International Journal of Research in Engineering, Science and Management
Volume-3, Issue-6, June-2020
www.ijresm.com | ISSN (Online): 2581-5792


# Fake News Detection Using Naive Bayes and Support Vector Machine Algorithm

Shruthi S. Shetty[1*], K. B. Shreejith[2], Deekshitha[3], Dhanusha[4], K. B. Gagana[5]
[1,3,4,5]*Student, Dept. of Information Science and Engineering, Srinivas Institute of Technology, Mangalore, India*
[2]*Assistant Professor, Department of Information Science and Engineering, Srinivas Institute of Technology, Mangalore, India*
*Corresponding author: shettyshruthi731@gmail.com

***Abstract*: A huge amount of information is generated on the social media platforms with various social media formats. Huge volumes of posts are rapidly increasing on social media. When some event takes place, many people discuss it on the internet through social networking sites. They search or retrieve and discuss the news events and make it as a routine of their daily life. However, very large volumes of news contents cause the users to face the problem of information overloading during searching and retrieving. Unreliable sources of information expose people to a large amount of fake news, rumors, hoaxes, conspiracy theories and misleading news. This fake news come from the misinformation, misunderstanding or the unreliable contents with the creditability source. This makes it difficult to detect whether to believe or not if the news is a fake or a real one when the news information is received. The aim of this research paper is to attempt to tackle the growing issues of fake news, which has been always been a problem by the wide-spread use of social media. In this paper we make use of two classification models, naïve Bayes and support vector machine (SVM). This approach enables us to classify the news contents into fake or real news.**

***Keywords*: Classification, Fake news, Naïve Bayes, Social media, Support Vector Machine.**

## 1. Introduction

Fake news detection on social media has recently become an emerging research that is capturing attention. Fake news is generated on purpose to mislead readers to believe false information, which makes it difficult and non-trivial to detect based on content.

Fake news on social media has been occurring for several years; however, there is no agreed definition of the term "fake news". For better guidance of the future directions of fake news direction research, appropriate classifications are necessary. Social media has proved to be a powerful source for spreading fake news. It is important to utilize some of the emerging patterns for fake news detection on social media.

## 2. Architecture of Proposed System

This paper aims at studying the detection of fake news using machine learning program in python. It uses Natural Language Processing for detecting the fake news. It takes input like training data and test data from different sources and analyses that data and detect the news as fake or real. A model is built based on the count vectorizer or a tf-idf matrix (i.e.) word tallies relative to how often they are used in other articles in the dataset used) can help. Since this problem is a kind of text classification, implementing a Naive Bayes classifier and Support vector machine will be best as this is standard for text-based processing.

The main motive is to develop a model which is the text transformation (count vectorizer vs. tf-idf vectorizer) and choosing which type of text to use i.e. headlines vs. full text. In the next step, we extract the most optimal features for count vectorizer or tfidf-vectorizer, this is done by using a n-number of the most used words, lower casing or not, and/or phases mainly removing the stop words which are common words such as "the", "when", and "there" and only using those words that appear at least a given number of times in a given text dataset.

### *A. Naïve Bayes Algorithm*

Naïve Bayes algorithm is a supervised classification technique. This algorithm is a simple but one of the most effective techniques of classification. We will use this algorithm in our paper for detecting the fake and real news. The formula for Naïve Bayes classifier is represented as given below:

P(H/D)= P(H) P(D/H)/ P(D)

Where,

P(H/D)= Posterior Probability

P(H)= Prior Probability i.e. what we believe before we see evidence.

P(D/H)= Likelihood of seeing the evidence if our hypothesis is correct.

P(D)= Likelihood of the evidence under any circumstances.

### *B. Support Vector Machine*

Support Vector Machines (SVM) are an arrangement of related supervised learning techniques operated for grouping and classification. There are some useless words that are present in the textual data. These useless words are called as stop words. These stop words are removed from the text data and then the features are extracted successfully. After the features are extracted from the text, SVM Classifier performs

classification on the data; and defines whether the given news is a fake news or real news. The below figure shows a pictorial representation of how textual data is classified implementing SVM Algorithm. The algorithm creates a line or a hyper plane based on which it splits the data into classes. Here, in our paper, as we know the textual data will be split into 2 classes to detect if it is fake or real news.

