Fake News Detection

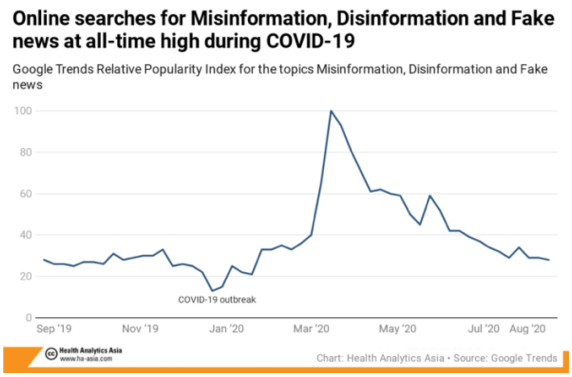
Anisha Yidala [G01223768]
Sravya Sangaraju [G01328406]

Abstract:

News articles, bulletins, and headlines i.e., information from across the world is so rapidly available than ever in the history and it is strongly impacting the society because of various factors like social media, news channels and other resources. Today the way any piece of information is projected has the power to make or break an issue from being heard. Fake news, deliberate disinformation, hoaxes, parodies, and satire are various ways to mislead people to gain financial or political benefits. Because of this extremely powerful nature we see a lot of misuse of this too. In today's world infinitely large amounts of data is available to us just a click away and it has become a boon and bane at the same time. Unlike the olden times when the resource of our information can be verified and the correctness of any data can be easily crosschecked, today because of the vast amounts of data, it has become tremendously difficult to verify the information we have. The objective of this project is to predict if any news article is reliable or unreliable as accurately as possible. Our study analyzes this information and extracts several key features which train various classifiers and build a model that efficiently predicts unseen data. Once maximum accuracy is achieved on real time data, the model can be developed into an automatic machine learning model (beyond the scope of this project) which works as an efficient way to combat the widespread dissemination of unreliable news by immediately alerting the user.

Introduction:

We now live in a digital era, where the news media evolved from newspapers and magazines to online news platforms, social media feeds, blogs, and other formats of digital media. Due to this transition, consumers are now able to acquire latest news at their fingertips. With the growing usage of World Wide Web and social media platforms (such as Facebook and Twitter), high volumes of false or misleading information are generated that has never been witnessed in the human history before. These social media platforms today have become extremely powerful and useful, as they allow users to discuss, debate and share ideas over various issues such as education, democracy, and health. Information that is being shared by the users on such platforms is usually unmonitored, some of which are misleading with no relevance to reality. Fake news can be described as any piece of disinformation which is in the form of hoaxes, frauds, or deceptions which are deliberately created to mislead or confuse the consumers, generally for monetary gain. Such unsubstantiated data can greatly impact the political, social, and economic relationships. The type of information that we consume directly effects our ability to take a decision. Our perceptions of the world are majorly shaped on the information we digest. In recent times, consumers have reacted ludicrously to the news that was later proved to be fake.



One such example is during the spread of COVID-19 pandemic, where false information was spread over the internet about the origin, nature, and behavior of the virus. The graph above depicts that the relative popularity index for misinformation, disinformation and fake news was all time high at the peak stage of COVID-19 during March 2020. It didn't stop with just that and misinformation was continuously generated about the availability of oxygen cylinders, beds in the hospitals and other medical resources, due to which the situation worsened as more people read about the fake content online. Identification and control of such news online is an overwhelming task.

Providentially, many computation techniques are now available which can be used to determine if a certain article as fake depending on their textual data using fact checking websites such as PolitiFact. There is also an availability of number of repositories that contain lists of websites that have been identified as fake or equivocal. However, there is a need for human expertise when using these resources to identify an article as fake or real. Moreover, these websites that check facts contain articles from only a particular domain such as politics and work poorly when identifying articles from other domains such as technology, sports, and entertainment. In our study, we try to solve this issue by developing two models that not only classify news articles as fake or real but also determine which domain of news is generating higher percentage of fake news.

Problem Statement:

The problem at hand is to detect fake news that is being generated in textual format on digital media platforms. We employ a two-phase approach to implement machine learning models where news articles are classified as fake or real and then we examine which category of news, for example political, entertainment, business, or technology contains the highest amount of fake news, thus helping us determine which domain needs the maximum regulation.

