Fake News detection using Python and Machine Learning

Mentor: Mr. Franklin Telfer
Dept of Electronics and Communication Engineering,
Rajalakshmi Institute of Technology, Chennai, India
franklintelfer@ritchennai.edu.in

Sathya Sivam L
Electronics and Communication Engineering
Rajalakshmi Institute of Technology, Chennai, India
sathyasivam.l.2021.ece@ritchennai.edu.in

Sarath Chandren S K
Electronics and Communication Engineering
Rajalakshmi Institute of Technology, Chennai, India
sarathchandren.s.k.2021.ece@ritchennai.edu.in

Santhosh R
Electronics and Communication Engineering
Rajalakshmi Institute of Technology, Chennai, India
santhosh.r.2021.ece@ritchennai.edu.in

***Abstract* — A fake news detection system is presented in this study because disinformation spreads quickly in today's digital environment and fake news has become a major problem. The goal of fake news detection is to distinguish between true news and false or misleading information. Machine learning algorithms are used in this procedure to examine and categorize news. To create reliable fake news detection systems, text preprocessing like data cleaning, feature extraction using the TF-IDF approach, and classification models (such as logistic regression, random forest, decision tree classifier, Gradient Boosting Classifier) are used. Logistic Regression and Random Forest Classifier use the "Intel Extension for Scikit Learn" to improve performance during training. The proposed system presents a viable approach to deal with fake news, promoting the spread of trustworthy information in the current media environment.**

Keywords—TF-IDF Vectorization, Intel Extension for Scikit-Learn, Logistic Regression, Random Forest Classifier, preprocessing.

I. INTRODUCTION

The expansion of the internet users and the quick uptake of social media platforms made it possible for the distribution of knowledge to reach unprecedented levels. Users of social media platforms are creating and disseminating more information than ever before, some of it false and unrelated to reality due to the widespread use of these platforms. 70% of traffic to news websites is generated by Facebook recommendations. Due to their potential to enable users to discuss, exchange, and debate topics like democracy, education, and health, these social media platforms are incredibly powerful and valuable. However, these platforms are also employed negatively for financial gain, and occasionally for slanted opinion formation and the dissemination of false information. We use some machine learning algorithms for classifying data, such as Decision trees, Logistic Regression, and Gradient Boost Classifiers, to identify fake news. We investigate "textual characteristics that can be used to distinguish fake contents from real" in our study. We train a variety of machine learning algorithms using a variety of techniques based on those properties and assess their effectiveness using datasets from the real world.

II. OBJECTIVE

The goal of fake news detection is to tell real information from fake information that is being spread on numerous channels, such as news websites, internet forums and social media. The term "fake news" describes intentionally false or misleading information that is presented as genuine news with the aim of misleading readers or swaying public opinion. "*The creation of trustworthy and precise tools for spotting and alerting erroneous material is the main objective of fake news detection*". Natural language processing (NLP), machine learning algorithms like Term Frequency - Inverse Document Frequency vectorization (TFIDF), N-grams vectorization, etc., model training, evaluation, and prediction of the trained data are frequently used in this process. By analyzing the content, sources, and contextual information, these methods aim to assess the credibility and reliability of the news.

The importance of detecting fake news lies in upholding the integrity of information, encouraging critical thought, and averting any potential harmful effects brought on by the dissemination of false information. People can make better judgements, lessen the impact of false information, and maintain a more accurate grasp of current events by identifying and alerting users to the presence of fake news. It can be challenging to identify fake news on social media and other platforms since disinformation tactics are dynamic and there is a lot of data to analyse. The data collection provided includes old news that may not be current. Effective data analysis requires a frequently updated data set.

III. OUTCOMES

Fake news detection aims to identify and categorize news articles as real or deceptive. The outcomes include classifying news, providing confidence scores, warning or flagging users on false information. These outcomes help users distinguish reliable sources, assess credibility, and combat the spread of misinformation.

A. *Classification*

Fake news detection algorithm typically classifies the news data provided as input into categories like “Real News”, “Fake News”. This classification algorithm helps users to distinguish between reliable and unreliable information.

B. *Prediction Scores*

This fake news detection system uses some classifiers like Logistic Regression, Decision Tree Classifier to predict the accuracy score of the algorithm. This information indicates the system level of certainty regarding the authenticity of the news.

Performance is improved by using the Intel Extension for Scikit Learn, which gradually reduces the run time of the classifiers for large data sets.