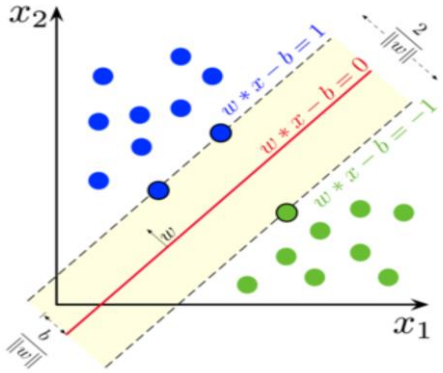


Fig. 1. SVM algorithm graph

We are training dataset of 'n' points of the form:
$(\overrightarrow{x_2},y_1),.....(\overrightarrow{x_n},y_n)$

*Pseudo code for SVM and Naïve Bayes*
*Step 1:* Extracting text from source.
*Step 2:* Preprocessing the textual data.
– Sampling information.
– Removing Stop words.
–Normalize textual data to form vector matrix using tfidf /count vectorizer.
– Generate feature matrix.
*Step 3:* Train-Test Split. (splitting the data for training and testing)
*Step 4:* Train the classifiers.
– Naïve Bayes.
– Support Vector Classifier.
*Step 5:* Test the algorithms.
*Step 6:* Evaluate performance of algorithms.

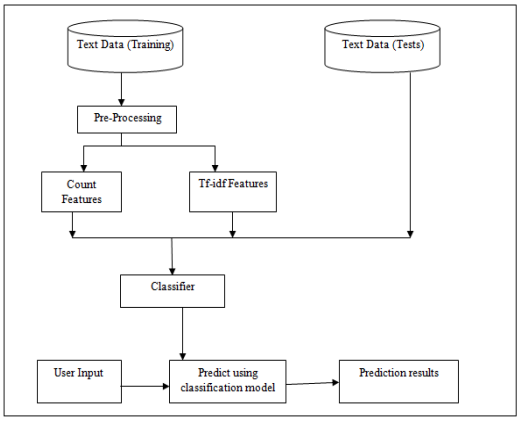


Fig. 2. Architecture of fake news detection

## 3. Methodology

The fake news detection system consists of the following modules listed below:

### A. *Pre-Processing*

The term Pre-processing the data is defined as the process of converting a data into an understandable format by cleaning it and preparing the text for classification. Texts from online contain usually lots of noise and uninformative parts such as scripts and advertisements. In addition, on word level, many words in the text do not have an impact on the general orientation of it. Keeping those words makes the dimensionality of the problem complex and hence the classification becomes more difficult since each word in the text is treated as single dimension. We learn about the hypothesis of pre-processing the data properly to reduce the noise in the text which helps to improve the performance of the classifier and speeds up the classification process, thereby aiding in real time sentiment analysis. The entire process involves several steps: online text cleaning, white space removal, expanding abbreviation, stemming, stop words removal, negation handling and finally feature selection. Features in the context of opinion mining are the words, terms, phrases that strongly express the opinion as positive or negative.

### B. *Count Features:*

The Count Featurizer allows you to convert complex categorical dimensions to simpler scalar dimensions which are easier and faster to train on while improving classification performance. It provides a simple way to do two things: tokenize a collection of text documents and build a vocabulary of known words, but also encoding new documents using that vocabulary. As a result, an encoded vector is returned with a length of the entire vocabulary and an integer count for the number of times each word appeared in the document. If an a-priori dictionary is not provided and you do not use an analyzer that does some kind of feature selection, then the number of features will be same as the vocabulary size found by analyzing the data.

