**CSCI- 538**

**TITTLE: FAKE NEWS DETECTION USING MULTI-MODEL ENSEMBLE LEARNERS**

**GROUP-4**

1. Rukmini Sai Mohan Jayanth Srikantam - 50298996

**APRIL 30,2023**

**Abstract**

In the present era, when the internet is everywhere, everyone relies on a variety of online sources for news. News has spread swiftly among millions of users in a short amount of time as the use of social media platforms like Facebook, Twitter, and others has increased. The propagation of false information has far-reaching effects, including affecting election results in favor of candidates and shaping public opinion. Additionally, spammers use captivating news headlines to generate revenue from click-bait advertisements. With the use of Artificial Intelligence ideas, we intend to perform binary categorization of various news items available online in this work. We used Machine Learning and Natural Language Processing for this project such as Gradient Booster, Linear SVM, Random Forest, Decision Trees, Logistic Regression, K-Nearest Neighbor Classifier.

**1 Introduction and Background**

The digital age has brought about many advancements, but with them come challenges, including the spread of fake news. Fake news can easily be disseminated through social media platforms, posing a danger to individuals and society. Artificial intelligence (AI) is a branch of computer science that can help detect and combat fake news. Machine learning algorithms are trained on datasets to recognize fake news, and researchers are constantly working to improve their accuracy. In this project, we propose a machine learning ensemble approach to detect fake news, using various datasets and the NLP technique. We have also added a manual testing block for fact-checking, making the system more useful for users to determine whether news is fake or real.

## NLP (Natural Language Processing)

* Natural Language Processing is an area of artificial intelligence that allows machines to read, understand, and interpret human languages.
* NLP combines linguistics and computer science to interpret language rules and structure and develop models that can comprehend, decipher, and distinguish crucial information from speech and text.
* People often exchange vast volumes of publicly available data on social media platforms in order to engage with one another. The insights gained from this information can help you understand people's behavior and client preferences much better.
* Data scientists and machine learning specialists use this data to teach machines how to replicate human verbal behavior, saving millions of dollars in labor costs and time.
* NLP makes algorithms to understand what is going on and process it into a form which is easily comprehensible by the machine. It works as shown in the Fig.1.



Fig.1

**1.1 The problem you tried to solve**

In the present era, when the internet is everywhere, everyone relies on a variety of online sources for news. News has spread swiftly among millions of users in a short amount of time as the use of social media platforms like Facebook, Twitter, and others has increased. Biased opinions are produced as a result of the dissemination of fake news. Moreover, spammers use captivating news headlines to generate revenue from click-bait advertisements. Hence, in order to prevent this, we must eliminate fake news, and in order to do so, we must develop certain techniques for identifying fake news.

**1.2 Results from the literature**

Three Authors from Vivekananda Training Society Institute of Technology, Mumbai, distributed their examination paper on counterfeit news finding in 2018. In their test paper, they stated that the web-based media age began in the twentieth century. Finally, the number of people using the internet is growing, as are the number of posts and articles. They used a variety of tactics and instruments to detect fake news, including NLP strategies, AI, and counterfeit knowledge.

In 2017, Ho Chi Minh City College of Technology (HCMUT) student Nguyen Vo did analysis on the distribution of fake news. Yang et al. initially proposed the Bi-directional GRU with Attention component, which he employed in his effort to identify bogus news. Additionally, he employed a few calculations for profound learning and tried to apply various profound learning models, including auto encoders, GAN, and CNN.

A study report on counterfeit news detection was distributed by Stanford University's Samir Bajaj. He used the Signal Media News dataset to create a true informational collection.

In our work we did the similar work that visit students did but the difference in our project is, we used the Chinese COVID 19 news Data set. We used the TF-IDF vectorizer. We elaborated about the vectorizer and explained the vectorization process for the understanding of the user. We used ensemble learners in this work for getting the accuracy of the model. In addition to it we added the manual testing block in our work where a user can give a news as an input and verify it over our dataset, also know whether the given news is fake or not.