IV. CHALLENGES

A. *Scalability and Coverage*

Building comprehensive and diverse data sets that encompass a wide range of fake news scenarios, topics, and languages is a challenging task. Collecting a large-scale, representative data set that covers various sources, domains, and regions requires significant resources and effort. Ensuring the data set is up-to-date and reflects the rapidly evolving landscape of fake news is an ongoing challenge...

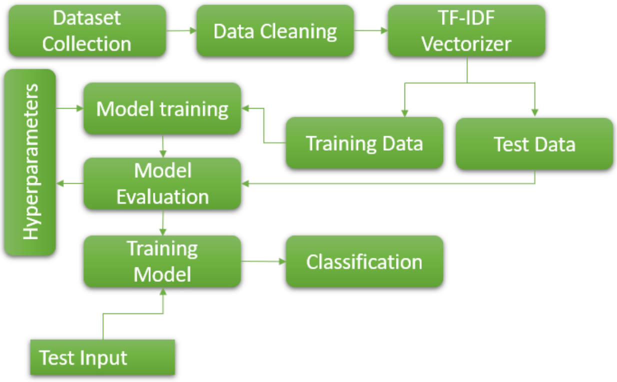
B. *Data Generalization*

Machine learning models trained on specific data sets may struggle to generalize to unseen data or adapt to new forms of fake news. Ensuring that data sets capture a wide variety of fake news scenarios and encompass various linguistic styles, sources, and media types is crucial for training models that can perform well in real-world settings.

C. *Labeling*

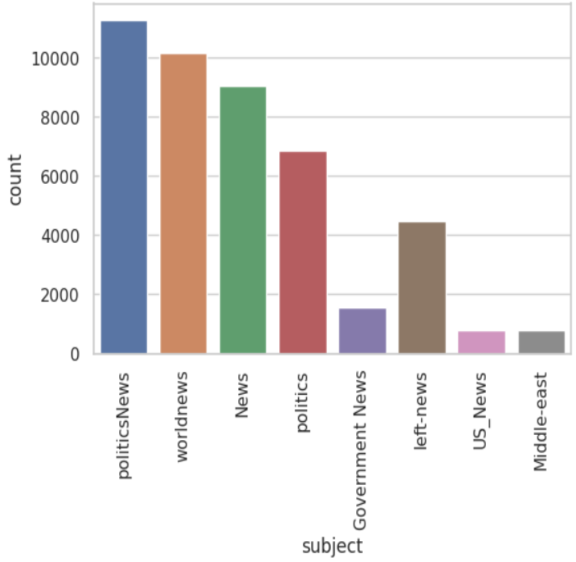
Obtaining accurate and reliable labels for fake news is a challenging task. It requires human experts to assess the veracity and credibility of each piece of news, which can be time-consuming and subjective. Differentiating between intentionally false information, biased reporting, and satire can be particularly challenging, leading to inconsistencies in labelling.

V. ARCHITECTURE



A. *Dataset Description*

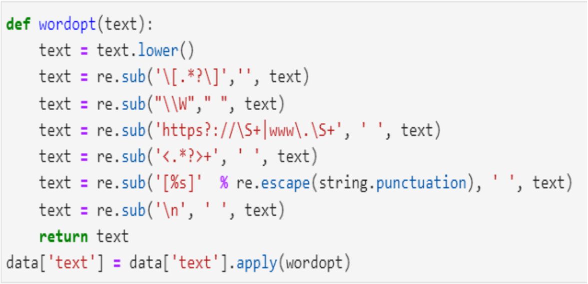
To differentiate the real news and fake news, the data has been collected from “The Online Academic Community”. The dataset contains about 40,000 articles among which includes both fake news as well as real news. The false news data and actual news data is separated into two different datasets, each with approximately 20,000 articles. The Real news consist of world-news and political news types. Fake news consists of Government news, middle-east, US news etc.



B. *Data Preprocessing*

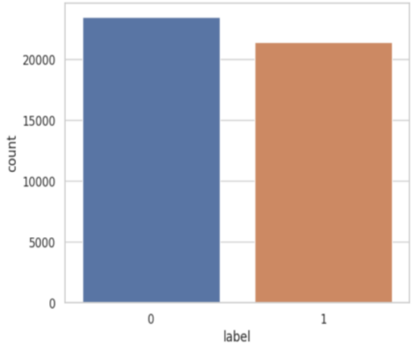
1. *Data Cleaning and Assigning classes*

The raw text data, such as news articles contains noise, irrelevant information, special characters. Text cleaning involves removing punctuation, tags, URLs and non-alphanumeric characters like hashtags, symbols, dollar signs. It includes techniques like lowercasing the text and removing stop-words to reduce noise.



```
def wordopt(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)
    return text
data['text'] = data['text'].apply(wordopt)
```

Fake news detection often involves using supervised learning techniques, where the machine learning model learns from labeled data to categorize news articles as either real or fake. Assigning classes to the dataset provides the necessary ground truth labels that guide the model's learning process. Real news is labelled as ‘1’ and fake news data is labelled as ‘0’. This labelled data act as a training set for building and training models. It allows models to learn patterns and features associates with real and fake articles, enabling it to make predictions on unseen instances.



