Fake News Analysis

Praveenaa Kulandhaivel, Haritha Athinarayanan
Team - 12 – Type 1
Social Media Mining

1 Abstract

In this era of social media, fake news is easy to fabricate and disseminate. Fake news combines pieces of news that may be hoaxes but pretend to be legitimate news and is generally spread through social media and other online platforms. Owing to the tremendous growth of the internet and social media usage, a large volume of fake news is generated and misleading information is circulated for personal and political gains. To detect and mitigate the spread of misinformation, and examine the veracity of a claim, key features such as the source of the news, claim and the content of the news, we classify whether a piece of news falls under the label (False, Misleading, True or Unproven) using Machine Learning Models.

2 Introduction

Over the last two decades, social media has become an inevitable addition to our daily lives, increasing our exposure to fake news and our chances of being persuaded by it. Using social media to get news updates is a doubtful advantage. Not all the news we read online is credible; with mobile phones, anyone may produce any content and share it. As a result of the quick attention from people, the dissemination of such news on platforms such as Twitter,and Facebook has instigated a serious crisis, resulting in political divisiveness, degrading people based on untrustworthy details, and losing faith in public institutions. The sources of this news are mainly the online platforms like Whatsapp, Twitter,and Facebook. Due to a lack of authenticity, it is difficult to figure out whether a piece of news is legitimate or a false assertion.

3 Related works

Many studies have recently sought to quantify the prevalence of false news on online platforms, how it influences people's perspectives and how it outperforms real news in terms of circulation. Few studies are conducted based on the source and nature of the claim of the deliberate misinformation and are used to identify hoaxes. Others include learning how susceptible individuals who are targeted with the fake news fall for it and are more prone to manipulation. Zhang, Dong, and Philip (2020) demonstrated effective fake news detection using a Deep diffusive Neural Network. The system extracts explicit features from textual information based on the credibility of the subject and creators and the diffusive model is made to learn these features in order to detect fallacious content. As Deep learning neural networks require a great quantity of data for learning, reinforced weakly supervised has been utilized for fake news detection Wang et al. (2020). Hybrid Deep models Ruchansky, Seo, and Liu (2017) employed RNN to capture the behavior of users who propagate fake news and learn the source attribute to identify false information. The dynamic nature of news creation and circulation makes obtaining labeled high-quality samples to train deep learning-based models infeasible. A Research paper on Natural Language Processing for Fake News Detection, Wang et al. (2020) presented the technical challenges involved in using NLP techniques for automatic fake news identification.

4 Model Description

4.1 Libraries

1. Numpy Van Der Walt, Colbert, and Varoquaux (2011)
2. Pandas McKinney et al. (2011)

3. Scikit Pedregosa et al. (2011)
4. Beautiful Soup Richardson (2007)
5. NLTK Loper and Bird (2002)
6. Matplotlib Hunter (2007)

4.2 Machine Learning Pipeline

For the given dataset, the following steps as shown in the pipeline were followed. Starting with reading the dataset, followed by Exploratory Data Analysis and Text pre-processing. The output is given for Text Representation and Feature Engineering, followed by Modelling and Pattern Mining. Finally, Evalutaion.

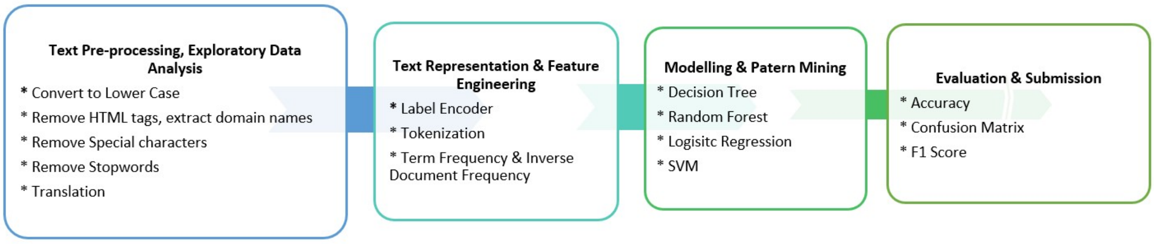

Figure 1: Overview of the steps done for Detection of fake news

4.3 Exploratory Data Analysis

Exploratory data analysis is the important process of looking at data for the first time to find patterns, spot outliers, test hypotheses, and check assumptions using summary statistics and graphical representations. It is a method for assessing every detail in the data at first encounter.

