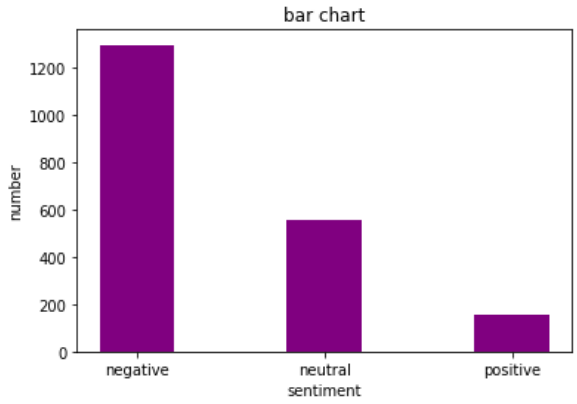
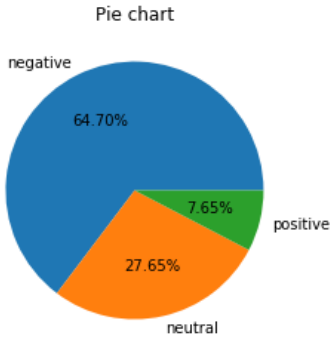
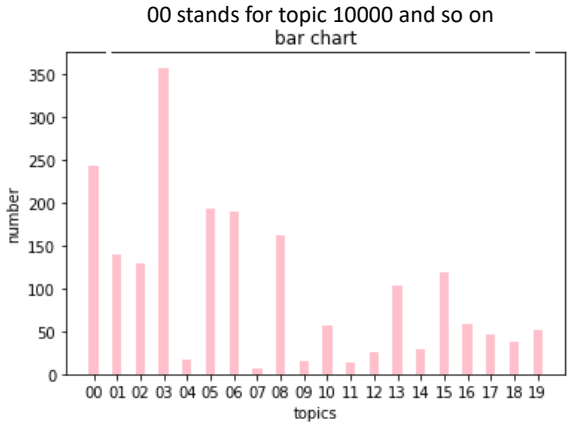
Comp9414 report Student: Keyu Yang ZID:z5177443

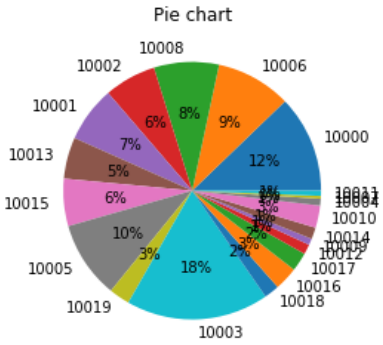
Item 1





z





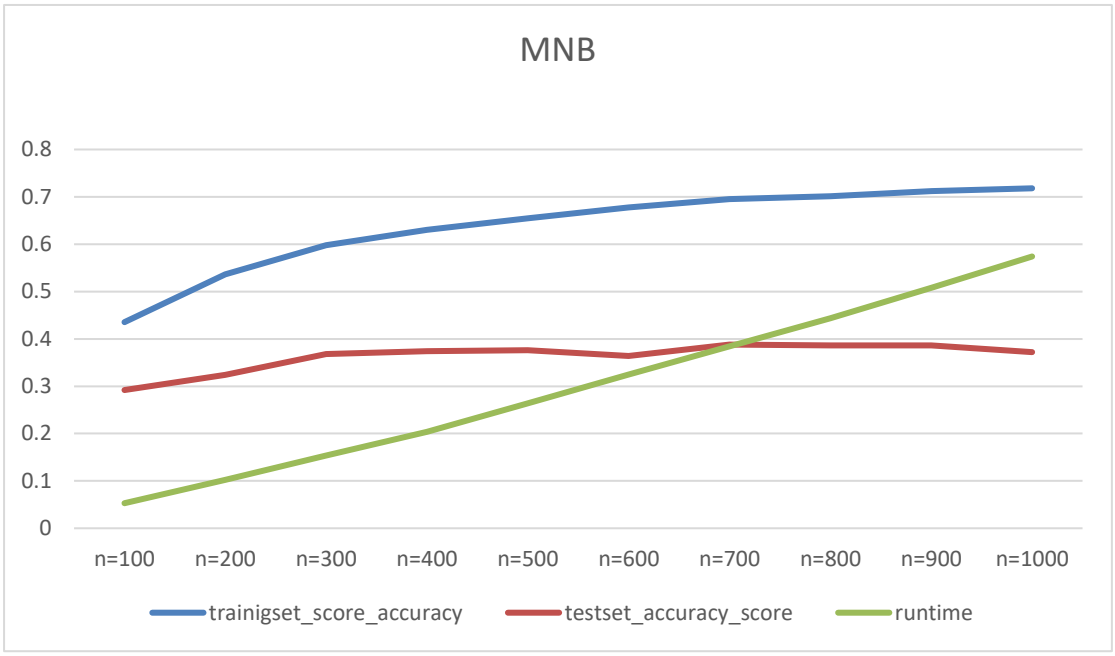
In the first bar chart and pie chart we can see that the negative sentiment occupies the largest part, which the number is 1294 of the whole dataset.tsv file. And positive sentiment occupies the smallest proportion, which only has 153 tweets. The neutral part accounts for one fourth of the whole file, there are 553 tweets.

