# An analysis of digital comments to detect the stance towards being pro or anti Covid-19-19-19 vaccines

**Anouka Ranby, Andreas Nilsson, Erik Rosvall**
Applied Machine Learning, University of Gothenburg
(gusranban@student.gu.se, gusnilanht@student.gu.se, gusrosver@student.gu.se)

**Abstract**

Presented here is a project designed to use comments from varied social media sources and online news articles to try to identify comments as either pro or anti Covid-19 vaccines. The text comments were crowdsourced and manually annotated and further processed and transformed with vectorizers. We explored five different classification models and were able to classify comments with up to 89% accuracy.

## 1. Introduction

As the global pandemic broke out early 2020 there has been a constant and ongoing conversation about how to best defeat Covid-19, if at all possible. The conversations on news sites and social media have overwhelmed us with information and comments about the diseases and its impact. When the conversations started about the possibilities to bring forward a vaccine, quicker than had ever been done before, the global population reacted even more. Since then it has been questioned whether the vaccine is trustworthy considering the fast developments of it and if there may be long-term impacts that are yet unknown. There have also been large positive reactions towards taking the vaccine and it being responsible for saving many lives from severe illness and death globally. The debate about Covid-19 vaccine has become somewhat polarized and people have taken a stance of being pro or anti vaccine, which the conversation on many digital platforms can witness about (Journals, 2021). In this study we investigate the possibility of detecting whether people are pro or anti Covid-19 vaccine by utilizing comments gathered from social media platforms and online news sites.

## 2. Methods

### 2.1 Crowdsourcing the data

The comments being used in this study were gathered in a collective exercise between the students taking the master course Applied Machine Learning at the University of Gothenburg. Every student had to select a minimum of 100 comments written in the English language and that were related to Covid-19 vaccinations. The comments had to be selected from any social media site or from online articles of the students choice and ideally from a variety of sources. Students were advised to collect a balanced data set of about 50% pro-vaccine and 50% anti-vaccine comments and avoid comments not expressing an opinion about the vaccine or that are taken out of its context as a response to other comments (e.g. “I don’t agree”). The source of the comments were not documented and is therefore unknown.

### 2.2 Annotations

The annotation of the crowdsourced data was carried out in two steps. Firstly each student was required to label each comment they had collected originally as pro-vaccine (1) or anti-vaccine (0) and hand it in. Secondly each student received a file with about 100 comments collected by other students and similarly marked them with 1 or 0. Any ambiguous comment had to be marked as -1. This resulted in files consisting of two columns: one for the labels (1,0,-1) and one column for the comments. The process of having multiple persons annotating the same data (comments) was a way to check the agreement of annotators and bring further clarity to whether the pro or anti stance in the comments were clear. This also resulted in a data set with multiple labels for each comment and had to be dealt with, which we'll cover below.

### 2.3 Data Labels

The data set that contained multiple labels for each comment was earmarked as the training data and was shared back with all students along with a separate test data set not having multiple labels. The training set consisted of a total of 26197 comments with a number of labels ranging between 1-23 annotations. The multiple labels for each comment had to be downsized to one label per comment. We took a strict approach and marked comments as follows:

- Assign final label as 1 where all annotations were 1 (minimum 2 annotations)
- Assign final label as 0 where all annotations were 0 (minimum 2 annotations)
- Assign the rest as -1

This resulted in 10791 comments labeled as 1, and 10491 comments labeled as 0, and 4915 comments labeled as -1 (totalling 26197), which means that 81% of the comments (10791+10491/26197) had consensus and were marked consistent between annotatores (minimum two) when computed simply. To compute a statistic to review the agreement among the annotaters we utilized Krippendorff's alpha. After computing Krippendorff's alpha we got approximately 68% which is within the reliability interval of acceptance 66,6-80% (Krippendorff, K. 2011) . All comments labeled as -1 were excluded from the analysis as annontator could not agree and were therefore considered as ambiguous in their stance towards Covid-19 vaccine, which could make the classification task more difficult. Any comments only having 1 annotation was also excluded since it had not been controlled by a second annotator resulting in a training data set of 21282 comments.

