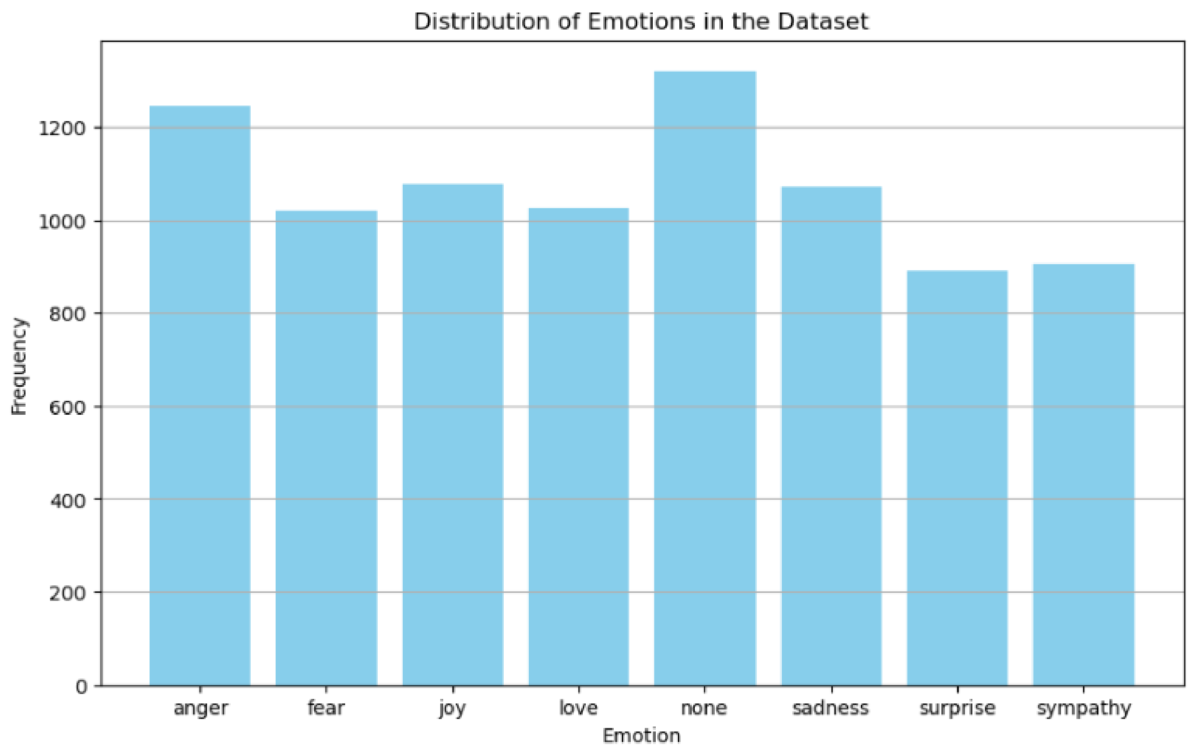
Exploratory data analysis

Yousef tag Eldin

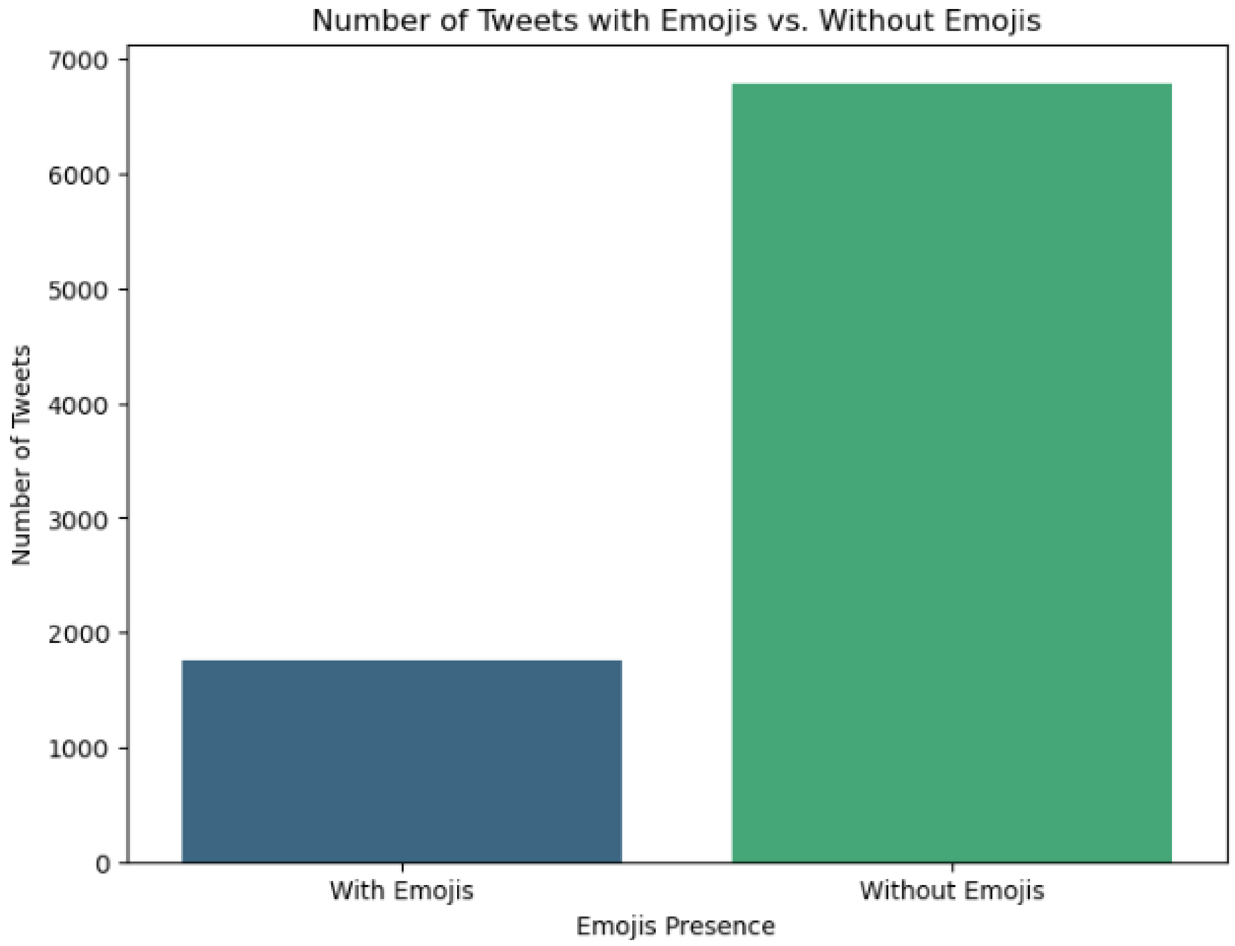
You can view the full analysis and run the plot in the notebook