**1.3 What tools and programs are already available for the problem, or for closely related ones?**

There are several pre-existing methods to detect fake news. Some of these methods include:

* Fact-checking by human journalists or fact-checking organizations.
* Using online tools like Snopes or PolitiFact, which are dedicated fact-checking websites.
* Cross-checking with multiple sources to verify the authenticity of the news.
* Analyzing the tone and language used in the news article to determine its credibility.
* Verifying the author and their credentials.
* Checking the date of the article to ensure it is relevant and not outdated.
* Looking for inconsistencies and logical fallacies within the article.
* Analyzing the sources used in the article to see if they are reputable and reliable.

Some of these methods apart from ML to detect fake news, such as fact-checking, expert reviews, and crowdsourcing, typically rely on human intervention and verification. These methods require humans to manually review the news and analyze its credibility. In contrast, ML methods use algorithms to automatically analyze large amounts of data and identify patterns to determine the credibility of the news. ML methods can process data faster than human review and can analyze larger amounts of data but may not be as accurate as human analysis in some cases. Ultimately, a combination of both pre-existing methods and ML methods may provide the most effective approach to detecting fake news.

**2 Overview of the architecture**

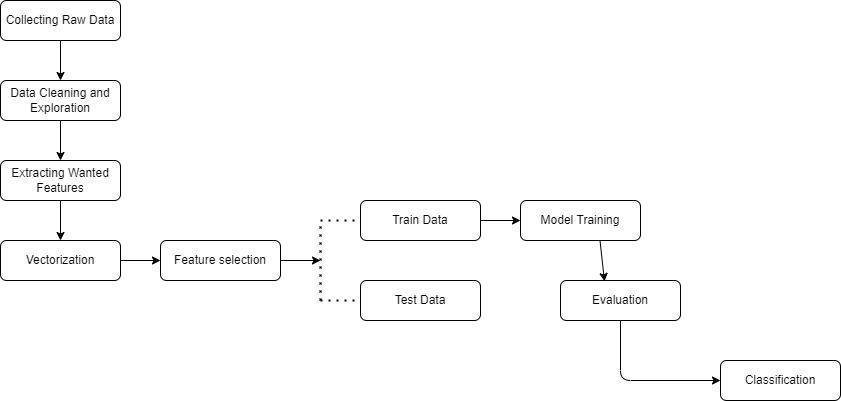


Fig.2

* As the above figure shows (Fig.2) our workflow.

* At First, we must collect the Raw data from the Open sources, and we have to explore the features of data and clean the data.
* We have extracted the wanted features and merged them for the selection process.
* We must select the merged features and drop the remaining features of the data set.
* We must split the following dataset into Test and Train data sets.
* We must train the data and use the Machine learning algorithms for finding the accuracy of the dataset.
* After finding the accuracy of data we must classify the data and use it for Manual testing.
* In manual testing the user will give a random data as input and our project will find the data among the available data set and classifies whether the given data is Fake or Not.

**2.1 Finished work: Running modules.**

All modules mentioned above are finished and running successfully.

**2.2 Work in progress: Modules designed but not implemented.**

All modules are successfully implemented.

**2.3 Future work: Modules a future continuation may have.**

Modules which might have a future continuation are Collection Raw Data and Model training. We can use some larger data and, we can use WebCrawler to collect the data directly from any website without doing it manually.

**3 Data Collection**

In total, we are taking two datasets, one is for real news and another is for fake news. The data in these datasets was taken from Kaggle website. These datasets contain the raw data of Chinese COVID 19 news data. Fake news dataset has 345 news and the real news dataset has more than 1000 news. The parameters present in these datasets are: Label, id, date, user\_id, text, pic\_url, video\_url, comment\_num, repost\_num, like\_num.

DATA PRE-PROCESSING

Initially we change the labels to 0 and 1 as real news dataset given label as 1 and fake news as 0.

|  |
| --- |
| fake["label"] = 0  true["label"] = 1 |

We merge the both datasets,

Then we merge the text and analysis column into news column.

|  |
| --- |
| df['News'] = df['text']+df['analysis'] |

Now we remove null values in the dataset by replacing it by space(‘ ’).

|  |
| --- |
| df['News'] = df['News'].fillna('') |

and we drop the unwanted columns like id, date, user\_id, pic\_url, vedio\_url, comment\_num, repost\_num, like\_num.

|  |
| --- |
| features\_dropped = ['id','date','user\_id','text','pic\_url','video\_url','comment\_num','repost\_num','like\_num','analysis',]  df = df.drop(features\_dropped, axis =1) |

Now the merged data set have only 2 columns: Label, News. And its ready to use now.

