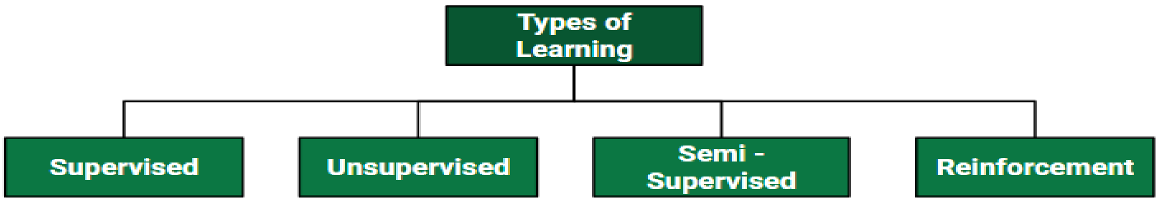


Figure 2.1: Types of Learning

2.2.1 Supervised Learning

Supervised learning is when the model is getting trained on a labelled dataset. A labelled dataset is one that has both input and output parameters. In this type of learning both training and validation, datasets are labelled as shown in the figures below.

While training the model, data is usually split in the ratio of 80:20 i.e., 80% as training data and the rest as testing data. In training data, we feed input as well as output for 80% of data. The model learns from training data only. We use different machine learning algorithms (which we will discuss in detail in the next articles) to build our model. Learning means that the model will build some logicof its own.

Once the model is ready then it is good to be tested. At the time of testing, the input is fed from the remaining 20% of data that the model has never seen before, the model will predict some value and we will compare it with the actual output and calculate the accuracy.

The types of supervised learning are:

1. Classification: It is a Supervised Learning task where output is having defined labels (discrete value). For example, in above Figure A, Output – Purchased has defined labels i.e., 0 or 1; 1 means the customer will purchase, and 0 means that the customer won't purchase. The goal here is to predict discrete values belonging to a particular class and evaluate them on the basis of accuracy. It can be either binary or multi-class classification. In binary classification, the model predicts either0 or 1; yes or no but in the case of multi-class classification, the model predicts more than one class. Example**:** Gmail classifies mails in more than one class like social, promotions, updates, and forums.

2. Regression: It is a Supervised Learning task where output is having continuous value. For example, in above Figure B, Output – Wind Speed is not having any discrete value but is continuous in a particular range. The goal here is to predict a value as much closer to the actual output value as our model can and then evaluation is done by

calculating the error value. The smaller theerror the greater the accuracy of our regression model.

These are examples of supervised learning algorithms:

1. Linear Regression
2. Logistic Regression
3. Nearest Neighbour
4. Gaussian Naive Bayes
5. Decision Trees
6. Support Vector Machine (SVM)
7. Random Forest

2.2.2 Unsupervised Learning

To anticipate the outcome from unlabelled datasets, unsupervised learning is utilized. The most frequent unsupervised learning technique is clustering. Consider a scenario; the unsupervised system is given aninput dataset that contains the photographs of various cats and dogs.

Because the algorithm is never trained on the given dataset, it has no idea what its characteristics are. The purpose of the unsupervised learning algorithm is to recognize visual elements independently. Thistask will be done by dividing the image collection into groups based on image similarity using an unsupervised learning method. The following are some of the most important arguments for the relevance of unsupervised learning:

1. Unsupervised learning is beneficial for extracting relevant information from data.
2. Unsupervised learning is analogous to how a human learns to think via their own experiences, bringingit closer to true Al.
3. Because unsupervised learning works with unlabeled and uncategorized data, it is more important.
4. In the real world, we don't always have input data that corresponds to output; hence we require unsupervised learning to handle these problems.

Some unsupervised learning algorithms are listed below:

1. K-means clustering KNN (k-nearest neighbors)
2. Hierarchical clustering
3. Anomaly detection Neural Networks
4. Principal Component Analysis
5. Independent Component Analysis

6. Apriori algorithm
7. Singular value decomposition

The classification based on unsupervised learning approaches is shown in Figure 2.2

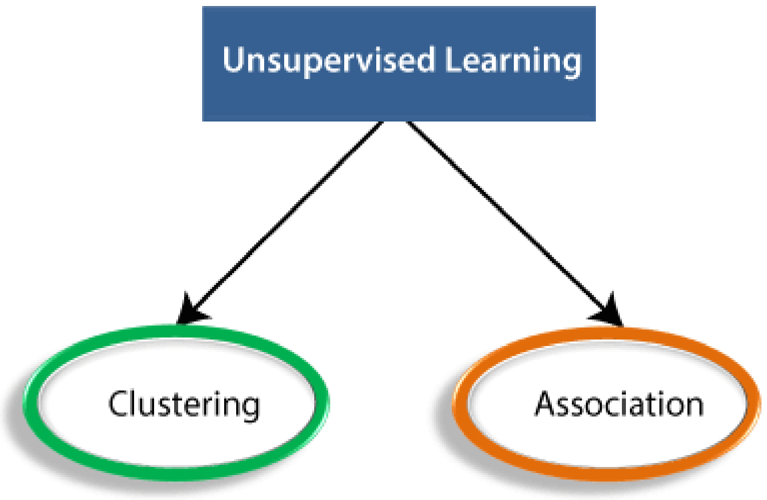


Figure 2.2: Types of Unsupervised learning

1. Clustering

It's essentially an unsupervised learning technique. Unsupervised learning is a technique for extractingreferences from datasets that contain input data but no labelled answers. It's a method for identifying significant structures, explaining underlying processes, and generating traits, and groups in a set of samples. Clustering is the practice of dividing a population or set of data points into various groups so that data points in the same group are more similar and data points in other groups are more dissimilar... It is essentially a collection of objects based on their similarity and dissimilarity.

2. Association

Association rule learning is an unsupervised learning technique that examines the dependency of one data item on another and maps accordingly to make it more profitable. It looks for intriguing relationships or associations between the dataset's variables. It finds interesting links between variablesin a database using a set of criteria. Association rule learning, which is utilized in Market Basket analysis, Web usage mining. continuous manufacturing, and other applications, is one of the most important issues in machine learning. Marketbasket analysis is a technique used by many large retailers to uncover product relationships. We may comprehend it by using the example of a supermarket, where all things purchased together are grouped.

2.2.3 Semi-Supervised Learning

Semi-supervised learning is similar to supervised learning in that it uses both labelled and unlabelled data as seen in figure 2.4. Because unlabelled data is less expensive and easier to obtain, it uses a littleamount of labelled data with a large volume of unlabelled data.

Classification, regression, and prediction can all be employed with this form of learning. When the expense of labelling is too high to allow for a fully labelled training procedure, semi-supervised learningcomes in handy. Identifying a person's face on a webcam is an early example of this. It would be nearly impossible to find a big number of tagged text documents, in this case, therefore semi-supervised learning is suitable. This is because having someone read through full-text documentsmerely to assign a simple classification is inefficient.

As a result, semi-supervised learning enables the algorithm to learn from a small number of labelledtext documents while classifying a large number of unlabelled text documents in the training set. By employing pseudo labelling, semi-supervised learning can train the model with less labelled trainingdata than supervised learning. Many neural network models and training methods can be combined in this way. The following is how it works: Just like in supervised learning, train the model with a limitedamount of labelled training data until it produces good results.

Then use it to forecast the outputs using the unlabelled training dataset, which are faux labels becausethey may not be completely accurate. Connect the labels from the labelled training data to the pseudo labels you made before. Connect thedata inputs in the labelled training data to the data inputs. Then, to reduce error and enhance model accuracy, train the model in the same way you did with the labelled set at the start.

2.2.4 Reinforcement Learning

When an agent learns through trial and error interactions with a dynamic environment, this is known asreinforcement learning. It is strongly related to statistics' decision theory, game theory, and engineering's control theory. The purpose of the machine is to learn to act in such a way that it reducespunishments while maximizing future rewards over its lifetime.

Machine learning includes reinforcement learning. It's all about taking the right steps to maximize yourreward in a given situation. It is used by a variety of software and computers to determine the best feasible action or path in a given situation. Reinforcement learning differs from supervised learning in that supervised learning includes the solution key, allowing the model to be trained with the right answer, whereas reinforcement learning does not. Instead, the reinforcement agent selects what to do to complete the job. It is obligated to learn from its experience in the absence of a training dataset.

Main Points in Reinforcement Learning

1. Input: The input should be a starting state for the model to work from.
2. Output: There are numerous possible outputs, just as there are numerous solutions to agiven problem.
3. Training: The model will return to a state after training, and the user will decide whether to reward orpunish the model depending on its output.
4. The model is always evolving.
5. The optimal solution is determined based on the maximal reward.

Two types of Reinforcement Learning

1. Positive Reinforcement

Positive reinforcement learning entails doing something to increase the likelihood of the desired behaviour occurring again. It improves the agents' behaviour and increases the strength of the conduct. This form of reinforcement can last a long period, but too much positive reinforcement might result inan overload of states, which can lessen the consequences.

2. Negative Reinforcement

The polar opposite of positive reinforcement learning, bad reinforcement learning the likelihood of thegiven behaviour reoccurring by avoiding the unfavourable scenario. Depending on the situation and conduct, it may be more successful than positive reinforcement, although it only offers reinforcement for the bare minimum of activity.

2.3 ALGORITHM

2.3.1 KNN Algorithm

1. Nearest Neighbor is one of the simplest Machine Learning algorithms based on Supervised Learning technique.
2. KNN algorithm assumes the similarity between the new case/data and available cases and put thenew case into the category that is most like the available categories.
3. KNN algorithm stores all the available data and classifies a new data point based on the

similarity.This means when new data appears then it can be easily classified into a well suite category by using KNN algorithm.

4. KNN algorithm can be used for Regression as well as for Classification but mostly it is used for Classification problems.
5. KNN is a non-parametric algorithm, which means it does not make any assumption on underlying data.
6. It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action onthe dataset.
7. KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much like the new data.

Example: Suppose, we have an image of a creature that looks similar to cat and dog, but we want to know either it is a cat or dog. So for this identification, we can use the KNN algorithm, as it works on asimilarity measure.

Our KNN model will find the similar features of the new data set to the cats and dogs images and based on the most similar features it will put it in either cat or dog category.
Suppose there are two categories, i.e., Category A and Category B, and we have a new data point x1, sothis data point will lie in which of these categories. To solve this type of problem, we need a K-NN algorithm. With the help of K-NN, we can easily identify the category or class of a particular dataset.
Consider the below figure of category graph:

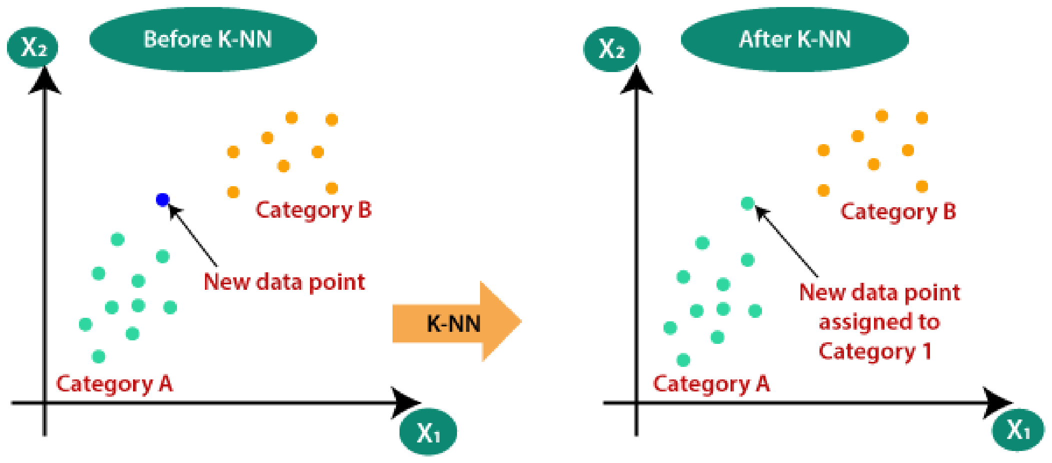


Figure 2.3: Category Graph

The KNN working can be explained on the basis of the below algorithm:

Step-1: Select the number K of the neighbors.

Step-2: Calculate the Euclidean distance of K number of neighbors.

Step-3: Take the K nearest neighbors as per the calculated Euclidean distance.

Step-4: Among these k neighbors, count the number of data points in each category.

Step-5: Assign the new data points to that category for which the number of the neighbor ismaximum.

Step-6: Our model is ready.

Firstly, we will choose the number of neighbors, so we will choose the k=5.

Next, we will calculate the Euclidean distance between the data points. The Euclidean distanceis the distance between two points, which we have already studied in geometry.

By calculating the Euclidean distance, we got the nearest neighbors, as three nearest neighbors incategory A and two nearest neighbors in category B.

Consider the below image:

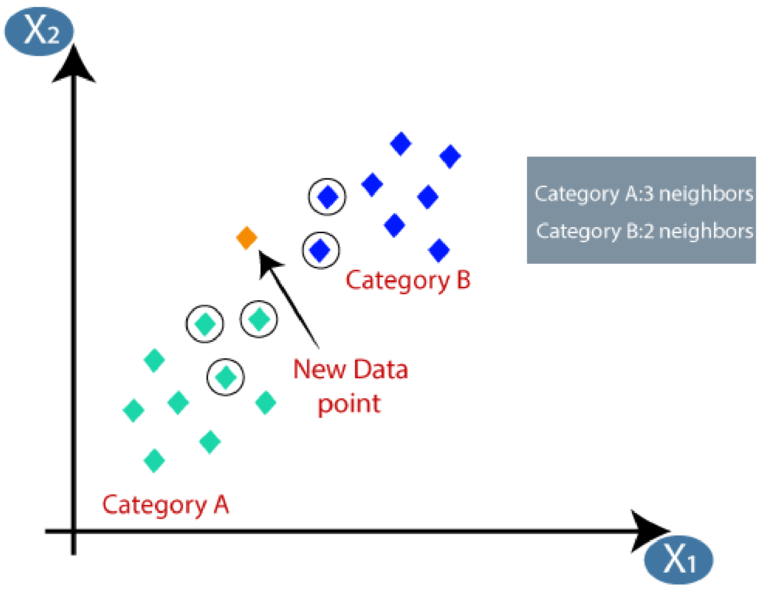


Figure 2.4 KNN Graph

As we can see the 3 nearest neighbors are from category A, hence this new data point must belong tocategory A.

Below are some points to remember while selecting the value of K in the KNN algorithm:

1. There is no particular way to determine the best value for "K", so we need to try some values to findthe best out of them. The most preferred value for K is 5.
2. A very low value for K such as K=1 or K=2, can be noisy and lead to the effects of -outliers in themodel.
3. Large values for K are good, but it may find some difficulties.

2.3.2 Support Vector Machine (SVM)

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classificationproblems in Machine Learning. The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n- dimensional space into classes so that we can easily put the new data point in the correct category in thefuture. This best decision boundary is called a hyperplane.

SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine. Consider the belowdiagram in which there are two different categories that are classified using a decision boundary or hyperplane:

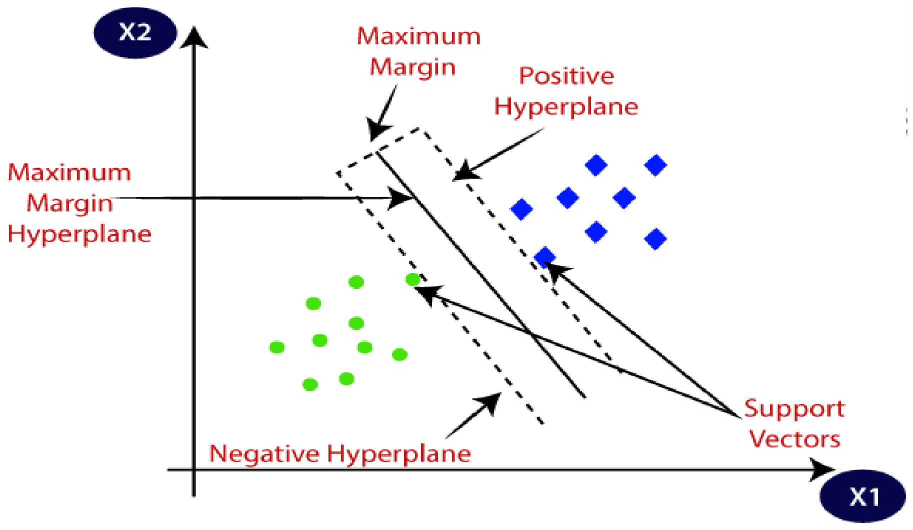


Figure 2.5: Hyperplane used in SVM

Example: SVM can be understood with the example that we have used in the KNN classifier. Supposewe see a strange cat that also has some features of dogs, so if we want a model that can accurately identify whether it is a cat or dog, so such a model can be created by using the SVM algorithm. We will first train our model with lots of images of cats and dogs so that it can learn about different features of cats and dogs, and then we test it with this strange creature. So as support vector creates a decision boundary between these two data (cat and dog) and choose extreme cases (support vectors), itwill see the extreme case of cat and dog. On the basis of the support vectors, it will classify it as a cat. Consider the below diagram:

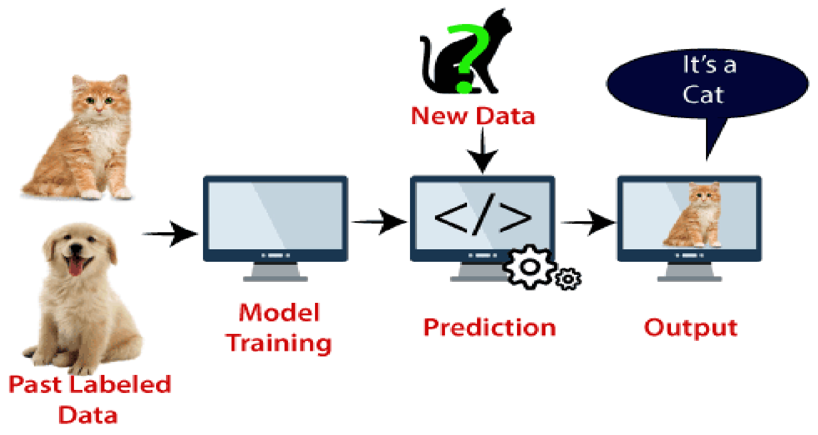


Figure 2.6: Working of SVM

SVM algorithm can be used for Face detection, image classification, text categorization, etc. The types of SVM are:

1. Linear SVM: Linear SVM is used for linearly separable data, which means if a dataset can be classifiedinto two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.
2. Non-linear SVM: Non-Linear SVM is used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and classifier used is called as Non-linear SVM classifier.