First,
we look at the distribution of the labels in the data

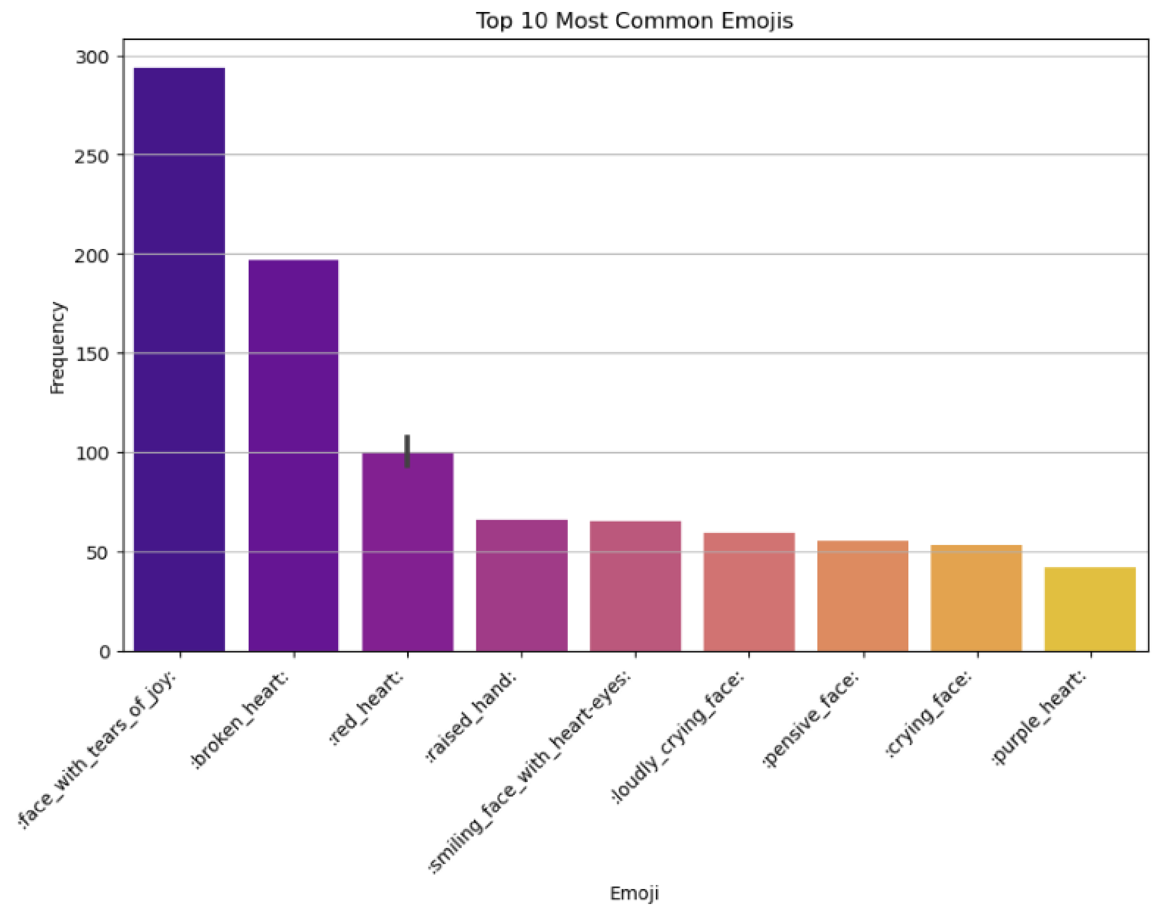


As we can see here the distribution of the labels is somewhat uniform with no big imbalances
We performed some data augmentation and generated 4000 sample 500 of each label that we can use later if needed
Now we will perform analysis on the text of the data first we will analyze the emojis in the text then we will look at the rest of the text