**4 Your methods and implementation**

### ML Learning Algorithms used in this Project

* **Logistic Regression**: Strategic relapse is one of the most obvious Machine Learning computations within the Supervised Learning methodology.
* It is a method for predicting a distinct ward variable from a collection of free components. The outcome of a direct subordinate variable is predicted using computed relapse.
* Therefore, the result should be a discrete or outright estimation. It frequently takes the form of Yes or No, 0 or 1, true or false, etc., but instead of communicating definite attributes like 0 and 1, it transmits probabilistic qualities that are somewhat similar to 0 and 1.
* In Logistic relapse, rather than fitting a relapse line, we fit an "S" molded calculated capacity, which predicts two greatest qualities (0 or 1).
* Strategic Regression can create probabilities and order new information utilizing both consistent and discrete datasets, calculated relapse is a key AI approach.

Importing Libraries from function,

|  |
| --- |
| from sklearn.linear\_model import LogisticRegression  from sklearn.metrics import accuracy\_score |

fit the model to Test and Train data,

|  |
| --- |
| LR\_model.fit(xv\_train,y\_train) |

predict the accuracy,

|  |
| --- |
| lr\_y\_pred = LR\_model.predict(xv\_test) |

getting the accuracy score and printing it,

|  |
| --- |
| score = accuracy\_score(y\_test,lr\_y\_pred)  print('Accuracy of Logistic Regression model is ', score) |

OUTPUT: Accuracy of Logistic Regression model is 0.9983230855226384

* **Decision Tree**: a regulated learning procedure might be utilized to take care of both order and relapse issues, but it is most normally utilized to address characterization issues.
* In this tree-organized classifier, the inside hubs address dataset ascription, the branches address choice principles, and each leaf hub provides the conclusion.
* There are two hubs: Leaf Node and Decision Node. While decision hubs are used to choose any option and have a few branches, leaf hubs are the outcome of those choices and don't have any extra branches.
* The qualities of the given dataset are used to guide the decisions or tests. It is called a choice tree because, like a tree, it starts with the root hub and grows into a structure resembling a tree with more branches.
* A choice tree essentially poses an inquiry and partitions the tree into subtrees dependent on the appropriate response (0/1) (Yes/No) (True/False) (Real/Fake).

importing libraries from function,

|  |
| --- |
| from sklearn.tree import DecisionTreeClassifier  from sklearn.metrics import accuracy\_score |

fitting the data to the model,

|  |
| --- |
| DT\_model.fit(xv\_train,y\_train) |

Prediction,

|  |
| --- |
| dt\_y\_pred = DT\_model.predict(xv\_test) |

Calculate the accuracy score and printing it,

|  |
| --- |
| score = accuracy\_score(y\_test,dt\_y\_pred)  print('Accuracy of Decision Tree model is ', score) |

OUTPUT: Accuracy of Decision Tree model is 0.9994410285075461

* **Random Forest** It is a striking ML estimation that uses the oversaw learning procedure.
* In ML, it will in general be utilized for both request and backslide issues.
* It depends on gathering realizing, which is the method involved with incorporating various classifiers to tackle a mind-boggling issue and work on the model's presentation.
* According to its name, Arbitrary Forest is a classifier that uses a variety of choice trees on various subsets of a given dataset and uses the normal to increase the predicted accuracy of that dataset.
* All things being equal, depending on a solitary choice tree, the arbitrary timberland gathers the estimates from each tree and predicts the last result dependent on the larger part votes of forecasts.
* The greater the quantity of trees in the backwoods, the more exact it is and the issue of overfitting is stayed away from.

