***Minor Project Report***

*On*

**FAKE\_NEWS\_DETECTION**

*Submitted in partial fulfilment of the requirements for the award of the degree*

*of*

**Bachelor of Technology**

in

**Computer Science & Engineering**

by

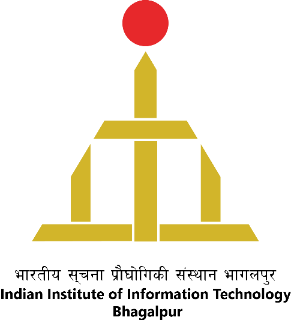
|  |  |
| --- | --- |
| **Ankush** | **(180101007)** |
| **Gautam Halder** | **(180101018)** |
| **Mayur Shakher Patel** | **(180101029)** |

Under the esteemed Supervision

of

**Dr. Dilip Kumar Choubey**

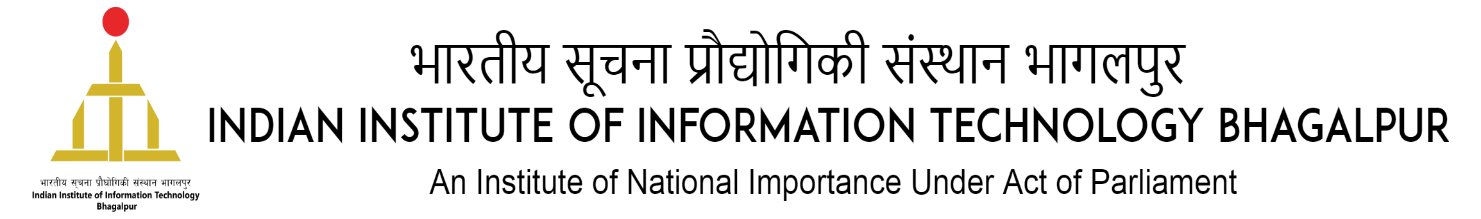
*(Assistant Professor IIIT Bhagalpur)*



**Department of Computer Science & Engineering**

Indian Institute of Information Technology Bhagalpur

**December, 2021**



# **Abstract**

In our modern era where the internet is ubiquitous, everyone relies on various online

resources for news. Along with the increase in the use of social media platforms like Facebook,

Twitter, etc. news spread rapidly among millions of users within a very short span of time. The

spread of fake news has far-reaching consequences like the creation of biased opinions to

swaying election outcomes for the benefit of certain candidates. Moreover, spammers use

appealing news headlines to generate revenue using advertisements via click-baits. In this paper, we aim to perform binary classification of various news articles available online with the help of concepts pertaining to Artificial Intelligence, Natural Language Processing and Machine Learning. We aim to provide the user with the ability to classify the news as fake or real and also check the authenticity of the website publishing the news.

# **Index**

[**Abstract** i](#_Toc90584002)

[**Index** ii](#_Toc90584003)

[**List of Figures** v](#_Toc90584004)

[**Chapter 1: Introduction** 6](#_Toc90584005)

[1.1 Overview 6](#_Toc90584006)

[1.2 Literature Survey 6](#_Toc90584007)

[1.3 Motivation 6](#_Toc90584008)

[1.4 Objective 7](#_Toc90584009)

[**Chapter 2: Project Initialization** 8](#_Toc90584010)

[2.1 Characteristics Of Fake News. 8](#_Toc90584011)

[2.2 Proposed Framework. 8](#_Toc90584012)

[2.3 Algorithms. 9](#_Toc90584013)

[2.4 Datasets 11](#_Toc90584014)

[2.5 Preformation Matrices 11](#_Toc90584015)

[**Chapter 3: Implementation & Result** 14](#_Toc90584016)

[3.1 Procedure 14](#_Toc90584017)

[3.1.1 Required Libraries 14](#_Toc90584018)

[3.1.2 Importing file from drive 14](#_Toc90584019)

[3.1.3 Inserting a column called "class" for fake and real news dataset to categories fake and true news. 15](#_Toc90584020)

[3.1.4 Removing last 10 rows from both the dataset, for manual testing 15](#_Toc90584021)

[3.1.5 Merging the manual testing dataframe in single dataset and save it in a csv file. 16](#_Toc90584022)

[3.1.9 Defining dependent and independent variable as x and y 18](#_Toc90584023)

[3.1.10 Splitting the dataset into training set and testing set 18](#_Toc90584024)

[3.1.11 Convert text to vectors 18](#_Toc90584025)

[3.2 Implementation of classifiers. 18](#_Toc90584026)

[3.2.1 Implementation Of Logistic Regression 18](#_Toc90584027)

[3.2.2 Implementation Of Decision Tree Classifier 19](#_Toc90584028)

[3.2.3 Implementation Of Random Forest Classifier 20](#_Toc90584029)

[3.3 Manual Testing 21](#_Toc90584030)

[3.3.1 Sample Input: 21](#_Toc90584031)

[3.3.2 Sample Output: 21](#_Toc90584032)

[**Chapter 4: Observation** 22](#_Toc90584033)

[4.1 Classification report 22](#_Toc90584034)

[4.2 Confusion Matrix 23](#_Toc90584035)

[4.3 Live prediction 24](#_Toc90584036)

[**Chapter 5: Conclusions** 25](#_Toc90584037)

[**References** 27](#_Toc90584038)

Learning Objectives/Minor Project Objectives

* Minor Projects are generally thought of to be reserved for college students looking to gain experience in a particular field. However, a wide array of people can benefit from Training Minor Projects in order to receive real world experience and develop their skills.
* An objective for this position should emphasize the skills you already possess in the area and your interest in learning more.
* Minor Projects are utilized in a number of different career fields, including architecture, engineering, healthcare, economics, advertising and many more.
* Some Minor Project is used to allow individuals to perform scientific research while others are specifically designed to allow people to gain first-hand experience working.
* Utilizing Minor Projects is a great way to build your resume and develop skills that can be emphasized in your resume for future jobs. When you are applying for a job, make sure to highlight any special skills or talents that can make you stand apart from the rest of the applicants so that you have an improved chance of landing the position.

