Fake News Detection

Leena Jawale
leenajawale@knights.ucf.edu

Sathya Narayanan Amarnath
sathyanarayanan@knights.ucf.edu

Srikar Sharma PV
srikar@knights.ucf.edu

Abstract

The proliferation of false information in social media news, other news articles affected the credibility of the news. Fabricated news contribute to confusion in the mind of the readers. The objective of this project is to build a classifier that can predict whether a piece of news is fake based only on its content, thereby approaching the problem from a purely Natural Language Processing (NLP) perspective. An important part of the goal is to compare and report the results from multiple different model implementations, and present an analysis of the findings. We use Logistic Regression, Random Forest and XGBoost classifier algorithms to detect the fakeness of the news. We classify the news based on headline and body of the news article. We calculate the performance of individual models based on F1 score and accuracy. We concluded that Logistic regression performs well when used on body of the news article compared to other classifiers.

1. Introduction

As an increasing amount of our lives is spent interacting online over the internet, more and more people tend to seek and consume news from social media, news agency homepages, search engines. On the other hand, it enables the proliferation of "fake news", i.e., low quality news with intentionally false information. Popular social media platforms such as Facebook, twitter have proven to be an effective means of channels for spreading these false news due to their wide reach and the speed in which information is spread. In this past election cycle for the 45th President of the United States, the world has witnessed a growing epidemic of fake news. The plague of fake news not only poses serious threats to the integrity of journalism, but has also created turmoil in the political world. The term 'fake news' became common parlance for the issue, particularly to describe factually incorrect and misleading articles published in social media feeds, news blogs and online newspapers, mostly for the purpose of making money through page views. The extensive spread of fake news has the potential for extremely negative impacts on individuals and society. Fake news is intentionally written to mislead readers to believe false information, which makes it difficult and nontrivial to detect based on news content. Therefore, the issue of fake new detection is both challenging and crucial. Hence, we decide to take up this challenge and find solution in an efficient way. From an NLP perspective, we identify the fake news using the classifiers Logistic regression, Random Forest, XGBoost. In our experiments, we used headline and body of the new articles to train the classifiers.

2. Background

A good distinction has been made between three types of fake news: serious fabrications (news about false events), hoaxes (providing false information via social media) and satire (mimicking genuine news items with irony and absurdity). Previous researchers have been successful in their attempt to differentiate satires from real news with promising results. Recently, a stylometric approach (writing-style) was proposed for identification of fake news from real news. Although stylometric approach was good for the classification of hyper partisan versus mainstream articles, both the approaches were not suitable for detection of fake news and real news. In this project, we successfully detect fake news using advanced machine learning algorithms like random forest, XGBoost classification techniques. We only focus on detecting fake news from "US News", "Politics", "Business" and "World", assuming that mostly fake news would be from these topics.

3. Data Source and Preparation

The data used for this project was drawn from three different sources. The real news datasets were collected from New York Times and The Guardian News articles. We used NYT API provided by New York Times, an American newspaper and Guardians API to collect the real news data. For collecting data from NYT, we obtained API key from their website, use that API key to get access to their database and then we scraped those articles. We use BeautifulSoup, an inbuilt python library for scraping, which visits the given URL and collects all the data from that page. Then we cleaned the scraped data by removing unwanted and unnecessary columns, replacing empty title and body of the news article by "DUMMY TEXT" and then storing the cleaned data in CSV file. We also calculate the TF-IDF scores, which tells the frequency of each term in the entire dataset. We visualized the most frequently occurring terms like TRUMP, CLINTON, RUSSIA, and SHOOTING etc.

For Guardian Dataset, we used the API key obtained from their website and accessed the dataset of articles. We obtained the data in json format. The cleaning process for this Guardian Dataset involves, reading the json data and putting in a dataframe, extracting headlines and body from the dataframe, removing all unwanted news sections, removing empty spaces and finally storing it in a csv file.

