Applied Machine Learning - Assignment 3

Ananthan Srinath Adhvait
Chalmers University of Technology
adhvait@chalmers.se

Lee Yu Xuan
Chalmers University of Technology
yuxua@chalmers.se

Teo Xuan Ming
Chalmers University of Technology
xuanm@chalmers.se

Abstract

In this assignment, we present an approach for classifying comments as pro-vaccination or anti-vaccination sentiments. Leveraging machine learning techniques, we developed a classifier using a small initial dataset, with plans to expand to a larger annotated dataset. Our methodology involves preprocessing the data to address uneven annotations and representing the text features using various vectorization techniques such as CountVectorizer and TfidfVectorizer. We employ several classification algorithms including Logistic Regression, Multinomial Naive Bayes, Perceptron, and a Multi-Layer Perceptron (MLP) Neural Network to determine which model performs best. Throughout our experiments, we evaluate the performance of each model using metrics such as accuracy, cross-validation scores, and classification reports. Additionally, we discuss the consensus among annotators in the dataset, the reliability of the data, and the impact of feature representation and algorithm selection on model performance. Our results indicate promising performance with the potential for further enhancement. Finally, we analyze errors made by the system, discuss the interpretability of the models, and propose avenues for future research to improve classification accuracy and model understanding.

1 Introduction

Vaccination remains one of the most effective public health measures, yet public discourse surrounding its efficacy and safety continues to be a topic of heated debate. It is viewed as unsafe and unnecessary by an increasing number of individuals (Nuwarda et al. 2022). In recent years, the proliferation of misinformation and skepticism regarding vaccination has led to an increase in anxiety about vaccines and vaccination programs leading to vaccine hesitancy. This has led to a resurgence of preventable diseases. In this paper, we tackle the challenge of classifying comments as pro-vaccination or anti-vaccination sentiments using machine learning techniques.

By analyzing sentiments expressed in online forums and social media platforms, we aim to provide valuable insights into the underlying factors driving vaccine-related attitudes and behaviors. Additionally, our study contributes to the broader field of natural language processing and sentiment analysis by exploring the applicability of various machine learning algorithms to a real-world classification task.

Our objectives include investigating the consensus and reliability of annotated data and developing robust classifiers capable of accurately distinguishing between pro-vaccination and anti-vaccination sentiments.

We begin by working with a small sample dataset, with plans to scale up to a larger annotated corpus to further refine and evaluate our models. The remainder of this paper is organized as follows: in Section 2, we describe the methodology employed in our study, including data preprocessing, feature representation, and model selection. Results and discussions are presented in Section 3, followed by conclusions and future directions in Section 4.

2 Methodology

In this section, we detail the methodology employed in our study for classifying comments as either pro-vaccination or anti-vaccination sentiments. Our approach consists of several key steps: data preprocessing, feature representation, and model selection.

2.1 Data Preprocessing

1-Jan	I'll only consume if I know what's inside it.
0/-1	It is easier to fool a million people than it is
0/0	NATURAL IMMUNITY protected us since e
0/-1	NATURAL IMMUNITY protected us since e
0/0	Proud to have resisted. Proud of my husbar
1/1/1/-1	The bigest sideffect of vaccines is fewer de
1/-1	Unvaccinated people are more likely to bec

Before training the classifiers, we preprocess the dataset to ensure consistency and eliminate potential sources of noise. This includes handling annotations that do not align and labeling them as "-1". An annotation is considered proper only when it is formatted as "sentiment_1/sentiment_2", where sentiment_1 and sentiment_2 are numerical values of either "1", "0" or "-1". Data with more than 2 sentiment annotations were dropped. Finally, we converted the numerical sentiment labels into string representations where 'Positive' denotes pro-vaccination sentiments, and 'Negative' represents anti-vaccination sentiments for Sklearn's classifiers. -1 were represented as 'Unclear' labels and were not used in training our models. After preprocessing, our usable data had decreased from 50k to 36k rows. Finally, we did an 80/20 split on the data to train and validation set respectively to ensure we are not overfitting during the training phase.

2.1 Feature Representation

The next step involves representing the textual comments as numerical features that can be used by machine learning algorithms. This is because the majority of the machine learning algorithms do not handle string data, instead they only handle numerical data. As such it is imperative that we convert our text data into numerical data by vectorizing them. We experiment with 2 main techniques for feature representation.

The CountVectorizer converts the text into a matrix of token counts, where each row corresponds to a document (comment) and each column represents a unique word (GeeksforGeeks, 2022). The value in each cell indicates the frequency of occurrence of a word in the corresponding document.

The Term Frequency-Inverse Document Frequency Vectorizer (TFIDF) is a statistical measure used to evaluate the importance of a word in a document relative to a collection of documents

(GeeksforGeeks, 2023). Similar to CountVectorizer, TfidfVectorizer converts the text into a matrix, however, commonly occurring words like articles are given a lower weightage.

We experimented with different parameters such as maximum features, n-gram range, and minimum document frequency to optimize the feature representation process.

2.3 Model Selection

For classification, we explore several machine learning algorithms to identify the most effective model for classifying the text data into 2 categories: pro-vaccination and anti-vaccination sentiments. The algorithms considered in our assignment include logistic regression, a linear classification model that estimates probabilities using a logistic function. Multinomial Naive Bayes, a probabilistic classifier based on Bayes' Theorem with strong independence assumptions between features. A perceptron and also a Multi-Layer Perceptron (MLP) Neural Network, composed of multiple layers of nodes, capable of learning complex patterns in the data.

We evaluate the performance of each model using metrics such as accuracy, classification reports, cross-validation scores, and fine-tuned hyperparameters as necessary to optimize performance.

