Data Preparation and Classification Report

Introduction

User-generated textual content from social media platforms has allowed researchers to collect valuable data about people's opinions, attitudes, and emotions towards anything. This mini-research takes a look into Natural Language Processing (NLP) in an attempt to automatically classify the perceived sentiment from recorded tweets (from Twitter, an established social media platform). That is done by studying the dataset of tweets, preparing it for machine learning models, and selecting the best textual classification for a given tweet. The machine learning process is supervised. The aim of this project is to automatically, and accurately, determine the sentiment displayed by the tweet. After that, evaluate three distinct classification models upon which Companies, politicians, advertisers, and many others could use this tool to understand better the public's opinions on their products or actions.

Data and preparation

The provided data is a `.csv` file with a simple table of tweets and the categorical sentiment associated with each tweet. In total there are 2 columns and 8040 rows, therefore column one contains the categorical *sentiment*, and the second column contains *Content* tweets as text. The tweets are organized in an alphabetical order, starting at the letter "F".

The four unique sentiments are enthusiasm, happiness, relief, and surprise. All categorical sentiments are found using the following line.

```
unique_labels = unique(all_sentiments);
```

In order for the machine learning model to utilize this information, the data is processed and prepared. First, a Bag-of-Words (BoW) model is prepared for this model. A BoW (also known as a term-frequency counter) records the number of times that a specific word appears in a given document of a collection. However, MATLAB function `bagOfWords()` does not split the text into words. For that, each individual tweet is tokenised using the `tokenizedDocument()` method.

Now that words from each tweet are tokenized and ready to be counted, all English punctuation, common stopWords and infrequent words (words with frequency less than 100 times) can be removed, once again using MATBAL NLP library. This filter allows separating the actual words from other parts of speech that might have little to no significance.

```
erasePunctuation(tokenizedDocument(data)
removeWords(bag,stopWords)
removeInfrequentWords(bag,99)
```

The next necessary step is to build a full TF-IDF matrix for the resulting BoW. The term frequency–inverse document frequency, is a numerical measurement that intends to reflect the importance of a string representation in a given document.

```
M1 = tfidf(bag); full(M1);
```

For most of the classification algorithms to recognise this data, the matrix has to be spread, therefore it is essential to use `full()`.

The final data preparation is to select the label vectors and feature matrix. There are two sets of these, one for training, and one for testing and evaluating. The training set creates a feature matrix and label vectors for training by selecting the first `m` rows of the TF-IDF matrix and all columns. The testing set of feature matrix and label vectors uses the rows after the `m` row.

Methodology

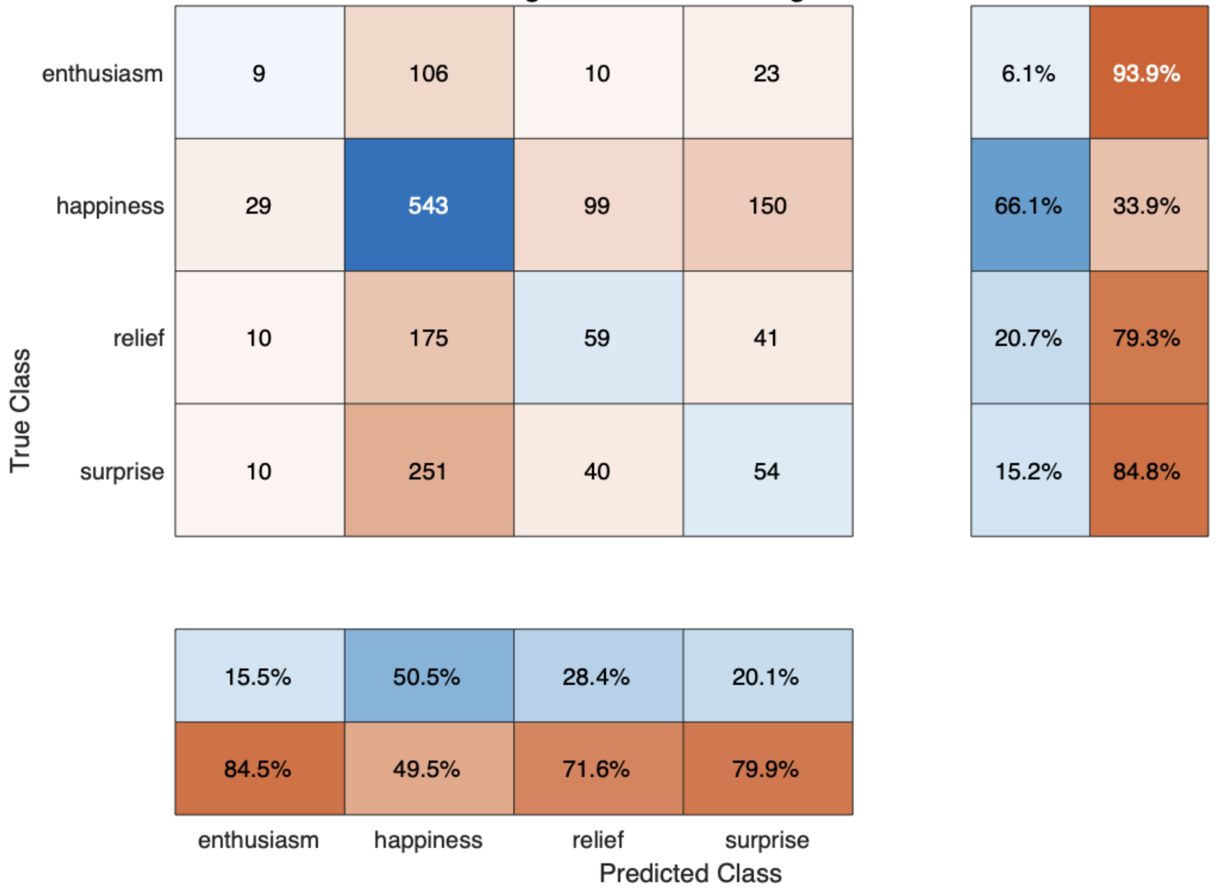
With the data preparation complete, these label vectors and features matrix can be used to classify the sentiment through several Classification Algorithms. Each classification method is fed with the prepared data, which returns a model. The model (which used the training data) then used that data to make predictions.