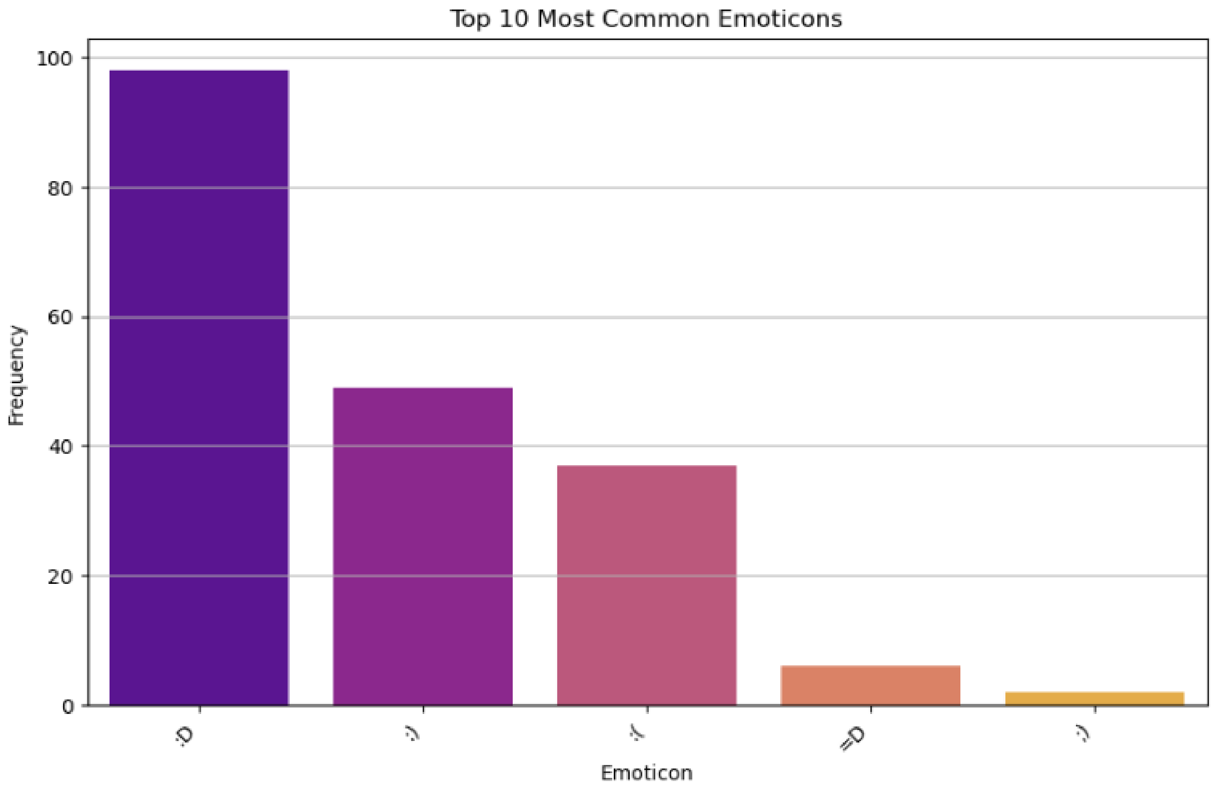
Emojis and emoticons

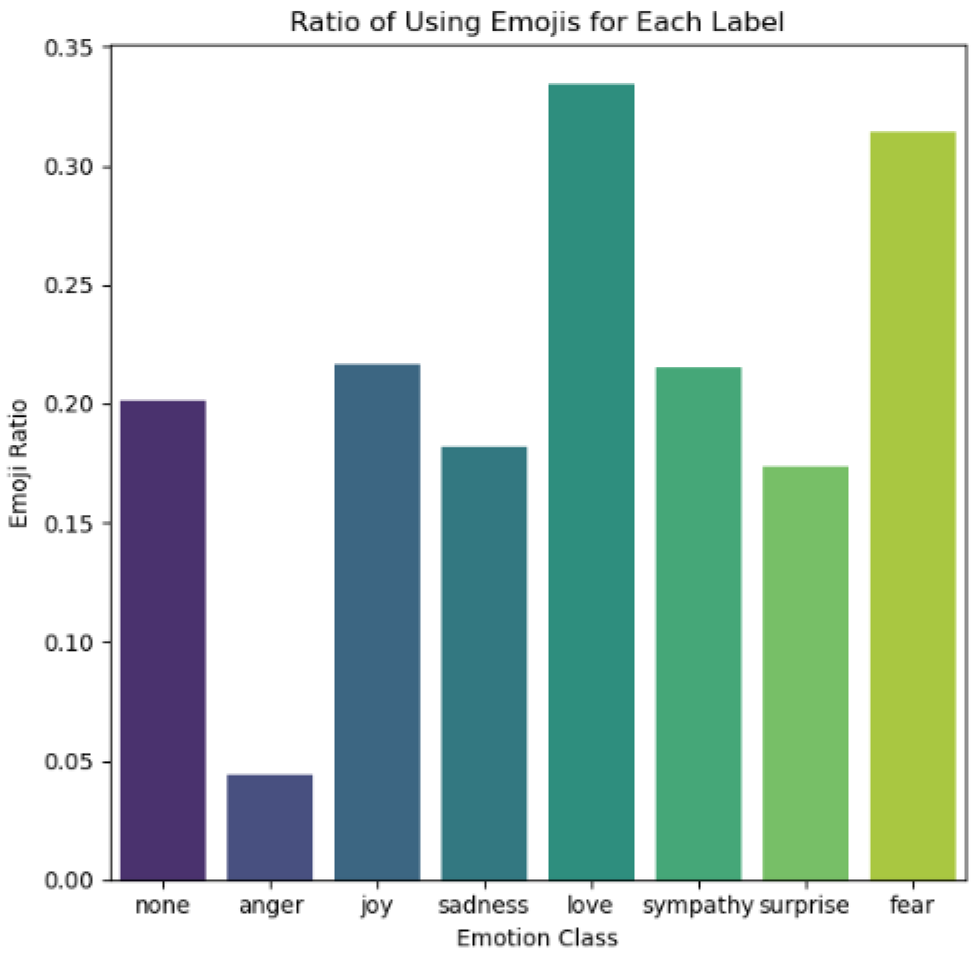


We can see that the majority of the data doesn't have emojis in it


Top 10 Most Common Emojis
Frequency
0
50
100
150
200
250
300
:face_with_tears_of_joy:
:broken_heart:
:red_heart:
:raised_hand:
:smiling_face_with_heart-eyes:
:loudly_crying_face:
:pensive_face:
:crying_face:
:purple_heart:
Emoji

```
{'😔', '❤', '😂', '❤', '✋', '😍', '😭', '💜', '💔', '😢'}
```


Top 10 Most Common Emoticons
Frequency
0
20
40
60
80
100
:D
:)
:(
=D
:)
Emoticon



Now we will look at the ratio of emoji usage in each emotion and we can clearly see that there is a relation between the label and the use of emojis for labels like anger we can see that they rarely contain emojis and for a class like love more than third of the tweets contain emojis so the use of emoji or not is an important indicator for the emotion of the tweet.

