TABLE OF CONTENT

ABSTRACT:

In the modern era where the internet is ubiquitous, everyone relies on various online resources for news. Together with the increase in the use of social media platforms like WhatsApp, Facebook, Twitter, etc. and Internet. Rumours spread rapidly among millions of users within a short period. The rise of fake news on the internet has raised serious questions about the trustworthiness of information in the public sphere. Fake news detection is one of the key areas of research that aims to combat this problem. A lot of research is already focused on detecting it. In this paper we have tried to aggregate news and subsequently use Statistical machine learning algorithms and make a model to determine whether the news is real or fake. Here we have used various statistical machine learning techniques like Logistics, Support Vector Machine, Random Forest, Decision Tree etc. to create a model that have the potential to detect the fake news by using tools like python scikit-learn, NLP for textual analysis. In this process features extraction and vectorization are done by using python scikit-learn library to perform tokenization and feature extraction of text data, because this library contains useful tool like Tiff Vectorizer. Then, we will perform feature selection methods, to experiment and choose the best fit features to obtain the highest precision and accuracy, according to confusion matrix result. This project's main goal is to give users a tool that can recognize false information and stop it from spreading.

Keywords: Internet, Social Media, Pre-Processing, Statistical Machine Learning, Fake News Detection.

1 INTRODUCTION:

World is converting rapidly. The fast improve of generations, particularly in communications and technology has turn out to be an indicator of the contemporary-day era. No doubt we've got some blessings of the virtual global however it additionally has its risks as well. There are different problems on this virtual global. One of them is fake news. These days fake news is developing different problems from sarcastic articles to a fabricated information and plan authorities' propaganda in a few outlets. It is not difficult for someone to create a fake news. A fake news is created to make someone or an organization less popular. There are different online platforms where the person can spread the fake news. This includes the Facebook, Twitter, WhatsApp etc. The spread of fake news is not a new phenomenon, because even before the Internet era, newspapers often manipulated information to serve their own purposes [9]. As more and more of our lives is spent communicating online through social media platforms, more people tend to point out and consume news from social media rather than traditional news organizations [8]. The explanations for the change in consumption behaviors lie in the nature of these social media platforms:(i) it is often more timely and fewer expensive to consume news on social media compared to traditional media such as newspapers or television; and (ii) social media makes it easy to share, discuss and discuss news with friends or other readers. For example, 62 percent of American adults get their news on social media in 2016, while in 2012 only 49 percent reported seeing news on social media [8]. Digital News Report 2022 from Reuters Institute for the Study of Journalism say 63% Indian respondents used social media to access news in the week preceding the survey period. The figure was 59% for TV, and 49% for print media. As many as 53% of the respondents used YouTube to access news, while 51% used WhatsApp, the report says [10]. It has also been observed that social media is now better than television as it is the main source of news. Despite of all the social media platforms WhatsApp's reach and features make it a top choice to spread fake news.

The app—which allows a user to send media (text, pictures, audio-video) via smartphones—has become a strong alternative of the traditional short message service (SMS), as it is a free application and allows more features. A user just needs an internet connection and a smartphone to communicate. Also, with internet and smartphone, the users are just not consumers of media; they can generate, modify, share and discuss content. There have few examples of spreading the fake news in WhatsApp ,1. In 2012 some videos, pictures and writings went viral on social media about the destruction of the world. 2. Share this image with 5 friends and you'll receive good news within 5 days! Despite the advantages that social media offers, the standard of stories on social media is less than traditional news organizations. Since online news is inexpensive and much faster and easier to spread through social networks, a large amount of fake news, i.e. news articles containing intentionally false information, are produced online for a spread of purposes, for financial and political gain. False information can spread widely and negatively affect people's lives as well as society. First, false information can undermine credibility equilibrium of the news ecosystem, for example, it is clear that the most widely shared fake news on Facebook was even more widespread than the most widely acknowledged legitimate mainstream news [12] Second, fake news changes people's perceptions of and reactions to actual news. For example, some fake news is intentionally designed to trigger people's distrust and make them confused; impeding their abilities to differentiate what's true from what's not. False information can be harmful to the general public's health. For example, false information about the effectiveness and safety of vaccinations can result in lower vaccination rates and greater spread of infectious illnesses [11]. Similar to this, people may be misled by inaccurate information regarding the causes and treatments of diseases, which could lead them to choose dangerous or ineffective treatments. The best example for fake news is that the pandemic situation occurring within the entire world. There are variant of news articles till presently that are falsified and used merely to create confusion and disturbance inside the minds of individual and to misguide their minds to believe that false news. However, can anyone perceive if it's fake or real? An example of fake news related to COVID-19 in India could be a viral message claiming that drinking a specific herbal concoction or tea can cure or prevent the virus. This message may falsely attribute healing properties to certain herbs or ingredients without any scientific evidence to support such claims. Promotions of unproven treatments or preventive measures for COVID-19, such as consuming cow urine, garlic, or certain herbs, without any scientific evidence to support their effectiveness [13]. Fake news has been shown to have a significant impact on public opinion and can cause negative consequences such as social unrest and political instability. The spread of fake news can have serious consequences, including social and political polarization, a decline in trust in institutions, and in some cases even physical harm. Stopping the spread of fake news requires a multi-faceted approach involving various stakeholders, including individuals, social media, traditional media and decision makers. Here are several strategies to prevent the spread of misinformation.

A. Media education: Promote media literacy in schools and communities so people can acquire the critical thinking skills needed to identify and evaluate reliable sources of information. Teaching techniques to spot misinformation, check claims, and understand bias can empower people to make informed decisions.

B. Fact-checking initiatives: Support and promote fact-checking organizations and initiatives that systematically verify the accuracy of news and debunk false claims. Encourage people to consult reliable fact-checking sources before sharing information online.

C. Transparency and Accountability: Promote transparency and accountability of media and social media platforms by promoting clear sourcing, disclosure of conflicts of interest, and adherence to ethical journalism standards. Hold platforms accountable for implementing policies to prevent the spread of false information and misinformation.

D. Notification and Reporting Mechanisms: Implement robust notification and reporting mechanisms on social media platforms to allow users to report suspicious or misleading content. Acting quickly for platforms to investigate and remove false information can help reduce its spread.

E. Collaboration: Promote collaboration between governments, civil society organizations, technology companies and academia to develop comprehensive solutions to combat disinformation. Combining resources, expertise and knowledge can improve the effectiveness of measures.

F. Algorithms: the development of effective algorithms for detecting fake news has become a critical need. One approach to fake news detection is through the use of statistical machine learning algorithms,

which can be trained on large datasets of labelled news articles to automatically identify patterns and features that distinguish between real and fake news.
Among all the strategies, using Algorithms is the best way to stop spreading fake news and detecting the ‘fake’ and ‘real'.
Statistical Machine learning techniques have emerged as promising tools for addressing the challenge of fake news detection due to their ability to analyse large volumes of textual data and identify patterns indicative of misinformation. This project provides an overview of the application of machine learning techniques in the detection of fake news. It explores various methodologies, including feature extraction, classification algorithms, and model evaluation, employed in fake news detection research. We will be training and testing the data, when we use supervised learning, it means we are labelling the data. By getting the testing and training data and labels we can perform different machine learning algorithms but before performing the predictions and accuracies, the data is need to be preprocessing i.e. the null values which are not readable are required to be removed from the data set and the data is required to be converted into vectors by normalizing and tokening the data so that it could be understood by the machine.

2 OBJECTIVE:

The main goal of the project is to use statistical machine learning techniques to develop and implement an efficient system for identifying fake news. We will to address the following particular issues:
1.Identification of Features: To identify and extract relevant features from textual data that can serve as indicators of fake news. This involves exploring various feature extraction methods one of them is TF-IDF (Term Frequency - Inverse Document Frequency), to capture the distinguishing characteristics of fake news articles.
2.Model Development: To create statistical machine learning models using the extracted features that can tell the difference between real and fake news articles. This means investigating various machine learning algorithms, including, logistic regression, random forests, support vector machines (SVM), K-Nearest Neighbour(K-NN), Decision Tree, Gradient Boosting, Bagging, to identify the most effective approach for fake news detection.
3.Train and evaluation: Train and evaluate the performance of developed models using labelled datasets of real and fake news articles. This involves dividing data sets into training and test sets, training models on training data, and evaluating their performance on test data using appropriate evaluation metrics such as precision, accuracy, recall, and F1 scores.
4.Application: Make a model by using the above algorithms, where we can detect which news is the real news or which one is the fake news.

3 LITERATURE REVIEW:

Dealing with the problem of fake news has become increasingly important in the digital era. Several research studies were performed with different perspectives, approaches, and techniques to counteract misinformation. Here are some notable contributions:
The study by Agudelo et al. [1] offered a novel method for identifying fake news using Python programming, natural language processing, and machine learning methods. On a dataset with over 10,000 news items, they were able to achieve high classification accuracy of 88.1% and 84.8%, respectively, by using the CountVectorizer, TfidfVectorizer, and Multinomial Naive Bayes models. However, the ethical ramifications of automatically detecting false news were not covered.

In their study Patel et al. [2] covered the use of different classifiers (i.e. K-NN, SVM) and natural language processing methods to identify false news. They underlined how crucial early detection is in stopping the spread of false information. With an accuracy of 90.42% for adaptive learning rate and 90.32% for constant learning rate, the SGD (Statistical Gradient Descent) method was shown to be the

most efficient. K-means had a very low accuracy of 40.37% in predicting fake news, whereas SVM and KNN also produced good results, with accuracy of 88.47% and 86.90%, respectively.

In their study [3], Mykhailo Granik et al. present a straightforward method for detecting fake news using a naïve Bayes classifier. This method was put into practice as a software solution and evaluated using a collection of Facebook news posts. They were gathered from three large mainstream political news pages (Politico, CNN, ABC News) as well as three large Facebook pages each from the right and from the left. A classification accuracy of almost 74% was attained. The accuracy of classification for bogus news is somewhat lower.
The skewness of the dataset only 4.9% of it contains fake news could be the reason for this.

Kulkarni et al [4] addressed the problem of fake news in online resources by proposing a machine learning based detection model. Their study used classifiers such as Random Forest, Logistic Regression, Decision Tree, KNN and Gradient Booster. The results showed that Logistic Regression achieved the highest accuracy of 85.04 percent, followed by Random Forest with 84.50 percent accuracy. The accuracy of the KNN algorithm and the decision tree was 80.20% and the decision tree was 78.11%, while the accuracy of the Gradient Boosting algorithm was the lowest at 77.44%.
Marco L. Della Vedova et al. to [5] first proposed a novel ML fake news detection method, which combining news content and social context features, outperforms existing methods found in the literature, increases its accuracy up to 78.8%. Second, they implemented their method on Facebook, Messenger Chabot and validated it with a real-world application, with an accuracy of 81.7% in detecting fake news.
Vikram Tembhurne and Almin [6] proposed a Multi-channel Deep neural network (McDNN) model to detect fake news, surpassing existing techniques. They presented potential applications of the McDNN model in various industries. The study showed that McDNN outperformed the most advanced machine learning and deep learning techniques in identifying fake news with 94.68% accuracy for fake news data (FND) and 99.23% accuracy for ISOT Fakes News data set.
Himank Gupta et. al. [7] gave a system based on different machine learning approach that deals with different issues including accuracy shortage, time lag (BotMaker) and high processing time to handle thousands of tweets in 1 sec. Firstly, they have collected 400,000 tweets from HSpam14 dataset. At that point they assist characterize the 150,000 spam tweets and 250,000 non- spam tweets. They moreover inferred a few lightweight highlights alongside the Top-30 words that are giving highest information pick up from Bag-of-Words model. 4. They were able to realize an accuracy of 91.65% and surpassed the existing solution of approximately 18% by approximately18%.