The following figures detail the composition of various columns in the train data set. It can be observed how much of the data is skewed.

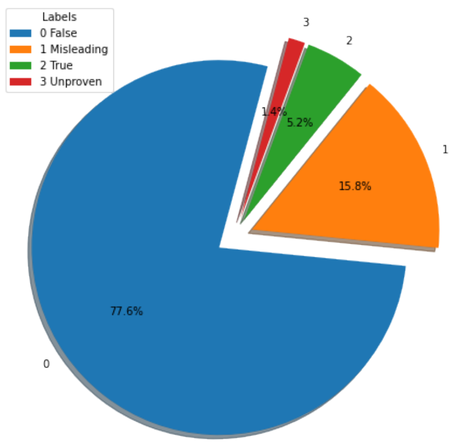

Figure 2: Pie Chart of Labels in the Train data set

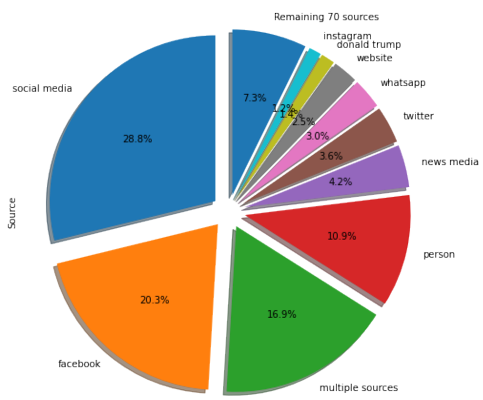

Figure 3: Pie chart of Top 10 sources in the Train data set

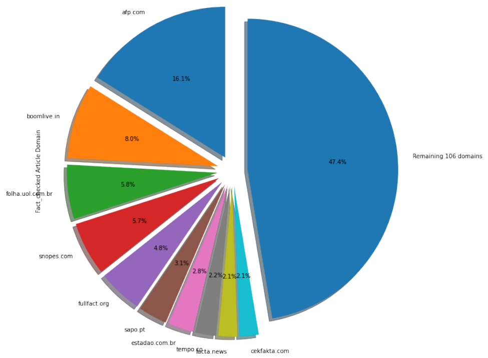

Figure 4: Pie chart of Top 10 domains in the Fact-Checked Article in Train dataset

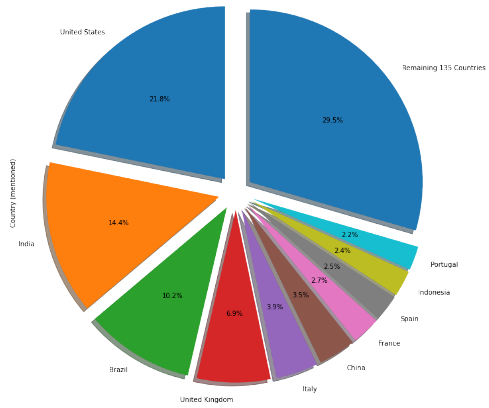

Figure 5: Pie chart of Top 10 Countries mentioned in the news articles in Train dataset