In the preprocessing, we remove the emojis and replace them with one of the emotions that we have that suits the emoji

One way to possibly improve the performance of the classifier is to encode whether the tweet had emojis in it or not

<u>Analysis on each label emoji use</u>

```
Class: joy
Most used emojis:
😂: 37
😍: 22
❤: 21
♥: 20
❤: 17
Most used emoticons:
:D: 45
:): 16
;): 1
:(: 1
```

```
Class: none
Most used emojis:
😂: 113
😅: 12
✋: 10
😄: 9
😊: 8
Most used emoticons:
:D: 23
:): 13
=D: 4
```

```
Class: love
Most used emojis:
❤: 55
❤: 51
💜: 32
😍: 32
💕: 28
Most used emoticons:
:): 3
:(: 1
;): 1
```

```
Class: sadness
Most used emojis:
💔: 41
😔: 19
😂: 16
😢: 14
😭: 10
Most used emoticons:
:(: 14
:D: 6
:): 4
=D: 1
```

```
Class: anger
Most used emojis:
✋: 6
😂: 5
[illegible]: 3
🙄: 2
😡: 2
Most used emoticons:
:D: 7
:): 4
:(: 4
=D: 1
```

```
Class: sympathy
Most used emojis:
💔: 51
😢: 20
❤: 18
❤: 12
😔: 12
Most used emoticons:
:(: 5
:): 2
```

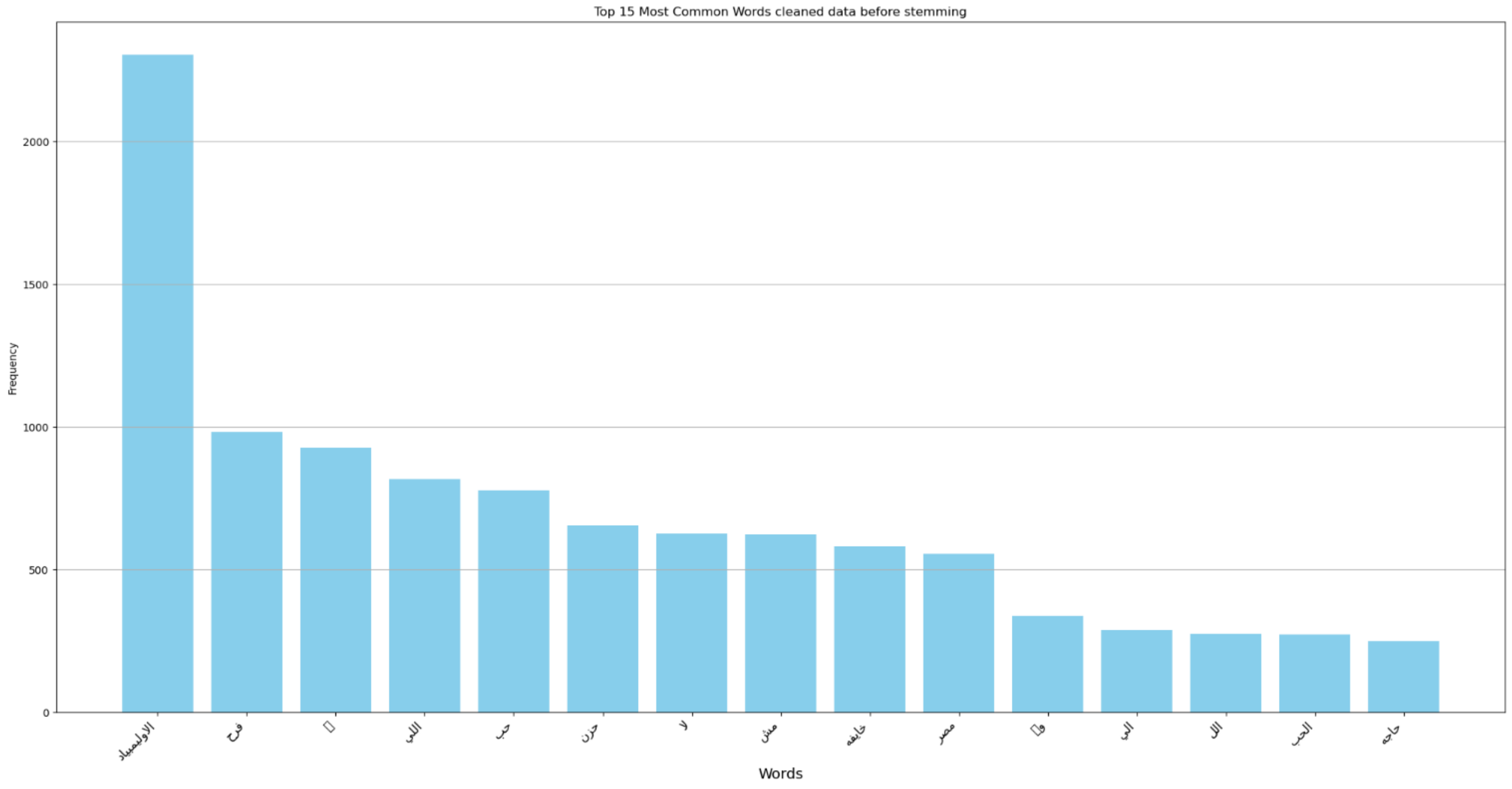
```
Class: surprise
Most used emojis:
😂: 46
🤭: 9
💔: 7
😳: 6
🙄: 6
Most used emoticons:
:D: 16
:(: 3
:): 2
```

