NLP Final Term Project Report

Topic Modeling on News Dataset and Fake News Detection

December 14, 2022
CIS 9665
Professor Deng

Team 4:
Suna Buse Agkoc, Syed Hossian, Yizhe Ling, Fengping Zhao

Introduction and background

With the development of networks and technology, information changes with each passing day, the amount of information on the network is also increasing, and there are a lot of people using the network. We live in the Internet age with information flooding. We can search the Internet quickly and get the information we need. The popularity of the Internet, the development of browsers, and the development of information technology have brought great convenience to our life. We can get information from different sources, including books, news, television, and the Internet. We can go to the library to read books and get information from them. We can also read the news to learn about domestic and international current affairs. The most commonly used information search channel is undoubtedly the Internet, you don't have to go to the library, don't turn on the TV, and you can get a lot of information. There is a wealth of information and news available just by swiping your phone or checking social media.

However, the flood of information is not necessarily a good thing. There are all kinds of information and news on the Internet. Real news can bring us the correct knowledge and information, while false news can lead us incorrectly. In view of this, while we enjoy the convenience of technology, we must also consider the harm of fake news. The online world is increasingly noisy, and there are plenty of malicious fabricators of fake news. According to our research in Statista, the data showed that 26% of Americans feel confident in their ability to identify fake news, but 67% of Americans believe that fake news causes great confusion. On an individual level, fake news can both mislead and confuse people's minds, making it difficult to distinguish between true and false.

Motivation

The harm of fake news cannot be ignored. Under the false guidance of fake news, some people are likely to listen to fake news and even forward it to their relatives and friends, so that the fake news will spread widely and cause serious social harm.

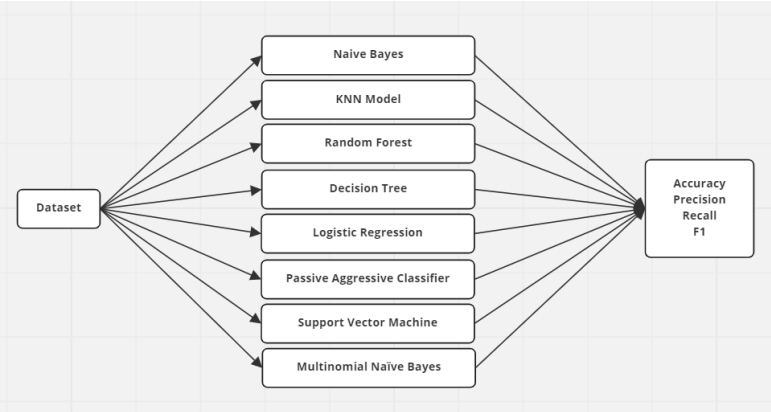
Our objective is to acquire a more comprehensive understanding of fake news and identify effective countermeasures and reduce or even eliminate the misinformation or fake news by applying our analysis in real life.

Text dataset description

To start off, we chose multiple data sources from Kaggle. We intended to start by analyzing the Fake and real news datasets by the owner Clément Bisaillon. It contains two CSV files named Real and Fake, where each CSV includes four columns of title, text, subject, and date. The dataset coverage started from December 31st,2015 to December 31st,2016. The Fake.csv is 62.79 MB and has 17903 unique values in titles; the Real.csv is 53.58 MB and has 20826 unique values in titles. In other words, there are 38729 rows, and each row represents different news in the pool.

The second dataset is called Fake or Real News, 30.7 MB, and has 6256 unique values (rows) in titles. This dataset is published by Jillani Soft Tech in Kaggle, and there are five columns which are number, title, text, and label. The number column in this dataset indicates the number of words for the text in that row.

Text preprocessing description



In order to have a more accurate analysis, we first make sure that our dataset is clean. More specifically, we're going to check whether there are any duplicates, missing values, or incorrect data types. After we make sure that our dataset is clean, we apply some preprocessing methods in NLP, such as regular expression, tokenization, normalization, removing stopwords, punctuation, and numbers, lemmatization, and stemming. Preprocessing allows us to adapt noise from high-dimensional features to low-dimensional areas to get more robust information from the text.

Research Question 1: Find out the hot topics in real/fake news and compare topics
Text analytic method

In this section, we want to learn about the hot topics from both the fake news data set and the real news data set. We would like to know if there is a significant difference between real and fake news topics. If there is, we can use these topics as a feature to determine labels. To accomplish this goal, we topic-model the data of both fake and real news datasets. Topic modeling is the statistical process of learning and extracting topics from a large number of documents in an unsupervised manner. Through topic modeling, we can cluster similar files from similar topics in the dataset and learn about the categories of news articles. In the previous step, we completed the preprocessing data, and the title and text of the dataset were cleaned. Therefore, we can directly use the cleaned data for topic modeling.

