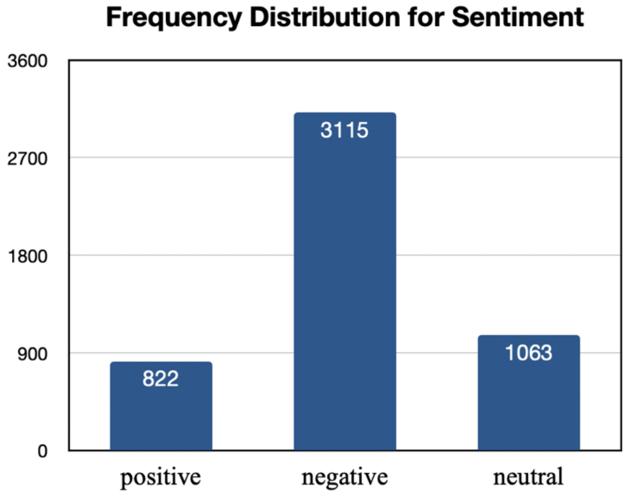
1.



In the sentiment class of 5000 tweets, “negative” is 3115, which is greatly larger than “positive”(822) and “neutral”(1063). It can be noticed that it is unbalanced dataset because ‘negative’ accounts for the majority part of 5000 tweets.

2.

BNB(whole)	precision	recall	f1-score	support
negative	0.67	0.99	0.80	628
neutral	0.78	0.19	0.30	210
positive	0.88	0.09	0.17	162
accuracy			0.68	1000
macro avg	0.78	0.42	0.42	1000
weighted avg	0.73	0.68	0.59	1000

BNB(1000)	precision	recall	f1-score	support
negative	0.87	0.83	0.85	628
neutral	0.60	0.67	0.63	210
positive	0.62	0.65	0.63	162
accuracy			0.77	1000
macro avg	0.70	0.71	0.70	1000
weighted avg	0.77	0.77	0.77	1000

MNB(whole)	precision	recall	f1-score	support
negative	0.72	0.99	0.83	628
neutral	0.81	0.25	0.38	210
positive	0.83	0.38	0.52	162
accuracy			0.73	1000
macro avg	0.79	0.54	0.58	1000
weighted avg	0.76	0.73	0.69	1000

MNB(1000)	precision	recall	f1-score	support
negative	0.84	0.89	0.86	628
neutral	0.63	0.54	0.58	210
positive	0.67	0.62	0.65	162
accuracy			0.77	1000
macro avg	0.71	0.69	0.70	1000
weighted avg	0.77	0.77	0.77	1000

The table below shows the metrics extracted from the tables above, which is related to BNB models using two different vocabulary.

BNB	accuracy	precision	recall	F1-score
micro(whole)	0.68	0.68	0.68	0.68
macro(whole)	0.68	0.78	0.42	0.42
micro(1000)	0.77	0.77	0.77	0.77
macro(1000)	0.77	0.70	0.71	0.70

The table below shows the metrics extracted from the tables above, which is related to MNB models using two different vocabulary.

MNB	accuracy	precision	recall	F1-score
micro(whole)	0.73	0.73	0.73	0.73
macro(whole)	0.73	0.79	0.54	0.58
micro(1000)	0.77	0.77	0.77	0.77
macro(1000)	0.77	0.61	0.69	0.70

From the tables above, it can be seen in both BNB and MNB models that the metrics (micro- and macro-accuracy, precision, recall and F1) except MNB macro-precision from the training set (b) the most frequent 1000 words from the vocabulary have slight increase than (a) the whole vocabulary.

Accuracy = $\frac{\mathrm{TP+TN}}{\mathrm{TP+TN+FP+FN}}$. Micro-accuracy = micro-precision= micro-recall= micro-F1=macro-accuracy = accuracy. Accuracy increasing after using (a) instead of (b) indicates that the proportion of number of correct predictions in total number of predictions increases.

Precision = $\frac{\mathrm{TP}}{\mathrm{TP+FP}}$. Macro-precision decreasing means that average of proportion of positive predictions which are actually correct decreases after using (b) instead of (a).

Recall = $\frac{\mathrm{TP}}{\mathrm{TP+FN}}$. Macro-recall increasing means increasing average of proportion of actual positives which are predicted correctly after using (b) instead of (a).

F1-score = $2*\frac{\text{precision*recall}}{\text{precision+recall}}$. Macro-F1 will give the same importance to each class. The increase indicates that BNB and MNB models perform better after using (b) than (a).

3. The table below shows all metrics on the test set of the three standard models with respect to the VADER baseline. Because micro-accuracy = micro-precision= micro-recall= micro-F1=macro-accuracy, "accuracy" will be used to represent micro-accuracy, micro-precision, micro-recall, micro-F1 and macro-accuracy in the table below.

model	accuracy	macro-precision	macro-recall	macro-F1
DT	0.70	0.62	0.54	0.56
BNB	0.68	0.78	0.42	0.42
MNB	0.73	0.79	0.54	0.58
VADER	0.54	0.54	0.60	0.51

VADER has the lowest accuracy among the four models because crowed-sourcing is in general highly unreliable and the dataset might not include much use of emojis and other markers of sentiment (which are mentioned in the assignment spec). At the same time, VADER has the highest macro-recall, which means that it has the highest average of proportion of actual positives which are predicted correctly among the four models.

