HIGH SCHOOL EDITION
Journal of Student Research
Volume 11 Issue 4 (2022)

Using Machine Learning Algorithms to Detect Fake News

Qiheng Gao[1] and Nicole Lantz[#]

[1]Lawrenceville School, Lawrenceville, NJ, USA
[#]Advisor

ABSTRACT

Fake news has been a growing threat in the modern world. A major reason why fake news is so dangerous and effective is due to the difficulties of distinguishing it from correct news, if there was a way to detect fake news accurately, its negative impact could be significantly minimized. Previous studies have already found that fake news differentiated itself substantially from real news in terms of words used and the structure of the texts, implying the possibility of differentiation. One possible method of detecting fake news is Machine Learning. Utilizing artificial intelligence to detect patterns within the text of fake and real news articles. In this paper, we test the capability of the Machine Learning Algorithms in detecting fake news using four different types of models, SVM, Multinomial NB, Gradient Boosting, and Gradient Boosting with LDA. We find that all four models had a high success rate of over 90%, with the LDA+Gradient Boosting model performing the best, and Multinomial NB being the least successful. We also attempt to determine the topics that fake news tends to cover and found that fake news is often about politics. While the model has proven to be successful, we recommend that future testing be done on other datasets with greater variety in news sources.

Introduction

Nowadays, people's actions are heavily influenced by the media that they consume. Thus, fake news has become a rising threat in recent times. The spread of false information through media outlets can have a huge impact. An example of the damage that fake news can cause is the spread of anti-vaccination misinformation, which has now become one of the top threats to global health (World Health Organization, 2019). Fake news can breed conflict, which can have drastic consequences, including tragedies such as the Pizzagate shooting (Haag & Salam, 2017). One of the main reasons that fake news is such a major problem is that it is very difficult for people to detect. In a study conducted by Stanford, it was found that Middle school, High school, and college students were shockingly bad at evaluating the credibility of information(Wineburg, McGrew, Breakstone, & Ortega, 2016). Most students were unable to identify differences between authentic and fake sources and couldn't tell apart a real and fake news sources on Facebook(Wineburg, McGrew, Breakstone, & Ortega, 2016)..

One possible way that we can attempt to remedy the problem of fake news is by creating a detection system using NLP(Natural Language Processing) to interpret and sort words and Machine Learning techniques such as SVM and gradient boosting to differentiate fake news from real news. This would greatly help tackle this issue by making it easier for people to recognize when articles are untrustworthy, greatly minimizing the impact of the deception. In our study, we will attempt to use NLP techniques to create an AI based classifier that can distinguish between real and fake news.

Several prior studies have already attempted to utilize AI tools to detect fake news. A key issue is the large variety of types of fake and real news. There can be serious fabrications written in tabloids; large scale hoaxes spread across multiple news sources and platforms; as well as satirical pieces are written to be intentionally absurd(Rubin, Conroy, Chen, & Cornwell, 2016). The large variety of types of fake news and their

variance in intent and wording makes it difficult for a classification model to take all types of fake news into account. In a study conducted by Rubin et al. (2016), when an SVM classification model was trained to distinguish between real news and satirical news, it was able to achieve an accuracy of 91%. However, when the same model was trained on all sorts of different fake news, the accuracy rate dropped to just 71%.

Horne and Adali (2017) used ANOVA and Wilcoxon rank sum tests to find the features that differentiate different categories of news. And helped make clear some of the main differences within the text of fake and real news. According to their observations, Fake news tends to have fewer stop words while having more nouns and verbs. However, their model only focused on differentiating fake news in only one topic at a time (Horne & Adali, 2017). Since it's a recent phenomenon, research into fake news is still in its early stages and it is clear that there is a current lack of research into developing a model that is able to differentiate between real and fake news regardless of topic and type. Our models will attempt to tackle this challenge.

Methods

Four models were created and trained using a dataset. The accuracy of these models was then tested

Dataset

The Fake and Real news datasets were compiled during studies conducted by Ahmed, Traore, and Saad (2017, 2018). The Real news were articles taken from Reuters.com whilst the Fake news was collected from sources flagged by PolitiFact (Ahmed, Traore, & Saad, 2017, 2018). For each article the Title, Text, Date of Publication, and Type were available. There were 23503 fake news samples and 21418 real news samples.

Our data was partitioned randomly into training and test sets at a 3:7 ratio using the Sci-Kit Learn library (Pedregosa et al., 2011). There were 17 different news subjects.