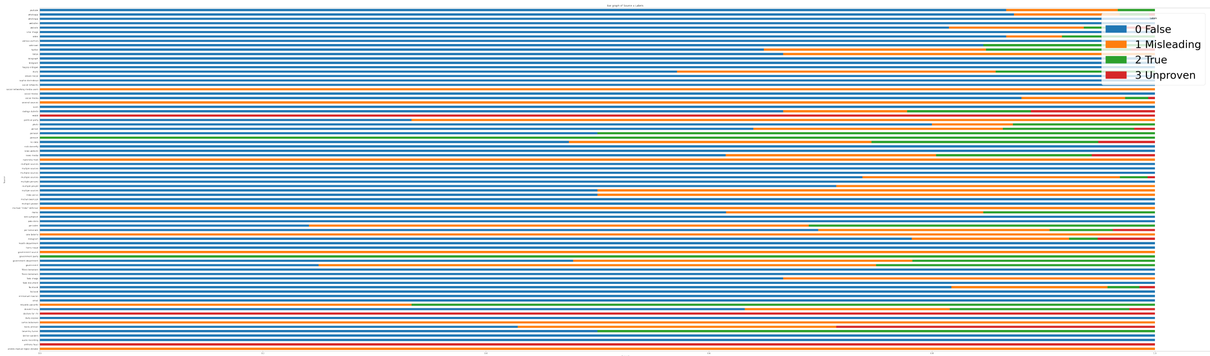

Figure 6: Stacked bar chart of all Sources and their correspoinding percentage of labels

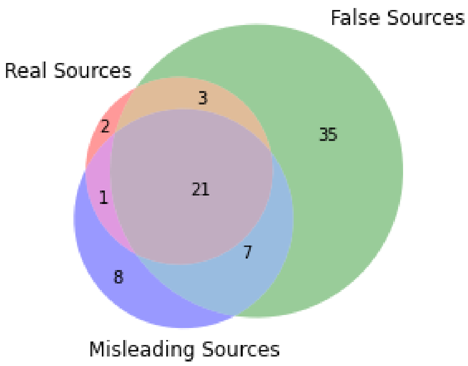

Figure 7: Venn diagram for Real, Misleading and False Sources

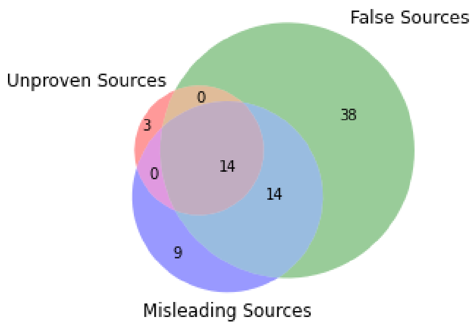

Figure 8: Venn diagram for Sources giving news that is not true

4.4 Text Pre-processing

4.4.1 Stop Words

NLTK package was used to import 'stopwords' corpus. Removing the stop words from the claim and the web-scraped information will let our model focus on the more important information making the claim.