4. The table below shows all metrics with and without preprocessing by applying NLTK English stop word removal then NLTK Porter stemming on the three standard models. Because micro-accuracy = micro-precision= micro-recall= micro-F1=macro-accuracy, "accuracy" will be used to represent micro-accuracy, micro-precision, micro-recall, micro-F1 and macro-accuracy in the table below.

model(preprocessing)	accuracy	macro-precision	macro-recall	macro-F1
DT (without)	0.70	0.62	0.54	0.56
DT (with)	0.70	0.64	0.59	0.61
BNB (without)	0.68	0.78	0.42	0.42
BNB (with)	0.69	0.78	0.46	0.47
MNB (without)	0.73	0.79	0.54	0.58
MNB (with)	0.75	0.76	0.58	0.62

A stop word is a commonly used word (such as "the", "a", "an", "in") that a search engine has been programmed to ignore on classifier performance. Stemming is the process of producing morphological variants of a root/base word. For example, stemming for root word "like" includes "likes", "liked", "likely", "liking".

After applying preprocessing of NLTK English stop word removal then NLTK Porter stemming, the accuracy of BNB and MNB has slight increase. Meanwhile, the macro-precision, macro-recall and macro-F1 of DT all have obvious increase after preprocessing above. Hence, the preprocessing of NLTK English stop word removal then NLTK Porter stemming may prompt the three standard models to learn better than before. However, it is unclear in the table above that which one of stopwords and stemming works in the "accuracy" increase, or that both of them work. Further tests are needed before confirming my best method.

5. The table below shows all metrics of three standard models before and after converting all letters to lower case. Because micro-accuracy = micro-precision= micro-recall= micro-F1=macro-accuracy, "accuracy" will be used to represent micro-accuracy, micro-precision, micro-recall, micro-F1 and macro-accuracy in the table below.

model(lowercase)	accuracy	macro-precision	macro-recall	macro-F1
DT (false)	0.70	0.62	0.54	0.56
DT (true)	0.71	0.64	0.58	0.59
BNB (false)	0.68	0.78	0.42	0.42
BNB (true)	0.72	0.83	0.50	0.54
MNB (false)	0.73	0.79	0.54	0.58
MNB (true)	0.76	0.81	0.59	0.64

From the table above, it is shown that all metrics on the training set increased slightly (such as the accuracy of three standard models) or greatly (such as macro-F1 of BNB) . This means that converting all letters to lowercase prompts the three standard models to learn better than before. Converting all letters to lowercase can be added to my best method.

6. From Q3, it is obvious that MNB model has the highest accuracy among the three standard models and VADER baseline. Therefore, I will choose MNB model in my method for sentiment.

From Q5, it shows that converting all letters to lowercase prompts MNB to learn better from the dataset. Therefore, converting all letters to lowercase would be added in my method.

From Q2, develop MNB model from the training set using the most frequent 1000 words from the vocabulary would bring about slight increase in metrics except macro-precision. Hence, I tried to change the max_features.

From Q4, applying NLTK English stop word removal then NLTK Porter stemming may have effects on giving rise to metrics.

The table below shows metrics of MNB model with different preprocessing and max_features. Because micro-accuracy = micro-precision= micro-recall= micro-F1=macro-accuracy, “accuracy” will be used to represent micro-accuracy, micro-precision, micro-recall, micro-F1 and macro-accuracy in the table below.

MNB with different circumstance (lowercase = true)	accuracy	macro-precision	macro-recall	macro-F1
1000	0.77	0.61	0.69	0.70
1000, stop word removal	0.77	0.71	0.69	0.70
1000, stemming	0.79	0.74	0.72	0.73
1000, stop word removal, stemming	0.77	0.71	0.69	0.70
2000	**0.81**	**0.77**	**0.72**	**0.74**
2000, stop word removal	0.78	0.72	0.69	0.70
2000, stemming	0.80	0.76	0.71	0.73
2000, stop word removal, stemming	0.78	0.73	0.69	0.70

From the table above, it is clear that MNB model with converting all letters to lowercase and setting max_features 2000 has the highest metrics between all Homogeneous values. **Hence, my best method for sentiment analysis is MNB model with converting all letters to lowercase, setting max_features 2000.**

The table below shows some experimental results for my method trained on the training set of 4000 tweets and tested on the test set of 1000 tweets.

some experimental results for my method		
test_id	predicted result(predicted_y)	actual result(y_test)
4001	negative	negative
4002	negative	negative
4003	negative	negative
4004	neutral	positive
4005	negative	negative
4006	negative	negative
4007	neutral	negative
4008	positive	positive
4009	negative	negative
4010	negative	neutral
4011	negative	negative
4012	negative	negative
4013	negative	negative
4014	negative	negative
4015	negative	negative

The table shows the comparison of my model to the standard models and the baseline. Because micro-accuracy = micro-precision= micro-recall= micro-F1=macro-accuracy, "accuracy" will be used to represent micro-accuracy, micro-precision, micro-recall, micro-F1 and macro-accuracy in the table below.

model	accuracy	macro-precision	macro-recall	macro-F1
DT	0.70	0.62	0.54	0.56
BNB	0.68	0.78	0.42	0.42
MNB	0.73	0.79	0.54	0.58
VADER	0.54	0.54	0.60	0.51
my model	**0.81**	**0.77**	**0.72**	**0.74**