Results

```
knn_accuracy = 0.4133
discr_accuracy = 0.5140
svn_accuracy = 0.5109
```

Continued on the next page…

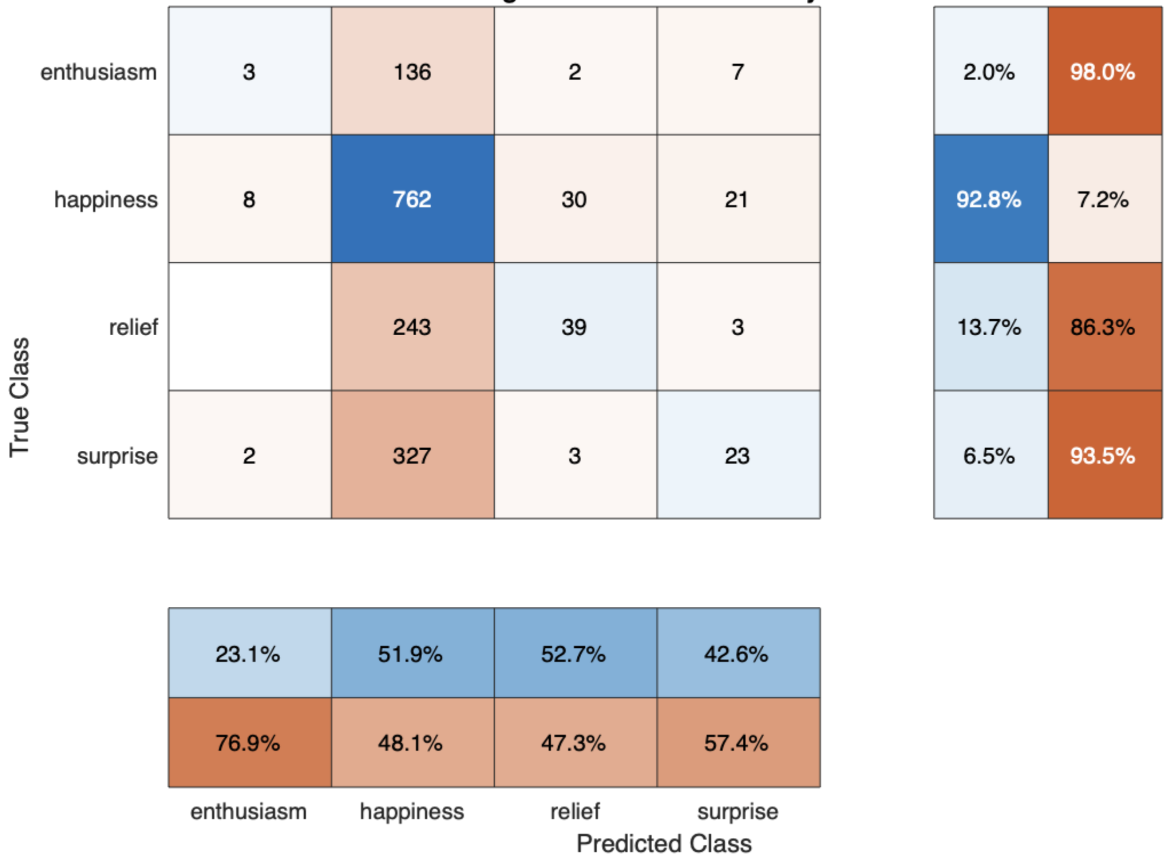
Figure 1. K-Nearest Neighbour



Class	n (truth) ⓘ	n (classified) ⓘ	Accuracy	Precision	Recall	F1 Score
1	58	148	88.32%	0.061	0.16	0.087
2	1075	821	49.66%	0.66	0.51	0.57
3	208	285	76.69%	0.21	0.28	0.24
4	268	355	67.99%	0.15	0.20	0.17

