ML Mod A Project : Disaster Tweets Classification

Munesu Maminimini
munesu.maminimini@studenti.unipd.it

Mohammad Huzaifa Fazal
mohammadhuzaifafazal.xxx@studenti.unipd.it

1. Introduction

Given the prevalent use of social media platforms in our daily lives and their integration into multiple human interactions and experiences, they have have become important channels for propagating news as well as information. Whilst recent events have shown that unmitigated use of this can be exploited and lead to the distribution of 'fake news' and misinformation by nefarious parties; it also presents an opportunity to utilise social media as an early warning and risk communication tool particularly during events such as health emergencies, natural disasters, etc. This can potentially lead to better response times in taking appropriate actions where needed, thus preserving human life.
Twitter, which is ranked as the world's 16th largest network, is one such platform. The aim of this project is to develop machine learning models that can correctly classify and accurately predict whether or not a tweet is about a disaster. Using text analysis algorithms which included Logistic Regression (LR), K-Nearest Neighbour (KNN), Random Forest, Support Vector Classification (SVC) and Gradient Boosting Classification (GBC);the ability to classify the tweets correctly was tested, the performance for each model compared and improvements for further experimentation proposed.

2. Dataset

The data-set analysed in this experiment was obtained from the Kaggle challenge: "Natural Language Processing with Disaster Tweets".

2.1. Data Structure and Feature Selection

The were 7613 total data entries. Each data entry had 5 features, namely:

- id - unique tweet identifier
- Keyword - key word from tweet
- Location - location from where tweet was sent
- Text - textual content of tweet
- Target - denotes class of tweet i.e. whether it is about a real disaster (1) or not (0)

In order to determine which features to use, a search of null values indicating missing data point was carried out and the results are displayed in Figure 1.

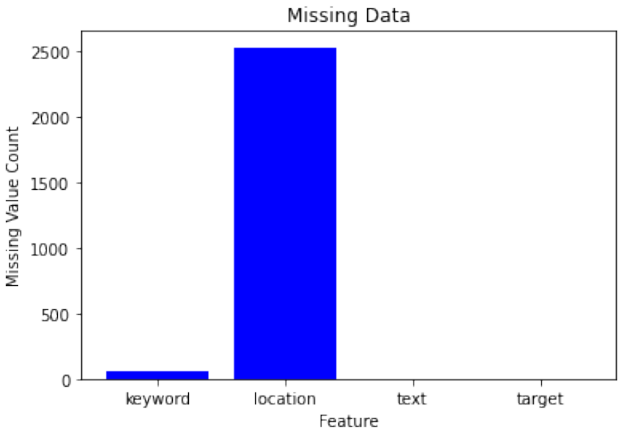

Figure 1. Missing data distribution

It can be noted that the *Location* and *Keyword* features have the most missing data. However since the *Keyword* feature has more values present than not it is possible to drop the data entries for which it is missing. For the analysis carried out in this report, only the *Text* and *Target* features were used as they were present for all entries.

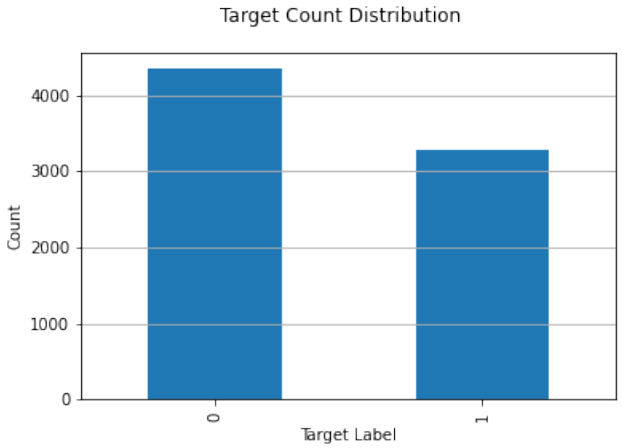

Figure 2. Distribution of tweets labelled '0' or '1'

Furthermore, the data was split between two classes - Class 0 (non-disaster): 4342 and Class 1(disaster): 3271, as illustrated in Figure 2. The data is skewed and slightly unbalanced which could lead to biased learning models. This could be addressed by either dropping some entries of label 0 since it is the majority class or use the F1 score to evaluate the models since it can handle data imbalances. For the experiments carried out in this report, the latter option was selected in addition to the accuracy score as well as the AUC ROC score.