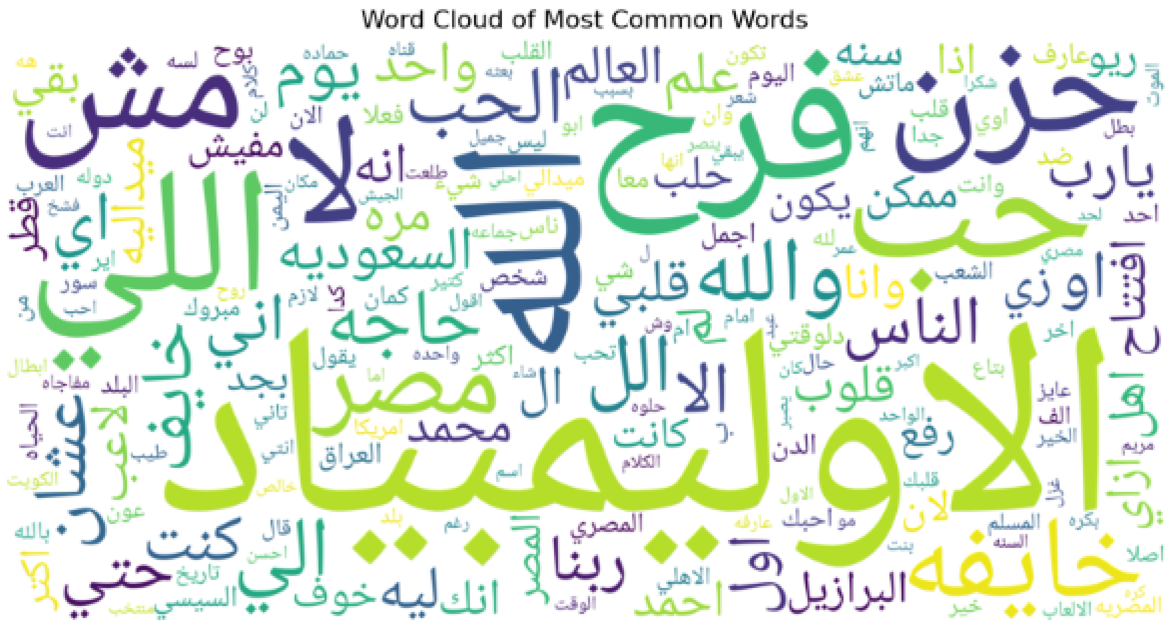
```
Class: fear
Most used emojis:
💔: 79
😂: 67
😭: 39
😔: 21
😢: 15
Most used emoticons:
:(: 9
:): 5
:D: 1
```

anger: has really low emoji use and only one of them is related to anger
None:we can see that this emotion uses a lot of emojis which is not expected for the none class espically the laugh emoji which can confuse the emoji

Also for the (surprise, fear, and sadness), we can see that the laughing face is the most used which is not consistent with the emotion
On the other hand we see that for the love class the emojis represent the emotion accurately

text analysis

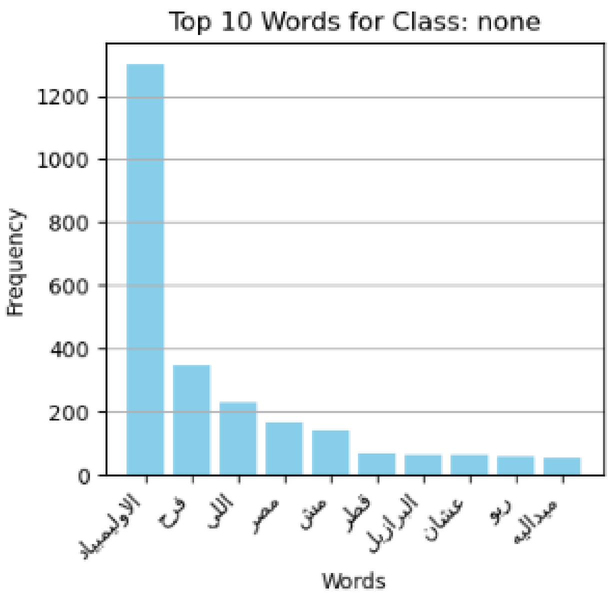


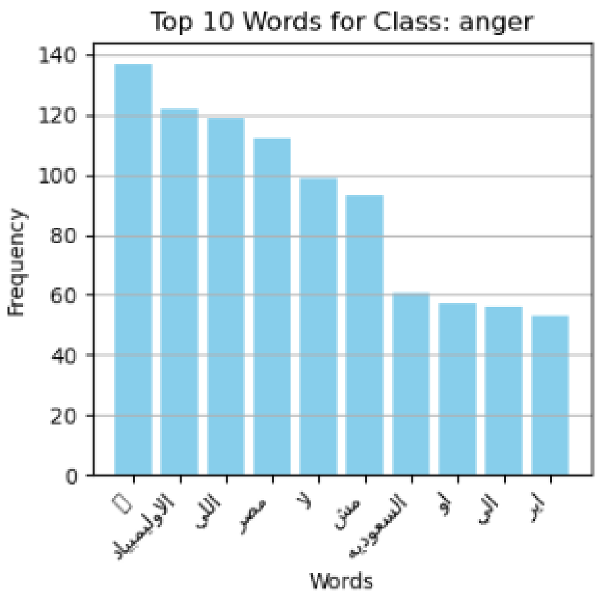


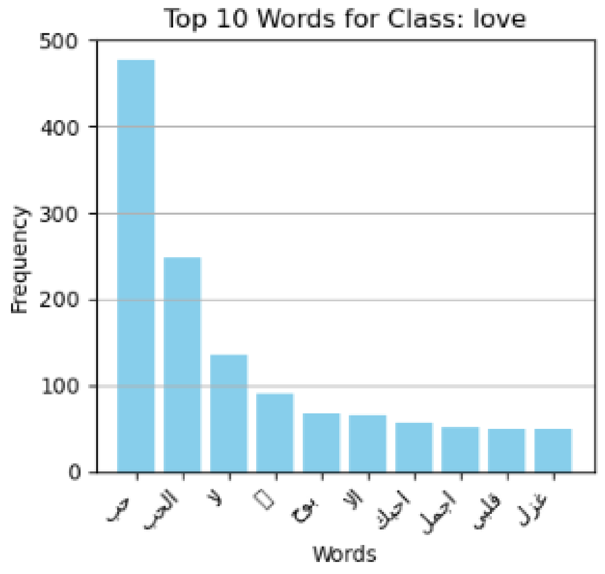
From the plot and the word cloud above we can see that the most common word in the dataset is words like (حب,حزن,فرح,الله,الاوليمبياد)