Table 1. Example of input data.

Fake or True	Title	Text	Subject	Date
Fake	Pope Francis Just Called Out Donald Trump During…	Pope Francis used his annual Christmas Day message…	Politics_news	December 25, 2017
True	LexisNexis withdrew two products from Chinese…	LexisNexis, a provider of legal, regulatory, and business information…	World_news	August 22, 2017

Data Preprocessing

Since we will be working mainly with text data, a lot of preprocessing must be done to convert the data into a form that is ready for modeling.

Date and Time

Date and Time data can help give us an idea of the recency of our data. The time and date in our dataset is in text form. To make it readable, we need to convert the text data into DateTime data. To do this, we utilize NumPy and the Python DateTime library to convert all the dates in the dataset into Datetime data(Harris et al., 2020).

Tokenization and Stemming

Tokenization is the process of reducing sentences into their smallest units (e.g. words and punctuation marks). This process is combined with stemming. As demonstrated in Figure 1 below, stemming is a technique that is used to remove prefixes and suffixes from words. Utilizing stemming, we can reduce more complicated words down to a single base form. To complete the Stemming we used the NLTK library and the Snowball Stemmer algorithm (Loper & Bird, 2002).

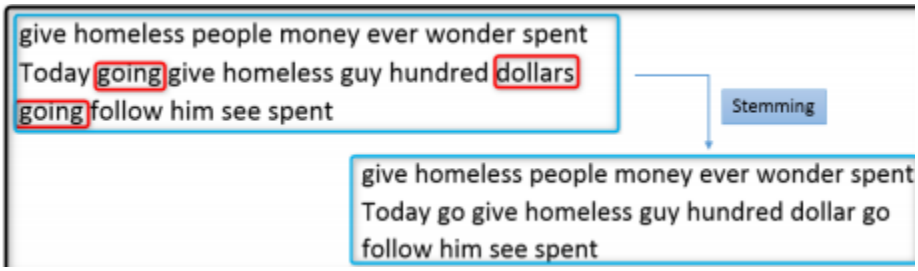

Figure 1. Example of stemming process. The figure above shows how prefixes and suffixes are removed to simplify and normalize words, making them more understandable. Reprinted from "Fake News Detection: A Deep Learning Approach," by A, Thota, P, Tilak, S, Ahluwalia, & N, Lohia, 2018, *SMU Data Science Review, Volume 1 (2018)*.. Copyright 2018 by Southern Methodist University. CC BY-NC 4.0

TF-IDF Vectorizor

Another technique we used was "Term Frequency Inverse Document Frequency" (TFIDF) for feature extraction. Term Frequency counts the number of times a word shows up in a document to determine its importance ranking. Inverse Document Frequency checks for words that do not appear often across all of the different documents. This distinguishes words that are important for the document but do not appear often across many texts. The importance ranking can then be used as an input for the classifier, which can lead to more accurate results. We utilized the Sci-kit Learn library to conduct the TF-IDF vectorization (Pedregosa et al., 2011).

Models

Our four models were SVM, Multinomial NB, Boosting Trees, and LDA+Gradient Boosting. These four models were selected due to their use in prior studies on text classification and how they fit our data (Horne & Adali, 2017; Ahmed, Traore, & Saad, 2017, 2018).

SVM

Among the 4 models, the first was a Support Vector Machine (SVM) for classification. SVMs take the data and fit a "best fit" hyperplane that divides the data and attempts to maximize the margin for categorization. This is shown in Figure 2 below. We pass our processed data through the SVM, and it classifies the data into fake and real news. Our SVM was created using the Sci-Kit Learn library (Pedregosa et al., 2011).