The fake news dataset is obtained from Kaggle.com. This dataset is cleaned and filtered to get only English language, to remove unwanted columns and to replace empty places with "DUMMY TEXT. The fakeness attribute in the dataset will indicate 0 for real news and 1 for fake news.

The number of news articles we obtained from NYT dataset were 8923, while the number of articles we obtained from Guardian dataset were 10124. We got 11841 fake news articles from Kaggle dataset. We combine all these fake and real news articles into a single csv file for implementation.

The attributes of the dataset are:

Id: A unique identifier for the article
Publication date: Article's publication date
Headline: Headline of the article
Body: The textual content of the article
Fakeness: binary 0/1 for real and fake news.

4. External Software

Third party libraries used in this project are as follows:

- BeautifulSoup – a python library to for pulling data out of HTML and XML files. To scrape or extract all the data from a particular website
- pymongo - To establish connection between python code and mongo database. It is used to retrieve the data from MongoDB and store locally.
- json - it is used to retrieve json data from the website
- requests - to get the requested URL for fetching the contents from the URL.
- time and datetime - To get the current date and time
- relativedelta- to calculate the absolute year, months, days, time between two given dates.
- pandas - for reading the csv and converting it to dataframe
- numpy - for array computation
- re- for regular expression
- sklearn- scikit learn for shuffling the data, performing cross validation, performance metrics
- matplotlib - for plotting graph
- xgboost - for performing XGBoost algorithm

5. System Implementation

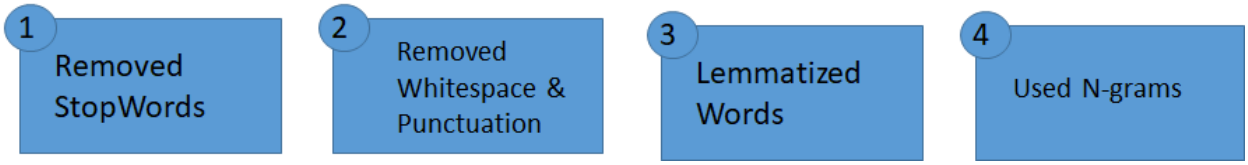
The system implementation for fake news detection is as follows:

a. **Data Collection –** To obtain fake news, we utilized an existing Kaggle dataset that had already collected and classified fake news. The articles were derived using the B.S. Detector, a browser extension that searches all links on a page for references to unreliable sources and checks them against a third-party list of domains. The fake news dataset from Kaggle contains 11000 articles. We obtained real news using NYT API and The Guardian Post API. We used NYT API and Guardians API to scrape the news articles from New York Times

and Guardians Post respectively. We scraped 9000 articles from NYT as real news and 12000 articles from Guardian Post API as real news. Total 10000 real news assuming that real news are more than fake news. We used BeautifulSoup, an inbuilt python library for scraping, which visits the given URL and collects all the data from that page.

b. **Data cleaning and Data Combining-** We parsed and cleaned the fake news dataset and datasets obtained from NYT API and The Guardians Post API i.e. real news datasets. We parsed the headline, bodyText attribute from these datasets to search for non-ascii characters and replace non-ascii characters by empty space. We added fakeness attribute as 0 for real news and 1 for fake news. The cleaning process involves removing of unnecessary columns, replacing empty title and body of the news article by "DUMMY TEXT". The Guardian Post datasets required reading the json files, putting in a dataframe, extracting headlines and body from the dataframe, removing all unwanted news sections, removing empty spaces and finally storing it in a csv file. After cleaning the fake and real news datasets, we combined and shuffled the datasets for further processing.

c. **Text Preprocessing -** We used scikit-learn in Python to tokenize the text and perform preprocessing steps like removing white space and punctuation, lemmatizing words, part of speech tagging. These steps helped reduce the size of our corpus and add context prior to feature conversion.



d. **Feature Generation -** To analyze and model text after it has been preprocessed, it must first be converted into features. Techniques we used TF-IDF. Using TF-IDF, we found the relative importance of words in both our fake news and real news datasets. TF-IDF is used to determine word importance in a given article in the entire corpus. There was significant overlap between the two - "trump" was the most important word in both types of articles, and words like "russia", and "women" also ranked highly. We used TfidfVectorizer to convert dataset to TF-IDF features. Below is TF-IDF results for fake news articles.