### C. *Tdif (Term Frequency –Inverter Document Frequency)*

TF-IDF which stands for Term Frequency – Inverse Document Frequency is a statistical method of evaluating the significance of word in given documents. Term frequency (tf) refers to the number of times a given term will appear in a document. Inverse document frequency measures the weight of each word in the document, i.e. if the word is common or rare in the entire document. The tf-idf intuition follows that the terms that appears rarely in a document are given more importance than the terms that appear frequently in a document. The tf-idf uses the vector space modeling technique for representing text document. TF-IDF is used in document classification, text summarization and recommender systems among other use cases. Here we will look at feature extraction with tf-idf, its application in text classification and how it can

be implemented using Python-based libraries. TF-IDF is a measure that uses two statistical methods, the Term Frequency and the Inverse Document Frequency. The term frequency denoted as *tf(t,d)* is the total number of times a given term t appears in the document d against the total number of all words in the document.

### *D. Classifier*

The classifier classifies the count and tf-idf outputs based on the features. A classifier utilizes some set of training data to understand how actually given input variables relate to the class. On training a classifier accurately, it can be used to detect an unknown email. Classification is a supervised learning method where the targets are provided with the input data. A classifier can also refer to the field in the data set which is the dependent variable of a statistical model. A classification technique (or classifier) is a systematic approach which can be used to build classification models from an input dataset. Some of the examples of classifier include decision tree classifiers, rule-based classifiers, neural networks, support vector machines, and Naıve Bayes classifiers.

### *E. User Input*

An Input is nothing but any information or data that is sent to a computer for processing. Using an input device, an input or a user input is sent to a computer. User Input is one of the most important aspects of programming concepts. Every program must have some kind of user interaction, from getting a character's name for a game to asking for a password to log into a database.

### *F. Predict using classification model*

The main aim of classification is to predict the target class for each case in the data accurately. A classification model obtains some conclusions from observed values. A classification model will try to predict the value of one or more outcomes with given one or more inputs. In short we can say that classification either predicts categorical class labels or classifies data (construct a model) based on the training set and the values or class labels in classifying different attributes and uses it to classify new data. There are a number of classification models. Some of classification model examples include logistic regression, decision tree, random forest, gradient boosted tree, multilayer perceptron, Naïve Bayes and support vector machine.

## 4. Results

Fake news and real news contents are substantially different. It is the title of the contents that are strong differentiating factors between fake and real news. Fake news is nothing new. But, what is new is how easy it has become to share information – both fake and real on a massive scale. Social media platforms like Twitter, Face book etc. almost allows anyone to publish their thoughts or share stories to the world. Most people before checking the source of material that they view online tend to share it, which can lead to fake news spreading quickly or even "going viral. Using fake news detection using appropriate algorithm we can avoid all the misleading news, rumors and hoaxes. The final result will show the fake and real news.

## 5. Conclusion

Fake news is the difficult problem because it is the rumors which are too hard to identify the fact in the contents. Motivation of the rapid increase of the true and fake news requires strenuous effort for detection. It will use fact checking to solve the fake news detection problem. Using machine learning methods, fake news can be accurately identified. In this experiment, selected data collected from social media such as twitter will be profiled with different attributes. From this information, all the machine learning methods: Naïve Bayes, Neural network, Support Vector Machine, are very good at detecting fake news with high confidence. Of course, it may not represent the whole spectrum of news in real world. However, we don't have enough evidence that detecting fake news is not too difficult, at least in some selected domain. It would be difficult to say with confidence how much the result of this experiment can be applied to real-world news.

## References

[1] Niall J Conroy, Victoria L Rubin, and Yimini Chen. 2015. Automatic deception detection: Methods for finding fake news. Proceedings of the Association for Information Science and Technology.
[2] Emma Haddia, Xiaohui Liua, Yong Shib (2013). "The Role of Text Pre-processing in Sentiment Analysis". Information Technology and Quantitative Mangaement (IQTM 2013).
[3] Pérez-Rosas, Verónica, et al., "Automatic Detection of Fake News," 2017.
[4] Qin Yumeng, Dominik Wurzer and TANG Cunchen. "Predicting Future Rumours", Chinese Journal of Electronics, Vol. 27, No. 3, May 2018.
[5] Eugenio Tacchini, Gabrisele Ballarin, Marco L. Della Vedova, Stefano Moret, and Luca de Alfaro, "Some like it hoax: Automated fake news detection in social networks," 2017.