ONE CLASS SUPPORT VECTOR MACHINE (OCSVM): NOVELTY DETECTION IN MISINFORMATION TWEETS

Rachel Rolle

American University

ABSTRACT

The objective of this study is to introduce a novel approach for accurately identifying misinformation disseminated through social media channels. Employing One-Class Support Vector Machine (OCSVM), the aim is to identify such misleading posts irrespective of the overall volume and the proportion of reliable versus unreliable content. While previous experiments have been conducted in this domain, this study endeavors to simulate real-world conditions by utilizing an imbalanced dataset. Specifically, the analysis will focus on tweets pertaining to the COVID-19 pandemic.

Index Terms**—** one class svm, social media, twitter, natural language processing, novelty detection, imbalance dataset, COVID-19

1. INTRODUCTION

In today's digital era, social media serves as a ubiquitous platform for millions worldwide. With information disseminating rapidly and minimal restrictions on user-generated content due to the principle of freedom of expression [1], the inadvertent or deliberate spread of false information can profoundly influence the behavior of readers [2]. Detecting misinformation swiftly on these platforms is paramount, yet it presents persistent challenges such as the overwhelming volume of data, the inherent ambiguity of content, the diverse structure of posts, and the imbalance between true and false classifications of tweets.

This challenge has been starkly evident during significant events like the COVID-19 pandemic. The proliferation of misinformation during this crisis has substantially shaped public perception and exacerbated social unrest. For instance, a single tweet alleging a scarcity of essential supplies led to widespread panic buying and heightened anxiety levels [3]. Such instances underscore the urgency of addressing malicious content dissemination, as it not only fuels negative emotions but also obstructs the acceptance of information.

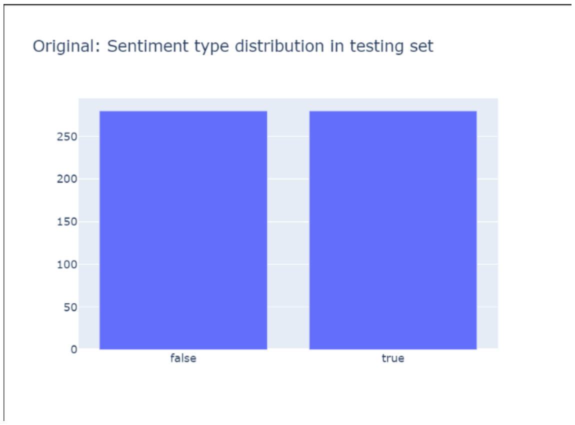


Figure 1: Distribution in Training Set

Hence, the primary aim of this project is to leverage an active learning approach known as One-Class Support Vector Machine (OCSVM) to identify tweets containing misinformation. If successful, this methodology promises rapid identification of false information within tweets.

In the realm of novelty detection, our model is tasked with discerning the outlier class [4], namely, false tweets, by training on a labeled dataset comprising such instances. Unlike conventional approaches that train on both true and false classes, our focus solely on false tweets aims to eliminate the risk of inadvertently reinforcing features of both reliable and unreliable content.

However, one inherent challenge of one-class novelty detection is the potential for overfitting, wherein the model becomes overly specialized on the training data, hindering its adaptability to new data introduced during testing. Additionally, given the multifaceted nature of Twitter posts, which may include videos, emojis, and images, the model's reliance solely on textual data may overlook crucial contextual factors influencing the categorization of posts as false.

In summary, while OCSVM presents a promising avenue for detecting misinformation, addressing the challenges posed by overfitting and the complexity of multimedia content remains essential to enhancing the robustness and effectiveness of such detection methods.

2. RELATED WORK

The 2018 research paper titled "Semi-supervised content-based detection of misinformation via tensor embeddings" delved into the realm of active learning and misinformation detection. The research team employed K-nearest neighbors (k-NN) and term frequency-inverse document frequency (tf-idf) techniques to identify misinformation within published news articles [2]. While considering a form of word-embedding method for my own research, I ultimately opted for Support Vector Machine (SVM) classification to discern unreliable tweets within the dataset. The decision was influenced by SVM's reputation for swift execution and its established efficacy in binary classification tasks [5]. Moreover, the study focused on news articles, which typically lack the dynamic and multifaceted nature of social media posts. I believe that the selection of these methodologies has contributed to an improved outcome. It will be intriguing to observe how my dataset behaves in the classification task compared to theirs.