### 2.4 Preprocessing

We performed processing of the data in order to create additional columns that could be used in the classification task. The original column containing all the comments was named 'Comment'. We created a new column named 'Cleaned' where we removed emojis, accented characters, punctuations, potential hyperlink symbols, leading and trailing spaces, unwanted characters and made all text to lowercase. We also created separate columns based on the Cleaned column where we removed stop words (e.g. 'a','the','are' etc. ), short words (<=3) and a combination of the three. Columns were added for the stemmed and lemmatized (reducing words to its dictionary baseline) versions of the 'Cleaned' column. Lastly numerical columns were added for

string length (including all characters and spaces in comments), word count and share of capital letters used based on the original 'Comment' column. Column names can be viewed in Figure 1. Comments that only consisted of emojis got a NaN value when calculating the upperRate that was replaced by 0 since it did not contain any capital letters. Preprocessing was applied to both the training and test data sets.

### 2.5 Features

As outlined in the preprocessing part we created additional feature columns that resulted in 10 columns of data including our original baseline column containing the comments without any processing. Seven of the columns were text based and somewhat similar to the original column and three of the columns were numerical. All text columns were further processed by two different vectorizers from the scikit-learn library. Both work well for processing text to be digestible by machine learning models and classification tasks hence the reason that we explored these two, along with that they are related in how they are using term frequency but different in weighting words, which is outlined below:

1. *CountVectorizer* converts text to a matrix with the count of each unique word (Figure 1).
2. *Tf-IdfVectorizer* is built on the base of CountVectorizer but applies weighting to give less weight to words that appear often in many comments (e.g. 'I', 'you') on top of the term frequency (see visualized example in Figure 2).

Both vectorizers were combined with all the text based columns and a classification model while the numerical columns were evaluated with a classification model.