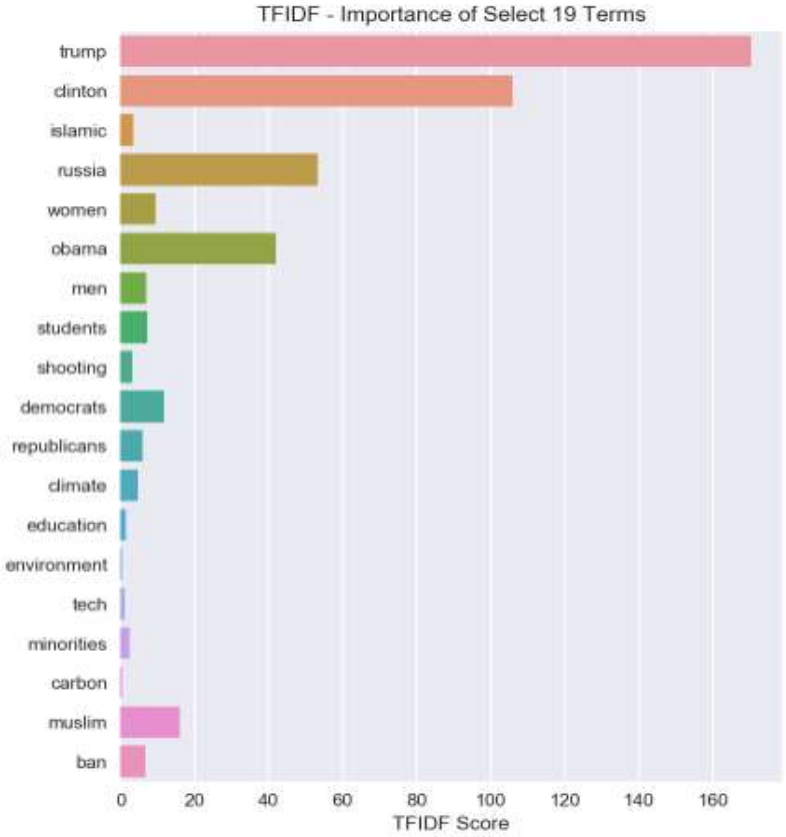


Fig 1. TF-IDF for fake news dataset

Below is TF-IDF results