Kernel Function: is a method used to take data as input and transform it into the required form of processing data. “Kernel” is used due to a set of mathematical functions used in Support Vector Machine providing the window to manipulate the data. So, Kernel Function generally transforms the training set of data so that a non-linear decision surface is able to transform to a linear equation in a higher number of dimension spaces. Basically, it returns the inner product between two points in a standard feature dimension.

Linear Kernel is used when the data is Linearly separable, that is, it can be separated using a singleLine. It is one of the most common kernels to be used. It is mostly used when there are a Large number of Features in a particular Data Set. One of the examples where there are a lot of features, is Text Classification, as each alphabet is a new feature. So we mostly use Linear Kernel in Text Classification.

Sigmoid Kernel function is equivalent to a two-layer, perceptron model of the neural network, which is used as an activation function for artificial neurons.

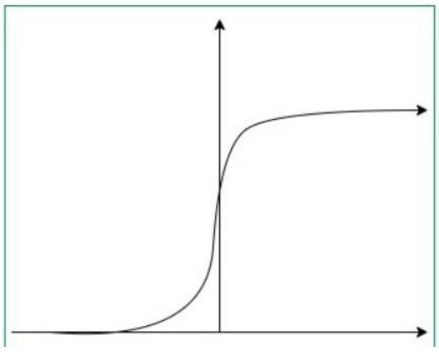


Figure 2.7: Sigmoidal Graph

Polynomial Kernel represents the similarity of vectors in the training set of data in a feature space over polynomials of the original variables used in the kernel.

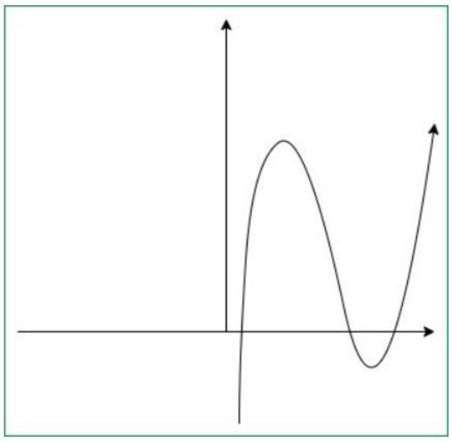


Figure 2.8: Polynomial Graph

Gaussian Kernel Radial Basis Function (RBF) is same as above kernel function, adding radial basismethod to improve the transformation.

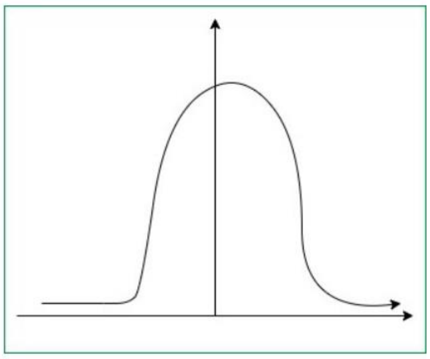


Figure 2.9: Gaussian Graph

2.3.3 Logistic Regression

Harrell et al. (2001) proposed a logistic regression model. It's a statistical data analysis method that works with binary dependent variables. Using logistic regression techniques, the logistic model parameter is estimated. A logistic model is a linear combination of independent variables that determines the likelihood of an event. It is not a classification model in general; rather, it models the output probability using the provided inputs. By adjusting the cut-off values, it is frequently used as a classifier. As a result, the variables below the cut-off values belong to one class, while those above thecut-off values belong to another. The two most common types of logistic regression are multinomial and ordinal logistic regression.

Multinomial logistic regression works with absolute values and divides output values into more thantwo groups. Ordinal logistic regression is the process of sorting the various outputs produced by a multinomial regression model. The techniques of logistic regression. A logistic model is a linear combination of independent variables that calculates the likelihood of an event.

It is not a classification model in general; rather, it models the output probability using the provided inputs. By adjusting the cut-off values, it is frequently used as a classifier. As a result, the variables below the cut-off values belong to one class, while those above the cut-off values belong to another. Thetwo most common types of logistic regression are multinomial and ordinal logistic regression.
Multinomial logistic regression works with absolute values and divides output values into more than two groups. The ordinal logistic regression process is used to order the many outputs produced by a multinomial regression model.

It is not a classification model in general; rather, it models the output probability using the provided inputs. By adjusting the cut-off values, it can be utilized as a classifier. As a result, the variables belowthe cut-off values belong to one class, while those above the cut-off values belong to another. Figure shows the logistic function.

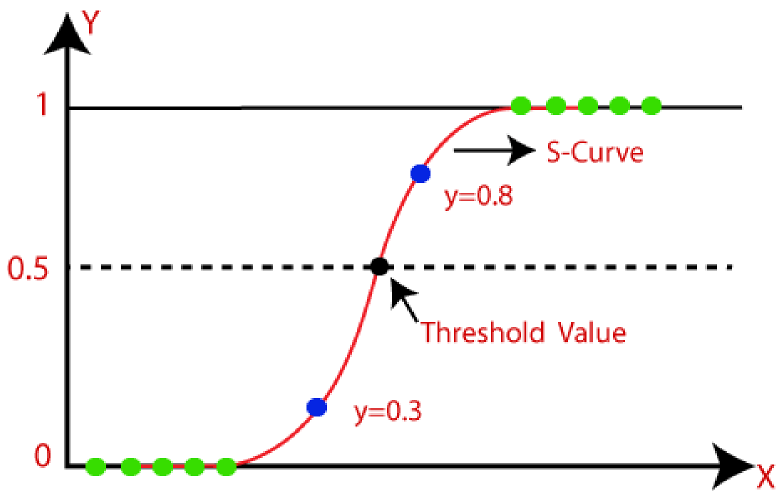


Figure 2.10: Logistic Function

The Linear Regression equation can be used to calculate the Logistic Regression equation. Mathematical steps to obtain Logistic Regression equations:

Equation for the straight line can be written as:

$$y \ = \ b0 + blxl + b2x2 + \ b3x3 + \ldots + box \tag{1}$$

Because y in Logistic Regression can only be between 0 and 1, divide the previousequation by (1-y):

$$y/1 - y : 0 \ for \ y = 0, and \ infinity \ for \ y = 1 \tag{2}$$

However, we require a range of infinity to [infinity, in which case the logarithmof the equation is:

$$log[y/1 - y] \ = \ b0 + b1x1 + b2x2 \ + b3x3 + \ldots + bnxn \tag{3}$$

Above is the final equation of logistic regression.

2.3.4 Naïve Bayes

Naive Bayes is a machine learning algorithm that is based on Bayes' theorem. It is used for classification problems, where the goal is to predict the class of a given input based on its features. Naive Bayes is called "naive" because it makes a strong assumption that the features are conditionally independent given the class. This means that the presence or absence of a particular feature does not affect the presence or absence of any other feature, given the class label. Although this assumption is often not true in practice, Naive Bayes has been shown to work well in many real-world applications. The algorithm works by first computing the prior probability of each class based on the training data. Then, for each feature, it computes the conditional probability of the feature given the class. Finally, it combines these probabilities using Bayes' theorem to compute the posterior probability of each class given the input.

For example, a fruit may be considered to be an apple if it is red, round, and about 3 inches in diameter. Even if these features depend on each other or upon the existence of the other features, all of these properties independently contribute to the probability that this fruit is an apple and that is why it is known as ‘Naive’.

An NB model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods.

Bayes theorem provides a way of computing posterior probability P(c|x) from P(c), P(x) and P(x|c).

Look at the equation below:

$$P\left(\frac{A}{B}\right) = \frac{P\left(\frac{B}{A}\right) \cdot P(A)}{P(B)} \tag{4}$$

Where
P(A | B) is the posterior probability of A given B,
P(B | A) is the likelihood of B given A,
P(A) is the prior probability of A,
and P(B) is the evidence probability (which is a normalization constant).

In Naive Bayes, we make a simplifying assumption that the features are conditionally independent given the class label. This means that the presence or absence of a particular feature does not affect the presence or absence of any other feature, given the class label. Mathematically, this can be expressed as follows:

$$P(x1, x2, \ldots, xn \mid A) = P(x1 \mid A) * P(x2 \mid A) * \ldots * P(xn \mid A) \tag{5}$$

where x1, x2, ..., xn are the features of the input.

Using this assumption, we can rewrite Bayes' theorem as follows:

$$P(A \mid x1, x2, \ldots, xn) = P(x1 \mid A) * P(x2 \mid A) * \ldots * P(xn \mid A) * P(A) / P(x1, x2, \ldots, xn) \quad (6)$$

We can compute the likelihood P(x | A) and prior P(A) from the training data. The evidence probability P(x) is a normalization constant that ensures that the probabilities add up to 1.

To classify a new input, we simply compute the posterior probabilities for each class and select the class with the highest probability. Naive Bayes has several advantages, including its simplicity, efficiency, and ability to handle high-dimensional data. It has been successfully applied to a wide range of applications, including text classification, spam filtering, and sentiment analysis. However, Naive Bayes also has some limitations, such as its assumption of independence, which can lead to poor performance when the features are strongly correlated. Additionally, it may not work well when the training data is imbalanced or when there are rare events that are not well represented in the training data.

Naive Bayes has been successfully applied to a wide range of applications, particularly in the fields of natural language processing and text classification. Some common applications of Naive Bayes include:

1. Email spam filtering: Naive Bayes can be used to classify emails as spam or not spam based on their content and other features.
2. Sentiment analysis: Naive Bayes can be used to classify text as positive, negative, or neutral based on the words and phrases used in the text.
3. News article categorization: Naive Bayes can be used to classify news articles into different categories such as sports, politics, entertainment, etc., based on their content.
4. Medical diagnosis: Naive Bayes can be used to diagnose diseases based on symptoms and other medical data.
5. Fraud detection: Naive Bayes can be used to detect fraudulent transactions based on various features such as transaction amount, location, etc.
6. Customer segmentation: Naive Bayes can be used to segment customers into different groups based on their demographics, buying habits, and other features.
7. Image classification: Naive Bayes can be used to classify images into different categories based on their features such as colour, texture, and shape.
8. Recommendation System: Naive Bayes Classifier and Collaborative Filtering together builds a Recommendation System that uses machine learning and data mining techniques to filter unseen information and predict whether a user would like a given resource or not.

Overall, Naive Bayes is a versatile algorithm that can be applied to many different types of classification problems, particularly those involving high-dimensional data such as text and images.

2.3.5 Decision Tree

These learning methods are commonly utilized in the healthcare industry. Figure 2.6 depicts the decision tree types model. A root node, branches, and leaf nodes are all present in each decision tree. The root node is at the top, while the remaining nodes are called leaf or branch nodes. On one or moreproperties of the given data, the internal node applies the decision rule or a test. The output is defined by the branch node.

Classification is the process of accurately classifying input data and mapping it to its appropriate classes in predictive analysis. Labelled and unlabelled data are the two most categories of data. There are several predictor attributes and a single target attribute in the labelled data. The class label is represented by each value of the target characteristics. Only the predictor attributes are present in theunlabelled attributes. The classification process' primary goal is to accurately predict the class of unlabelled data using classification models developed from labeled instances (historical data).

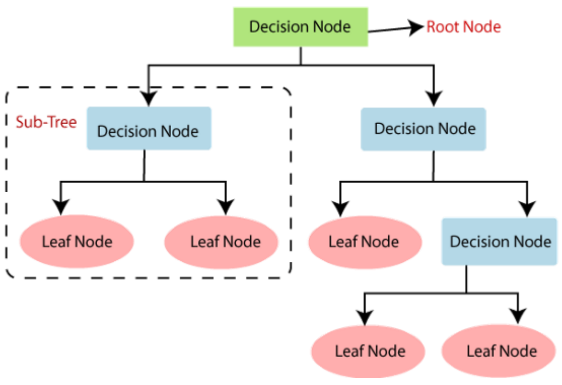


Figure 2.11: Decision Tree

The Decision Tree classifier will be our initial algorithm. It is currently one of the most used machine learning algorithms. They're used to solve both classification and regression issues. Now you might bewondering why we chose the Decision Tree Classifier over other classifiers. We can give two reasons to answer the question. One example of an algorithm is decision trees, which attempt to replicate the way the human brain thinks so that it is relatively straightforward to comprehend the data and reach sound conclusions or interpretations. Second, rather than being a black box algorithm like SVM. NN, and others, decision trees allow us to understand the reasoning for the data to interpret. It has the advantage of being straightforward, making it one of the most popular among programmers of this generation.

Now that we've established why a Decision Tree is beneficial, let's take a closer look at what a Decision Tree Classifier is. To begin, a decision tree is a tree with several nodes, each of which represents a feature (attribute), each link (branch) represents a decision (rule), and each leaf of the treerepresents an outcome (categorical or continuous value). The idea is to make a tree out of all of the data and receive a result at each leaf. We now have a better understanding of what a decision tree is. Let's get started talking about how to make a decision tree classifier. Two alternative algorithms can beused to create a decision tree. CART (Classification and Regression Trees) is one, and ID3 is another (iterative Dichotomiser 3). For ID3, we start with the column's x value and a y value that stays at the column's last position and only has "YES" or "NO" values. The x values for the chart above are (outlook, temp, humidity, and windy) and the value is play, which only has two options: "YES" or "NO" and is at the bottom of the column. Now we must map the x and y coordinates. Because this is a binary classification problem, we'll use the ID3 technique to construct the tree.

As a general guideline, the root node should be the characteristic that has the most impact on the value y. Then we move on to the next node, which is the most influential feature. We'll employ the concept of entropy, which is a measure of how much uncertainty there is in a data set. For the binary classification task, we must calculate the entropy for all categorical values. To summarize, we must first compute the entropy for the data set. Then, for each attribute/feature, we must first calculate entropy for all categorical values, then take the average value information entropy for the current attribute, and then determine how much we have gained. Then we need to choose the highest gain attribute and repeat it till we have the tree we want. That is the ID3 process now. As previously stated,the Decision Tree Classifier is based on a different method known as CART, which stands for classification and regression trees. We utilize the Gini Index as our cost function to evaluate splits in the dataset in our approach. Because our target variable is binary, it can only have two values (yes and no).

Now we must calculate the Gini score, which will help us determine how to divide the data. If the Giniscore is 0, we can consider it a perfect separation, while a 50/50 split is the worst-case scenario. The difficulty now is how to determine the Gini index value. The Gini index will remain identical if the target variable is a category variable with numerous levels. The initial step in this procedure is to compute the gain index for the data- set. Then we must calculate the Gini index for all category values,determine the average information entropy for the current attribute, and finally calculate the Gini gain for each feature.

2.3.6 Random Forest Algorithm

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model. As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output. The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

Since the random forest combines multiple trees to predict the class of the dataset, it is possible that some decision trees may predict the correct output, while others may not. But together, all the trees predict the correct output. Therefore, below are two assumptions for a better Random forest classifier:

1. There should be some actual values in the feature variable of the dataset so that the classifier can predict accurate results rather than a guessed result.
2. The predictions from each tree must have very low correlations.

The Working process can be explained in the below steps:

Step-1: Select random K data points from the training set.

Step-2: Build the decision trees associated with the selected data points (Subsets).

Step-3: Choose the number N for decision trees that you want to build.

Step-4: Repeat Step 1 & 2.

Step-5: For new data points, find the predictions of each decision tree, and assign the new data pointsto the category that wins the majority votes.

Advantages of Random Forest

1. Random Forest is capable of performing both Classification and Regression tasks.
2. It is capable of handling large datasets with high dimensionality.
3. It enhances the accuracy of the model and prevents the overfitting issue

2.3.7 Bagging

Bagging, also known as Bootstrap aggregating, is an ensemble learning technique that helps to improve the performance and accuracy of machine learning algorithms. It is used to deal with bias-variance trade-offs and reduces the variance of a prediction model.