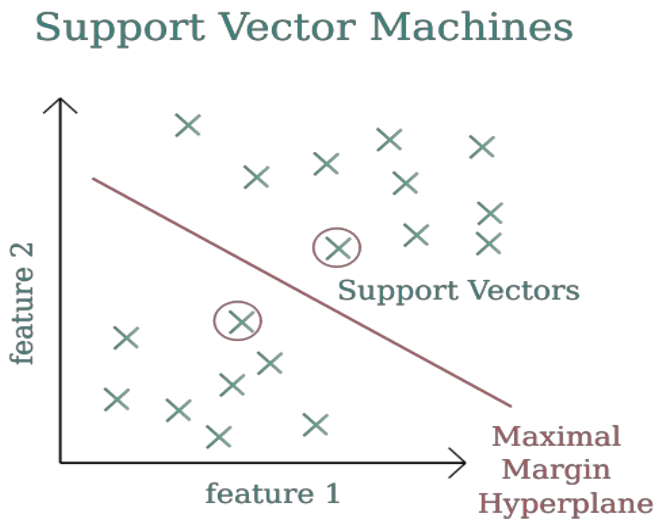

Figure 2. Visual representation of SVM. As seen in Figure 2, SVMs use a hyperplane to split the data into their respective classifications. From SVM (Support Vector Machines) diagram vector image, by OpenClipart, 2014, FreeSVG (https://freesvg.org/svm-support-vector-machines-diagram-vector-image). CC0 1.0.

MultinomialNB

Naive Bayes methods are a set of supervised learning algorithms based on applying Bayes' Theorem, shown in Figure 3 below, with the assumption that the features of measurement are conditionally independent of each other. Multinomial Naive Bayes is used when the features follow a multinomial distribution, which fits our data well. In the case of our model, it utilizes the frequency of words appearing in the text to calculate the probability that that piece of text belongs to an article of fake news. We used the Sci-Kit Learn machine learning library to create the model (Pedregosa et al., 2011).

Equation 1. Bayes' Theorem. Naïve Bayes classifiers are probabilistic models that apply Bayes' Theorem to classify data.

$$P(A|B) = \frac{P(A)P(B|A)}{P(B)}$$

Boosting Trees

Boosting is a type of supervised learning algorithm that attempts to predict a target value by combining the predictions of several weaker models to make a stronger ensemble prediction. When boosting, each subsequent decision tree is made to improve upon its predecessor and avoid making the same mistakes. As more and more effective decision trees are built, the classification gets more and more accurate. Gradient boosting is a specialized type of boosting in which a gradient descent algorithm is used to minimize loss as the number of models increases. Gradient boosting was done with XGBoost due to its optimization and algorithmic enhancements (Chen & Guestrin, 2016).

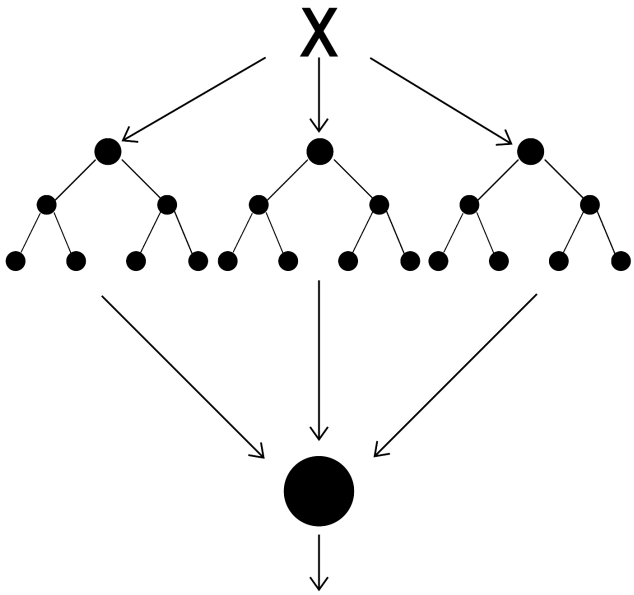

Figure 3. Visual representation of Boosting. Multiple weak decision trees are combined to create a single powerful prediction rule.

Latent Dirichlet Allocation + Gradient Boosting

Latent Dirichlet Allocation (LDA) is a topic model and is used to classify texts to a topic. LDA assumes that the distribution of topics in the document and the distribution of words in topics are Dirichlet distributions as there is a large amount of variability in the occurrence of words in the documents. As shown in Figure 4 below, LDA uses the frequency of these words to assign topics to texts. LDA gives each text a percentage of which it fits into each topic. After running the data through LDA using PyCaret, a new topic feature is generated (Ali, 2020). . We then run our data through another Gradient Booster, this time Catboost, as it works well with the categorical variables generated by LDA (Prokhorenkova, Gusev, Vorobev, Dorogush, & Gulin, 2019). In this test the data has an extra feature which is its topic.