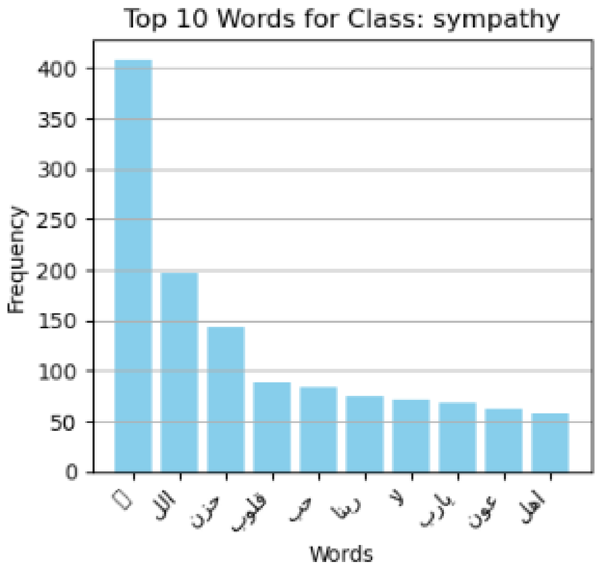
The prevalence of the word olympics is due to the time period when the data was collected as it was collected during the olympics
The words love and sadness are prevalent probably due to the use of emojis as the emojis are replaced with word that represents the emojis like love and sadness