# **List of Figures**

Figure 1: System Architecture…………………………………………………………………8

Figure 2: Confusion Matrix …………………………………………………………………..11

Figure 3: True dataset structure………………………………………………………………14

Figure 4: Fake dataset structure…………………………………………………………….. 15

Figure 5: Merged dataset structure ..........................................................................................16

Figure 6: Manual dataset structure……………………………………………………………17

Figure 7 Testing dataset structure…………………………………………………………….18

Figure 8 Logistic Regression classification report…………………………………………...19

Figure 9 Decision Tree classification report………………………………………………… 20

Figure 10 Random Forest classification report…………………………………………… ….20

Figure 11 Confusion Matrix(LR)……………………………………………………………..21

Figure 12 Confusion Matrix(DT)…………………………………………………………….22

Figure 13 Confusion Matrix(RFC)…………………………………………………………...23

# **Chapter 1: Introduction**

### 1.1 Overview

As an increasing amount of our lives is spent interacting online through social media platforms, more and more people tend to hunt out and consume news from social media instead of traditional news organizations. To assist mitigate the negative effects caused by fake news (both to profit the general public and therefore the news ecosystem). It's crucial that we build up methods to automatically detect fake news broadcast on social media.

### 1.2 Literature Survey

There has been a rapid increase in the spread of fake news in the last decade, most prominently observed in the 2016 US elections. Such proliferation of sharing articles online that do not conform to facts has led to many problems not just limited to politics but covering various other domains such as sports, health, and also science. One such area affected by fake news is the financial markets, where a rumour can have disastrous consequences and may bring the market to a halt.

Our ability to take a decision relies mostly on the type of information we consume; our world view is shaped on the basis of information we digest. There is increasing evidence that consumers have reacted absurdly to news that later proved to be fake. One recent case is the spread of novel corona virus, where fake reports spread over the Internet about the origin, nature, and behaviour of the virus. The situation worsened as more people read about the fake contents online. Identifying such news online is a daunting task.

### 1.3 Motivation

The World Wide Web contains data in diverse formats such as documents, videos, and audios. News published online in an unstructured format (such as news, articles, videos, and audios) is relatively difficult to detect and classify as this strictly requires human expertise. However, computational techniques such as natural language processing (NLP) can be used to detect anomalies that separate a text article that is deceptive in nature from articles that are based on facts. Other techniques involve the analysis of propagation of fake news in contrast with real news. More specifically, the approach analyzes how a fake news article propagates differently on a network relative to a true article. The response that an article gets can be differentiated at a theoretical level to classify the article as real or fake. A more hybrid approach can also be used to analyze the social response of an article along with exploring the textual features to examine whether an article is deceptive in nature or not.

### 1.4 Objective

In this paper, we propose a solution to the fake news detection problem using the machine learning ensemble approach. Our study explores different textual properties that could 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 that are not thoroughly explored in the current literature. The ensemble learners have proven to be useful in a wide variety of applications, as the learning models have the tendency to reduce error rate by using techniques such as bagging and boosting. These techniques facilitate the training of different machine learning algorithms in an effective and efficient manner. We also conducted extensive experiments on real world publicly available datasets. The results validate the improved performance of our proposed technique using the 4 commonly used performance metrics (namely, accuracy, precision, recall, and F-1 score).

# **Chapter 2: Project Initialization**

### 2.1 Characteristics of Fake News.

They often have grammatical mistakes. They are often emotionally colored. They often try to affect readers’ opinion on some topics. Their content is not always true. They often use attention seeking words and news format and click baits. They are too good to be true. Their sources are not genuine most of the times

### 2.2 Proposed Framework.

The architecture of Static part of fake news detection system is quite simple and is done keeping in mind the basic machine learning process flow. The system design is shown below and self- explanatory. The main processes in the design are

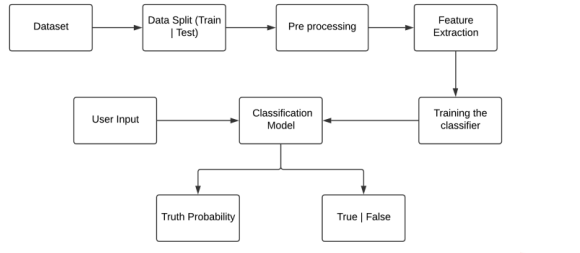
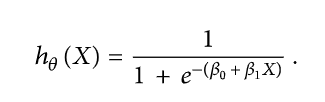


Figure 1. System Architecture

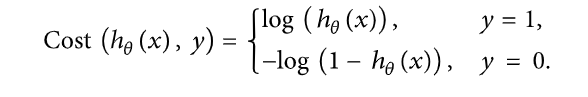
### 2.3 Algorithms.