for real news

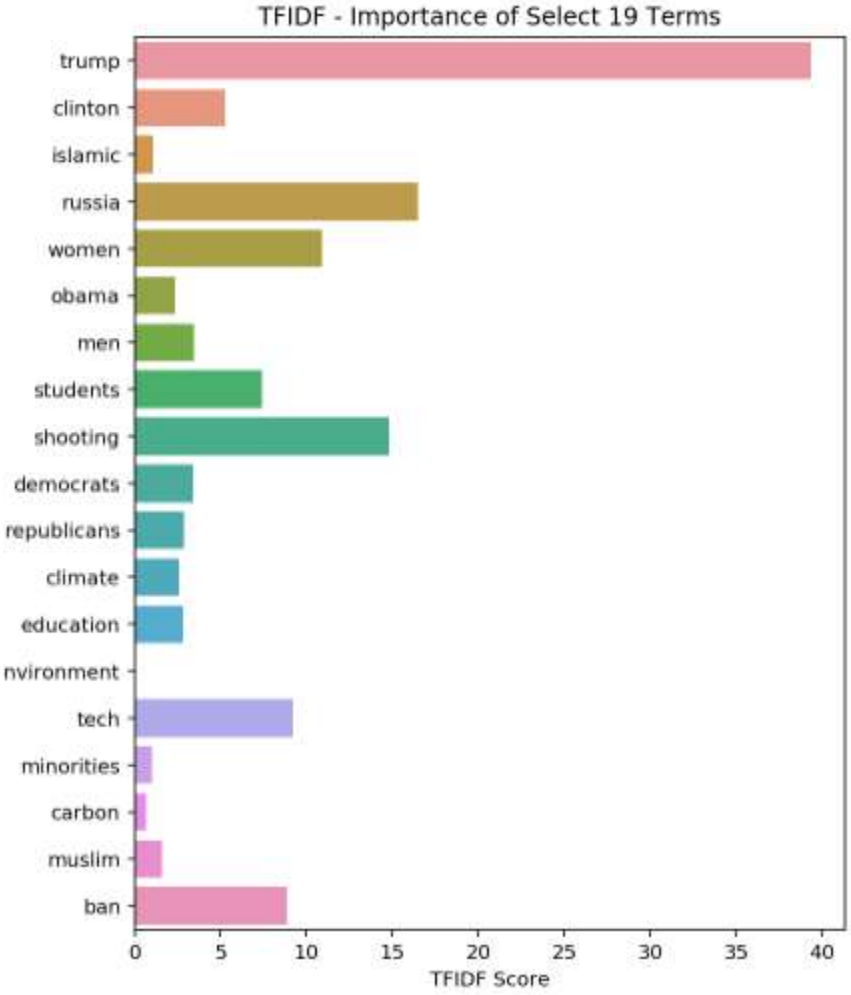
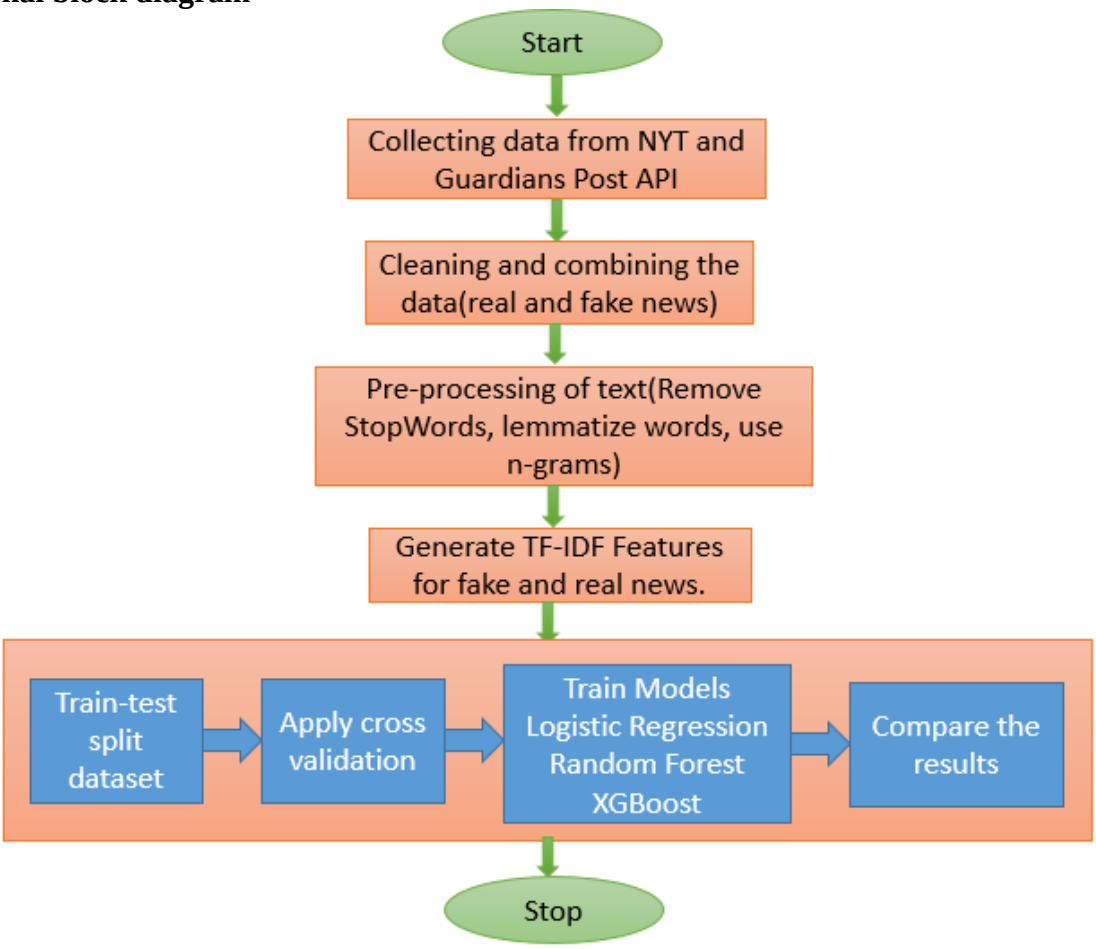


