

# **FAKE NEWS DETECTION USING MULTIPLE MACHINE LEARNING ALGORITHM**

**Submitted in partial fulfillment of the requirements for the award of  
Bachelor of Engineering degree in Computer Science and Engineering**

**By**

**Nikhil Joshi(38110368)  
Nikhil Singh (38110370)**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING  
SCHOOL OF COMPUTING**

## **SATHYABAMA**

**INSTITUTE OF SCIENCE AND  
TECHNOLOGY (DEEMED TO BE  
UNIVERSITY)**

**Accredited with Grade "A" by NAAC | 12B Status by UGC | Approved by  
AICTE**

**JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119**

## **MARCH-2022**



**SATHYABAMA**  
**INSTITUTE OF SCIENCE AND TECHNOLOGY**  
(DEEMED TO BE UNIVERSITY)



**Accredited with Grade “A” by NAAC**  
(Established under Section 3 of UGC Act, 1956)  
JEPPIAAR NAGAR, RAJIV GANDHI SALAI  
CHENNAI– 600119  
[www.sathyabama.ac.in](http://www.sathyabama.ac.in)

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**BONAFIDE CERTIFICATE**

This is to certify that this Project Report is the bonafide work of **Nikhil Joshi(Reg. No: 38110368)** and **Nikhil Singh(Reg. No: 38110370)** who carried out the project entitled “**Fake news detection using multiple Machine Learning algorithm** ” under my supervision from November 2021 to May 2022.

**Guide**

**Dr. KALAIARASI G M.E.,Ph.D.,**

**Head of the Department**

**Dr. L. LAKSHMANAN M.E., Ph.D.,**

**Dr. S. Vigneshwari M.E.,Ph.D.,**

**Submitted for Viva voce Examination held on\_\_\_\_\_**

**Internal Examiner**

**External Examiner**

## **DECLARATION**

I, **Nikhil Joshi and Nikhil Singh** hereby declare that the project report entitled “**Fake news detection using Multiple Machine Learning Algorithm**” done by us under the guidance of **Dr. Kalaiarasi G M.E.,Ph.D.**, at “Sathyabama Institute of Science and Technology (Deemed to be University), Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600119” is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering/Technology degree in Information Technology.

**DATE:**

**PLACE:**

**SIGNATURE OF THE CANDIDATE**

## **ACKNOWLEDGEMENT**

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph.D, Dean, School of Computing, Dr. S. Vigneshwari, M.E., Ph.D. and Dr. L. Lakshmanan, M.E., Ph.D., Heads of the Department of Computer Science and Engineering** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. Kalaiarasi G M.E.,Ph.D.**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

## **ABSTRACT**

Fake news has been in our society for a long time but with the introduction of social media, internet and mobile phones the spread of fake news has severely increased. Social media sites are being used to effectively spread misinformation and hoaxes around the world which not only causes people to change their thinking but also manipulates their opinions and decisions. In today's world it has become nearly impossible to detect if the given news is fake or real. With the arrival of the novel coronavirus-19 pandemic the propagation of fake news is now more than ever. In this time there is a need for something which can classify if a given news is real or not. In this article we aim to develop a model which, using some algorithms, determines if the given news is fake or not.