##### 2.3.1 Logistic Regression

As we are classifying text on the basis of a wide feature set, with a binary output (true/false or true article/fake article), a logistic regression (LR) model is used, since it provides the intuitive equation to classify problems into binary or multiple classes. We performed hyperparameters tuning to get the best result for all individual datasets, while multiple parameters are tested before acquiring the maximum accuracies from LR model. Mathematically, the logistic regression hypothesis function can be defined as follows:



Logistic regression uses a sigmoid function to transform the output to a probability value; the objective is to minimize the cost function to achieve an optimal probability. The cost function is calculated as shown in



2.3.2 Decision Tree

**Decision Trees (DTs)** are a non-parametric supervised learning method used for [classification](https://scikit-learn.org/stable/modules/tree.html#tree-classification) and [regression](https://scikit-learn.org/stable/modules/tree.html#tree-regression). The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. A tree can be seen as a piecewise constant approximation. A Decision tree is a flowchart like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label.

2.3.3 Random Forest

Random forest (RF) is an advanced form of decision trees (DT) which is also a supervised learning model. RF consists of large number of decision trees working individually to predict an outcome of a class where the final prediction is based on a class that received majority votes. The error rate is low in random forest as compared to other models, due to low correlation among trees. A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.

### 2.4 Datasets

The datasets we used in this study are open source and freely available online. The data includes both fake and truthful news articles from multiple domains. The truthful news articles published contain true description of real-world events, while the fake news websites contain claims that are not aligned with facts.

The data source used for this project is dataset which contains 2 files with .csv format for true and fake news. Below is some description about the data files used for this project.

A full training dataset with the following attributes:

**id:** unique id for a news article

**title:** the title of a news article.

**text:** the text of the article.

**label:** a label that marks the article as potentially unreliable

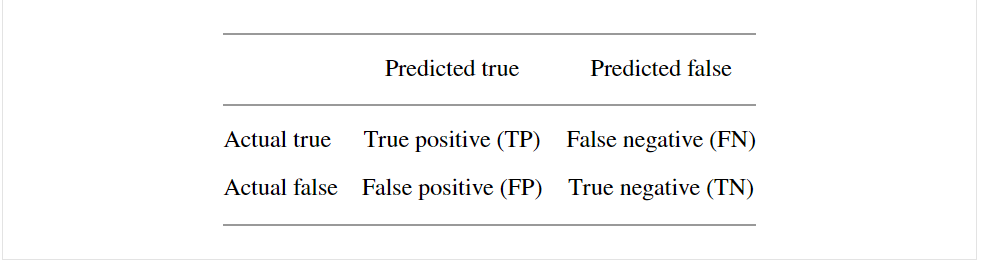
**1:** Fake news

**0:** Real news

* Fake.csv - This dataset contains a list of articles considered as "fake" news.
* True.csv - This dataset contains a list of articles considered as "real" news.
* Manual.csv - We also perform manual testing on dataset by removing 10 records from both True and Fake dataset and merging the records in single dataset and save it as manual.csv file.

### 2.5 Preformation Matrices

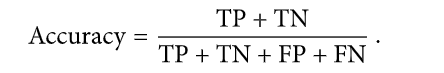
To evaluate the performance of algorithms, we used different metrics. Most of them are based on the confusion matrix. Confusion matrix is a tabular representation of a classification model performance on the test set, which consists of four parameters: true positive, false positive, true negative, and false negative (see Table 1).



###### Figure 2

##### 2.5.1 Accuracy

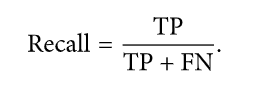
Accuracy is often the most used metric representing the percentage of correctly predicted observations, either true or false. To calculate the accuracy of a model performance, the following equation can be used:



In most cases, high accuracy value represents a good model, but considering the fact that we are training a classification model in our case, an article that was predicted as true while it was actually false (false positive) can have negative consequences; similarly, if an article was predicted as false while it contained factual data, this can create trust issues. Therefore, we have used three other metrics that take into account the incorrectly classified observation, i.e., precision, recall, and F1-score.

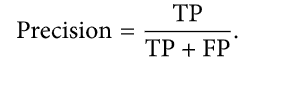
##### 2.5.2 Recall

Recall represents the total number of positive classifications out of true class. In our case, it represents the number of articles predicted as true out of the total number of true articles.



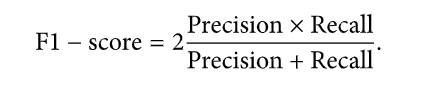
##### 2.5.3 Precision

Conversely, precision score represents the ratio of true positives to all events predicted as true. In our case, precision shows the number of articles that are marked as true out of all the positively predicted (true) articles:



##### 2.5.4 F1-score

F1-score represents the trade-off between precision and recall. It calculates the harmonic mean between each of the two. Thus, it takes both the false positive and the false negative observations into account. F1-score can be calculated using the following formula:



# **Chapter 3: Implementation & Result**

### 3.1 Procedure

So, we have to make a machine learning model and we have to train it, so that it can detect the Fake news.