3. METHODOLOGY

In addition to employing One-Class SVM for model training, this project utilized several additional methodologies, including handling imbalanced classes, text cleaning, Word2Vec, One-Class novelty detection, and Principal Component Analysis (PCA).

3.1. Data

The dataset was sourced from Dr. Zois Boukouvalas, who oversaw the Natural Language Processing course. These tweets were extracted from Twitter, specifically focusing on the COVID-19 event. Each tweet was categorized as either "unreliable" (1) or "reliable" (0). Additionally, another corpus exclusively comprised false posts, spanning across various social media platforms. This dataset specifically filtered false tweets from Twitter. Henceforth, the dataset containing only false tweets will be denoted as the training set, while the dataset comprising both false and true tweets will be referred to as the test set throughout this paper.

3.2. Imbalance Classes

The test set, initially obtained from my professor, was originally balanced, consisting of 280 false and 280 true tweets. However, for the purposes of this project, we sought to introduce an imbalance in the classes. To achieve this, a Python function was developed to randomly remove 250 tweets from the false class. To ensure consistency across multiple runs of the code, the "random state" parameter was set to forty-two.

As depicted in Figure 2, the number of false tweets has significantly decreased, comprising approximately 11% of the test set. With our imbalanced set now established, we can proceed to utilize One-Class SVM, renowned for its efficacy in addressing imbalanced datasets.

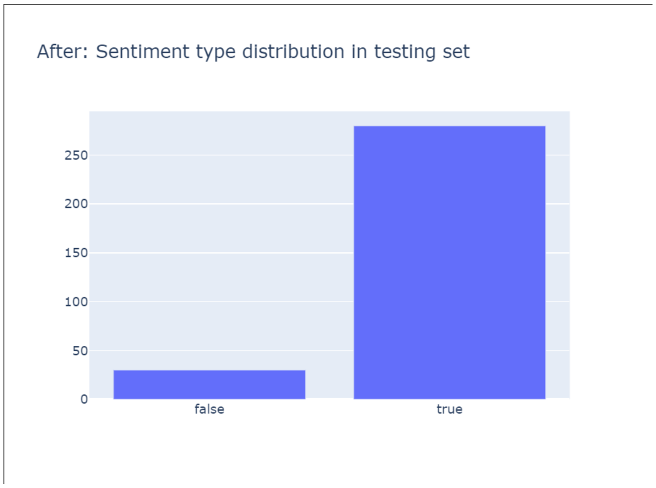


Figure 2: Create Imbalance in False Tweets

3.3. Text Cleaning

When dealing with text data, it's crucial to normalize each feature to enhance the effectiveness of our analysis. Normalization techniques such as lowercasing, stemming, and removing stop words can help minimize irrelevant aspects of a corpus and ultimately improve results. For instance, common stop words like "and," "or," and "so" are ubiquitous and unlikely to contribute to distinguishing between false and true tweets. Therefore, their inclusion in the evaluation may not be necessary.

In the clean text function that we've developed, stop word removal is integrated, along with the removal of hashtags, links, and emojis. While these elements may not be entirely insignificant, they tend to be too unique to establish meaningful relationships across posts. For example, the likelihood of multiple posts sharing the same link is minimal, making this feature less relevant. While the decision to retain emojis and hashtags could vary based on the project's requirements and time constraints, for this assignment, we opt not to analyze posts further using hashtags and emojis. The removal of these features is implemented using the "re" package.

Another text normalization technique involves employing the "WordNetLemmatizer" function from the "nltk.stem" package. This process involves reducing words to their root form by removing suffixes, thereby consolidating various forms of a word into a single entity. To facilitate each normalization operation, the text must first be tokenized so that individual elements can be processed effectively.

