Natural Language Processing: Sentiment Analysis

Introduction

This project is in the Natural Language Processing sphere and engages in the idea of automatic classification of sentiment from recorded tweets coming directly from Twitter, as our aim is to try out different techniques and determine the best classification model for doing that. The goal is to go through the classification models and analyse the results and confusion charts we generate in order to see which model generates the least mistakes in its predictions, thus being the most accurate. All of this is extremely useful in the sphere of information extraction and text analysis, as by using such algorithms, patterns can be discovered, and decision-making is improved. The data can be used to teach an algorithm to distinguish between serious and joke tweets, as well as, be deployed as a regulating mechanism if the sentiment of the tweet is extremely aggressive or abusive, it could automatically flag it and begin the process of removing the message.

Data and Preparation

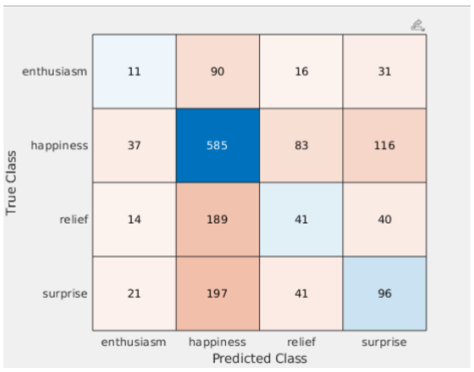
The dataset for this problem must contain a collection of a large number of tweets, which are specifically tweets expressing feelings in some form. Those expressions should be noted as a sentiment along their corresponding tweet, so each one has a feeling attached to the text, it can be either relief, happiness, surprise or enthusiasm. Our text_emotion_data_filtered.csv dataset has exactly that, as it consists of two columns, one being the sentiment, the other the text of the tweet and has exactly 8040 entries in it. Data preparation might be one of the most important things when doing projects like that and especially in our, we've gone through the following steps. Firstly, we import the .csv file into MATLAB with the readtable() function, as it directly transfers our data into a table. We then tokenized the column with the content of the tweets and created a bag of words with it. With the help of the bag of words, we removed all stop words, as well as any words with less than 100 occurrences and build the full Term Frequency-Inverse Document Frequency matrix (tf-idf). After the corresponding label vector from the column of sentiments was completed, the next and final step of this was to do the features and the labels which were going to be used for training and testing the models. I created one feature matrix and one corresponding label vector for training by selecting the first 6432 rows and all the columns, and one feature matrix and corresponding label vector for testing as well, however, it was done with the remaining rows from the dataset, which were 1608 entries. Now that I had those, I was ready to begin the model training and evaluation.

Methodology

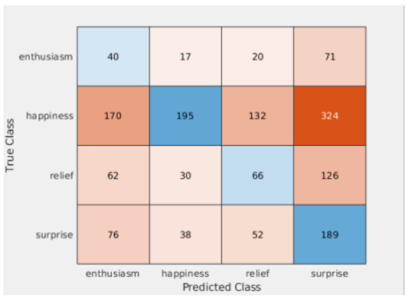
After the data was prepared and the feature and labels were ready for use, the classification models had to be trained and evaluated. For this project, I chose the Decision Trees, Multiclass SVM and Naïve Bayes models to be trained, as I believe they had the capacity to show both sides of the coin and prove there are both advantages and disadvantages of the use of each one of those. All 3 of those models have multiclass support, and categorical predictor support, are relatively fast to medium with their prediction speeds, consume from small to medium memory and have an easy level of interpretability. Decision trees predict responses to data by following the decisions in a tree-like model from the beginning node to the very last leaf node based on a true or false structure. If the next node meets the requirements, we go through it to the other choice presented in front of the algorithm and so on, until the very end node. The Multiclass SVM is different to it, in a way that it finds the best hyperplane that split ups all data points of one class from those of another class. The largest margin or rather a maximal width of the slab parallel to the hyperplane, between the two classes is usually the best hyperplane for a Support Vector Machine (SVM). The Naïve Bayes model applies Bayes theorem with naïve independence assumptions between the features, so it is designated as a tool to be used whenever the predictors are independent of one another within each class. After training the models and getting the correct predictions from each, I was able to calculate the accuracy and see how useful each one of them is. Each of those models gave different results, from which interesting conclusions could be drawn.

Results

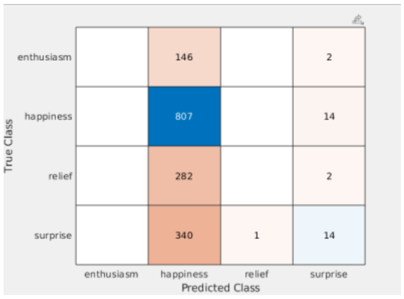
Starting with the first classification model – the Decision Trees, I've successfully calculated that there are 733 correct predictions with this model while maintaining an accuracy of 0.46 or 46%. Attaching the confusion chart, I've made to be able to analyse the results clearly and more easily, we have the most matches for the happiness sentiment while getting the least matches for predicting happiness when the actual class is a surprise:



Going on the Naïve Bayes model, we got only 490 correct predictions with 30% accuracy. It is the lowest of the three models in terms of accuracy and looking at its confusion chart, it can be seen again how most correct predictions go for happiness on happiness, however, the number of mistakes and the high failure ratio on a surprise prediction whenever happiness is the true class is truly detrimental for this model.



The SVM model proves to be the best out of the three in terms of accuracy, as it has the most correct predictions – 821 and the highest percentage of accuracy – 51%. However, looking at its confusion chart, we can see how there are no hits for so many of the categories, no matter whether right or wrong, it doesn't recognise enthusiasm and relief.



Although SVM has the most correct predictions, it fails to look and distinguish between all the sentiments, unlike the Tree and Bayes models. While Bayes on the other hand has the lowest correct predictions, the Decision Tree model gets the best of both worlds, it goes through all the sentiments and has a high accuracy ratio.

Conclusion

Overall, the classification model that I would recommend being used within our project would be the Decision Trees model, as unlike the other two it can identify all the sentiments and successfully go over all their predictions and actual classes, while also maintaining a good accuracy percentage over its predictions. As further research into the project, I would add that other classification models shall be tried the same way, such as K-Nearest Neighbour, Discriminant Analysis and Ensembles and to be seen if any of them would have a better accuracy percentage while going over all the pairs of predictions.