### 3.1.1 Required Libraries

import pandas as pd

import numpy as np

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import classification\_report

import re

import string

### 3.1.2 **Importing file from drive**

from google.colab import drive

drive.mount('/content/drive')

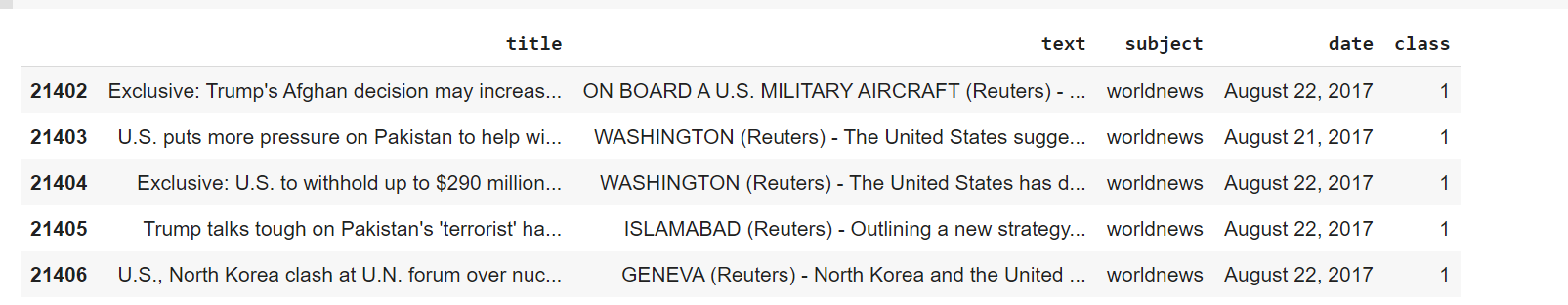
path\_fake='/content/drive/MyDrive/Colab Notebooks/Fake.csv'

path\_true='/content/drive/MyDrive/Colab Notebooks/True.csv'

df\_fake=pd.read\_csv(path\_fake)

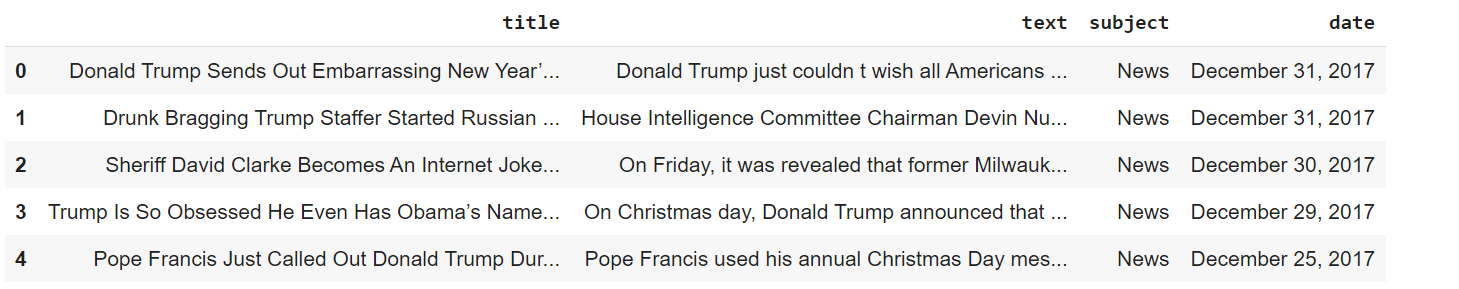
df\_true=pd.read\_csv(path\_true)

df\_true.tail(5)



###### Figure 3

df\_fake.head(5)



###### Figure 4

### 3.1.3 Inserting a column called "class" for fake and real news dataset to categories fake and true news.

df\_fake["class"] = 0

df\_true["class"] = 1

df\_fake.shape,df\_true.shape



### 3.1.4 **Removing last 10 rows from both the dataset, for manual testing**

df\_fake\_manual\_testing=df\_fake.tail(10)

for i in range(23480,23470,-1):

  df\_fake.drop([i], axis=0, inplace=True)  #delete the records row\_wise

df\_true\_manual\_testing=df\_true.tail(10)

for i in range(21416,21406,-1):

  df\_true.drop([i], axis=0, inplace=True)  #delete the records row\_wise

### 3.1.5 Merging the manual testing dataframe in single dataset and save it in a csv file.

df\_manual\_testing=pd.concat([df\_fake\_manual\_testing,df\_true\_manual\_testing],axis=0)

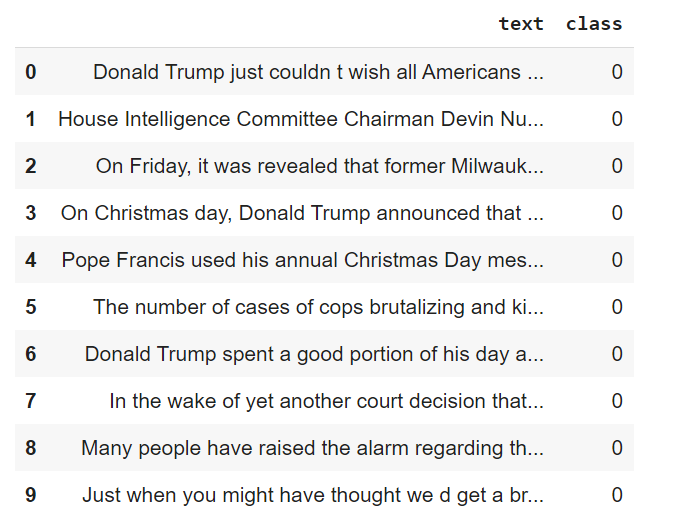
df\_manual\_testing.to\_csv("manual\_testing")

