DAT341 Programming assignment 3: Stance classification

Yahui Wu
Chalmers University of Technology
MPMOB
yahuiw@chalmers.se

Tianshuo Xiao
Chalmers University of Technology
MPMOB
tianshuo@chalmers.se

Abstract

Different people have different opinions about vaccines. In this paper, by collecting a large number of comments on whether to vaccinate or not, and using it as a training set. The text is vectorized by TfidfVectorizer and try different classifiers algorithms, such as BernoulliNB, MultinomialNB, LinearSVC, LogisticRegression and MLPClassifier, to comapre cross-validate score by the training set. The linearSVC get the higher score and tune hyperparameters to get the best better performance. Plot confusion matrix and ROC curve to evaluate the model. In addition, analyze different models' algorithms and explore the causes of prediction errors. Finally, in order to investigate the important influencing factors in the text, identify the words with higher weights given by different word classifiers for both support and opposition cases, and remove some high frequency words for comparison with the trained models.

1 Introduction

The Covid-19 epidemic led to a global catastrophe. With the development and promotion of a vaccine, the outbreak was gradually controlled and disseminated. However, due to the uncertainty and side effects of the vaccine, different people have different views on whether to get vaccinated or not. Therefore, we collected a large number of comments from websites such as YouTube and chose a classifier to predict people's comments.

2 Data collection and processing

2.1 Data consensus

In our data, there are 37,885 data sets in total. People who have the same sentiment marker for 34,182 data sets, which consensus rate is about 90.23%. In the non-consensus data, there are 2087 data sets, which means that 56.36% of people have the opposite sentiment indicators. So, we think the data is reliable.

2.2 Data processing

There is a consensus issue that the annotations of some comments by different users are conflict. In some cases the comments have the annotation where one of the label 0, 1, -1 occupies the leading position and it is easy to determine what label we should take. However, in some worse cases, the labels show a tie like -1/0/1. We would like to have no conflict training set that is beneficial to the prediction without lots of data wasted at the same time. To do the pre-process job, we came up with a voting policy that the comments showing a tie or comments with more or equal to two dominated labels could be dropped while the rest comments should be assigned a label equal to the majority label in the annotation.

2.3 Data classification

After processing, we have 32,452 sets of data in total. There are 16340 sets of positive data, which represent 50.35%, and 16112 sets of negative data, which represent 49.65%. Therefore, this data set is relatively balanced through our data processing.

3 Data features

The original training set we handled with is comments about Covid-19 vaccine consisting of a list of documents, where each document (comment of each instance in this case) is represented as a single string in English. To make the training set interpretable for machine learning classifiers, we would like to process it and obtain numerical features for each instance in the data.

At first, we removed some special symbols, converting uppercase to lowercase, and emoji symbols, but the accuracy of the model decreases. This may be because some emoji have been capitalized with the commenter’s sentiment, which may be either supportive or ironic. Therefore, we decided not to process our training set.

We used TfidfVectorizer to convert the comments to rows of vectors. The number of rows equals to the number of instances from training set and each element in the row vector is a numerical value that represents the probability of a certain word appearing in the document. By processing the comments in training set using TfidfVectorizer, we could obtain the feature of each instance, where the number of features reflected how many words in the bag-of-words and the feature value was the word frequency.

4 Learning algorithms select and tuning

4.1 Different learning algorithms

The task is to predict a given comment whether it is pro-vaccine or anti-vaccine. We have to do the prediction and label a test comment as 1 (pro-vaccine) or 0 (anti-vaccine). Therefore, we considered some machine learning algorithms that were widely used to solve such a binary classification problem. When it comes to the text classification problem, the first thing we may come up with is Naïve Bayes model. In this case, we tried two different types – BernoulliNB and MultinomialNB. Instead of only considering whether a certain word in the document has appeared or not using BernoulliNB, MultinomialNB can take the frequency of a word being used in the document into account. Therefore, MultinomialNB could have better performance than BernoulliNB. LogisticRegression is another useful way to handle binary classification problem. It calculates the probability that a certain comment belongs to the positive class (pro-vaccine) using sigmoid(1) function. Perception algorithms from scikit-learn uses sign function to assign the instances to positive class and negative class which could be suitable for our task. Similarly, LinearSVC uses sign function to classify instances but what different is that it makes the distance between those points near the hyperplane as short as possible thus LinearSVC has better robust. Besides the machine learning algorithms, we can also implement some neural networks method like MLPClassifier(2).