Fig 2. TF-IDF for real news dataset

e. Classification of Data- We used Logistic Regression, Random Forest, and XGBoost classifier. Logistic regression is a simple algorithm whereas Random Forest and XGBoost are more advanced. We used scikit-learn library to train the Logistic Regression and Random Forest classifier and xgboost to train XGBoost classifier. We split the dataset into training, test and validation sets. The testing set is used to test after the models were trained. The validation set is used to select hyper parameters. We used grid search to perform cross validation and hyper parameter selection.

Functional block diagram



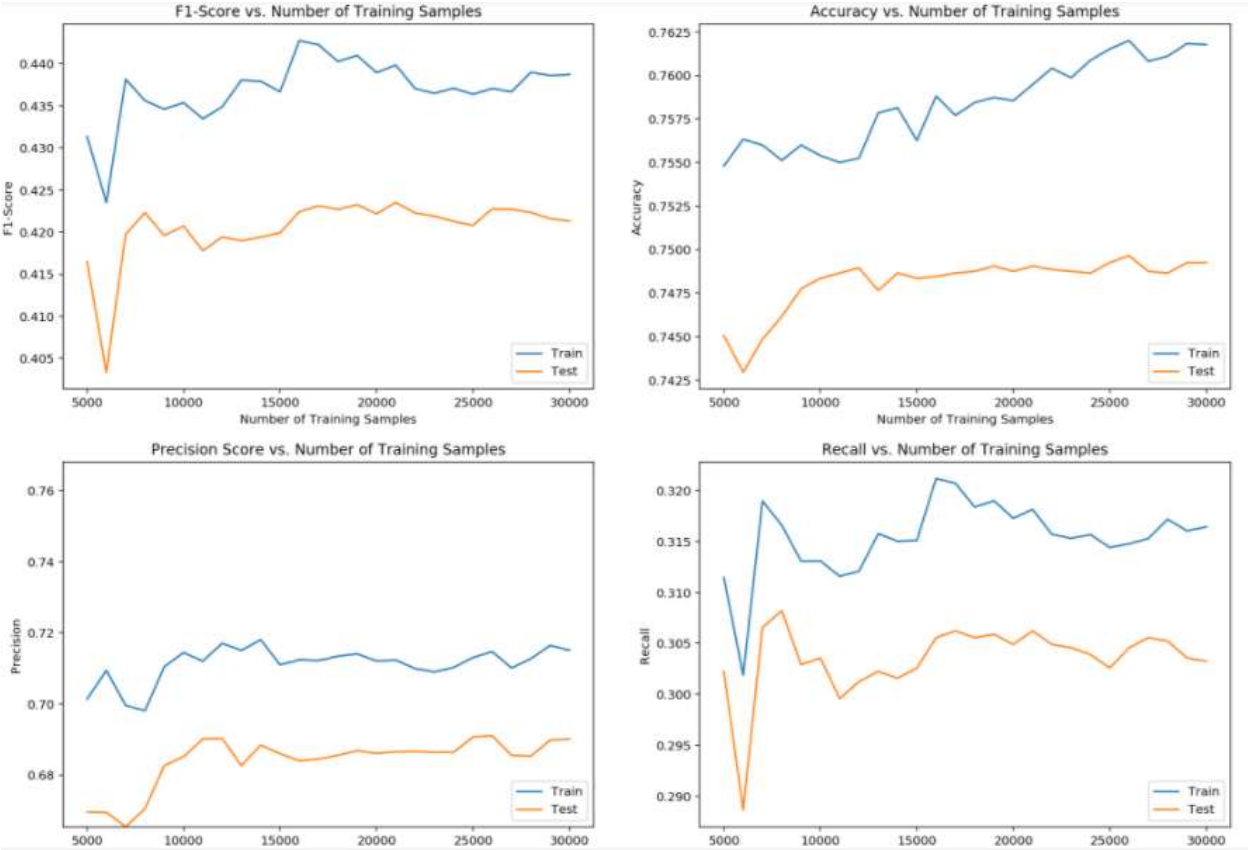
6. Experiments and Results

We trained the dataset on three machine learning algorithms - **Logistic Regression, Random Forest** and **XGBoost Classifier** to detect the fake news. All algorithms are applied for both headlines and