In 1996, Leo Breiman introduced the bagging algorithm, which has three basic steps:

1. Bootstrapping: Bagging leverages a bootstrapping sampling technique to create diverse samples. This resampling method generates different subsets of the training dataset by selecting data points at random and with replacement. This means that each time you select a data point from the training dataset, you are able to select the same instance multiple times. As a result, a value/instance repeated twice (or more) in a sample.
2. Parallel training**:** These bootstrap samples are then trained independently and in parallel with each other using weak or base learners.
3. Aggregation: Finally, depending on the task (i.e., regression or classification), an average or a majority of the predictions are taken to compute a more accurate estimate. In the case of regression, an average is taken of all the outputs predicted by the individual classifiers; this is known as soft voting. For classification problems, the class with most votes is accepted; this is known as hard voting or majority voting.

Bagging avoids overfitting of data and is used for both regression and classification models, specifically for decision tree algorithms. Bagging requires more computational resources than a single model, but it can significantly improve the prediction accuracy and generalization of the model, especially for large and complex datasets.

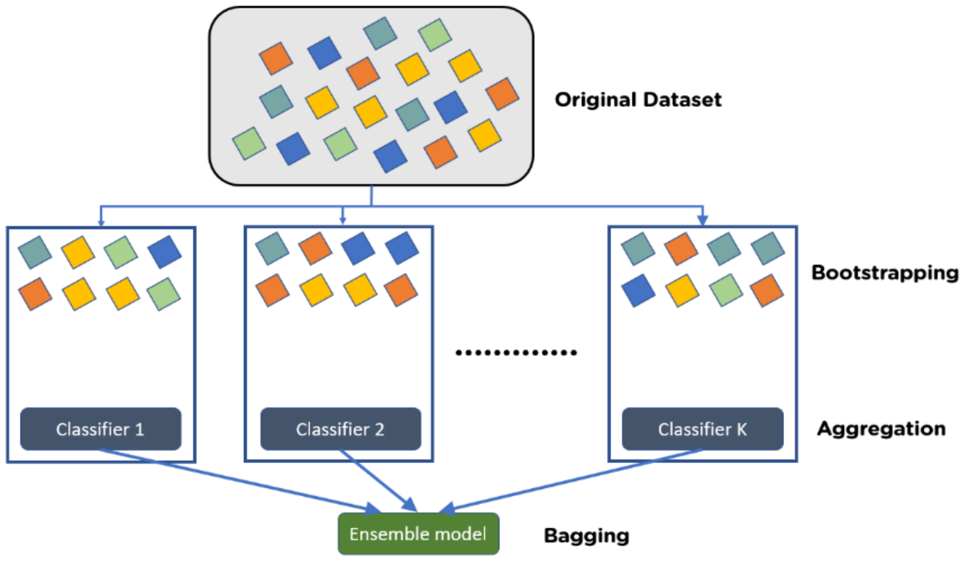


Figure 2.12: Bagging Classifier

Advantages of Bagging are:

1. Improved prediction accuracy: Bagging can improve the accuracy of a model by reducing the variance and bias of the individual models. This is especially effective when the individual models have high variance and are prone to overfitting.
2. Increased stability: Bagging can make the model more stable by reducing the impact of outliers and noise in the training data. This is because it averages the predictions of multiple models that are trained on different subsets of the data.
3. Feature importance: Bagging, particularly Random Forest, provides feature importance scores that can be used for feature selection and understanding the importance of each feature in the prediction.

Disadvantages of Bagging are:

1. Increased complexity: Bagging requires training multiple models, which increases the computational complexity and time required for training.
2. Reduced interpretability: Bagging can make it harder to interpret the model because it combines the predictions of multiple models. It can be difficult to understand which features are most important and how they contribute to the prediction.
3. Limited improvement: Bagging may not improve the model's accuracy if the individual models are already accurate or if the dataset is small.

In summary, bagging can be an effective method for improving prediction accuracy and stability, but it comes with some trade-offs such as increased complexity and reduced interpretability. Its usefulness depends on the specific problem and dataset being used.

2.3.8 XGBoost

XGBoost is an optimized distributed gradient boosting library designed for efficient and scalable training of machine learning models. It is an ensemble learning method that combines the predictions of multiple weak models to produce a stronger prediction. XGBoost stands for "Extreme Gradient Boosting" and it has become one of the most popular and widely used machine learning algorithms due to its ability to handle large datasets and its ability to achieve state-of-the-art performance in many machine learning tasks such as classification and regression.

XGBoost is used for supervised learning tasks such as regression and classification. It is based on the idea of iteratively adding weak learners to a strong learner, where each weak learner attempts to correct the errors of the previous ones. XGBoost builds a model that predicts the target variable by combining the predictions of the weak learners. It uses a gradient descent approach to minimize the loss function and update the model parameters.

One of the advantages of XGBoost is its scalability, as it can handle large datasets with many features and observations. It also has a built-in parallel processing capability that can speed up the training process on multi-core CPUs or distributed systems. XGBoost can handle various types of data, such as numerical, categorical, and binary data, without requiring any pre-processing or feature engineering. It can also handle missing values by assigning them to the most similar group based on the available data.

XGBoost is efficient handling of missing values, which allows it to handle real-world data with missing values without requiring significant pre-processing. Additionally, XGBoost has built-in support for parallel processing, making it possible to train models on large datasets in a reasonable amount of time.

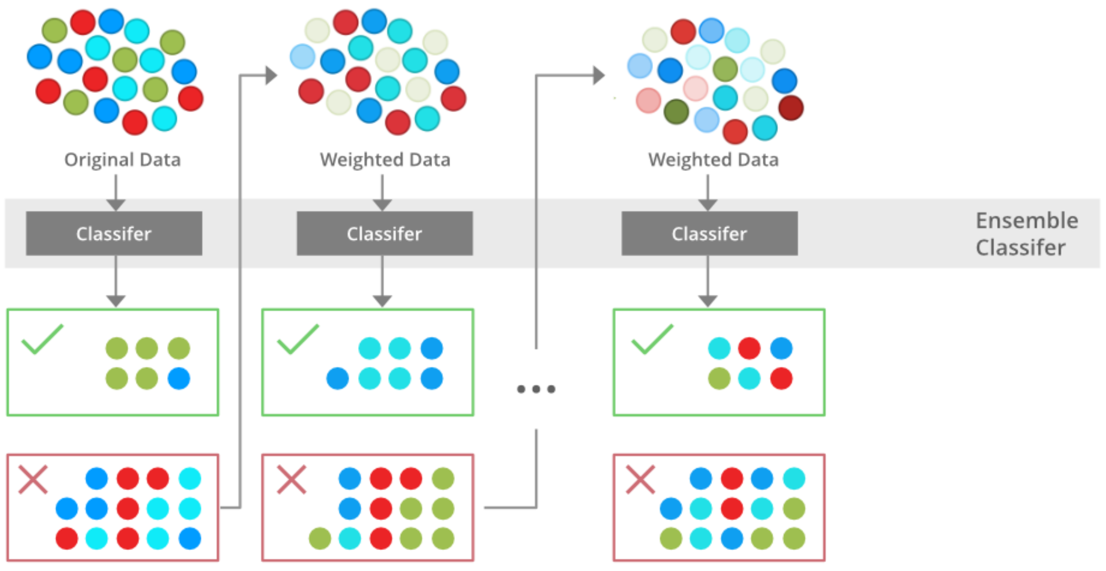


Figure 2.13: XGBoost Classifier

Advantages of XGBoost are:

1. Performance: XGBoost has a strong track record of producing high-quality results in various machine learning tasks, especially in Kaggle competitions, where it has been a popular choice for winning solutions.
2. Scalability: XGBoost is designed for efficient and scalable training of machine learning models, making it suitable for large datasets.

3. Customizability: XGBoost has a wide range of hyperparameters that can be adjusted to optimize performance, making it highly customizable.
4. Handling of Missing Values: XGBoost has built-in support for handling missing values, making it easy to work with real-world data that often has missing values.
5. Interpretability: Unlike some machine learning algorithms that can be difficult to interpret, XGBoost provides feature importance, allowing for a better understanding of which variables are most important in making predictions.

Disadvantages of XGBoost are:

1. Computational Complexity: XGBoost can be computationally intensive, especially when training large models, making it less suitable for resource-constrained systems.
2. Overfitting: XGBoost can be prone to overfitting, especially when trained on small datasets or when too many trees are used in the model.
3. Hyperparameter Tuning: XGBoost has many hyperparameters that can be adjusted, making it important to properly tune the parameters to optimize performance. However, finding the optimal set of parameters can be time-consuming and requires expertise.
4. Memory Requirements: XGBoost can be memory-intensive, especially when working with large datasets, making it less suitable for systems with limited memory resources.

XGBoost can be used in a variety of applications, including Kaggle competitions, Predictive modelling, Image and speech recognition, Natural Language Processing (NLP), Recommender systems and click-through rate prediction, among others. It is also highly customizable and allows for fine-tuning of various model parameters to optimize performance.

2.3.9 AdaBoost

AdaBoost (Adaptive Boosting) is a machine learning algorithm that is used for classification and regression problems. It is an ensemble learning algorithm, which means that it combines the predictions of several weak learners (i.e., classifiers that perform only slightly better than random guessing) to make a final prediction.

The basic idea behind AdaBoost is to iteratively train a series of weak classifiers on the same dataset, with each subsequent classifier focusing more on the misclassified examples from the previous classifiers. At each iteration, the algorithm assigns a weight to each example in the training set based on its difficulty to classify, where difficult examples are given higher weights. The algorithm then trains a new classifier on the weighted training set and updates the weights of the examples based on the classifier’s performance.

During the testing phase, the algorithm combines the predictions of the weak classifiers using a weighted majority voting scheme, where the weight of each classifier is proportional to its performance on the training data.

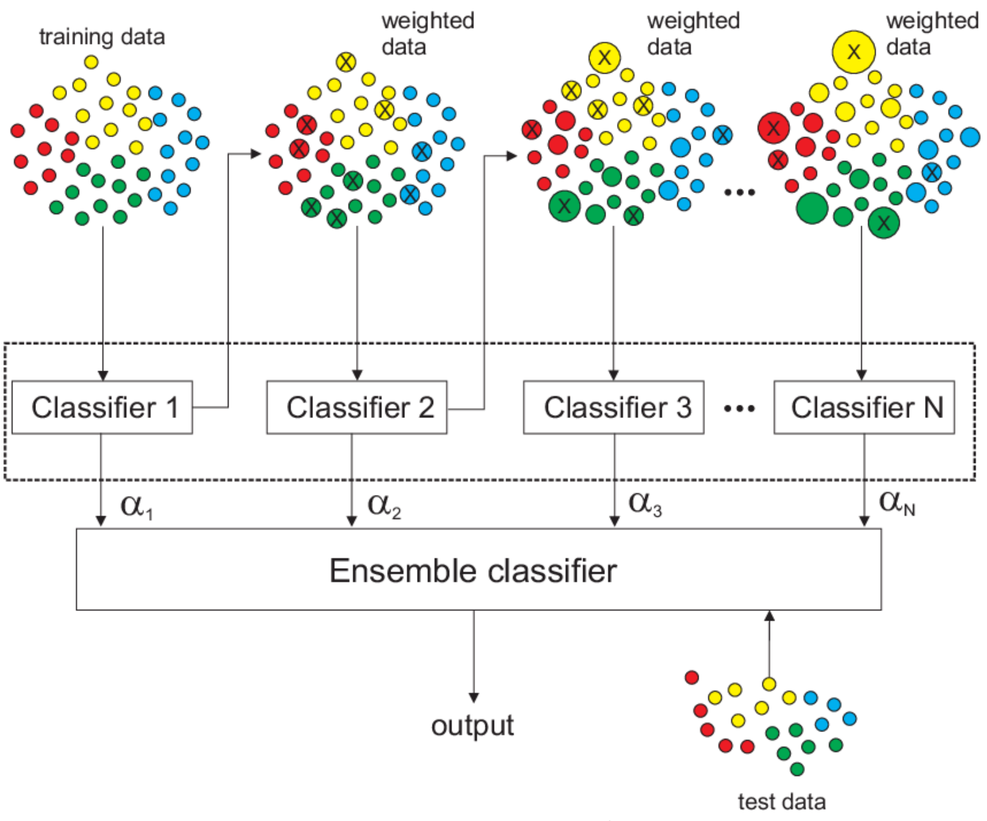


Figure 2.14: AdaBoost Classifier

There are several advantages to using AdaBoost (Adaptive Boosting) in machine learning, including:

1. Improved classification accuracy: AdaBoost can significantly improve the accuracy of classification compared to using a single classifier or a non-ensemble method.
2. Handles complex data: AdaBoost can handle complex classification problems, including those with high-dimensional data and non-linear decision boundaries.
3. Reduces overfitting: AdaBoost has a low risk of overfitting, which is a common problem with other machine learning algorithms.
4. No prior knowledge required: AdaBoost does not require prior knowledge about the data or the classification problem, making it useful for a wide range of applications.
5. Handles noisy data: AdaBoost is robust to noisy data and outliers, which can be a problem for other machine learning algorithms.
6. Can be combined with other algorithms: AdaBoost can be combined with other machine learning algorithms to further improve classification accuracy.

Although AdaBoost (Adaptive Boosting) is a powerful machine learning algorithm with many advantages, there are also some potential disadvantages to using it.

Here are a few:

1. Sensitive to noisy data: While AdaBoost is generally robust to noisy data and outliers, it can also be sensitive to them. In some cases, noisy data can actually decrease the performance of the algorithm.
2. Computationally intensive: AdaBoost can be computationally intensive, especially when dealing with large datasets. This is because it requires multiple iterations to train the weak classifiers, which can be time-consuming.
3. Requires sufficient training data: AdaBoost requires a sufficient amount of training data to perform well. If the dataset is too small, or if there are too few examples of certain classes, the algorithm may not be able to learn an accurate classifier.
4. Biased towards over-represented classes: AdaBoost can be biased towards over-represented classes in the training data, which can lead to poor performance on under-represented classes.
5. Can be sensitive to the choice of weak learner: AdaBoost is sensitive to the choice of weak learner used to train each iteration. If the weak learner is too complex or not expressive enough, the algorithm may not be able to learn an accurate classifier.

Despite these potential disadvantages, AdaBoost remains a popular and effective machine learning algorithm for many classification problems. As with any machine learning method, it is important to carefully consider the strengths and limitations of the algorithm and to choose the appropriate approach for the specific problem at hand.

CHAPTER - 3

PROPOSED METHODOLOGY

3.1 Proposed methodology

The framework as reflected in figure 3.1 depicts that the data is collected from various credible sources which include Kaggle [20] and the University of California [21]. The data is then preprocessed using the inbuilt library in Python NLTK (Natural Language Toolkit) [22] then feature extraction is performed on text using another inbuilt library of Python called TF-IDF Vectorizer [23]. Following this 9 machine learning algorithms have been used to train a model to classify if a news article is either reliable or non-reliable.

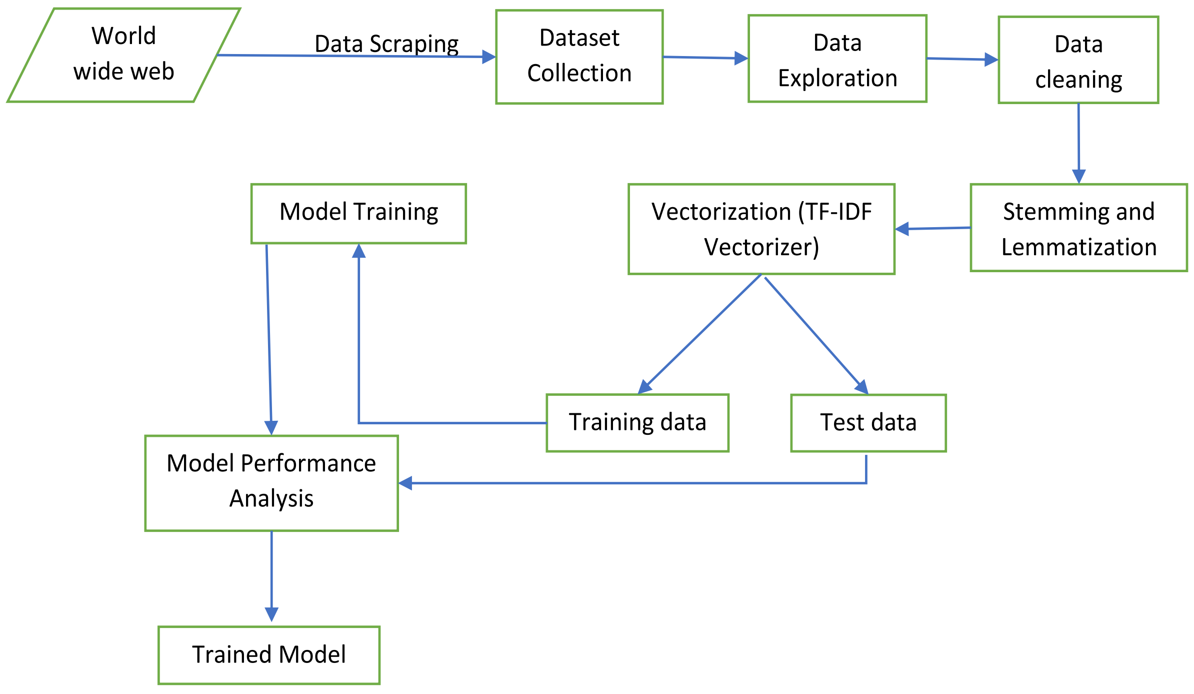
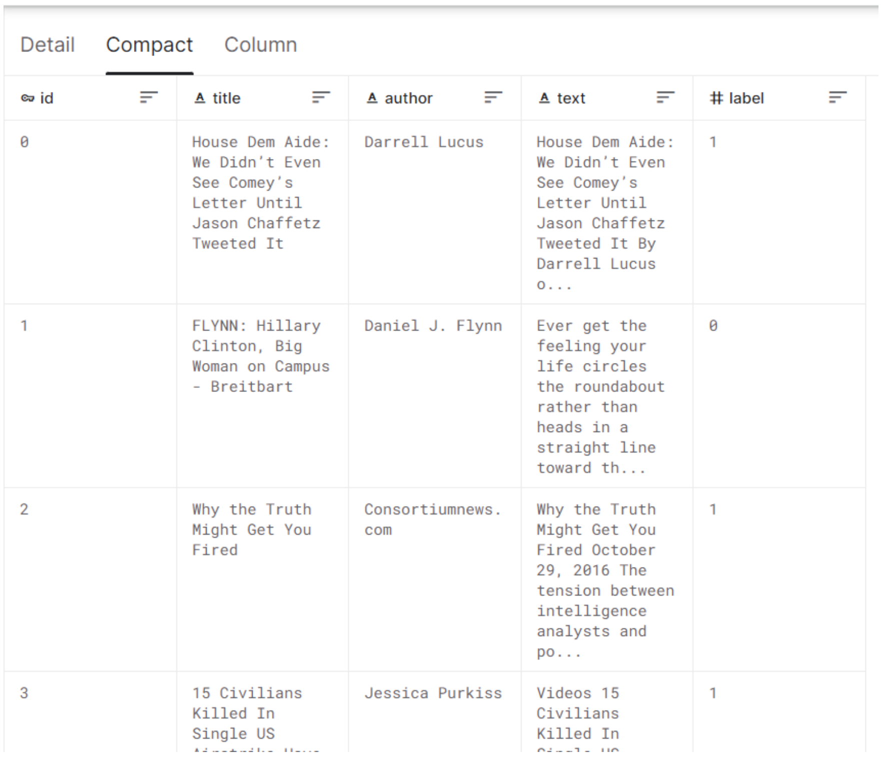


Figure 3.1: Proposed framework

3.2 Data Gathering

This work includes 3 datasets that are used for training the model and finally testing it. The datasets consist of news from various genres which include news related to politics, entertainment, technology, and sports. The datasets are also having a mix of both true and fake news that is well labelled.