4.2 Learning algorithms selection

We use cross-validation to evaluate our models. Cross-validation evaluates the performance of a model, such as whether the model generalizes well and whether its over-fits. By the cross-validation method, we derive the specific score values in 4.2, as shown in Table 1. The LinearSVC has the highest value, so we choose LinearSVC as our computational model.

Classifier	cross_val_score
Perceptron	0.771
BernoulliNB	0.795
MultinomialNB	0.807
LinearSVC	0.815
LogisticRegression	0.813
MLPClassifier	0.803

Table 1: Cross-validation scores of different algorithms

4.3 Hyperparameters tuning

We search for the optimal parameters by GridSearchCV. We select C and Penalty to optimize. Penalty specifies the norm used in the penalization. About C, the larger the C, the greater the penalty for misclassified samples, and therefore the higher the accuracy in the training samples, but the lower the generalization ability, i.e., the lower the classification accuracy of the test data. On the contrary, decreasing C allows for some misclassified samples in the training sample and has a high generalization ability.

First, we take the value of ’C’ as [1, 10, 100, 1000] and ’Penalty’ as [’l1’, ’l2’]. Then, we found the best ’C’ was 1 and ’Penalty’ was ’l2’. Next, We narrowed the range of values of C to [0.5, 0.6, 0.7, 0.8, 0.9, 1, 1.1, 1.2, 1.3] and the best ’C’ was 0.5. Finally, We narrowed the range of values of C again to [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7], the best ’C’ was 0.4.

Overall, the best parameters are as follows: ’C’ was 0.4 and ’Penalty’ was ’l2’.After optimization of the parameters, the cross validation score is 0.818, which has improved. The accuracy score is 0.85367 and F1 score is 0.8526.

5 Model evaluations

5.1 Compare to a trivial baseline

We compare our classifier to a baseline by DummyClassifier, which is a trivial majority-class classifier.We mainly compared the accuracy score and F1 score between them, as Table ??

From the table we can see that the DummyClassifier accuracy score is 0.5 with F1 score of 0.667.We can compare our classifier with this baseline.

The accuracy of LinearSVC with default parameters is 0.834, and after we adjust the parameters, the accuracy is 0.854, which has improved by 0.02. For F1 score, the LinearSVC with default parameters is 0.845, and after adjusting the parameters, the value is 0.853, which has improved by 0.008. The correct rate is not significantly improved, but it is much better than the trivial baseline.

Classifier	accuracy score	F1 score
DummyClassifier	0.5	0.667
LinearSVC	0.834	0.845
New LinearSVC	0.854	0.853

Table 2: Compare to a trivial baseline

In order to compare with the trivial baseline, we plotted the ROC curve as Figure 1 The ROC curve can remain unchanged when the distribution of positive and negative samples in the test set changes. Class imbalance often occurs in real data sets, where there are many more negative samples than positive samples (or conversely), and the distribution of positive and negative samples in the test data may also change over time.

In the ROC curve of our model, the AUC value is 0.928, which means that our model outperforms random guesses. This model can have predictive value if the threshold value is properly set.

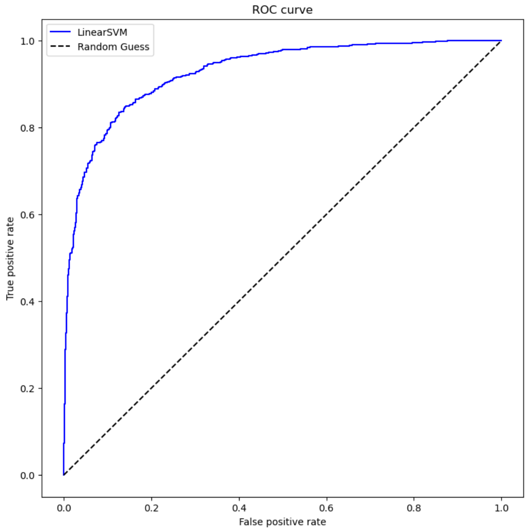

Figure 1: ROC curve

5.2 Model's confusion matrix

We plot the confusion matrix as Figure 2. The system generates some classification errors when it performs classification, with 106 sets of false positive data and 117 sets of false negative data. Then, the true positive rate is calculated to be 0.8425 and the true negative rate is 0.8464, which shows that our designed classifier can classify the data well and the quality of the system is good.