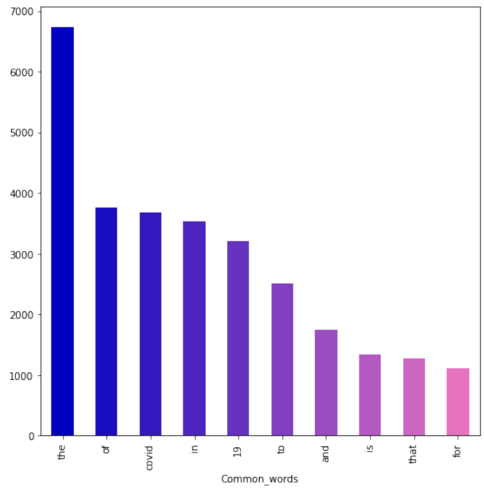

Figure 9: Bar chart of common words used in the claim for the train data set

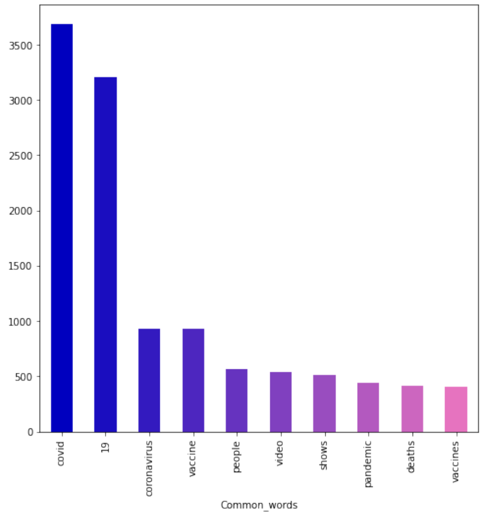

Figure 10: Bar chart of common words used in the claim without stop words for the train data set

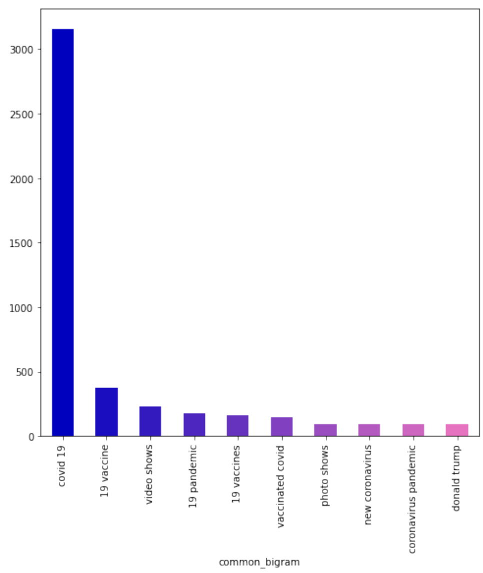

Figure 11: Bar chart of frequently occuring bigrams in the Claim without stop words in the train dataset

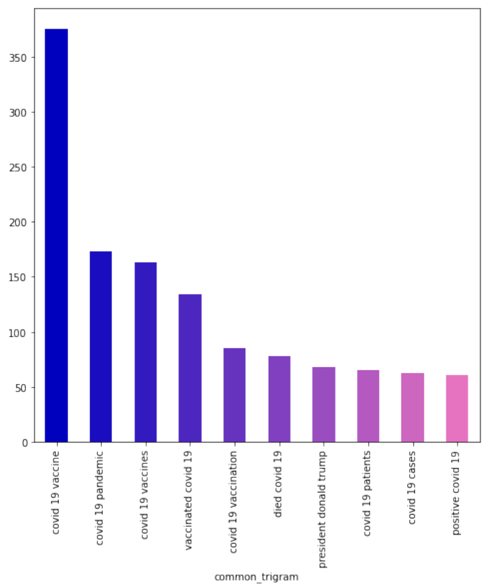

Figure 12: Bar chart of frequently occuring trigrams in the Claim without stop words in the train dataset

4.4.2 Web-Scraping

The Python library, 'Beautiful Soup' is used to scrape information from the Fact-checked articles. The title tag (headline of the news article) was scraped and concatenated with the claim.

```
    url = row['Fact-checked Article']
    reqs = requests.get(url, verify = False)
    soup = BeautifulSoup(reqs.text, 'html.parser')
    title = soup.find('title')
```

Fact-checked Article	Claim	Web-scraped Headline
https://leadstories.com/hoax-alert /2021/09/fact-check-harris-did -not-admit-vaccine- does-not-work.html	vice president kamala harris "admits" that covid vaccines don't work.	Fact Check: 'Kabala' Harris Did NOT Admit COVID Vaccine Doesn't Work — Lead Stories

4.4.3 Translate

Since, many fact-checked articles were not in English, the extracted headline had to translated to English for the model to train better. Google Translate library was used to perform this.

```
from googletrans import Translator
translator = Translator()
```

```
result = translator.translate(title.string, dest='en')
```

Polish	English
Prezydent Meksyku nie sprzeciwia sie szczepieniu dzieci na COVID-19	The President of Mexico does not oppose the vaccination of children at Covid-19