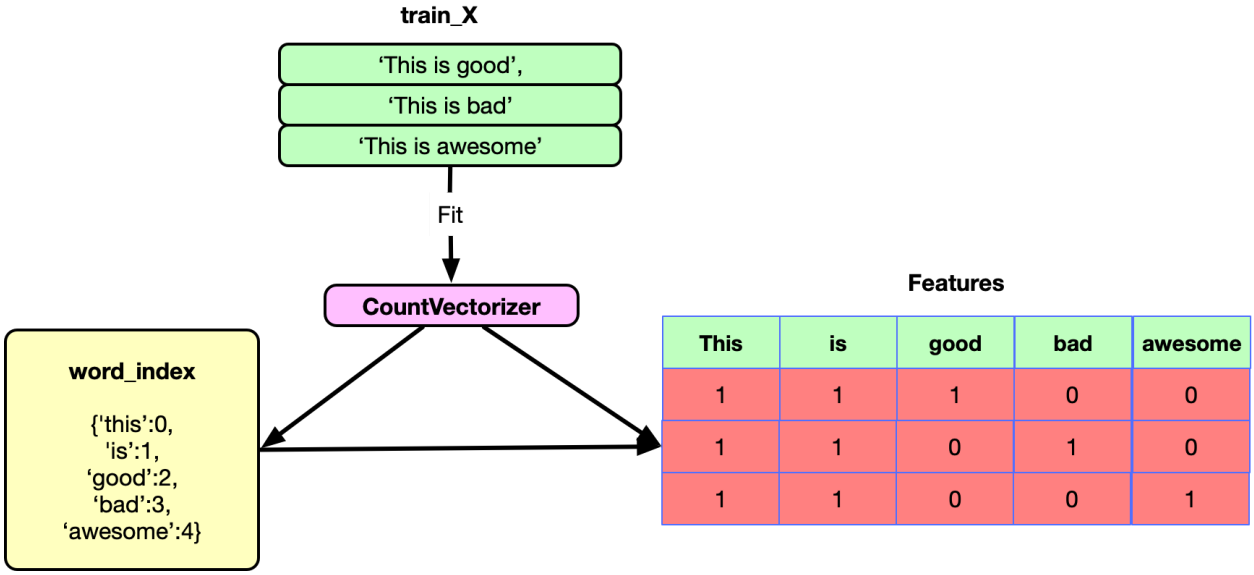


*Figure 1. Visualization of how the CountVectorizer works (ML Whiz, 2019)*

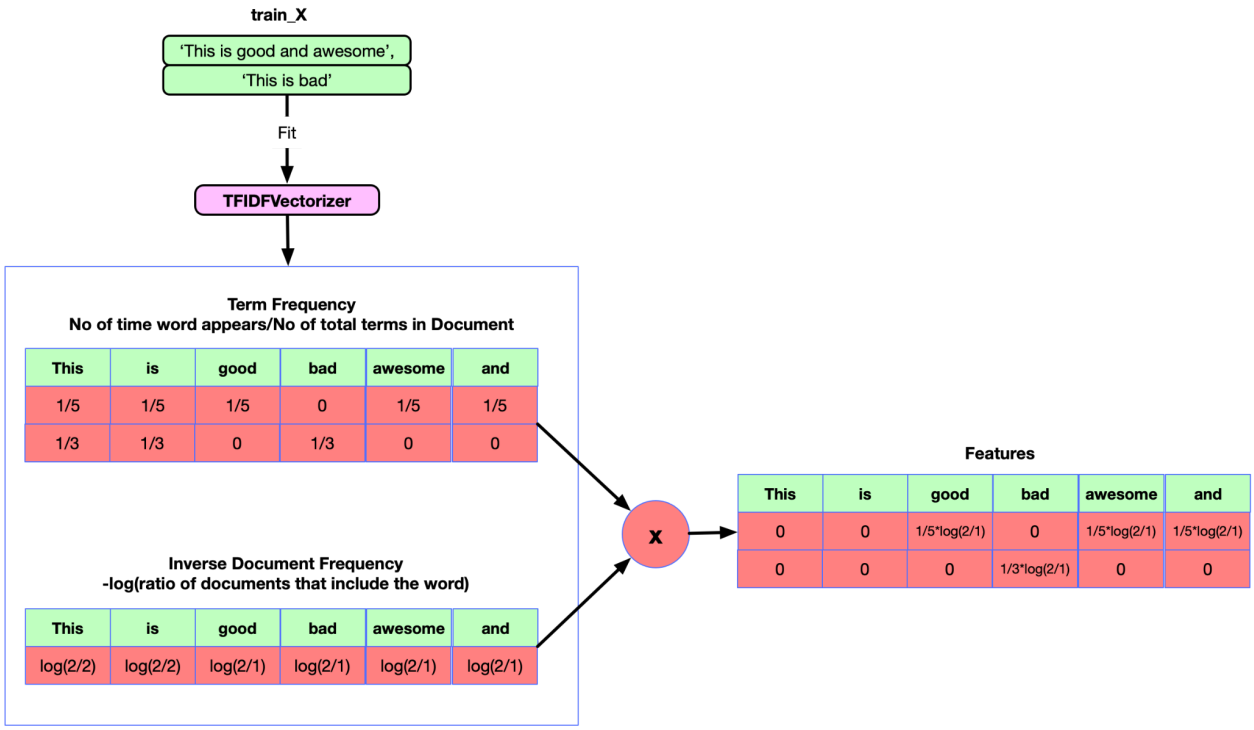


*Figure 2. Visualization of how the Tf-idfVectorizer works. (ML Whiz, 2019)*

### 2.6 Classification Framework

We initially considered five different classification algorithms from the scikit-learn library in the study: DummyClassification (only as trivial baseline), Logistic Regression (LR), Support Vector Classifier (SVC), Random Forest Classifier (RFC), Perceptron Classifiers, and Multinomial Naive Bayes Classifier (MNB) implemented in a pipeline where they were combined with one feature column and vectorizer (only for text columns). The three algorithms that then performed best were brought forward for additional hyperparameter tuning with scikit-learn's GridSearchCV. The reason we started out with so many models was based on our curiosity of comparing models we had recently covered in the course and other courses. While the main focus of models selected was their relative simplicity (linear and statistical).