##### **3.1.6** Merging the main fake and true data frame and "title", "subject" and "date" columns is not required for detecting the fake news, so I am going to drop the columns.

df\_merge=pd.concat([df\_fake,df\_true],axis=0)

df=df\_merge.drop(["title","subject","date"],axis=1)

df.head(10)



###### Figure 5

##### 3.1.7 **Randomly shuffling the dataframe**

df = df.sample(frac = 1)

df.tail(10)



###### Figure 6

3.1.8 Creating a function to convert the text in lowercase, remove the extra space, special chr., ulr and links.

def word\_drop(text):

  text = text.lower()

  text = re.sub('\\[.\*?\\]', '', text)

  text = re.sub("\\W"," ", text)

  text = re.sub('https?://\\S+|www\\.\\S+', '', text)

  text = re.sub('<.\*?>+', '', text)

  text = re.sub('[%s]' % re.escape(string.punctuation), '', text)

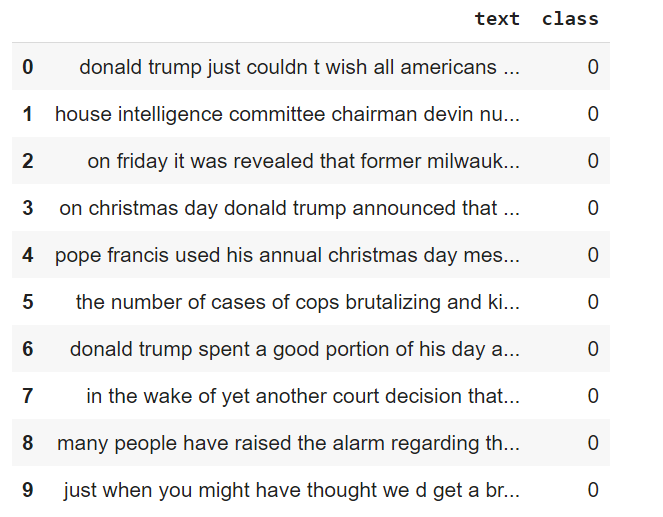
  text = re.sub('\\n', '', text)

  text = re.sub('\\w\*\\d\\w\*', '', text)

  return text

df["text"]=df["text"].apply(word\_drop)

df.head(10)



###### Figure 7

### 3.1.9 Defining **dependent and independent variable as x and y**

x=df["text"]

y=df["class"]

### 3.1.10 Splitting the dataset into training set and testing set

x\_test,x\_train,y\_test,y\_train=train\_test\_split(x,y,test\_size=.25)

### 3.1.11 **Convert text to vectors**

from sklearn.feature\_extraction.text import TfidfVectorizer

vectorization= TfidfVectorizer()

xv\_train=vectorization.fit\_transform(x\_train)

xv\_test=vectorization.transform(x\_test)

### 3.2 Implementation of classifiers.

### 3.2.1 **Implementation of Logistic Regression**

from sklearn.linear\_model import LogisticRegression

LR=LogisticRegression()

LR.fit(xv\_train,y\_train)

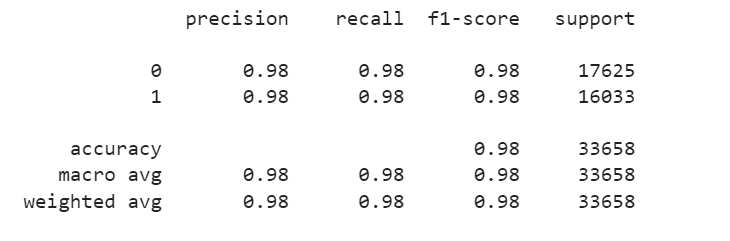
LogisticRegression()

LR.score(xv\_test,y\_test)

**0.978846039574544**

predict\_LR=LR.predict(xv\_test)

print(classification\_report(y\_test,predict\_LR))

****

###### Figure 8

### 3.2.2 **Implementation Of Decision Tree Classifier**

from sklearn.tree import DecisionTreeClassifier

DT=DecisionTreeClassifier()

DT.fit(xv\_train,y\_train)

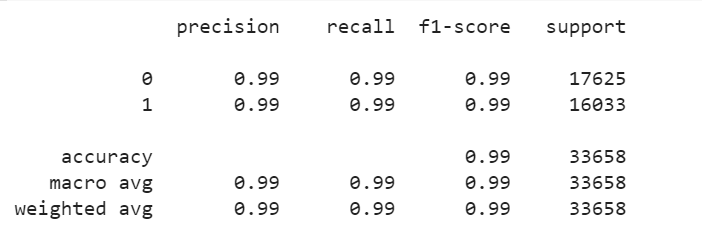
DecisionTreeClassifier()

DT.score(xv\_test,y\_test)

**0.9925723453562303**

predict\_DT=DT.predict(xv\_test)

print(classification\_report(y\_test,predict\_DT))

****

###### Figure 9

### 3.2.3 **Implementation Of Random Forest Classifier**

from sklearn.ensemble import RandomForestClassifier

RFC=RandomForestClassifier(random\_state=0)