# TABLE OF CONTENTS

|                                                              |           |
|--------------------------------------------------------------|-----------|
| <b>ABSTRACT</b>                                              | <b>5</b>  |
| <b>TABLE OF CONTENTS</b>                                     | <b>6</b>  |
| <b>LIST OF FIGURES</b>                                       | <b>7</b>  |
| <b>LIST OF TABLES</b>                                        | <b>7</b>  |
| <b>ABBREVIATIONS</b>                                         | <b>8</b>  |
| <b>KEYWORDS</b>                                              | <b>10</b> |
| <b>1. Introduction</b>                                       | <b>11</b> |
| 1.1. Introduction                                            |           |
| 1.2. Aim and Objective                                       |           |
| 1.3. Motivation                                              |           |
| 1.4. Software Requirements Specification                     |           |
| <b>2. Literature Review</b>                                  | <b>14</b> |
| 2.1. Different approaches used                               |           |
| 2.2. Comparison of Existing methods with merits and demerits |           |
| <b>3. Methodology</b>                                        | <b>16</b> |
| 3.1. Models Used                                             |           |
| 3.2. Architecture diagram                                    |           |
| <b>4. Implementation</b>                                     | <b>19</b> |
| 4.1. Data collection and analysis                            |           |
| 4.2. Data Preprocessing                                      |           |
| 4.3. Tokenization                                            |           |
| 4.4. Feature Extraction                                      |           |
| 4.5. Splitting the dataset                                   |           |
| 4.6. Model Training and verification                         |           |
| <b>5. Libraries Used</b>                                     | <b>22</b> |
| 5.1. Numpy                                                   |           |
| 5.2. Pandas                                                  |           |
| 5.3. Sklearn                                                 |           |
| <b>6. Coding and Testing</b>                                 | <b>25</b> |
| <b>7. Conclusion and Enchantment</b>                         | <b>28</b> |
| <b>8. References</b>                                         |           |

## **LIST OF FIGURES**

- Architecture Diagram 18

## **LIST OF TABLES**

- Literature Review 14
- Accuracy Table 28

## **ABBREVIATIONS**

**LR** - Logistic Regression

**ML**- Machine Learning

**NLP**- Natural Language processing

**np** - Numpy

**pd** - Pandas

# 1.INTRODUCTION

## 1.1 INTRODUCTION:

As we spend more time connecting online through social media platforms, more people are seeking out and consuming news through social media outlets rather than traditional news organisations. [1] The reasons for this shift in consuming habits are built into the nature of those social media platforms: (i) consuming news on social media is typically more timely and less expensive than traditional journalism, such as newspapers or television; and (ii) sharing, discussing, and discussing the news with friends or other readers is easier on social media. For example, in 2016, 62 percent of adults in the United States received news on social media, compared to only 49 percent in 2012 [1].

It was also discovered that, as a key news source, social media currently exceeds television. Despite the advantages of social media, the quality of social media stories is lower than that of traditional news organisations. Large volumes of fake news, i.e. news pieces with purposely incorrect material, are created online for a variety of reasons, including financial and political gain, because it is inexpensive to supply news online and significantly faster and easier to distribute through social media. Over 1 million tweets have been linked to the false news "Pizzagate" scandal, according to estimates "by the end of the presidential election's first round Given the popularity of this new phenomena, "fake news" is a term that has to be defined "In 2016, the Macquarie dictionary named it the word of the year [2].

The widespread dissemination of fake news has the potential to harm both individuals and society. For starters, fake news has the potential to disrupt the news ecosystem's authenticity equilibrium; for example, during the 2016 presidential election in the United States, the most popular false news was even more widely shared on Facebook than the most widely accepted genuine mainstream news. Second, fake news is designed to induce customers to accept biased or incorrect information. Propagandists use fake news to disseminate

political ideas or influence. For example, according to some reports, Russia has developed fake accounts and social bots to spread misleading tales.

Third, false news alters how people interpret and respond to actual news. For example, some fake news was generated solely to instil distrust and confusion in people, making it difficult for them to tell what is true from what is not.

## **1.2 AIM AND OBJECTIVES**

The main objective behind the development and upgradation of existing projects are the following smart approaches:

- Be Aware of such articles while forwarding to others
- Reveal True stories
- Prevent from false crisis events
- Be Informative

## **1.3 MOTIVATION :**

Machine learning (ML) is a sort of artificial intelligence (AI) that allows software applications to improve their prediction accuracy without being expressly designed to do so.

In order to forecast new output values, machine learning algorithms use historical data as input.

The widespread dissemination of fake news has the potential to harm both individuals and society. For starters, false news has the potential to disrupt the news ecosystem's authenticity balance.

Understanding the truth of new information and its message can have a positive impact on society when used in conjunction with news detection.

