International Journal of Recent Advances in Multidisciplinary Topics
Volume 3, Issue 6, June 2022
https://www.ijramt.com | ISSN (Online): 2582-7839

Fake News Detection Using Machine Learning

Malsawmsanga Sailo[1], C. Lallungmuana[2*]
[1,2]Student, Department of Computer Engineering, Mizoram university, Aizawl, Mizoram, India

***Abstract*: In our post modern era where majority of people have access to the internet, many people relies on various online resources for news. This can have a positive and negative effect, on the positive side people have access to the latest news much quicker than ever before, on the negative side this type of news are unverified and can be a fake news based on a single person's opinion or view of the situation, or it can be a lie to sway others view of the situation. In this paper, we aim to detect the fake news using Machine Learning.**

***Keywords*: Fake news, internet, social media, classification, machine learning.**

1. Introduction

As the internet becomes more and more accessible, we spent an increasing amount of our lives interacting through social media paltforms such as Facebook, Twitter, Whatsapp, etc. more and more people tend to consume news from social media instead of traditional news organizations.

As the majority of people are busier now-a-days more than ever, we rarely have time to consume news the traditional way such as reading a newspaper. Social media sites such as Facebook, Whatsapp, Twitter, etc. makes it easier to consume news on the go. Not only does social media makes news easier to consume it is also inexpensive than the traditional ways. As we see the rise of news and information in social media, one question occurs, which is whether the given news is Real News or Fake News.

As we see the growth of news or information on social media, it is hard to verify the credibility of the source of information and whether the given information is real or fake. There are also people who tends to post misleading information to sway other people's opinion through social media. In the recent elections of India, there was a lot of discussion regarding the credibility of different news report, as some news media reports not only using facts but also mixing their opinions.

In this paper we proposed a methodology to create a model whether a news is fake or real based on the words it contains, by applying supervised machine learning algorithms on the dataset. We vectorized the data after pre-processing the dataset and feed it to the algorithm. We propose to create the model using different classification algorithms. The model then will test the unseen data and try to predict if the data is fake or real.

There are various form of news i.e. audio, video, image, and text. In this project we will focus on a detecting a fake news in text form by using Natural Language Processing and different classification algorithms.

2. Methodology

In this project we will classify a fake news from a real news using an existing dataset from Kaggle.com. We will be using Python Jupyter Notebook, Scikit-Learn, Natural Language Toolkit to implement the code.

We will be using Term Frequency- Inverse Document Frequency to vectorized our dataset, and Regular Expression to remove unwanted data, and Natural Language Toolkit to remove stop words and lemmatize the data.

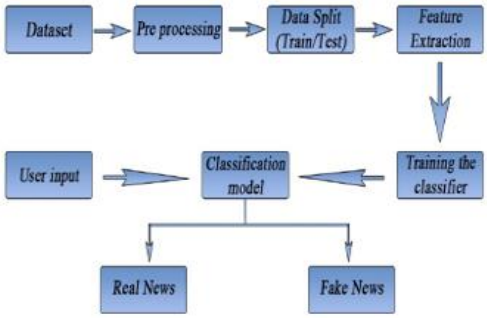

Fig. 1. Workflow

3. Implementation

A. Data Collection

The first step in this process is the data collection. The machine learning algorithm used in this project is supervised learning. Machine learning is said to be supervised if the model is trained on a dataset which contains the input and output parameters. To train our model we have taken the dataset from https://www.kaggle.com/datasets/clmentbisaillon/fake-and-real-news-dataset?select=Fake.csv for fake news and

https://www.kaggle.com/datasets/clmentbisaillon/fake-and-real-news-dataset?select=True.csv for real news

B. Pre-Processing

The second step in this process is the pre-processing of the data. Since, we have two datasets one for fake news and one for real news, both of the datasets have the attributes `title`, `text`, `subject`, and `date`, we drop the unnecessary attributes from the datasets and add a label attribute and concatenate the dataset into one.

In pre-processing we remove stop words from the data.

*Corresponding author: lallungmuanachhante11@gmail.com

Stopwords are words which are common and doesn't have much meaning and don't disturb the correctness of the information such as a, an, is, the, etc.

After this, we apply lemmatization to our data. Lemmatization is the process of turning a word into its canonical form. e.g., since words like playing, played, players all have the same meaning, they will be treated as one word `play`. This process helps to reduce the feature dimensionality and increase the efficiency of the model.

After doing the above process, we remove a non-text, url links, and other unwanted data using Regular Expression, and lowercase the data to reduce the feature dimensionality.

C. Data Split (Train/Test):

In the third step, we will split the data randomly for training and testing the model. For this process we will be using SKlearn train_test_split module from sklearn.model_selection to split our data.