Importing libraries from function,

|  |
| --- |
| from sklearn.ensemble import RandomForestClassifier  from sklearn.metrics import accuracy\_score |

Fitting the data to the model,

|  |
| --- |
| RFC\_model.fit(xv\_train, y\_train) |

Prediction,

|  |
| --- |
| rfc\_y\_pred = RFC\_model.predict(xv\_test) |

Calculating the accuracy and printing it,

|  |
| --- |
| score = accuracy\_score(y\_test,rfc\_y\_pred)  print('Accuracy of Random Forest Classifier model is ', score) |

OUTPUT: Accuracy of Random Forest Classifier model is 0.9994410285075461

* **K-Nearest Neighbour**: One of the most fundamental Machine Learning algorithms, it is based on the Supervised Learning method.
* The K-NN approach places the new case in the category that is most similar to the existing categories on the assumption that the new case/data and previous cases are similar.
* The K-NN algorithm stores all previously collected data and classifies new data points according to how similar they are.
* This means that as fresh data is generated, the K-NN algorithm can quickly classify it into a suitable category.
* The K-NN approach can be utilized for both relapse and arrangement, yet it is all the more usually used for characterization errands.
* The K-NN calculation is a non-parametric calculation, which implies it makes no suppositions about the hidden information.
* It is sometimes referred to as a lazy learner algorithm since it saves the dataset and only uses it when it is time to categorize it, deferring learning from the training set.

Importing libraries from functions,

|  |
| --- |
| from sklearn.neighbors import KNeighborsClassifier  from sklearn.metrics import accuracy\_score |

Fitting the data to the model,

|  |
| --- |
| KNN\_model.fit(xv\_train,y\_train) |

Prediction,

|  |
| --- |
| knn\_y\_pred = KNN\_model.predict(xv\_test) |

Calculating the Accuracy and printing it,

|  |
| --- |
| score = accuracy\_score(y\_test,knn\_y\_pred)  print('Accuracy of K Nearest Neighbour model is ', score) |

OUTPUT: Accuracy of K Nearest Neighbor model is 0.9586361095584125

* **Gradient Boosting** is a famous helping calculation. In inclination helping, every indicator revises its archetype's mistake.
* As opposed to Adaboost, the loads of the preparation examples are not changed, all things being equal, every indicator is prepared utilizing the remaining mistakes of archetype as marks.
* The ensemble comprises of N trees. Tree1 is prepared utilizing the component network X and the marks y.
* The forecasts named y1(hat) are utilized to decide the preparation set leftover mistakes r1. Tree2 is then prepared utilizing the component network X and the leftover mistakes r1 of Tree1 as names. The anticipated outcomes r1(hat) are then used to decide the lingering r2.
* The interaction is rehashed until all the N trees shaping the outfit are prepared.
* There is a significant boundary utilized in this strategy known as Shrinkage. Shrinkage alludes to the way that the expectation of each tree in the group is contracted after it is increased by the learning rate (estimated time of arrival) which ranges between 0 to 1.
* There is a compromise among estimated time of arrival and number of assessors, diminishing learning rate should be remunerated with expanding assessors to arrive at specific model execution. Since all trees are prepared now, forecasts can be made.

Importing libraries from function,

|  |
| --- |
| from sklearn.ensemble import GradientBoostingClassifier  from sklearn.metrics import accuracy\_score |

Fitting data to the model,

|  |
| --- |
| GB\_model.fit(xv\_train,y\_train) |

Prediction,

|  |
| --- |
| gb\_y\_pred = GB\_model.predict(xv\_test) |

Calculating the accuracy and printing,

|  |
| --- |
| score = accuracy\_score(y\_test,gb\_y\_pred)  print('Accuracy of Gradient Boosting model is ', score) |

OUTPUT: Accuracy of Gradient Boosting model is 0.9994410285075461

* **Support Vector Machine** or SVM is conceivably the most well-known Supervised Learning calculation, which is used for Classification similarly as Regression issues.
* In any case, fundamentally, it is used for Classification issues in Machine Learning.
* The objective of the SVM computation is to determine the best line or decision limit that can categorize the n-dimensional space, allowing us to confidently organize the new component later on. A hyperplane is the name for this best decision boundary. SVM picks the ridiculous centers/vectors that help with making the hyperplane.
* These unbelievable cases are called help vectors, and in this manner, computation is named as Support Vector Machine.
* Linear SVM is used for straight-line detachable data, meaning that if a dataset can be divided into two classes using a single straight line, it is straight-line detachable data, and the classifier employed is called a Linear SVM classifier.
* Non-straight SVM is used for non-straightforwardly disconnected data, which implies that when a dataset cannot be obtained using a straight line, it is referred to as non-straight data and the classifier employed is known as a Non-straight SVM classifier.

