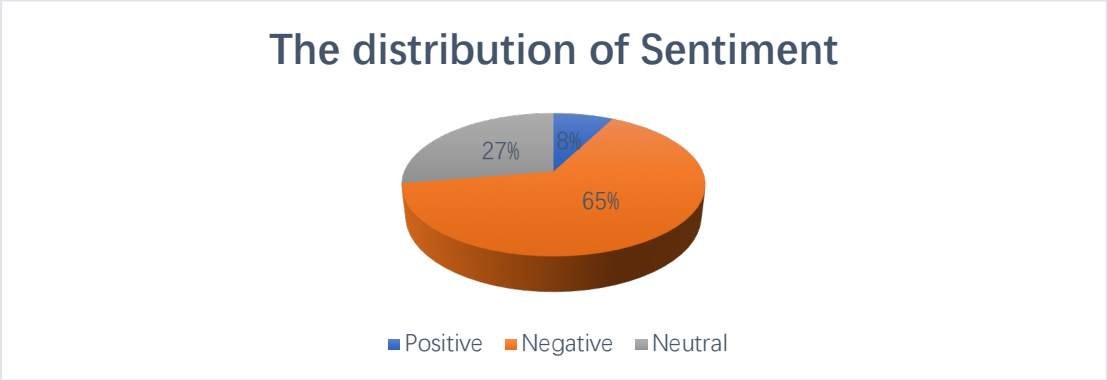
Student ID: Z5189310
Student Name: Yuchen Yang
1) Question 1
Sentiment Chart