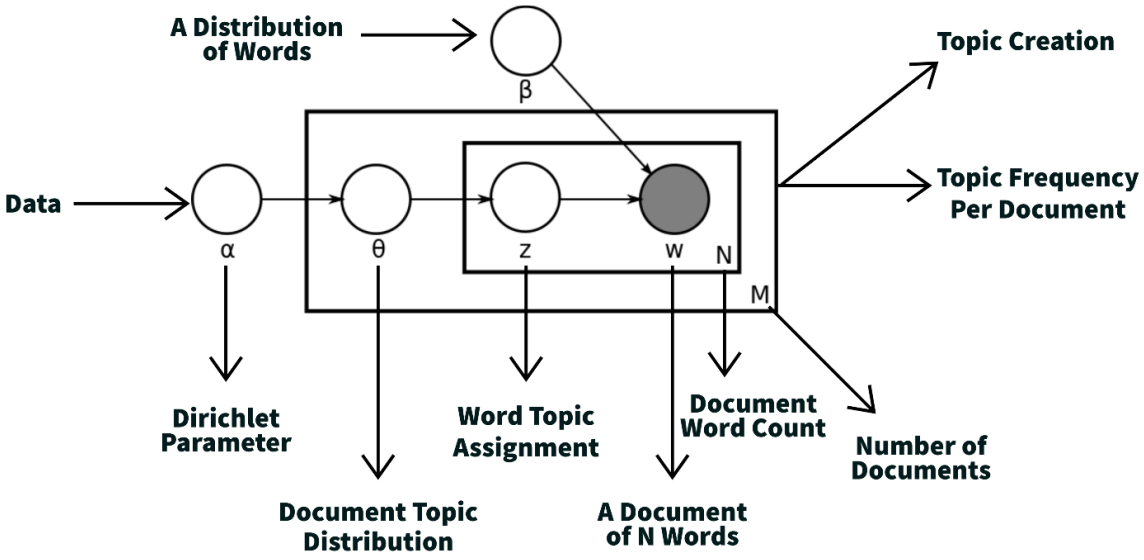

Figure 4. Visual representation of LDA. LDA uses the frequency of words to assign topics to a text. Adapted from "Latent Dirichlet allocation" by Bkkbrad, 2008, Wikimedia Commons (https://commons.wikimedia.org/wiki/File:Latent_Dirichlet_allocation.svg). CC BY-SA 4.0

Results and Discussion

After testing all four of our models, all four of them had outstanding results. LDA+Gradient Boosting performed the best followed by Boosting Trees and SVM. Multinomial NB was the worst performing model out of the 4. The models took around 2 minutes to train.

Table 2. Error metrics for our 4 different models.

Model	Precision	Recall	F1 Score	Accuracy
SVM	0.99	0.99	0.99	0.99
MultinomialNB	0.97	0.91	0.94	0.93
Boosting Trees	0.99	0.99	0.99	0.99
LDA+Gradient Boosting	1.00	1.00	1.00	1.00

Equation 2. Formula for Precision.

$$Precision = \frac{True\ Positive}{True\ Positive + Fale\ Positive}$$

Equation 3. Formula for Recall.

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

Precision is a measure of the accuracy of the positive predictions of a model. It's a good measure for when the costs of false positives are high. Recall calculates the true positive rate of the model and is a good measure to use when the costs of false negatives are high.

Equation 3. Formula for F1.

$$F1 = 2 \times \frac{Precision * Recall}{Precision + Recall}$$

Equation 3. Formula for Accuracy.

$$Accuracy = \frac{True\ Positive + True\ Negative}{True\ Positive + True\ Negative + False\ Positive + False\ Negative}$$

F1 is another error metric that seeks to balance precision and recall, and is a good model to follow when balance is needed between precision and recall and there is a high amount of true negatives. Accuracy measure both the true positive and true negative rate of the model, and is highly influenced by true negatives. For three out of the four models, all error metrics performed exceptionally well. Multinomial NB was the outlier. While its precision score was still high, the recall was quite low, suggesting that the false negative rate of the model was relatively high, meaning that it miscategorized a lot of fake news as real news.