4 METHODOLOGY:

This section presents the methodology used for the classification. In this method statistical machine learning is used for classifying the dataset. The first step in this classification problem is dataset collection phase, followed by preprocessing, implementing features selection, then perform the training and testing of dataset and finally running the classifiers. Figure [1] describes the proposed system methodology. The methodology is based on conducting various experiments on dataset using the algorithms like Logistics Regression, Decision Tree, SVM, K-NN, Random Forest etc. which are describe below. The experiments are conducted individually on each algorithm

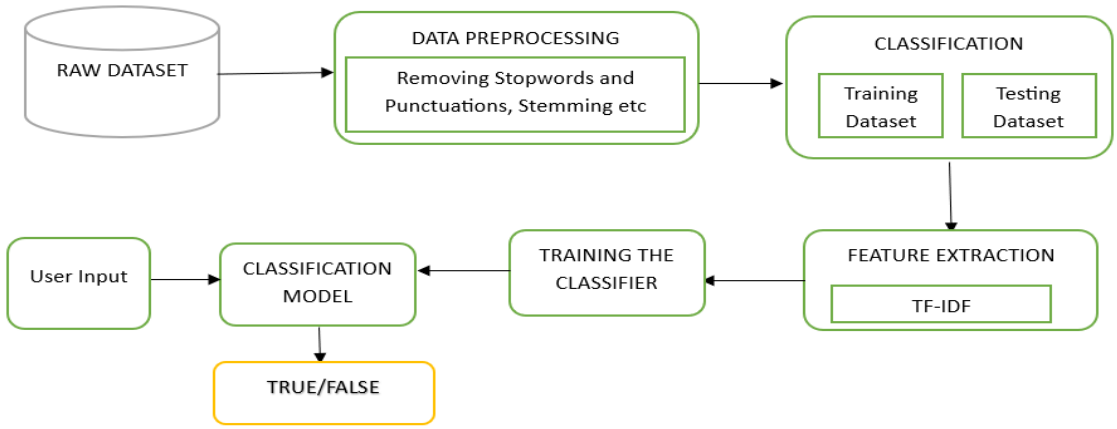


Figure 1: System Architecture

The essential aim is to use a fixed model so as to be used as a scanner for a fake news through news of news detection and embed the version in python software for use as a discovery for the fake news data. The classification algorithms applied in this model are k-Nearest Neighbors (k-NN), Logistics Regression, Gradient Boosting, Bagging, Decision Tree, Random Forests and Support Vector Machine (SVM).
As shown in the figure [2], the dataset is implemented to exclusive algorithms so that all can discover a fake news

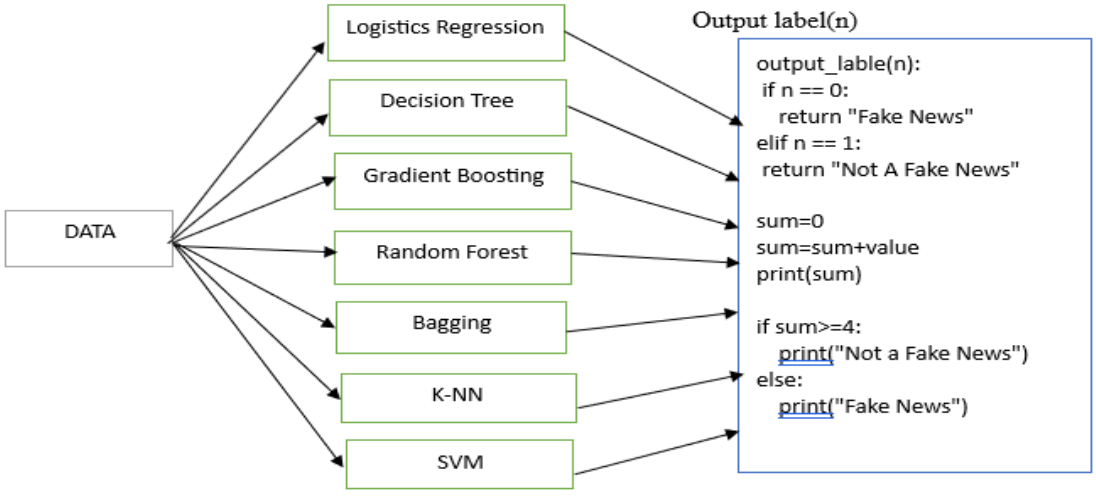


Figure 2: Classification Algorithms

The process of model creation, the approach of detecting fake news is as follow:

4.1 MODEL DESCRIPTION:

A DATA USE:

Here we have taken two dataset one is fake news dataset and other one is true news dataset of America from the online website https://www.simplelearn.com/tutorials/machine-learning-tutorial/how-to-create-a-fake-news-detection-system. The datasets include 8429 fake news which in csv file and 11677 true news which in xlsx file .Here we can read the .csv file and .xslx file then we display the first few

datapoints of our datasets to display the shape of the dataset. We can divide the dataset into two parts training and testing. By getting the testing and training data and labels we can perform different machine learning algorithms but before performing the predictions and accuracies, the data is need to be preprocessing i.e. the null values which are not readable are required to be removed from the data set and the data is required to be converted into vectors by normalizing and tokening the data so that it could be understood by the machine.

**We use the same methodology for Indian Data. Here we have taken two dataset one is fake news dataset and other one is true news from the online platform and download the archive.zip file. The datasets include 7172 fake news and 9500 true news.

B DATA PREPROCESSING:

Cleaning the dataset is the first step of the project. The data contains the undesirable characters like stop words, digits etc. which creates hindrance while fake news detection. By removing unnecessary features from our text, we can reduce complexity. Removing punctuation, special characters and filter words (i.e. the, a, an etc) does not drastically change the meaning of a text.

- Data Cleaning: -We read data in structured or unstructured formats. Structured formats have a well-defined structure, while unstructured formats have no structure at all. In between the two formats, we have a semi-structured format. Semi-structured formats are comparable to structured formats. The purpose of cleaning text data is to highlight the attributes that we want our machine learning system to recognize. Clearing (or pre-processing) text data typically involves several steps:

- Lowercase Text: - We lowercase the text to reduce the size of the vocabulary of our text data.
  Example: - "Donald Trump Sends Out Embarrassing New Year Eve Message" in lowercase it is "donald trump sends out embarrassing new year eve message"
- Removal Number: - We can either delete the numbers or you can convert them into their text representation. we can also use regular expressions to get remove of the numbers.
  Example: - "10 Reasons Donald Trump And Kim Jong-Un Are Basically The Same Person." In Removal Number it is "Reasons Donald Trump And Kim Jong-Un Are Basically The Same Person."
- Remove Punctuations: - Punctuation can provide grammatical context to a sentence that supports our understanding. But for our vectorizer that counts the number of words and not the context, it does not value, so we remove all the special characters.
  Example: - "Trump Is Being Sued By A Watchdog For Mar-A-Lago Corruption: Will Visitor Logs Show Pay-To-Play?" the text transfer "Trump Is Being By A Watchdog For Mar A Lago Corruption Will Visitor Logs Show Pay To Play"
- Remove Whitespace: - We can use the join and split function to remove all the white spaces in a string.
  Example: - "washington reuters andrew olmem an attorney" the text transfer "washington reute r andrew olmem an attorney"

- Tokenization: - If the text is not in characters, we need to convert it to characters. After converting text strings to characters, we can convert word characters to their root form. There are mainly three root algorithms. They are Porter Stemmer, Snowball Stemmer and Lancaster Stemmer. Porter Stemmer is the most common of them.
  Example: - "The Eighth Person In Trump Tower Meeting Is Linked To Money Laundering" after Tokenize the text convert to 'The', 'Eighth', 'Person', 'In', 'Trump', 'Tower', 'Meeting', 'Is', 'Linked', 'To', 'Money', 'Laundering'.
- Remove Stopwords: - Stop words are words that do not affect the meaning of the sentence. So, they can be safely deleted without changing the meaning of the sentence. The NLTK library has a set of stopwords that allow us to remove stopwords from text and return a list of word tokens.
  Example: - "Racist Chicago Cop Who Beat A Handcuffed Black Woman Found With Trove Of Slur-Filled Websites" After Stopwords remove the text looks like "Racist Chicago Cop Beat Handcuffed Black Woman Found With Trove Slur-Filled Websites."

- Stemming: - Stemming is the process by which the root form of a word is obtained. A stem or stem is the part to which case suffixes (-ed, -ize, -de, -s, etc.) are added. A word stem is created by removing a prefix or suffix from a word. So, word may not lead to true words.
  Example: - 'Returns'-> 'Return', 'Tired'-> 'Try' etc.
- Lemmatization: - Like stemming, lemmatization turns a word into its root form. The only difference is that lemmatization ensures that the root word belongs to the language. If we use lemmatization, we get valid words. In NLTK, we use WordNetLemmatizer to get word lemmas. We also need to provide a context for lemmatization. So, we add the part of speech as a parameter.
  Example: - "Trump Just Accidentally Confirmed Russian Contact With Sessions In Latest Twitter Fit" convert to 'Trump', 'Just', 'Accidentally', 'Confirm', 'Russian', 'Contact', 'With', 'Sessions', 'In', 'Latest', 'Twitter', 'Fit'.

After the preprocessing step we divided the dataset in two parts
1.Training Dataset
2.Testing Dataset
The dataset is divided into 75% for train data and 25% for test data using python sklearn. In the train data section, the algorithm detects the real news and fake news, the data is labelled in the form of 0 and 1 where 0 is for fake news and 1 for true news. After that, the rest of the data, which is 25% of it, will do a test on it, and we compute the accuracy of the test data so that it is sure whether the news is fake or true, and then return it in case it was right or wrong, the algorithm percentage will be formed.

C FEATURE EXTRACTION:
Feature extraction is the process of selecting a subset of relevant features for use in model construction. Feature extraction methods helps in to create an accurate predictive model. They help in selecting features that will give better accuracy. When the input data to an algorithm is too large to be handled and it is supposed to be redundant then the input data will be transformed into a reduced illustration set of features also named feature vectors. Altering the input data to perform the desired task using this reduced representation instead of the full-size input. Feature extraction is performed on raw data prior to applying any machine learning algorithm, on the transformed data in feature space.

- Vectorizing Data: Vectorization is the process of encoding text as integers i.e. a numeric format to generate feature vectors so that machine learning algorithms can make sense of our data.
- Vectorizing TF-IDF: Term Frequency - Inverse Document Frequency (TF-IDF) is a widely used statistical method in natural language processing and information retrieval. Measures how important a term in a document is relative to a set of documents (i.e. relative to a corpus).