## **1.4 SOFTWARE REQUIREMENTS SPECIFICATION:**

- Anaconda Navigator

- Jupyter
- Google Collab
- Windows 10
- Mouse,Keyboard, Laptop

## 2. LITERATURE REVIEW

### 2.1 DIFFERENT APPROACHES USED:

| Paper Id | Title                                                          | Source                                                                            | Task                                                                                                |
|----------|----------------------------------------------------------------|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------|
| 1        | Fake news detection using discourse segment structure analysis | 2020 10th International Conference on Cloud Computing, Data Science & Engineering | to propose and implement a method for automated deception detection                                 |
| 2        | Fake News Detection Using Deep Learning Techniques             | 2021 3rd International Congress on Human-Computer Interaction                     | Classification techniques like LR, RF, SVM NB and DNN for feature selection and extraction utilized |
| 3        | Fake News Detection by Decision Tree                           | IEEE Southeastcon                                                                 | Our model can identify the reality of the newest news on the website                                |

|   |                                                                         |                                                                                          |                                                                                                                  |
|---|-------------------------------------------------------------------------|------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------|
| 4 | Fake News Detection Using Deep Learning Techniques                      | International Conference on Advances in Information Technology                           | proposed system LR, RF, SVM NB and DNN classification techniques are utilized that will help to detect fake news |
| 5 | Fake News Detection on Social Media using K-Nearest Neighbor Classifier | International Conference on Advances in Computing and Communication Engineering (ICACCE) | a specific frame to predict fake news on social media.                                                           |
| 6 | A Deep Transfer Learning Approach for Fake News Detection               | 2020 International Joint Conference on Neural Networks (IJCNN)                           | fake information detection through stance detection                                                              |

## **2.2 COMPARISON OF EXISTING METHODS WITH MERITS AND DEMERITS:**

### **Limitations of linguistic analysis-based methods**

- Although this method is frequently regarded as superior to cue-based methods, it still fails to extract and fully exploit the content's rich semantic and grammatical information.
- E.g.: The N-gram approach is simple, however it cannot model more complicated contextual dependencies of the text.
- Syntactic features used alone are also less powerful than word-based n-grams and a superficial combination of the two would not be effective in capturing the complex interdependence.

### **Limitations of cue and feature-based methods**

- Various language cues necessitate the creation of a new cue set for each new situation, making it challenging to generalise cues and feature engineering methods across themes and domains.

### 3. METHODOLOGY

#### 3.1 MODELS USED:

- **Logistic Regression:**

The logistic function, also called the sigmoid function, was developed by statisticians to describe properties of population growth in ecology, rising quickly and maxing out at the carrying capacity of the environment. It's an S-shaped curve that can take any real-valued number and map it into a value between 0 and 1, but never exactly at those limits.  $\text{sigmoid}(Z) = 1 / (1 + e^{-Z})$  Hypothesis  $\Rightarrow Z = WX + B$   $h\Theta(x) = \text{sigmoid}(Z)$

- **Decision Tree Classification :**

Decision Tree is a supervised learning technique that may be used to solve both classification and regression problems, however it is most commonly employed to solve classification issues. Internal nodes represent dataset attributes, branches represent decision rules, and each leaf node provides the conclusion in this tree-structured classifier.