Continued on the next page…

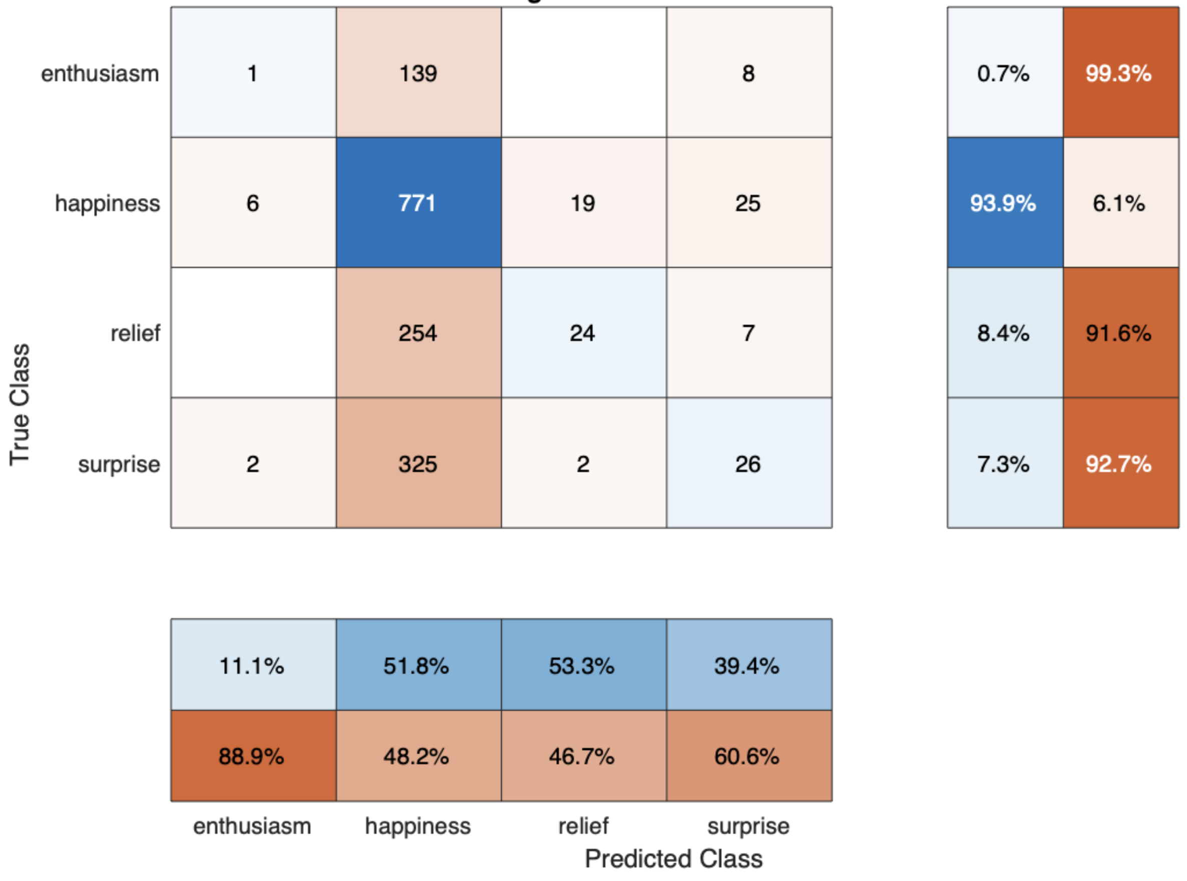
Figure 2. Discriminant Analysis



Class	n (truth) ?	n (classified) ?	Accuracy	Precision	Recall	F1 Score
1	13	148	90.37%	0.020	0.23	0.037
2	1468	821	52.45%	0.93	0.52	0.67
3	74	285	82.54%	0.14	0.53	0.22
4	54	355	77.44%	0.065	0.43	0.11

Continued on the next page…

Figure 3. SVM for Multiclass



Class	n (truth) ⓘ	n (classified) ⓘ	Accuracy	Precision	Recall	F1 Score
1	9	148	90.37%	0.0068	0.11	0.013
2	1489	821	52.27%	0.94	0.52	0.67
3	45	285	82.47%	0.084	0.53	0.15
4	66	355	77.07%	0.073	0.39	0.12