Words in a text document are converted to importance numbers in the text vectorization process. There are many different scoring systems for text vectorization, with TF-IDF being one of the most common. As the name suggests, TF-IDF vectorizes/scores a word by multiplying the term frequency (TF) of the word by the inverse density (IDF) of the document.
The TF of a term or word is the number of times the term occurs in a document compared to the total number of words in the document.

$$TF=\frac{\text{Number of Times The Term Appeares in the Document}}{\text{Total Numebr of Terms in the Document}}$$

Inverse document density: The IDF of a term reflects the proportion of documents containing that term in the corpus. Words that are unique to a small percentage of documents (e.g. technical jargon terms) are given more importance than words that are common to all documents (e.g. a, the, and).

$$IDF=\log\left(\frac{\text{Number of Documents in the Corpus}}{\text{Number of documents in the Corpus contain the term}}\right)$$

The TF-IDF expression is calculated by multiplying the TF. and IDF scores.

$$TF\text{-}IDF= TF*IDF$$

D TRAINING AND CLASSIFIER:

In this project we are using Scikit-Learn Machine learning library for implementing the architecture. This just needs importing the packages and we can compile the command as soon as we write it. If the command doesn't run, we can get the error at the same time. We are using in my project 7 different algorithms and we have trained these 7 models i.e. Decision Tree Classifier, Gradient Boost Classifier, Random Forest Classifier, Bagging Classifier, Support Vector Machine, K Nearest Neighbors and Logistic Regression are very popular methods for document classification problem. Once the classifiers are trained, we can check the performance of the models on test-set.

E ALGORITHMS:

This section deals with training the classifier. Different classifiers were investigated to predict the class of the text. We explored specifically seven different machine learning algorithms-

❖ LOGISTICS REGRESSION (LR):

Logistic regression is one of the most popular statistical machine learning algorithms that belongs to the supervised learning technique. It is used to predict a categorical dependent variable using a given set of independent variables. LR predicts the output of a categorical dependent variable. Therefore, the result must be a categorical or discrete or categorical value. It can be either yes or no, 0 or 1, true or false, etc. Instead of giving an exact value of 0 and 1, it gives probability values that range from 0 to 1. Logistic regression is similar to linear regression except for how they are used. Linear regression is used to solve regression tasks, while logistic regression is used to solve classification tasks. In LR, instead of the regression line, an "S"-shaped logistic function is fitted, which predicts two maximum values (0 or 1). A logistic function curve shows a certain probability, such as whether cells are cancerous or not, emails are spam or not, etc. Logistic regression is a notable machine learning algorithm for its ability to represent probabilities and classify new data, using continuous and discrete data sets. Logistic regression uses the concept of predictive modelling as regression; therefore, it is called logistic regression, but it is used to classify samples; Therefore, it is categorized under the classification algorithm. LR can be used to classify observations using different types of data and allows one to easily determine the most effective variables to use for classification.
The below image is showing the logistic function:

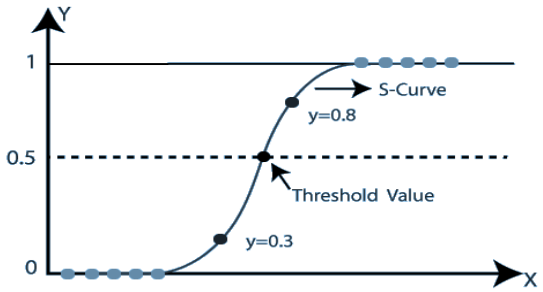


Figure 3: Logistic Function

- Logistic Function (Sigmoid Function): The sigmoid function is a mathematical function used to map predicted values to probabilities. It maps any real value to another value between 0 and 1. The LR value must be between 0 and 1, which cannot exceed this limit. So, it forms a curve like an "S" shape. An S-shaped curve is called a sigmoid function or logistic function. In LR, the concept of a threshold value is used, which defines a probability of either 0 or 1. For example, values above the threshold tend to be 1 and values below the threshold tend to be 0.

- Assumptions for Logistic Regression: We will explore the assumptions of logistic regression as understanding these assumptions is important to ensure that we are using appropriate application of the model. The assumption include:

1. Independent observations: Each observation is independent of the other. meaning there is no correlation between any input variables.
2. Binary dependent variables: It takes the assumption that the dependent variable must be binary or dichotomous, meaning it can take only two values. For more than two categories SoftMax functions are used.
3. Linearity relationship between independent variables and log odds: The relationship between the independent variables and the log odds of the dependent variable should be linear.
4. No outliers: There should be no outliers in the dataset.
5. Large sample size: The sample size is sufficiently large.

- Logistic Regression Equation: Logistics Regression is a statistical machine learning techniques used for binary classification tasks. The logistics Regression equation can be represented as:

$$P(Y = 1|X) = \frac{1}{1+e^{-(\beta 0+\beta 1x1+\beta 2x2+\cdots.\beta nxn)}}$$

Where:
P(Y=1|X) is the probability of the dependent variables (y) being 1 given the independent variables(x).
(e) is the base of the natural logarithm.
(β_0, $\beta_{1\,,\,\ldots}$, $\beta_{n)\ are}$ the coefficient of the independent variables
(x_1, x_2, x_3…, x_n) are the independent variables.

- Type of Logistic Regression: On the basis of the categories, Logistic Regression can be classified into three types:
  - Binary logistic regression: In this approach, the response or dependent variable is dichotomous in nature—i.e. it has only two possible outcomes (e.g. 0 or 1). Some popular examples of its use include predicting if an e-mail is spam or not spam or if a tumour is malignant or not malignant.
  - Multinomial logistic regression: In this type of logistic regression model, the dependent variable has three or more possible outcomes; however, these values have no specified order. A multinomial logistic regression model can help the studio to determine the strength of influence a person's age, gender, and dating status may have on the type of film that they prefer.
  - Ordinal logistic regression: This type of logistic regression model is leveraged when the response variable has three or more possible outcome, but in this case, these values do have a defined order. Examples of ordinal responses include grading scales the exam number such as "low", "Medium", or "High".

- Process of Logistic Regression work in the project: Logistic regression is a statistical model used to analyze the relationship between input characteristics and a binary outcome variable.
  1. In this case, the input functions represent the news articles and the output variable indicates whether the article is genuine or fake. We denote the input characteristics by X, where X = [x_1, x_2, x_3……x_n] and the output variable by y, where y takes the binary values either 1 (indicates a real news article) or 0 (indicates a fake news article). The purpose of the logistic regression model is to estimate the probability that the article is fake (y = 0) using the vectorized input characteristics X.
  2. If the logistic function returns the value >0.5 then it is considered to be as real news (1) otherwise it is detected as fake news (0)
  3. Evaluation: From test data set we validate for how many cases it results properly.
     To do so we first predict for the test data set the output that is fake or not fake and compare them with the original output and check how many percent it provides correct results.

❖ DECISION TREE (DT):

Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome. In a Decision tree, there are two nodes, which are the 1) Decision Node and 2) Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches. The decisions or the test are performed on the basis of features of the given dataset. It is called a decision tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure. A decision tree can contain categorical data (YES/NO) as well as numeric data. In order to build a tree, we use the CART algorithm, which stands for Classification and Regression Tree algorithm. A decision tree simply asks a question, and based on the answer (Yes/No), it further split the tree into subtrees.

Below diagram explains the general structure of a decision tree:

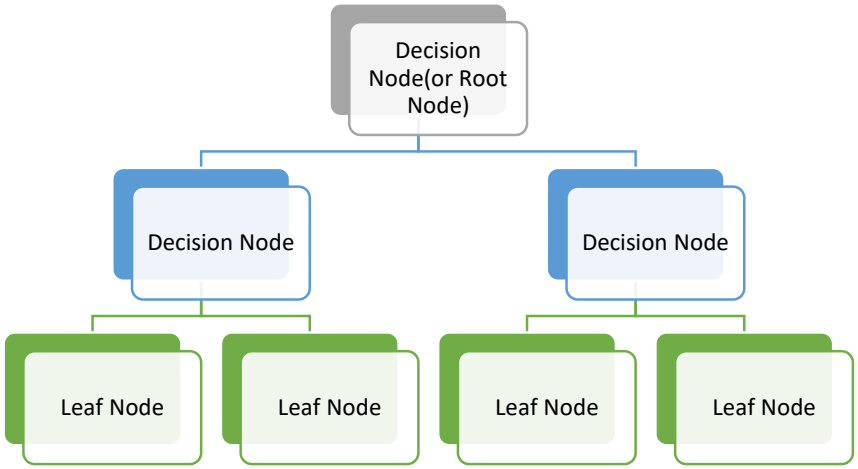


Figure 4: Structure Of Decision Tree

- Example: -In the following example, we've to approve a loan on the basis of the age, salary, and no. of children the person has. We ask a conditional question at each node and make splits accordingly, till we reach a decision at the leaf node (i.e., Get loan/Don't get loan)

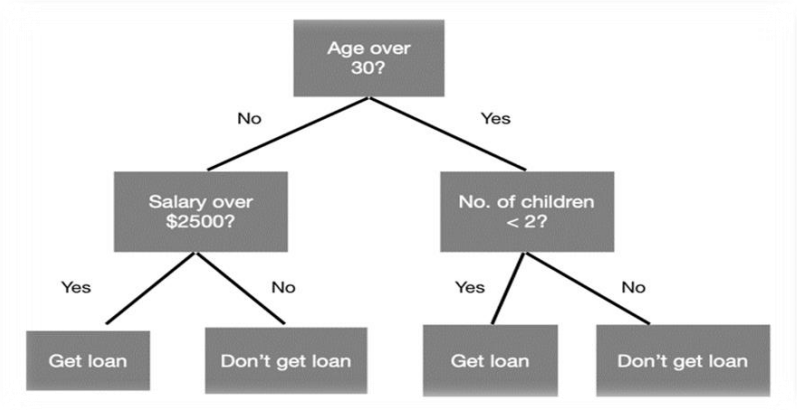


Figure 5: Example of a Decision Tree

- Why use Decision Trees?

1.Decision Trees usually mimic human thinking ability while making a decision, so it is easy to understand.
2.The logic behind the decision tree can be easily understood because it shows a tree-like structure.

- Mathematical Analysis: -The decision tree algorithm works by recursively partitioning the data into subsets based on the values of the input variables, using a tree-like structure to represent the decision-making process. At each node of the tree, a decision is made based on the values of one of the input variables, with each possible outcome leading to a new branch in the tree. The algorithm continues to split the data into smaller and smaller subsets until a stopping criterion is met

- Decision Tree Terminologies: -

>>Root Node: Root node is from where the decision tree starts. It represents the entire dataset, which further gets divided into two or more homogeneous sets.
>>Leaf Node: Leaf nodes are the final output node, and the tree cannot be segregated further after getting a leaf node.
>>Splitting: Splitting is the process of dividing the decision node/root node into sub-nodes according to the given conditions.
>>Branch/Sub Tree: A tree formed by splitting the tree.
>>Pruning: Pruning is the process of removing the unwanted branches from the tree.
>>Parent/Child node: The root node of the tree is called the parent node, and other nodes are called the child nodes.

- Steps of Decision Tree: -In a decision tree, for predicting the class of the given dataset, the algorithm starts from the root node of the tree. This algorithm compares the values of root attribute with the real dataset attribute and, based on the comparison, follows the branch and jumps to the next node.

For the next node, the algorithm again compares the attribute value with the other sub-nodes and move further. It continues the process until it reaches the leaf node of the tree. The complete process can be better understood using the below algorithm:

Step-1: Begin the tree with the root node, says F, which contains the complete dataset.
Step-2: Find the best attribute in the dataset using Attribute Selection Measure (ASM).
Step-3: Divide the F into subsets that contains possible values for the best attributes.
Step-4: Generate the decision tree node, which contains the best attribute.
Step-5: Recursively make new decision trees using the subsets of the dataset created in Step-3. Continue this process until a stage is reached where you cannot further classify the nodes and called the final node as leaf node.