Since the dataset was so large, we only used "title_clean" text data for topic modeling, otherwise; the data modeling would take a long time to run. First, we applied word cloud to the "title_clean" data of fake news and real news. The results showed that the most popular topics in both fake and real news were politically related. Word cloud's data visualization helps us more intuitively feel the thematic differences between fake news and real news. For example, in this word cloud (Figure 1), we can see which words appear in the fake news but not in the real news.

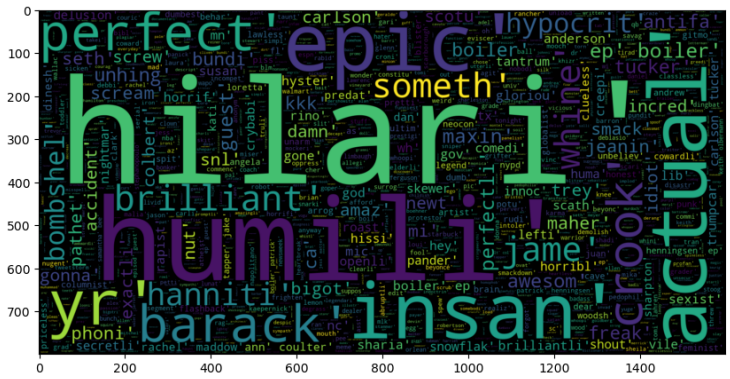


Figure 1

Then, we created LDA models for data from fake news and real news. “LDA mode is one of the most popular topic modeling methods. Each document is made up of different words, and each topic has a different word that belongs to it. “The ultimate goal of LDA is to find an optimal representation of the document-topic matrix and the topic-term matrix. The purpose of the LDA model is to find the topic of the document based on the words in the document.”(Figure 2) To generate the LDA model, we need to know how often each term appears in each document by building a document term matrix. Also, we need two key main inputs, the dictionary(id2word) and the corpus. “The dictionary is unique ids for each unique word. The corpus shows for each document number of times a particular word appeared.”(Mahajan). In addition, in order to better create LDA models, we also need to determine the number of topics, which is the k value in the model. We will explain how we determined the number of LDA model topics during the model-tuning process in this section.

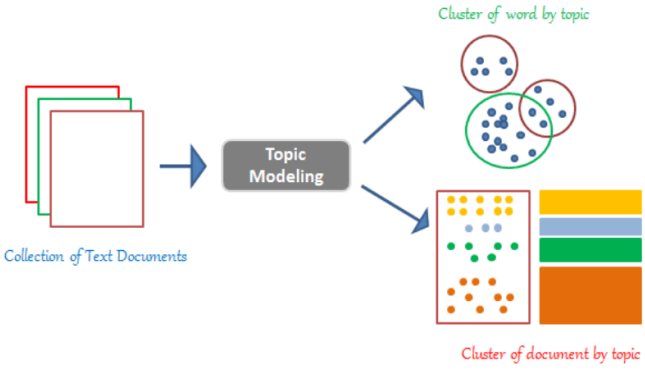


Figure 2

After creating the LDA model for the text data of true and fake news, we applied pyLDAvis to visualize the results of the LDA model. “pyLDAvis is designed to help users explain topics that have been fitted into the topic model of the text data corpus. The package extracts information from the fitted LDA topic model to provide information for web-based interactive visualization.” (“Pyldavis.”).Through pyLDAvis, we can more intuitively understand the relevant words in each hot topic and determine the proportion of each hot topic.

Model evaluation and Model tuning process

To determine the number of topics needed in the LDA model, we need to make some improvements to the model. The LDA model is an unsupervised learning model, so we can run some tuning processes to improve the model and select the number of topics that can be used. First, We randomly shuffled the entire data order, which might have helped us get more accurate results. Then, we create a function named "make_bigrams". In this function, we implemented bigrams using Gensim's Phrases model. "Gensim is a Python library for topic modeling, document indexing, and similarity retrieval of large corpora. Bigrams are two words that often appear together in a document." ("Gensim."). We applied bigrams in the text to help us better create the model. This step can help the model better understand the correlation between words and their usage, which helps us build the model better. We will call this function for "title_clean" text data as the argument.

We then created functions for "corpus_dict" and "build_lda_model". These two functions are to create the most basic LDA model for real and fake news data. The "Topic_Coherence" function is to see how good is the basic LDA model. In this step, the function "scores individual topics by measuring the semantic similarity between high-scoring words" (Kapadia). In addition, we created the "compute_coherence_values" function to evaluate the model scores for the different numbers of topics. The "show_graph" function helps us visualize how many topics will achieve the highest coherence score in the LDA model.