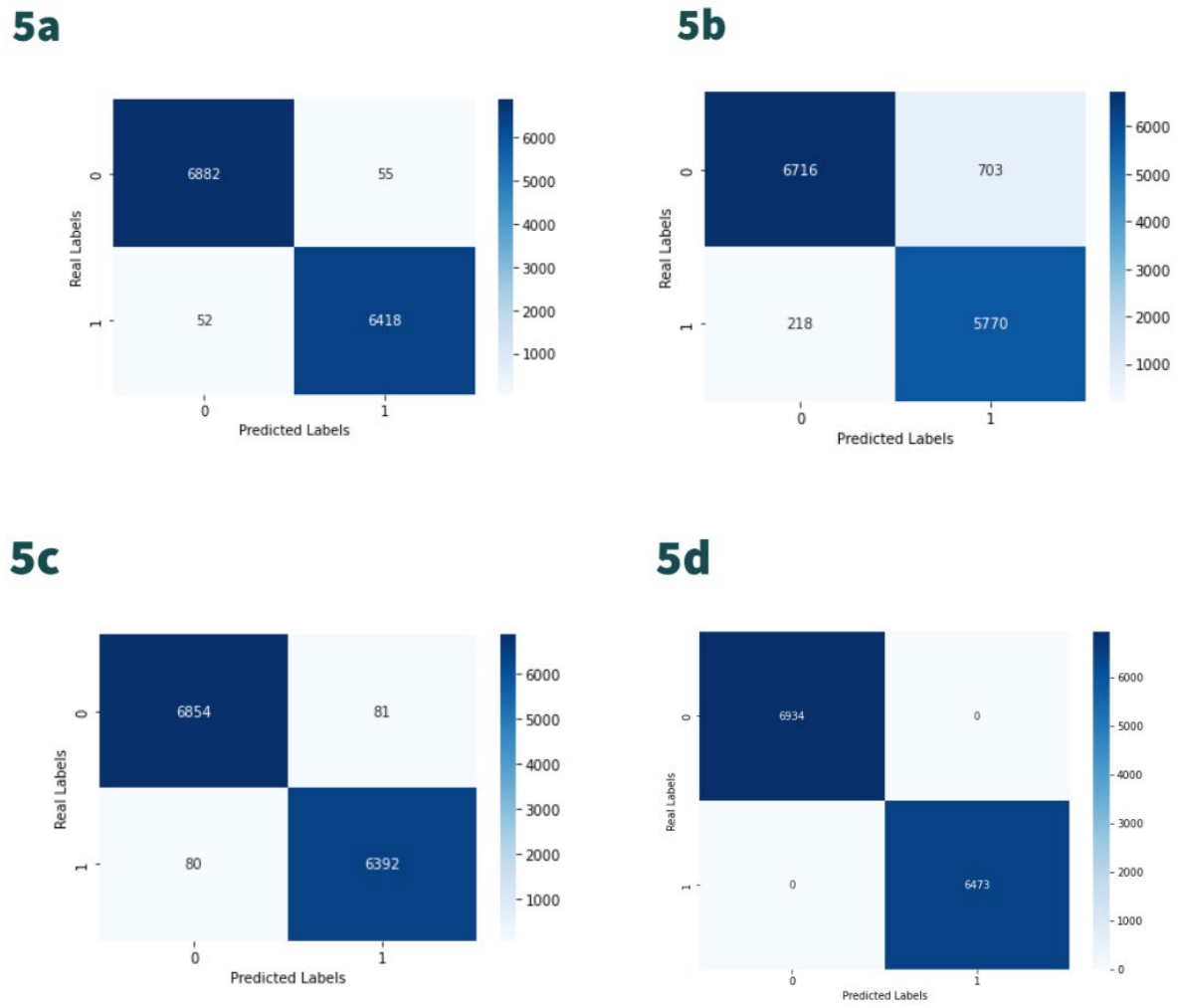

Figure 5. Confusion matrices to assess the performance of the four models. 5a is SVM, 5b is MultinomialNB, 5c is Boosting Trees, and 5d is LDA+Gradient Boosting. For the predicted and real labels, 0 represents fake news and 1 represents real news. The top left in each matrix represents the number of true positives, bottom right represents the number of true negatives, top right represents the number of false negatives, and bottom left represents the number of false positives.

LDA+Gradient Boosting was by far the best performing model out of the four with an abnormally high 100% accuracy. We believe that it functioned well since according to Horne and Adali (2017), fake news shared common traits in word use such as a lesser amount of stop words, that would allow the LDA to better group together the fake news into the same topic. Additionally, since all of the real news data was taken from a single source (Reuters), trends in the editing or writing style of that publication may have also allowed the model to perform extremely well.

The success of our models shows that there are commonalities among all fake news regardless of topic that can be recognized. It also suggests that NLP and machine learning based solutions can be effective in combating and detecting fake news.