- Application of Decision Tree in the project:
  1. Start with the root node
  2. And its choses the content as the feature value and choses a threshold by default.
  3. And on the basis of the above criteria, it detects fake news.

❖ GRADIENT BOOST CLASSIFIER (GBC):

Gradient Boosting is a popular Boosting algorithm in machine learning algorithm that is applied to regression and classification problems. One type of ensemble learning technique is called "boosting," in which the model is trained successively, with each new model attempting to improve upon the one before it. It turns a number of ineffective learners into effective ones. The two most widely used boosting algorithms are:

1.AdaBoost
2.Gradient Boost
Gradient Boosting is combines several weak learners into strong learners, wherein every new model is trained to minimize the loss function such as mean squared error or cross-entropy of the preceding model the usage of gradient descent. In every iteration, the set of rules computes the gradient of the loss function with recognize to the predictions of the current ensemble after which trains a new weak model to decrease this gradient. The predictions of the new model are then introduced to the ensemble, and the procedure is repeated till a stopping criterion is met.
GBM helps us to get a predictive model in form of an ensemble of weak prediction models such as decision trees. Whenever a decision tree performs as a weak learner then the resulting algorithm is called gradient-boosted trees.

- Real world Application of Gradient Boosting:

1.Retail and e-commerce: personalized recommendations, inventory tracking, fraud detection.
2.Finance and insurance: credit risk assessment, volatility forecasting, algorithmic trading.
3.Healthcare and medicine: disease diagnosis, drug development, personalized medicine.
4.Search and online advertising: search ranking, ad targeting.

- How do Gradient Boosting Classifier?

In general, most supervised learning algorithms are based on a single predictive model such as linear regression model, penalized regression model, decision trees, etc. But in ML, there are some supervised algorithms that depend on a combination of several different models. Ensemble another words, as multiple base models increase their predictions, an average of all predictions is adjusted by the boosting algorithms.
Gradient machines consist of 3 elements as follows:
1.Loss function: Machine learning, however, has a large family of Loss functions that can be used depending on the type of problem to be solved. The use of the loss function is evaluated based on special properties of the conditional distribution, such as the robustness requirement. If we use a loss function in our problem, we need to define a loss function and a function to calculate the corresponding negative gradient. Once we get these two functions, they can be easily implemented in gradient machines. However, several loss functions have already been proposed for GBM algorithms.
2.Weak learners: Weak learners are basic learning models that learn from past errors and help build a strong predictive model that supports machine learning algorithms. In general, decision trees act as weak learners when improving algorithms. Boosting is defined as the framework that continuously tries to improve the output of base models. Many gradient boosting applications allow you to "plugin" weak learners to different classes at your disposal. Therefore, decision trees are mostly used for weak (basic) learners.
3.Additive model: A composite model is defined as the addition of trees to a model. Although we should not add several trees at once, only one tree should be added so as not to change existing trees in the model. In addition, we can also prefer the gradient descent method by adding trees to reduce the loss.

- Algorithm of Gradient Boosting Classifier:

1.Initial Prediction: Start with an initial model that makes a simple prediction for each instance. For binary classification, this is often a constant value that represents the average prediction.

$$F_0(x) = \text{argmin}_\gamma \sum_{i=1}^{n} L(y_i, \gamma)$$

where L is the log loss function for classification, y_i are the actual labels, and γ is the initial prediction.

2.Residuals Calculation: Calculate the residuals (errors) for each instance based on the current model's predictions. These residuals indicate how far off the current predictions are from the actual values.

$$R = -\left[\frac{\partial F(xi)\, \partial L(yi, F(xi))}{\partial F(xi)}\right]_{F(xi)\text{-}Fm\text{-}1(xi)}$$

where R are the residuals for iteration m, L is the loss function, and $F_{m-1}(xi)$ is the prediction from the previous model.

3.Train Weak Learner: Fit a new weak learner (e.g., a decision tree) to the residuals from the previous step. This learner tries to predict the residuals of the current model.

$H_m(x)=\text{argmin}\gamma \sum_{i=1}^{n}(R - h(xi))2$

where $H_m(x)$ is the new weak learner fitted to the residuals.

4. Update Predictions: Update the current model by adding the predictions from the new weak learner, scaled by a learning rate η. This helps to control the contribution of each weak learner.

$F_m(x)=F_{m-1}(x)+\eta \cdot H_m(x)$

where $F_m(x)$ is the updated model, $F_{m-1}(x)$ is the previous model, η is the learning rate, and $H_m(x)$ is the new weak learner.

5.Iterate: Repeat steps 2-4 for a predefined number of iterations M or until the residuals are minimized sufficiently.

$F_M(x)=F_{M-1}(x)+\eta \cdot H_M(x)$

6.Combine Models: The final model is the sum of the initial model and all the weak learners, each scaled by the learning rate.

$F_M(x)= F0(x) + \eta \sum_{i=1}^{n} Hm(x)$

- Application of Gradient Boosting Classifier in the project:

1.At first, we fixed an initial value, suppose we predict every news article is "real" initially.
2. Calculate the residuals (errors) for each news article based on the initial prediction. If the actual label is "fake" and the initial prediction is "true", the residual will show this error.
3. Train a decision tree based on these residuals to predict whether an article is fake or true. The tree learns patterns that distinguish between the two classes.
4. Update the model by adding predictions from the new decision tree scaled by learning rate. This adjusts the model to better predict residuals.
5. Repeat the process several times. Each iteration adds a new decision tree that corrects the errors of the combined previous models.
6. The final prediction for each news article is the sum of the initial prediction and all adjustments made by the decision trees. This final model should be able to distinguish between fake and true news articles more accurately.

❖ RANDOM FOREST CLASSIFIER (RF):

Random Forest (RF) is a popular machine learning algorithm Random Forest (RF) is an advanced form of decision trees (DT) which is also a supervised learning model. RF can be used for both Regression and Classification problems in Machine Learning. It is mainly based on the concept of Ensemble learning. It is a combination of multiple decision tree that are created using different random subsets of the features and training data. The RF takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

The Diagram explains the working of the Random Forest algorithm.

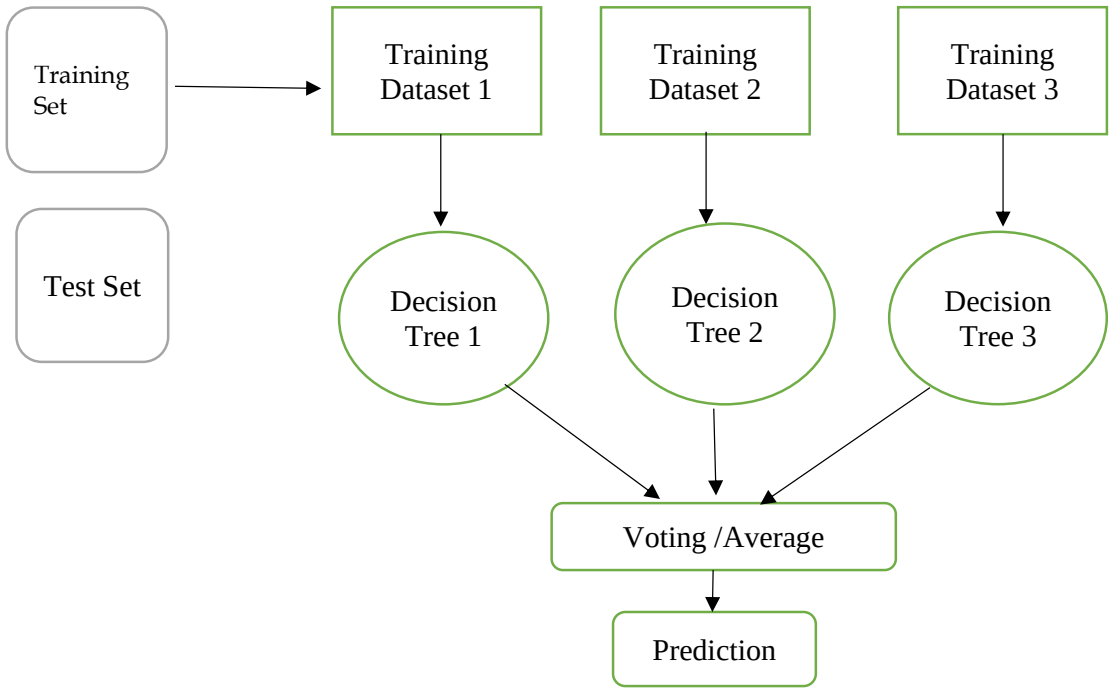


Figure 6: Structure of Random Forest

➢ How does Random Forest algorithm work?

Random Forest works in two-phase at first is to create the random forest by combining N decision tree, and second is to make predictions for each tree created in the first phase.
The Working process can be explained in the below steps and diagram:
Step-1**:** Select random K-data points from the training set.
Step-2: Build the decision trees associated with the selected data points.
Step-3**:** Choose the number N for decision trees that you want to build.
Step-4: Repeat Step 1 and 2.
Step-5: For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.

➢ Application of Random Forest in the project:

1. From the training dataset we select random K-data points and make K decision tree.
2. We input some test data then we get the K number of output (suppose some decision trees are provide the news is fake). And in this algorithm, we combined all the decision tree.
3. we predict the output by Voting (i.e. the news is fake).

❖ BAGGING CLASSIFIER (BC):

Bagging is an ensemble method that involves training multiple models independently on random subsets of the data, and aggregating their predictions through voting or averaging. Bagging is also known as Bootstrap aggregating. In the case of the bagging classifier, the final prediction is made by aggregating the prediction of all the base model using majority voting. In the. In the models by averaging the predictions of the all-base model and known as bagging regression. The Bagging avoids overfitting of data and improve accuracy, especially

in models that have high variance and it is used for each classification and regression of the class, in particular for the decision tree algorithms.

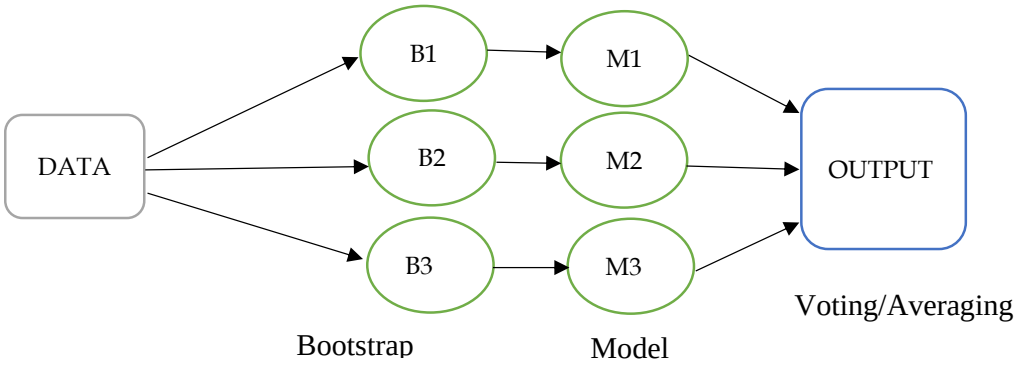


Figure 7: Structure Of Bagging Classifier

➢ The steps of bagging are as follows:

1. Multiple subsets are created from the original dataset with the same tuples, selecting observations with replacement. .
2. A base model is built on all subsets.
3. Each version is found simultaneously with each training set.
4. Each version is unbiased.
5. The final predictions are calculated by combining the predictions from all models.