Importing Libraries from functions,

|  |
| --- |
| from sklearn.svm import SVC  from sklearn.metrics import accuracy\_score |

Fitting the data to the model,

|  |
| --- |
| svm\_model.fit(xv\_train,y\_train) |

Prediction,

|  |
| --- |
| svm\_y\_pred = svm\_model.predict(xv\_test) |

Calculating the accuracy and printing it,

|  |
| --- |
| score = accuracy\_score(y\_test,svm\_y\_pred)  print('Accuracy of SVM model is ', score) |

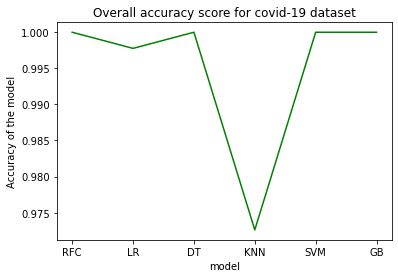
OUTPUT: Accuracy of SVM model is 0.9994410285075461

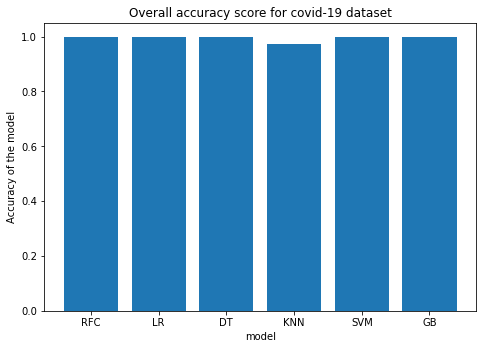
**5 Results and Evaluation**

In this part of the report, we discuss about the final result of the project and analysis of the project. The final result of our project is finding the news whether it is fake or not which is given by the user. We do this at the Manual testing block, as of our study with respect to all other reports and research papers no one had added this block in their project. Not only this we added the vectorization part too for the understanding of user. The focus of our project's entire investigation is on how to identify fake news using the provided data from the internet. Python is being used in this project for coding. To make the project work, we import a lot of libraries. We utilise machine learning methods to assess the dataset's accuracy and determine whether it can be used for manual testing. Finally, we perform manual testing to determine user preferences (including whether the provided news is true or false). Here the table named “Overall Accuracy score for Covid-19 dataset” shows the detailed score of the different models that we had used [Table 1]. Alongside the Graph shows the graphical representation of the same table [Graph1 and Graph2].

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Overall Accuracy score for Covid-19 dataset   |  |  | | --- | --- | | Model | Accuracy of Model | | Random Forest | Accuracy of RFC model is 0.9994410285075461 | | Logistic Regression | Accuracy of LR model is 0.9983230855226384 | | Decision Tree | Accuracy of DT model is 0.9994410285075461 | | KNN Classifier | Accuracy of KNN model is 0.9586361095584125 | | Linear SVM | Accuracy of SVM model is 0.9994410285075461 | | Gradient Boosting | Accuracy of GB model is 0.9994410285075461 | |

Table1



Graph 1 

Graph 2

**6 Achievements and Observations:**

* **Jayanth Srikantam:**

I in this project is to identify the problem, conduct research, develop a solution for the problem and, I have contributed my work in the document writing and Power point preparation. From this project I have gained some knowledge upon the algorithms, Critical Thinking and Communication with the Teammates.

**Data gathering:** Gathering pertinent data is one of the first steps in identifying fake news. Since different team members may have specialization in different areas, such as web scraping, data analysis, and data visualization, this process necessitates teamwork.