2.4 Model Hyperparameters

For the Logistic Regression model, we have decided to set the *max_iter* value from the default of 100 to 1000, allowing the model more time to converge its value. By allowing the regressor to iterate more, we were able to yield a higher accuracy as the regressor was able to converge on the value even more.

For the MLP neural network (NN) model, ReLU was chosen as the activation function due to its mathematical nature of no plateau given a large positive output. Adam was chosen as the optimizer as it allows for an adaptive learning rate to allow our model to converge quickly. Early stopping was adopted to prevent overfitting when the model's validation score has not improved in over 20 iterations. After much trial and error, we decided that a 5 by 30 architecture focusing on depth in the NN was the most optimal model. Finally, various random states as a different set of initialization yield different performances due to the abundance of local minima in the non-convex objective functions of NN.

3 Results and Discussion

In this section, we present the results of our findings with different classifiers and discuss their performance in classifying comments as pro-vaccination or anti-vaccination sentiments.

3.1 Classifier Performance

We evaluated the performance of five classifiers using two main feature representation techniques: CountVectorizer and TfidfVectorizer. The following are the validation accuracies obtained for each classifier:

Vectorizer	Model	Validation	Test	F1 Score Positive	F1 Score Negative

TFIDIF	Dummy Classifier	51.36%	49.97%	0%	67%
Count	Logistic Regression	82.78%	84.26%	84%	84%
TFIDIF	Logistic Regression	84.45%	85.23%	86%	85%
TFIDIF	Multinomial Naive Bayes	82.68%	83.96%	84%	84%
TFIDIF	Perceptron	79.90%	80.53%	81%	80%
TFIDIF	MLP NN	84.54%	85.68%	86%	86%

3.2 Discussion

The dummy classifier serves as a trivial baseline. All the models performed much better than the dummy in terms of accuracy and F1 scores. Before settling on TFIDF as the vectorizer, it was tried alongside CountVectorizer across all the models and performed worse in every instance. This suggests that the TfidfVectorizer feature representation technique is more effective for capturing the distinguishing characteristics of pro-vaccination and anti-vaccination sentiments due to the weightage and penalization properties.

We then focused our efforts on model hyperparameter tuning with TFIDF vectorized data. Logistic regression was a strong start as it was suited for binary classification tasks. The performance of the Multinomial Naive Bayes classifier was slightly lower compared to logistic regression, indicating that the assumption of independence between features may not hold as strongly in this context. However, the relatively high accuracy of 82.6% suggests that it remains a competitive option for text classification tasks. We then explored perceptron and it was evident that a single layer was insufficient to truly capture the relationship of the dataset. The idea was then extended to MLP NN with the focus on crafting a deep NN to capture the complex relationship between the corpus and sentiment. Our hypothesis worked as MLP NN was the best-performing model in both the test accuracy and F1 scores. Further experimentation with more sophisticated neural network architectures and hyperparameter tuning may lead to improved performance.

However, there is still room for improvement, particularly in addressing the nuances and complexities inherent in natural language processing tasks.

3.3 Limitations and Future Directions

Currently, the utilization of the CountVectorizer and TfidfVectorizer only takes into account the number of occurrences that each feature has shown in the input data. These two vectorizers do not

account for the fact that certain features have greater importance over another feature in helping to determine the classification of the data.

Another limitation of using these two vectorizers is the fact that they also do not factor in the context of the data and order of the words as it considers each feature independently. Given the data “The vaccine isn’t all that bad”, the vectorizers would only look at each word independently and the classifier then decides whether this statement is pro-vaccine or against-vaccine. The word “bad” might trigger the classifier to classify this statement as against vaccines even though we know that this is not the sentiment of the statement.

Given more time, our hyperparameter tuning for the MLP NN could be much more systematic and utilize grid search. This would alleviate the time-consuming process of fine-tuning each hyperparameter manually and eliminate human errors. Moreover, it would increase the scalability and sustainability of future NN projects.

As such, to improve the performance of the classifier we can look at Large Language Models (LLM) such as BERT which comes with its vectorizer/encoders. As such LLMs have been fine-tuned on a large corpus of text data, the way that these models interpret a statement will vary greatly from how either the CountVectorizer or TfidfVectorizer does as the majority of these LLMs provide a weight for each of the words that had been used in the input. Some of these LLMs have functions that allow the user to perform sentiment analysis. But fine-tuning these LLMs requires a lot of time and energy.

Another method to help classify these data could be through Semantic Search. Semantic Search is often used by search engines (i.e. Google) to provide users with results that are similar to the user’s query.

An alternative would be to consider techniques such as Long-Short Term Memory (LSTM) to provide more context for the model to use or provide another method that calculates the importance of each word that helps it to classify the input data. This would require less resources compared to the other alternatives.

4 Conclusion

In conclusion, language processing tasks are complex and uprising given the popularity of ChatGPT in the past year. This assignment has allowed us to explore how different algorithms could be used in NLP tasks beyond just NNs. Certain tasks can be completed using simpler methods and one does not always need to rely on resource-intensive methods such as NNs or LLMs to carry out NLP tasks.

Citations:

1. Nuwarda, R. F., Ramzan, I., Weekes, L., & Kayser, V. (2022). Vaccine hesitancy: Contemporary issues and historical background. Vaccines, 10(10), 1595.
2. GeeksforGeeks. (2022, July 7). *Using CountVectorizer to Extracting Features from Text*. https://www.geeksforgeeks.org/using-countvectorizer-to-extracting-features-from-text/
3. GeeksforGeeks. (2023, January 19). *Understanding TF IDF term frequency-inverse document frequency*. https://www.geeksforgeeks.org/understanding-tf-idf-term-frequency-inverse-document-frequency/