➢ Application of Bagging in the project:

1. At first, we input a data or a news. multiple subsets are created from the original dataset, selecting observations with replacement.
2. A model is built on all subsets.
3. The final Predictions are calculated by combining the predictions from all models. If most of the models are say the news are fake then the output must be fake.

❖ K-NEAREST NEIGHBOR (K-NN):

K-Nearest Neighbor is one of the simplest Machine Learning algorithms based on Supervised Learning technique. It assumes the similarity between the new case and available cases and put the new case into the category that is most similar to the available categories. K-Nearest Neighbor 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 K-NN algorithm. It can be used for Regression as well as for Classification but mostly it is used for the Classification problems. K-NN is a non-parametric algorithm, which means it does not make any assumption on underlying data. 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 on the dataset. K-NN 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 similar to the new data.

➢ Why do we need a K-NN Algorithm?

Suppose there are two categories, i.e., Category A and Category B, and we have a new data point x1, so this 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 diagram:

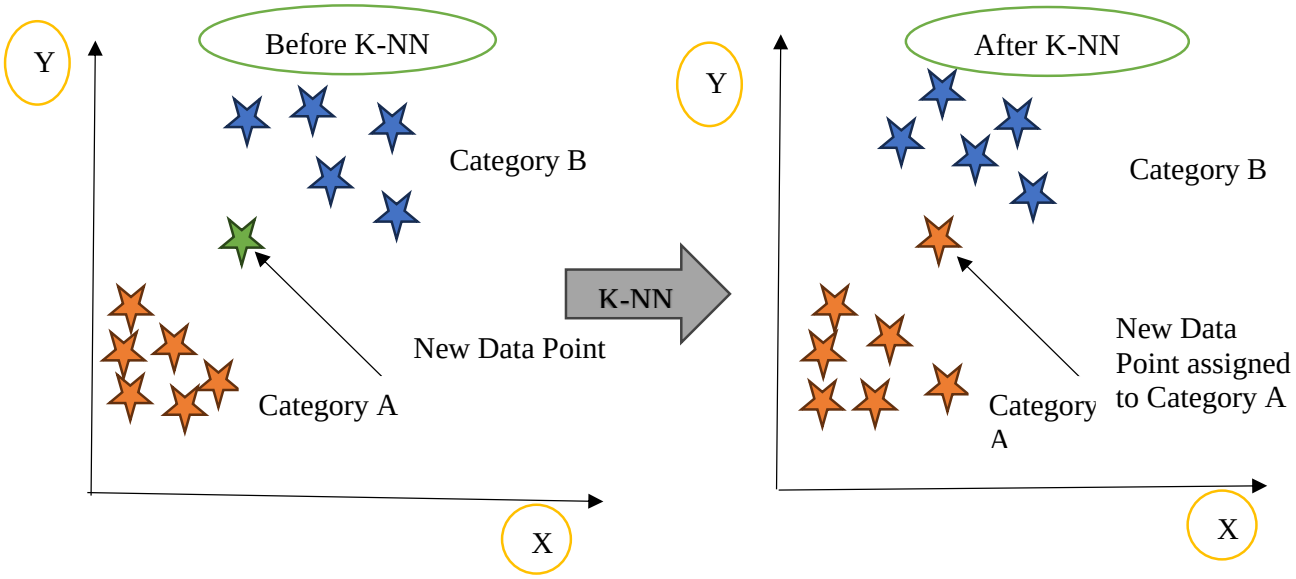


Figure 8: Detecting a particular category of a particular datapoint using K-NN

➢ Mathematical Analysis: -The first step is to calculate the distance between the new point and each training point. There are various methods for calculating this distance, of which the most commonly known methods are — A) Euclidian, B) Manhattan (for continuous) and C) Hamming distance (for categorical).

Euclidean distance is calculated as the square root of the sum of the squared differences between a new point (x) and an existing point (y).

Manhattan Distance: This is the distance between real vectors using the sum of their absolute difference.

Distance Function:

- Euclidean Distance: $\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$
- Manhattan Distance: $\sum_{i=1}^{k}|x_i - y_i|$

➢ Steps of K-NN Algorithm:

1: Select the number K of the neighbors.

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

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

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

5: Assign the new data points to that category for which the number of the neighbor is maximum.

6: The model is ready.

➢ Application of K-NN in the project:

1. From the Figure 8 we can elaborate that, suppose we take Category B as a Fake News and Category A as a True News. Now at first, we choose the number of neighbors, the value K (Python will choose the value by itself)
2. After that we calculating the distance between the data points and get the nearest neighbors. If maximum number of nearest neighbors in category A (i.e. True News), then we can say that the new dataset must be True.

❖ SUPPORT VECTOR MACHINE(SVM):

Support Vector Machine (SVM) is a most popular supervised machine learning technique. It is effective device getting to know set of rules used for linear or nonlinear classification, regression, or even outlier detection tasks. SVMs may be used for loads of tasks, along with textual content classification, photo classification, junk mail detection, handwriting identification, gene expression analysis, face detection, and anomaly detection. SVMs are adaptable and green in loads of programs due to the fact they are able to manipulate high-dimensional information and nonlinear relationships.

The SVM algorithm's primary goal is to locate the best hyperplane in an N-dimensional space that may be used to divide data points into various feature space classes. The hyperplane attempts to maintain the largest possible buffer between the nearest points of various classes. The number of features determines the hyperplane's dimension. The hyperplane is essentially a line if there are just two input features. The hyperplane transforms into a 2-D plane if there are three input features. If there are more than three features, it gets hard to imagine.

Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane:

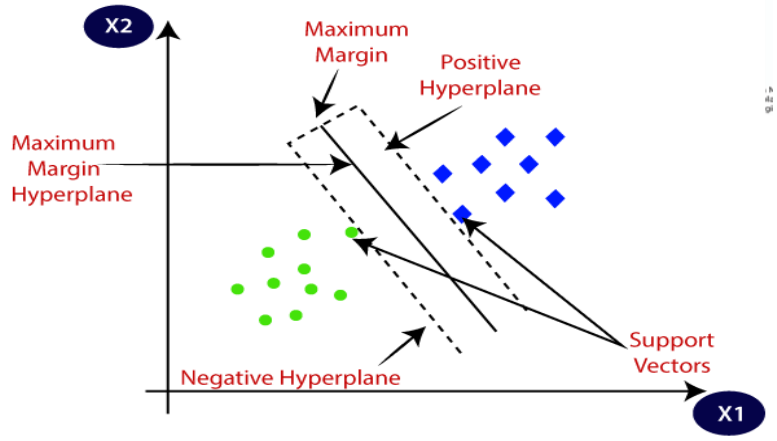


Figure 9: Diagram of classified two different categories using SVM

- Mathematical intuition of Support Vector Machine: Consider of a binary classification task where there are two classes, denoted by the labels +1 and -1. The input feature vectors (G) and the matching class labels (H) comprise our training dataset.

The linear hyperplane equation is expressed as follows: -

$$c^T G + d = 0$$

The direction perpendicular to the hyperplane, or the normal vector, is represented by the vector C. The offset, or distance, of the hyperplane from the origin along the normal vector c is represented by the parameter d in the equation.

The distance between a data point G_i and the decision boundary can be calculated as:

$$X_i = \frac{cTG + d}{||c||}$$

where ||c|| represents the Euclidean norm of the weight vector w. Euclidean norm of the normal vector c.

For Linear SVM classifier:

$$Y = \begin{cases} 1; c^T G + d \geq 0 \\ 0; c^T G + d < 0 \end{cases}$$

- Steps of SVM Algorithms:

1.Open the sklearn fake news dataset and true news dataset.

2. Keep target variables and input features apart.

3.Use the RBF kernel to construct and train the SVM classifiers.
4.Plot the input feature scatter plot.
5.Draw the boundaries of the decision.

- Application of SVM in the project:
  1.We have two different sets fake news and true news. The news(paragraphs or texts) are convert in vectors Then these vectors are measured with their norm and direction. Now we make a hyperplane which divides the vectors containing news(both of type fake or real).

  2.The hyperplane is used as a classifier between the fake and real news, which takes a certain vector of news and calculates the value of the classifier addressed beforehand. Now we input a test data which pass through the classifier. If the classifier takes 1 then we address the news as real news and if it takes 0 then we address the news as fake news.

5 RESULT AND ANALYSIS:

This section aims to evaluate the prediction capabilities of the suggested methodology. The trials were carried out using Python as the programming language.

5.1 PERFORMANCE METRICES: The result of the proposed work is analysed using Confusion Matrix and its different statistical metrics are shown below.

5.1.1 CONFUSION MATRIX: A confusion matrix is a table used to evaluate the performance of a classification model in machine learning. The number of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) for each class in the dataset is displayed in a matrix. The confusion matrix is a helpful tool for assessing a classification model's performance parameters, including accuracy, precision, recall, and F1 score, and pinpointing areas in need of development. Table 1 presents a confusion matrix for a binary classification problem with two classes, denoted as positive and negative.

Table 1 shows the diagram of the Confusion Matrix.

Table 1: CONFUSION MATRIX

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positive (TP)	False Positive (FP)
Predicted Negative (0)	False Negative (FN)	True Negative (TN)

From the Table1 accuracy, precision, recall and F1-score are defined as:

- ACCURACY: -The percentage of accurately identified cases is used to calculate accuracy. This is calculated by dividing the number of correct predictions by the total number of instances in the dataset. The equation can therefore be used to determine the accuracy.

  $$\text{Accuracy} = (TP + TN)/ (TP + TN + FP+ FN)$$

Where TP = True Positive; FN = False Negative; FP = False Positive; TN = True Negative.

- PRECISION: -Its primary purpose is to get above accuracy's constraints. The accuracy of a positive prediction is determined by its precision. It can be found by comparing the true positives, or predictions, to the total number of positive predictions. The formula can be used to calculate the precision.

  $$\text{Precision} = TP/ (TP + FP)$$

- RECALL/SENSITIVITY: -It is used to determine the proportion of actual positive values that were identified incorrectly. It can be calculated as the ratio of true positives, or predictions that come true, to all the positives. The formula can be used to calculate the recall.

  $$\text{Recall/Sensitivity} = TP/ (TP + FN)$$

- F1-SCORE: -It is a measurement method for evaluating a classification based on the harmonic mean of precision and recall. It is determined by looking at the test's recall and precision. With an F1 score of 1 representing the best and 0 representing the worst, it may be thought of as a weighted average of precision and recall. It gives a better measure of the incorrectly classified cases than the accuracy metric.

F1 Score = {2 × (Recall × Precision)}/(Recall + Precision)

The confusion matrix is automatically obtained by Python code using the cognitive learning library when running the algorithm code in Anaconda platform.

