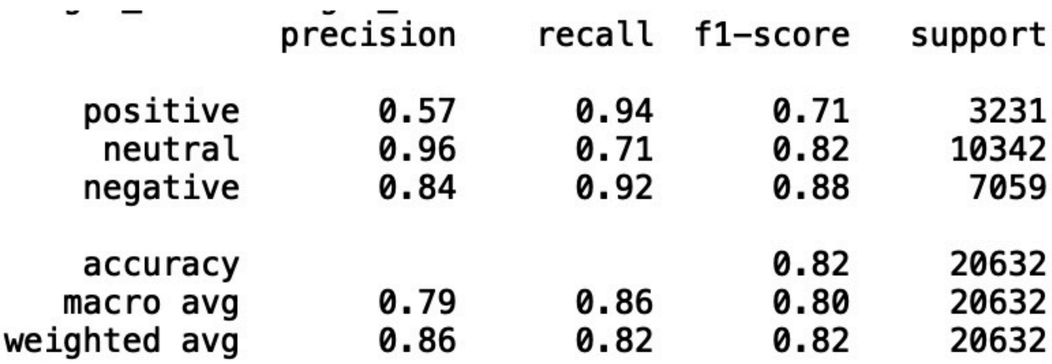
1. **Create a results table that compares performance of the algorithms you chose for analysis.**

Naive Bayes

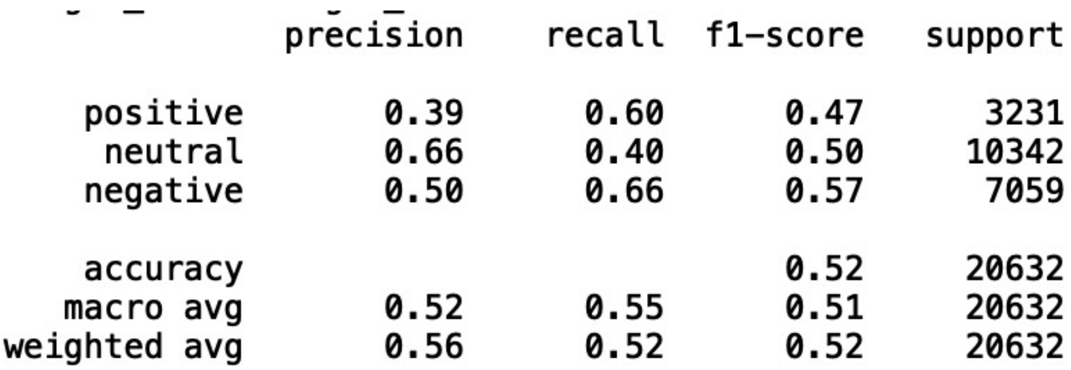


```
              precision    recall  f1-score   support

    positive       0.57      0.94      0.71      3231
     neutral       0.96      0.71      0.82     10342
    negative       0.84      0.92      0.88      7059

    accuracy                           0.82     20632
   macro avg       0.79      0.86      0.80     20632
weighted avg       0.86      0.82      0.82     20632
```

Vader



```
              precision    recall  f1-score   support

    positive       0.39      0.60      0.47      3231
     neutral       0.66      0.40      0.50     10342
    negative       0.50      0.66      0.57      7059

    accuracy                           0.52     20632
   macro avg       0.52      0.55      0.51     20632
weighted avg       0.56      0.52      0.52     20632
```

2. **Speculate on the differences between the two performance measures above.**

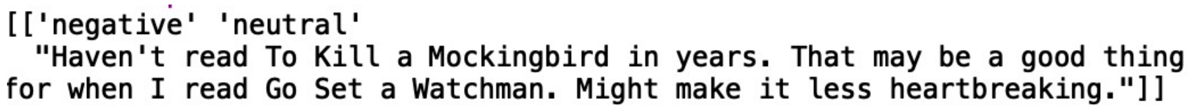
From the definition, Recall is the ratio of correctly predicted positive observations to the all observations in actual class. And Precision is the ratio of correctly predicted positive observations to the total predicted positive observations.

F1 Score = 2*(Recall * Precision) / (Recall + Precision)

From the formula of F1-measure, we can see that it’s the is the weighted average of Precision and Recall. This score takes both false positives and false negatives into account.

In contrast, the average recall only concerns the recall.

3. **Look at your results and find / show examples where your classifiers have mis-performed. What sorts of phenomena do you see and speculate on why you see these errors. Are there distinct differences between classifiers or are differences difficult to see from your results?**



```
[['negative' 'neutral'
  "Haven't read To Kill a Mockingbird in years. That may be a good thing
for when I read Go Set a Watchman. Might make it less heartbreaking."]]
```

Here is the example that Naive Bayes model regards this piece of twitter as the negative tone but the original table for it is neutral. Maybe there are too many words that seems negative words such as “haven’t,” “kill,” “less,” “heartbreaking.” But in fact, “kill a mockingbird” is the name of a book, it’s neutral. And the tone of the whole sentence is not negative at all. Some words in the name of the books or the name of the songs may be positive or negative, but actually they are neutral. This kind of situation may lead to errors. In the result table, it seems that Naive Bayes is good at recognizing neutral and negative sentiments, Vader is fine with neutral and negative emotions. But it still not distinct differences from our results.

4. **How important was tokenization / feature extraction?**

Tokenization is breaking the texts such as sentences, paragraphs, and articles into the minimal meaningful units: words. Tokenizing means splitting your text into minimum meaningful units. And also, tokenize can make the whole text easier and make analyzing more accuracy. We need to combine all the words with the same meaning into one token (such as the same words with different capitalization or different tense) and remove all words without any meaning. For example, we can remove the URL and ‘unknown’ in the INPUT.txt, because they won’t help us to analyze the sentiment and the public opinion in the text. In contrast, they may disturb some analyze process, such as catching the most frequency words. And also, they may weaken the accuracy of sentiment analysis.

5. **If you had more time, what might you do differently? What questions do you know have about your analysis that you didn't have before starting?**

If we have more time, we will try more classifiers and compare them. And also, we want to try those classifiers in the different twitter txt file to see if the difference between them are biased or not. We have some questions about how the different ways to do tokenization influence the result of the different parts of the project.