Natural Language Processing with Disaster Tweets

Mojtaba Amini
Mojtaba.amini.1995@gmail.com
https://github.com/mojeee/Machine_and_Deap_Learning_ModeA.git

1. Introduction

Natural Language Processing refers to understanding the text and spoken words like a human. There are a bunch of ambiguities in human language processing because of irregularities in language, idioms, different types of usage expectations, etc. [1]. Nowadays, several NLP projects have been developed to improve the language understanding of the computer. Sentiment analysis, speech recognition, word sense disambiguation, natural language generation, etc., are examples of subjects in this area. In this study, NLP will be used for sentiment analysis to predict disaster tweets.

In this day and age, Twitter has become an important communication channel in which also people uses Twitter in emergency situation. To get informed about an emergency as soon as possible, many companies use the NLP to monitor Twitter. Monitoring the online data of Twitter is a little bit tricky, but a supervised learning model can be developed and can be used for prediction in the future.

2.1. Dataset

The dataset consists of ID, keyword, location, text, and target columns. The last column is the label assigned to the text that represents an emergency. The ID is just a number that is set to a tweet. The keyword column represents the categorical tag of a tweet. Location refers to the tweet's location, and text column refers to the tweet's text.

First of all, the ID, keyword, and location column has been removed because the ID and location column cannot give us too much information, at least in this step of work, and will be used for more advanced analysis. In addition to that, there are plenty of details in the tweet's text. Because of that, the keyword column has been ignored in NLP analysis. Furthermore, the text column needs some cleaning in the dataset to be prepared for study. The data cleaning stage removes the special punctuation characters, and other words like an auxiliary verb cannot help the model.

After a simple data cleaning, the tweets are ready for the primary analysis. In this stage, the first thing that comes to mind is plotting a word cloud from all tweets to see which words are more common and which type of data cleaning the data sets might need before running a Machine Learning (ML) on the data. Focusing on the word cloud shows that more data cleaning can be helpful, but it takes too much time, and in this stage, it's better to leave it like that and focus on it in the future.

Figure 1 Word Cloud representation of all tweets

To prepare data for a ML model, the CountVectorizer function has been used in this study. Since values in the columns don't have different ranges, they don't need any scalar function to be run on it. The last analysis that comes to mind is checking the number of data values in two categories to see whether the data is balanced. Figure 2 shows that the data is balanced.

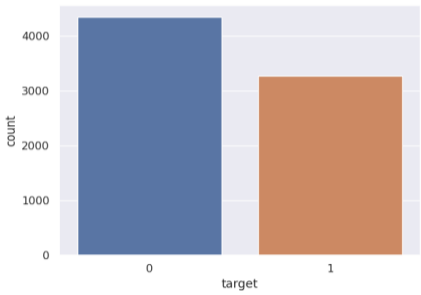

Figure 2 Number of training data in each category

2.2. Method

For choosing an ML model in this part, different options are needed to evaluate NLP in more detail. The first model that comes to mind is the Logistic Regression model. This model can easily be implemented for classification and has a lower cost than the other advanced model with many features and samples [2]. To achieve a better result, it is better to compare the performance of the different methods for this specific study. Implementing the different approaches is not reasonable in any study. It is better to delve into the details of the machine learning methods to have a clear idea about every method [3].

The logistic regression model is simple and has a lower cost, but in some cases where we have several features, the overfitting of the problem is too probable. The KNN and decision tree may also have an overfitting problem here. There are many samples and features, so KNN may need a larger K for this method, or the tree more complex, so the cost and probability of the overfitting will increase. Among other ML methods for a classification problem, although the SVM has more cost, it seems that it can be a good choice for this study because it is less prone to overfitting and it has less error in comparison to other models.

The Neural Network is the last approach that comes to mind in this study. The NN is more complex and needs more time for choosing the proper parameter for it, and it takes time to find the best parameter. By the way, in this study, there is this expectation that a good result will be achieved from SVM and NN. In conclusion, in this study, we work more on SVM and NN models, and also the Logistic regression and KNN model will be implemented to compare with them. A summary of the essential ideas considered in this stage has been reported in the table 1.