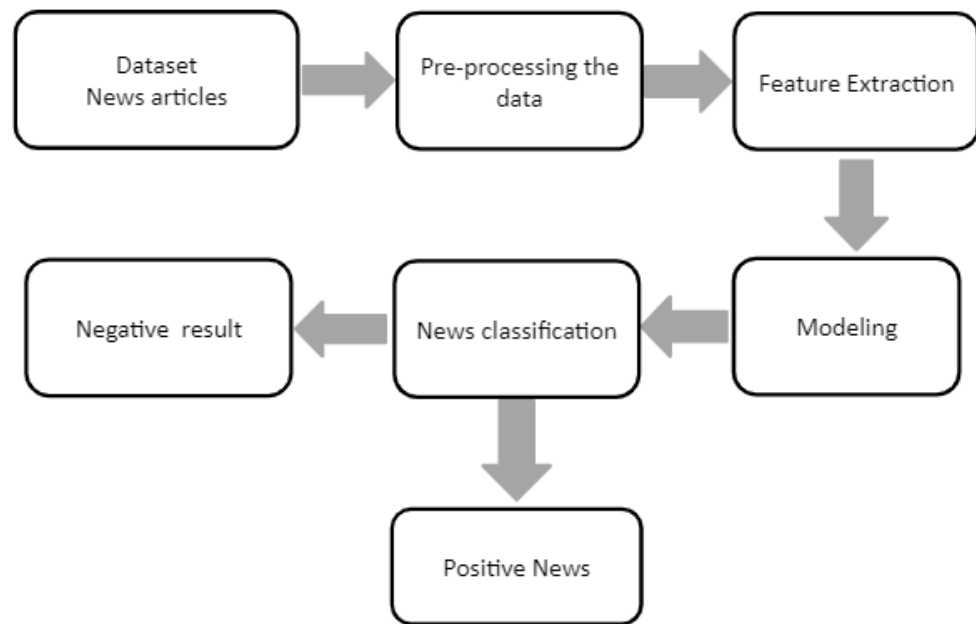
- **Gradient Boosting Classifier :**

Gradient Boosting is a popular boosting algorithm. In gradient boosting, each predictor corrects its predecessor's error. In contrast to Adaboost, the weights of the training instances are not tweaked; instead, each predictor is trained using the residual errors of its predecessor as labels.

- **Random Forest Classifier :**

Random Forest is a trademark term for an ensemble of decision trees. In Random Forest, we've collection of decision trees (so known as "Forest"). To classify a new object based on attributes, each tree gives a classification and we say the tree "votes" for that class. The forest chooses the classification having the most votes (over all the trees in the forest). The random forest is a classification algorithm consisting of many decision trees. It uses bagging and feature randomness when building each individual tree to try to create an uncorrelated forest of trees whose prediction by committee is more accurate than that of any individual tree. Random forest, like its name implies, consists of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model's prediction. The reason that the random forest model works so well is: A large number of relatively uncorrelated models (trees) operating as a committee will outperform any of the individual constituent models.

### 3.2 ARCHITECTURE DIAGRAM



ARCHITECTURE DIAGRAM

## **4. IMPLEMENTATION**

### **4.1 DATA COLLECTION AND ANALYSIS**

We can get online news from different sources like social media websites, search engines, homepage of news agency websites or fact-checking websites. On the Internet, there are a few publicly available datasets for Fake news classification like BuzzFeed News, BS Detector etc.

### **4.2 DATA PREPROCESSING :**

In data preprocessing the next step is to clean up the data that already exists. It's critical to remember that we're working with a computer that we're educating to distinguish between the phoney and the real. We have textual data right now, but computers work with numbers. So first we must convert them to numbers, and then we must ensure that we are only converting those texts that are required for comprehension.

The first stage in the cleaning pipeline is to see whether there are any superfluous symbols in the dataset that need to be removed. Web addresses or any other referring symbol(s), such as at(@) or hashtags, could be used.

After we've removed that, we'll move on to the following step: removing the other symbols, which are the punctuations. When we think about it, punctuation has no discernible impact on our ability to comprehend the truth of a particular piece of news. When there are a lot of punctuations, for example, excess of exclamations, it's conceivable that the news isn't true.

However, those are unusual circumstances that would necessitate a rule-based approach. As a result, we'll be eliminating the punctuation marks for our fake news detecting effort.

### **4.3 TOKENIZATION:**

The next step is to tokenize the words and stem the word to its core. Tokenization is the process of converting each sentence into a list of words or tokens.

### **4.4 FEATURE EXTRACTION**

The following step is critical. The process of converting tokens into numbers. Feature extraction is another name for this step. We're using the TF-IDF approach to extract and build features for our machine learning pipeline in our application.

The acronym TF-IDF stands for term frequency-inverse document frequency. We scoop information about the dataset via its frequency of terms as well as the frequency of terms across the full dataset, or collection of documents, as the name suggests.