In the first phase, we train several linear models and ensemble models like K-NN (K-Nearest Neighbours) Classifier, Decision Trees Classifier, Random Forest Classifier, Naïve Bayes Classifier, XGB Classifier, which is a gradient boosting classifier and some other classifiers using the train data. Then, we use them to make predictions on our test data. We observe that using ensemble methods has been more efficient than using linear classifiers as they improve themselves constantly. We use the train-test split technique and F1-score metric to measure the performance of the models and select a model that most accurately predicts real and fake news.

In the second phase, we extract all the news that has been classified as fake in phase one and classify them into major categories of interest like politics, technology, entertainment, or business. We use several models like linear regression model, a multinomial naïve bayes classifier, decision tree classifier, random forest classifier, and ADA Boost. We train these models with a new dataset that has category labels and then we give the fake news extracted in phase one as input to these to predict the news category.

Literature Review:

[1] In an approach to identify Fake News on Social Networking, Nicollas R. de Oliveira, Dianne S. V. Medeiros, and Diogo M. F. Mattos proposed a method that uses text extracted from social media platforms to detect fake news. In their research, the selected Twitter to perform a check on whether the information posted on it is fake or real. The approach used by them for this study, helped them achieve an accuracy of 86% with minimum overhead. They have used three different methodologies in this research for the detection of fake news. The first two techniques being the execution of the machine learning algorithms for unsupervised clustering and classification. Their third technique take the representation in the vector space of frequent words into account to expand the detection procedure. To conclude, the outcome of their research clearly detects and differentiates fake news from real when the three methodologies are implemented on the text data collected from Twitter.

[2] In another research by Jayashree M Kudari, Varsha V, Monica BG, and Archana R, Fake News Detection using Passive Aggressive and TF- IDF Vectorizer was introduced as a fake news detection method. This paper is gives more

importance to fake news detection with their effects on social media and to differentiate between fake and real news. This paper says that passive- aggressive and TF-IDF vectorizer is efficient and through this, we get 90% accuracy. Passive-aggressive learning is the family of large-scale learning and they do not require any learning rate.

Dataset Description:

Datasets that we used for this project is publicly available and have been downloaded from the Kaggle repository. We used two datasets for two different phases – one for classifying whether the news is fake or real, second for categories news articles into various domains like politics, technology, entertainment, and business. Both the datasets contain textual data related to the news articles.

Dataset 1

This dataset has two files for train and test data. The train.csv file contains a full training data with 20800 records and 6 attributes as the following –

- id: Unique id for a news article
- title: The title of a news article
- author: Author of the news article
- text: The text of the article
- label: A label that marks the article as potentially unreliable; where 1 denotes that the news is unreliable and 0 denotes that the news is reliable.

The test.csv file consists of a testing training dataset with all the same attributes as train.csv but without the label.

Dataset 2

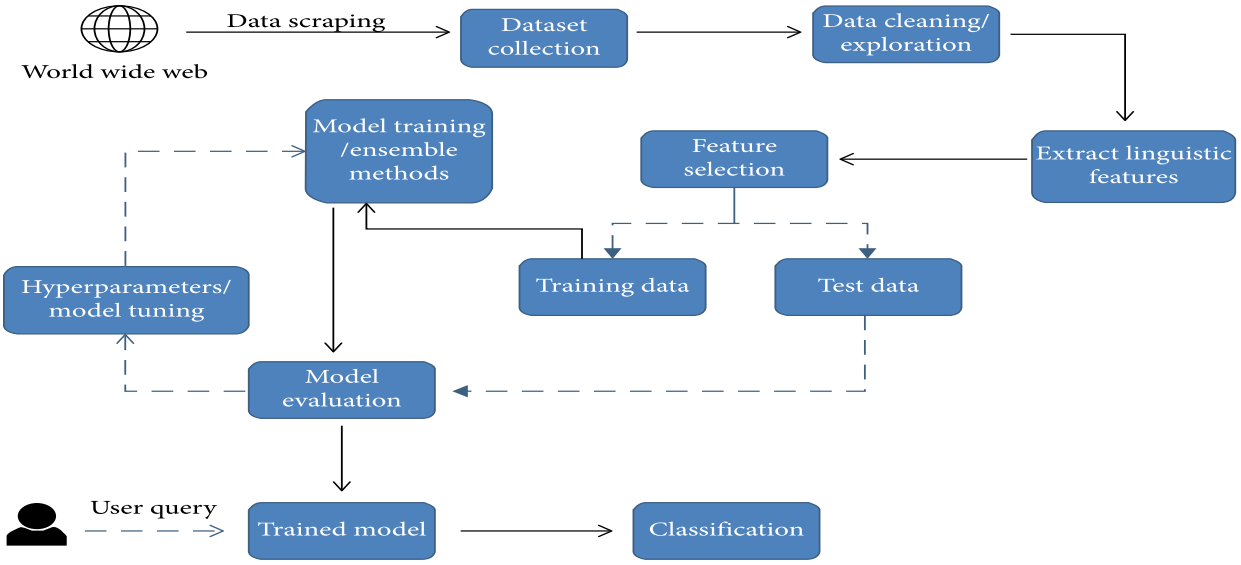
The second dataset that we used has two files as well for training and testing respectively. The Data_train.csv has 7628 records and 2 features namely story, which represents the news article and section, which represents the category that a particular news article falls under. The section attribute of the data has the following values corresponding to the categories mentioned below –