body of the news. We then analyzed the performance from each of those algorithms and compared the performances to determine the best algorithm. We evaluated the performance of the models based on F1 score and accuracy of the model. The F1 score helps strike a balance between precision (fake articles classified correctly over the total number of articles predicted as fake) and recall (the proportion of fake articles classified correctly). We used the F1 metric as our optimization parameter when using cross-validation to tune our hyper parameters. Below table shows the F1 score and accuracy for all three classifiers.

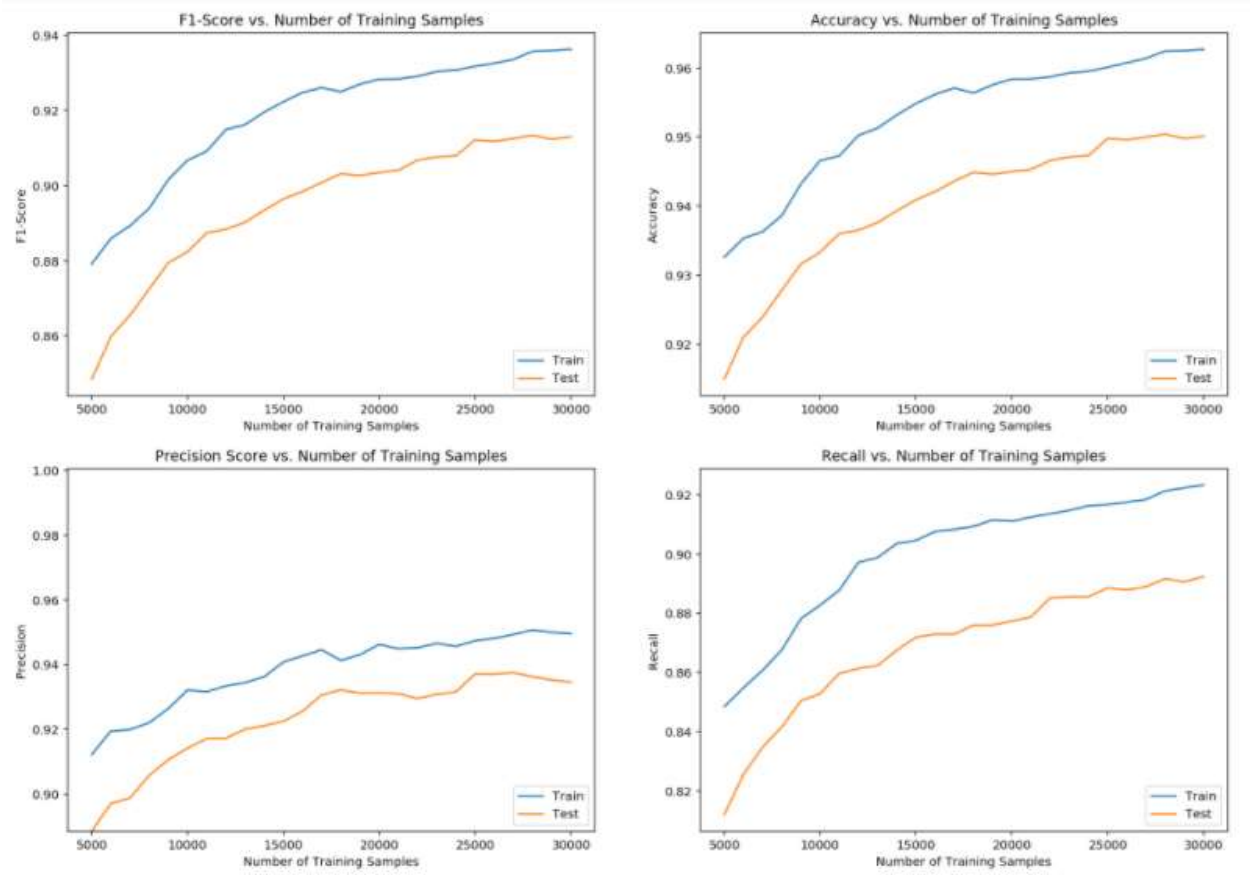
Model	**Headlines of the news**		**Body of the news**	
	F1 Score	**Accuracy Score**	**F1 Score**	**Accuracy Score**
Logistic Regression	63.73%	75.44%	**94.66%**	**95.58%**
Random Forest	**64.39%**	75.87%	91.62%	93.34%
XGBoost	61.75%	**75.93%**	89.98%	91.73%