Table 1 Comparing different ML models in this study

Logistic Regression	Support Vector Machine	K-Nearest Nodes	Decision Tree	Neural Networks
Simple to implement	Less error compares to logistic regression	Work well the features are not too much	Simple to implement	Difficult to implement
Lower cost	Higher cost	Higher cost	Higher cost	Higher cost
Lower parameter for tuning	Work better for unstructured data like image and text	Work well when the number of samples is not too much	Expensive for large sample and large feature	A lot of functions and parameters for tuning
Vulnerable to overfitting	Less vulnerable to overfitting		vulnerable to overfitting	Vulnerable to overfitting

3. Experiments

In this section the performance of mentioned approach will be assessed to reach a final conclusion that which one is better for the NLP of tweets. Since every method has several parameters, we consider the critical parameter and evaluate their impact on performance of the related method.

3.1. Logistic Regression

The important parameter for logistic regression is the lambda coefficient, which controls regularization in logistic regression. The results showed that the best performance is achieved when the inverse of the lambda coefficient is equal to one. Also, the total accuracy of the model with this lambda for the test set was 78%.

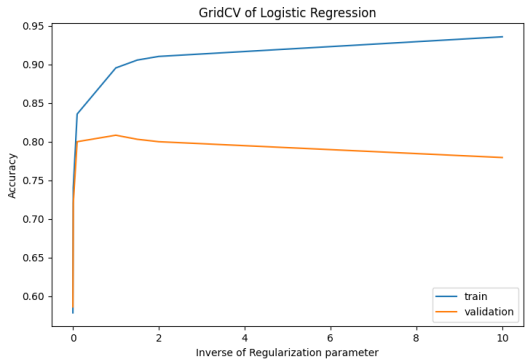

Figure 3 Effect of Lambda in Logistic Regression

3.2. K-Nearest Machine (KNN)

The number of nodes considered for calculation in the calculation of KNN has an important effect on the performance of this method. It is evident in figure 4 that by increasing the number of K the performance of the model tends to decrease for the train and evaluation dataset. Furthermore, the performance of the KNN for the evaluation set is about 70%, in the best case, which there is a considerable difference between this model and logistic regression.

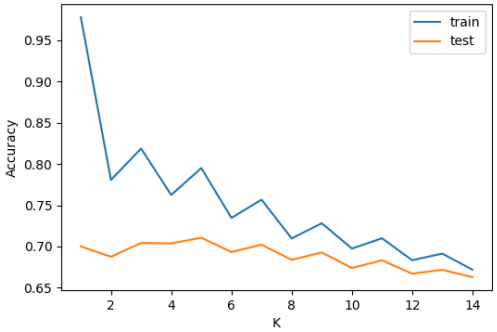

Figure 4 Accuracy of KNN model in different K

3.3. Support Vector Machine (SVM)

Two critical parameters of the SVM are the function that SVM use in the method and constant C. the performance of the method has been evaluated in different C for two functions, linear and RBF. The results showed that the non-linear function, RBF, had better performance and better results when C was equal to 1.

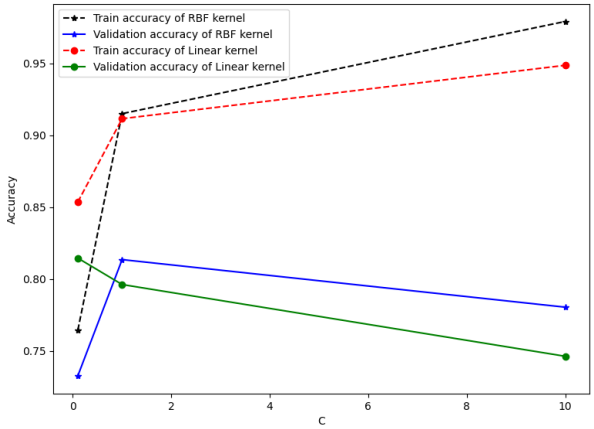

Figure 5 Analysis of the performance of SVM with different parameters

Figure 5 shows that the training accuracy improves with increasing C, but the validation accuracy reduces due to the overfitting problem.

3.4. Neural Networks (NN)

The behavior of epoch and batch size are studied in this study. These two parameters have a significant impact on the results. The results showed that by increasing the number of epochs, the accuracy of the train data set increases, but the validation set's accuracy decreases. Furthermore, similar behavior can be seen in figure 7. In figure 7, by increasing the number of the epoch, the loss of the NN model for the train data set decreases, but the loss of the validation set increases due to the overfitting problem.