As shown in the graph, the highest coherence score in the fake news data is 14 topics(Figure 3.1), and the highest coherence score in the real news data is 14 topics (Figure 3.2). Since the data is randomly shuffled, the number of topics with the highest coherence score may be different for each run, but for each run, the results are always around 12 topics. Therefore, we created a variable "fake_news_num_of_topics" and "real_news_num_of_topics" to store these results and use them in topic modeling steps for more convenience. It is important how many topics we choose to build the model, which is a key input in the LDA model.

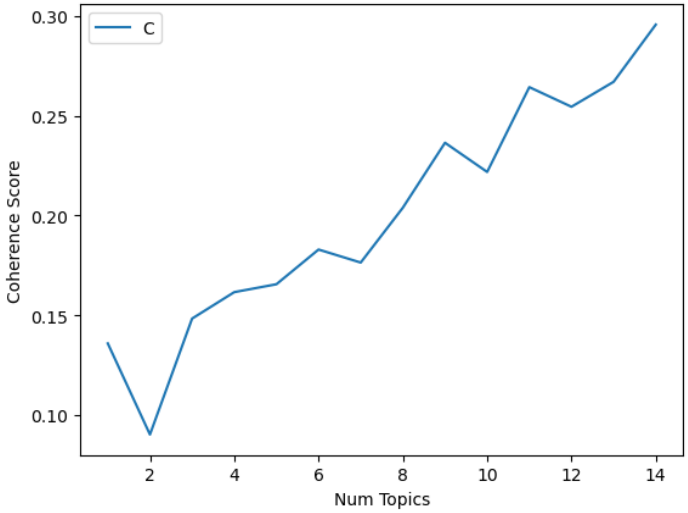


Figure 3.1 (number of topics for fake news LDA model)

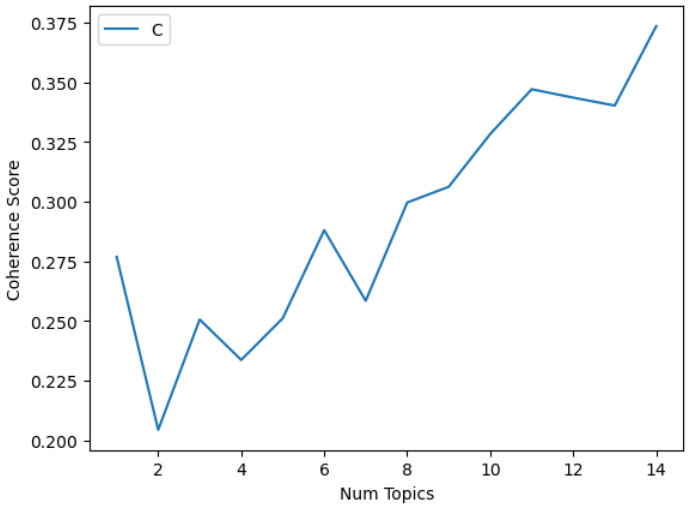


Figure 3.2 (number of topics for real news LDA model)

Conclusions

From the results of the word cloud (Figure 4.1 and Figure 4.2), we can see that there is no significant difference between the top topics of fake news and real news. They all revolve around political topics. In the results shown in pyLDAvis, we can see the proportion of each topic, the overlap, and the frequency of their keywords. Visualization allows us to understand the topics in the data better. For example, it can be seen from figure 5.1 that among the 14 topics in fake news data, the second topic does some overlap with topic 14. The top three keywords that often appear on this topic are “obama”,“syria” and “terrorist”. On the other hand, figure 5.2 showed that among the 14 topics in real news data, the second topic does not overlap with other topics. The top three keywords that often appear on this topic are “hous”, “trump” and “vote”.In addition, as we mentioned above, we believe that the keywords of the topic can be a feature to distinguish between real and fake news. So we put the words that appeared in the fake news but did not appear in the real news into a list. We decided to turn these words into a feature that can distinguish real news from fake news, and a column will appear in the later data to indicate whether the news contains these words.

