Natural Language Processing with Disaster Tweets

Sina Ainesazi Dovom
Sina.ainesazidovom@studenti.unipd.it

1. Introduction

Natural Language Processing is the task of making computers to be able to understand the words and text spoken and written by humans.

Currently, there are many projects about this task, and companies showing a very strong enthusiasm for making models on text data to extract knowledge and make predictions about the future of any field and anything.

As a communication tool, Twitter is increasingly important in times of crisis. People use this social network to tweet about disasters and spread the news about them.

In this task sentiment analysis has been used on the data set provided by the company figure-eight to predict to see whether a tweet is about a natural disaster or not.

2. DataSet

The dataset that is used in this project, includes 7613 observations and 5 variables which are, ID, keyword, location, text, and target.

Each tweet has an ID, a keyword variable that indicates the type of tweet, a location column that describes the place about which the tweet is about, and a text column containing the tweet's text and lastly, the target variable that consists of two values 0 and 1, which 0 has been assigned to the tweets which are not about a disaster but vice versa the number 1 has been assigned to the tweets which are about a disaster.

Since the majority of the values for keyword and location columns were empty, and the ID variable does not have any useful information, these three columns have been dropped from the data set. The text variable itself contains all the required useful information, but the text data is in raw text which by itself, cannot be used as a feature. So it needs to be preprocessed.

For the sake of checking whether the data is balanced or not, the distribution of the target class has been checked by using a bar plot. Figure 1 shows the number of training data in 2 categories.

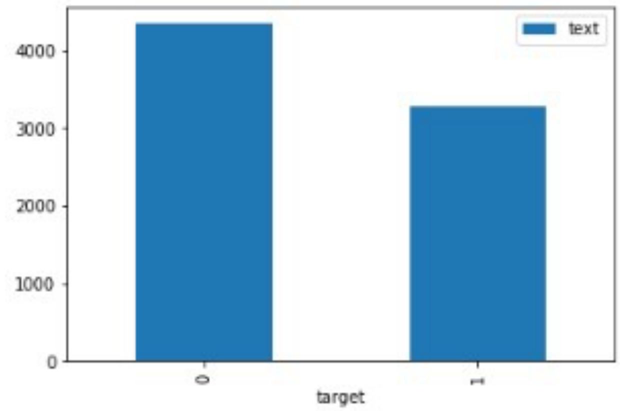


Figure 1 Distribution of the two categories

It is evident that there are more occurrences of 0 than 1, in the target variable. Still, the good thing is that the difference is not significant and the data is relatively balanced.

Before starting the preprocessing steps, the raw text should be examined to see what words are more frequent. The benefit of this is that it gives an idea of which words and signs are not helpful to use. Figure 2 illustrates the result of using the word cloud for this purpose.

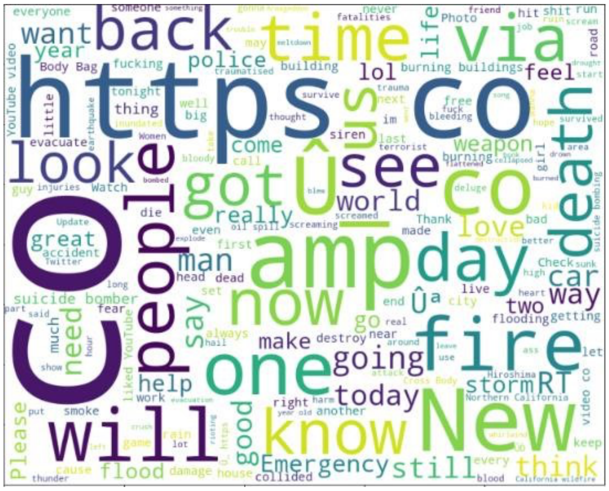


Figure 2 Most common words in tweets.

The preprocessing steps which have been taken into account are Removing punctuation, Removing stopwords, Conversion to lower case, and Stemming, for fulfilling these steps the nltk package has been loaded and used in this project. After having done the preprocessing step, for building machine learning models, there is a need to convert the raw text to word frequency vectors. There are several ways to do this, such as using CountVectorizer and HashingVectorizer, but the TfidfVectorizer has been used in this task since it's the most popular one and it is used as a weighting factor in text mining applications. In simple words, TF-IDF attempts to highlight important words which are frequent in a document but not across documents. By using TF-IDF, we are able to extract the important words from the corpus, reducing the data dimensions by removing words that are less important for analysis.