Table 1: Models performance with TF-IDF features

We utilized cross-validation and a grid search to find the best parameters for the TF-IDF algorithm for each individual model. We found that Logistic regression produced the best results among the other models when used body of the news using TF-IDF to convert our text to features. The Logistic Regression model was much faster to train, which is important from a time-complexity standpoint when evaluating model performance.
We found that all the three models perform better on body of the news article. We generated below learning curves using the headline, which shows the bias-variance trade off. In this graph, we can see the difference train and test curves are far apart, which shows us the level of bias is high.



We generated below learning curves using body of the news articles. But using the body, we can see that there is low bias and low variance and we drive the conclusion as more data helps to improve the metrics.



7. Conclusion

In this work, we addressed the task of identifying fake news. The rise of fake news has become a major problem, which even the tech giants like Google and Facebook are finding difficult to solve. We introduced two genuine news datasets, one from NYT and other from Guardian Post and one fake news dataset from Kaggle. We used headline and bodytext from news articles to classify news into fake and real. We trained our dataset using three classification models - Logistic Regression, Random Forest, XGBoost. All these models perform almost equally but better than human ability to spot the fake content on body of news article. Some of the most frequently occurring terms cause a problem in classifying. For example, words like Trump, Clinton, Russia, Obama, etc, occur very often in our fake dataset corpus. Therefore, our model might have trouble classifying new real articles about those subjects correctly because they are so prevalent in fake news.

Future work include the addition of features such as the source of the news, including any associated URLs, the topic (e.g., science, politics, sports, etc.), publishing medium (blog, print, social media), country or geographic region of origin, publication year, as well as linguistic features which not exploited in this project—use of capitalization, fraction of words that are proper nouns (using gazetteers), and others.

References

[1] Rubin, Victoria L., Niall J. Conroy, and Yimin Chen. "Towards News Verification: Deception Detection Methods for News Discourse". *Hawaii International Conference on System Sciences.* Web.

[2] Volkova, *Svitlana, et al. "Separating Facts from Fiction: Linguistic Models to Classify Suspicious and Trusted News Posts on Twitter". Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). Web.*

[3]Pérez-Rosas, Verónica, et al. "Automatic Detection of Fake News." *arXiv preprint arXiv:1708.07104* (2017)Web.

[4] https://www.kaggle.com/mrisdal/fake-news/data