By combining the TF and IDF data, TF-IDF may be simply determined. Simple ratios are used in both calculations.

$TF = \text{total number of terms} / \text{number of times the term appears in the document}.$

$IDF = \log \text{ of } (\text{total number of documents} / \text{number of documents containing the phrase})$

The TF-IDF conversion now has two Python implementations. The TF-IDF vectoriser comes first, followed by the TF-IDF transformer. The transformer, on

the other hand, requires a bag-of-words implementation before the transformation, but the vectoriser merges both stages into one. We might also use the count vectoriser, which is a simple bag-of-words implementation. However, the TF-IDF would be more appropriate for this dataset.

#### **4.5 SPLITTING THE DATASET:**

It can be achieved by using sklearn's preprocessing package and importing the train test split function.

#### **4.6 MACHINE LEARNING MODEL TRAINING AND VERIFICATION:**

In this step we import different types of classification models and feed the model the training data and wait for the model to be trained and after the training of the model we use the accuracy matrix and the test data to test the accuracy of each model.

We repeat the same process for each of the 4 classification models and get the accuracy score of the models.

## **5.LIBRARIES USED: NUMPY , PANDAS, SCIKIT-LEARN (SKLEARN)**

### **5.1 NUMPY :**

Numpy is an array processing library that may be used to do a wide range of tasks. It comes with a high-performance multidimensional array object and tools to manipulate it. For scientific computing, it is the most important Python module. In addition to its obvious scientific applications, Numpy can be used as a multi-dimensional container of common data.

Some of the following features of Numpy are:

- Numpy Array is a fast and space-efficient multidimensional array with vectorized arithmetic operations and advanced broadcasting features.
- Standard mathematical functions can be utilised to conduct rapid operations on large arrays of data without the need for loops.
- Reading/writing array data to disc and working with memory-mapped files
- Toolkits for integrating C, C++, and Fortran code Capabilities in linear algebra, random number generation, and the Fourier transform

### **5.2 PANDAS :**

Pandas is a Python package that allows you to work with large data sets. It offers tools for data analysis, cleansing, exploration, and manipulation.

Wes McKinney came up with the name "Pandas" in 2008, which refers to both "Panel Data" and "Python Data Analysis."

Pandas provides you with data-related responses. Like:

- Is there a link between two or more of the columns?
- What is the average value of something?
- What is the maximum value?
- What is the minimum value?

Pandas can also delete rows that are no longer relevant or contain incorrect values, such as empty or NULL values. This is referred to as data cleaning.

### 5.3 SCIKIT-LEARN (SKLEARN)

In Python, Scikit-learn (Sklearn) is the most usable and robust machine learning library. It uses a Python consistency interface to give a set of efficient tools for machine learning and statistical modelling, such as classification, regression, clustering, and dimensionality reduction. NumPy, SciPy, and Matplotlib are the foundations of this package, which is mostly written in Python.

Almost all common supervised learning algorithms, such as Linear Regression, Support Vector Machine (SVM), Decision Tree, and others, are included in scikit-learn.

On the other hand, it includes all of the popular unsupervised learning algorithms, including clustering, factor analysis, PCA (Principal Component Analysis), and unsupervised neural networks.

Some of the important features of Sklearn are:

**Clustering** is a model for grouping data that hasn't been labelled.

**Dimensionality reduction** is a technique for lowering the amount of qualities in data so that it can be summarised, visualised, and features selected.

