Stance Classifier
Analyzing Comments on COVID-19 Vaccination through Text Classification

Dylan Osolian
MPALG
Chalmers University of Technology
dylan.osolian@gmail.com

Lisa Samuelsson
MPDSC
Chalmers University of Technology
lisasamuelssons@gmail.com

1 Introduction

The COVID-19 pandemic had a profound impact on people's daily life and caused a lot of health issues worldwide. Vaccines against the disease were quickly developed and the spread slowed down, but the vaccines generated polarized opinions. While some people were enthusiastic about getting vaccinated, others took a strong stance against the vaccines.

This report covers the development of a text classifier that has the goal of determining whether a textual comment expresses a positive or a negative opinion towards COVID-19 vaccinations. Given data collected and annotated by course participants, different classification models were trained and evaluated.

2 Method

2.1 Data preprocessing

A large group of students collaborated on collecting the training data. Each student collected and annotated 100 comments each. After the first round of annotations, the data was then annotated by other students who had not seen the initial labels. If an annotator couldn't understand if the comment was for or against vaccines they left the value -1. This meant that the final training set had a number of annotations for each comment, and to handle these agreements four differents methods were decided on:

1. Selecting the majority label (that isn't -1)
2. Removing all rows with a label -1, select the majority label for the remaining rows
3. Removing all rows that contain -1 and those with equally many 0's and 1's, select the majority label for the remaining rows
4. Keeping only the rows where all annotators have agreed

All models were evaluated using all training sets, and the best results were obtained when keeping only the rows with agreeing labels.

2.2 Encoding

In order to use the collected data in the common machine learning models it has to be converted to a format which is supported, more precisely it has to be turned into numerical data. Vectorization is a feature extraction step commonly used to get numerical features out of text data. To do this a Term Frequency-Inverse Document Frequency (TFIDF) vectorizer was used. TF-IDF vectorizers are one of the most commonly used vectorizers to represent text data as features. The TF-IDF vectorizer considers the frequency of words and gives a numerical vector with weighted values representing the importance of the words. In order to get the most out of the data some data cleaning was done before vectorizing it, all of this was handled directly by the vectorizer. All the data was converted to lowercase by letting the *lowercase* parameter be *True* and the parameter *strip_accents* = *unicode* removes all the accents and normalizes some characters.

2.3 Models

Six different classifiers from scikit-learn were used:

- Random Forest Classifier
- Bernoulli NB
- Multinomial NB
- Linear SVC
- SGD Classifier
- Logistic Regression

These classifiers have been covered in the course material and are commonly used for text classification.

A dummy classifier from scikit-learn that always predicts the most frequent label was also used as a baseline model. This model gave accuracies of around 50 percent for all four datasets, which ensured us that they were still close to balanced.

2.4 Hyperparameter tuning

The hyperparameters of all selected models were tuned by doing a grid search. Grid search is a way to find the optimal hyperparameters for a model by trying all of the possible combinations of the different parameters. To do the search, GridSearchCV from scikit-learn's library was used. GridSearchCV runs a cross validation on the model for every combination of the specified parameter grid and then returns the parameters which maximizes the score.

2.5 Evaluation

To evaluate the models cross-validation with 10 folds was performed, and both the accuracy and the F1 score was measured. Confusion matrices were plotted to see if any class was more prone to misclassification, but misclassification generally happened equally often to both classes.

As a further analysis of the chosen model a plot of the feature importances were created, which allows us to look at what words the model finds important in order to classify a social media comment as pro or anti vaccination. Some examples of misclassified comments are also looked at and discussed further to try and reason about out why the model behaves as it does.

3 Result

After tuning and finding the optimal hyperparameters of all selected models the results shown in Table 1 were obtained. The table shows the accuracy and F1 score for the classifiers on all of the 4 datasets. Random Forest was the worst performing classifier achieving an accuracy score of 0.8002 and a F1 score of 0.7974, but it still performs better than the dummy classifier. The best performing model was logistic regression which got an accuracy score of 0.8336 and a F1 score of 0.8342 on training dataset 4 which is signifanctly better than the trivial baseline of the dummy classifier as can be seen in the table. A confusion matrix was created for the logistic regression model to check whether the classifier had some sort of bias in what kind of errors it makes, see Figure **??**.