D. Feature Extraction

In the fourth step, we have feature extraction. Since, a machine learning algorithm understand only numerical values we need to transform the text into something that a machine can understand. We use natural language processing to transform the text into a meaningful vector of numbers for the machine. In this process we will be using Term Frequency-Inverse Document Frequency (TF-IDF) for feature extraction.

TF (Term Frequency): It is the number of terms a word appears in the document.

IDF (Inverse Document Frequency): It measures the importance or uniqueness of a term in the document.

$$\mathbf{Tf(t,d)} = \frac{\text{Number of times t occurs in document `d`}}{\text{Total word count of document `d`}}$$

$$\mathbf{IDF(t,d)} = \frac{\text{Total number of documents}}{\text{Number of documents which term t in it}}$$

$$\mathbf{TFIDF\ (t,d) = TF\ (t,d) * IDF\ (t)}$$

4. Classification

Before training our model we first split our dataset into two parts, 80% for training the model and 20% for testing the model. For classification algorithms we use the following algorithms:

A. Logistic Regression

Logistic Regression in Machine Learning, despite its name, is a classification model rather than a regression model. It is one of the more prominent binary classification algorithms.

Logistic Regression is a specific type of Generalized Linear Model (GLM). Generalized Linear Models are a generalization of the concepts and abilities of regular Linear Models.

In machine learning Logistic Regression is a model used for binary classification. A binary classification is a classification where the predicted output can have either one of two values. In example, a binary classification can be used to predict whether an email is a spam or not, or whether a patient has cancer or not. In this project we will be using it to predict whether a given data is fake or not.

In this project we will be using the Logistic Regression module from SKLearn to classify fake news using an existing dataset. We perform hyperparameters tuning to get the best results from our dataset.

Logistic Regression uses a sigmoid function to transform the output to a probability value, using the equation:

$$\sigma(x) = \frac{1}{1+e^{-x}}$$

Cost function is an error representation in machine learning. It shows how a model is predicting compare to the original given dataset. The lesser the cost function, the more accurate the model, to calculate the cost function for global minimum, we use the equation:

$$\text{Cost} = \frac{1}{m}\sum_{i=1}^{m} [y \log(y_{pred}) + (1-y)\log(1-y_{pred})]$$

where m is the number of iteration and,

$$y_{pred} = \sigma(w^T x + b)$$

B. Decision Tree Classifier

A Decision Tree can perform classification and regression tasks. When a decision tree classifies things into categories, it is called a classification tree and when a decision tree predicts numeric value it is called a regression tree. Since, we are going to classify fake news in this project, we will be using the decision tree using Decision Tree Classifier module form Sklearn.

Decision tree first calculates the entropy of the known output using the equation:

$$\text{Entropy} = \sum - P_i \log(P_i)$$

Then, calculates the information gain value of all the attributes using the equation:

$$IG = E(\text{parent}) - \text{Summation } w_i\ E(\text{child}_i)$$

Where E is the entropy of the attributes and wi is the weight of the child, which is a relative size of a child node w.r.t the parent node. The greatest information gain value attribute will be used as the parent node.

The model compares every possible split and takes the one with the maximum information gain.

It is a greedy algorithm, it selects the current best split that maximizes information gain, it does not backtrack and change a previous split. So, all the following split will depend on the current one.

C. Random Forest Classifier

Random Forest is build from decision tree. Decision Tree works great with data used to train the model, but they are not

flexible when it comes to classifying new samples. Random Forest combine the simplicity of decision tree with flexibility resulting in a vast improvement in accuracy.

Steps to make a random forest:

i) Create bootstrapped dataset. To create a bootstrapped dataset the same size as the original, we just randomly select samples from the original dataset. The important detail is that we are allowed to pick the same sample more than once.

ii) Create a decision tree using the bootstrapped dataset, but only used a random subset of variables (or columns) at each step.

iii) Go back to step i) and repeat. Make a new bootstrapped dataset and build a tree considering a subset of variables at each step.

Using a bootstrapped sample and considering only a subset of the variables at each step a results in a wide variety of trees. The variety is what makes random forest more effective than individual decision trees. Generally, about one-third of the original data does not end up in the bootstrapped dataset, this are called `Out-Of-Bag-Dataset`. Ultimately, we can measure how accurate our random forest is by the proportion of Out-Of-Bag samples that were correctly classified by the random forest. The proportion of Out-Of-Bag samples that were incorrectly classified is the `Out-Of-Bag-Error`.