Sentiment	Twitter numbers
Positive	153
Negative	1294
Neutral	553

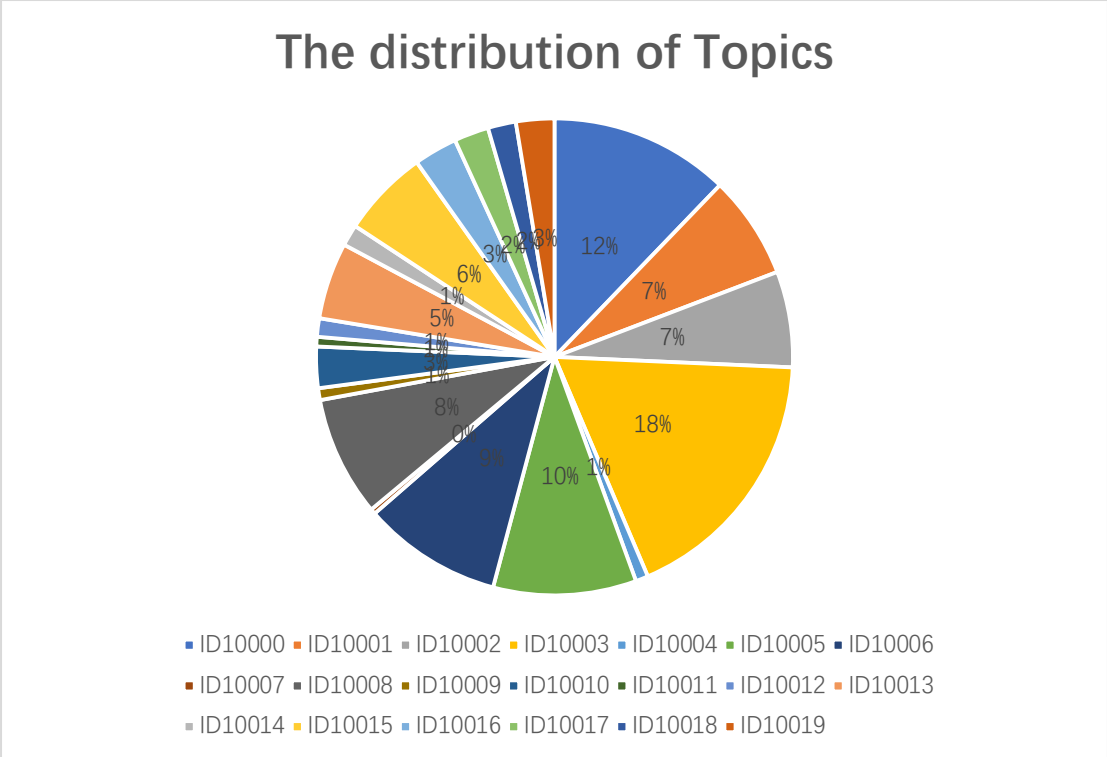


As for the sentiment chart, we can easily see that most of the data we get are negative, accounting for 65%, while neutral ranks second, accounting for 27% roughly, while positive only accounts for 8%. Therefore, for our model, in extreme cases, there may be a lack of positive examples, which will directly affect the establishment of our model and the next sentiment prediction. If this data model is used for prediction, its next prediction will tend to be negative in the largest proportion.

Topics Chart

Topic ID	Twitter numbers
10000	244
10001	140
10002	130
10003	358
10004	17
10005	194
10006	189
10007	7
10008	163
10009	16
10010	56
10011	13
10012	25
10013	104
10014	29
10015	119

10016	59
10017	47
10018	38
10019	52



For topic this chart, the number of some topics in the model accounts for less than the average (number of total amount divided by the sum of the topic data), such as the topic with ID 10007 topic in only seven of the total data, its proportion in the whole model is 0.35%, almost can be ignored, then the forecast for the next data, it is hard to predict, because its proportion is too small in the model.

Therefore, we should increase the amount of their data to reach the average amount, to ensure the integrity, feasibility and high accuracy of our model.

2) Question 2

Model_Name	max_features	accuracy_score	precision_score (macro)	recall_score (macro)	f1_score(average='macro')	f1_score(average='micro')
DT_sentiment	100	0.658	0.476	0.426	0.658	0.431
	200	0.658	0.476	0.426	0.658	0.431
	400	0.658	0.476	0.426	0.658	0.431
	800	0.658	0.476	0.426	0.658	0.431
	1000	0.658	0.476	0.426	0.658	0.431
BNB_sentiment	100	0.712	0.588	0.521	0.712	0.541
	200	0.722	0.64	0.544	0.722	0.57
	400	0.726	0.631	0.534	0.726	0.558
	800	0.728	0.63	0.521	0.728	0.538
	1000	0.728	0.73	0.525	0.728	0.544
MNB_sentiment	100	0.708	0.589	0.486	0.708	0.511
	200	0.736	0.646	0.555	0.736	0.584
	400	0.732	0.645	0.566	0.732	0.594
	800	0.736	0.635	0.562	0.736	0.588
	1000	0.738	0.654	0.558	0.738	0.583

As for the sentiment data, we can find according to the above data table that changing the value of feature will not affect the accuracy of DT model or other measurement results.

However, for BNB and MNB models, the effect is significant, especially for MNB model. Increasing the maximum number of features of MNB model directly affects the occurrence frequency of its feature, so when the occurrence frequency of feature is higher, the corresponding accuracy is higher and the accuracy is higher.

Model_Name	max_features	accuracy_score	precision_score (macro)	recall_score (macro)	f1_score(average='macro')	f1_score(average='micro')
DT_topics	100	0.37	0.231	0.206	0.37	0.208
	200	0.358	0.217	0.209	0.358	0.205
	400	0.358	0.217	0.209	0.358	0.205
	800	0.358	0.217	0.209	0.358	0.205
	1000	0.358	0.217	0.209	0.358	0.205
BNB_topics	100	0.348	0.205	0.194	0.348	0.195
	200	0.41	0.24	0.237	0.41	0.232
	400	0.428	0.242	0.232	0.428	0.23
	800	0.41	0.257	0.208	0.41	0.21
	1000	0.374	0.208	0.18	0.374	0.175
MNB_topics	100	0.33	0.195	0.192	0.33	0.191
	200	0.418	0.287	0.278	0.418	0.278
	400	0.446	0.268	0.256	0.446	0.254
	800	0.444	0.261	0.246	0.444	0.245
	1000	0.432	0.274	0.241	0.432	0.245

For topic data, according to the data table above, we can find that changing the value of feature will hardly affect the accuracy and other measurement results of DT model.
However, for BNB and MNB, the effect was significant, especially for MNB model. Increasing the maximum number of BNB and MNB model features to a reasonable number will improve the accuracy of the model.
According to the above data table, we can find that when the max feature is 20% of the overall data, the accuracy reaches the highest level. It can be seen that this is the sample feature quantity we hope to obtain, which is also the most reasonable.