- 0 – Politics
- 1 – Technology
- 2 – Entertainment
- 3 – Business

The Data_test.csv on the other hand contains 2748 records of news articles without the section labels.

Methods and Techniques:

This project is implemented in two phases, where we first classify whether a given news article is fake or real and in the second phase, we figure out which category of news consists of maximum number of fake news. In both the phases, the initial step is to pre-process the data which is done using Natural Language Processing techniques since we are dealing with textual data. The following chart depicts the flow of the project.



Data Pre-processing:

NLP Text preprocessing is a method to clean the text to make it ready to feed to the machine learning models. Noise in the text data comes in varied forms like emojis, punctuations, special characters, and different cases. Properly cleaned data help us in good text analysis and making accurate decisions for real world business problems. Therefore, preprocessing of text is one of the important steps in machine learning before training the models. We first load the datasets using read_csv() from the Pandas library. The cleaning of textual data can be done by performing certain techniques as discussed below on the raw data that directly comes from World Wide Web.

- Removing all sorts of punctuations, and URLs.
- Removing Stop-words – They are the words in any language that do not add much meaning to a sentence. For example, articles, prepositions, pronouns, conjunctions can be considered as stop-words in English.
- Converting the entire data into lower case to deal with redundant words.
- Perform Tokenization – This is a process of splitting a large sentence into words, where each word is a token.
- Perform Stemming / Lemmatization – This is a procedure where each word is converted into to its root form by trimming the words.

All the techniques mentioned are performed by using various libraries in Python such as NLTK, re and wordcloud.

In the next step, tokenized vector is converted into a term frequency – inverse document frequency (tf-idf) matrix using TfdifVectorizer from Sci-kit Learn library for featuring extraction. An inverse document frequency weight is assigned to each word from the training text's vocabulary. This is then converted into a sparse matrix. Once, the text processing is finished, we proceed to train various classification models with the clean data.

Phase 1:

This is the phase where we predict whether a news article is fake or real by giving the preprocessed training data of Dataset-1 to various classifiers and training them. First, we perform a train-test split on the data which is used for cross validation once the models are trained. We then used the following classifiers from Sci-kit Learn library and performed a fit() method with the training data.

Models Used:

1. **Random Forest Classifier –** Random Forest is a supervised learning model which is an advanced form decision trees classification. It consists of large number of decision trees that work individually to predict an outcome of a class where the final prediction is based on the class that receives the majority votes. Random forest gives low error rates when compared to the other models, due to low correlation among trees. Since it is an ensemble model, we imported RandomForestClassifier() class from sklearn.ensemble module and selected the parameters as n_estimators = 500 which selects 500 decision trees. This model gives an F1-Score of 95% with the given parameters.
2. **XG Boost Classifier –** Boosting is another ensemble method that is widely used to train weak models to become strong models. This allows weak learners to classify data points correctly in an incremental approach that are usually misclassified. In successive rounds, the weighted coefficients are decreased for data points that are correctly classified and are increased for data points that are misclassified. In our project, we used XG boost classifier and achieved an accuracy of 96%.
3. **Logistic Regression –** Another model that we tried is Logistic Regression which is a classification algorithm. This is used when the data given has binary output. It uses a Sigmoid function to transform the output into a probability value. The objective is to minimize the cost function to achieve an optimal probability. Using this model, we could get an F1-score of 96%.
4. **Naïve Bayes Classifier –** It is one of the simplest and effective classification techniques that helps in building fast machine learning models that can make quick predictions. It is a collection of classification algorithms based on Bayes' Theorem, where each pair of features being classified is independent of each other. But unfortunately, this model couldn't perform well as others and generated an F1-Score of only 85%.