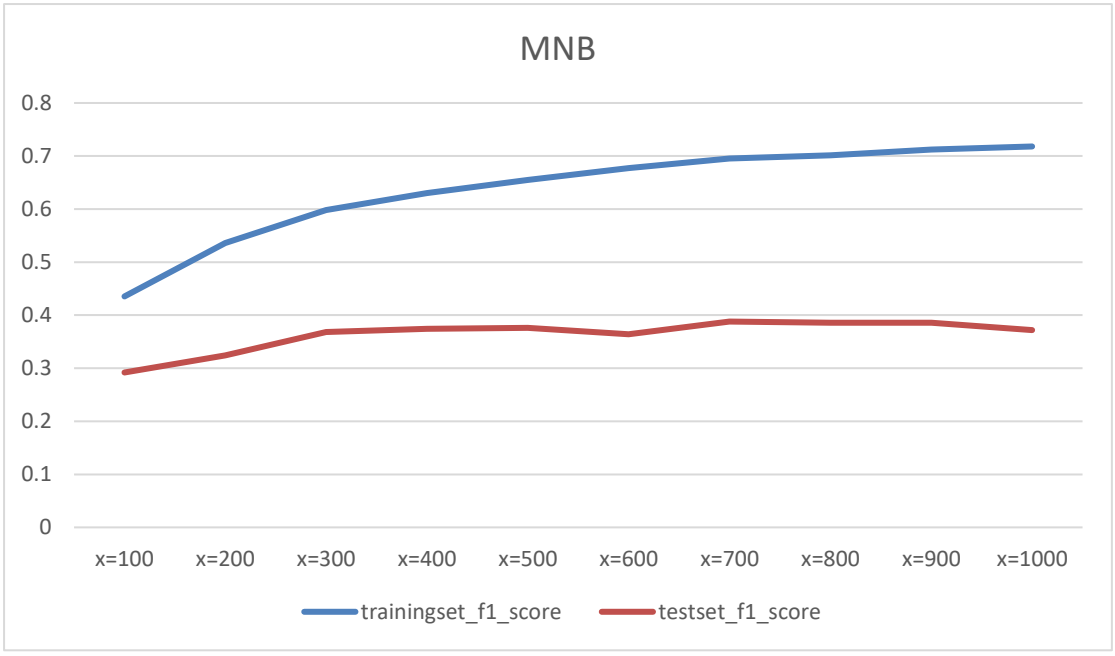
In the second chart we can see that the topic distribution is not even, which the topic 10003(economic management:358) occupies the largest part and topic 10007 takes up the minimum part, which only has 52 tweets. And here is the statistic of the topic 10000 to 10019,