Here we make a table of Accuracy for all the algorithms (INDIAN DATA):

Machine Learning Algorithms	Accuracy
Logistics Regression	0.925
Decision Tree	0.907
Gradient Boosting Classifier	0.909
Random Forest	0.933
Bagging Classifier	0.921
K-NN Classifier	0.865
SVM Classifier	0.935

Table 2: Accuracy Table for all the algorithms

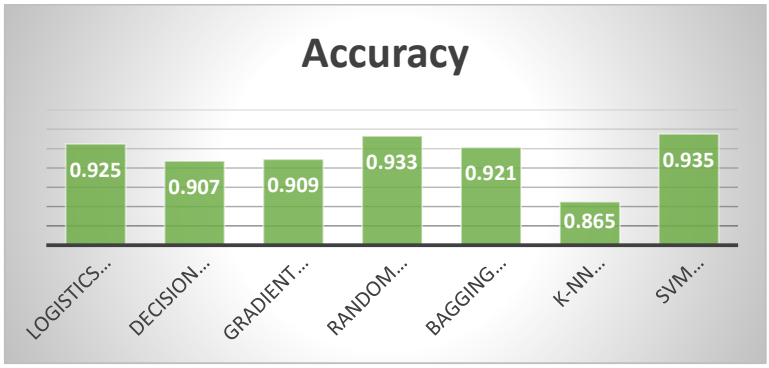


Figure 9: Obtained Percentage in Accuracy for all the algorithms

Figure 9 expresses the accuracies of these algorithms. As shown that Random Forest and SVM classifier are depicting the highest accuracy with more than 93%, next are Logistic Regression and Bagging Classifier with more than 92% accuracy and after the 4 algorithms there have Gradient Boosting Classifier with 91% accuracy and Decision Tree Classifier with 90% accuracy. At last K-NN is depicting the lowest accuracy with 85%. This indicates that Random Forest and Support Vector Machine (SVM) are better at correctly classifying fake news.

We make a table of Precision, Recall, f1-Score for all the algorithms:

Machine Learning Algorithms	Precision	Recall	f1-score
Logistics Regression	0.885	0.994	0.936
Decision Tree	0.909	0.924	0.916
Gradient Boosting Classifier	0.862	0.994	0.924
Random Forest	0.916	0.967	0.941
Bagging Classifier	0.913	0.948	0.930
K-NN Classifier	0.909	0.840	0.873
SVM Classifier	0.901	0.989	0.943

Table 3: Precision, Recall, f1-score for all the algorithms

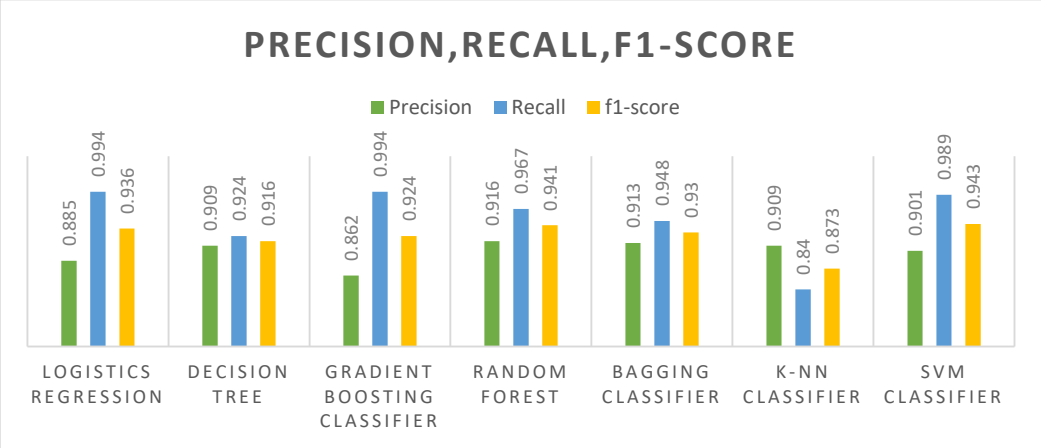
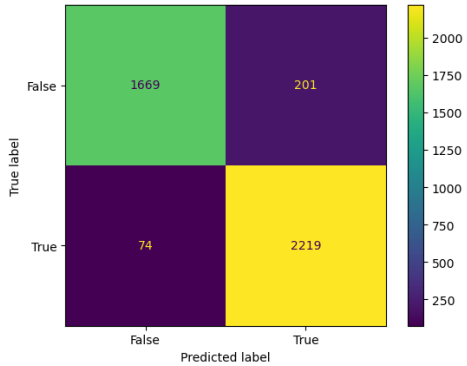


Figure 10: Obtained Percentage in Precision, Recall and f1-score for all the algorithms

From the above table we can say that precision score for Random Forest is 92%, which is better than other algorithms. So, it is indicated that Random Forest is more precise in identifying fake news than others.

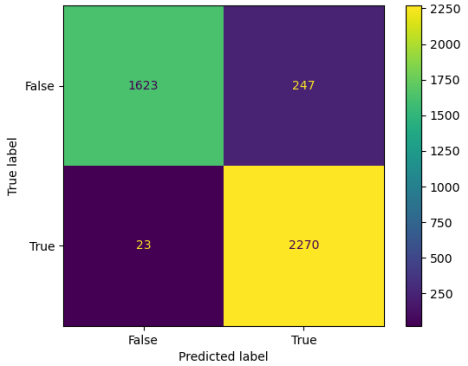
The Confusion Matrix of the Random Forest and Support Vector Machine (SVM) for Indian News are depict below:

Figure 11: CONFUSION MATRIX FOR RANDOM FOREST



The Random Forest algorithm predict that 201 fake news as true news and 74 true news as fake news, as well as 1669 fake news detect as fake and 2219 true news detects as real.

Figure 12: CONFUSION MATRIX FOR SVM



The SVM predict that 247 fake news as true news and 23 real news as fake news as well as 1623 fake news detect as fake and 2270 true news detect as true.

Here we make a table of Accuracy for all the algorithms (AMERICAN DATA):

Machine Learning Algorithms	Accuracy
Logistics Regression	0.980
Decision Tree	0.997
Gradient Boosting Classifier	0.996
Random Forest	0.984
Bagging Classifier	0.998
K-NN Classifier	0.612
SVM Classifier	0.992

Table 4: Accuracy Table for all the algorithms

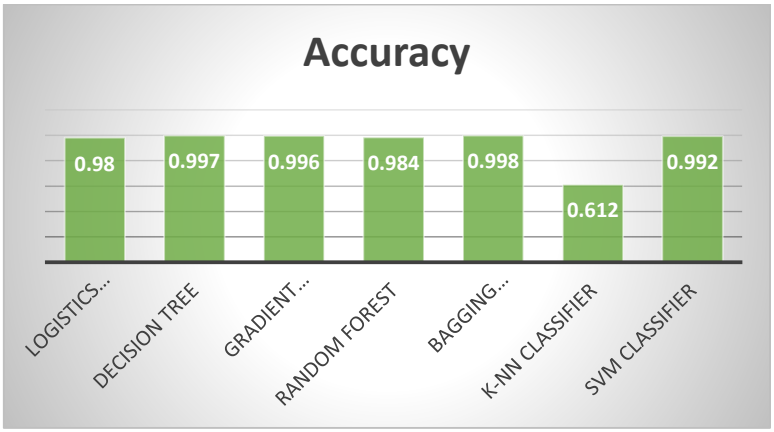


Figure 13: Obtained Percentage in Accuracy for all the algorithms

Figure 13 expresses the accuracies of these algorithms. As shown that most of the algorithms are depicting the accuracy with approx.100%, like, Decision Tree, Gradient Boosting, Bagging and SVM, next are Logistic Regression and Random Forest with more than 98% accuracy and at last K-NN is depicting the lowest accuracy with 61%. This indicates that Gradient Boosting, Bagging, Decision Tree and Support Vector Machine (SVM) are better at correctly classifying fake news.
We make a table of Precision, Recall, f1-Score for all the algorithms:

Machine Learning Algorithms	Precision	Recall	f1-score
Logistics Regression	0.980	0.986	0.983
Decision Tree	0.997	0.998	0.997
Gradient Boosting Classifier	0.995	0.998	0.927
Random Forest	0.980	0.992	0.986
Bagging Classifier	0.997	0.998	0.998
K-NN Classifier	0.939	0.353	0.513
SVM Classifier	0.993	0.993	0.993

Table 5: Precision, Recall, f1-score for all the algorithms

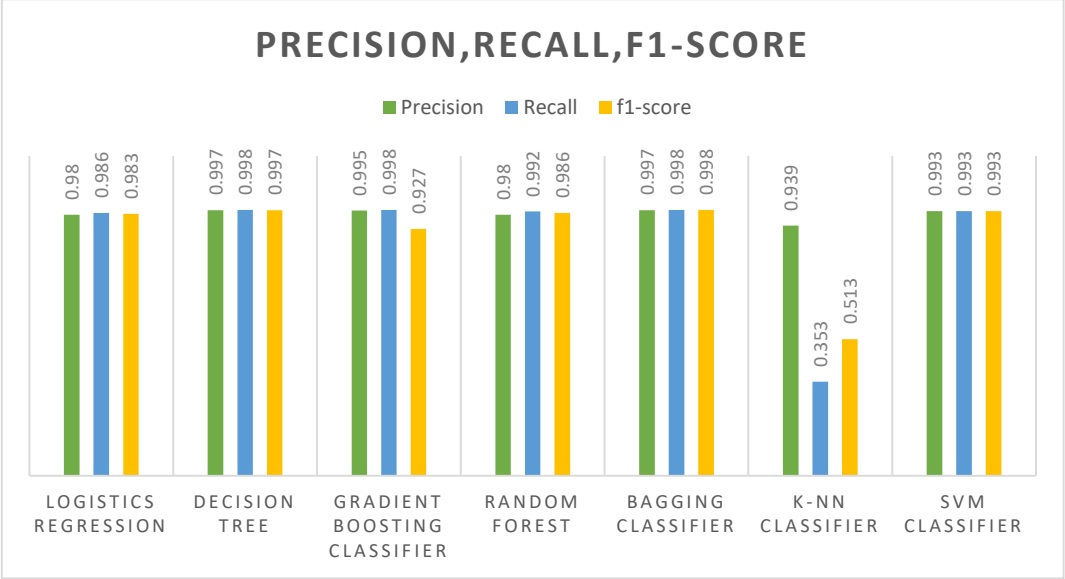
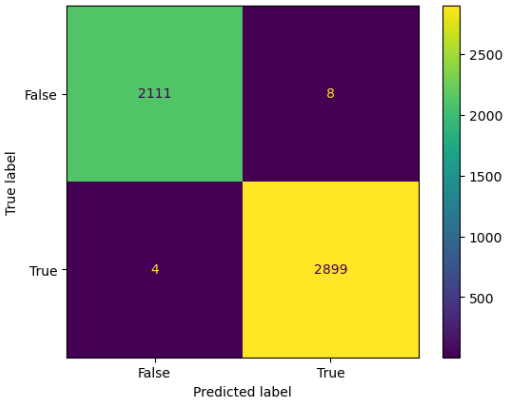


Figure 14: Obtained Percentage in Precision, Recall and f1-score for all the algorithms

From the above table we can say that precision score for Decision Tree and Bagging Classifier are approx. 100%, which is better than other algorithms. So, it is indicated that Decision Tree, Bagging Classifier are more precise in identifying fake news than others.