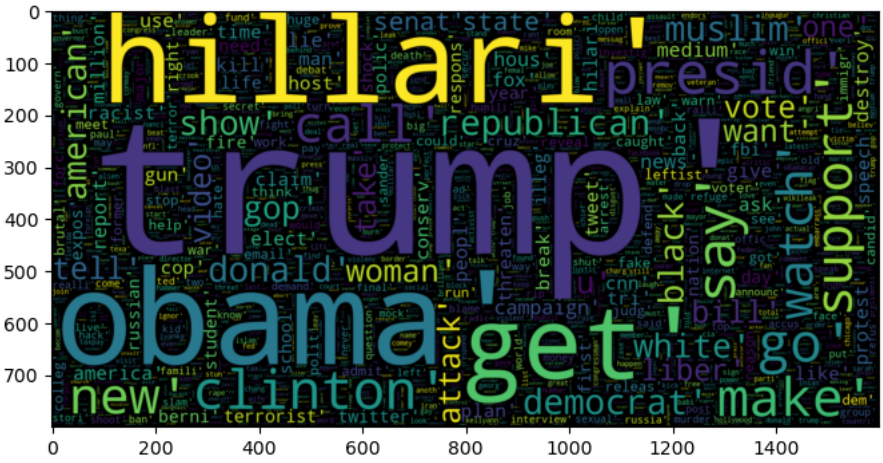


Figure 4.1 (word cloud for fake news data)

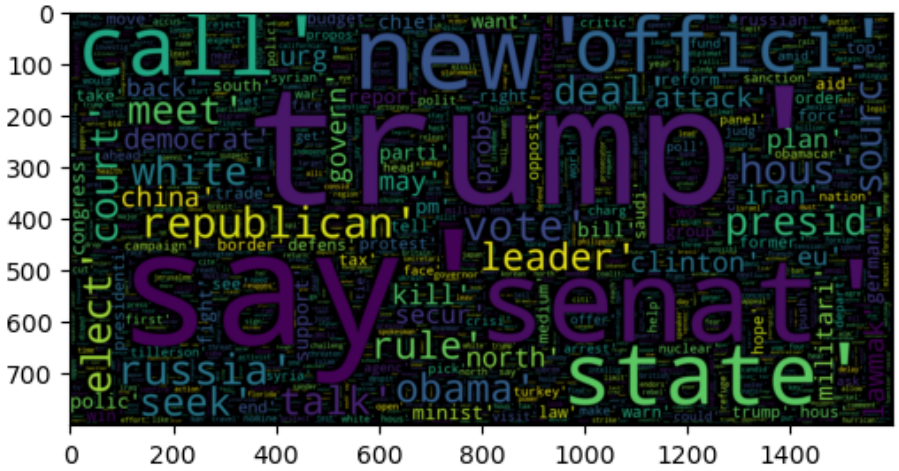


Figure 4.2 (word cloud for real news data)

