UDACITY

Fake News Detection Using Machine Learning Approaches

Udacity Nanodegree Capstone Project

Thura Aung

Abstract. Fake news detection is still in the early age of development, and there are still many challenging issues that need further investigations. It is necessary to discuss potential research directions that can improve fake news detection and mitigation capabilities. Binary classification is done by using different machine learning algorithms.

1. Definition

1.1 Project Overview

The fake news on social media and various other media is widespread and is a matter of serious concern due to its ability to cause a lot of social and national damage with destructive impacts. A lot of research is already focused on detecting it.

This paper makes an analysis of the research related to fake news detection and explores the traditional machine learning models to choose the best, in order to create a model of a product with supervised machine learning algorithm, that can classify fake news as true or fake, by using tools like python scikit-learn, NLP for textual analysis.

This process will result in feature extraction and vectorization; we propose using Python scikit-learn library to perform tokenization and feature extraction of text data, because this library contains useful tools like Count Vectorizer and Tiff Vectorizer. Then, we will perform feature selection methods, to experiment and choose the best fit features to obtain the highest precision, according to confusion matrix results.

1.2 Problem Statement

Fake news data are pervasive, and it has become an exploration challenge to consistently check the data, content, and distribution to label it as right or wrong. Many researchers have been trying to work on this problem, and they have also somehow been successful. Some have researched the field of machine learning, and some have explored deep learning. Still, no one has ever produced research in the field of sentiment analysis or sentiment information.

2. Dataset

2.1 Dataset Description and Architecture

The dataset used in this study consists of fake news and real news. Each file of the dataset consists of more than twenty thousand examples of fake news and real news. The dataset considers the title, text, subject, and date that the articles were posted, and the dataset comprises information used from the fake and real news datasets used for Ahmed, Traore and Saad [1].

2.2 Data Preprocessing

The data needs to be pre-processed before the training, testing, and modeling phases. Before moving to these phases, the real news and fake news are concatenated. In the dataset cleaning process, we removed the columns from the datasets that were not needed for processing. The punctuation and stop words were also removed. Stop-words are those words that frequently occur, such as "I, are, will, Shall, is it, etc. Uppercase letters were converted into lowercase letters. After the dataset was cleaned, it looked good and was ready for the exploration step. However, for the sake of more in-depth research, the dataset exploration was completed on both the cleaned and uncleaned data. For the exploration process, both the fake and real datasets were grouped into a data frame to make the processing easier.

Table 1 .The number of real and fake news

Article Title	**Frequency**
Fake news articles	23,481
Real (True) news articles	21,417

2.3 Data Exploration

Data exploration stage is used to explore and visualize the data to identify patterns and insights from fake and real news. We plotted various charts using Matplotlib[2] and Seaborn[3] using the Python libraries.