[“ I went to a birthday party.”]

would be first tokenized and would be represented in a list as:

[‘I’, ‘went’, ‘to’, ‘a’, ‘birthday’, ‘party’, ‘.’]

After being processed from the “clean_text” function, the individual text with be joined into a sentence again. We will get this.

[‘went birthday party’]

3.4. Word2Vec Feature Extraction

Once our data has been cleaned, the next step is to preprocess the text to prepare it for classification. Since machines only interpret numerical data, we must convert our text data into features. One effective method for this purpose is Word2Vec, a widely-used word embedding technique that captures semantic relationships between words, essential for our text classification task. In this approach, each post is represented as a vector, and the Word2Vec model captures these representations using cosine similarity. Mathematically, cosine similarity is expressed as follows [6]:

Equation 1: Cosine Similarity Equation [6]

$$similarity = \cos\theta = \frac{\bar{x} \bullet \bar{y}}{\| \bar{x} \| \| \bar{y} \|}$$

where :

$\bar{x} \bullet \bar{y}$: Vector dot product from x and y. $\sum_{k=1}^{n} x_k y_k$

$\|x\|$: Long vector x. $\sum_{k=1}^{n} x_k^2$

$\|y\|$: Long vector y, $\sum_{k=1}^{n} y_k^2$

I obtained a pretrained Word2Vec model provided by Google, which encompasses a vast vocabulary of 100 billion words sourced from newsletters and web text [7]. This resource is expected to significantly enhance the accuracy of our experiment, particularly beneficial given the relatively small corpus of data at our disposal.

3.5. One-Class Novelty Detection

One-class novelty detection shares similarities with One-Class classification [4]. The underlying concept involves focusing on a decision boundary for a single class, achieved through training the model exclusively on instances of that class. In our case, we're utilizing an imbalanced dataset and introducing only false tweets to the One-Class SVM.

Conversely, the positive class, comprising true tweets in the test set, has not yet been introduced. This constitutes the "novelty" data, representing new and unseen patterns within the dataset [8]. The objective is for the model to become adept at identifying false tweets to the extent that it can detect the introduction of new tweets. Consequently, this results in the formation of a decision boundary around the novel class.

This approach is advantageous because attempting to train a model on the rare false class, which constitutes only 10% of the population, while also conducting a random test-train-split, would likely lead to significant challenges in the model's ability to learn about the test class without overfitting. Indeed, such attempts were made in this experiment, yet the false tweets remained undetected. While there was an endeavor to increase the detection threshold, as will be elaborated on in the subsequent section, it proved ineffective in detecting false tweets.

A common solution to address this issue involves balancing the dataset of false and true tweets using techniques such as Under-Sampling or Over-Sampling. However, this approach may compromise the authenticity of the real-world scenario we aim to replicate.

3.6. One-Class Support Vector Machine (OCSVM)

The method selected for this experiment is Support Vector Machines (SVM), a component of the Scikit-learn library renowned for its effectiveness in solving classification problems and identifying patterns within robust datasets [4]. Given the inherent complexity of tweets, SVM proves to be an ideal choice for distinguishing false tweets from reliable ones.

Several parameters require adjustment, including "nu," "kernel," and "gamma." The relationship between the two classes is best described using the 'rbf' (radial basis function) option. Unlike other methods such as linear, polynomial, and sigmoid, the 'rbf' kernel method offers greater flexibility in capturing the nuances between false and reliable posts, making it the preferred choice for fitting the data.

The equation for the SVM kernel method with a Gaussian (RBF) kernel is as follows:

Equation 2: OCSVM Equation

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right)$$

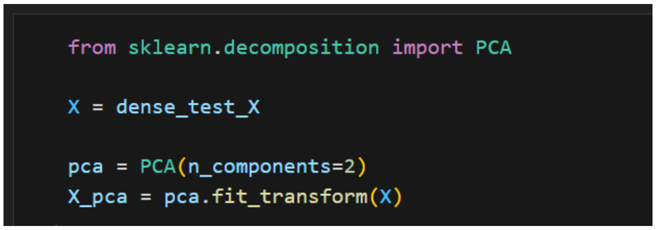
where x_i and x_j are input feature vectors, $\|x_i - x_j\|$ denotes the Euclidean distance between x_i and x_j , and σ is a hyper parameter controlling the kernel width [4].