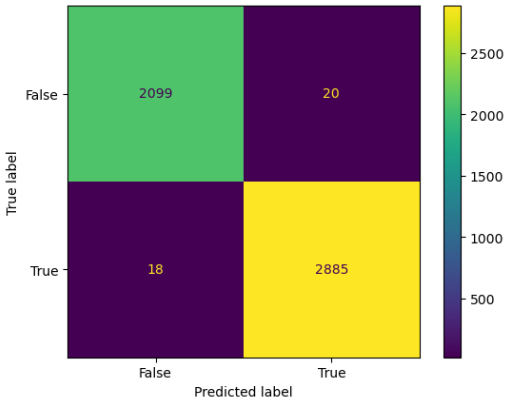
The Confusion Matrix of the Decision Tree, Gradient Boosting, Bagging Classifier and Support Vector Machine (SVM) for American News are depict below:

Figure 15:CONFUSON MATRIX FOR DECISION TREE



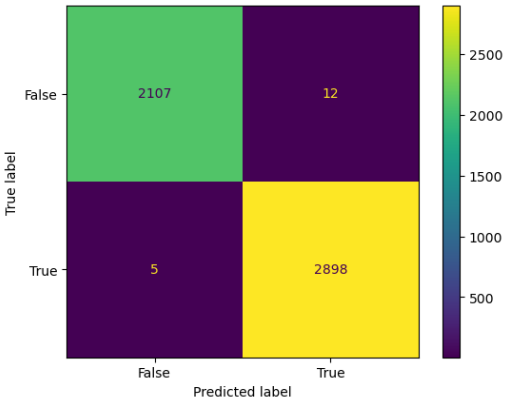
The Decision Tree algorithm predict that 8 fake news as true news and 4 true news as fake news , as well as 2111 fake news detect as fake and 2899 true news detects as true.

Figure 16: CONFUSION MATRIX FOR SVM



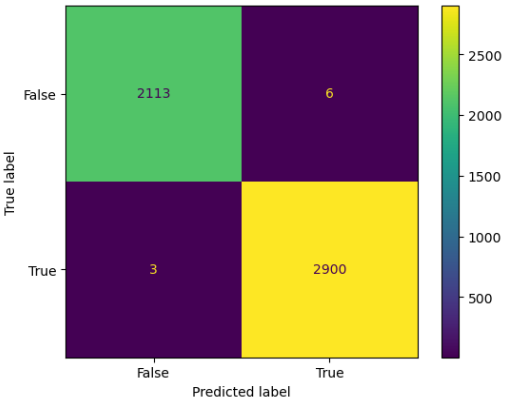
The SVM predict that 20 fake news as true news and 18 true news as fake news as well as 2099 false news detect as false and 2885 true news detect as true.

Figure 17: CONFUSION MATRIX OF GRADIENT BOOSTING



The Gradient Boosting algorithm predict that 12 fake news as true news and 5 true news as fake news, as well as 2107 false news detect as fake and 2898 true news detects as true.

Figure 18: CONFUSION MATRIX OF BAGGING



The Bagging Classifier predict that 6 fake news as true news and 3 true news as fake news, as well as 2113 fake news detect as fake and 2900 true news detects as true.

We make a model where we detect the news are true or fake. In our algorithm we assigned some different weights to the different models based on the accuracy of the models. We assigned the weight in such a way that the sum of the weights is 1.

```
#Model Testing
def output_lable(n):
    if n == 0:
        return "Fake News"
    elif n == 1:
        return "Not A Fake News"

def manual_testing(news):
    testing_news = {"Statement":[news]}
    new_def_test = pd.DataFrame(testing_news)
    new_def_test["Statement"] = new_def_test["Statement"].apply(wordopt)
    new_x_test = new_def_test["Statement"]
    new_xv_test = vectorization.transform(new_x_test)
    pred_LR = LR.predict(new_xv_test)
    pred_DT = DT.predict(new_xv_test)
    pred_GBC = GBC.predict(new_xv_test)
    pred_RFC = RFC.predict(new_xv_test)
    pred_BC = BC.predict(new_xv_test)
    pred_KNN = KNN.predict(new_xv_test)
    pred_SV = SV.predict(new_xv_test)

    data={'Method':['LR Prediction','DT Prediction','GBC Prediction','RFC Prediction','BC Prediction','KNN Prediction','SV Predic
          [pred_LR[0],pred_DT[0],pred_GBC[0],pred_RFC[0],pred_BC[0],pred_KNN[0],pred_SV[0]]}

    return data

    #return print("\n\nLR Prediction: {} \nDT Prediction: {} \nGBC Prediction: {} \nRFC Prediction: {} \nBC Prediction: {} \nKNN



news = str(input())
output_r=manual_testing(news)
```

```
PM Modi wishes Russian President Vladimir Putin on 68th birthday.
```

```
#Make Dataframe of the outputs
DF=pd.DataFrame(output_r)
```

```
#Make sum of the outputs and rechack the structure of the test data
sum=0
for index, value in DF['Report_Bin'].items():
    sum=sum+value


print(sum)


if sum>=3:
    print("Not a Fake News")
else:
    print("Fake News")
```

```
4
Not a Fake News
```

```
#Inputs some weights and check the structure of the test data
weight=np.array([0.160,0.150,0.150,0.170,0.150,0.05,0.170])

DC=np.dot(weight,DF['Report_Bin'])

if DC>=0.11:
    print("Not a Fake News")
else:
    print("Fake News")
```

```
Not a Fake News
```

Figure 19: Detecting The news are Fake or Not

6 CASE STUDY:

In this project we also work with Bengali dataset we have taken two dataset one is fake news dataset and other one is true news dataset of Bangladesh from the online website https://www.kaggle.com/datasets/cryptexcode/banfakenews/code.The datasets include 1300 fake news and 7203 true news. We done this project in the online platform colab. In this project we are using different packages and to load and read the data set we are using pandas. By using pandas, we can read the .xslx file of the dataset then I display the first few datapoints of our datasets and we can display the shape of the dataset. we can divide the dataset into two parts training and testing, when we use supervised learning, it means we are labelling the data. By getting the testing and training data and labels we can perform different machine learning algorithms but before performing the predictions and accuracies, the data is need to be

preprocessing i.e. the null values which are not readable are required to be removed from the data set and the data is required to be converted into vectors by normalizing and tokening the data so that it could be understood by the machine.
we collect the data online, it sometimes contains the undesirable characters like stop words (i.e. অতএব', 'অথচ', 'অথবা', 'অনুযায়ী', 'অনেক', 'অনেকে', 'অনেকেই', 'অন্তত', 'অন্য', 'অবধি', 'অবশ্য', 'অর্থাত', 'আই', 'আগামী', 'আগে', etc), digits (i.e. ১২৩৪৫৬৭৮৯), etc. which creates hindrance while fake news detection. We have removed Stop Words through bnlp Library. Removing punctuation, special characters and filter words (i.e. অআইঈউঊঋএঐওঔকখগঘঙচছজঝঞটঠডঢণতথদধনপফবভমযরলশষসহড়ঢ়য় etc) does not drastically change the meaning of a text. By removing unnecessary features from our text, we can reduce complexity. We have endeavoured for Stemming through the bnlp library. After the pre-processing we done same things which are mention above in the methodology part. Here we make a table of Accuracy for all the algorithms: -

Machine Learning Algorithms	Accuracy
Logistics Regression	0.924
Decision Tree	0.898
Gradient Boosting Classifier	0.928
Random Forest	0.913
Bagging Classifier	0.924
K-NN Classifier	0.914
SVM Classifier	0.945

Table 6: Accuracy Table for all the algorithms

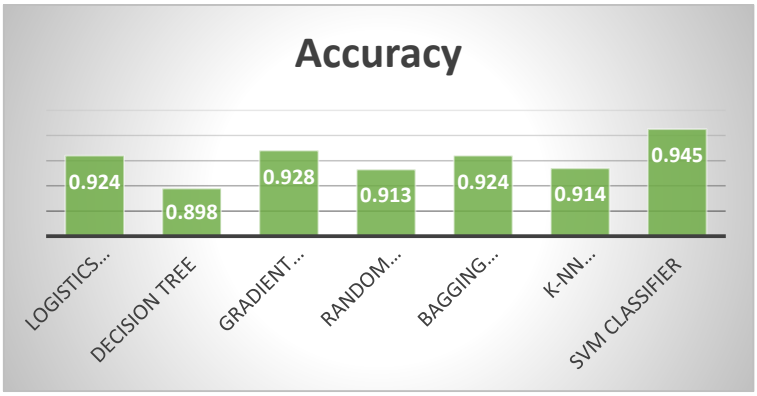


Figure 20: Obtained Percentage in Accuracy for all the algorithms

Figure 20 expresses the accuracies of these algorithms. As shown that SVM classifier is depicting the highest accuracy with more than 94%, next are Logistic Regression and Bagging Classifier and Gradient Boosting Classifier with more than 92% accuracy and after the 3 algorithms there have Random Forest and K-NN Classifier with 91% accuracy at last Decision Tree is depicting the lowest accuracy with 89%. This indicates that Support Vector Machine (SVM) is better at correctly classifying fake news.
We make a table of Precision, Recall, f1-Score for all the algorithms:

Machine Learning Algorithms	Precision	Recall	f1-score
Logistics Regression	0.924	0.991	0.957
Decision Tree	0.933	0.947	0.940
Gradient Boosting Classifier	0.929	0.991	0.959
Random Forest	0.907	0.999	0.951
Bagging Classifier	0.942	0.970	0.956
K-NN Classifier	0.921	0.982	0.951
SVM Classifier	0.950	0.987	0.968

Table 7: Precision, Recall, f1-score for all the algorithms

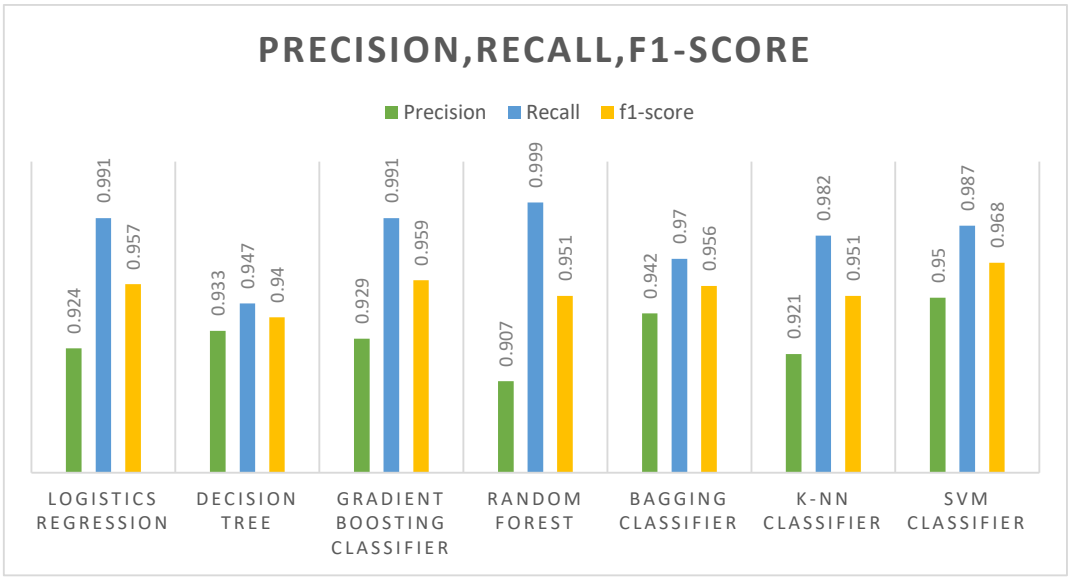
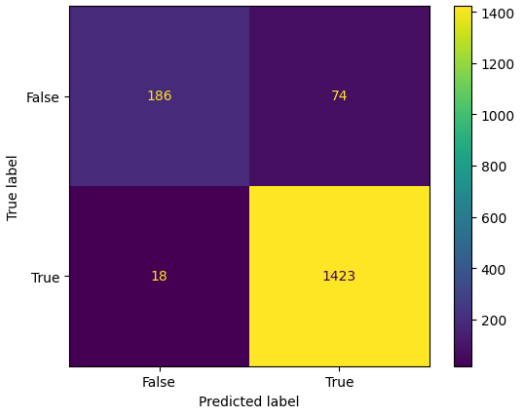


Figure 21: Obtained Percentage in Precision, Recall and f1-score for all the algorithms

From the above table we can say that precision score for SVM is 95%, which is better than other algorithms. So, it is indicated that SVM is more precise in identifying fake news than others.