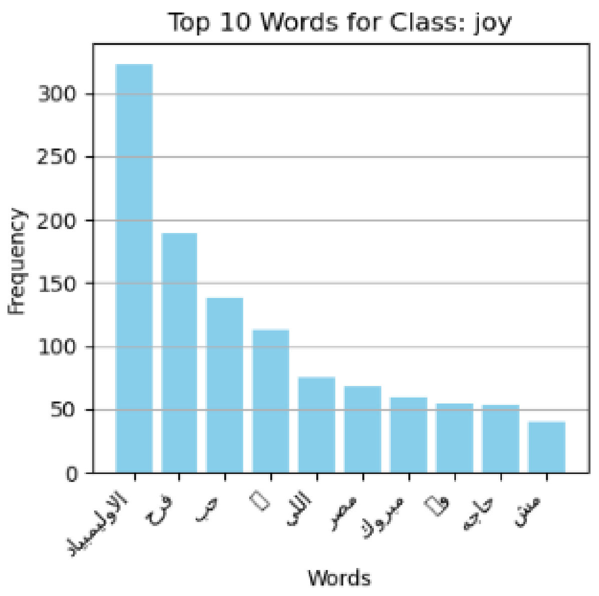
<u>Most common for each class</u>

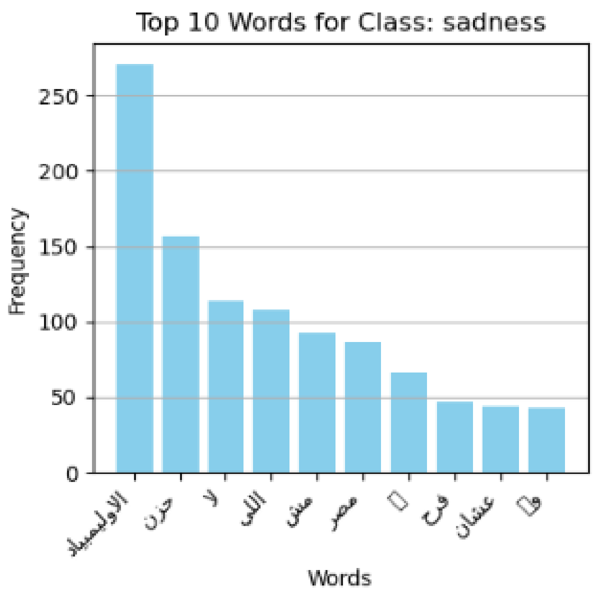


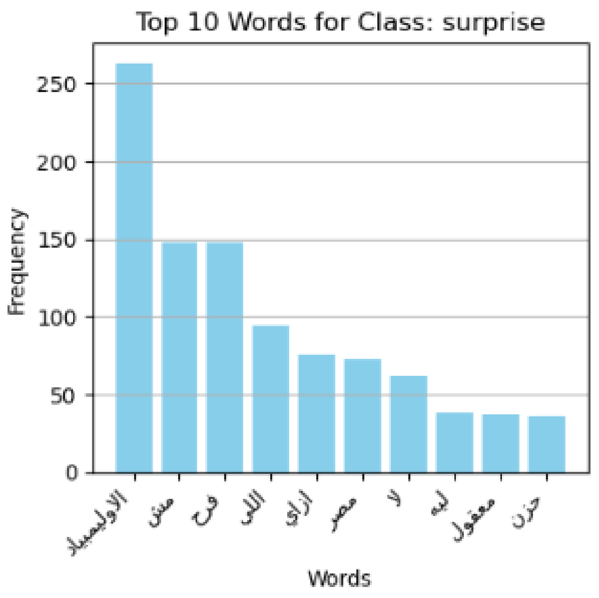


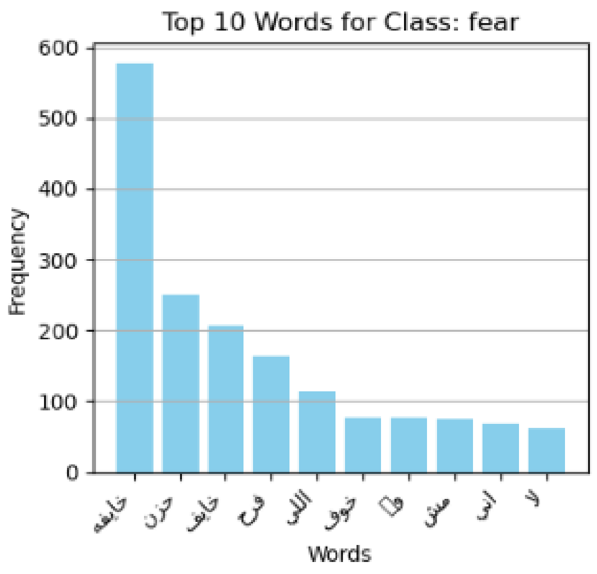












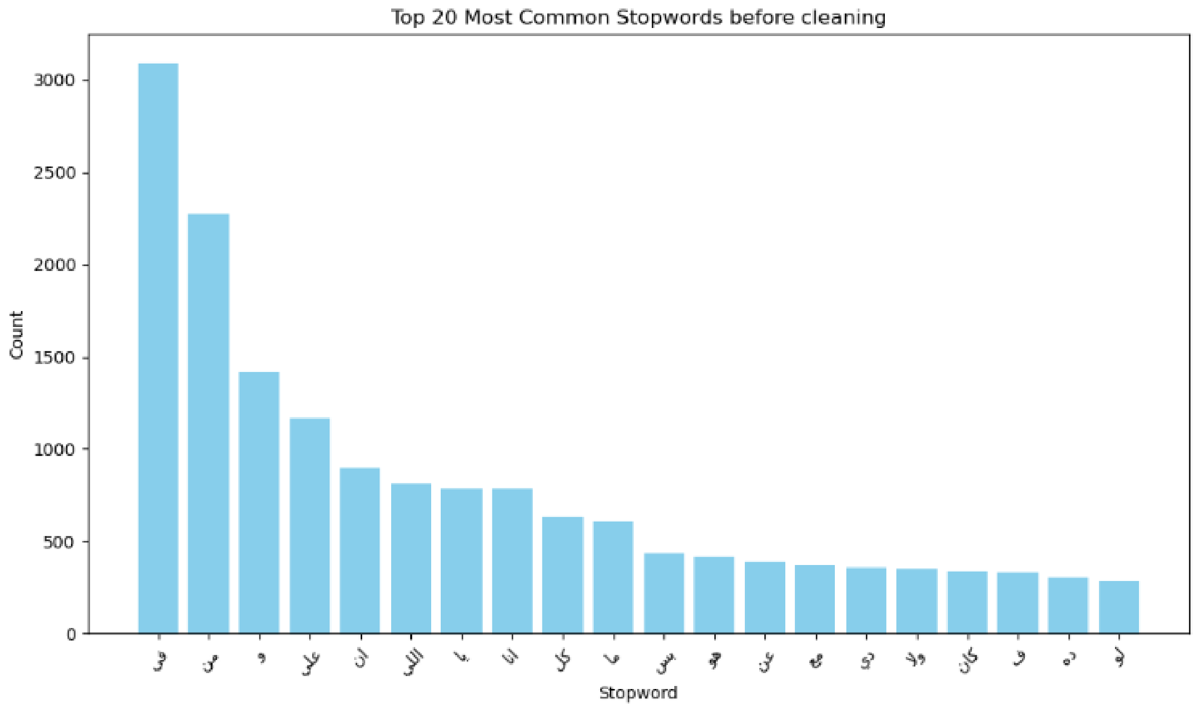
We can see that the word olympics is used alot with classes none,joy,sadness,anger,suprise which is expected because people talk about the event with a wide range of emotions and opinions

Also for classes like (fear,love,sympathy) we can see that the words used represent the emotion

For sadness we can see the word “فرح” used frequently with is counterintuitive

Also we see word like “فرح” used in the class none which is counterintuitive

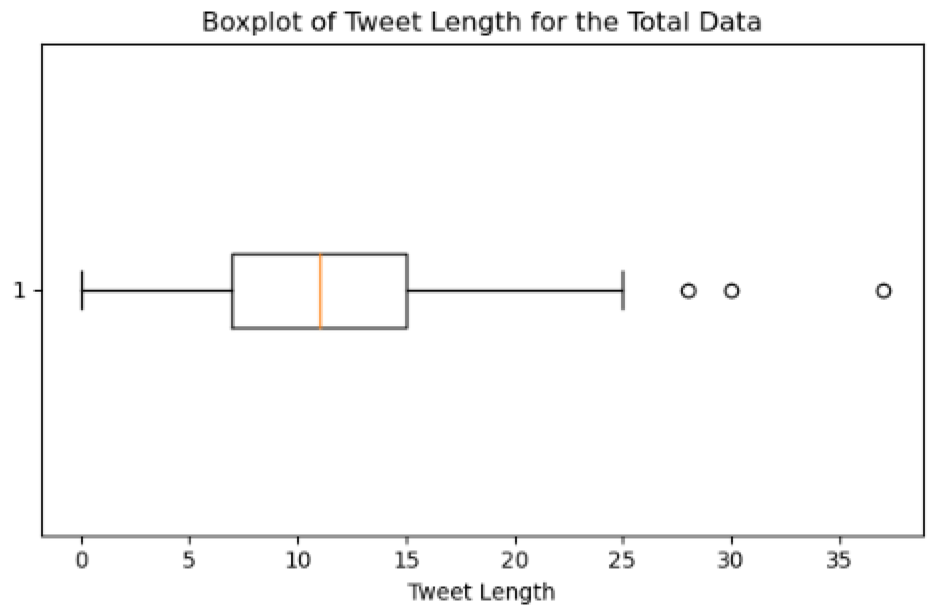
The most common stop words before cleaning