C. Converting Text into Vectors

TF-IDF Vectorization: We use Scikit learns TF-IDF Vectorization. TF stands for Term Frequency, IDF stands for Inverse Document Frequency. TFIDF vectorization converts news articles or social media posts into matrix, which captures the importance of words in distinguishing between real and fake news.

Term Frequency (TF)

Term Frequency, is a numerical representation that measures the frequency of a term in a document. TF is calculated by counting occurrences of a term in a document and dividing it by the total number of terms in that document.

$$\text{TF} = \frac{\text{Number of times term appears in a document}}{\text{Total number of terms in the document}}$$

Words that appear more frequently in a document are assumed to be more indicative of the document content.

Inverse Document Frequency (IDF)

Document Frequency refers to the number of documents in a corpus that contain a specific term. IDF is calculated as the logarithm of the total number of documents in the corpus divided by the document frequency.

$$\text{IDF} = \log\left(\frac{\text{Total number of documents}}{\text{Document Frequency of the term}}\right)$$

IDF assigns higher weights to terms that appear in fewer documents. It captures the rareness or uniqueness of a term, considering that terms that appear in a limited number of documents are often more informative and distinctive.

TF-IDF Calculations

Compute the TF-IDF value for each term in each document by multiplying the Term Frequency (TF) with the Inverse Document Frequency (IDF). The resulting TF-IDF values capture both the local importance (TF) of a word within a document and its global importance (IDF) across the entire corpus.

The training set will be used to train the fake news detection model, while the test set will be used to evaluate its performance. The split ratio typically ranges **75%** for training data and **25%** for test data.

The training set is trained by suitable model such as Logistic Regression, Random Forest. During training, the model learns from the features of datasets. The model optimizes its parameters to minimize the error, aiming to classify news articles as real or fake accurately.

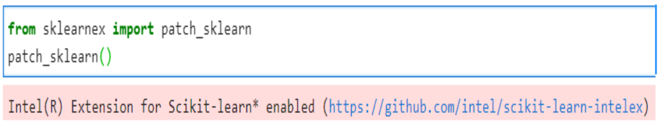
The trained model is applied to test set to predict the class label of the test samples. The predicted label is compared with the true label of the test set to evaluate the model performance.

D. Classification Algorithms

1) Logistic Regression

Logistic regression is a supervised learning algorithm used in fake news detection. It learns a binary classification model to separate real and fake news based on given features. The algorithm uses a logistic function to map the input features to probabilities, indicating the likelihood of an instance being fake news. A decision boundary is set to classify instances into real or fake news. Logistic regression is interpretable and can provide insights into feature importance.

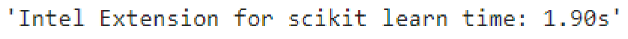
The algorithm for logistic regression uses ***"Intel Extension for Scikit-Learn."*** The performance of Scikit-Learn is frequently subpar. It's largely used now with Python. On huge datasets, some ML algorithms may take longer to process. Significant speed gains are possible using Intel Extension for Scikit-Learn.



```
from sklearnex import patch_sklearn
patch_sklearn()
```

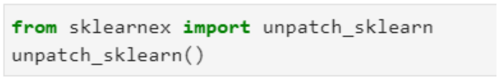
```
Intel(R) Extension for Scikit-learn* enabled (https://github.com/intel/scikit-learn-intelex)
```

The efficiency of the scikit learn algorithms is optimized by the scikit-learn-Intelex addition, which is added to scikit-learn by executing *'**patch_sklearn()**'*.

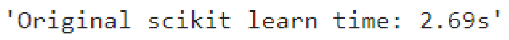


```
'Intel Extension for scikit learn time: 1.90s'
```

The training time for Logistic Regression using the Intel Extension for Scikit-Learn is around 1. seconds.