3. Methods

Many different models can be chosen to build a machine learning model for Natural Language Processing tasks, the first algorithm is the Logistic Regression. In natural language processing, logistic regression is the baseline supervised machine learning algorithm for classification, and also has a very close relationship with neural networks. This method in comparison to the other methods used in this study is easy to implement but it's sensitive to overfitting especially when the number of features is high.

The second algorithm which has been used in this project is the Random Forest algorithm, a random forest is an ensemble classifier that estimates based on the combination of different decision trees. In simple words, it fits several decision tree classifiers on various subsamples of the dataset. Also, each tree in the forest is built on a random best subset of features. Finally, the act of enabling these trees gives the best subset of features among all the random subsets of features. This method in comparison to the other algorithms used in this task is expensive if the number of observations and number of features is high in the data set, it's easy to implement and it's sensitive to overfitting.

The third method is the K Nearest Neighbors (KNN) algorithm which is a simple, supervised machine learning algorithm that can be used to solve classification problems. It's easy to implement and understand but has a major drawback of becoming significantly slow as the size of that data in use grows. In this case, having a large K may increase the risk of overfitting in KNN. So this method in comparison to the others works well when the number of both the features and the observations in use are not too big.

The last method which has been used in this task is Support Vector Machine, SVM is a supervised machine learning model that uses classification algorithms for two-group classification problems. Overfitting is less likely to occur with this method in comparison to alternative methods and it works better on unstructured data like text.

4. Experiments

In this part, the performance and the accuracy of the methods which have been used in this project will be discussed and in the end, the best and the most efficient algorithm will be revealed.

4.1. Logistic Regression

After fitting this model on the training data set, the accuracy of the prediction on the training data set was 0.89577, the accuracy for the validation data set was 0.79096 and also the total accuracy of this model on the test set was 0.78394.

4.2. Regularized Logistic Regression

The Logistic regression with different hyper parameters has been fitted on the training data set and the result of the accuracy after making the prediction on both the training and validation sets, it was obvious that the best performance happened when the constant C is equal to 5, which constant C is the inverse of the lambda coefficient. The accuracy of prediction on the training data set with this method was 0.96426 and the accuracy of the prediction on the validation data set was 0.79149. The performance of this model on the test set was 0.78853. Figure 3 shows the results of different values of C on both the training and validation data sets. Figure 3 shows the effects on the hyper parameter.

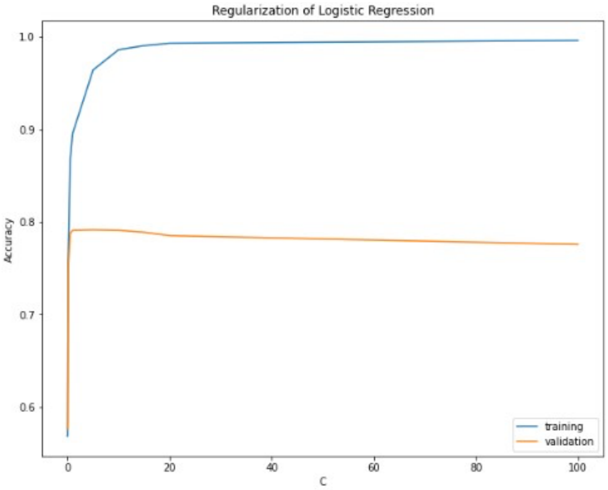


Figure 3 Effect of the hyperparameters

4.3. Random Forest Classifier

Also, the Random Forest classifier with the number of estimators equal to 10 and the entropy as a criterion has been fitted on the training data set and the accuracy of prediction on the training data set with this method was 0.97355 and since the accuracy of the prediction on validation data set was 0.78571, it seemed to have an overfitting problem. Also, the total accuracy of this model on the test set was 0.76831

4.4. K-Nearest-Neighbor

After fitting this model on the training data set and predicting on the training and validation data sets it was obvious that, as the number of nodes increases, the algorithm performs worse. However, the best prediction accuracy was obtained when the value of K was equal to 9 and the accuracy of prediction on the training data set with this method was 0.82168 and the accuracy of the prediction on the validation data set was 0.77993 and the performance of this model on the test set was 0.71958. Figure 4 shows the performance of this algorithm with different values of K.