D. Multinomial Naive Bayes

It is one of the algorithms under Naive Bayes. It is based in the Bayes theorem and predicts the tag of a text. It calculates the probability of each tag for a given sample and then gives the tag with the highest probability as output. A Naive Bayes classifier is naive in that it treats all words the same, which means that each feature being classified is not related to any other feature. Treating all word orders is very different from how we communicate. It ignores all grammar rules and common phrases, it instead treats them like a bag of words. Although Naive Bayes is naive, it performs surprisingly well. By ignoring relationships among words, we'd say Naive Bayes has high bias, but because it works well in practice, it has a low variance.

To understand Multinomial Naive Bayes lets first take a look at the Bayes Theorem, which calculates the probability of an event occurring based on prior knowledge of conditions related to an event using the equation:

$$P(A|B) = P(A) * P(B|A)/P(B)$$

Where we are calculating the probability of class A when predictor B is already provided.

P(B) = prior probability of B
P(A) = prior probability of class A
P(B|A) = occurrence of predictor B given class A probability

This formula helps in calculating the probability of the tags in the text.

The Multinomial Naive Bayes use the equation:

$$P(\text{word}|\text{class}) = \frac{\text{number of occurrence of word}}{\text{Total number of words in class}}$$

To calculate the probability or likelihood of an individual word. To calculate the probability of a data being fake, it uses the equation:

$$P(\text{class}_i) * P(\text{word}_1) * P(\text{word}_2) * ... * P(\text{word}_n),$$

Where i is the number of class and word is each word present in the data. The data will be classified to the class with the highest value result.

E. Gradient Boosting Classifier

Gradient Boost is used for classification. It has a lot in common with Logistic Regression. When using Gradient Boost for classification, the initial prediction for every individual is the log (odds). The easiest way to use the log(odds) for classification is to convert it to a probability, and we do that using the logistic function.

$$\text{Probability of fake news} = \frac{e^{\log(\text{odds})}}{1+e^{\log(\text{odds})}}$$

We can measure how bad the initial prediction is by calculating pseudo residuals, the difference between the observed and the predicted values.

Residual = (Observed - Predicted)

We can measure how bad the initial prediction is by calculating pseudo residuals, the difference between the observed and the predicted values.

Residual = (Observed - Predicted)

Since, the predictions are in terms of log (odds) and the leaf is derive from a probability, we can't just at log (odds) Prediction without transformation. When we use Gradient Boost for classification, most common transformation is the following formula:

$$\frac{\Sigma \text{Residual}_i}{\Sigma\ [\text{Previous Probability}_i * (1-\text{Previous Probability}_i)]}$$

After transforming we calculate the log(odds) Prediction using the equation:

log(odds) Prediction = Previous Prediction + (Output value from tree*Scaled by learning rate)

After calculating for each data, we calculate the probability of the data being a fake news again. We repeat the above steps multiple times.

5. Results

The results of the model achieved by using the above algorithms is shown using its accuracy score and confusion matrix, in confusion matrix the real news is represented as 0, and fake news is represented as 1, the results are as shown below.

A. Logistic Regression

The Logistic Regression model shows 99.58% accuracy and the confusion matrix is,

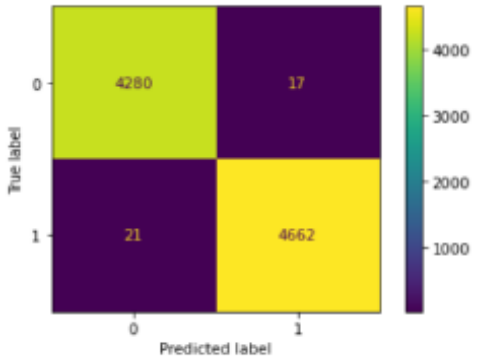

Fig. 2. Confusion matrix of logistic regression model

B. Decision Tree Classifier

The decision tree classifier model shows 99.73% accuracy and the confusion matrix is,

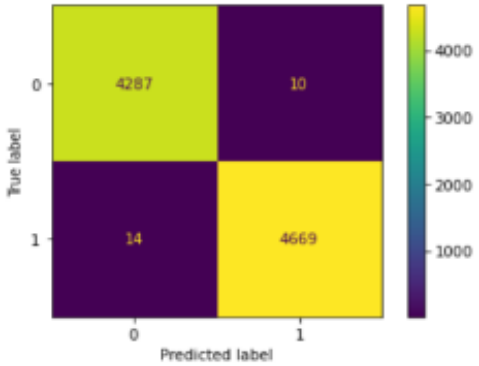

Fig. 3. Confusion matrix of decision tree classifier model

C. Random Forest Classifier

The random forest classifier model shows 98.46% accuracy and the confusion matrix is,