4.4.4 HTML Cleaning & Domain Extraction

The domain of the fact checked article was extracted and given as one feature to the model for training.

```
df['Fact_checked Article Domain'] = df['Fact-checked Article'].str.extract('//(?:\w+\.)
    ?(.*?\..*?)/')
```

Original URL	Extracted Domain
http://checkyourfact.com/2021/08/11/fact-check-marine-corps-general-david-berger-no-mandatory-vaccinations-marines/	checkyourfact.com

4.5 Feature Engineering

4.5.1 Label-Encoder

All the non-numerical categorical data, namely, Country, Source and Article Domain were converted to integer values using Label Encoder.

Index	Country (mentioned)	Source	Article Domain
1	Germany	person	dpa-factchecking.com
2	United States	website	leadstories.com
3	United States	Multiple people	afp.com

Index	Country (mentioned)	Source	Article Domain
1	45	50	30
2	130	73	64
3	130	27	4

4.5.2 Term Frequency - Inverse Document Frequency

Tf-idf measure how important a term is within the document relative to the collection of documents.

```
tfidf_vect = TfidfVectorizer(stop_words = 'english')
tfidf_train = tfidf_vect.fit_transform(x_train)
tfidf_test = tfidf_vect.transform(x_test)
tfidf_df = pd.DataFrame(tfidf_train.A, columns=tfidf_vect.get_feature_names())
```

The tf-idf performs the below steps with the help of TfidfVectorize.

1. Tokenization of text
2. Counting of tokens
3. Transforming raw tokens into tf-idf values

index	...	zookeepers	zoom	zum
0	...	0.0	0.0	0.0
1	...	0.0	1.0	0.0
2	...	0.0	0.0	0.0
3	...	0.0	0.0	0.0
4	...	0.0	0.0	0
...	...	...	...	...

Table 1: Dataframe after transforming using tf-idf

5 Experiment

The following classification models were trained and evaluated on 3 metrics, Accuracy, Precision, and F1 score.

5.1 Logistic Regression

```
LR = LogisticRegression()
LR.fit(tfidf_train, y_train)
y_pred_LR = LR.predict(tfidf_test)
```