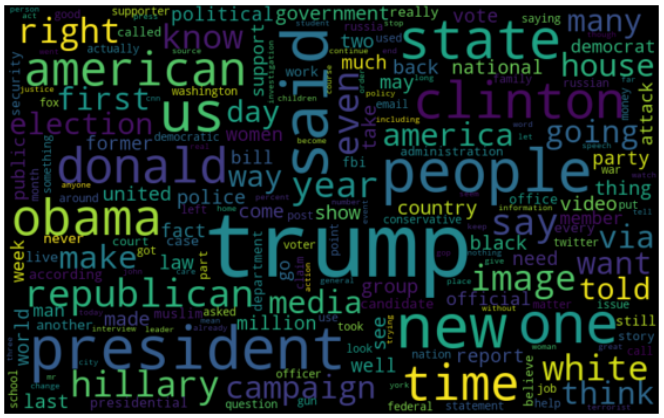

Figure 1. Word clouds comprising title words for the fake news

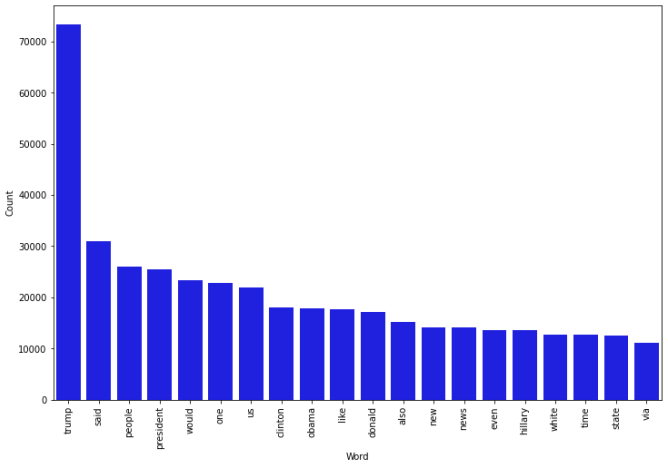

Figure 2. Frequent words in fake news

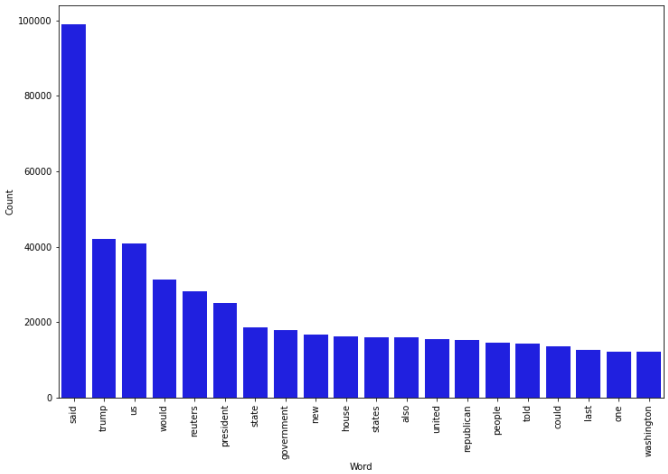

Figure 3. Frequent words in real news

3. *Methodology*

3.1 Feature extraction

A python library known as Scikit learn was used [4]. This library is perfect when performing any task with the TF-IDF model. This method includes TF-IDF vectors that represent a term's relative significance in the record or as a whole. The next factor of this method is that term frequency is very important (TF). It represents the frequency of a word occurring in the dataset (we determined the word frequency in an article when undergoing data exploration) [5].

The formula for finding the TF is

$$\mathrm{TF}(t,d) = \frac{Number\ of\ times\ t\ occurs\ in\ a\ document\ 'd'}{Total\ word\ count\ of\ document\ 'd'}$$

IDF is Inverse Document Frequency used to measure how notable a term is in the entire dataset.

$$\mathrm{IDF}(t) = \log_e\left(\frac{Total\ number\ of\ documents}{Number\ of\ documents\ with\ term\ t\ in\ it}\right)$$

TF- IDF is equal to the inverse document frequency integrated into term frequency.

$$\text{TF-IDF}(t,d) = \mathrm{TF}(t,d) * \mathrm{IDF}(t)$$

The TF-IDF model extracted the feature engineering and counted the most relevant terms from both datasets to achieve better performance. TF-IDF tokenized records and archived recurrent weightings.[6]

3.2 Machine Learning Approaches

3.2.1 Naive Bayes Classifier

This algorithm works on Bayes theorem under the assuming that its free from predictors and is used in multiple machine learning problems [7]. Simply put, Naive Bayes assumes that one function in the category has nothing to do with another. For example, the fruit will be classified as an apple when its of

red color, swirls, and the diameter is close to 3 inches. Regardless of whether these functions depend on each other or on different functions, and even if these functions depend on each other or on other functions, Naive Bayes assumes that all these functions share a separate proof of the apples

$$P(c \mid x) = \frac{P(x \mid c)P(c)}{P(x)}$$

$$P(c \mid X) = P(x1 \mid c) \times P(x2 \mid c) \times \ldots \times P(x2 \mid c) \times P(c)$$

3.2.2 Logistic Regression

The third technique that we are using to make this model work correctly is the logistic regression technique. Logistic regression in machine learning dictates that logistic regression can discover a connection among the highlights (probability) and likelihood (outcome) of a specific result. A logistic regression classifier is used when the predicting value is categorical. For instance, when predicting the value, it will give either a true or false response. Logistic regression can discover a connection among the highlights (probability) and likelihood (outcome) of a specific result [8]. The logistic regression model can be imported from the sklearn linear_model.

3.2.3 Decision Tree Classifier

As we know, this classifier is one of the best classifiers in machine learning. Decision trees are known for their non-parametric supervised learning methods that can be used for processes such as classification and regression tasks. It works in a model way [9]. Tree models where the objective variable can take a discrete arrangement of qualities are called order trees. Decision trees perform with good results and can be made quickly based on Gini index The last machine learning algorithm we will be using is the decision tree classifier. Decision trees are known for their non-parametric supervised learning methods that can be used for both processes, such as classification and regression tasks. Additionally, a decision tree may be suitable for detecting fake news [10]. First of all, it is essential to import the decision tree classifier from the sklearn tree model.