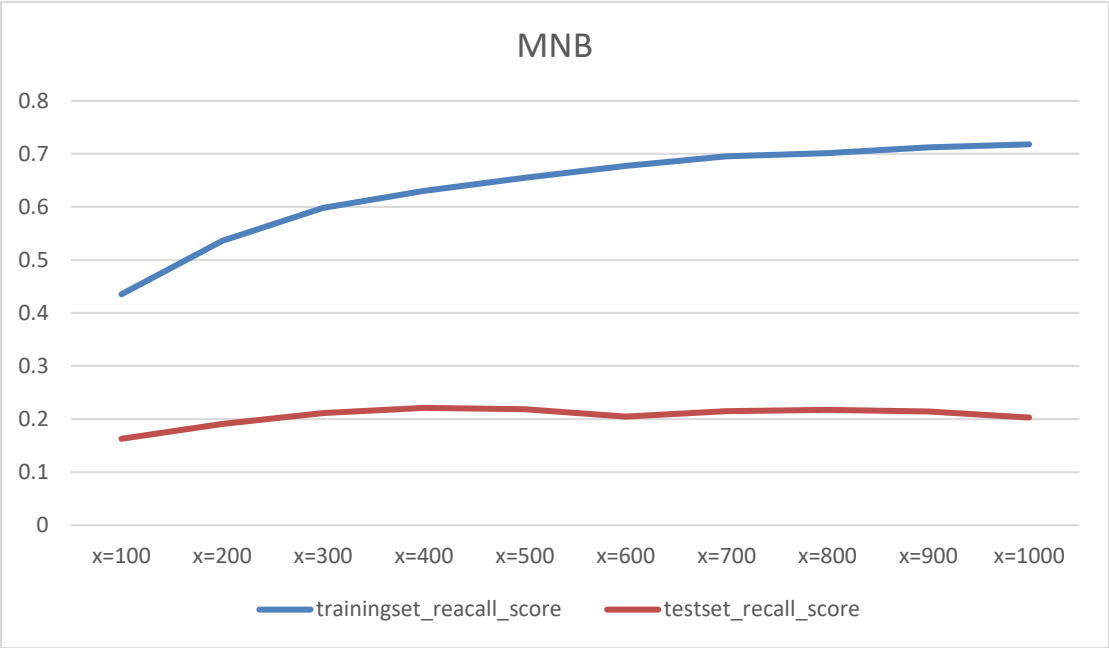
00. corruption/governance	244
01. employment/jobs	140
02. tax/negative gearing	130
03. economic management	358
04. superannuation	17
05. healthcare/medicare	194
06. social issues/marriage equality/religion	189
07. indigenous affairs	7
08. asylum seekers/refugees	163
09. early education and child care	16
10. school education	56
11. higher education	13
12. innovation/science/research	25
13. environment/climate change	104
14. infrastructure	29
15. telecommunications/nbn	119
16. terrorism/national security	59
17. foreign policy	47
18. agriculture/irrigation/dairy industry	38
19. mining and energy	52