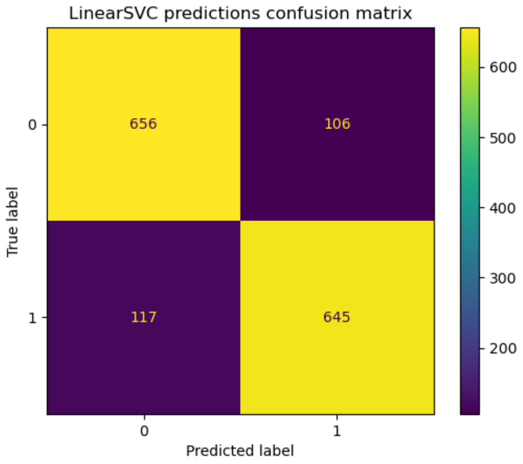

Figure 2: Example of a figure.

5.3 Discussion on errors system makes

The classifier sometimes has the problem of prediction errors, predicting positive attitudes as negative or negative attitudes as positive. We think it is possible that in the SVC classifier, the words with different frequencies of appearance are given different weights, and the words in some sentences appear more frequently, which leads to prediction errors. We use two examples to explain this below. ”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.”is labeled 1 in the test set, but 0 in the prediction set.The sentence does not contain words that express the commenter’s obvious attitude, but contains stupidity, a word that is given high weight in the category of 0, causing the classifier to predict it as negative.(Table 5)

Similarly, ’95% effective for a virus that kills at 0.5%.’ In this sentence effective is present, in general effective occurs more frequently under pro-vaccine comments(Table 4), so the classifier predicts it as positive. The implication of this statement is that the virus is not as lethal as it could be, with a certain irony. Actually this sentence indicates that the commenter is against vaccines.

5.4 Important feature

To find which features the model considers important, we selected some of the more frequently occurring words and then presented them before calculating the accuracy score as Table 3.

From the table, we find that when we removed some words that appear more frequently, the correct rate of the model decreases or remains constant, which means that some important words can affect the correct rate of the classifier, but the effect is not significant.

Remove word	accuracy score	F1 score
should	0.849	0.849
the	0.827	0.821
vaccine	0.833	0.829
and	0.847	0.844
sick	0.852	0.852
safe	0.851	0.849
virus	0.854	0.854
Covid	0.848	0.848
healthy	0.853	0.852
against	0.848	0.846

Table 3: Important features

In addition, we list the weighted values of different words and the top ten words in each weighted value.

We can see that words like get vaccinated are pro-vaccination and have a higher weight within the group that is in support of vaccination. However, words like poison, which have negative emotional overtones, have a higher weight in the group that is against vaccination. These features the model considers important.

word	weight
antivaxxers	3.1036755455328047
anti	2.1493604951225636
vaxxers	1.8828262457137213
available	1.8749738707142205
ventilator	1.8704272508732038
getvaccinated	1.8593770943293597
selfish	1.8445388107664902
response	1.7924575542006274
antivaxers	1.7576980310846586
understand	1.729812000658641

Table 4: Important features of pro-vaccine comments

word	weight
poison	-2.5849514724546676
never	-2.341733213443355
not	-2.332811747834075
forced	-2.1682645915674654
experimental	-2.1283070222593476
rushed	-2.0324628611877986
liability	-1.9552883836986452
pressured	-1.901983996627582
experiment	-1.8756326027538859
force	-1.8505314831038615

Table 5: Important features of anti-vaccine comments

6 Conclusion

In this paper, the data are processed to obtain a better training set and the text is processed by TfidfVectorizer. After comparing many classifier algorithms, LinearSVC was chosen as the model for hyperparameter tuning, and got a high accur score. The model was compared with DummyClassifier by confusion matrix and ROC curve, and the model performed well. Finally, the weighted values of keywords under different attitude tendencies were obtained, and the model was evaluated again after removing some high-frequency words, which was found to have little effect on it.

References

[1] Summa, M. G. (2012). Statistical Learning and Data Science. CRC Press.

[2] Raschka, S., amp; Mirjalili, V. (2019). Python machine learning: Machine learning and deep learning with python, scikit-learn, and tensorflow 2. Packt.