After these models are trained, we used the test data to predict the outputs for whether the news article is fake or real. Then we extracted only those records that were classified as fake to feed to the model in phase 2 and check in which category most of the fake news fall under.

Phase 2:

In this phase, we intend to train a machine learning model that categorizes news into different domains like technology, politics, entertainment, and business based on Dataset -2 by performing multi-class classification. Like phase 1, we first perform a train-test spilt on the cleaned training data from Dataset – 2. We have tried and evaluated F1-scores of the following machine learning algorithms in this phase:

Models Used:

1. **Random Forest Classifier –** Just like in phase 1, we imported RandomForestClassifier() class from sklearn.ensemble module and selected the parameters as n_estimators = 500 which selects 500 decision trees. This model generated a test accuracy of 95%.
2. **ADA Boost Classifier –** We used another Gradient Boosting algorithm in phase 2 known as ADA boosting classifier which is a meta-estimator. It fits a classifier on the top of original dataset and then fits additional copies of classifier on the same dataset. But we could only get a test accuracy of 86% and hence this model did not work well in this situation.
3. **Logistic Regression –** The last model that we tried is Logistic Regression classification algorithm and when trained with the preprocessed data, we achieved the test accuracy as 96.33% which is the highest in comparison to the other models that we tested.

Once the model selection was done, we tested the model with the data extracted from phase 1 that contains 2967 records, consisting of only those news articles that were classified as fake and visualized some interesting results.

Discussions and Results:

For plotting the graphs and charts we used Matplotlib library, and we also used Sci-kit learn library in python for the evaluation metrics.

Evaluation Metrics:

For evaluating the performance of the models that were used in this project, we used different metrics which were based on confusion matrix. It is a tabular representation which consists of four parameters – true positive, false positive, true negative and false positive. Using these we calculate the following metrics:

		Predicted	
		Negative	**Positive**
Actual	**Negative**	True Negative	False Positive
	Positive	False Negative	True Positive

- **Accuracy:** It represents the percentage of correctly predicted observations.
- **Recall:** It is the total number of positive classifications out of the true class.
- **Precision:** It represents the ratio of true positives to all the events predicted as true.
- **F1- Score:** It calculates the harmonic mean between precision and recall.

The following formulae represent the above-mentioned metrics,

- tp = true positive
- fp = false positive
- tn = true negative
- fn = false negative

$$\text{precision} = \frac{\text{tp}}{\text{tp} + \text{fp}}$$

$$\text{recall} = \frac{\text{tp}}{\text{tp} + \text{fn}}$$

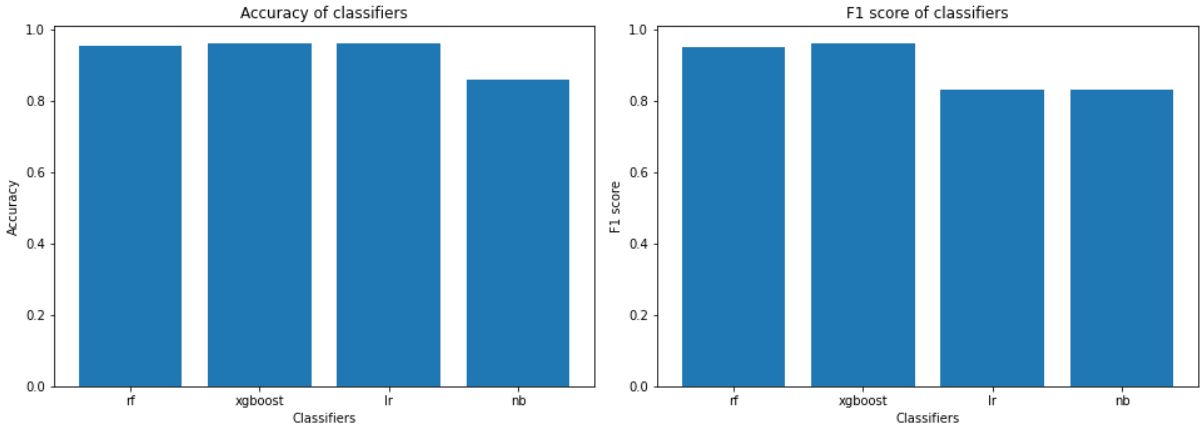
$$\text{accuracy} = \frac{\text{tp} + \text{tn}}{\text{tp} + \text{tn} + \text{fp} + \text{fn}}$$

$$\text{F}_1 \text{ score} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

Experimental results:

We visualized quite interesting results for both phase 1 and phase 2 of this project, which are represented in the graphs and table below:

Phase 1:



The bar graphs depict the F1- Scores and accuracy of the models that we tested in phase 1 and observed that XG Boost classifier performed the best when compared to the rest with an F1-Score of 96.15%.

