Political stance classification

Nilufar Hatamova
Department of Computer Science and Engineering, University of Gothenburg, Sweden
gusniluha@student.gu.se

Abstract

Text classification is a process of training a supervised machine learning algorithm in order to classify text data into groups defined by labels. In this project, I used a dataset that is collected from different social media networks such as Twitter and Facebook by students of Applied Machine Learning course at GU and Chalmers University. The dataset contains positive and negative comments and labels about the opinions of different peoples regarding BREXIT.
I applied different text classification algorithms as Multinomial NB, Logistic regression, Linear SVC, KNN, Random Forest Classifier, etc. to define whether a given comment expresses a pro-Brexit or anti-Brexit position. The goal of this paper is comparing the performances of specified machine learning approaches and selecting the best technique among them.

Introduction

In this project I have experimented with various text classification techniques to classify text data. The data contains positive and negative comments about whether UK should leave the EU which is collected from social media.

The rest of the paper is organized as follows. It starts by data pre-processing which includes the list of pre-processing steps. The next section discusses model selection and contains information about which classification algorithms are used and how the final model is selected. Finally, the results of selected model are presented.

Data pre-processing

Data pre-processing is one of the essential steps in the text classification process that prepares data for algorithms. The biggest advantage of data pre-processing is raising the accuracy and generalizability of the model. The applied pre-processing techniques include:

- punctuation removal
- stemming words to their "root" form
- removing stop words such as "*I, the, is* "
- digit removal

In the training set, the labels of comments include the annotations from different annotators. There are some disagreements between annotations which is 18.6% of all annotations. The steps for making labels cleaner are defined as:

- If all annotations are the same, the label is selected as that annotation
- If there are some disagreements between annotations the label is selected same as the majority of annotations.

After completing the steps mentioned above, the data points is removed

- If the label is -1 meaning that it's impossible to understand a comment as pro-Brexit or anti-Brexit
- Number of pro-Brexit and anti-Brexit annotations are the same

Balanced data

The number of comments in the training set is almost equally distributed between classes which means both pro-Brexit and anti-Brexit comments have the same effect to train the model. After cleaning the text corpus, the percentages of pro-Brexit and anti-Brexit comments are distributed as 51.6% and 48.4%, respectively in the training set.

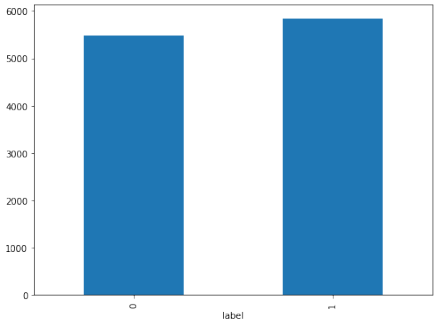

Figure 1: The distribution of training set.

Text representation

The features for text classification are non-numerical. TfidfVectorizer() is used to represent these non-numerical features as a matrix of TF-IDF features. TfidfVectorizer() measures the weight of a word in the document. This weight is a statistical measure used to evaluate how important a word is to a document in a collection or corpus.[1]

Model Selection

Different classification techniques were experimented by building a pipeline. Several models are trained for comparison reasons.

Trivial baseline

The decision tree classifier was used as a baseline classifier which is one of earlier classification algorithm for text mining. The classification pipeline was built with the following steps:

1. To represent these features as a matrix of TF-IDF weight: TfidfVectorizer(),
2. Run the classifier: DecisionTreeClassifier(),

After fitting the model, the baseline test accuracy was 0.69.

Trying different classification techniques.

When working on the text classification problem with a given data, I tried different text classification techniques - Multinomial NB, Logistic regression, Linear SVC, KNeighborsClassifier, Random Forest Classifier, compared their classification performance to choose the most accurate one. The Table 1 shows their accuracy for both training and testing set.

Algorithm	Train_accuracy	Test_accuracy
MultinomialNB	0.880612	0.766379
LogisticRegression	0.862937	0.779310
GradientBoostingClassifier	0.736303	0.705172
KNeighborsClassifier	0.550990	0.556034
LinearSVC	0.940880	0.787069
RandomForestClassifier	0.999381	0.774138

Table 1: Training and testing set accuracy

KNeighbors Classifier has the least accuracy for both training and testing set among the used algorithms. LinearSVC and RandomForestClassifier has the highest accuracy for training set, 0.94 and 0.99, respectively.

The model was selected based on test set accuracy. Because LinearSVC has highest test accuracy that equals 0.78, LinearSVC selected as best model.

Hyperparameter Tuning

The performance of an algorithm can be highly related to the choice of hyperparameters of the algorithm. A good selection of hyperparameters can increase the accuracy of the learning algorithm. The hyperparameter optimization was executed by using Grid Search over the selected-Linear SVC algorithm.