Overall it can be seen that the majority of models performed best on dataset 4, which was the dataset where only the rows where all annotators agreed were kept.

Based on these results the decision to use logistic regression for the classification task was made, furthermore we decided to apply the preprocessing used on dataset 4.

The model was then used to classify the comments in the test set and achieved an accuracy score of 0.8589 and a F1 score of 0.8588.

4 Discussion

4.1 Misclassifications

A confusion matrix was also created to analyse how the model performed on the test set, see Figure 2. Looking at the confusion matrix we can see that the model performs very evenly. The amount of false positives are very similar to the amount of false negatives which we think is a good sign for our model since it makes the model balanced and shows that it doesn't have some kind of big flaw or misses out on some important distinction.

To look at what type of comments the model misclassified we looked at some random misclassified comments (see Table 3) and analyzed if we could think of any reason why. In general we found that many of the misclassified comments were really hard to classify as they were written in a way that was hard to understand, some also had abbreviations or were based on sarcasm which makes it more understandable as to why the model couldn't predict the correct label. Many misclassified comments also brought up both positive and negative sides with the vaccine but stated their final opinion in the end which of course makes it very difficult for our model to predict. However there were also cases which we thought weren't very challenging and thought that the model should have done better.

For example looking at comment 1 in Table 3, we can see that it includes the abbreviation FTW which is positive but the model probably has a very hard time to classify this comment because the comment is so short and FTW plays a big role in it. Comment 2 is a difficult one since it brings up negative aspects with the vaccine, but the comment mentions getting the vaccine which probably confuses the model, however we are not sure how we would classify this comment ourselves. Comment 3 does not say much at all and lacks context and therefore it is impossible to classify both for us and for the model. Comment 4 is an example of a comment which we believe the classifier should have classified correctly since there is no obvious part of the comment which makes it hard to classify.

4.2 Feature importance

We have plotted the 10 most important features for classifying a comment as pro vaccination and the 10 most important features for classifying the comment as anti vaccination, see Figure 3. "Posion" is the most important word to help identify if a comment is anti vaccination which seems logical. Other important words for identifying anti vaccination comments are for example "never", "experimental" and "forced". The most important feature for finding pro vaccination comments turned out to be "ventilator" among other words like "get", "selfish" and "antivaxxers".

4.3 Tuning the vectorizer

Some tuning of the vectorizer was done during the development of our system. As previously mentioned we decided to vectorize the data with the parameter *lowercase=True* meaning that all the words get converted to lowercase. An idea was to set this parameter to False since maybe some words written in uppercase might be an indicator of the label. However we tried this and saw slightly worse results than with *lowercase=True*. We also use stop words in our implementation, meaning that we filter out the most common words which do not say anything meaningful about the data. We got the best results with filtering out the 3 most common words. It is probable that even more tuning with the vectorizer would give a slightly better performing model.

5 Conclusion

The goal of the experiment was to classify social media comments as pro or anti COVID-19 vaccinations. The data was processed and a logistic regression model was developed. The final model achieved an accuracy score of 0.8589 and a f1 score of 0.8588 on the test set. Even though the model is not perfect we can conclude that it is possible to build a fairly accurate model to classify social media comments about vaccination.

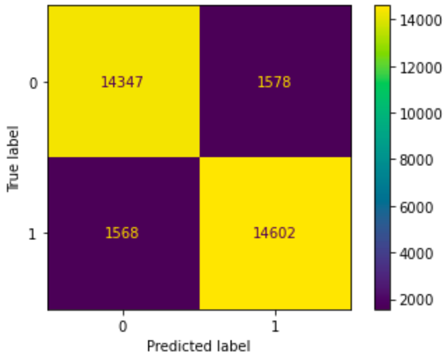

Figure 1: *Confusion Matrix for the training set. 0 are anti-vaccination comments, 1 are pro-vaccination comments.*