2.2. Data Cleaning and Preprocessing

In order to conduct the experiments, it was necessary to first clean the tweet text. All characters of the text were changed to lowercase and special characters, numbers, URLs and the most common English words (stop words) were removed. The tweets also underwent the following processing techniques:

- Stemming - reduces extra characters from a word to its root or base of a word
- Tokenisation - breaks a phrase into smaller sections
- Vectorisation - maps words to real numbers

3. Method

The data was split into training and tests sets to be used for the different learning models.

3.1. Algorithm Selection

The following models were used in the experiments carried out:

3.1.1 Logistic Regression

Logistic Regression is one of the main types of algorithms used in data classification. Since the task was to classify tweets as "disaster" and "not-disaster", logistic regression was a good starting point. Furthermore, in NLP, it is considered as a "base-line supervised machine learning algorithm for classification" [1].

3.1.2 K-Nearest Neighbour

K-nearest neighbour is a type of non-generalising or instance-based classification learning [2]. A simple majority vote of the k-nearest neighbours is used to classify the different classes based on a similarity measure. In this case, the similarity measure is the proximity/closeness amongst the data points that is calculated by taking the Euclidean distance between the points. This supervised learning algorithm is based on the the notion that neighbours close to each other should share similar characteristics and behaviours - i.e. disaster tweets should have common phrases and word groupings [3].

3.1.3 Random Forest

Random Forest algorithms construct a multitude of decision trees on sub-samples of the dataset and then averages the scores to classify the data - as a disaster or not disaster tweet. It is among the best performing classification algorithms and therefore was an important model to include in this project [4].

3.1.4 Support Vector Classification

Support Vector Classification is based on the SVM approach - which is a classification algorithm which finds a hyperplane in an N-dimensional space that classify the points in the dataset distinctively. The algorithm tries to form the smallest sphere that can enclose the data's image. SVC was included in this analysis since it is known to produce more accurate results with less computation power. [5]

3.1.5 Gradient Boosting Classification

Gradient Boosting Classifiers work by learning iteratively from various weak learning models such as decision trees to produce a strong additive predicting model.

3.2. Performance Measurement

The performance of the models was measured using the following parameters:

- Accuracy score - The ratio of the number of correct predictions over the total number of input
- AUC ROC - AUC measures the degree of separability and ROC is a probability curve. Together they measure the trade off between sensitivity and specificity.
- F1 Score - Harmonic mean of the recall and precision. Computes the balance between recall(sensitivity) and precision.
- Precision - The proportion of positive class over all positives classified by the model (TP/(TP+FP)).
- Recall - The proportion of positive class that got classified correctly (TP/(TP+FN)).

4. Experiments

4.1. Logistic Regression

The grid search yielded C= 0.5 as the best 'lambda coefficient' parameter for LR. The model was able to correctly predict more correct values for '0' **(787/879)** than '1'

(456/644) as illustrated by the confusion matrix in Figure 3. This would suggest that either there exists a better C parameter for which the performance will improve or the initial data imbalance has resulted in a model that has a bias towards classifying 0's and is inexperienced in classifying 1's.

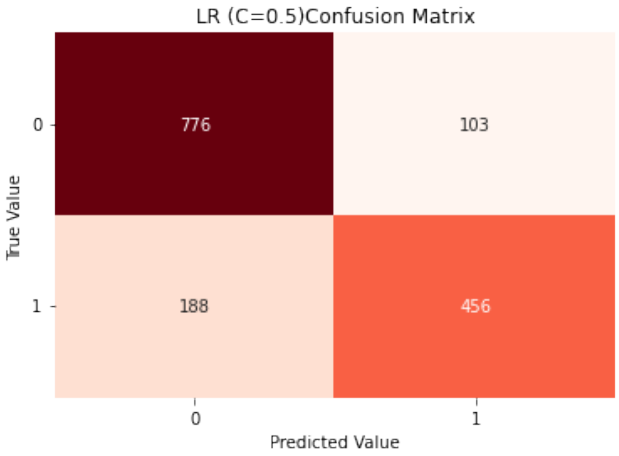

Figure 3. Confusion matrix for LR

4.2. K-Nearest Neighbour

The optimal number of nodes (neighbours) for the KNN algorithm from Figure 4 was determined to be between ~5-10 and so **k=5** was used for the experiment.