## 3. Result

### 3.1 Best vectorizer and feature column

We did some initial testing on a smaller training set (2500 comments to not be too cumbersome to process) to find the best feature column and vectorizer combination with a default LR classifier (no hyperparameter tuning). Based on the average accuracy score generated from five-fold cross-validation the winner was the 'Lemmatized' column combined with the Tf-IdfVectorizer. Even though the performance for the 'Stemmed' column was more or less the same we decided to continue with the 'Lemmatized' columns as it does not have the same tendency to create an in-correct base of words. Figure 3 shows the result from the Tf-Idf Vectorizer performance.

| Column processed by Tf-IdfVectorizer & LG | Mean Accuracy Score |
|---|---|
| 'Comment' | 76,3% |
| 'Cleaned' | 76,1% |
| 'Stemmed' | 77,6% |
| 'Lemmatized' | 77,7% |
| 'Clean-NoStopWords' | 75,0% |
| 'Clean-NoShortWords' | 74,2% |
| 'Clean-NoStop-NoShort' | 74,2% |
| Numerical columns (StrLen, WordCount, upperRate) | 53,6% |

*Figure 3. Column and the mean cross-validation accuracy score from being processed by Tf-IdfVectorizer (except for numerical columns) and classified with Logistic Regression.*

**3.2 Top algorithms**

Once we had the best column feature ('Lemmatized') and vectorizer ('Tf-idf') defined we continued to test the five different classification algorithms on the smaller training set used in initial testing. These models were tested with their default hyperparameters and based on the average accuracy score generated from five-fold cross validation the best models were MNB (78,6%), LR (77,7%) and SVC (78,0%). Figure 4 shows all the five models' average accuracy performance.

| Classification Algorithm & Tf-IdfVectorizer | Mean Accuracy Score |
|---|---|
| Multinomial Naive Bayes | 78,6% |
| Logistic Regression | 77,7% |
| Support Vector Classifier | 78,0% |
| Random Forest Classifier | 74,5% |
| Perceptron Classifier | 73,7% |
| Dummy Classifier | 50,2% |

*Figure 4. Classification algorithm with Tf-IdfVectorizer and best column 'Lemmatized' along with their corresponding mean accuracy score from five-fold cross-validation.*

The three winning models we brought forward to perform additional hyperparameter tuning with GridSearchCV. We still utilized the smaller training set and best feature column and vectorizer identified earlier. The hyperparameters for the vectorizers ngram range were set similarly for all

three algorithms while each model had different hyperparameters available to be tuned. The MNB was limited to try out different values for its smoothing parameter alpha. For SVC we tested different values for C (how much to avoid misclassification), tolerance for stopping criterion, and different kernels. LR also had the same values tested for C and tolerance as the SVC and also tested different penalties (l1, l2) for the liblinear solver. The models were then trained again and evaluated on a separate validation set where MNB had the highest validation accuracy score of 83,4%, LG 80,8% and SVC 79,2%. The MNB was significantly faster to train and evaluate on the validation data compared to the LR and SVC.

**3.3 Final Model and evaluation**

The MNB algorithm had the best alpha hyperparameter value of 0.1 and the vectorizer had the best ngrams 1,3 and was finally trained on the full training data set (21282 comments) and evaluated along with best hyperparameters we identified on the unseen test data (1136 comments). The final accuracy score on the test data was 88,6% along with a F1 score (harmonic mean of precision and recall) of 89% for pro vaccine comments (1) and 88% for anti vaccine comments (0). Both precision and recall were high, around 90% respectively. The results generated from this final evaluation can be seen in the classification report in figure 5.

```
### Final Model: Multinomial Naive Bayes ###
Final Model Test Accuracy: 0.886443661971831

Classification report:
              precision    recall  f1-score   support

           0       0.90      0.87      0.88       568
           1       0.88      0.90      0.89       568

    accuracy                           0.89      1136
   macro avg       0.89      0.89      0.89      1136
weighted avg       0.89      0.89      0.89      1136
```

*Figure 5. Final model performance of the Multinomial Naive Bayes (alpha=0.1) classifier with 'Lemmatized' column processed by the Tf-IdfVectorizer using 1,3 ngram_range trained on the full training set and tested on the full test set.*

We also computed a confusion matrix to be able to evaluate True negative, False negative, False positive and True positive with the corresponding volume and share of comments that were placed in each category. The darker blue areas in the confusion matrix in figure 6 represent the anti-vaccine comments correctly labeled and the bottom right box represents the pro-vaccine comments correctly labeled. The color indicates the volume of comments and the lighter blue boxes indicate that very few comments out of the total were labeled incorrectly by the model. Since this classification task aims to discover if comments are pro or anti Covid-19 vaccine we accept the accuracy score at nearly 90% as very good in doing so.