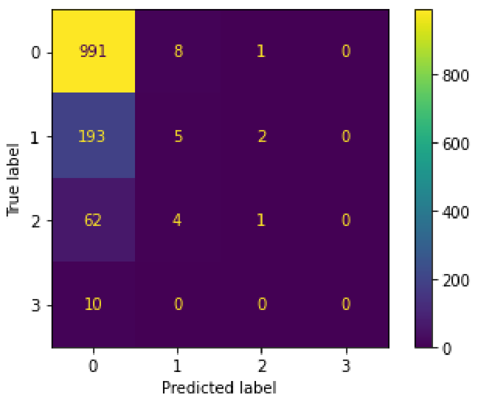

Figure 13: Bar chart of common words used in the claim for the train data set

Metric	Score
Accuracy	0.774
Precision	0.774
F1- score	0.774

5.2 Support Vector Machine

```
clf = svm.SVC()
clf.fit(tfidf_train, y_train)
y_pred_SVM = clf.predict(tfidf_test)
```

5.3 Random Forest Classifier

```
Rando = RandomForestClassifier(n_estimators=100,random_state=0)
```

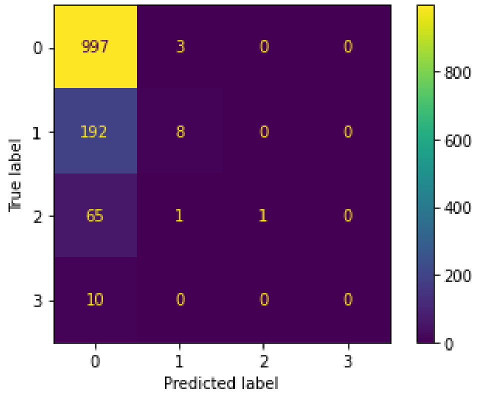

Figure 14: Bar chart of common words used in the claim for the train data set

Metric	Score
Accuracy	0.782
Precision	0.782
F1- score	0.782

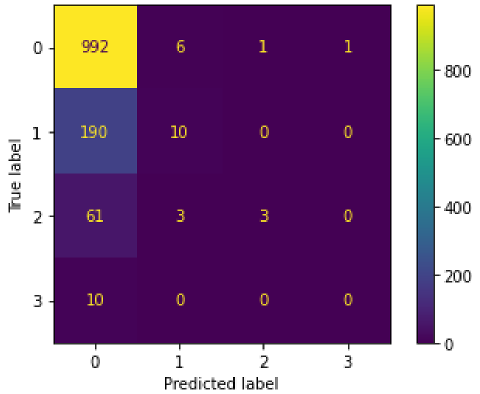

Figure 15: Bar chart of common words used in the claim for the train data set

Metric	Score
Accuracy	0.779
Precision	0.779
F1- score	0.779

5.4 Adaptive Booster

```
AdaBoostClassifier(DecisionTreeClassifier(max_depth=10),n_estimators=5,random_state=1)
```

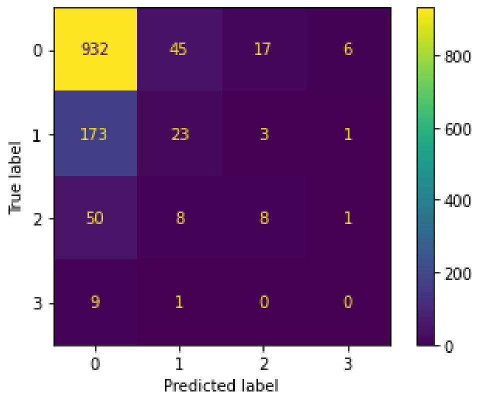

Figure 16: Bar chart of common words used in the claim for the train data set

Metric	Score
Accuracy	0.727
Precision	0.727
F1- score	0.727

Although there was no significant difference in the metrics of all four classifiers, SVM perfomed the best comparatively. The values predicted by SVM was extracted to CSV and submitted for the Kaggle competition.

6 Future Works

The work can be extended to creating a balanced data set where all classes are equally represented. Furthermore, Deep learning models such as Diffusion Neural Networks, autoencoder can be used for training. This work can also be extended to more datasets containing more records.

References

Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Computing in science & engineering*, *9*(03), 90–95.

Loper, E., & Bird, S. (2002). Nltk: The natural language toolkit. *arXiv preprint cs/0205028*.

McKinney, W., et al. (2011). pandas: a foundational python library for data analysis and statistics. *Python for high performance and scientific computing*, *14*(9), 1–9.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... others (2011). Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, *12*, 2825–2830.

Richardson, L. (2007). Beautiful soup documentation. *Dosegljivo: https://www. crummy. com/software/BeautifulSoup/bs4/doc/.[Dostopano: 7. 7. 2018]*.

Ruchansky, N., Seo, S., & Liu, Y. (2017). Csi: A hybrid deep model for fake news detection. In *Proceedings of the 2017 acm on conference on information and knowledge management* (pp. 797–806).

Van Der Walt, S., Colbert, S. C., & Varoquaux, G. (2011). The numpy array: a structure for efficient numerical computation. *Computing in science & engineering*, *13*(2), 22–30.

Wang, Y., Yang, W., Ma, F., Xu, J., Zhong, B., Deng, Q., & Gao, J. (2020). Weak supervision for fake news detection via reinforcement learning. In *Proceedings of the aaai conference on artificial intelligence* (Vol. 34, pp. 516–523).

Zhang, J., Dong, B., & Philip, S. Y. (2020). Fakedetector: Effective fake news detection with deep diffusive neural network. In *2020 ieee 36th international conference on data engineering (icde)* (pp. 1826–1829).