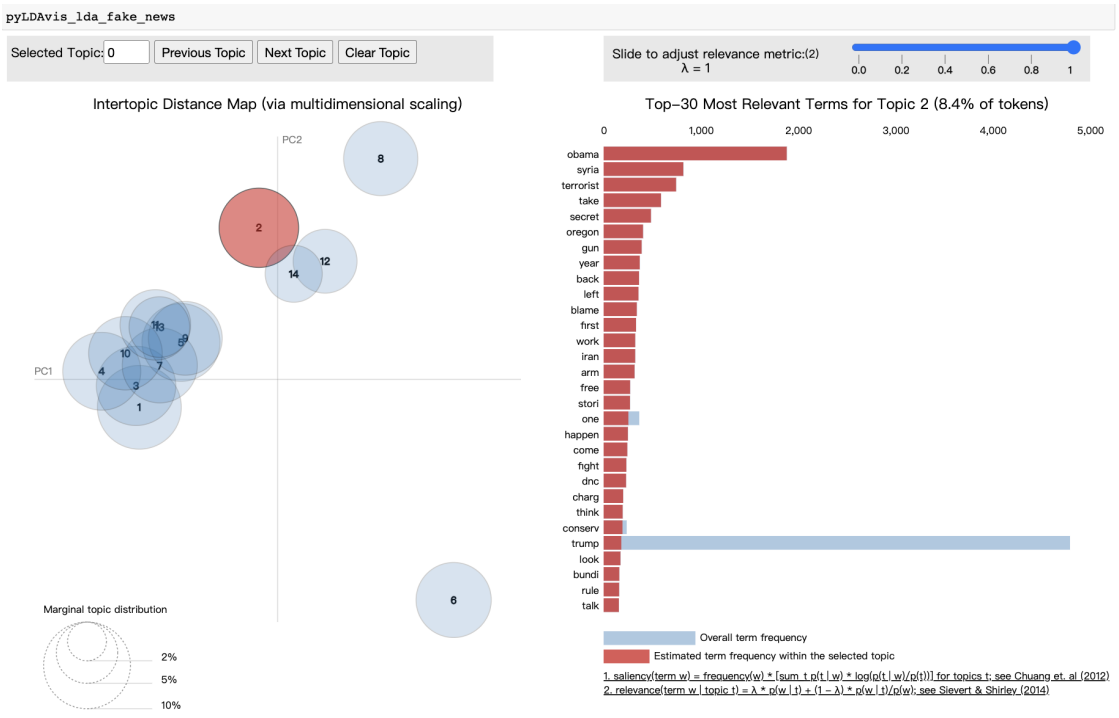


Figure 5.1

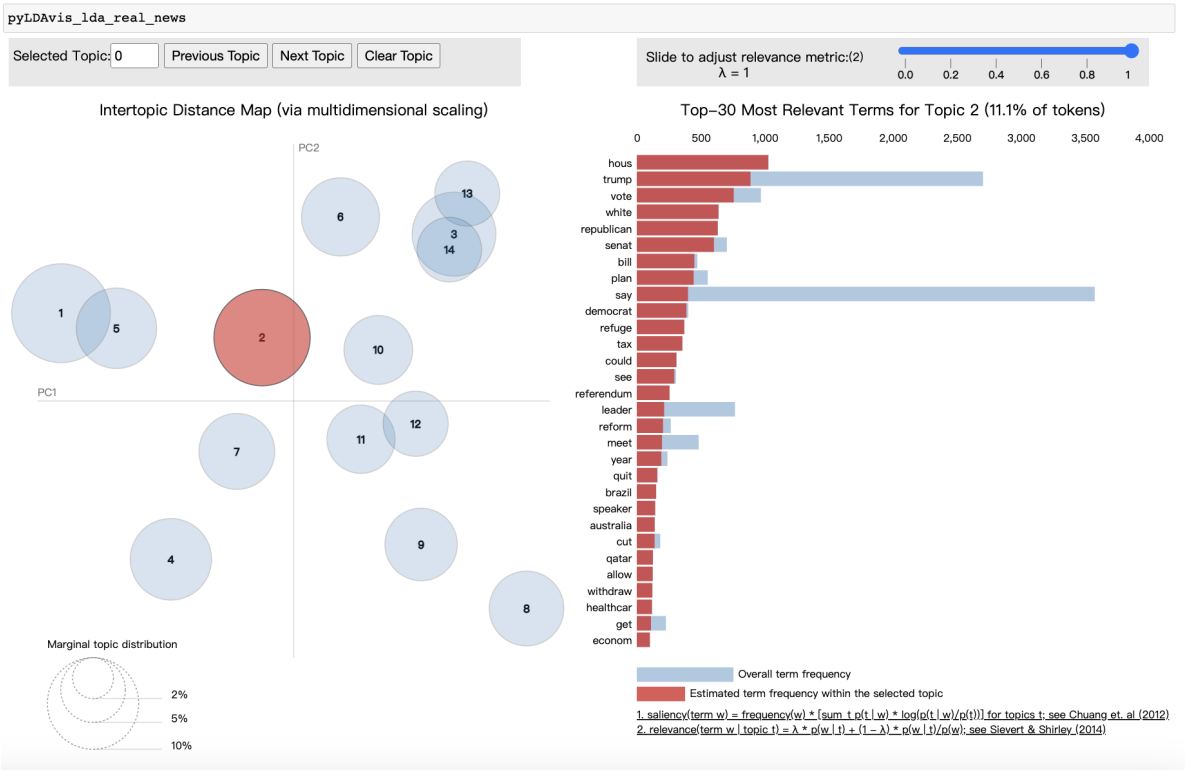


Figure 5.2

Research Question 2: Does fake news has more negative connotation words than real news?
Text analytic method and Model tuning process

In this section, we decided to use sentiment analysis to look at the emotional expression between fake and real news. The built-in functions SentimentIntensityAnalyzer () and polarity_scores helps us understand the different emotional score of the text. Due to the huge amount of data, it might take more than one hour to run the codes for the entire dataset, so we kept the output but commented out the code after the first successful run for the entire dataset. This gives us the brief idea that fake news does have a higher negative sentiment score than real news. Fake news also scored higher on positive emotions than real news. In order to check the more obvious gaps, we tested a few hundred of the most frequent words in the text data. The results clearly showed that fake news had higher negative and positive emotions than real news, and real news had higher neutral scores (Figure 6).

```
sa.polarity_scores(text=str(fake_news_sa))
```

```
{'neg': 0.234, 'neu': 0.652, 'pos': 0.114, 'compound': -0.999}
```

```
sa.polarity_scores(text=str(real_news_sa))
```

```
{'neg': 0.165, 'neu': 0.758, 'pos': 0.077, 'compound': -0.9974}
```

Figure 6 (polarity scores)

Conclusions

This result may indicate that fake news is more emotional, meaning that fake news is more likely to drive readers' emotions. On the other hand, real news is more emotion-neutral, it describes a fact rather

than arousing readers' emotions. We think polarity scores can be features that are part of the distinction between real and fake news. We will create three new columns to represent the three sentiment scores for each title, which are positive, negative, and neutral scores.