3) Question 3

Sentiment Model	accuracy_score	precision_score (macro)	recall_score (macro)	f1_score(average='macro')
Majorityclass_baseline_sentiment	0.67	0.223	0.333	0.267
Vader_Sentiment	0.43	0.41	0.46	0.37
DT_sentiment	0.658	0.476	0.426	0.658
BNB_sentiment	0.716	0.455	0.421	0.716
MNB_sentiment	0.722	0.716	0.51	0.722
Topic Model	**accuracy_score**	**precision_score (macro)**	**recall_score (macro)**	**f1_score(average='macro')**
Majorityclass_baseline_Topic	0.174	0.009	0.05	0.015
DT_topics	0.358	0.217	0.209	0.358
BNB_topics	0.192	0.035	0.058	0.192
MNB_topics	0.326	0.194	0.148	0.326

1. For the sentiment table, all models are more accurate than Vader sentiment and Majority class baseline sentiment. The accuracy of DT model is closer to that of Majority class baseline sentiment.
   For precision and recall score, the BNB and DT models were closer to Vader sentiment. MNB is more sensitive to sentiment, with higher precision and recall score.
   For F1_score, the Vader sentiment data is not significantly related to the three standard models.
2. For the topic table, major class baseline has a great influence on the BNB model, especially accuracy score and recall score, which makes the data valuable for reference.
   But for DT and MNB models, their scores are very close to each other, which is far from

majority class baseline data

4) Question 4

Model_Name	preprocessing	accuracy_score	precision_score (macro)	recall_score (macro)	f1_score(average='macro')	f1_score(average='micro')
DT_sentiment	with	0.668	0.487	0.36	0.668	0.334
	without	0.658	0.476	0.426	0.658	0.431
BNB_sentiment	with	0.724	0.429	0.415	0.724	0.406
	without	0.716	0.455	0.421	0.716	0.415
MNB_sentiment	with	0.736	0.634	0.536	0.736	0.554
	without	0.722	0.716	0.51	0.722	0.525

As for the sentiment data, we can find from the above data table that preprocessing can correspondingly improve the accuracy, but different models have different sensitivity to it.
As for BNB and MNB model, the accuracy of preprocessing decreases, while the accuracy of DT model increases correspondingly.

Model_Name	preprocessing	accuracy_score	precision_score (macro)	recall_score (macro)	f1_score(average='macro')	f1_score(average='micro')
DT_topics	with	0.364	0.222	0.219	0.364	0.213
	without	0.358	0.217	0.209	0.358	0.205
BNB_topics	with	0.208	0.086	0.065	0.208	0.036
	without	0.192	0.035	0.058	0.192	0.028
MNB_topics	with	0.408	0.302	0.212	0.408	0.22
	without	0.326	0.194	0.148	0.326	0.148

For Topic data, we can find from the above data table that the accuracy of preprocessing is greatly improved, especially for MNB model, which is more sensitive to preprocessing.

5) Question 5

Model_Name	estion numbreprocessing	Sentiment Model	accuracy_score	precision_score (macro)	recall_score (macro)	f1_score(average='macro')	f1_score(average='micro')
DT_sentiment	2 ——	——	0.878	0.554	0.524	0.878	0.527
	3 ——	Majorityclass_baseline_sentiment					
	——	Vader_Sentiment					
	——						
	4 with	——	0.9	0.669	0.536	0.9	0.544
	without	——	0.878	0.554	0.524	0.878	0.527
BNB_sentiment	2 ——	——	0.903	0.452	0.5	0.903	0.475
	3 ——	Majorityclass_baseline_sentiment					
	——	Vader_Sentiment					
	——						
	4 with	——	0.903	0.452	0.5	0.903	0.475
	without	——	0.903	0.452	0.5	0.903	0.475
MNB_sentiment	2 ——	——	0.875	0.614	0.587	0.875	0.597
	3 ——	Majorityclass_baseline_sentiment					
	——	Vader_Sentiment					
	——						
	4 with	——	0.875	0.652	0.663	0.875	0.657
	without	——	0.875	0.614	0.587	0.875	0.597