**Content analysis:** After data has been gathered, it must be examined to ascertain whether it is real or not.

* **Sankalpitha Vangala ,**

I Sankalpitha Vangala have played an important role in developing the models and implementing them to detect fake news using ensemble learning. Our contribution has helped in the successful implementation of various machine learning algorithms such as logistic regression, decision tree, KNN classifier which are used for binary classification of news items based on their authenticity.

* **Srija Gunturu:**

I Srija Gunturu have played important role in involving with other project mates to develop models to detect fake news using ensemble learning. So, I too contributed in developing Word Document Writing and Power Point Presentation. I mainly Concentrated in contributing Implementation Part ,how to use various ML algorithms such as Random forest , Linear SVM and Gradient Boosting and also involved in going through all the algorithms with team mates and testing them together.

**7 Discussion and Conclusions**

This Fake News detection system makes more change in the society as it was used as more as possible. We did it for the COVID 19 news data set as nowadays the fake news about the covid is spreading rapidly to make people know about the truth about the news. We used these algorithms and found the accuracy of the data and we had created a manual testing block where pc asks user to give an input (Any news) and verify the user given news with respect to dataset and gives output as “Real News” if the news is real and “Fake News” if the news is fake.

Our future scope is to use a big dataset having the various type of features and also, we are going to use WebCrawler method and we import the news to the dataset from the website directly and use it for the classification.

**8 References**

* Shu, Kai, et al. [2017], "Fake news detection on social media: A data mining perspective," ACM SIGKDD explorations newsletter, vol. 19, no. 1, pp. 22-36, <https://dl.acm.org/doi/abs/10.1145/3137597.3137600>.

The most important reference to this paper in our document is in Section Introduction

* Reis, Julio CS, et al. [2019], "Supervised learning for fake news detection," IEEE Intelligent Systems, vol. 34, no. 2, pp. 76-81, <https://ieeexplore.ieee.org/document/8661773>.

The most important reference to this paper in our document is in Section Methods

* Pérez-Rosas, Verónica, et al. [2017], "Automatic detection of fake news," arXiv preprint arXiv:1708.07104, <https://arxiv.org/abs/1708.07104>.

The most important reference to this paper in our document is in Section Methods

* Palacio Marín, Ignacio, and David Arroyo. [2019], "Fake news detection," Computational Intelligence in Security for Information Systems Conference, Springer, Cham, pp. 169-177, <https://link.springer.com/chapter/10.1007/978-3-030-33716-2_16>.

The most important reference to this paper in our document is in Section Methods

* Karimi, Hamid, et al. [2018], "Multi-source multi-class fake news detection," Proceedings of the 27th international conference on computational linguistics, pp. 2865-2876, <https://www.aclweb.org/anthology/C18-1258.pdf>.

The most important reference to this paper in our document is in Section Methods

* R. V. L, C. Yimin, and C. N. J. [2016], "Deception detection for news: Three types of fakes," Proceedings of the Association for Information Science and Technology, vol. 52, no. 1, pp. 1-4, <https://doi.org/10.1002/pra2.2015.1450520100127>.

The most important reference to this paper in our document is in Section Introduction

* T. Granskogen and J. A. Gulla [2017], “Fake news detection: Network data from social media used to predict fakes,” CEUR Workshop Proc., vol. 2041, no. 1, pp. 59–66.

The most important reference to this paper in our document is in Section 1.2

* R. V. L, C. Yimin, and C. N. J [2016], “Deception detection for news: Three types of fakes,” Proc. Assoc. Inf. Sci. Technol., vol. 52, no. 1, pp. 1–4.

The most important reference to this paper in our document is in Section 1.2

* V. Rubin, N. Conroy, Y. Chen, and S. Cornwell [2016], “Fake News or Truth? Using Satirical Cues to Detect Potentially Misleading News,” pp. 7–17.

The most important reference to this paper in our document is in Section 1.2

* Jain, Anjali, et al. [2019], "A smart system for fake news detection using machine learning." International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT). Vol. 1. IEEE, 2019.

The most important reference to this paper in our document is in Section 1.2.