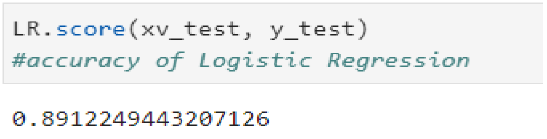
```
from sklearnex import unpatch_sklearn
unpatch_sklearn()
```



```
'Original scikit learn time: 2.69s'
```

The ***'unpatch_sklearn()'*** function is used to remove the Intel extension from Scikit-Learn and return to the normal implementation. The two figs show that the Intel extension for Scikit-Learn consumes less time than the original Scikit-Learn.

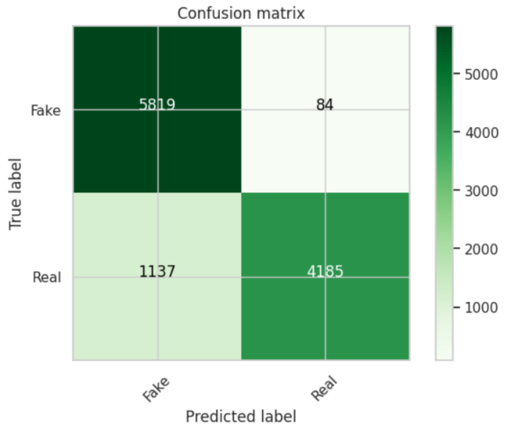
Accuracy of Logistic Regression :



```
LR.score(xv_test, y_test)
#accuracy of Logistic Regression

0.8912249443207126
```

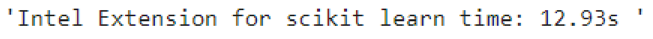
Confusion Matrix :



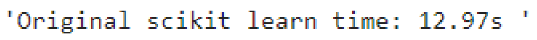
2) Random Forest Classifier

Ensemble learning involves combining the predictions of multiple individual models to make final predictions. The Random Forest Classifier utilizes the concept of ensemble learning by combining multiple decision trees. A random forest is a collection of decision trees, where each tree is trained on a random subset of the trained data and features. The randomness introduces during training helps to reduce overfitting and imporve generalization.

Intel Extension for Scikit-learn is also used in Random Forest Classifier like Logistic Regression. At the first step, Intel extension is used to patch the scikit learn for enhancement of performance.



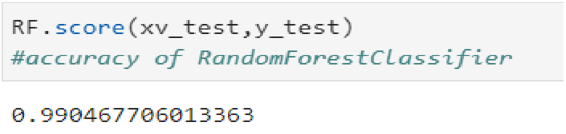
```
'Intel Extension for scikit learn time: 12.93s '
```



```
'Original scikit learn time: 12.97s '
```

In the above fig time taken for training the model by Intel extension for scikit learn is comparatively lesser the original scikit learn training. For larger datasets the difference in time for training will be higher.

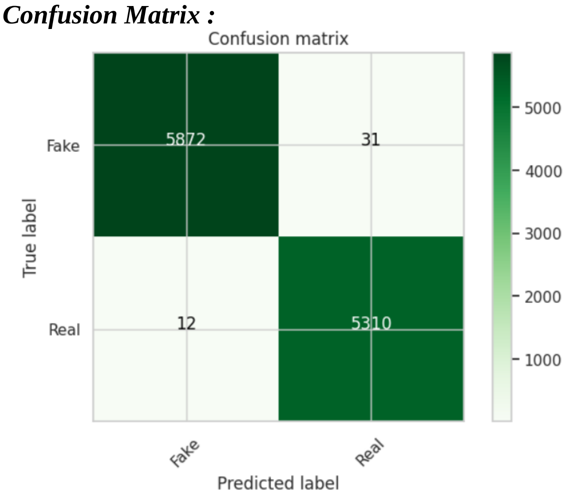
Accuracy of Random Forest Classifier :



```
RF.score(xv_test,y_test)
#accuracy of RandomForestClassifier

0.990467706013363
```

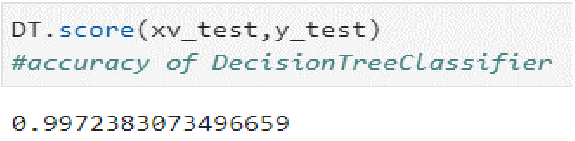
Confusion Matrix :



3) ***Decision Tree Classifier***

The decision tree algorithm will learn to make splits in the data based on the features and labels, creating a tree-like structure that represents decision rules for classifying news articles as fake or real.