Design Thinking:

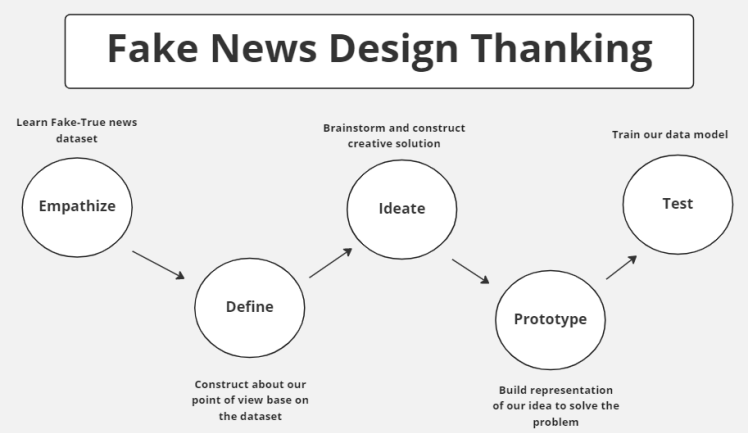


Figure 5.3

When we are starting our fake news project in the natural language process, we keep thinking about how we can start. First, we think about how we can add design thinking in our process. In order to solve our problem, we can grow our team's creative capacity by focusing on three core design thinking principles, or the 3 E's: empathy, expansive thinking, and experimentation. In our research question, we demonstrate our design thinking.

Research Question 3:

Text analytic method and Model tuning process

For our research question 3, classifying fake/real news, we utilize different classification methods and take this question into the data mining problem. In other words, we use 5 different classification models, which are Logistic Regression, Decision Tree, Random Forest, KNN, and Naive Bayes (GNB). In order to classify fake/real news, we use the scores from parts 1 and 2 in our project. More specifically, we use different polarity scores (i.e., positive, neutral, and negative) from our sentiment analysis and a dummy variable from our topic modeling that shows the results from the list of words that fake news has but real news does not. In this dummy variable, 1 indicates the news contains the words that appeared in that list, and 0 means the news does not contain the words that appeared in that list. Lastly, we used TF-IDF scores for our feature in classification models. Instead of using tokenization of words, we tokenize the sentence in measuring TF-IDF scores. After tokenization, we calculate the frequency of words in each sentence (TF), and calculate the IDF part, which shows the size of sentences containing a word, and lastly multiply these two matrices to estimate TF-IDF scores for each word.

After choosing our final variables, we split the data into one training set containing a random sample of 80% observations and one testing set with the remaining 20%. Our team also conducted a 5th Fold Cross Validation split and randomly shuffled the data. We chose the 5th Fold Cross Validation model since the most preferred k value was 1. Cross Validation is used to measure the test error related to a model to evaluate its performance. In order to compare our models, resampling is an effective strategy that shows the error of a model on unobserved data, helps to maintain the model's flexibility and provides

an effective model selection. In other words, by training five different parts, we can better evaluate what is going on in algorithms and it helps us to observe more information about our models' performance.

Conclusion

According to the results of our different models (i.e., Logistic Regression, Decision Tree, Random Forest, KNN, and Naive Bayes (GNB)), we created a comparison table with cross-validated accuracy, accuracy, precision, recall, and F1 scores of each model. The results in Table 1 show that the proposed Random forest is significantly better than all other methods in terms of all performance matrices. Also, in Fig 7 you can compare the models' performance metrics more clearly.

Model	CV Accuracy	Accuracy	Precision	Recall	F1
LOG-R	0.689794	0.678365	0.679716	0.679699	0.678364
DT	0.725642	0.742536	0.742954	0.740662	0.741071
RF	0.736115	0.754456	0.754436	0.753071	0.753427
KNN	0.736115	0.704323	0.704207	0.704595	0.704142
GNB	0.542839	0.541555	0.754361	0.560823	0.446545

Table 1:Performance Metrics(Part 3)