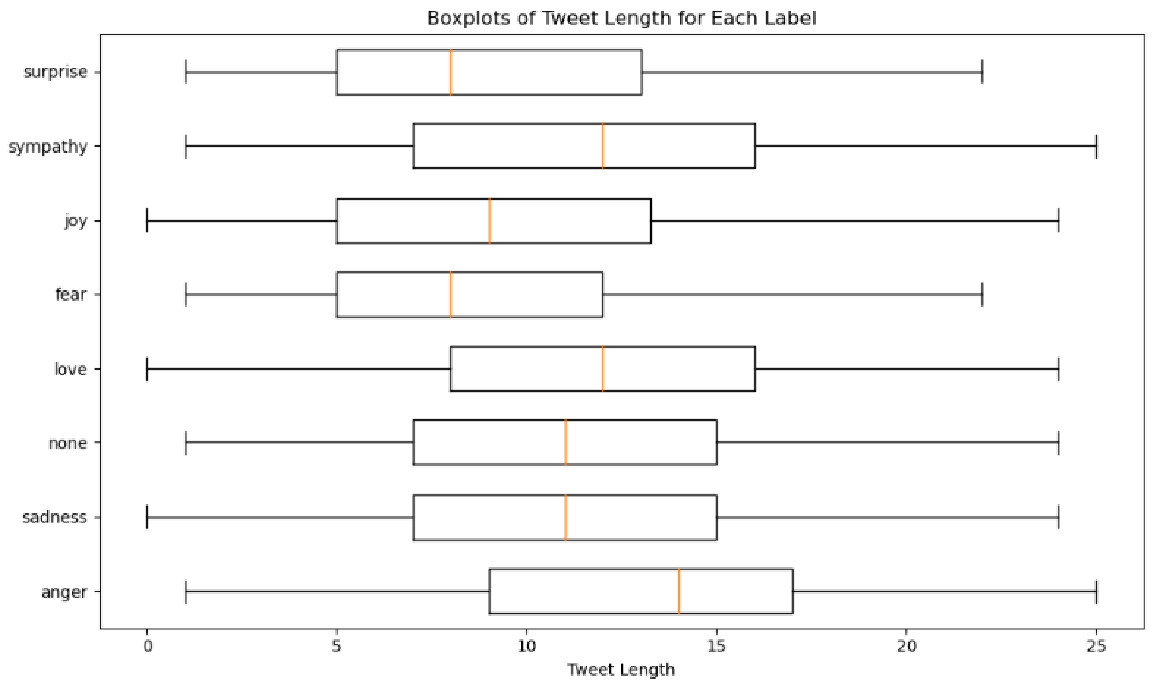


At first we didnt remove all the stop words well enough due to the nature of the tweets being written in a common dialect which resulted in alot of the stop word not being detected due to word being written with an additional "و" is prefixed. Due to this, I manually included the necessary words since I couldn't locate any pre-existing list capable of addressing this issue.
But this resulted in a decrease in performance with gru architecture but also resulted in a boost in performance in the logistic regression classifier but this is still not thorough explored so we will do further analysis in it

Furthermore, we retained negation words as they contribute significantly to sentence meaning and are essential to retain.

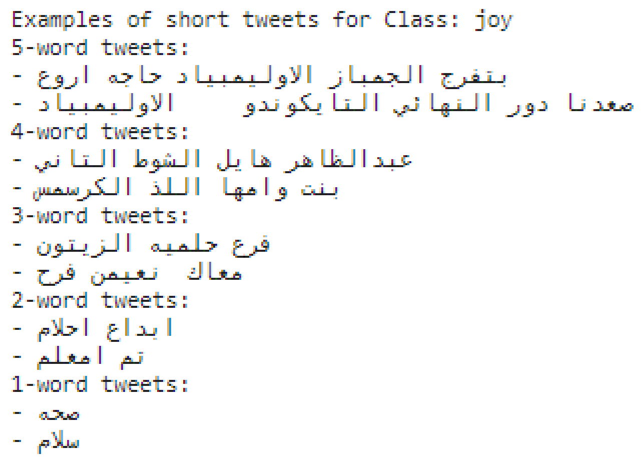
Analysis in the length of the tweets





the average length of the data is 10.81 words with a max of 37 words and a min of 0 words and the rest of the classes some skew to be a bit longer or shorter further analysis is in the notebook

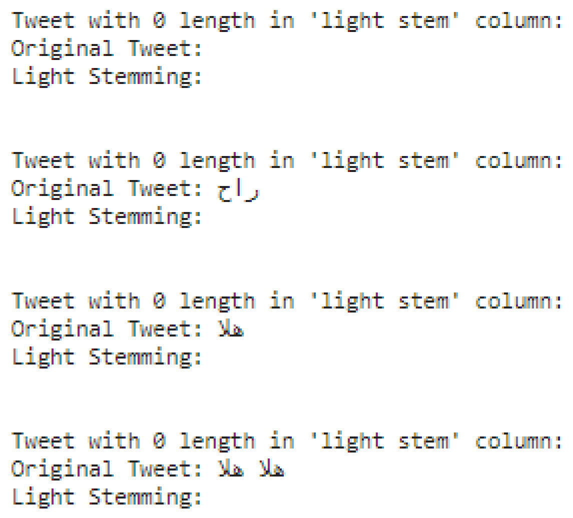
Here are some example of different lengths of the class joy



```
Examples of short tweets for Class: joy
5-word tweets:
- بتفرج الجمباز الاوليمبياد حاجه اروع
- صعدنا دور النهائي التايكوندو    الاوليمبياد
4-word tweets:
- عبدالظاهر هايل الشوط التاني
- بنت وامها اللذ الكرسمس
3-word tweets:
- فرع حلميه الزيتون
- معاك  نعيمن فرح
2-word tweets:
- ابداع احلام
- تم امعلم
1-word tweets:
- صحه
- سلام
```

We can see that the shorter the tweet is the more ambiguous its and harder its to predict the correct class if even we can say that it's possible to predict the correct class from one-word tweet which we can say provides too little information

When we inspect the data length we see that there are 4 sample with 0 length which is weird and as we can see here all of these sample are stop word that got remove or it was already empty before preprocessing



```
Tweet with 0 length in 'light stem' column:
Original Tweet:
Light Stemming:


Tweet with 0 length in 'light stem' column:
Original Tweet: راح
Light Stemming:


Tweet with 0 length in 'light stem' column:
Original Tweet: هلا
Light Stemming:


Tweet with 0 length in 'light stem' column:
Original Tweet: هلا هلا
Light Stemming:
```

The embedding

We employed the Marbert Transformer to generate embeddings. This was achieved by extracting the output from the final layer of the model for each word, subsequently averaging these word embeddings to create a comprehensive tweet embedding.

Upon comparing the performance of these two methods, depending on the model used, we observed a minor performance advantage along with a significant efficiency boost for the sentence embedding approach.a significant efficiency boost for the sentence embedding approach.
We also tried to remove the special tokens the tokenizer add like the [UNK] [CLS]
But removing these tokens resulted in a significantly worse performance meaning that these tokens help the model make a better embedding

NOTES

I also did a small experiment by replacing the word olympics by one of four word ["", "رياضة", "مسابقة", "منافسة", "لعبة"] randomly and this lead to a small boost in performance
Which only validated the big effect the word olympics have on the data
This can be explored further later on in the project