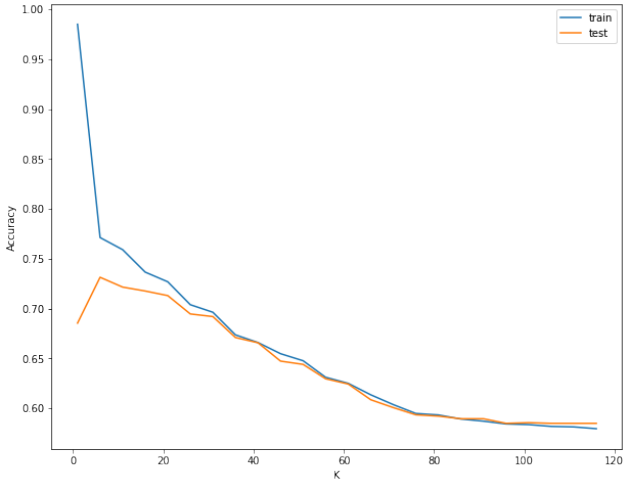

Figure 4. KNN train-test accuracy plot

In Figure 5 the performance of the model for class '0' is better than LR **(c=0.5)** but the performance for class '1' is much less. When K is varied to less or more than 5, the performance continues to worsen. This is due to the nature of the algorithm whereby either too few or too many neighbors results in incorrect classification decisions.

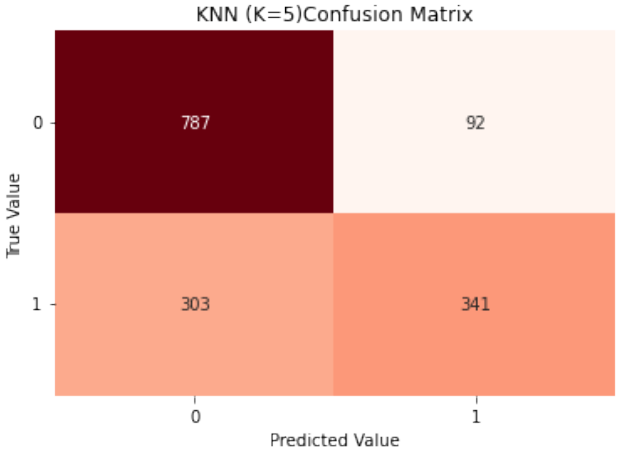

Figure 5. Confusion matrix for KNN (N=5)

4.3. Random Forest

The best performance is at a maximum depth of 64. For the purposes of this experiment, due to computational as well as time limitations no further maximum depth values were considered. The RFC was able to correctly predict **822/879** class '0' and **(340/644)** class '1' tweets. There is a slight improvement in class '0' and decrease in class '1' as shown in Figure 6. This again suggests that either the data imbalance is resulting in a biased model or some of the preprocessing techniques such as vectorisation may have some undesirable effects on the training models.

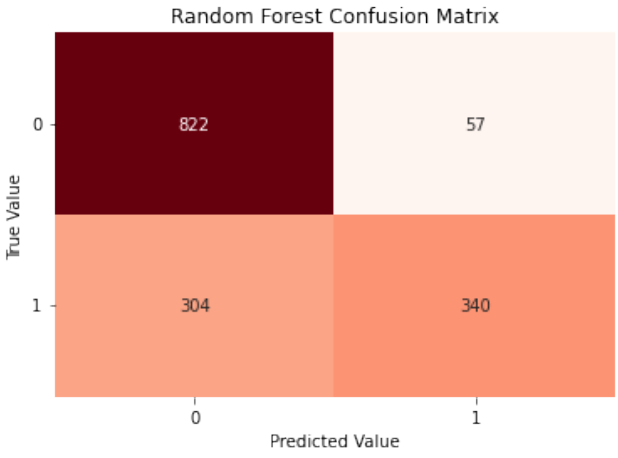

Figure 6. Confusion matrix for RFC

4.4. Support Vector Classification

The SVC model correctly predicts **797/879** class '0' tweets and **426/644** class '1' tweets as illustrated in Figure 7. There is a major improvement for the Class '1' predictions.