The first dataset is a dataset taken from Kaggle. The data consists of 5 columns which include fields id, author, title, text, and a label. The dataset is divided into train and test data in which the training dataset contains 20387 rows of data, on the other hand, the testing data contains 5127 rows of data.



id	title	author	text	label
0	House Dem Aide: We Didn't Even See Comey's Letter Until Jason Chaffetz Tweeted It	Darrell Lucus	House Dem Aide: We Didn't Even See Comey's Letter Until Jason Chaffetz Tweeted It By Darrell Lucus o...	1
1	FLYNN: Hillary Clinton, Big Woman on Campus - Breitbart	Daniel J. Flynn	Ever get the feeling your life circles the roundabout rather than heads in a straight line toward th...	0
2	Why the Truth Might Get You Fired	Consortiumnews.com	Why the Truth Might Get You Fired October 29, 2016 The tension between intelligence analysts and po...	1
3	15 Civilians Killed In Single US	Jessica Purkiss	Videos 15 Civilians Killed In	1

Figure 3.2: Dataset 1

Link to the dataset: https://www.kaggle.com/competitions/fake-news/overview
Description of the dataset: It contains 2 csv files which are train.csv and test.csv.
train.csv is a full training dataset with the following attributes:

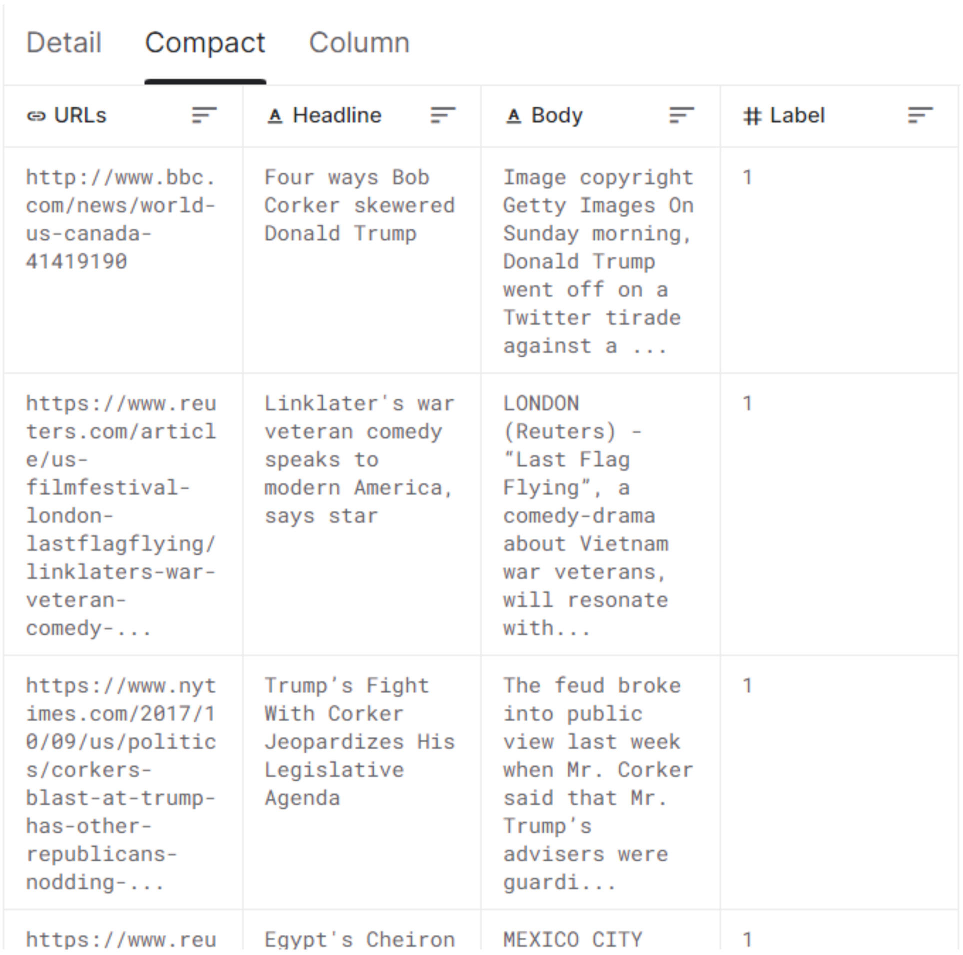
1. id: unique id for a news article
2. title: the title of a news article
3. author: author of the news article
4. text: the text of the article; could be incomplete.
5. label: a label that marks the article as potentially unreliable.

    1: unreliable

0: reliable

test.csv: A testing training dataset with all the same attributes at train.csv without the label.

The second dataset (containing 3,352 articles) is accessible on Kaggle. The authentic articles are sourced from The New York Times, CNN, Reuters, and other reputable online sources, whereas false news is sourced from news websites that have been known to publish unreliable information. This dataset covers a variety of topics including politics, entertainment, and sports.



URLs	Headline	Body	Label
http://www.bbc.com/news/world-us-canada-41419190	Four ways Bob Corker skewered Donald Trump	Image copyright Getty Images On Sunday morning, Donald Trump went off on a Twitter tirade against a ...	1
https://www.reuters.com/article/us-filmfestival-london-lastflagflying/linklaters-war-veteran-comedy-...	Linklater's war veteran comedy speaks to modern America, says star	LONDON (Reuters) - "Last Flag Flying", a comedy-drama about Vietnam war veterans, will resonate with...	1
https://www.nytimes.com/2017/10/09/us/politics/corkers-blast-at-trump-has-other-republicans-nodding-...	Trump's Fight With Corker Jeopardizes His Legislative Agenda	The feud broke into public view last week when Mr. Corker said that Mr. Trump's advisers were guardi...	1
https://www.reu	Egypt's Cheiron	MEXICO CITY	1

Figure 3.3: Dataset 2

Link to the dataset: https://www.kaggle.com/datasets/jruvika/fake-news-detection

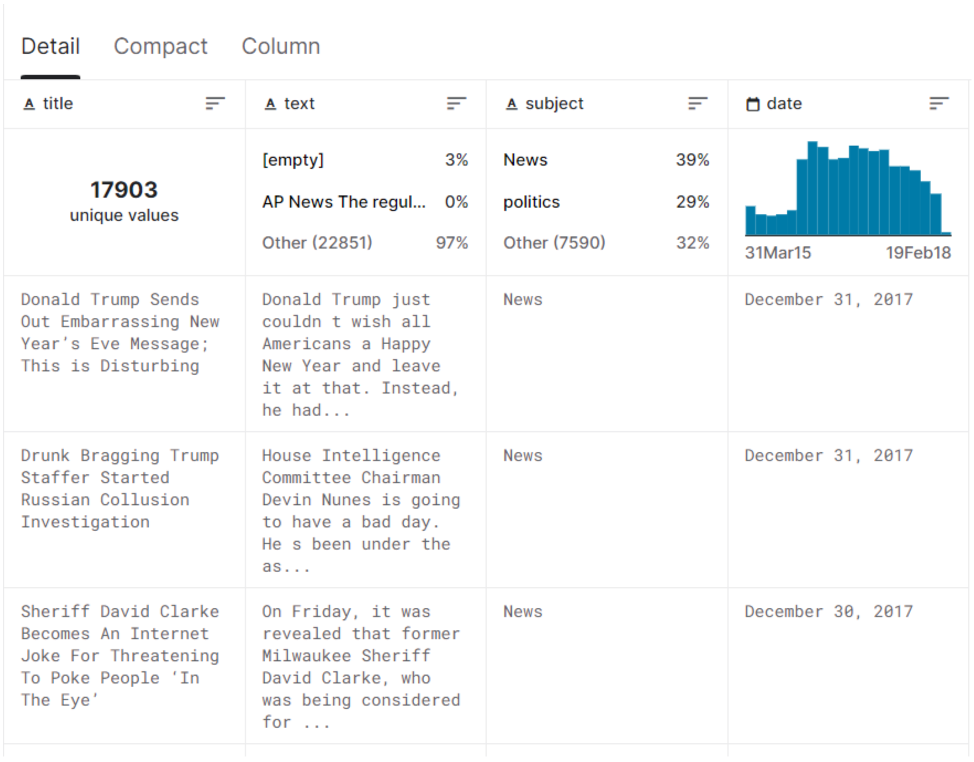
Description of the dataset: It contains 2 csv files which are train.csv and test.csv.

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

1. URL: Link for the news article
2. Headline: the title of a news article
3. Body: the text of the article; could be incomplete
4. label: a label that marks the article as potentially unreliable.

test.csv: A testing training dataset with all the same attributes at train.csv without the label.

The third dataset "ISOT Fake News Dataset", is a dataset that is widely used as a benchmark for evaluating fake news classification problems. The dataset is a collection of articles that are present on a news website called reuters.com. On the other hand, there are fake articles that are a compilation of news from sources that Politifact.com has identified as unreliable. The articles are a mix of various genres but most of them focus on political news. A total of 44,898 articles are present of which 21,417 are true while the rest are false.



title	text	subject	date
17903 unique values	[empty] 3% AP News The regul... 0% Other (22851) 97%	News 39% politics 29% Other (7590) 32%	31Mar15 19Feb18
Donald Trump Sends Out Embarrassing New Year's Eve Message; This is Disturbing	Donald Trump just couldn t wish all Americans a Happy New Year and leave it at that. Instead, he had...	News	December 31, 2017
Drunk Bragging Trump Staffer Started Russian Collusion Investigation	House Intelligence Committee Chairman Devin Nunes is going to have a bad day. He s been under the as...	News	December 31, 2017
Sheriff David Clarke Becomes An Internet Joke For Threatening To Poke People 'In The Eye'	On Friday, it was revealed that former Milwaukee Sheriff David Clarke, who was being considered for ...	News	December 30, 2017

Figure 3.4: Dataset 3

Link to the dataset: https://www.kaggle.com/datasets/csmalarkodi/isot-fake-news-dataset

Description of the dataset:

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

1. subject: genre for the news article
2. title: the title of a news article
3. text: the text of the article; could be incomplete.
4. Date: date of publishing of the article
5. label: a label that marks the article as potentially unreliable.

   1: unreliable

   0: reliable

test.csv: A testing training dataset with all the same attributes at train.csv without the label.

3.3 Data Pre-Processing

The data that is present in the dataset is in form of published articles hence it is still having many inconsistencies and comprises null values, non-ASCII characters, meaningless spaces, hashtags, and other symbols that are not useful in the evaluation process. The pre-processing includes 7 steps that ensure the removal of all inconsistencies and unwanted characters.

1. The first step was removing all the null-valued rows, which are the rows that have no values entered in them. Null-valued rows provide a false output when any function is applied.
2. The second step was to add an empty string to the columns where a value is missing to remove inconsistencies.
3. The third step was to convert the data into lowercase so that it can be easily operated on and to maintain a standard for the data.
4. The fourth step was removing all non-ASCII characters found in the data. The non-ASCII characters have no meaning in the evaluation since they cannot be handled through a vectorizer.
5. The fifth step includes the removal of all mentions (names following @) and all the URLs in the data. The next step was to filter out hashtags, extra numbers, and spaces that have no significant meaning. All the above-mentioned pre-processing is done using a library called NLTK (Natural Language Toolkit) It has applications in semantic reasoning, categorization, tokenization, stemming, tagging, and parsing, among others.
6. The final step uses port stemmer to convert all the words and reduce them into their stem words or root words. Following stemming, the data at hand is a group of words that are in their root form, but a machine learning algorithm can only work on numerical data that is where the vectorizer comes in. The pre-processed data is then given to the vectorizer known as TF-IDF Vectorizer. This commonly used technique transforms text into equivalent numerical representations to be used in machine learning algorithms.

3.3.1 TF-IDF Vectorizer

TF-IDF stands for Term Frequency Inverse Document Frequency of records. It can be defined as the calculation of how relevant a word in a series or corpus is to a text. The meaning increases proportionally to the number of times in the text a word appears but is compensated by the word frequency in the corpus (dataset).

Terms used are:

1. Term Frequency: In document d, the frequency represents the number of instances of a given word t. Therefore, we can see that it becomes more relevant when a word appears in

the text, which is rational. Since the ordering of terms is not significant, we can use a vector to describe the text in the bag of term models. For each specific term in the paper, there is an entry with the value being the term frequency. The weight of a term that occurs in a document is simply proportional to the term frequency.

$$\mathrm{tf}\,(\mathrm{t},\mathrm{d}) \;=\; \text{count of t in d / number of words in d} \tag{7}$$

2. Document Frequency: This tests the meaning of the text, which is very similar to TF, in the whole corpus collection. The only difference is that in document d, TF is the frequency counter for the term t, while df is the number of occurrences in the document set N of the term t. In other words, the number of papers in which the word is present is DF.
   df(t) = occurrence of t in documents
3. Inverse Document Frequency**:** The key aim of the search is to locate the appropriate records that fit the demand. Since tf considers all terms equally significant, it is therefore not only possible to use the term frequencies to measure the weight of the term in the paper. First, find the document frequency of a term t by counting the number of documents containing the term:

$$df(t) \;=\; N(t) \tag{8}$$

   where
   df(t) = Document frequency of a term t
   N(t) = Number of documents containing the term t

Term frequency is the number of instances of a term in a single document only; although the frequency of the document is the number of separate documents in which the term appears, it depends on the entire corpus. The IDF of the word is the number of documents in the corpus separated by the frequency of the text.

$$idf(t) \;=\; N/\,df(t) \;=\; N/N(t) \tag{9}$$

Code for TF-IDF Vectoriser:

```
from sklearn.feature_extraction.text import TfidfVectorizer
vectorizer = TfidfVectorizer()
vectorizer.fit(X)
X = vectorizer.transform(X)
```

3.3.2 Data Before Preprocessing:

Table 3.1: Data Before Preprocessing

Id	text	label	content
0	Donald Trump Sends Out Embarrassing New Yearâ€™s Eve Message; This is Disturbing	1	Donald Trump just couldn’t wish all Americans a Happy N Drunk Bragging Trump Staffer Started Russian Collusion Investigation New Year and leave it at that.
1	Drunk Bragging Trump Staffer Started Russian Collusion Investigation	1	House Intelligence Committee Chairman Devin Nunes is going to have a bad day. He s been under the assumption, like many of us
2	Sheriff David Clarke Becomes An Internet Joke For Threatening To Poke People â€˜In The Eyeâ€™	1	On Friday, it was revealed that former Milwaukee Sheriff David Clarke, who was being considered for Homeland Security Secretary in Donald Trump s administration
3	Trump Is So Obsessed He Even Has Obamaâ€™s Name Coded Into His Website (IMAGES)	1	On Christmas day, Donald Trump announced that he would be back to work the following day, but he is golfing for the fourth day in a row.
4	Pope Francis Just Called Out Donald Trump During His Christmas Speech	1	Pope Francis used his annual Christmas Day message to rebuke Donald Trump without even mentioning his name.