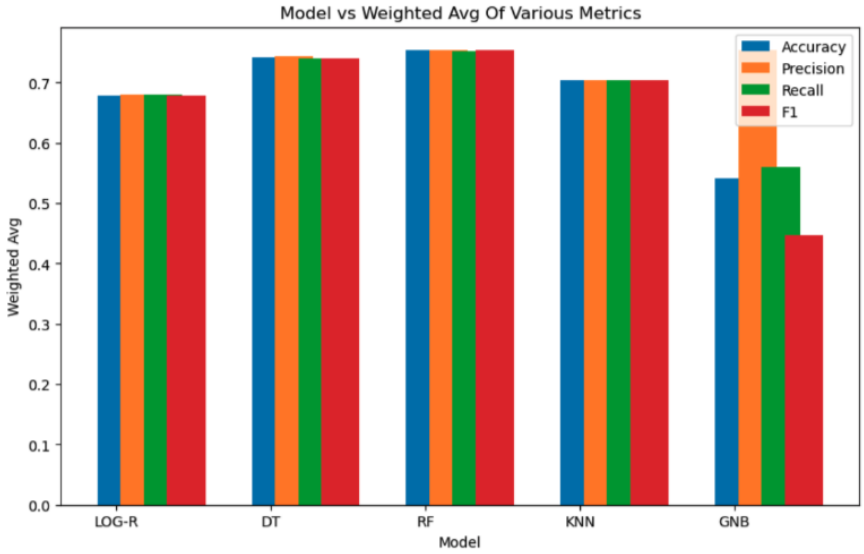


Figure 7: Visualization of Performance Metrics

Research Question 4:

Text analytic method and Model tuning process

For the last part of our project, we try to use different approaches to classify fake/real news. Instead of including the scores from parts 1 and 2 as our feature, we used only the bag of words(CountVectorizer) and TF-IDF(Tfidftransformer()). In this part, we used different models from part 3 except the Random Forest model, which was the best model in part 3. In Part 4, besides the Random

Forest model, we used a Support Vector Machine (SVM), MultinomialNB, and Passive Aggressive Classifier.

Even though using only CountVectorizer and Tfidftransformer increases the accuracies in the models, there are also some overfitting problems in each model. In other words, as seen in Table 2, our models work perfectly on the training dataset and perform poorly on test set signals that the models are overfitting. Overfitting problem means that our models don't generalize sufficiently and therefore just learn thoroughly how to predict well on specific sets of samples, but not perform well on unseen data. We can conclude that our models suffer from an overfitting problem since our dataset doesn't show any class imbalance, which means that there are almost similar amounts of fake and real news in our dataset.

Lastly, we can also observe this overfitting problem by using Local Interpretable Model-agnostic Explanations (LIME). It is an algorithm that describes the predictions by showing a local linear approximation of the model's behavior. The LIME explanation shows the probabilities predicted by the model for each sample. As seen in Figure 8.1 and 8.2, the probabilities predicted by the models for the sample of "fake news" is highly low (0.04 and 0.01) although we don't have any class imbalance in our dataset.

Name	Accuracy	Precision	Recall	F1 Score
Random Forest Classifier (Train)	1.000000000	1.000000000	1.000000000	1.000000000
Random Forest Classifier (Valid)	0.889523470	0.890210067	0.867851623	0.878888671
Multinomial Naïve Bayes (Train)	0.952281787	0.954953799	0.942853523	0.948865085
Multinomial Naïve Bayes (Valid)	0.892022131	0.894548349	0.868624420	0.881395805
Support Vectory Machine (Train)	0.978163860	0.975964579	0.977572225	0.976767741
Support Vectory Machine (Valid)	0.910405140	0.900230238	0.906491499	0.903350019
Passive Aggressive Classifier (Train)	0.997501041	0.997339078	0.997339078	0.997339078
Passive Aggressive Classifier (Valid)	0.893092986	0.887417219	0.880216383	0.883802134

Table 2: Performance Metrics (Part 4)

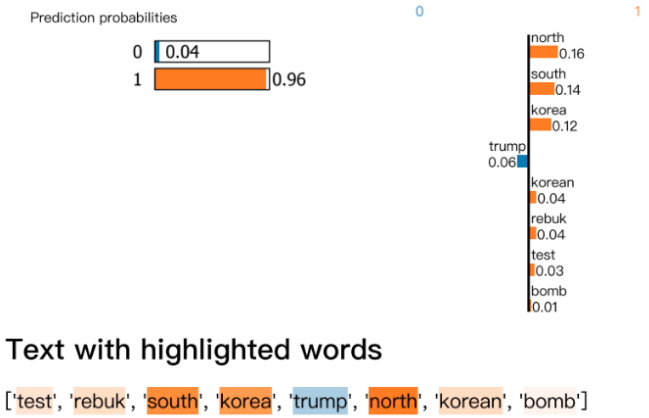


Figure 8.1: LIME (Random Forest)

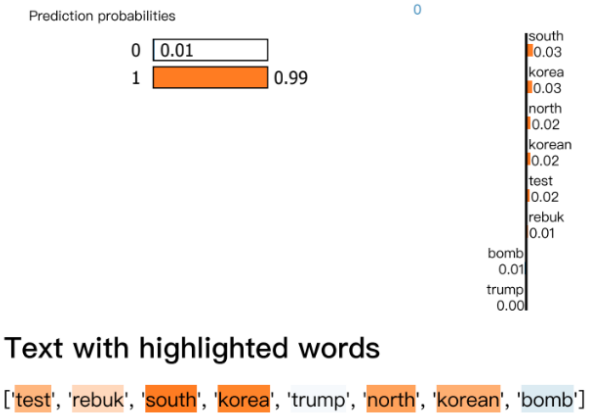


Figure 8.1: LIME (MultinominalNB)