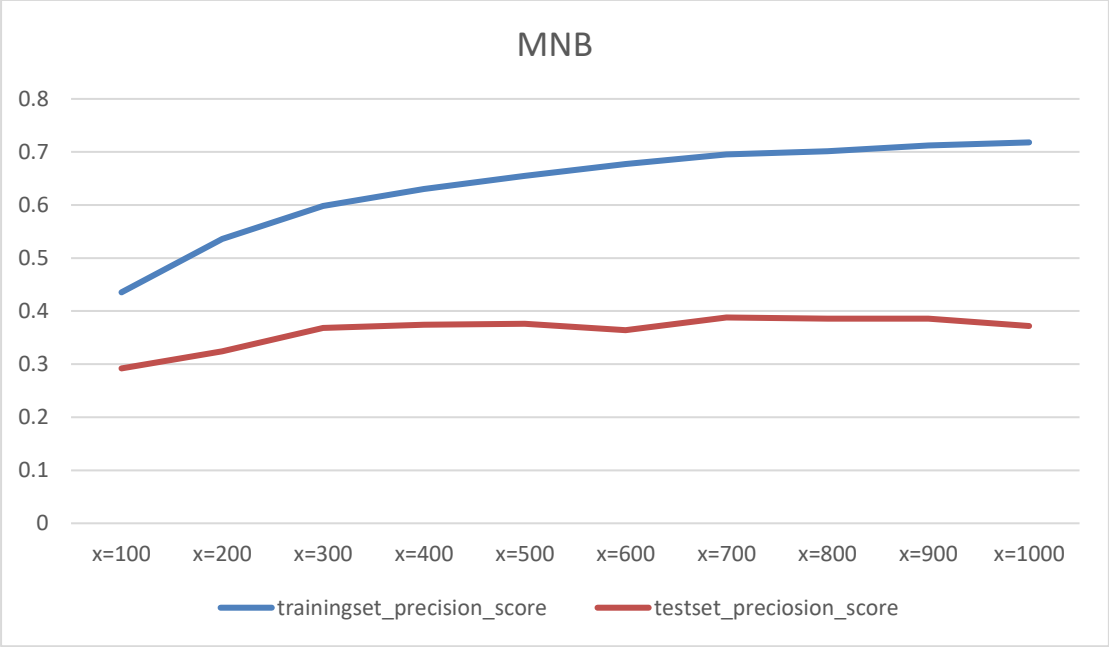
Item 2

For MNB topic analysis, I have got these charts





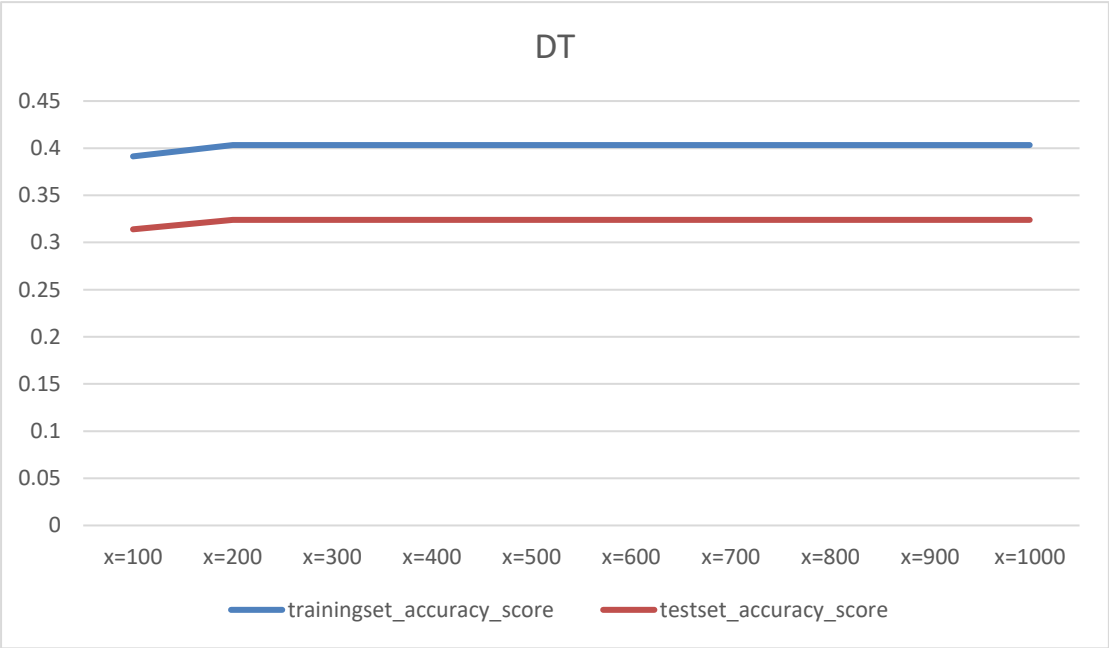




We can see that the score of training set is higher than test set in any metrics, I think this is because the prediction is based on the training set, so the score is relatively high.

And for BNB it is similar to the pattern of MNB, but in DT model it is totally different, and the graph below is the compare of accuracy score of the DT

model between in training set and test set.



For sentiment analysis,

Accuracy score	N =100	N =200	N =500	N =1000
DT				
Training_set	0.712	0.712	0.712	0.712
Test_set	0.678	0.678	0.678	0.678
BNB				
Training_set	0.735	0.768	0.82	0.85
Test_set	0.714	0.716	0.73	0.736
MNB				
Training_set	0.727	0.768	0.83	0.85
Test_set	0.728	0.724	0.74	0.754

Topic

Accuracy score	N =100	N =200	N =500	N =1000
DT				
Training_set	0.403	0.403	0.403	0.403
Test_set	0.324	0.324	0.324	0.324
BNB				
Training_set	0.442	0.512	0.594	0.582
Test_set	0.302	0.324	0.374	0.334
MNB				
Training_set	0.434	0.534	0.613	0.723
Test_set	0.292	0.324	0.376	0.372

I have calculated all the metrics of the training set and test set of all models, and I set the max features vary between 100 to 1000 , these two graphs represent the accuracy score of three models which I show the max features between 100 to 1000, and other metrics I have put it in the end of my report(less than 10 chart).

As we can see in these graphs, there are serval pattern that may be obvious.

The first it that the accuracy score of DT models in training set and test set

will not change if I raise the max features. But in BNB and MNB the metrics of training set and test set are basically remain the same or increase slowly. Secondly, the training set metrics overall higher than the test set metrics. Thirdly , as the increase of max features ,runtime will also increase in all models.

And the overfitting problem may exist, I think it is because the training data is still not enough.

Item 3

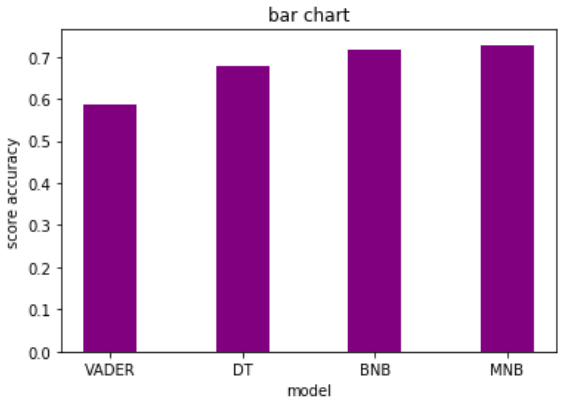
Sentiment

Vadar:

I set three area to distinguish three sentiment for VADAR sentiment analysis, which positive sentiment is between 0.05 to 1, neutral sentiment is between -0.05 to 0.05, positive sentiment is between 0.05 to 1.By calculating the number of compound tweets, and I have found that the accuracy score of the VARER sentiment analysis is about 0.43. Compared to VADER sentiment analysis, the accuracy score of DT,BNB,MNB model respectively is 0.678,0.718,0.728. It is obviously that the performance of these three models are all better than VADAR sentiment analysis and MNB model is the best model among these four.