3.2.4 Random Forest Classifier

The random forest has almost the same hyperparameters as a decision tree or a sacking classifier. This technique adds more arbitrariness to the model while developing the trees. First of all, a random forest classifier is a technique that makes different choice trees and consolidates them to produce a more exact and stable prediction. The random forest has hyperparameters that are almost the same as a decision tree or a sacking classifier. This technique adds more arbitrariness to the model while developing the trees [11]. There are diverse arbitrary trees that provide worth, and worth with more votes is the genuine after effect of this classifier [12]. It can also be imported from sklearn, as was the linear model.

3.2.5 Support Vector Machine

In machine learning, support-vector machines are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Developed at AT&T Bell Laboratories by Vapnik with colleagues, it presents one of the most robust prediction methods, based on the statistical learning framework or VC theory proposed by Vapnik and Chervonenkis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier.[13]

4. Experimental Results

This section has two different sections about the experimental setup in Sections 4.1 and 4.2 is related to the results.

4.1. Experimental Setup

All four models were implemented on Google Colab, which provided a cloud environment. For this, we used python 3.5 and above. The libraries that we used for training and testing were Numpy, Pandas, Scikit learning, Natural language Toolkit (NLTK), Matplotlib, and Seaborn. We divided the dataset into the training and test set with a ratio of 80:20.

4.2 Experimental Results

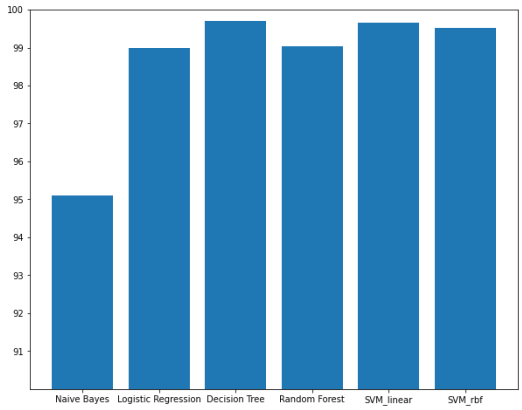

Figure 4 . Accuracy comparison of the classifiers

Table 2 . Experimental results of machine learning algorithms

Model	**Accuracy**
Naive Bayes	95.11%

Logistic Regression	98.98%
Decision Tree	99.69%
Random Forest	99.04%
Linear SVM	99.65%
SVM with RBF kernel	99.52%

Conclusion

Data preprocessing and TFIDF make machine learning classifiers more precise.Decision Tree Classifier would be the best fit on the dataset.

References

---

[1] Ahmed H, Traore I, Saad S. “Detecting opinion spams and fake news using text classification”, Journal of Security and Privacy, Volume 1, Issue 1, Wiley, January/February 2018.

[2] Hunter, J.D. Matplotlib: A 2D graphics environment. Comput. Sci. Eng. 2007, 9, 90–95.

[3] Waskom, M.L. seaborn: Statistical data visualization. J. Open Source Software. 2021, 6, 3021.

[4] Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V. Scikit-learn: Machine learning in Python. J. Mach. Learn. Res. 2011, 12, 2825–2830.

[5] Singh, A.K.; Shashi, M. Vectorization of Text Documents for Identifying Unifiable News Articles. Int. J. Adv. Comput. Sci. Appl. 2019, 10

[6] Dey, A.; Jenamani, M.; Thakkar, J.J. Lexical TF-IDF: An n-gram feature space for cross-domain classification of sentiment reviews. In Proceedings of the International Conference on Pattern Recognition and Machine Intelligence, Kolkata, India, 5–8 December 2017; pp. 380–386

[7] Jasmin Kevric et al. "An effective combining classifier approach using tree algorithms for network

intrusion detection." Neural Computing and Applications , 1051–1058. 2017.

[8] Menard, S. Applied Logistic Regression Analysis; Sage: London, UK, 2002; Volume 106.

[9] Safavian, S.R.; Landgrebe, D. A survey of decision tree classifier methodology. IEEE Trans. Syst. Man, Cybern. 1991, 21, 660–674

[10] Lyu, S.; Lo, D.C.T. Fake News Detection by Decision Tree. In Proceedings of the 2020 SoutheastCon, Raleigh, NC, USA, 28–29 March 2020; pp. 1–2.

[11] Manzoor, S.I.; Singla, J.; Nikita. Fake News Detection Using Machine Learning approaches: A systematic Review. In Proceedings of the 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, 23–25 April 2019; pp. 230–234

[12] Segal, M.R. Machine Learning Benchmarks and Random Forest Regression; Kluwer Academic Publisher: Amsterdam, The Netherlands, 2004.

[13] Deokate, Suchitra. (2019). Fake News Detection using Support Vector Machine learning Algorithm. International Journal for Research in Applied Science and Engineering Technology. 7. 438-444. 10.22214/ijraset.2019.7067.