It is easy to see from the graph above that after removing neutral data from the sample data and test data, the accuracy rate increased significantly to more than 80 %, However, precision and recall scores were not high, remaining at 50-60 percent.
So in data processing, we need to remove some ambiguous intermediate data to build a better model.

6) Question 6

1. The sentiment model use the MNB standard model

Model_Name	Action	accuracy_score	precision_score (macro)	recall_score (macro)	f1_score(average='macro')
Majorityclass_baseline_sentiment	——	0.67	0.223	0.333	0.267
Vader_Sentiment	——	0.43	0.41	0.46	0.37
MNB_sentiment	——	0.722	0.716	0.51	0.722
	Move stopwords	0.726	0.615	0.531	0.541
	Porter	0.736	0.634	0.536	0.554
	Change the train_test_split and stratify=y, test_size=0.25	0.752	0.675	0.572	0.604

My action

1) Move the stop words
2) stem the porter
3) Change the train_test_split and let the data distribution be based on the proportion of the respective data, make sure my develop model' s training data is reasonable.

Conclusion:

According to the summary table above, we can find that each adjustment increases the accuracy of the prediction, but not by much..

Move stop words which is not help us to find the sentiment could reduce the time we spend processing data. Stemming porter helps us find sentiment more precisely. Compared with two actions accuracy score, it's easy to see that the next step improves more accuracy than the previous one.

However, it is easier to improve the accuracy of the model operation, or to improve the proportion of data in the model. Like for the sentiment and dataset, we already knew The proportion for the three classes(in q1), Positive is 153, Negative is 1294, Neutral is 153.

In extreme cases, our training data does not contain positive data (because the data volume is too small), so for testing data, if there is positive data, our prediction will fail directly. That is why I change the train_test_split, in this way, the data model structure is more complete and the accuracy is higher（0.752）.

Compared with the baseline model, the accuracy and precision of the model were improved significantly. For Vader sentiment, the scores are closer to those I did not deal with in the former standard model.

2. The topics model use the DT standard model

Model_Name	Action	accuracy_score	precision_score (macro)	recall_score (macro)	f1_score(average='macro')
DT_topics	——	0.358	0.217	0.209	0.205
	Move stopwords and stem porter	0.364	0.222	0.219	0.213
	change the train_test_split and stratify=y, test_size=0.25	0.405	0.268	0.236	0.237
	Extend the Twitter number and adjust the sum of Twitter which is below the average (Sum of Twitter number / Topics number)	0.458	0.532	0.489	0.489

My action

1) Move the stop words and stem porter
2) Change the train_test_split and let the data distribution be based on the proportion of the respective data, make sure my develop model' s training data is reasonable
3) To calculate the average tweet number for all topics and increase the sum of the tweets which sum of topics numbers is lower than the average. Make sure the model is evenly distributed and the model is relatively complete.

Conclusion:

According to the summary table above, we can find that these changes have improved accuracy, and the increase is larger compared with the sentiment chart.

The basic 2 operations are the same as sentiment, but it's worth noting that I've increased the number of tweets.

Because when I see the question 1, I find that Topic (ID = 10007) only have 7 Twitters, this means that for 2,000 tweets, the topic features are almost negligible. Besides, it is hard to predict when the next tweet belongs to this topic.

So I calculate the average for all tweets according to the number of topics, the formula is all tweets number / all topics number. Then I make a dictionary to save those topics which is lower than this average. After, I add tweets according to topics and topics ID.
It works well and improve every scores higher.
I think the key to modeling is to make sure that your sample is evenly distributed.