Detection of Fake News Using MNB, Passive Aggresive Classifier and LSTM

Maman Yusuf Khan
Department of CSE,
Islamic University of Technology
Gazipur, Bangladesh
mamanyusuf@iut-dhaka.edu

Sadik Yasin Eftee
Department of CSE,
Islamic University of Technology
Gazipur, Bangladesh
sadikyasin@iut-dhaka.edu

Tasfia Tahsin
Department of CSE,
Islamic University of Technology
Gazipur, Bangladesh
tasfiatahsin@iut-dhaka.edu

***Abstract*—One of the rising global crises is dealing with fake news. Often meant to create chaos and cause harm to others, the spread of fake news is excessively fast. Considering this ongoing problem, this program experiments with three different machine learning techniques and models to classify fake news. Using these three models, we got an accuracy score of 95.8 percent using Passive Aggressive Classifier making it the highest among the models used, with LSTM giving an accuracy score of 91.8 percent and Multinomial Naive Bayes Algorithm giving an accuracy score of 90 percent.**

***Index Terms*—News, Fake News, Detection, TF-IDF Vectorizer, One Hot Encoding, Multinomial Naive Bayes, Passive Aggressive Classifier, LSTM**

I. Introduction

A. Problem Statement

News is a very reliable way to be kept updated about everything that is going on around us. With the rise of social media and other online platforms, we happen to come across various types of news everyday. But with this rise in popularity, there comes an issue with authenticity. As much as a news article can be real, there is even a higher probability that it is a hoax. Therefore, in recent times, the trust in the media is at an all time low. In 2012, due to a post from a fake facebook account, 25 thousand people joined in abolishing Buddhist Temples in Ramu, Bangladesh [1].

B. Domain

Our Application uses three different Machine Learning Models to produce a result to check which of these models give the most accurate prediction. The domain of the test is English Real and Fake News.

C. Challenges

Certain challenges for this project is to prepare a proper impure dataset to train the model with the best possible spread of data. As the machine learning model only deals with categorical features, it was a challenge to extract features to convert it into one hot encoding. Another possible challenge is how to apply feature engineering to the raw data so that they turn into something meaningful for our model for a better score.

II. Background Study

A. Overview of the Project

An Overview of our project is that it aims to solve the issue of questioning the authenticity of a news article using a machine learning model. Some of the term that we need to be familiar with to have a better understanding of our project is:

TF (Term Frequency): The number of times a word appears in a document is its Term Frequency. A higher value means a term appears more often than others, and so, the document is a good match when the term is part of the search terms.

IDF (Inverse Document Frequency): Words that occur many times in a document, but also occur many times in many others, may be irrelevant. IDF is a measure of how significant a term is in the entire corpus.

One Hot encoding: In one hot encoding, each category is represented as a binary vector, with a 1 in the position corresponding to the category and 0s in all other positions. For example, if there are three categories (A, B, and C), the one hot encoding of category A would be [1, 0, 0], the one hot encoding of category B would be [0, 1, 0], and the one hot encoding of category C would be [0, 0, 1].

Stop Words: Stop words are common words in a language that are generally not useful for information retrieval or natural language processing tasks. They are usually removed from the text before further processing because they do not add any meaningful information and can interfere with the effectiveness of some algorithms.

B. Related Works

Many Researchers have previously worked with various Machine Learning techniques to come up with a solution to detect fake news. A. S. Sharma et al. proposed a hybrid extraction technique from text documents, combining Word2Vec and TF-IDF [2] which may detect with standard CNN architecture whether a text document is satire or not with a precision of more than 96 percent. Similarly, Alrubaian et al. used a combination of random forest, naive Bayes and decision trees to detect tweets containing malicious information [3]. Hakak et al. presented an ensemble classification model for detecting fake news that outperforms the current state-of-the-art models in terms of accuracy. The proposed technique collects key

features from fake news datasets, which are then identified using an ensemble model that combines three common machine learning models: decision tree, random forest, and extra tree classifier. [4]

C. Data Collection and Feature Analysis

The Dataset is collected from Kaggle. The dataset contains the following attributes:

- 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; could be incomplete
- label: a label that marks the article as potentially unreliable with 1 being reliable and 0 being unreliable.

We worked with a total of 18258 news. Among these, 10361 of them were labeled as fake and 7924 of them were labeled as real. The percentage of Real News was 43.3 percent and the percentage of Fake News was 56.67. Hence we can see that the dataset is quite impure so it would be good for the learning process.

For both Multinomial naive Bayes and Passive Aggressive Classifiers, we used the text column of the dataset and for the LSTM, we used the title column of the dataset as the title would take less time due to it being shorter than the test. We want to use these texts and titles as features to be transformed into meaningful numerical feature to be fit into the models.