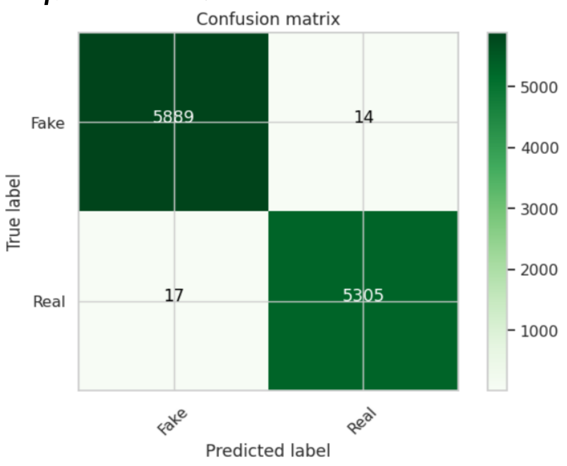
Accuracy of Decision Tree Classifier :



```
DT.score(xv_test,y_test)
#accuracy of DecisionTreeClassifier

0.9972383073496659
```

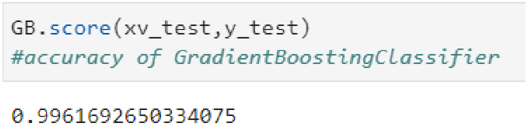
Confusion Matrix :



4) Gradient Boosting Classifier

This algorithm will sequentially build an ensemble of weak decision trees, where each subsequent tree corrects the mistakes made by previous tree. This process creates a strong predictive model.

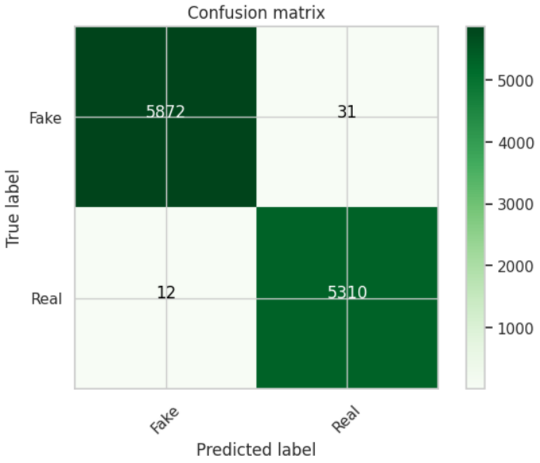
Accuracy of Gradient Boosting Classifier :



```
GB.score(xv_test,y_test)
#accuracy of GradientBoostingClassifier
```

```
0.9961692650334075
```

Confusion Matrix :



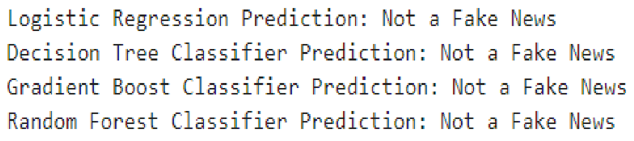
E. Comparision Among Different Classifiers

The below table represents the Accuracy among Different Classifiers used in this fake news detection project.

Accuracy among different classifiers			
Logistic Regression	Decision Tree Classifier	Gradient Boost Classifier	Random Forest Classifier
0.89122	0.99723	0.99616	0.99046

VI. TESTING

The user can provide input to the prompts either a news article or any text from the datasets. The user's input is preprocessed using the same preprocessing steps explained above like data cleaning applied during the training phase. The preprocessed input is then transformed into feature vector using the method of TFIDF vectorization. The trained model predicts whether the input is classified as fake or real based on the feature vector, and the results is displayed to the user.



```
Logistic Regression Prediction: Not a Fake News
Decision Tree Classifier Prediction: Not a Fake News
Gradient Boost Classifier Prediction: Not a Fake News
Random Forest Classifier Prediction: Not a Fake News
```

VII. REFERENCE PAPERS

1. "Fake News Detection on Social Media: A Review" by Srijita, Ghosh and Niloy Ganguly. (Link: https://arxiv.org/abs/1902.06673)

2. "Fake News Detection Using Machine Learning Ensemble Methods" by Iftikhar Ahmad, Muhammad Yousaf. (Link:https://www.hindawi.com/journals/complexity/2020/8885861/)

3. "Optimization and improvement of fake news detection using deep learning approaches for societal benefit" by Tavishee Chauhan, Hemant Palivela. (Link:https://www.sciencedirect.com/science/article/pii/S2667096821000446)

4. "Fake News Detection Using Deep Learning Approaches: A Review" by Shervin Minaee, Yuanfang Guan, and Amirali Abdolrashidi. (Link:https://ieeexplore.ieee.org/abstract/document/9054594)