Conclusion

Lastly, we wanted to present ROC curves which show a better indicator than overall accuracy since the area underneath (AUC) displays an overall measure of performance across all feasible classification thresholds unlike accuracy based on only a particular cut point. As seen in Figure 9, all of our models have similar performance, however, the Support Vector Machine (SVM) model seems to have the best ROC AUC result in our analysis.

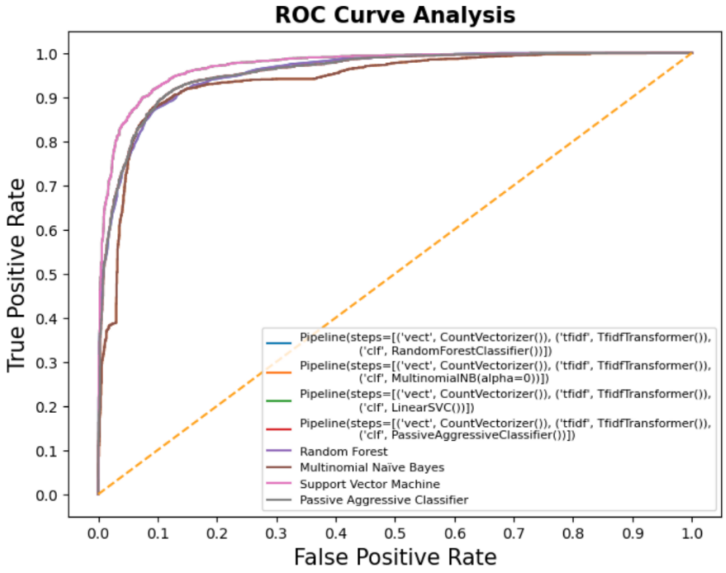


Figure 9: Roc Curve Analysis

Limitations and Improvements

Even so, our project still has many shortcomings and deficiencies. For example, our dataset is not big and up-to-date enough to support our analysis and research. Therefore, we need to retrieve more fresh data to develop stronger computational models that detect fake information. Improving the feature extractions from texts and images in news articles is necessary. Further strengthening the key features is also very important for our model development. For instance, the sentiments of news articles, the frequency of the words used, user information, and who left comments on the news articles, and social network information among users (who were connected based on activities of commenting, replying, liking, or following) could be used as key features. Furthermore, considering the characteristics of news consumption environments leveraging insights from social science.

Practical implications

Our initial practical plan is to build a tool that can Launching Applications, Mini Programs, or Extensions used on various platforms, such as mobile phones, tablets, laptops, chrome, etc.

This important step can improve people's ability to identify fake news and their perception of real news. More practically, the news media can help society reduce social conflict by filtering news that contains more negative connotations. Which leads to reducing socially harmful controversies or polarizations. Ensure we all have a healthier society!

Work Cited

Chiusano, Fabio. "Two minutes NLP — Explain predictions with LIME" *Medium,* Feb 2021 https://medium.com/nlplanet/two-minutes-nlp-explain-predictions-with-lime-aec46c7c25a2#:~:text=LIME%20

"Gensim." *PyPI*, https://pypi.org/project/gensim/.

Kapadia, Shashank. "Evaluate Topic Models: Latent Dirichlet Allocation (LDA)." *Medium*, Towards Data Science, 29 Dec. 2020, https://towardsdatascience.com/evaluate-topic-model-in-python-latent-dirichlet-allocation-lda-7d57484bb5d0.

"Latent Dirichlet Allocation." *Latent Dirichlet Allocation - an Overview | ScienceDirect Topics*, https://www.sciencedirect.com/topics/computer-science/latent-dirichlet-allocation#:~:text=An%20advantage%20of%20the%20LDA,formation%20and%20resulting%20document%20clusters.

Mahajan, Sahil. "NLP-a Complete Guide for Topic Modeling- Latent Dirichlet Allocation (LDA) Using Gensim!" *LinkedIn*, 20 Jan. 2021, https://www.linkedin.com/pulse/nlp-a-complete-guide-topic-modeling-latent-dirichlet-sahil-m.

"Pyldavis." *PyLDAvis*, https://pyldavis.readthedocs.io/en/latest/readme.html.

Watson, Amy. "Topic: Fake News in the U.S.", *Statista*, 21 June 2022, https://www.statista.com/topics/3251/fake-news/.