The Confusion Matrix of the Support Vector Machine (SVM) is depicted below:

Figure 22: CONFUSION MATRIX OF SVM



The SVM predict that 74 false news as real news and 18 real news as fake news, as well as 186 false news detect as false and 1423 true news detects as real.

We make a model where we detect the news are true or fake. In our algorithm we assigned some different weights to the different models based on the accuracy of the models . We assigned the weight in such a way that the sum of the weights is 1.

```
[80] #Model Testing
     def output_lable(n):
         if n == 0:
             return "Fake News"
         elif n == 1:
             return "Not A Fake News"

     def manual_testing(news):
         testing_news = {"text":[news]}
         new_def_test = pd.DataFrame(testing_news)
         new_def_test["text"] = new_def_test["text"].apply(wordopt)
         new_x_test = new_def_test["text"]
         new_xv_test = Tf_Idf.transform(new_x_test)
         pred_LR = LR.predict(new_xv_test)
         pred_DT = DT.predict(new_xv_test)
         pred_GBC = GBC.predict(new_xv_test)
         pred_RFC = RFC.predict(new_xv_test)
         pred_BC = BC.predict(new_xv_test)
         pred_KNN = KNN.predict(new_xv_test)
         pred_SV = SV.predict(new_xv_test)

         data={'Method':['LR Prediction','DT Prediction','GBC Prediction','RFC Prediction','BC Prediction','KNN Prediction','SV Prediction'],'Report
               [pred_LR[0],pred_DT[0],pred_GBC[0],pred_RFC[0],pred_BC[0],pred_KNN[0],pred_SV[0]]}

         return data
```

```
[80]     #return print("\n\nLR Prediction: {} \nDT Prediction: {} \nGBC Prediction: {} \nRFC Prediction: {} \nBC Prediction: {} \nKNN Prediction: {}



     news = str(input())
     output_r=manual_testing(news)
```

ক্রিকেট বিশ্বের নতুন চমকের নাম আফগানিস্তান। কয়েক বছরে তাদের পারফরম্যান্স নজর কেড়েছে ক্রিকেট জগতের

```
[81] #Make Dataframe of the outputs
     DF=pd.DataFrame(output_r)
```

```
[82] #Make sum of the outputs and rechack the structure of the test data
     sum=0
     for index, value in DF['Report_Bin'].items():
         sum=sum+value


     print(sum)


     if sum>=1:
         print("Not a Fake News")
     else:
         print("Fake News")
```

```
7
Not a Fake News
```

```
#Inputs some weights and check the structure of the test data
weight=np.array([0.160,0.075,0.165,0.130,0.160,0.130,0.180])

DC=np.dot(weight,DF['Report_Bin'])

if DC>=0.14:
    print("Not a Fake News")
else:
    print("Fake News")
```

```
Not a Fake News
```

Figure 23: Detecting The news are Fake or Not for Bengali Data

7 CONCLUSION:

In the 21st century, the majority of the tasks are done online. Newspapers that were earlier preferred as hardcopies are now being substituted by applications like Facebook, Twitter, and news articles to be read online. An enormous number of people frequently use social media and the internet. When posting news on these platforms, there are no restrictions. Thus, some people start spreading false information about specific people or organizations by taking advantage of these platforms. This has the potential to ruin someone's reputation or have an impact on a company. Through fake news, the opinions of the people can also be changed for a political party. There must be a method for spotting these false reports. In addition to being used for other tasks, machine learning classifiers can also be used to identify false news.

This paper presents a novel model that uses machine learning algorithms to detect fake news. This model classifies user-provided data as either true or false. A variety of machine learning techniques—including Logistics Regression, Decision Trees, Random Forests, Support Vector Machines, Gradient Boosting Classifiers, K Nearest Neighbors, and Bagging Classifiers—are used to assess the model's performance. We used natural language processing (NLP) techniques to extract and represent features from a diverse dataset of news articles. This allowed the models to identify linguistic patterns that could be indicative of misinformation. Here we also make a model on Bengali Dataset to detect the fake and real news. For Bengali Dataset we used Bengali natural language processing (BNLP) techniques to extract and represent features from a dataset.

Our findings show encouraging performance across a variety of machine learning models. For Indian data, Random Forest and SVM showed the best accuracy in fake news detection tasks, while for American data, Gradient Boosting, Decision Tree, Bagging, and SVM showed the best accuracy in fake news detection tasks. Additionally, other machine learning techniques like K-NN and Logistics Regression produced results that were competitive as well, highlighting the significance of model aggregation and diversity in enhancing classification performance. For Bengali data, SVM showed the best accuracy in fake news detection tasks.

In conclusion, even though machine learning algorithms show promise in detecting fake news, the multifaceted nature of misinformation necessitates a comprehensive strategy that includes public awareness campaigns, legislative action, and technological innovation to protect the integrity of information and maintain democratic societies in the digital age.

8 REFERANCE:

[1] G. Agudelo, O. Parra, and J. Baron Velandia, “Raising a Model for Fake News Detection Using Machine Learning in Python,” pp. 596–604, 2018, doi: 10.1007/978-3-030-02131-3_52ï.

[2] J. Patel, M. Barreto, U. Sahakari, and Dr. S. Patil, “Fake News Detection with Machine Learning,” International Journal of Innovative Technology and Exploring Engineering, vol. 10, no. 1, pp. 124– 127, Nov. 2020, doi: 10.35940/IJITEE.A8090.1110120.

[3] M. Granik and V. Mesyura, "Fake news detection using naive Bayes classifier," 2017 IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON), Kiev, 2017, pp. 900-903

[4] P. Kulkarni, S. Karwande, R. Keskar, P. Kale, and S. Iyer, “Fake News Detection using Machine Learning,” ITM Web of Conferences, vol. 40, p. 03003, 2021, doi: 10.1051/itmconf/20214003003.

[5] M. L. Della Vedova, E. Tacchini, S. Moret, G. Ballarin, M. DiPierro and L. de Alfaro, "Automatic Online Fake News Detection Combining Content and Social Signals," 2018 22nd Conference of Open Innovations Association (FRUCT), Jyvaskyla, 2018, pp. 272-279.

[6] J. Vikram Tembhurne and M. M. Almin, “Mc-DNN: Fake News Detection Using Multi-Channel Deep Neural Networks,” Int J Semant Web Inf Syst, vol. 18, no. 1, doi: 10.4018/IJSWIS.295553.

[7] H. Gupta, M. S. Jamal, S. Madisetty and M. S. Desarkar, "A framework for real-time spam detection in Twitter," 2018 10th International Conference on Communication Systems & Networks (COMSNETS), Bengaluru, 2018, pp. 380-383

[8] Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu, ―Fake News Detection on social media: A Data Mining Perspective‖

[9] P. Goyal, S. Taterh, and A. Saxena, "Fake News Detection using Machine Learning: A Review," International Journal of Advanced Engineering, Management and Science (IJAEMS), vol. 7, no. 3, pp. 2454–1311, 2021, doi: 10.22161/ijaems.

[10]https://www.asianmedia.org.in/acj/home/reuters_institute#:~:text=India%20was%20also%20one%20of,%2C%20and%2049%25%20from%20print.

[11] f87824c6-4d9e-4f19-a27f-818a851b1de9

[12] IJERT_Fake_News_Detection_using_Machine%20

[13] https://covid-19-constitution.in/analyses/fake-news-misinformation-the-law-covid-19

9 APPENDIX:

SOFTWARE USES:
Here I use PYTHON Software.
PYTHON: Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages. Here we use many libraries.
We import the libraries.

Libraries	Function
pandas	Working with "relational" or "labelled" data can be simple and intuitive thanks to the Python module pandas, which offers quick, adaptable, and expressive data structures. It has functions for analysing, cleaning, exploring and manipulating data
numpy	The Python package NumPy is used to manipulate arrays. Additionally, it has matrices, Fourier transform, and functions for working in the area of linear algebra.
seaborn	A package called Seaborn uses Matplotlib as its foundation to plot graphs. In order to see random distributions, it will be used.
matplotlib	For the Python programming language & its NumPy numerical mathematics add-on, Matplotlib is a graphing library. It offers an object-oriented API for integrating charts into programs utilizing all-purpose GUI toolkits like Tkinter, python, Qt, or GTK
skylearn	t includes a variety of classification, regression, and clustering methods, such as support vector machines, random forests, gradient boosting, k-means, and DBSCAN, and is built to work with Python's NumPy and SciPy scientific and numerical libraries.

train_test_split ()	Machine Learning algorithms applicable to prediction-based algorithms and applications are evaluated using the train-test split. We can compare the output of our own machine-learning model to that of other machines using this quick and simple process.
accuracy score	This function computes subset accuracy in multilabel classification: the set of labels predicted for a sample must exactly match the corresponding set of labels in y true.
classification report	A classification report is used to assess the accuracy of a classification algorithm's predictions. How many predictions are correct and how many are incorrect? True Positives, False Positives, True Negative, and False Negatives are specifically utilized to predict the metrics of a classification report.
re	The functions in this module allow you to determine whether a given text fits a given regular expression, known as a regular expression.
string	You can use the Python library NLTK, or Natural Language Toolkit, for NLP. A large portion of the data you might be examining is unstructured and contains text humans can read. Preprocessing that data is necessary before you can programmatically evaluate it.

For importing the Dataset, we use the functions

Function	Description
pd.read_csv()/ pd.read_excel	Data Frame is read from a comma-separated values (csv or xlsx) file. csv is used for importing csv file and xlsx are used for importing excel file. Additionally enables optional file iteration or file segmentation.
head()	The top five rows of the data frame are shown by default when using Python's head method.
shape	Using the shape function, we can check how many rows and columns are present in the dataset.
Manual testing	The process of manually checking software for faults is known as manual testing. It requires a tester to act like an end user, using the majority of the application's capabilities to ensure proper behaviour.
pd.concat	We can add or merge two dataset using concat function.
drop	The drop() function deletes the given row or column. The drop() method eliminates the selected column by specifying the column axis (axis='columns').
lower()	An all-lower-case string is produced by the lower () function.
re.escape()	Automatically escaping each space.
re.sub()	The Python Regular Expressions (re) module contains the sub () method. All instances of the supplied pattern that match is replaced by the replace string in the returned string. We must import the re-module first before we can utilize this function.
string. Punctuation	A pre-initialized string called punctuation is utilized as a string constant. Python's string. Punctuation function returns all available punctuation
apply()	In Python, this method serves the same purpose as map (). It applies a function that is provided as input to a whole Data Frame. When working with tabular data, you need to define the axis your function should
TfidVectorizer	The TfidfVectorizer turns a set of raw documents into a TF-IDF feature matrix. Python implementation of Us with and Word2Vec word embeddings.
fit_transform	It is used to train data in order to scale it and learn the scaling parameters.