Another set of parameters to consider are the gamma and nu parameters. The "nu" parameter determines the allowance of outliers in the dataset. Given that false tweets constitute approximately 10% of the data, setting this parameter to one-tenth is appropriate. Regarding the gamma parameter, it collaborates with the kernel method in shaping the decision boundary. Several decisions were made while testing the gamma parameter, including options such as "auto", .05, 1, 2, and 5. Selecting the correct value was crucial, as too low a value hindered the detection of outliers, while overshooting led to overfitting, making it challenging to interpret the decision boundary. The value of 1 emerged as the most suitable choice.

3.6. Principal Component Analysis (PCA)

To assess the parameters, one approach involved plotting their features to observe how the model differentiated between true and false tweets. Since the data existed in multiple dimensions, applying Principal Component Analysis (PCA) facilitated this process by simplifying the data into its lower-dimensional form. PCA extracts the most relevant information from the data [9].

In this experiment, the Word2Vec feature extraction yielded multiple features representing individual words. While this is advantageous for the model to capture the semantic relationships between words, it complicates the interpretation of the model's decision-making process. Figure 3 illustrates the application of PCA from the "sklearn.decomposition" package to the text data using the ".fit_transform" method.



```
from sklearn.decomposition import PCA

X = dense_test_X

pca = PCA(n_components=2)
X_pca = pca.fit_transform(X)
```

Figure 3: Applying PCA to Text Features

4. RESULTS

After applying the methods to the tweets, the following is the classification report output of the 1% outlier detection with a default score threshold, as shown in Table 1.

Table 1: Score Threshold At Default				
	Precision	Recall	F1-score	Support
0 (True)	0.88	0.35	0.50	280
1 (False)	0.08	0.53	0.14	30
accuracy			0.37	310
Macro avg.	0.48	0.44	0.32	310
Weighted avg	0.80	0.37	0.47	310

In Table 2, these are the results of the customized score for 1% of outliers.

Table 2: Customized Score Threshold Classification Report				
	Precision	Recall	F1-score	Support
0 (True)	0.91	1.00	0.95	280
1 (False)	0.75	0.10	0.18	30
accuracy			0.91	310
Macro avg.	0.83	0.55	0.56	310
Weighted avg	0.90	0.91	0.88	310

Comparing the decision boundaries based on the parameters, with original data, Figure 4.

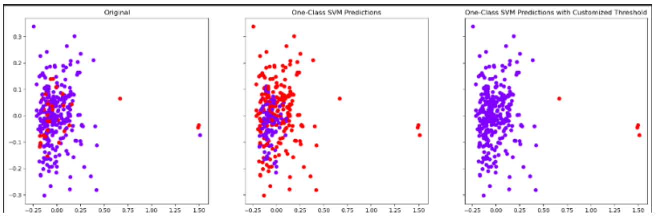


Figure 4: Comparing Thresholds with Original Data

The classification scores exhibit fluctuations when the researcher opts to detect more outliers. For instance, at a "nu" value of 1%, the accuracy is 0.37, whereas with a customized score threshold also at 1% outlier detection, the accuracy jumps to 0.91. With the customized score threshold, 91% of all tweets classified as reliable were indeed true. The recall score reflects that 100% of true tweets were correctly identified by the model, indicating its effectiveness in capturing most true tweets. The F1-score, providing a balance between precision and recall, stands at

95%, signifying good overall performance in identifying true tweets. Conversely, when the outlier rate was set to 1% with the default score threshold, the precision, recall, and F1-scores experienced a significant drop, ranging from 35% to 88%.

While the model exhibited high scores for all three metrics for the true class, it struggled with classifying false tweets accurately. With the customized threshold, only 75% of tweets classified as unreliable were actually false, indicating a high number of false positives. This suggests that many unreliable tweets were incorrectly classified as true. Furthermore, the recall for false tweets was a mere 10%, indicating that only a small portion of actual false tweets were correctly identified by the model. The low F1-score for false tweets reflects the model's poor performance in accurately identifying them.