Phase 2:

In this phase, we observed that the highest accuracy is achieved by Logistic Regression Classifier with a test accuracy of 96.33%. The following report describes the evaluation metrics for each domain of news.

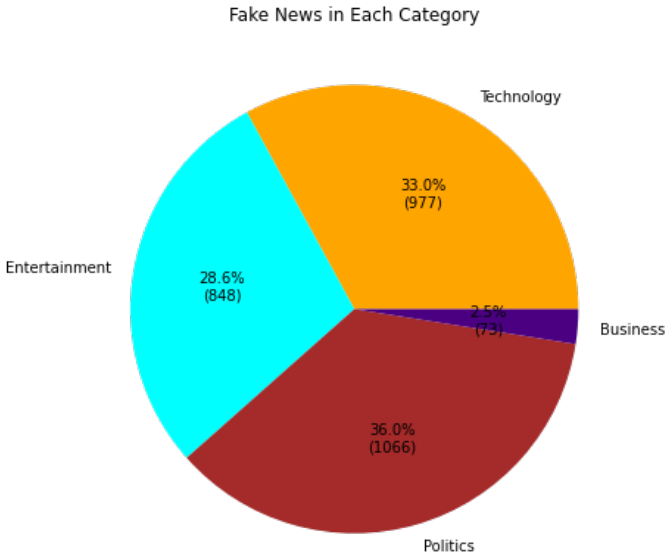
```
Train Accuracy: 98.92%
Test Accuracy : 96.33%


******** Classification Report ***************
               precision    recall  f1-score   support

     Politics       0.97      0.93      0.95       323
   Technology       0.97      0.97      0.97       549
Entertainment       0.94      0.98      0.96       402
     Business       0.96      0.96      0.96       252

     accuracy                           0.96      1526
    macro avg       0.96      0.96      0.96      1526
 weighted avg       0.96      0.96      0.96      1526
```

After the data from phase 1 is fed to the model in phase 2, the news articles which are classified as fake initially are distributed into different domains as represented in the pie-chart below.



We can observe that the highest amount of fake news falls under the political domain at 36%. This is followed by technology domain at 33%, entertainment domain at 28.6% and lastly business domain with a very minimal amount of fake news at 2.5%.

From these results, it is evident that political and technological news needs the most monitoring as compared to the rest to avoid the consequences that might occur due to the spread of fake news.

Conclusion:

Throughout this project, we had a huge learning curve of understanding the Natural Language Processing techniques and how classification models can be trained and tuned to achieve better results using different libraries in python. We had to skip over some other techniques due time constraints, which we will be working on later for improvising this project. The already existing techniques only detect fake news from one domain but in our study, we extended the solution to this problem by not only classifying the news but also by determining which domain is the most prone to generating false information so that necessary actions can be taken accordingly. By being informed about where the most percentage of fake news is originating from, more stringent checks can be placed to combat the dissemination of unreliable news in such categories, which is the need of the hour.

Directions for future work:

We further plan to improvise this project by implementing deep learning models and detecting news which is not only present in textual content but also in the form of videos, podcasts, and images. We also intend to design an automated system that sends warning notification whenever a user is exposed to false information in real time.

Dataset Availability:

The datasets used for this project are publicly available at https://www.kaggle.com/c/fake-news/overview and https://www.kaggle.com/akash14/news-category-dataset

File Structure:

The src folder contains two files –

- **Phase1.ipynb:** This file contains the implementation of Phase 1 discussed in this report where classification of news articles as fake or real is done.
- **Phase2.ipynb:** This contains the implementation of phase 2, where fake news is categorized into different domains.

References:

[1] Nicollas R. de Oliveira, Dianne S. V. Medeiros and Diogo M. F. Mattos,Member, IEEE – A Sensitive Stylistic Approach to IdentifyFake News on Social Networking- 2020 IEEE.

[2] Vectorizer Jayashree M Kudari, Varsha V, Monica BG and Archana R- Fake News Detection using Passive Aggressive and TF-IDF, 2020 IEEE.

[3] https://scikit-learn.org/stable/modules/classes.html

[4] https://www.python.org/doc/

[5] https://www.nltk.org