3.3.3 Data After Preprocessing:

Table 3.2: Data After Preprocessing

id	text	label	content
0	donald trump wish american happi new year leav thatinsteadg	1	donald trump send embarrass new year eve message disturb donald trump wish american happi new year
1	hous intellig committe chairman devin nune go bad dayassumptionlik mani uschristoph steel	1	drunk brag trump staffer start russian collus investig hous intellig committe chairman devin nune go bad
2	fridayrev former milwauke sheriff david clarkeconsid homeland secur secretari donald trump administrationemail	1	sheriff david clark becom internet joke threaten poke peopl eye fridayrev former milwauke sheriff david clarkeconsid
3	christma daydonald trump announc wouldback workfollow daygolf fourth day rowform realiti show star blast former	1	trump obsess even obama name code websiteimag christma daydonald trump announc wouldback workfollow daygolf fourth day
4	pope franci use annual christma day messag rebuk donald trump without even mention namepop deliv messag day member unit nation condemn	1	pope franci call donald trump christma speech pope franci use annual christma day messag rebuk donald trump without even mention

3.3.4 Splitting Dataset:

Data splitting is when data is divided into two or more subsets. Typically, with a two-part split, one part is used to evaluate or test the data and the other to train the model.

Data splitting is an important aspect of data science, particularly for creating models based on data. This technique helps ensure the creation of data models and processes that use data models such as machine learning are accurate.

3.4 Algorithms Used:

The problem at hand puts forth a scenario where two classes are there, and machine learning is used to classify the news articles into one of these classes. The first class is reliable and the other is unreliable or fake news. In these types of scenarios, we use an algorithm that can perform the task of binary classification which include Logistic regression, K-nearest neighbors, Linear SVM, Random Forest, Bagging classifier, Decision trees, AdaBoost, XGBoost, and Naive Bayes Classifier.

3.4.1 Logistic Regression:

Logistic regression implementation code:

```
from sklearn.linear_model import LogisticRegression
model=LogisticRegression()
model.fit(X_train, y_train)
```

3.4.2 K-Nearest Neighbors:

KNN implementation code:

```
from sklearn.neighbors import KNeighborsClassifier
knn_model=KNeighborsClassifier(n_neighbors=1)
knn_model.fit(X_train,y_train)
```

3.4.3 Random Forest:

Random Forest Algorithm implementation code:

```
from sklearn.ensemble import RandomForestClassifier
model1=RandomForestClassifier(n_estimators=200)
```

3.4.4 Decision Tree:

Decision Tree implementation code:

```
from sklearn.tree import DecisionTreeClassifier
dtc = DecisionTreeClassifier(criterion='entropy')
dtc.fit(X_train, y_train)
```

3.4.5 Linear Support Vector Machine:

The SVM algorithm helps to find the best line or decision boundary; this best boundary or region is called a hyperplane. SVM algorithm finds the closest point of the lines from both the classes. These points are called support vectors. The distance between the vectors and the hyperplane is called margin. And the goal of SVM is to maximize this margin. The hyperplane with maximum margin is called the optimal hyperplane.

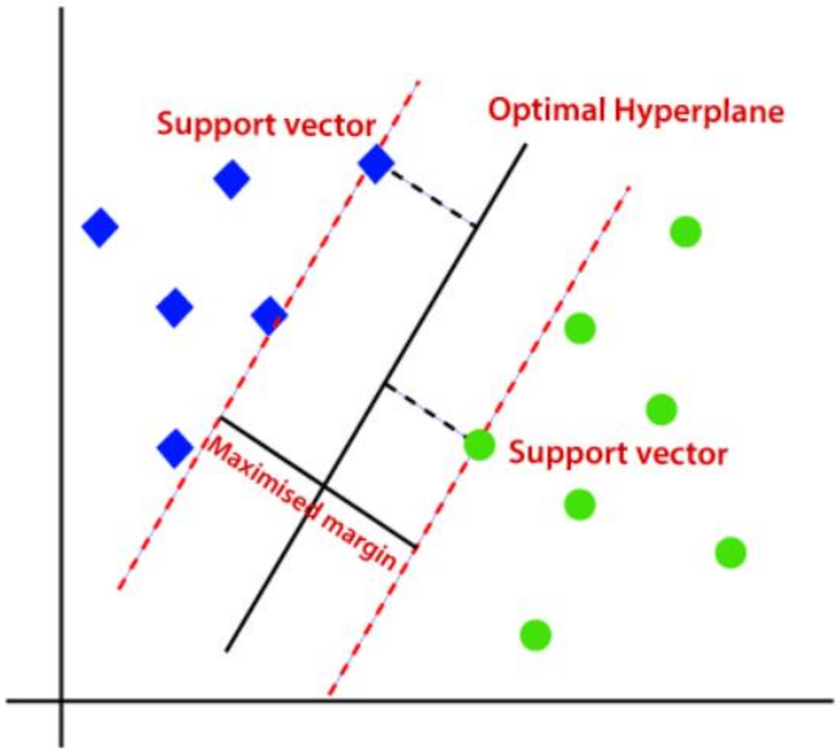


Figure 3.5: SVM Algorithm

SVM implementation code:

```
from sklearn.svm import LinearSVC
svc_model=LinearSVC()
```

3.4.6 Bagging:

Bagging algorithm implementation code:

```
from sklearn.ensemble import BaggingClassifier
bagging = BaggingClassifier(LinearSVC(), max_samples=0.5, max_features=0.5)
```

3.4.7 AdaBoost:

AdaBoost implementation code:

```
from sklearn.ensemble import AdaBoostClassifier
clf1 = AdaBoostClassifier(n_estimators=100)
```

3.4.8 XGBoost:

XGBoost implementation code:

```
from sklearn.ensemble import GradientBoostingClassifier
clf2=GradientBoostingClassifier(n_estimators=100, learning_rate=1.0,max_depth=1)
```

3.4.9 Naïve Bayes Classifier:

Naïve Bayes Classifier implementation code:

```
from sklearn.naive_bayes import MultinomialNB
clf=MultinomialNB()
clf.fit(X_train,y_train)
```

CHAPTER 4

RESULTS AND DISCUSSION

The accuracy of each method on the three datasets under consideration is summarized in Table 4.1. The Linear SVM attained a maximum accuracy of 96.5% on DS1 (Kaggle Fake News Dataset 1). The accuracy of boosting algorithms like Adaboost and XGboost was 95%. The least accurate models were K-Nearest Neighbors and Naive Bayes, with scores of 45% and 71%, respectively. On DS1, ensemble learning algorithms achieve an accuracy of 93.5% on average, compared to an accuracy of 66.7% for individual learners. Individual and Ensemble learners differ significantly, with a 32.2% absolute difference between the two groups. The algorithms with the best results on Dataset 2 are Linear SVM and Bagging Classifier, which achieve an accuracy of 97%. Although individual learners reported an average accuracy of 94.3%, the average accuracy of the ensemble learners was 95.6%, making the difference between the two non-significantly different by 1.3%. On Dataset 3, logistic regression. and bagging algorithms both achieved an accuracy of 98%, while Linear SVM reported an accuracy of 99%. K-Nearest Neighbors reported the lowest accuracy, at 57.5%. The average accuracy for individual and ensemble learners was 88.7% and 97.1%, respectively.

Table 4.1: Overall accuracy for each dataset

Accuracy Score	**Dataset 1**	**Dataset 2**	**Dataset 3**
K- Nearest Neighbors	0.45	0.905	0.575
Linear SVM	0.965	0.973	0.99
Logistic regression	0.939	0.961	0.979
Naïve-Bayes	0.711	0.94	0.947
Decision Tree	0.913	0.936	0.946
Random Forest	0.885	0.943	0.976
Bagging	0.944	0.97	0.98
XG boost	0.957	0.945	0.963
Ada boost	0.956	0.966	0.965

The average accuracy of each algorithm across the three datasets is shown in Figure 4.1. Overall, the bagging classifier and linear SVM algorithms perform the best among ensemble learners (accuracy of 96.4% and 97.6%, respectively), while K-Nearest Neighbors perform the worst (accuracy of 65.8%). The accuracy of individual learners is 87.8%, whereas that of ensemble learners is 95.37%. Only dataset 2 (90.5%) had a high accuracy score with K-Nearest Neighbors, while datasets 1 and 3 received poor results. On datasets 2 and 3, naive Bayes also performed well (accuracy of 94%), but not on dataset 1 (accuracy of 71%). Recall, precision, and F1-score are additional metrics we use to evaluate the performance of learning models because the accuracy score alone is not a trustworthy measure of a model's performance.

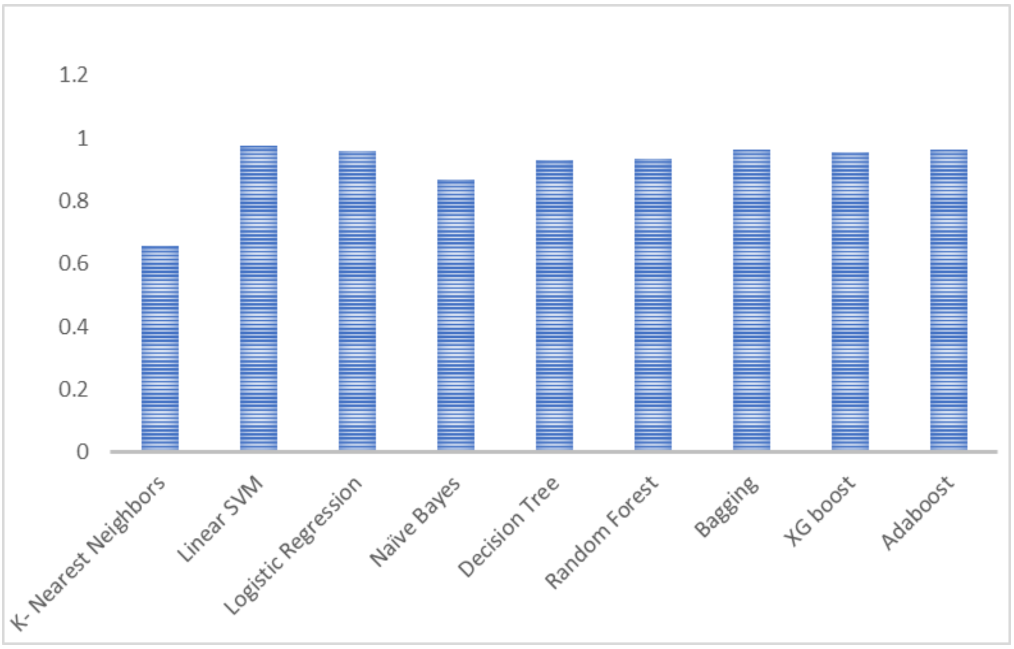


Figure 4.1: Average Accuracy Score over all datasets

Tables 4.2-4.4 review each algorithm's precision, recall, and f1-score for each of the three datasets. In terms of average precision (Table 3), Linear SVM performed best among individual learners, while the Bagging classifier performed best among ensemble learners. On all three datasets, the linear SVM and bagging classifiers' average precision is 97%. The average precision of random forest (RF) increased to 96.9% on the three datasets after deleting the dataset with the lowest score, DS2. Random forest (RF) had previously attained a precision of 95%. The K-Nearest Neighbors algorithm has the lowest precision score.

Table 4.2: Overall precision for each dataset

Precision	**Dataset 1**	**Dataset 2**	**Dataset 3**
K-Nearest Neighbors	0.44	0.902	0.55
Linear SVM	0.969	0.97	0.989
Logistic regression	0.96	0.95	0.978
Naïve-Bayes	0.998	0.903	0.933
Decision Tree	0.907	0.954	0.939
Random Forest	0.959	0.916	0.979
Bagging	0.967	0.97	0.983
XG Boost	0.957	0.962	0.964
Ada boost	0.953	0.929	0.963

According to the recall performance metric, the Boosting Algorithm (Adaboost) outperforms the Ensemble Learners with a recall score of 0.96, while the Linear SVM performs best among Individual Learners with a recall score of 0.971. Following closely behind are the K-Nearest Neighbors technique and the boosting classifier (XGBoost), both of which had recall values of 0.953 and 0.963 respectively. On the F1- score, the algorithms performed similarly to how they did for precision. The best F1-score of any technique was attained by the linear SVM, which had an F1-score of 0.97. The bagging classifier and Adaboost learning algorithm came in second with F1-scores of 0.959.

Table 4.3: Overall Recall for each dataset

Recall	**Dataset 1**	**Dataset 2**	**Dataset 3**
K-Nearest Neighbors	0.998	0.893	0.998
Linear SVM	0.95	0.973	0.992
Logistic regression	0.89	0.967	0.981
Naïve-Bayes	0.334	0.975	0.969
Decision Tree	0.891	0.906	0.959
Random Forest	0.768	0.967	0.976
Bagging	0.902	0.965	0.979
XG Boost	0.943	0.954	0.964
AdaBoost	0.945	0.965	0.97

Table 4.4: Overall F1-Score for each dataset

F1-Score	**Dataset 1**	**Dataset 2**	**Dataset 3**
K-Nearest Neighbors	0.61	0.897	0.71
Linear SVM	0.959	0.971	0.99
Logistic regression	0.928	0.958	0.98
Naïve-Bayes	0.5	0.938	0.951
Decision Tree	0.89	0.93	0.949
Random Forest	0.85	0.941	0.977
Bagging	0.93	0.967	0.981
XG Boost	0.95	0.941	0.964
Ada boost	0.949	0.963	0.966

Figure 4.2 shows the average performance of learning algorithms utilizing precision, recall, and f1-score across the three datasets. It is evident that the effectiveness of learning algorithms utilizing different performance criteria does not differ significantly with K- Nearest Neighbors and Naïve Bayes being exceptions.

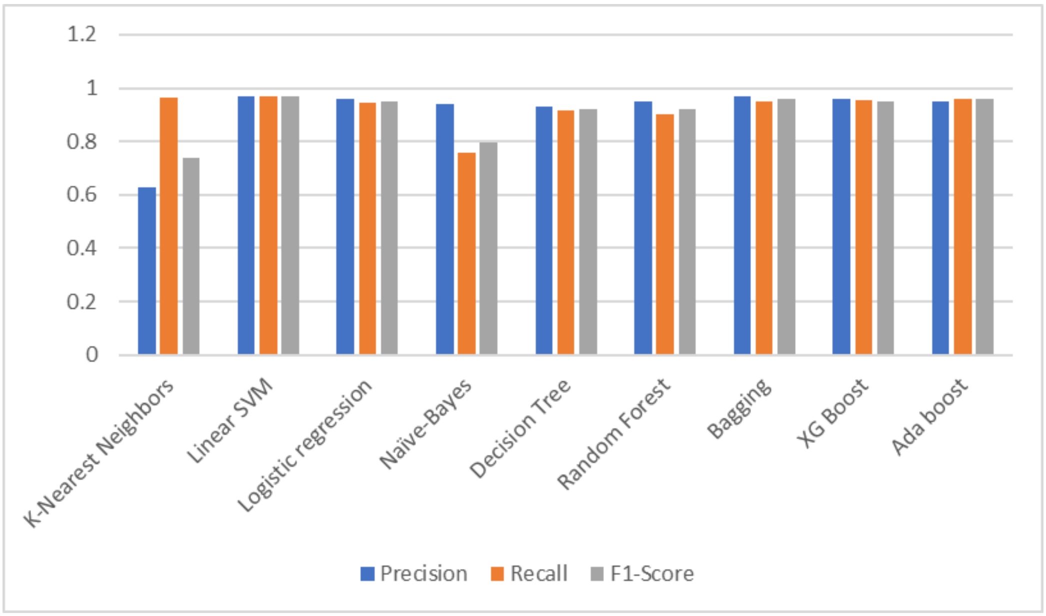


Figure 4.2: Overall Precision, Recall, and F1-Score over all datasets

On all performance criteria, the ensemble learner XGboost outperforms other learning models. The main reason for the increased performance of the software is XGBoost operating philosophy, which efficiently discovers defects and minimizes them in each cycle. Using multiple classifications and regression trees (CART), which include numerous weak learners, to give misclassified data points higher weights, is the essential notion behind how XGBoost functions. For the model to reliably identify the erroneously classified points on each successive iteration, overfitting is reduced using regularization parameters.

Despite being a relatively straightforward model, logistic regression has an average accuracy of nearly 95% across the three datasets. The high average accuracy can be attributed to a few factors. First, a thorough grid search with numerous hyperparameters is used to optimize the logistic regression model. Second, the similar writing styles of some authors in some datasets (like DS2) help explain the logistic regression model's 96% accuracy.

CHAPTER 5

CONCLUSION AND FUTURE SCOPE

Fake news identification and detection, when performed manually, require a lot of detailed knowledge of the field and proficiency to recognize abnormalities in the news article. This work tries to consider the problem of identifying fake news using machine learning algorithms. We used data from Kaggle and the ISOT Research Lab for our work, which contained news articles from various fields, to deal with the majority of the news and not any specific news. The research's main goal is to find textual characteristics that distinguish genuine news from fraudulent news. The different features required for classification were identified using the TF-IDF vectorizer, and the identified feature set was used as input to the machine learning models. Comparatively speaking, some models achieved higher accuracy than others. Results were compared using different performance metrics such as precision, recall, and f1-score for each algorithm. On overall performance, ensemble learning algorithms achieved a much better score than individual learning algorithms.

Fake news identification currently has many problems that require attention from research teams. Machine learning can be used to identify the causes of the spread of unreliable news. Future work can be done to identify fake news in live news articles and stop the spread of fake news.