The combination of words that helped the model to classify comments were largely combinations of 2-3 words containing words like ‘no vaccine’, ‘die more’, ‘take vaccine’. While misclassification and errors produced by the algorithm were comments where the user was e.g. using irony or using metaphors for the vaccine like: “I don't know what's in it. As if they know what's in the hamburgers they eat, the milk they drink, the high fructose corn syrup they consume... what a tragedy of stupidity.” and “Vaccines are saving lives meanwhile romania being the least vaccinated and also least infected country in the whole europe”.

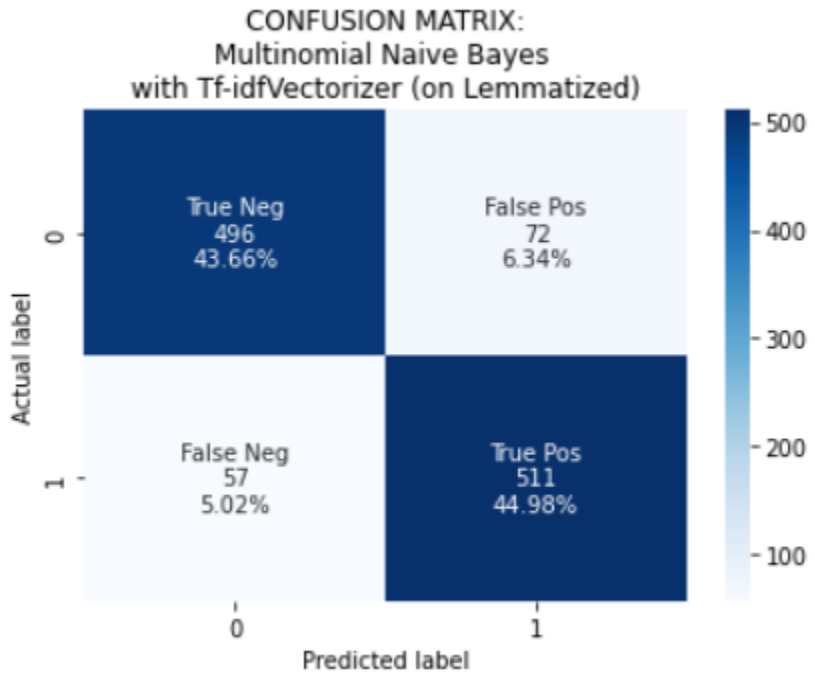


*Figure 6. Confusion Matrix showing the result from the final model configuration classifying comments from the test data.*

**4. Conclusions**

We can conclude that this classification model can help detect whether comments on social media or online news articles are pro or anti Covid-19 vaccines with 89% accuracy. This is considered good and far better than the result when compared to the dummy classifier and null accuracy score which were both around 50%. The most crucial part of this study and project was to perform processing of the text, which is the core of natural language processing tasks in order to generate strong features.

With the limited time we had we did not perform tests to prove that the performance of the initial testing for the best feature column and vectorizer were statistically significant. If this would have shown that the preprocessed columns did not have a statistically greater accuracy the effort to transform large volumes of text should probably be avoided as the original column could have been used for this without too much loss in accuracy.

The initial testing was limited to a sample of the training data as it was too cumbersome to run the full training data on the many columns, vectorizers and models tested. We could have tested fewer feature columns and models to afford to train on a larger data set, which could have resulted in other results and choices. We could also see that training on a larger data set generated better results as the corpus gets much larger, helping the algorithm to learn patterns and better label test comments.

Finaly future analysis could involve creating additional features related to sentiment to discover if positive or negative sentiment can help the algorithm detect whether the comments are anti or pro Covid-19 vaccines.

**References**

Journals (2021, 3 Mar). *The anti-vaccination infodemic on social media: A behavioral analysis.* *https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0247642*

Krippendorff, K. (2011). Computing Krippendorff's Alpha-Reliability. Retrieved from: *https://repository.upenn.edu/cgi/viewcontent.cgi?article=1043&context=asc_papers*

ML Whiz (2019, 8 Feb). *NLP Learning series: Part 2 - Conventional Methods for Text Classification.* *https://mlwhiz.com/blog/2019/02/08/deeplearning_nlp_conventional_methods/*