The label column is extracted as the output that we want our model to predict. The other columns are dropped as they are not required in training the model.

D. Description of the Models

The Models that are used here are:

- **Multinomial Naive Bayes**: Multinomial Naive Bayes is a classification algorithm that is based on the Naive Bayes theorem, which states that the probability of an event occurring can be calculated by considering the independent probabilities of each of the individual factors that contribute to the event.
  In the context of classification, the Naive Bayes theorem can be used to predict the class of an example (such as a document or image) based on the features (such as words or pixels) that it contains. The ”multinomial” part of the name refers to the fact that this variant of the Naive Bayes algorithm is used for classification problems where the features are discrete and multivariate (i.e., there are multiple features).
- **Passive Aggressive Classifier**: The passive-aggressive classifier uses a type of online learning algorithm that can be used for classification tasks. It is an example of a ”lossy” learning algorithm, meaning that it makes a trade-off between the accuracy of the model and the speed at which it can be trained.
  The passive-aggressive algorithm works by updating the model’s parameters based on each new training example it sees. If the model correctly predicts the class of the example, the parameters are not updated. However, if the model makes an incorrect prediction, the parameters are updated in such a way as to try to correct the mistake.
  The passive-aggressive algorithm has a number of interesting properties that make it attractive for certain applications. For example, it can be used to learn a linear model in a kernel space, meaning that it can learn a non-linear decision boundary even if the original data is not linearly separable. It is also relatively simple to implement and can be used with large datasets.
- **Long Short Term Memory (LSTM)**: Long Short-Term Memory (LSTM) is a type of Recurrent Neural Network (RNN) that is capable of learning long-term dependencies in data. LSTMs are particularly useful for tasks that involve sequential data, such as language translation, language generation, and time series forecasting.
  An LSTM consists of three gate mechanisms (input, output, and forget gates) that control the flow of information into and out of the LSTM cell. The gates allow the LSTM to ”remember” certain information for long periods of time, while also being able to forget unnecessary information. This ability to selectively retain and forget information makes LSTMs particularly effective for modeling sequential data.
  LSTMs are often used in combination with other machine learning techniques and are widely used in natural language processing.

III. Implementation

A. Overview of the Experiment

The experiment has been conducted using Python Notebook using the dataset stated. The libraries used were: ”pandas” to import the csv file, ”matplotlib” to plot, ”sklearn” to split train and test data and use the Multinomial naive Bayes and Passive Aggressive Model, ”tensorflow” to implement the LSTM model and its layers. After training the models, their accuracy, precision, recall and f1 scores have been calculated to come to a conclusion about which model works the best for this.

B. Feature Engineering

The following steps were done as part of the feature Engineering:

- The first part of Feature Engineering that has been done is by removing the missing values from the dataset.
- The independent and dependent features have been extracted from the dataframe and stored
- The indexing has been reset to make sure there are no errors after removing the missing values.
- The textual part of the independent feature have undergone procedures to get rid of all the characters other than the letters of the English Alphabet. They have also been converted to lowercase and split into words.
- The stopwords have been removed and then the remaining words have been stemmed by removing the commoner morphological and inflexional endings.

- For Multinomial naive Bayes and Passive Aggressive Algorithm, we use the 'text' column and apply all the feature engineering above. Then, we take the top most frequently used words and convert the list we got into an array. As a result, each feature is converted into an independent feature vector or the TF-IDF Vector. If the term denoted as 't', a particular document as 'd'and the whole document as 'D', then the formula [5] is,

$$tfidf(t, d, D) = tf(t, d) \cdot idf(t, D)$$

  Here, tf (t, d) is the frequency of 't' in 'd'
  idf (t, D) is how 't' is common or rare across 'D'
- For Long Short Term Memory (LSTM), we use the 'title' column and apply all the feature engineering above. The datas are then encoded using 'One Hot Encoding'. Since the encoded titles are of different length we bring them to a common fixed length of 20 i.e., representing them in an embedded form. We then build a neural network model with an Embedding Layer, LSTM Layer, and a dense layer with sigmoid function as activation function.

C. Train and Test Set Generation

The train and test split generation has been done using the "sklearn" library. 33% of the data has been set for testing.

D. Running the Classifier

- For Multinomial naive Bayes, we fit the training set on the Multinomial naive Bayes model from "sklearn" library.
- For Passive Aggressive Classifier, we fit the training set on the Passive Aggressive model from "sklearn" library.
- For LSTM, we fit the training set on the LSTM model from "Tensorflow.keras" library with the testing set as validation and number of epochs as 10.