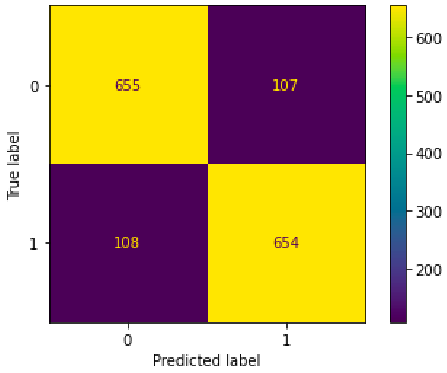

Figure 2: *Confusion Matrix for the test set. 0 are anti-vaccination comments, 1 are pro-vaccination comments.*

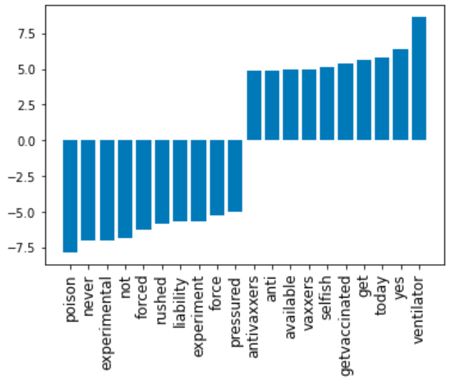

Figure 3: *Feature importances for the logistic regression classifier. A large positive number indicates that the word is important for classifying a comment as pro-vaccination. A large negative number indicates that the word is important for classifying a comment as anti-vaccination.*

	Dataset 1	Dataset 2	Dataset 3	Dataset 4
Dummy Classifier	Accuracy: 0.5265 F1 score: 0.0	Accuracy: 0.5247 F1 score: 0.0	Accuracy: 0.5036 F1 score: 0.6699	Accuracy: 0.5038 F1 score: 0.6701
Random Forest Classifier	Accuracy: 0.7681 F1 score: 0.7421	Accuracy: 0.7761 F1 score: 0.7521	Accuracy: 0.7957 F1 score: 0.7936	Accuracy: 0.8002 F1 score: 0.7974
Bernoulli NB	Accuracy: 0.7795 F1 score: 0.7571	Accuracy: 0.7893 F1 score: 0.7698	Accuracy: 0.8086 F1 score: 0.8029	Accuracy: 0.8091 F1 score: 0.8035
Multinomial NB	Accuracy: 0.7795 F1 score: 0.7571	Accuracy: 0.7893 F1 score: 0.7698	Accuracy: 0.8086 F1 score: 0.8029	Accuracy: 0.8091 F1 score: 0.8035
Linear SVC	Accuracy: 0.8002 F1 score: 0.7875	Accuracy: 0.8819 F1 score: 0.801	Accuracy: 0.8322 F1 score: 0.8331	Accuracy: 0.8319 F1 score: 0.833
SGD Classifier (SVM)	Accuracy: 0.796 F1 score: 0.7804	Accuracy: 0.8069 F1 score: 0.7939	Accuracy: 0.8262 F1 score: 0.8259	Accuracy: 0.8267 F1 score: 0.827
Logistic Regression	Accuracy: 0.8008 F1 score: 0.7876	Accuracy: 0.8099 F1 score: 0.7984	Accuracy: 0.8333 F1 score: 0.834	Accuracy: 0.8336 F1 score: 0.8342

Table 1: Accuracies and F1 scores for each of the models. (See descriptions of the data sets in Table 2).

Dataset 1	Dataset 2	Dataset 3	Dataset 4
Selecting the majority label (that isn't -1)	Removing all rows with a label -1, select the majority label for the remaining rows	Removing all rows with label -1 and those with equally many 0's and 1's, select the majority label for the remaining rows	Keeping only the rows where all annotators have agreed

Table 2: Descriptions of each of the versions of data sets.

	Comment	Actual Label	Classified Label
1	A big thanks to Germany! Biontech FTW! :-)	1	0
2	About 4 months ago, I had COVID-19. I just got my first vaccine and I became very ill with fever, chills, body aches and dizzy spells. Now I'm afraid to get the second dose. Any suggestions?	0	1
3	All I can say is wow!!	1	0
4	Am vaccine free and glad i didn't comply. Feel bad for those who did.	0	1

Table 3: Examples of some comments that got misclassified by the logstic regression model.