The most significant parameter that was tuned for the Linear SVC algorithm is the regularization parameter – *C* and for TfidfVectorizer() is *ngram_range*. After performing the grid search, the best hyperparameter was chosen as C =1 and ngram_range=(1,2) and the model accuracy increased to 0.81. In comparison with the baseline model, the accuracy of the selected model is 10% higher.

Model Evaluation

Several evaluation metrics were used to assess the quality of the selected method such as accuracy, recall, precision, F1score which are widely used in binary classification problems. Selected model has measurements as in Table 2.

classes	precision	recall	f1score
anti-Brexit	0.81	0.79	0.80
pro-Brexit	0.82	0.83	0.83

Table 2: Classification report

From the Table 2, It can be seen that the model has better results for pro-Brexit comments.

Additionally, AUC - ROC curve is also used as a performance measurement for this classification problem. AUC close to 1 means the model is good at distinguishing between pro-Brexit and anti-Brexit comments. Figure 2 contains the AUC - ROC curve for the selected model. The AUC value is equal to 0.9.

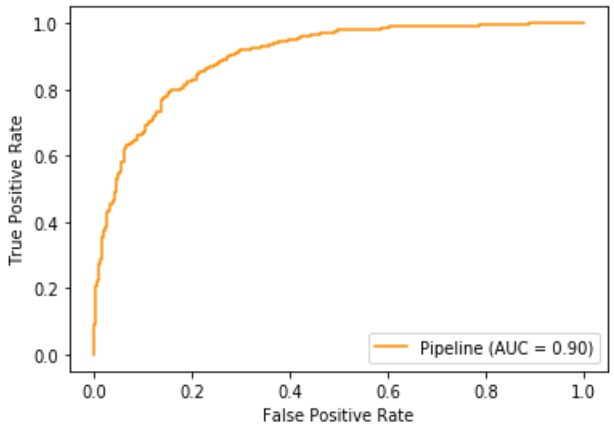

Figure 2: AUC - ROC curve

Confusion matrix
Confusion matrix, also known as an error matrix which is a specific table that visualize of the performance of an algorithm [2]. It gives information about both incorrectly and correctly predicted cases. Most of the performance metrics are calculated based on Confusion matrix.

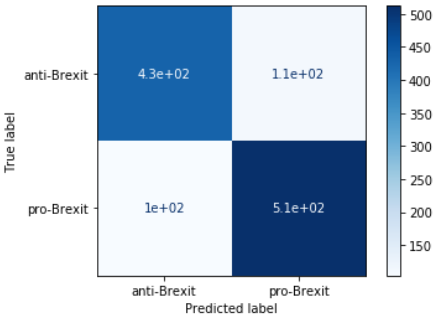

Figure 3: confusion matrix

The confusion matrix in Figure 3, It can be seen that almost 950 comments are classified correctly, however more than 200 comments are misclassified.

Misclassification
As mentioned above, approximately 18% of comments are misclassified. One of these comments is:

"so many remainers that hate our country" I think you will find that those who love their country voted Remain.

There reason for this misclassification can be that this comment contains both negative and positive words. Because the text classification algorithms classified based on words, the comments with allusion can also be cause misclassification.

Feature importance
Each feature in the dataset has a score that identifies its importance the classification. The feature with higher score has a more significant effect in the classification process. The Figure 4 presents what features (words) are being used to make pro-Brexit and anti-Brexit classifications from the dataset.

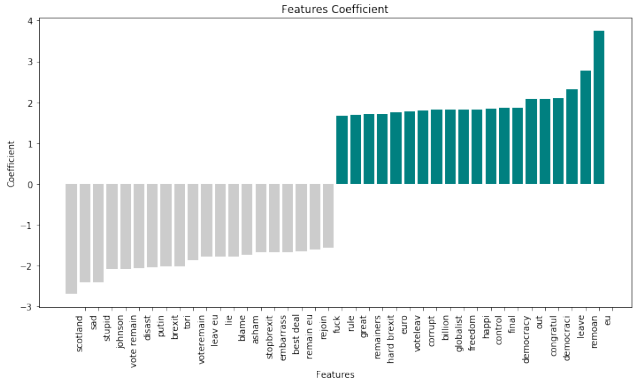

Figure 4: Top Features

The words like Scotland, sad, stupid, Putin has negative coefficients, which are used to classify comment as anti-Brexit. The words like freedom, democracy, leave, etc. have positive coefficients, which have significant influence to classify comment as pro-Brexit.

Conclusion

It was possible to get reasonable results from classifiers such as Logistic Regression, Linear SVC. The test accuracy of those algorithm was around 78% which is reasonable considering the narrow margin between positive and negative comments in many cases. Furthermore, the hyperparameter tuning was not done for all algorithms due to time and processing power constraints. It might be possible that other algorithms which are not selected perform better with optimal hyperparameters.

References

[1]. http://www.tfidf.com/

[2]. https://en.wikipedia.org/wiki/Confusion_matrix