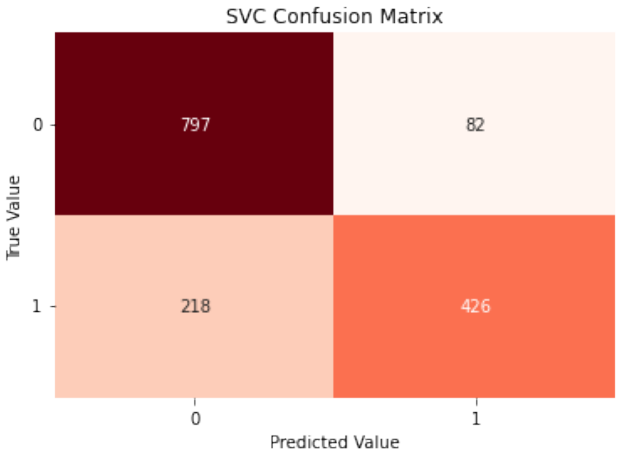

Figure 7. Confusion matrix for SVC

4.5. Gradient Boosting Classification

The performance of the GBC model shown in Figure 8, correctly predicts **782/879** class '0' tweets and **434/644** class '1' tweets. This is the second best prediction for class '1' tweets after LR. It is interesting to note that for all the models tested, an improvement on either class results occurs with a decreased performance in the other one i.e. no model improves on the predicting both classes simultaneously.

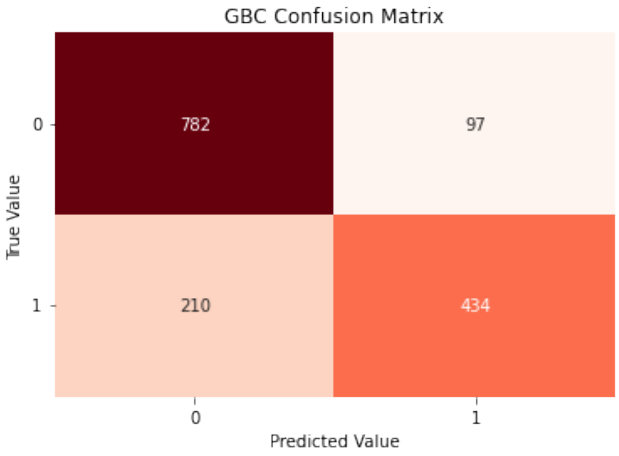

Figure 8. Confusion matrix for GBC

4.6. Summary of Results

The performance of all the five models is summarised in Figure 9 below.

	LR	KNN	RFC	SVC	GBC
Accuracy	0.809	0.741	0.763	0.803	0.798
AUC ROC	0.795	0.858	0.837	0.858	0.853
F1	0.758	0.633	0.633	0.74	0.736

Figure 9. Summary of model performance for Accuracy, AUC ROC and F1

The accuracy and AUC ROC scores are more optimistic about the performance for each model but the F-1 score presents a more realistic depiction. The final best and worst models according to each performance measure are as follows:

Parameter	Best models	Worst Models
Accuracy	LR,SVC,GBC	KNN,RFC
AUC ROC	SVC,KNN,GBC	LR,RFC
F1	LR,SVC,GBC	KNN,RFC

5. Conclusion

From the results above, it can be determined that the SVC, GBC, and LR models have the best overall performance on the test set since they are consistently better for all the performance measurement parameters. The most consistently bad performer is the Random Forest, closely followed by KNN. However, KNN has amongst the highest AUC ROC scores, demonstrating a high sensitivity/recall. This is particularly important to the project since there is a much higher cost of False negatives compared to False positives. The results have also highlighted the importance of having multiple comparison parameters in order to reach a final verdict on model performance since relying on only one can be misleading. It can also be concluded that the data imbalance, even though seemingly insignificant, appears to have lead to the models to having a bias and identifying many false class '0' data; and that preprocessing techniques play an important part in the final result and model performance. For improvements, the data should be balanced, the keyword feature can be included in the training and other parameters such as max depth can be varied to test the effect. Other preprocessing techniques and models can be explored as well as tested over multiple epochs in order to get a fair idea of the best performance.

References

[1] Daniel Jurafsky James H. Martin. Speech and language processing, 2021. Chapter 5 Logistic Regression.

[2] Scikit Learn. Nearest neighbors. https://scikit-learn.org/stable/modules/neighbors.html.

[3] Harshiv Patel. Text classification using k nearest neighbors (knn). https://iq.opengenus.org/text-classification-using-k-nearest-neighbors/.

[4] et al Nitin Hardeniya, Jacob Perkins. Natural langugage processing:python and nltk. https://www.oreilly.com/library/view/natural-language-processing/9781787285101/ch06s04.html.

[5] Asa Ben-Hur. Support vector clustering. http://www.scholarpedia.org/article/Support$_vector_clustering$", $addendum =$ "$(accessed : 28.01.2022)$.