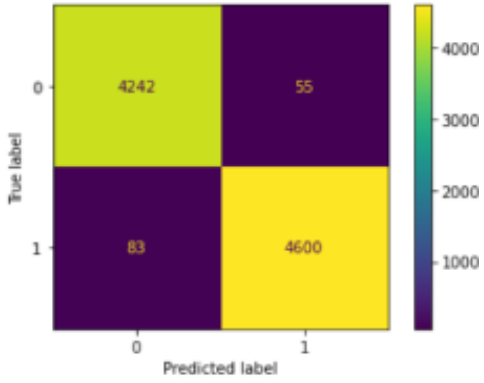

Fig. 4. Confusion matrix of random forest classifier

D. Multinomial Naive Bayes

The multinomial naive bayes model shows 94.78% accuracy and the confusion matrix is,

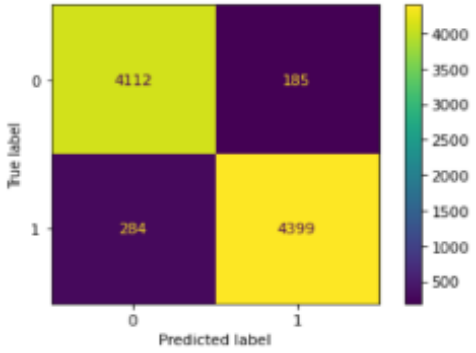

Fig. 5. Confusion matrix of multinomial naive bayes model

E. Gradient Boosting Classifier

The gradient boosting classifier model shows 99.82% accuracy and the confusion matrix is,

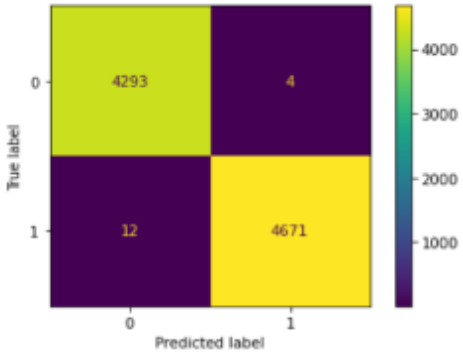

Fig. 6. Confusion matrix of gradient boosting classifier model

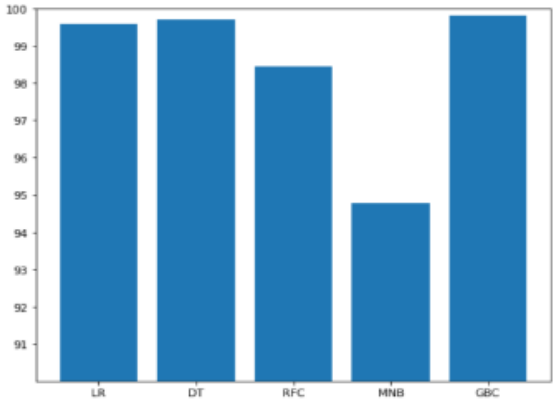

Fig. 7. Accuracy results of all the algorithms

6. Conclusion

In this project we try to classify fake news using different algorithms. In this day and age, the majority of news we get are from online such as social media. Whatsapp's forwards are also major source. After training our model, we classify a fake news of a different dataset and got the accuracy score for Logistic Regression 52.8%, Decision Tree Classifier 48.31%, Random Forest Classifier 53.3%, Multinomial Naive Bayes 52.55%, Gradient Boosting Classifier 47.44%.

The best model with the highest accuracy is used to classify a fake news, as evident from above our best model came out to be Random Forest Classifier with accuracy of 53.3% on an unknown data. Although Random Forest Classifier and Gradient Boosting Classifier was the highest on the training dataset, Gradient Boosting Classifier score the lowest and Decision Tree Classifier score the second lowest among the algorithms we used on an unknown data.

References

[1] Uma Sharma, Sidarth Saran, Shankar M. Patil "Fake News Detection using Machine Learning Algorithms," 2021.
[2] Nicole O'Brein, "Machine Learning for detection of Fake News."
[3] Mayur Bhogade, Bhushan Deore, Abhisek Sharma, Omkar Sonawane, "A Research Paper on Fake News Detection," 2021.
[4] Chih-Chien Wang, "Fake News and Related Concepts: Definitions and Recent Research Development."
[5] Iftikhar Ahmad, Muhammad Yousaf, Suhail Yousaf, and Muhammad Ovais Ahmad, "Fake News Detection Using Machine Learning Ensemble Methods."
[6] Z. Khanam, B. N. Alwasel, H. Sirafi and M. Rashid, "Fake News Detection Using Machine Learning Approaches," 2021.