REFERENCES

[1] Z. Shahbazi and Y. C. Byun, "Fake Media Detection Based on Natural Language Processing and Blockchain Approaches," *IEEE Access*, vol. 9, pp. 128442–128453, 2021, doi: 10.1109/ACCESS.2021.3112607.

[2] M. F. Mridha, A. J. Keya, M. A. Hamid, M. M. Monowar, and M. S. Rahman, "A Comprehensive Review on Fake News Detection with Deep Learning," *IEEE Access*, vol. 9, pp. 156151–156170, 2021, doi: 10.1109/ACCESS.2021.3129329.

[3] N. Seddari, A. Derhab, M. Belaoued, W. Halboob, J. Al-Muhtadi, and A. Bouras, "A Hybrid Linguistic and Knowledge-Based Analysis Approach for Fake News Detection on Social Media," *IEEE Access*, vol. 10, pp. 62097–62109, 2022, doi: 10.1109/ACCESS.2022.3181184.

[4] T. Jiang, J. P. Li, A. U. Haq, A. Saboor, and A. Ali, "A Novel Stacking Approach for Accurate Detection of Fake News," *IEEE Access*, vol. 9, pp. 22626–22639, 2021, doi: 10.1109/ACCESS.2021.3056079.

[5] D. Rohera *et al.*, "A Taxonomy of Fake News Classification Techniques: Survey and Implementation Aspects," *IEEE Access*, vol. 10, pp. 30367–30394, 2022, doi: 10.1109/ACCESS.2022.3159651.

[6] A. Tariq, A. Mehmood, M. Elhadef, and M. U. G. Khan, "Adversarial Training for Fake News Classification," *IEEE Access*, vol. 10, pp. 82706–82715, 2022, doi: 10.1109/ACCESS.2022.3195030.

[7] W. Shishah, "JointBert for Detecting Arabic Fake News," *IEEE Access*, vol. 10, pp. 71951–71960, 2022, doi: 10.1109/ACCESS.2022.3185083.

[8] M. Umer, Z. Imtiaz, S. Ullah, A. Mehmood, G. S. Choi, and B. W. On, "Fake news stance detection using deep learning architecture (CNN-LSTM)," *IEEE Access*, vol. 8, pp. 156695–156706, 2020, doi: 10.1109/ACCESS.2020.3019735.

[9] H. Ali *et al.*, "Analyzing the Robustness of Fake-news Detectors under Black-box Adversarial Attacks," *IEEE Access*, 2021, doi: 10.1109/ACCESS.2021.3085875.

[10] M. Carter, M. Tsikerdekis, and S. Zeadally, "Approaches for Fake Content Detection: Strengths and Weaknesses to Adversarial Attacks," *IEEE Internet Comput*, vol. 25, no. 2, pp. 73–83, Mar. 2021, doi: 10.1109/MIC.2020.3032323.

[11] K. A. Qureshi, R. A. S. Malick, M. Sabih, and H. Cherifi, "Complex Network and Source Inspired COVID-19 Fake News Classification on Twitter," *IEEE Access*, vol. 9, pp. 139636–139656, 2021, doi: 10.1109/ACCESS.2021.3119404.

[12] E. Elsaeed, O. Ouda, M. M. Elmogy, A. Atwan, and E. El-Daydamony, “Detecting Fake News in Social Media Using Voting Classifier,” *IEEE Access*, vol. 9, pp. 161909–161925, 2021, doi: 10.1109/ACCESS.2021.3132022.

[13] K. Xu, F. Wang, H. Wang, and B. Yang, “Detecting Fake News Over Online Social Media via Domain Reputations and Content Understanding,” 2020.

[14] Ravish, R. Katarya, D. Dahiya, and S. Checker, “Fake News Detection System Using Featured-Based Optimized MSVM Classification,” *IEEE Access*, vol. 10, pp. 113184–113199, Nov. 2022, doi: 10.1109/access.2022.3216892.

[15] M. Sudhakar and K. P. Kaliyamurthie, “A Machine Learning Framework for Automatic Fake News Detection in Indian News,” 2022, doi: 10.21203/rs.3.rs-2268597/v1.

[16] S. Lorent and A. Itoo, “Fake News Detection Using Machine Learning,” 2018.

[17] A. Preda, S. Ruseti, S.-M. Terian, and M. Dascălu, “Romanian Fake News Identification using Language Models,” in *RoCHI - International Conference on Human-Computer Interaction*, 2022, pp. 73–79. doi: 10.37789/rochi.2022.1.1.13.

[18] D. Dementieva, M. Kuimov, and A. Panchenko, “Multiverse: Multilingual Evidence for Fake News Detection,” Nov. 2022, [Online]. Available: http://arxiv.org/abs/2211.14279

[19] D. Li, H. Guo, Z. Wang, and Z. Zheng, “Unsupervised Fake News Detection Based on Autoencoder,” *IEEE Access*, vol. 9, pp. 29356–29365, 2021, doi: 10.1109/ACCESS.2021.3058809.

[20] “Kaggle: Your Home for Data Science.” https://www.kaggle.com/ (accessed Nov. 23, 2022).

[21] “Fake News Detection Datasets - University of Victoria.” https://www.uvic.ca/ecs/ece/isot/datasets/fake-news/index.php (accessed Nov. 23, 2022).

[22] “NLTK :: Natural Language Toolkit.” https://www.nltk.org/ (accessed Nov. 23, 2022).

[23] “sklearn.feature_extraction.text.TfidfVectorizer — scikit-learn 1.1.3 documentation.” https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html (accessed Nov. 23, 2022).

[24] “Fake News | Kaggle.” https://www.kaggle.com/competitions/fake-news/overview (accessed Nov. 23, 2022).

[25] “Fake News detection | Kaggle.” https://www.kaggle.com/datasets/jruvika/fake-news-detection (accessed Nov. 24, 2022).

[26] E. Loper and S. Bird, “NLTK,” 2002, pp. 63–70. doi: 10.3115/1118108.1118117.

[27] S. Qaiser and R. Ali, “Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents,” *Int J Comput Appl*, vol. 181, no. 1, pp. 25–29, Jul. 2018, doi: 10.5120/ijca2018917395.

[28] C. Y. J. Peng, K. L. Lee, and G. M. Ingersoll, “An introduction to logistic regression analysis and reporting,” *Journal of Educational Research*, vol. 96, no. 1, pp. 3–14, 2002, doi: 10.1080/00220670209598786.

[29] P. Hilaire Torr, ubor Ladick, and P. H. Torr, “Linear Support Vector Machines Learning Lightweight Neural Networks: Pruning and Quantization View project Continual Learning for Scene Understanding Agents View project Locally Linear Support Vector Machines,” 2011. [Online]. Available: https://www.researchgate.net/publication/221345963

[30] G. Guo, H. Wang, D. A. Bell, Y. Bi, D. Bell, and K. Greer, “KNN Model-Based Approach in Classification,” 2004. [Online]. Available: https://www.researchgate.net/publication/2948052

[31] P. Kaviani, B. Vidyapeeth, and M. S. Dhotre, “Short Survey on Naive Bayes Algorithm Study of Scheduling Algorithm to assess the performance of a processess in Operating Systems View project E-Learning System View project Short Survey on Naive Bayes Algorithm,” *International Journal of Advance Engineering and Research Development*, vol. 4, no. 11, 2017, [Online]. Available: https://www.researchgate.net/publication/323946641

[32] H. H. Patel and P. Prajapati, “Study and Analysis of Decision Tree Based Classification Algorithms,” *International Journal of Computer Sciences and Engineering*, vol. 6, no. 10, pp. 74–78, Oct. 2018, doi: 10.26438/ijcse/v6i10.7478.

[33] G. Biau and G. B. Fr, “Analysis of a Random Forests Model,” 2012.

[34] C. Tu, H. Liu, and B. Xu, “AdaBoost typical Algorithm and its application research,” in *MATEC Web of Conferences*, Dec. 2017, vol. 139. doi: 10.1051/matecconf/201713900222.

[35] T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Aug. 2016, vol. 13-17-August-2016, pp. 785–794. doi: 10.1145/2939672.2939785.

APPENDIX

Fake News Detection Using Machine Learning Techniques

Author: Sanjiv Sharma[1], Vidit Upreti[1], Shwetank Pratap Tiwari[1], Lakshya Yadav[1], Umang Rastogi[1], Dr. B. Barani Sundaram[2]

1. **Department of Computer Science & Engineering, KIET Group of Institutions Ghaziabad, India**
2. **Department of Computer science, College of Informatics, Bule hora University, Bule hora ,Ethiopia**

Abstract

The proliferation of fake news has become a major concern in recent years, making it imperative to develop methods for detecting and combatting it. This research paper investigates the effectiveness of machine learning techniques in detecting fake news. In this study three datasets were used to evaluate the performance of various algorithms, including linear support vector machines (SVM), boosting algorithms like Adaboost and XGboost, K-Nearest Neighbors, Naive Bayes, and ensemble learning algorithms. The results show that the Linear SVM and Bagging Classifier algorithms performed best among ensemble learners, with an average accuracy of 96.4% and 97.6%, respectively, while K-Nearest Neighbors performed the worst with an accuracy of 65.8%. Logistic regression also had a high average accuracy of 95% across the three datasets. The study also analyzed the precision, recall, and F1-score of each algorithm on the three datasets. Overall, the findings suggest that machine learning techniques can effectively detect fake news and assist in combating its spread.

Keywords: Fake news, Social media, Authenticity, Artificial Intelligence, Logistic regression, Support vector machine, Naïve Bayes algorithm, Random Forest algorithm,

Introduction

False information spreading among the masses is nothing new and has been around since ancient times, long before the advent of the Internet. Misleading and false information is disseminated for financial gain through the use of fake news. A couple of years ago the news used to disseminate from one person to another by word of mouth, but in the current scenario, the dissemination of information has come a long way since then. Because of the development of the World Wide Web and the rapid proliferation of online social media platforms (such as Facebook, Twitter, and WhatsApp), it is now possible for information to be transmitted in a manner that has never been witnessed before in the annals of human history. As a result of increased connectivity and intelligent automation, now more than a million users get their news and other information primarily from social networking sites and other social platforms. Contrary to other internet applications, news organizations benefited from the extensive usage of social media platforms by updating their subscribers' news in almost real-time. Newspapers, tabloids, and magazines are the prime sources of online news, and blogs, social media feeds, and other digital media formats are the latest news media. With the help of social media, any idea can quickly spread to millions of people all over the world. These social media platforms, in their present phase, are very successful and helpful in providing users with the opportunity to debate, exchange, and discuss themes such as democracy, education, and health. However, some people or organizations also utilize these platforms negatively, frequently to obtain financial advantage, occasionally to influence public opinion, and people's attitudes, or to propagate satire or outrageousness. In the past ten years, false news has spread quickly.

A lot of issues have arisen in politics, sports, health, and science, as a result of the widespread circulation of fake news online. The financial markets are one of these areas where fake news is a problem. A rumor can have terrible results or even stop the market altogether. Someone's ability to make decisions is significantly influenced by the information they take in. There are mounting pieces of evidence that people have behaved unwisely in response to news that afterward turned out to be false. One recent example is the propagation of the new coronavirus, in which false information regarding the virus's origin and behavior is propagated across the Internet, which further leads to chaos and unrest in society.

It would take a tremendous amount of time and resources for people to verify the veracity of every social media post or article. Therefore, more effort should be put into developing automated applications and techniques for determining whether or not news stories shared on social media are authentic. There is an irresistible need for a low-cost, high-performing model that can accurately categorize the proliferation of fake news or articles while requiring minimal manual effort.

This research article is organized as follows: Section 1 gives a brief introduction to the research area and its importance in our society; Section 2 gives an insight into the contemporary research work available in the literature. Further section 3 discusses the proposed framework along with the data sets, data pre-processing, selected machine learning algorithms for the study, and different performance matrices for performance evaluation. Section 4 discusses the comparative analysis of the result achieved. In the last section, 5 presents a conclusion of this research work.

2. Related works

Machine learning is a tool that is often used to spot fake news. Many researchers have tried to come up with different methods and frameworks for figuring out whether the news is real or fake. This section has talked about some of these ways of doing things.

Shahbazi Z et al. [1] proposed a method for fake media detection using natural language processing and blockchain approaches where they collected data from online sources and social media networks which include Twitter, BBC, and Facebook. From this data, they have extracted 5 features that have been used to train blockchain-based reinforcement learning. The proposed solution states that proof of authority and used validation are useful tools for dealing with fake news.

Mridha M et al. [2] reviewed the use of deep learning algorithms in fake news detection using a combination of various NLP techniques and deep learning algorithms on a different dataset which is available for benchmarking. As a result, they have tried to uncover the weaknesses and strengths of these algorithms.

Seddari N et al. [3] created a technique that incorporates linguistic and fact-checking elements into the process of training a model that employs four different machine learning algorithms logistic regression, random forest, additional trees discriminant, and XGBoost. The result shows that Random Forest achieves an accuracy of 94% which outperforms the other algorithms used.

Jiang T et al. [4] proposed a Novel stacking approach for the accurate detection of fake news. They used 5 machine learning and 3 deep learning algorithm and evaluated their performance on two different datasets in terms of recall, accuracy, f1-score, and precision. Then they proposed a stacking approach model and used McNemar’s test to determine if the performance is any different.

Rohera D et al. [5] gave a taxonomy of fake news classification techniques. They trained 4 machine learning models to detect fake news. Later they improved its performance by hyper tuning parameters like drop-out factor, smoothening, and batch size. This new mode, when implemented gave better results in the evaluation phase.

Tariq A et al. [6] proposed adversarial training to regularize fake news classification tasks. This paper evaluates the performance of two transformed modes on publicly available datasets. The results of the evaluation process show that on a long text, the model performs better than the baseline in accuracy, precision, and recall but degrades for short text classification.

Shishah W [7] proposes a joint BERT application to perform automatic fake news detection on Arabic news. The proposed framework is tested on 4 real-world Arabic datasets and the test are compared to other Arabic fake news techniques such as Qarib, and araGPT2. This joint BERT model works better than all the others in terms of recall, precision, and F-1 score.

Umer M et al. [8] gave a fake news stance detection architecture using Deep learning. They use a combination of deep learning algorithms such as CNN and LSTM and neural networks. This approach gave an accuracy of 97.8% which they consider to be better than the previous study.

Ali H et al. [9] analyzed fake news detection models using black box testing where they used 4 deep learning methods which include Multi-layer Perceptron, Convoluted Neural Network, Recurrent Neural Network, and a hybrid CNN-RNN-based model for robustness. In their research, they have tried to give various length inputs and use different loss functions that prove that CNN is the most robust model followed by RNN, but the MLP and hybrid model drop in accuracy after an increase in the input size.

Carter M et al. [10] have tried to uncover the strengths of various machine learning and deep learning methods which are used in fake content detection. Their research provides insight into the fields in which supervised and unsupervised learning methods provide better results. They point out the key loopholes in these methods and ways we can try to improve the performance.

Qureshi K et al. [11] have created a model that used the information regarding the sources and the connectivity of the news propagators to classify the news present on Twitter and used it on the COVID-19-based news dataset. They used a series of algorithms in their evaluation and concluded that CatBoost and RNN were the most effective with an accuracy score of 98%.

Elsaeed E et al. [12] have used voting classifiers for the problem in which they have given a framework to pre-process the data using lemmatization and vectorization algorithms and built a model using their voting classifier on 3 datasets. The proposed model provides improved accuracy over the existing solutions on the ISOT dataset by 0.2%.

Xu K et al. [13] put forth a framework that is built around the domain of the news to be classified. They have studied features including domain popularity, domain ranking, and the probability of the news disappearing for classification. Other key components in their model were TF-IDF vectorizer and LDA topic modeling as a result they concluded that the fake and real news only had subtle differences in their data and have given a prospect to use doc2vec for vectorization.

Ravish et al. [14] have created a model that incorporates the use of Multi-layered Principal Component Analysis that determines the features to be used to train the model based on Multi-Support Vector Machines. They have used this model to classify news in 10 different datasets with varying features. They conclude based on their evaluation that the accuracy is only low in the case of the datasets having less number of features.

Sudhakar M et al. [15] suggested an automatic detection technique that makes use of algorithms for machine learning in order to identify false news stories in Indian media information and sources. The mechanism that is now being presented helps control bogus information. Classification models built using machine learning, such as the Naive Bayes, Support Vector Machine, and Logistic Regression algorithms. The LSTM Convolutional Neural Network model that was proposed has delivered much-improved accuracy.

Lorent S and Itoo A [16]evaluated the effectiveness of the Attention Mechanism for detecting fake news on two datasets, one named “Fake News Corpus” which contains articles from online sources, and the second “Liar-Liar Corpus” which is described as a collection of short news articles from various sources such as political interviews, TV ads. The results of the Attention Mechanism and results on both datasets are compared against Deep learning models and methods of machine learning. Word2vec Embedding is used along with the attention mechanism to enhance the performance of the approach.

Preda A [17] describes various techniques used in the classification of text and examines the performance of different machine learning and neural networks on a dataset containing articles written in Romanian. BERT used transformers alongside other machine learning techniques to categorize articles into real news, fake news, and other types. BERT used after parameter tuning performed much better than other machine learning techniques.