RFC.fit(xv\_train,y\_train)

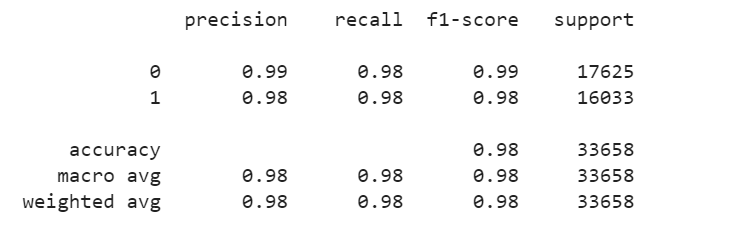
RandomForestClassifier(random\_state=0)

RFC.score(xv\_test,y\_test)

**0.9845801889595341**

predict\_RFC=RFC.predict(xv\_test)

print(classification\_report(y\_test,predict\_RFC))

****

###### Figure 10

### 3.3 Manual 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(word\_drop)

    new\_x\_test = new\_def\_test["text"]

    new\_xv\_test = vectorization.transform(new\_x\_test)

    pred\_LR = LR.predict(new\_xv\_test)

    pred\_DT = DT.predict(new\_xv\_test)

    pred\_RFC = RFC.predict(new\_xv\_test)

    return print("\n\nLR Prediction: {} \nDT Prediction: {} \nRFC Prediction: {}".format(output\_lable(pred\_LR[0]), output\_lable(pred\_DT[0]),  output\_lable(pred\_RFC[0])))

news = str(input())

manual\_testing(news)

### 3.3.1 Sample Input:

21413,LexisNexis withdrew two products from Chinese market,"LONDON (Reuters) - LexisNexis, a provider of legal, regulatory and business information, said on Tuesday it had withdrawn two products from the Chinese market in March this year after it was asked to remove some content. The issue of academic freedom in China hit the headlines this week after the leading British academic publisher, Cambridge University Press, said it had complied with a request to block online access to some scholarly articles in China. It later reversed its position. Earlier this year LexisNexis Business Insight Solutions in China was asked to remove some content from its database, LexisNexis said in a statement. In March 2017, the company withdrew two products (Nexis and LexisNexis Academic) from the Chinese market. LexisNexis is owned by information group Relx. ",worldnews,"August 22, 2017 "

### 3.3.2 Sample Output:

LR Prediction: Not A Fake News

DT Prediction: Not A Fake News

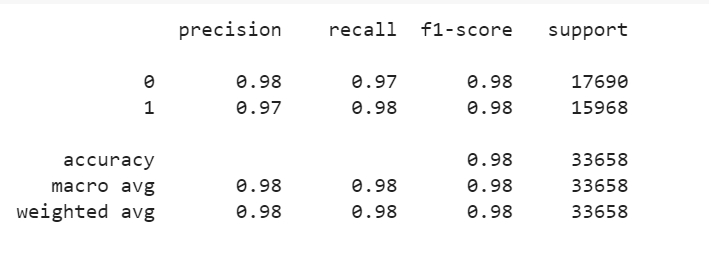
RFC Prediction: Not A Fake News

# **Chapter 4: Observation**

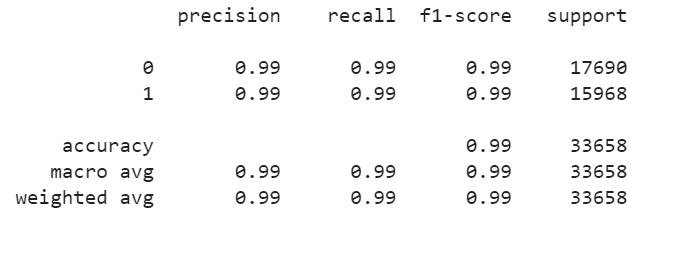
Our classifier work efficiently on the given dataset. We have successfully implemented our classifier algorithm on the datasets and the results that we get from our classifiers are as follows: -