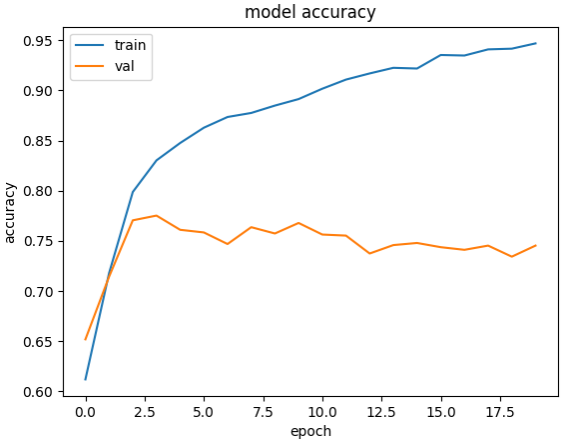

Figure 6 Accuracy of NN with different epoch

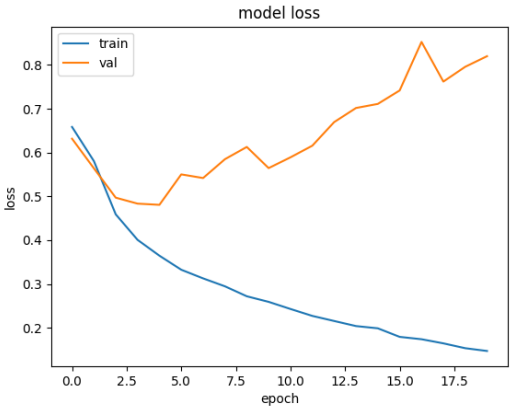

Figure 7 Loss of NN with different epoch

Results of the study of the NN model in different batch sizes and epochs showed that the NN model has better performance when the epoch is 5, and the batch size is equal to 4.

```
Best: 0.734011 using {'batch_size': 4, 'epochs': 5}
0.734011 (0.025241) with: {'batch_size': 4, 'epochs': 5}
0.716804 (0.021173) with: {'batch_size': 4, 'epochs': 10}
0.704588 (0.020954) with: {'batch_size': 4, 'epochs': 20}
0.725342 (0.024197) with: {'batch_size': 8, 'epochs': 5}
0.730068 (0.011588) with: {'batch_size': 8, 'epochs': 10}
0.706555 (0.012877) with: {'batch_size': 8, 'epochs': 20}
0.661772 (0.051950) with: {'batch_size': 16, 'epochs': 5}
0.731646 (0.019017) with: {'batch_size': 16, 'epochs': 10}
0.722452 (0.021694) with: {'batch_size': 16, 'epochs': 20}
0.667543 (0.005082) with: {'batch_size': 20, 'epochs': 5}
0.707741 (0.033176) with: {'batch_size': 20, 'epochs': 10}
0.729544 (0.016357) with: {'batch_size': 20, 'epochs': 20}
```

Figure 8 Accuracy of the NN with different epoch and batch size

4. Conclusion

In conclusion, the SVM with RBF kernel function and C equal to 1 had better performance on the test set. Figure 9 shows the result of the SVM with the mentioned specification. The codes also uploaded in the GitHub. The address of the GitHub can be found in the first page and in the reference section.

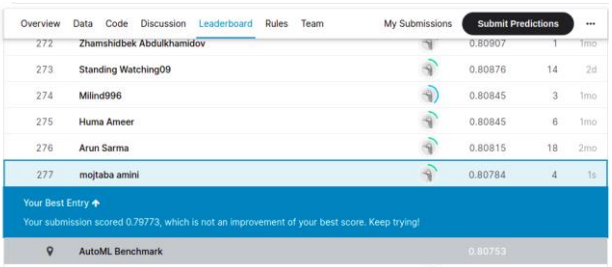

Figure 9 Result of the test data in Kaggle

References

[1] https://www.ibm.com/cloud/learn/natural-language-processing.

[2] http://cs229.stanford.edu/syllabus-spring2020.html

[3] Hastie, T., Tibshirani, R., & Friedman, J. H. (2017). The elements of statistical learning: Data mining, inference, and prediction. New York, NY, USA: Springer.

[4] https://github.com/mojeee/Machine_and_Deap_Learning_ModeA.git