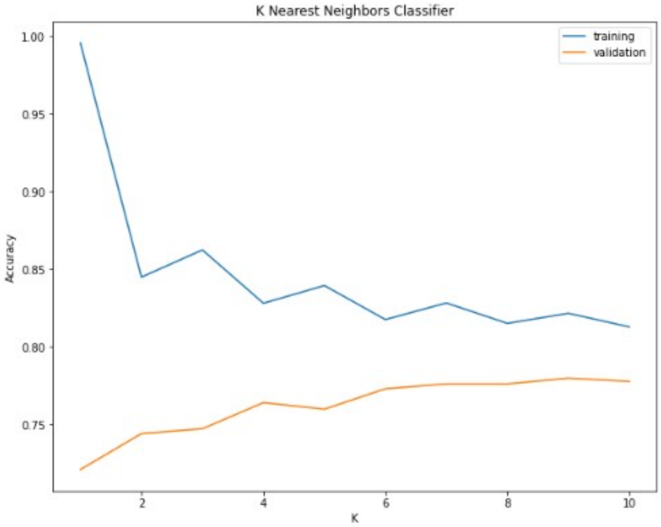


Figure 4 KNN accuracy with several K values

4.5. Support Vector Machine with a linear function

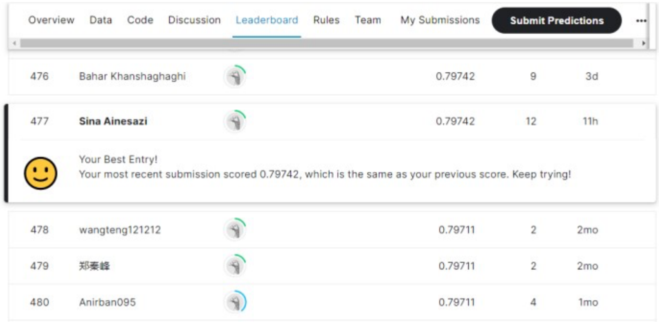
The support vector machine with a linear function and the constant C equal to 1 fitted on the training data and the accuracy of prediction on the training data set with this method was 0.93589 and the accuracy of the prediction on the validation data set was 0.8004 and the performance of this model on the test set was 0.79436.

4.6. Support Vector Machine with RBF function

Once again the Support Vector Machine algorithm with RBF function and different values for constant C fitted on the training data set and it was clear that the performance of this model was better when the constant C was equal to 1. The accuracy of prediction on the training data set with this method with C = 1 was 0.97284 and the accuracy of the prediction on the validation data set was 0.79779 and the performance of this model on the test set was 0.79742. There might be an overfitting problem here.

5. Conclusion

From the results of all the methods, it was concluded that the support vector machine algorithm with RBF function and C = 1 had the best accuracy and result on the test set of all the methods used. Figure 6 shows the accuracy and ranking of the submission on the Kaggle website.



476	Bahar Khanshaghaghi	0.79742	9	3d
477	**Sina Ainesazi**	0.79742	12	11h

Your Best Entry!
Your most recent submission scored 0.79742, which is the same as your previous score. Keep trying!

478	wangteng121212	0.79711	2	2mo
479	郑秦峰	0.79711	2	2mo
480	Anirban095	0.79711	4	1mo

Figure 6 Results on the Kaggle submission

References

[1] https://monkeylearn.com/blog/introduction-to-support-vectormachines-svm/
[2] CS229: Machine Learning (stanford.edu)
[3] Pattern Recognition and Machine learning, Christopher M. bishop,2006. https://www.microsoft.com/en-us/research/uploads/prod/2006/01/Bishop-Pattern-Recognitionand-Machine-Learning-2006.pdf
[4] https://www.techtarget.com/searchenterpriseai/definition/natural-language-processing-NLP