5. DISCUSSION

Upon reviewing the findings, it becomes clear that the choice of outlier detection method is paramount. Notably, there was a substantial enhancement in the precision score, increasing from 0.08 to 0.75, upon lowering the score threshold. This shift signifies a reduction in false positives, highlighting the critical role of fine-tuning the detection threshold for optimal performance.

Figure 4 illustrates that the most effective model validation approach does not involve a deep dive into individual tweets. The dataset lacks clear boundaries between unreliable and reliable tweets, with posts from both categories intermingled. This inherent complexity poses a challenge for simplistic models, as depicted in Figure 4, where subtle differences between classes may not be easily discerned.

Nevertheless, despite these hurdles, it's important to acknowledge that this model can still be valuable in specific contexts. Considering our primary aim of identifying false tweets, there lies an opportunity to identify reliable ones as well. By meticulously examining tweets and retaining those that significantly deviate from the dataset's norm, we can effectively isolate and eliminate false tweets that starkly contrast with authentic ones.

6. CONCLUSION

In essence, this study aims to confront the pervasive issue of fake news on social media by advancing and expanding upon existing Natural Language Processing (NLP) techniques. Recognizing the nuanced complexities of social media posts compared to traditional news articles, our research endeavors to bridge this gap through the adaptation and refinement of methodologies. Through meticulous empirical investigation and analysis, our goal is to deepen the understanding of misinformation detection techniques and contribute to the development of more potent strategies for combating misinformation across online platforms.

Our findings indicate that the OCSVM classifier encounters challenges in discerning the intricate relationships between the two classes. As depicted in Figure 4, there is a notable difficulty in identifying false tweets positioned at the center of the plotted points. Moreover, efforts to bolster detection by raising the score threshold often lead to diminished classification scores. Hence, it is imperative to explore and deploy more sophisticated models capable of adeptly managing the inherent complexities of this task.

11. REFERENCES

[1] Boukouvalas, Z., Shafer, A.: Role of statistics in detecting misinformation: A review of the state of the art, open issues, and future research directions. Annual Review of Statistics and Its Application 11 (2024)

[2] Guacho, G.B., Abdali, S., Shah, N., Papalexakis, E.E.: Semi-supervised contentbased detection of misinformation via tensor embeddings. In: 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), pp. 322–325 (2018). IEEE

[3] Leung, J., Chung, J.Y.C., Tisdale, C., Chiu, V., Lim, C.C., Chan, G.: Anxiety and panic buying behaviour during covid-19 pandemic—a qualitative analysis of toilet paper hoarding contents on twitter. International journal of environmental research and public health 18(3), 1127 (2021)

[4] Perera, Pramuditha, Poojan Oza, and Vishal M. Patel. "One-class classification: A survey." *arXiv preprint arXiv:2101.03064* (2021).

[5] Huang, S., Cai, N., Pacheco, P.P., Narrandes, S., Wang, Y., Xu, W.: Applications of support vector machine (svm) learning in cancer genomics. Cancer genomics & proteomics 15(1), 41–51 (2018) [5] Hamilton, D., Pacheco, R., Myers, B., Peltzer, B.: knn vs. svm: A comparison of algorithms. Fire Continuum-Preparing for the future of wildland fire, Missoula, USA 78, 95–109 (2018)

[6] Jatnika, Derry, Moch Arif Bijaksana, and Arie Ardiyanti Suryani. "Word2vec model analysis for semantic similarities in english words." *Procedia Computer Science* 157 (2019): 160-167.

[7] Rezaeinia, Seyed Mahdi, et al. "Sentiment analysis based on improved pre-trained word embeddings." *Expert Systems with Applications* 117 (2019): 139-147.

[8] Pimentel, Marco AF, et al. "A review of novelty detection." *Signal processing* 99 (2014): 215-249.

[9] Shlens, Jonathon. "A tutorial on principal component analysis." *arXiv preprint arXiv:1404.1100* (2014).