IV. Result Analysis

A. Overview

For Analysis of the result, we need the following terms:

Accuracy: Accuracy is a measure of how well a model is able to correctly predict the output for a given set of inputs. Accuracy is calculated by dividing the number of correct predictions made by the total number of predictions made.

Precision: Precision is a measure of the accuracy of a model's predictions. It is defined as the number of true positive predictions (number of times the model correctly predicts the positive class) made by the model divided by the total number of positive predictions made.

Recall: Recall is a measure of the ability of a model to correctly identify all instances of the positive class. It is defined as the number of true positive predictions made by the model divided by the total number of actual positive instances in the data.

F1 Score: The F1 score is a metric that is used to evaluate the performance of a model. It is a combination of precision and recall, and is defined as the harmonic mean of precision and recall.

$$\text{Accuracy} = \frac{(TP + TN)}{(TP + FP + TN + FN)}$$

Fig. 1. Formula of Accuracy.

$$\text{Precision} = \frac{TP}{(TP + FP)}$$

Fig. 2. Formula of Precision.

$$Recall = \frac{True\ Positive(TP)}{True\ Positive(TP) + False\ Negative(FN)}$$

Fig. 3. Formula of Recall.

$$F_1 = 2 * \frac{precision * recall}{precision + recall}$$

Fig. 4. Formula of F1 score.

B. Confusion Matrix

The Confusion Matrix has been constructed using the testing values and graphically represented using the "sklearn" library for a better analysis.

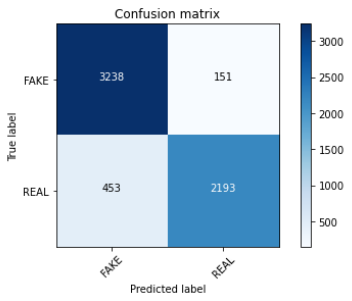

Fig. 5. Confusion Matrix of Multinomial Naive Bayes

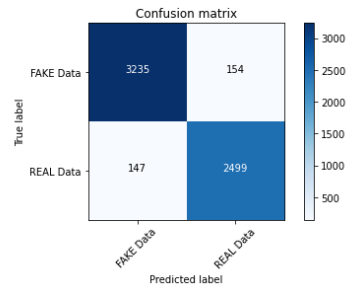

Fig. 6. Confusion Matrix of Passive Aggressive Algorithm

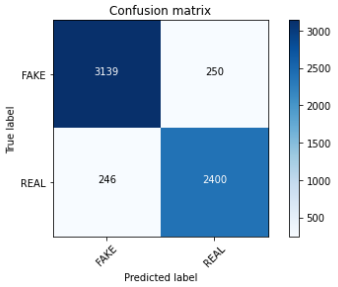

Fig. 7. Confusion Matrix of LSTM

C. Result Analysis

	Accuracy	Precision	Recall	F1 Score
Multinomial naive Bayes	0.900	0.829	0.879	0.936
Passive Aggressive Classifier	0.950	0.944	0.943	0.942
Long Short Term Memory	0.918	0.906	0.907	0.906

From the results, we can see that the Passive Aggressive Classifier can give the most accurate results among all the classifiers. It can also predict the positive class better than the other classifiers due to the high precision and recall value. The F1 score is also the highest in Passive Aggressive Classifier.

V. Conclusion

Thus we can conclude from our research that the Passive Aggressive Algorithm is doing somewhat better than the other models in predicting whether a news is real or fake. Hopefully, our research of this domain can extend to support news from languages other than English and more enhanced state-of-the-art models can be used for the classification process of the news with a large dataset available to us.

References

[1] Hossain, M.Z., et al., BanFakeNews: A dataset for detecting fake news in bangla.arXiv preprint arXiv:2004.08789, 2020.

[2] . S. Sharma, M. A. Mridul, and M. S. Islam, “Automatic detection of satire in bangla documents: A cnn approach based on hybrid feature extraction model,” arXiv preprint arXiv:1911.11062, 2019.

[3] Alrubaian, M., Al-Qurishi, M., Hassan, M. M., and Alamri, A. 2018. “A Credibility Analysis System for Assessing Information on Twitter,” IEEE Transactions on Dependable and Secure Computing (15:4),Institute of Electrical and Electronics Engineers Inc., pp. 661–674.(https://doi.org/10.1109/TDSC.2016.2602338).

[4] Hakak, S.; Alazab, M.; Khan, S.; Gadekallu, T.R.; Maddikunta, P.K.R.; Khan, W.Z. An ensemble machine learning approach through effective feature extraction to classify fake news. Future Gener. Comput. Syst. 2021, 117, 47–58.

[5] Wikipedia. Tfidf - wikipedia. [Online]. Available: https://en.wikipedia.org/wiki/Tf%E2%80%93idf