Dementieva D [18] proposed a new feature Multiverse, which uses cross-lingual evidence for the detection of unreliable news and enhances existing techniques. Firstly, a manual study was conducted on twenty news datasets to check whether a real-life user can detect fake news using cross-lingual evidence. Then, using two strategies, content similarity estimation was experimented on. First, based on cosine distance between news text embeddings. Second, based on Natural language inference scores when scraped news was used as hypothesis h, and original news was used as premise p. Lastly integration of cross-lingual evidence features into an automated pipeline for fake news detection. Cross-lingual features in combination with linguistic features bested both deep learning and statistical fake news classification systems.

Li D [19] proposed an autoencoder based on an unsupervised fake news detection method (UFNDA). This research studies various news kinds in social networks and incorporates text content, images, user information, and propagation to increase the effectiveness of fake news identification. UFNDA is used to assess fake news by utilizing it as anomalous data. Compared to unsupervised unreliable news detection techniques, UFNDA produces better outcomes. To perform in-depth categorization of news, UFNDA could yet be developed.

3. Material and methods

In this section, the proposed framework is explained in detail. The framework as reflected in *Figure-1* depicts, that the data is collected from various credible sources which include Kaggle [20] and the University of California[21]. The data is then preprocessed using the inbuilt library in Python NLTK (Natural Language Toolkit)[22] then feature extraction is performed on text using another inbuilt library of Python called TF-IDF Vectorizer[23]. Following this 9 machine learning algorithms have been used to train a model to classify if a news article is either reliable or non-reliable.

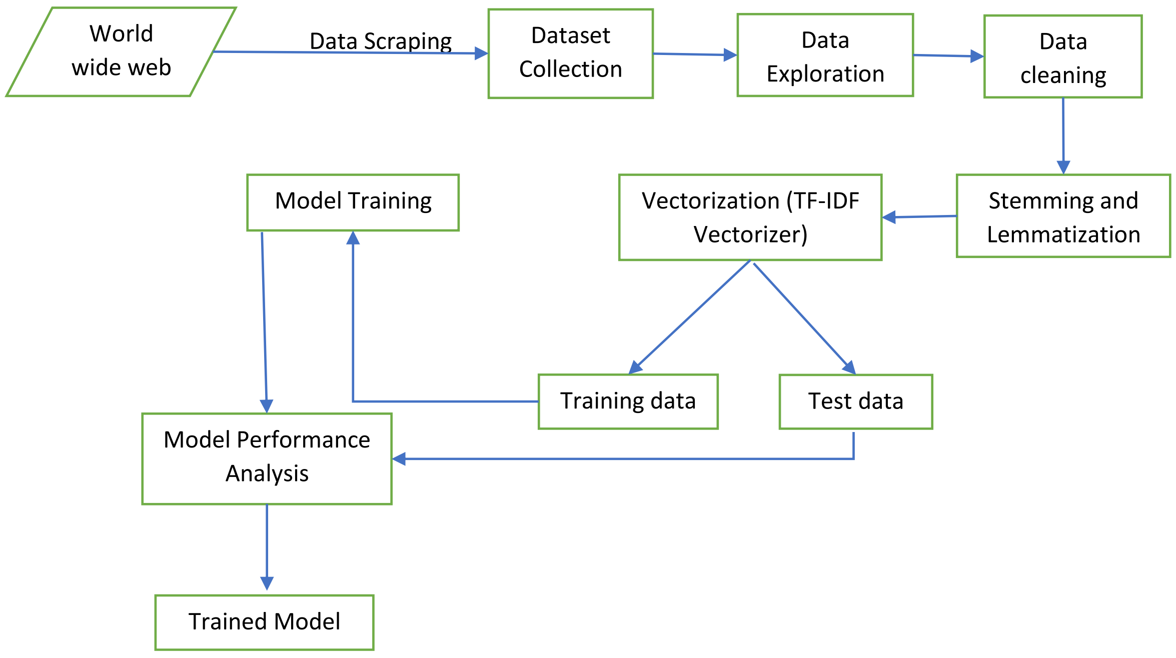


Figure 1 Proposed Framework

3.1. Datasets

This work includes 3 datasets that are used for training the model and finally testing it. The datasets consist of news from various genres which include news related to politics, entertainment, technology, and sports. The datasets are also having a mix of both true and fake news that is well labeled.

The first dataset is a dataset taken from Kaggle[24]. The data is consisting of 5 columns which include fields id, author, title, text, and a label. The dataset is divided into train and test data in which the training dataset contains 20387 rows of data, on the other hand, the testing data contains 5127 rows of data.

The second dataset (containing 3,352 articles) is accessible on Kaggle[25]. The authentic articles are sourced from The New York Times, CNN, Reuters, and other reputable online sources, whereas false news is sourced from news websites that have been known to publish unreliable information. This dataset covers a variety of topics including politics, entertainment, and sports.

The third dataset "ISOT Fake News Dataset"[21], is a dataset that is widely used as a benchmark for evaluating fake news classification problems. The dataset is a collection of articles that are present on a news website called reuters.com. On the other hand, there are fake articles that are a compilation of news from sources that Politifact.com has identified as unreliable. The articles are a mix of various genres but most of them focus on political news. A total of 44,898 articles are present of which 21,417 are true while the rest are false.

3.2. Data pre-processing

The data that is present in the dataset is in form of published articles hence it is still having many inconsistencies and comprises null values, non-ASCII characters, meaningless spaces, hashtags, and other symbols that are not useful in the evaluation process. The pre-processing includes 7 steps that ensure the removal of all inconsistencies and unwanted characters. The first step was removing all the null-valued rows, which are the rows that have no values entered in them. Null-valued rows provide a false output when any function is applied. The second step was to add an empty string to the columns where a value is missing to remove inconsistencies. The third step was to convert the data into lowercase so that it can be easily operated on and to maintain a standard for the data. The fourth step was removing all non-ASCII characters found in the data. The non-ASCII characters have no meaning in the evaluation since they cannot be handled through a vectorizer. The fifth step includes the removal of all mentions (names following @) and all the URLs in the data. The next step was to filter out hashtags, extra numbers, and spaces that have no significant meaning. All the above-mentioned pre-processing is done using a library called NLTK (Natural Language Toolkit)[26]. It has applications in semantic reasoning, categorization, tokenization, stemming, tagging, and parsing, among others.. The final step uses port stemmer to convert all the words and reduce them into their stem words or root words.

Following stemming, the data at hand is a group of words that are in their root form, but a machine learning algorithm can only work on numerical data that is where the vectorizer comes in. The pre-processed data is then given to the vectorizer known as TF-IDF Vectorizer[27]. This commonly used technique transforms text into equivalent numerical representations to be used in machine learning algorithms. Machine-learning models are then constructed using the input features. Each dataset is split into a training set and a testing set, each with an 80/20 split.

The problem at hand puts forth a scenario where two classes are there, and machine learning is used to classify the news articles into one of these classes. The first class is reliable and the other is unreliable or fake news. In these types of scenarios, we use an algorithm that can perform the task of binary classification which include Logistic regression, K-nearest neighbors, Linear SVM, Random Forest, Bagging classifier, Decision trees, AdaBoost, XGBoost, and Naive Bayes Classifier.

3.3. Algorithms

3.3.1. Logistic Regression.

Logistic Regression[28] is one of the most frequently applied classification methods. It is a supervised classification algorithm that takes a set of inputs as features and maps the data into classes that are our required outputs. The algorithm although called regression is used in classification because it uses a sigmoidal function to model the data and divide it into classes depending on a threshold value.

$$sig(x) = \frac{1}{1+e^{-x}} \tag{1}$$

Using the probability value, we aim at minimizing the cost function that results in the optimal probability.

$$C(sig(x), y) = log\big(sig(x)\big), \quad y = 1 \tag{2}$$

$$C(sig(x), y) = -\log\big(1 - sig(x)\big), \quad y = 0$$

Here C (x, y) states the cost when the sigmoidal function returns the value x.

The tuning we require is to determine the outcome based on the requirement. We can have either a high Precision or a high Recall result. The difference between these results determines the number of false positives and false negatives we will get in our results.

3.3.2. Linear Support Vector Machine Algorithm.

The linear support vector machine (LSVM) is an example of a supervised learning method that is useful for solving classification issues [29]. It does so by plotting the data points into the n-dimensional graph for the supplied set of n characteristics. Support vectors are known to be the most extreme points on a graph, which is where the term "support vector machine" comes from. The data is classified based on the hyperplane that runs through the middle of the two categories. When working with data that can be linearly divided into two classes, linear support vector machines, or linear SVMs, perform exceptionally well. This program makes use of the RBF kernel because it is the one that performs the best when dealing with huge applications such as a corpus of news items. Given these two samples, x and x', the Radial Basis function can be written as:

$$f(x, x') = e^{\frac{\left\| x - x' \right\|^2}{2a^2}} \qquad (3)$$

3.3.3. K-Nearest Neighbour(KNN) Algorithm

KNN algorithm[30] is a supervised machine learning algorithm that is used for solving a classification problem. KNN is an instance-based learning algorithm that uses the complete data for training the model. It classifies each new case into a category based on the knowledge gained from previous cases. This algorithm maintains the data throughout the training phase and then classifies the data based on the value of K at the time of classification. The number of neighbours we consider when determining the most similar class is K. The value of K can be determined by trying out different values.

To find the k most similar instances we use Euclidian's distance formula (5). The Euclidian distance is the square root of the sum of squares of the difference between the testing data and training data.

$$Euclidean\ distance\ = \sqrt{\sum_{i=1}^{i=n} (x_i - y_i)^2} \qquad (4)$$

3.3.4. Naïve Bayes Classifier Algorithm

Naïve Bayes Classifier Algorithm[31] is a supervised learning algorithm and is used for solving classification problems. This Classifier uses the probability of data to predict its class. The probability is calculated using Bayes theorem (6). In this algorithm, all the features of an instance contribute independently to the determination of its class and that is why it is named Naïve. This algorithm is useful for text classification over a large dataset.

Bayes theorem is used to find the posterior probability p(a/x), using p(a),p(x), and p(x/a). Given a predictor x, the posterior probability of the class (a, target) is denoted by the symbol P(a|x) (x, attributes). The prior probability of class is denoted by the symbol P(a). The likelihood, denoted by the symbol P(x|a), is the probability of a predictor being given a class. The prior probability of the predictor is denoted by the symbol P(x).

$$P\left(\frac{a}{x}\right) = P\left(\frac{x}{a}\right) . P(a)/P(x) \qquad (5)$$

3.3.5. Decision Tree Classification Algorithm

Classification issues can be tackled with the help of the decision tree classification method [32], a supervised learning approach. After inferring the decision rules from the training data, the resulting decision tree may accurately predict the target class. Each branch of a decision tree is either a decision node or a leaf node. The decision nodes are where the data is divided, and the leaves are the decisions or the results. It offers a top-down strategy in terms of its organizational structure.

The tree divides the dataset into a small set of records by applying the inferred decision rules. These small sets of records are the different classes that are predicted by the tree.

3.3.6. Random Forest Algorithm.

Random forest[33] is a supervised machine learning algorithm that is used to solve classification problems as well as regression problems. It is an ensemble of a decision tree that gives better performance for a complex problem compared to a decision tree. Instead of relying on one decision tree for predictions, Random Forest uses predictions from many decision trees and the result depends on the majority, among the predictions of all the trees.

The accuracy of this algorithm depends on the number of decision trees used in the algorithm. Increasing the decision trees in the forest also prevents the problem of overfitting. The random forest algorithm works because it uses many relatively uncorrelated trees operating together. If similar trees are used in the forest, then the result will be no different from the decision tree algorithm. We can get uncorrelated trees can be obtained by using feature randomness and bootstrap aggregation.

Bootstrap Aggregation

Decision trees are extremely sensitive to the data that they are trained on, and even the smallest of modifications to the data set that they are trained on can result in dramatically different tree architectures. The random forest algorithm makes use of this fact by enabling each individual tree to randomly pick data from the dataset while maintaining replacement, which ultimately results in a variety of trees. Bagging, or bootstrapping, is another name for this process.

Feature Randomness

When it comes time to split a node in a typical decision tree, we take into account all of the alternative features and select the one that creates the greatest amount of differentiation between the observations included in the left node and those contained in the right node. In contrast, only a random subset of characteristics is available for selection by each individual tree in a random forest. Because of this, there is an even greater degree of variety among the trees in the model, which ultimately leads to a weaker correlation between the trees and a greater degree of diversification.

3.3.7. Boosting Classifier

In this method of ensemble learning, we initially have stumps (1-level trees) these are the basic tree that we initially use in the training process. The training data is then classified, and the results are checked the data that has been misclassified is then given a higher weight and we focus on reducing these misclassified points, then we further use these baselevel trees to create a classifier that can handle the classification problem more efficiently. Some examples of boosting include AdaBoost[34] and XGBoost[35]. AdaBoost (Adaptive boosting) functions by changing the sample distribution by updating the weights during each iteration of its execution. The 3 main steps in this are using stump (1-level tree) as weak learners then influencing the next stump based on modification on the first stump and finally a weighted vote to select the stump that will be used in the final class. XGBoost (eXtreme Gradient Boosting) is another widely used boosting algorithm that is based on Gradient boosting and uses more regularized model formalization to control overfitting which yields better results. Sequential patterns are used to construct decision trees.

3.4. Performance Metrics

The performances of the algorithms (in section 2.3) for fake news detection have been compared using several assessment metrics. Evaluation metrics are widely utilized to assess the performance of supervised machine learning algorithms as they enable us to compare the efficiency of all the algorithms. In the proposed model, As can be seen in Table 1, an evaluation of the effectiveness of the false news detection model was carried out with the use of a confusion matrix. Samples are determined to be either real or fake using this matrix.TP, FP, TN, and FN concepts are explained as follows:

• True Positive (TP): The predicted fake news is actually fake.

• False Positive (FP): The predicted fake news is actually real.

• True Negative (TN): The predicted real news is actually real.

• False Negative (FN): The predicted real news is actually fake.

Table 1 Confusion Matrix

	Actual Positive Class	**Actual Negative Class**
Predicted Positive Class	True Positive	False Positive
Predicted Negative Class	False Negative	True Negative

Accuracy: Accuracy measures how often a prediction turns out to be true or false for a set of observations. In order to determine how well a model performs, Equation (6) can be used to measure accuracy.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \qquad (6)$$

Recall: The recall measure is the proportion of correct classifications that correspond to the actual class. Number of projected true articles divided by the total number of true articles in the suggested model. Equation (7) is used to find out the value of the recall:

$$\textit{Recall} = \frac{TP}{TP+FN} \tag{7}$$

Precision: The precision score is the ratio of the number of true positives to the total number of events that were correctly forecasted as true. In the model that has been proposed, it indicates the percentage of articles that have been determined to be accurate out of the total number of articles that have been accurately predicted. Equation 8 can be used to find out the recall value.

$$\textit{Precision} = \frac{TP}{TP+FP} \tag{8}$$

F1-Score: The F1 score is a representation of the balance that was made between precision and recall. It performs the calculation necessary to determine the harmonic mean between the two. This method takes into consideration both the false positive and the false negative observations. The equation 9 below can be used to get an individual's F1 score.:

$$\textit{F1-score} = \frac{TP+TN}{TP+TN+FP+FN} \tag{9}$$

Accuracy refers to the rate at which news is correctly expected for all samples when discussing the difficulty of identifying false news. In relation to the overall number of fake news items, the recall value shows what percentage of false news stories may have been successfully predicted. Based on the overall amount of fake news that can be foreseen, the Precision measure assesses the amount of fake news that can be reliably predicted. The harmonic average of the precision value and recall value is used to calculate the F-measure, a method for detecting instances of fake news.

4. Results

The accuracy of each method on the three datasets under consideration is summarised in Table 2. The Linear SVM attained a maximum accuracy of 96.5% on DS1 (Kaggle Fake News Dataset 1). The accuracy of boosting algorithms like Adaboost and XGboost was 95%. The least accurate models were K-Nearest Neighbors and Naive Bayes, with scores of 45% and 71%, respectively. On DS1, ensemble learning algorithms achieve an accuracy of 93.5% on average, compared to an accuracy of 66.7% for individual learners. Individual and Ensemble learners differ significantly, with a 32.2% absolute difference between the two groups. The algorithms with the best results on Dataset 2 are Linear SVM and Bagging Classifier, which achieve an accuracy of 97%. Although individual learners reported an average accuracy of 94.3%, the average accuracy of the ensemble learners was 95.6%, making the difference between the two non-significantly different by 1.3%. On Dataset 3, logistic regression. and bagging algorithms both achieved an accuracy of 98%, while Linear SVM reported an accuracy of 99%. K-Nearest Neighbors reported the lowest accuracy, at 57.5%. The average accuracy for individual and ensemble learners was 88.7% and 97.1%, respectively.

Table 2: Overall accuracy for each dataset

Accuracy Score	Dataset 1	Dataset 2	Dataset 3
K- Nearest Neighbors	0.45	0.905	0.575
Linear SVM	0.965	0.973	0.99
Logistic regression	0.939	0.961	0.979
Naïve-Bayes	0.711	0.94	0.947
Decision Tree	0.913	0.936	0.946
Random Forest	0.885	0.943	0.976
Bagging	0.944	0.97	0.98
XG boost	0.957	0.945	0.963
Ada boost	0.956	0.966	0.965

The average accuracy of each algorithm across the three datasets is shown in Figure 2. Overall, the bagging classifier and linear SVM algorithms perform the best among ensemble learners (accuracy of 96.4% and 97.6%, respectively), while K-Nearest Neighbors perform the worst (accuracy of 65.8%). The accuracy of individual learners is 87.8%, whereas that of ensemble learners is 95.37%. Only dataset 2 (90.5%) had a high accuracy score with K-Nearest Neighbors, while datasets 1 and 3 received poor results. On datasets 2 and 3, naive Bayes also performed well (accuracy of 94%), but not on dataset 1 (accuracy of 71%). Recall, precision, and F1-score are additional metrics we use to evaluate the performance of learning models because the accuracy score alone is not a trustworthy measure of a model's performance.