**Ensemble** approaches combine the predictions of numerous supervised models, as the name implies.

**Feature extraction** is a technique for extracting data features and defining properties in picture and text data.

**Feature selection** is a technique for identifying interesting qualities that can be used to build supervised models.

## **6. CODING AND TESTING :**

### **STEP 1 : MAKE NECESSARY IMPORTS**

```
import NumPy as np
import pandas as pd
import itertools
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import PassiveAggressiveClassifier
from sklearn.metrics import accuracy_score, confusion_matrix
```

### **STEP 2 : READ THE DATA INTO A DATA FRAME, AND GET THE SHAPE OF THE DATA .**

```
#Read the data
df=pd.read_csv('D:\\DataFlair\\news.csv')
#Get shape and head
df.shape
df.head()
```

### **STEP 3: GETTING THE LABELS FROM THE DATA FRAME**

```
#DataFlair - Get the labels
labels=df.label
labels.head()
```

### **STEP 4: SPLITTING THE DATASET**

```
# Split the dataset
x_train,x_test,y_train,y_test=train_test_split(df['text'], labels,
test_size=0.2, random_state=7)
```

### **STEP 5: INITIALIZE A TFIDFVECTORIZER**

Terms with a higher document frequency will be discarded

```
#Initialize a TfidfVectorizer
tfidf_vectorizer=TfidfVectorizer(stop_words= 'english' , max_df=0.7)
#Fit and transform train set, transform test set
tfidf_train=tfidf_vectorizer.fit_transform(x_train)
tfidf_test=tfidf_vectorizer.transform(x_test)
```

### **STEP 6: WE'LL INITIALIZE CLASSIFIER**

```
from sklearn.tree import DecisionTreeClassifier
DT = DecisionTreeClassifier()
DT.fit(xv_train, y_train)
pred_dt = DT.predict(xv_test)
DT.score(xv_test, y_test)
print(classification_report(y_test, pred_dt))
```

## OUTPUT SCREENSHOT

### 1. Logistic Regression

```
In [28]: from sklearn.linear_model import LogisticRegression
```

```
In [29]: LR = LogisticRegression()  
LR.fit(xv_train,y_train)
```

```
Out[29]: LogisticRegression()
```

```
In [30]: pred_lr=LR.predict(xv_test)
```

```
In [31]: LR.score(xv_test, y_test)
```

```
Out[31]: 0.9866310160427807
```

```
In [32]: print(classification_report(y_test, pred_lr))
```

|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0            | 0.99      | 0.99   | 0.99     | 5729    |
| 1            | 0.99      | 0.99   | 0.99     | 5491    |
| accuracy     |           |        | 0.99     | 11220   |
| macro avg    | 0.99      | 0.99   | 0.99     | 11220   |
| weighted avg | 0.99      | 0.99   | 0.99     | 11220   |

### 2. Decision Tree Classification

```
In [33]: from sklearn.tree import DecisionTreeClassifier
```

```
In [34]: DT = DecisionTreeClassifier()  
DT.fit(xv_train, y_train)
```

```
Out[34]: DecisionTreeClassifier()
```

```
In [35]: pred_dt = DT.predict(xv_test)
```

```
In [36]: DT.score(xv_test, y_test)
```

```
Out[36]: 0.995632798573975
```

```
In [37]: print(classification_report(y_test, pred_dt))
```

|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0            | 0.99      | 1.00   | 1.00     | 5729    |
| 1            | 1.00      | 0.99   | 1.00     | 5491    |
| accuracy     |           |        | 1.00     | 11220   |
| macro avg    | 1.00      | 1.00   | 1.00     | 11220   |
| weighted avg | 1.00      | 1.00   | 1.00     | 11220   |

### 3. Gradient Boosting Classifier

```
In [38]: from sklearn.ensemble import GradientBoostingClassifier
```

```
In [39]: GBC = GradientBoostingClassifier(random_state=0)
GBC.fit(xv_train, y_train)
```

```
Out[39]: GradientBoostingClassifier(random_state=0)
```

```
In [40]: pred_gbc = GBC.predict(xv_test)
```

```
In [41]: GBC.score(xv_test, y_test)
```

```
Out[41]: 0.9959893048128342
```

```
In [42]: print(classification_report(y_test, pred_gbc))
```

|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0            | 1.00      | 0.99   | 1.00     | 5729    |
| 1            | 0.99      | 1.00   | 1.00     | 5491    |
| accuracy     |           |        | 1.00     | 11220   |
| macro avg    | 1.00      | 1.00   | 1.00     | 11220   |
| weighted avg | 1.00      | 1.00   | 1.00     | 11220   |