Conclusion and Future Work

Our project attempted to find out if it was possible to use machine learning to distinguish between real and fake news. To do this, we used NLP to process the news data and created four classification models, SVM, Multinomial NB, Gradient Boosting, and LDA + Gradient Boosting. Out of the four models, the LDA and Gradient boosting performed the best out of the four while Multinomial NB was the worst performing model, but the results were still extremely good.

Although the model performed well, there may be issues with the dataset that caused the model to form exceptionally well. Since the real news was all taken from the same source, the writing may be similar. This may result in the writing of the articles being extremely similar leading to the model performing well. Further testing of the model must be done on more well compiled fake news datasets. In addition to that, the amount of people that still read traditional news through articles. Most people now are getting their daily news through social media. A further extension of this work would be to test to see if it can have similar results on datasets compiled from sources such as Twitter and Facebook.

Acknowledgments

I would like to thank my advisor for the valuable insight provided to me on this topic.

References

Ahmed, H., Traoré, I., & Saad, S. (2017). Detection of Online Fake News Using N-Gram Analysis and Machine Learning Techniques. *ISDDC*. https://doi.org/10.1007/978-3-319-69155-8_9

Ahmed, H., Traore, I., & Saad, S. (2018). Detecting opinion spams and fake news using text classification. *Security and Privacy, 1(1), e9.* https://doi.org/10.1002/spy2.9

Ali, M. (2020). *PyCaret: An open source, low-code machine learning library in Python.* https://www.pycaret.org

Bansal, H. (2020, November 25). *Latent Dirichlet allocation.* Medium. https://medium.com/analytics-vidhya/latent-dirichelt-allocation-1ec8729589d4

Bkkbrad. (2008, February 24). *Latent Dirichlet allocation* [Diagram]. Wikimedia Commons. https://commons.wikimedia.org/wiki/File:Latent_Dirichlet_allocation.svg

Harris, C. R., Millman, K. J., Van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., Van Kerkwijk, M. H., Brett, M., Haldane, A., Del Río, J. F., Wiebe, M., Peterson, P., … Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*(7825), 357-362. https://doi.org/10.1038/s41586-020-2649-2

Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785–794). ACM. https://doi.org/10.1145/2939672.2939785.

Haag, M., & Salam, M. (2017, June 22). *Gunman in 'Pizzagate' Shooting Is Sentenced to 4 Years in Prison.* The New York Times - Breaking News, US News, World News and Videos. https://www.nytimes.com/2017/06/22/us/pizzagate-attack-sentence.html

Horne, B., & Adali, S. (2017). This Just In: Fake News Packs a Lot in Title, Uses Simpler, Repetitive Content in Text Body, More Similar to Satire than Real News. https://doi.org/10.48550/arXiv.1703.09398

Loper, E., & Bird, S. (2002). NLTK: The Natural Language Toolkit. *CoRR,* https://doi.org/10.48550/arXiv.cs/0205028
.
Mishra, K. (2019, November 29). *Machine learning : Bayes theorem.* Medium. https://seeve.medium.com/machine-learning-bayes-theorem-2f48c33d51e5

OpenClipArt. (2014, September 4). *SVM (Support Vector Machines) diagram vector image.* FreeSVG. https://freesvg.org/svm-support-vector-machines-diagram-vector-image

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research, 12, 2825–2830.* https://doi.org/10.48550/arXiv.1201.0490

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A., & Gulin, A. (2019). CatBoost: unbiased boosting with categorical features. https://doi.org/10.48550/arXiv.1706.09516
Rubin, V., Conroy, N., Chen, Y., & Cornwell, S. (2016). Fake News or Truth? Using Satirical Cues to Detect Potentially Misleading News. In *Proceedings of the Second Workshop on Computational Approaches to Deception Detection* (pp. 7–17). Association for Computational Linguistics. https://doi.org/10.18653/v1/W16-0802

Thota, A., Tilak, P., Ahluwalia, S., & Lohia, N. (2018) Fake News Detection: A Deep Learning Approach. In *SMU Data Science Review*: Vol. 1: No. 3, Article 10. https://scholar.smu.edu/datasciencereview/vol1/iss3/10

Wineburg, S., McGrew, S., Breakstone, J., & Ortega, T. (2016, November 22). *Evaluating information: The cornerstone of civic online reasoning*. Stanford Digital Repository. https://purl.stanford.edu/fv751yt5934

World Health Organization. (2019). *Ten health issues WHO will tackle this year.* WHO | World Health Organization. https://www.who.int/news-room/spotlight/ten-threats-to-global-health-in-2019