### 4.1 Classification report

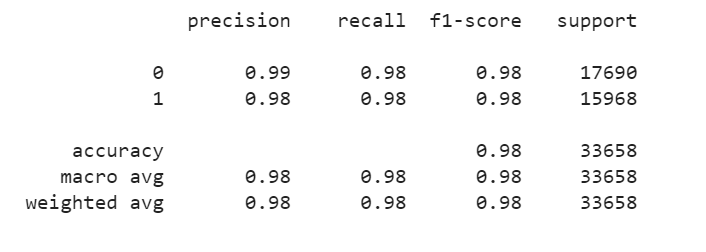
* Logistic Regression



* Accuracy: - 0.98
* Decision Tree



* Accuracy: - 0.99
* Random Forest



* Accuracy: -0.98

### 4.2 Confusion Matrix

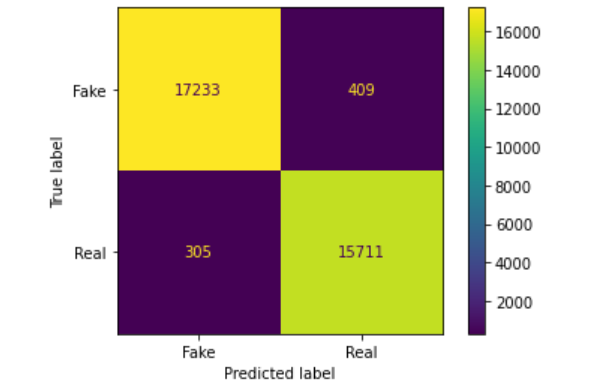
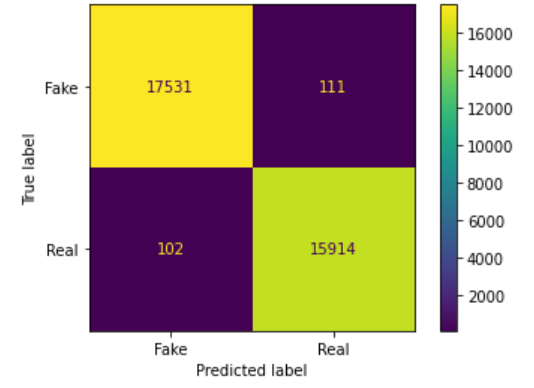
 

Figure 12. Logistic Regression Figure 13. Decision Tree

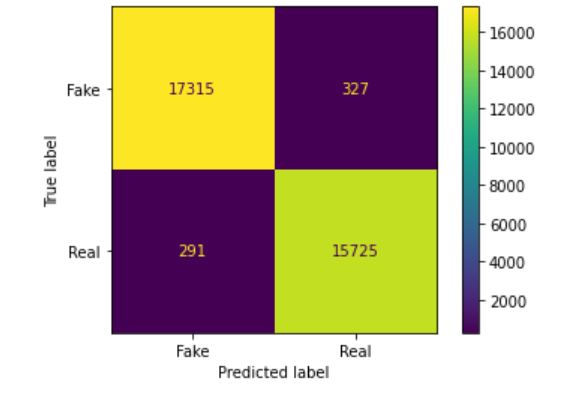


Figure 14. Random Forest Classifier

With the help Machine Learning we have created 3 prediction model which gives the accuracy above 90% and it covers all latest political and covid19 news. We have also generated manual.csv files from the given dataset and perform manual testing on it. We get correct output prediction on the given input samples.

### 4.3 Live prediction

Designed a web application which receive input text(news) and predict whether the news is fake or real.

Created a html page for frontend and flask for connection to localhost server. Choose the best classifier from the given three classifiers. The accuracy score of the classifiers are as follows: -

Logistic Regression - 91.55%

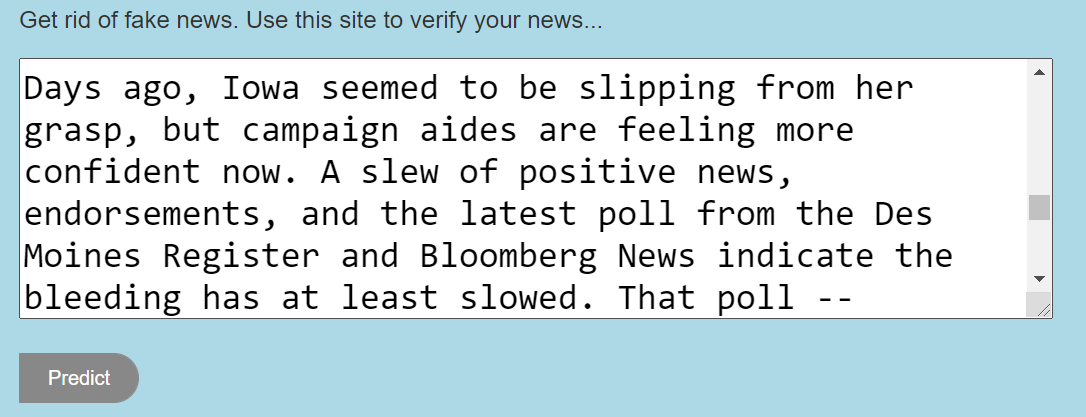
Decision Tree – 83.5%

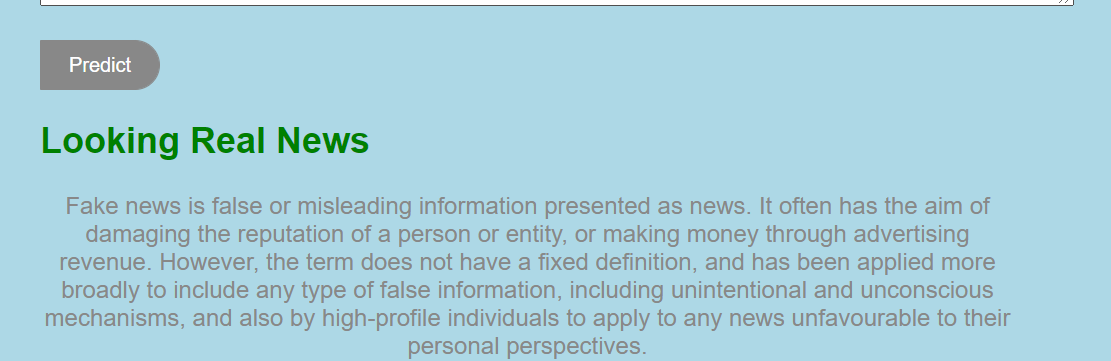
Random Forest Classifier -90.21%

We choose Logistic regression as it has better accuracy.