### 4. Random Forest Classifier

```
In [43]: from sklearn.ensemble import RandomForestClassifier
```

```
In [44]: RFC = RandomForestClassifier(random_state=0)
RFC.fit(xv_train, y_train)
```

```
Out[44]: RandomForestClassifier(random_state=0)
```

```
In [45]: pred_rfc = RFC.predict(xv_test)
```

```
In [46]: RFC.score(xv_test, y_test)
```

```
Out[46]: 0.9914438502673797
```

```
In [47]: print(classification_report(y_test, pred_rfc))
```

|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0            | 0.99      | 0.99   | 0.99     | 5729    |
| 1            | 0.99      | 0.99   | 0.99     | 5491    |
| accuracy     |           |        | 0.99     | 11220   |
| macro avg    | 0.99      | 0.99   | 0.99     | 11220   |
| weighted avg | 0.99      | 0.99   | 0.99     | 11220   |

## 7.CONCLUSION AND FUTURE ENCHANTMENT :

We created 4 prediction models using Machine Learning that have an accuracy of above 90% which predicts if the given news is either fake or not. We intend to create our own dataset, which will be updated as new information becomes available in the future.

Accuracy Score of the different models:

| Classifier          | Accuracy |
|---------------------|----------|
| Logistic Regression | 98%      |
| Gradient Boosting   | 99%      |
| Random Forest       | 99%      |
| Decision Tree       | 99%      |

## 8. References

- [1] Li, Xiangyang, et al. "Exploring text-transformers in aaai 2021 shared task: Covid-19 fake news detection in english." arXiv preprint arXiv:2101.02359 (2021).
- [2] Kamal, Ojasv , Adarsh Kumar, and TejasVaidhya. "Hostility Detection in Hindi leveraging Pre-Trained Language Models." arXiv preprint arXiv:2101.05494 (2021).
- [3] Yang, Kai-Chou, Timothy Niven, and Hung-Yu Kao. "Fake news detection as natural language inference." arXiv preprint arXiv:1907.07347 (2019).
- [4] Das, SouryaDipta, AyanBasak, and Saikat Dutta. "A heuristic-driven ensemble framework for covid-19 fake news detection." arXiv preprint arXiv:2101.03545 (2021).
- [5] Vicario, Michela Del, et al. "Polarization and fake news: Early warning of potential misinformation targets." ACM Transactions on the Web (TWEB) 13.2 (2019): 1-22.
- [6] Qi, Peng, et al. "Exploiting multi-domain visual information for fake news detection." 2019 IEEE International Conference on Data Mining (ICDM). IEEE, 2019.
- [7] Mladenova, Tsvetelina, and Irena Valova. "Analysis of the KNN Classifier Distance Metrics for Bulgarian Fake News Detection." 2021 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA). IEEE, 2021.
- [8] Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).
- [9] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." Neural computation 9.8 (1997): 1735-1780.
- [10] Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems. 2017.
- [11] Radford, Alec, et al. "Improving language understanding by generative pre-training." (2018).
- [12] Sun, Yu, et al. "Ernie: Enhanced representation through knowledge integration." arXiv preprint arXiv:1904.09223 (2019).

- [13] Sun, Yu, et al. "Ernie 2.0: A continual pre-training framework for language understanding." Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 34. No. 05. 2020.
- [14] Ceron, Wilson, Mathias-Felipe de-Lima-Santos, and Marcos G. Quiles. "Fake news agenda in the era of COVID-19: Identifying trends through fact-checking content." Online Social Networks and Media 21 (2021): 100116.
- [15] Hamid, Abdullah, et al. "Fake news detection in social media using graph neural networks and NLP Techniques: A COVID-19 use-case." arXiv preprint arXiv:2012.07517 (2020).
- [16] Yuan, Chunyuan, et al. "Early detection of fake news by utilizing the credibility of news, publishers, and users based on weakly supervised learning." arXiv preprint arXiv:2012.04233 (2020).