Tables 3-5 review each algorithm's precision, recall, and f1-score for each of the three datasets. In terms of average precision (Table 3), Linear SVM performed best among individual learners, while the Bagging classifier performed best among ensemble learners. On all three datasets, the linear SVM and bagging classifiers' average precision is 97%. The average precision of random forest (RF) increased to 96.9% on the three datasets after deleting the dataset with the lowest score, DS2. Random forest (RF) had previously attained a precision of 95%. The K-Nearest Neighbors algorithm has the lowest precision score.

According to the recall performance metric, the Boosting Algorithm (Adaboost) outperforms the Ensemble Learners with a recall score of 0.96, while the Linear SVM performs best among Individual Learners with a recall score of 0.971. Following closely behind are the K-Nearest Neighbors technique and the boosting classifier (XGBoost), both of which had recall values of 0.953 and 0.963 respectively. On the F1- score, the algorithms performed similarly to how they did for precision. The best F1-score of any technique was attained by the linear SVM, which had an F1-score of 0.97. The bagging classifier and Adaboost learning algorithm came in second with F1-scores of 0.959.

Figure 3 shows the average performance of learning algorithms utilizing precision, recall, and f1-score across the three datasets. It is evident that the effectiveness of learning algorithms utilizing different performance criteria does not differ significantly with K-Nearest Neighbors and Naïve Bayes being exceptions.

On all performance criteria, the ensemble learner XGboost outperforms other learning models. The main reason for the increased performance of the software is XGBoost's operating philosophy, which efficiently discovers defects and minimizes them in each cycle. Using multiple classifications and regression trees (CART), which include numerous weak learners, to give misclassified data points higher weights, is the essential notion behind how XGBoost functions. For the model to reliably identify the erroneously classified points on each successive iteration, overfitting is reduced using regularisation parameters.

Despite being a relatively straightforward model, logistic regression has an average accuracy of nearly 95% across the three datasets. The high average accuracy can be attributed to a few factors. First, a thorough grid search with numerous hyperparameters is used to optimize the logistic regression model. Second, the similar writing styles of some authors in some datasets (like DS2) help explain the logistic regression model's 96% accuracy.

Table 3: Overall precision for each dataset

Precision	**Dataset 1**	**Dataset 2**	**Dataset 3**
K-Nearest Neighbors	0.44	0.902	0.55
Linear SVM	0.969	0.97	0.989
Logistic regression	0.96	0.95	0.978
Naïve-Bayes	0.998	0.903	0.933
Decision Tree	0.907	0.954	0.939
Random Forest	0.959	0.916	0.979
Bagging	0.967	0.97	0.983
XG Boost	0.957	0.962	0.964
Ada boost	0.953	0.929	0.963

Table 4: Overall Recall for each dataset

Recall	Dataset 1	Dataset 2	Dataset 3
K-Nearest Neighbors	0.998	0.893	0.998
Linear SVM	0.95	0.973	0.992
Logistic regression	0.89	0.967	0.981
Naïve-Bayes	0.334	0.975	0.969
Decision Tree	0.891	0.906	0.959
Random Forest	0.768	0.967	0.976
Bagging	0.902	0.965	0.979
XG Boost	0.943	0.954	0.964
AdaBoost	0.945	0.965	0.97

Table 5: Overall F1-Score for each dataset

F1-Score	Dataset 1	Dataset 2	Dataset 3
K-Nearest Neighbors	0.61	0.897	0.71
Linear SVM	0.959	0.971	0.99
Logistic regression	0.928	0.958	0.98
Naïve-Bayes	0.5	0.938	0.951
Decision Tree	0.89	0.93	0.949
Random Forest	0.85	0.941	0.977
Bagging	0.93	0.967	0.981
XG Boost	0.95	0.941	0.964
Ada boost	0.949	0.963	0.966

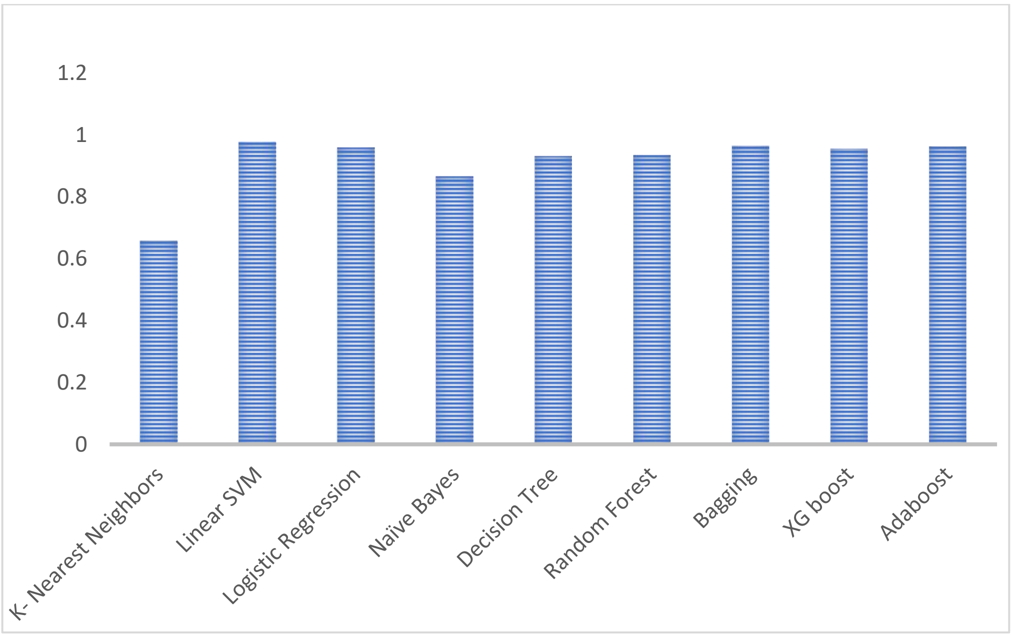


Figure 2: Average Accuracy Score over all datasets

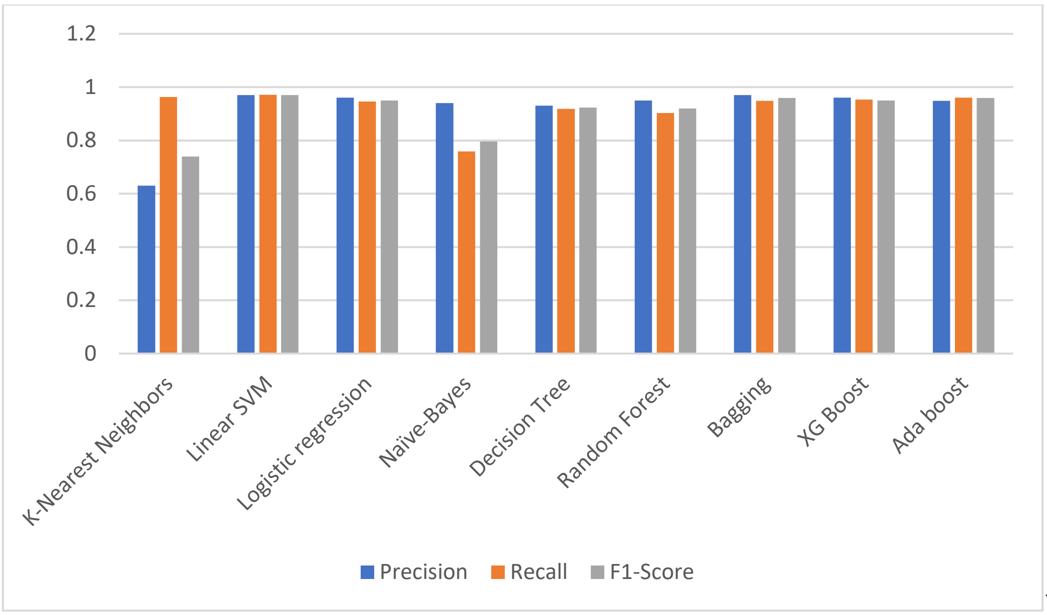


Figure 3: Overall Precision, Recall, and F1-Score over all datasets

5. Conclusion

Fake news identification and detection, when performed manually, require a lot of detailed knowledge of the field and proficiency to recognize abnormalities in the news article. This work tries to consider the problem of identifying fake news using machine learning algorithms. We used data from Kaggle and the ISOT Research Lab for our work, which contained news articles from various fields, to deal with the majority of the news and not any specific news. The research's main goal is to find textual characteristics that distinguish genuine news from fraudulent news. The different features required for classification were identified using the TF-IDF vectorizer, and the identified feature set was used as input to the machine learning models. Comparatively speaking, some models achieved higher accuracy than others. Results were compared using different performance metrics such as precision, recall, and f1-score for each algorithm. On overall performance, ensemble learning algorithms achieved a much better score than individual learning algorithms.

Fake news identification currently has many problems that require attention from research teams. Machine learning can be used to identify the causes of the spread of unreliable news. Future work can be done to identify fake news in live news articles and stop the spread of fake news.

6. References

[1] Z. Shahbazi and Y. C. Byun, “Fake Media Detection Based on Natural Language Processing and Blockchain Approaches,” *IEEE Access*, vol. 9, pp. 128442–128453, 2021, doi: 10.1109/ACCESS.2021.3112607.

[2] M. F. Mridha, A. J. Keya, M. A. Hamid, M. M. Monowar, and M. S. Rahman, “A Comprehensive Review on Fake News Detection with Deep Learning,” *IEEE Access*, vol. 9, pp. 156151–156170, 2021, doi: 10.1109/ACCESS.2021.3129329.

[3] N. Seddari, A. Derhab, M. Belaoued, W. Halboob, J. Al-Muhtadi, and A. Bouras, “A Hybrid Linguistic and Knowledge-Based Analysis Approach for Fake News Detection on Social Media,” *IEEE Access*, vol. 10, pp. 62097–62109, 2022, doi: 10.1109/ACCESS.2022.3181184.

[4] T. Jiang, J. P. Li, A. U. Haq, A. Saboor, and A. Ali, “A Novel Stacking Approach for Accurate Detection of Fake News,” *IEEE Access*, vol. 9, pp. 22626–22639, 2021, doi: 10.1109/ACCESS.2021.3056079.

[5] D. Rohera *et al.*, “A Taxonomy of Fake News Classification Techniques: Survey and Implementation Aspects,” *IEEE Access*, vol. 10, pp. 30367–30394, 2022, doi: 10.1109/ACCESS.2022.3159651.

[6] A. Tariq, A. Mehmood, M. Elhadef, and M. U. G. Khan, “Adversarial Training for Fake News Classification,” *IEEE Access*, vol. 10, pp. 82706–82715, 2022, doi: 10.1109/ACCESS.2022.3195030.

[7] W. Shishah, “JointBert for Detecting Arabic Fake News,” *IEEE Access*, vol. 10, pp. 71951–71960, 2022, doi: 10.1109/ACCESS.2022.3185083.

[8] M. Umer, Z. Imtiaz, S. Ullah, A. Mehmood, G. S. Choi, and B. W. On, “Fake news stance detection using deep learning architecture (CNN-LSTM),” *IEEE Access*, vol. 8, pp. 156695–156706, 2020, doi: 10.1109/ACCESS.2020.3019735.

[9] H. Ali *et al.*, “Analyzing the Robustness of Fake-news Detectors under Black-box Adversarial Attacks,” *IEEE Access*, 2021, doi: 10.1109/ACCESS.2021.3085875.

[10] M. Carter, M. Tsikerdekis, and S. Zeadally, “Approaches for Fake Content Detection: Strengths and Weaknesses to Adversarial Attacks,” *IEEE Internet Comput*, vol. 25, no. 2, pp. 73–83, Mar. 2021, doi: 10.1109/MIC.2020.3032323.

[11] K. A. Qureshi, R. A. S. Malick, M. Sabih, and H. Cherifi, “Complex Network and Source Inspired COVID-19 Fake News Classification on Twitter,” *IEEE Access*, vol. 9, pp. 139636–139656, 2021, doi: 10.1109/ACCESS.2021.3119404.

[12] E. Elsaeed, O. Ouda, M. M. Elmogy, A. Atwan, and E. El-Daydamony, “Detecting Fake News in Social Media Using Voting Classifier,” *IEEE Access*, vol. 9, pp. 161909–161925, 2021, doi: 10.1109/ACCESS.2021.3132022.

[13] K. Xu, F. Wang, H. Wang, and B. Yang, “Detecting Fake News Over Online Social Media via Domain Reputations and Content Understanding,” 2020.

[14] Ravish, R. Katarya, D. Dahiya, and S. Checker, “Fake News Detection System Using Featured-Based Optimized MSVM Classification,” *IEEE Access*, vol. 10, pp. 113184–113199, Nov. 2022, doi: 10.1109/access.2022.3216892.

[15] M. Sudhakar and K. P. Kaliyamurthie, “A Machine Learning Framework for Automatic Fake News Detection in Indian News,” 2022, doi: 10.21203/rs.3.rs-2268597/v1.

[16] S. Lorent and A. Itoo, “Fake News Detection Using Machine Learning,” 2018.

[17] A. Preda, S. Ruseti, S.-M. Terian, and M. Dascălu, “Romanian Fake News Identification using Language Models,” in *RoCHI - International Conference on Human-Computer Interaction*, 2022, pp. 73–79. doi: 10.37789/rochi.2022.1.1.13.

[18] D. Dementieva, M. Kuimov, and A. Panchenko, “Multiverse: Multilingual Evidence for Fake News Detection,” Nov. 2022, [Online]. Available: http://arxiv.org/abs/2211.14279

[19] D. Li, H. Guo, Z. Wang, and Z. Zheng, “Unsupervised Fake News Detection Based on Autoencoder,” *IEEE Access*, vol. 9, pp. 29356–29365, 2021, doi: 10.1109/ACCESS.2021.3058809.

[20] “Kaggle: Your Home for Data Science.” https://www.kaggle.com/ (accessed Nov. 23, 2022).

[21] “Fake News Detection Datasets - University of Victoria.” https://www.uvic.ca/ecs/ece/isot/datasets/fake-news/index.php (accessed Nov. 23, 2022).

[22] “NLTK :: Natural Language Toolkit.” https://www.nltk.org/ (accessed Nov. 23, 2022).

[23] “sklearn.feature_extraction.text.TfidfVectorizer — scikit-learn 1.1.3 documentation.” https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html (accessed Nov. 23, 2022).

[24] “Fake News | Kaggle.” https://www.kaggle.com/competitions/fake-news/overview (accessed Nov. 23, 2022).

[25] “Fake News detection | Kaggle.” https://www.kaggle.com/datasets/jruvika/fake-news-detection (accessed Nov. 24, 2022).

[26] E. Loper and S. Bird, “NLTK,” 2002, pp. 63–70. doi: 10.3115/1118108.1118117.

[27] S. Qaiser and R. Ali, “Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents,” *Int J Comput Appl*, vol. 181, no. 1, pp. 25–29, Jul. 2018, doi: 10.5120/ijca2018917395.

[28] C. Y. J. Peng, K. L. Lee, and G. M. Ingersoll, “An introduction to logistic regression analysis and reporting,” *Journal of Educational Research*, vol. 96, no. 1, pp. 3–14, 2002, doi: 10.1080/00220670209598786.

[29] P. Hilaire Torr, ubor Ladick, and P. H. Torr, “Linear Support Vector Machines Learning Lightweight Neural Networks: Pruning and Quantization View project Continual Learning for Scene Understanding Agents View project Locally Linear Support Vector Machines,” 2011. [Online]. Available: https://www.researchgate.net/publication/221345963

[30] G. Guo, H. Wang, D. A. Bell, Y. Bi, D. Bell, and K. Greer, “KNN Model-Based Approach in Classification,” 2004. [Online]. Available: https://www.researchgate.net/publication/2948052

[31] P. Kaviani, B. Vidyapeeth, and M. S. Dhotre, “Short Survey on Naive Bayes Algorithm Study of Scheduling Algorithm to assess the performance of a processess in Operating Systems View project E-Learning System View project Short Survey on Naive Bayes Algorithm,” *International Journal of Advance Engineering and Research Development*, vol. 4, no. 11, 2017, [Online]. Available: https://www.researchgate.net/publication/323946641

[32] H. H. Patel and P. Prajapati, “Study and Analysis of Decision Tree Based Classification Algorithms,” *International Journal of Computer Sciences and Engineering*, vol. 6, no. 10, pp. 74–78, Oct. 2018, doi: 10.26438/ijcse/v6i10.7478.

[33] G. Biau and G. B. Fr, “Analysis of a Random Forests Model,” 2012.

[34] C. Tu, H. Liu, and B. Xu, “AdaBoost typical Algorithm and its application research,” in *MATEC Web of Conferences*, Dec. 2017, vol. 139. doi: 10.1051/matecconf/201713900222.

[35] T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Aug. 2016, vol. 13-17-August-2016, pp. 785–794. doi: 10.1145/2939672.2939785.