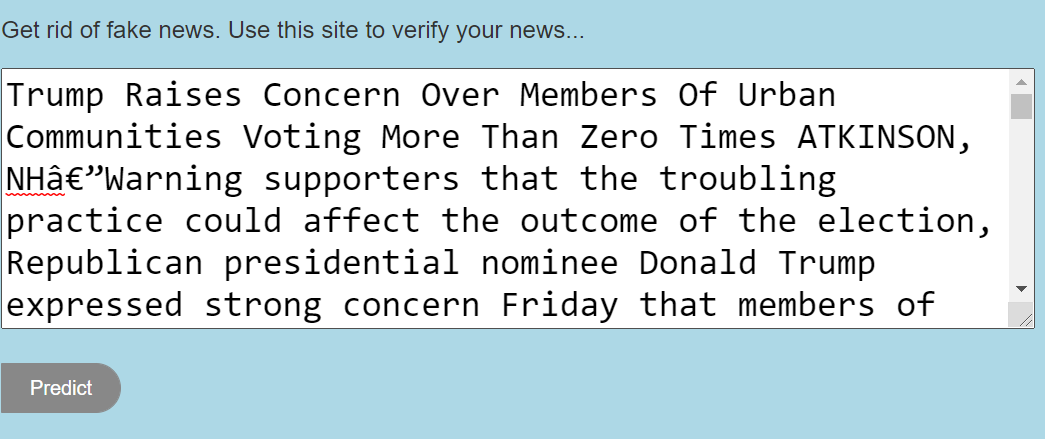
**Prediction**

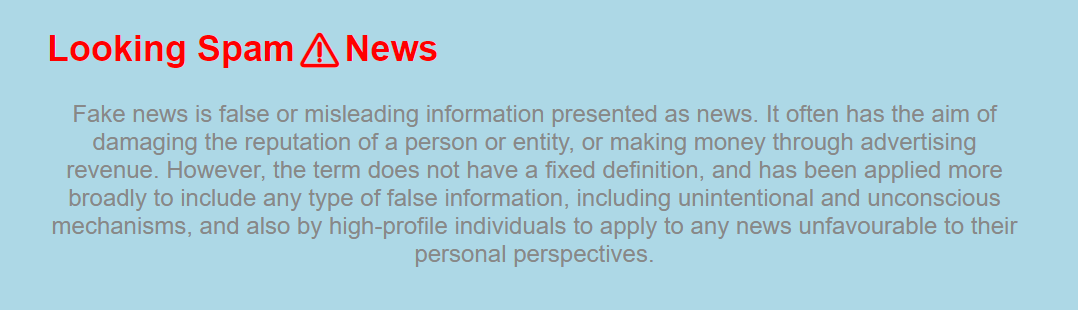
* **Real News**





* **Fake News**





# **Chapter 5: Conclusions**

The task of classifying news manually requires in-depth knowledge of the domain and expertise to identify anomalies in the text. In this research, we discussed the problem of classifying fake news articles using machine learning models and ensemble techniques. The data we used in our work is collected from the World Wide Web and contains news articles from various domains to cover most of the news rather than specifically classifying political news. The primary aim of the research is to identify patterns in text that differentiate fake articles from true news. The learning models were trained and parameter-tuned to obtain optimal accuracy. Some models have achieved comparatively higher accuracy than others. We used multiple performance metrics to compare the results for each algorithm.

Fake news detection has many open issues that require attention of researchers. For instance, in order to reduce the spread of fake news, identifying key elements involved in the spread of news is an important step. Graph theory and machine learning techniques can be employed to identify the key sources involved in spread of fake news. Likewise, real time fake news identification in videos can be another possible future direction

# **References**

1. **Dataset Availability-1** [**https://www.kaggle.com/clmentbisaillon/fake-and-real-news-dataset**](https://www.kaggle.com/clmentbisaillon/fake-and-real-news-dataset)
2. **Database\_Availability-2** [**https://github.com/Spidy20/Fake\_News\_Detection/blob/master/news.csv**](https://github.com/Spidy20/Fake_News_Detection/blob/master/news.csv)
3. **Tutorial** [**https://www.coursera.org/lecture/machine-learning/classification-wlPeP**](https://www.coursera.org/lecture/machine-learning/classification-wlPeP)
4. **Pandas Documentation** [**https://pandas.pydata.org/docs/user\_guide/index.html**](https://pandas.pydata.org/docs/user_guide/index.html%20)
5. **Scikit Learn Documentation**

[**https://scikitlearn.org/stable/supervised\_learning.html#supervised-learning**](https://scikitlearn.org/stable/supervised_learning.html#supervised-learning)

1. **NumPy Documentation** [**https://numpy.org/doc/**](https://numpy.org/doc/)
2. [**https://www.researchgate.net/publication/343916946\_Fake\_News\_Detection\_Using\_Machine\_Learning\_Algorithms**](https://www.researchgate.net/publication/343916946_Fake_News_Detection_Using_Machine_Learning_Algorithms)
3. [**https://data-flair.training/blogs/advanced-python-project-detecting-fake-news/**](https://data-flair.training/blogs/advanced-python-project-detecting-fake-news/)
4. [**https://www.hindawi.com/journals/complexity/2020/8885861/**](https://www.hindawi.com/journals/complexity/2020/8885861/)
5. **Code Link** [**https://github.com/ANKUSHSINGH-PAT/Fake\_News\_Detection**](https://github.com/ANKUSHSINGH-PAT/Fake_News_Detection)