(max_features =100)

positive	negative	neutral
125	335(0.67)	40



majority class classifiers:

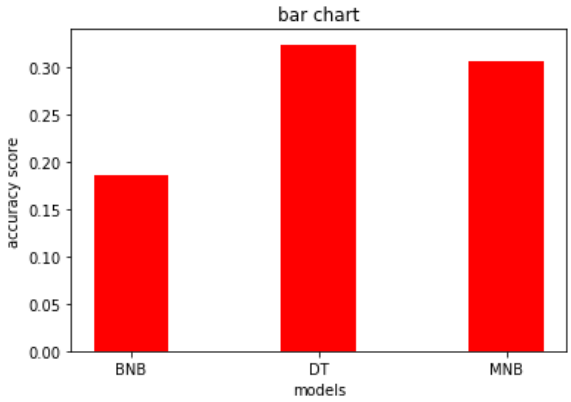
For all models, if I use majority class classifiers, which means the predicted test set should be filled by negative sentiment. And the accuracy score of this is 0.67 for all models. And this score is obviously t less than DT, BNB and MNB model, so we can know these three models can be the decent models for analyzing sentiment.

N=100	Precision_score	Recall_score	F1-score
DT	0.74	0.84	0.72
BNB	0.77	0.88	0.71
MNB	0.75	0.89	0.82

Topic

By calculating the accuracy score of three different models for topic analysis, the performance of DT model is the best one, which the accuracy score is 0.324. Then followed by MNB model(0.306),

and the last one is BNB, which the score is about 0.186.



majority class classifiers:

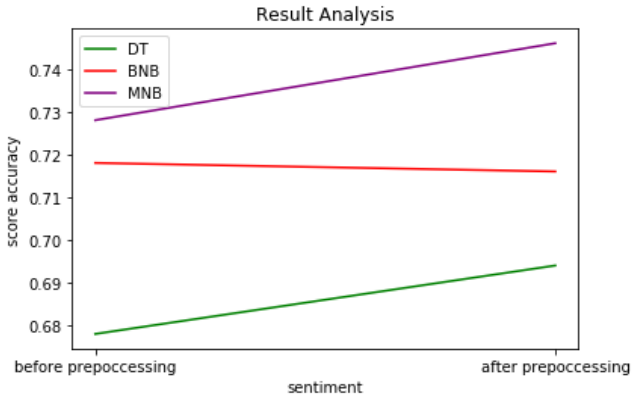
Same as the analysis of sentiment, the topic 10003 accounts for the largest part in dataset.tsv file, therefore our predicted set should be filled by this topic. And after calculating, the accuracy score is 0.174. And this score is lower than any of these three, so these three models are meaningful.

Item 4

There are several graphs below showing that the statistic of the performance of three models before preprocessing and after preprocessing(Max_features = 200).

Sentiment(for test set):

Sentiment	Before	after
DT	0.678	0.694
BNB	0.718	0.726
MNB	0.728	0.746

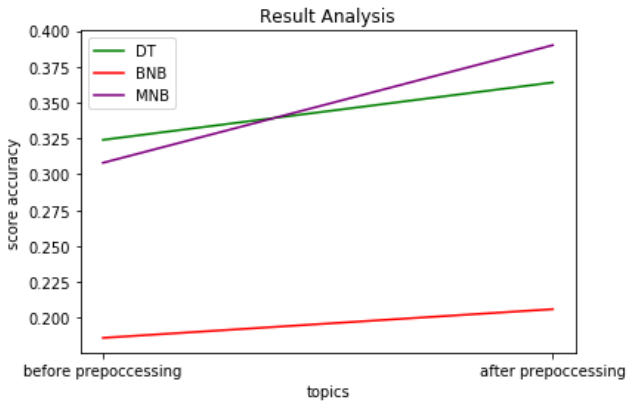


As we can see from these graphs, it is obvious that most of the score accuracy of models have been increased after preprocessing except BNB model in sentiment analysis, which the score accuracy decreases about 0.02. And the other two models' accuracy in sentiment analysis also rise.

Topic(test set):

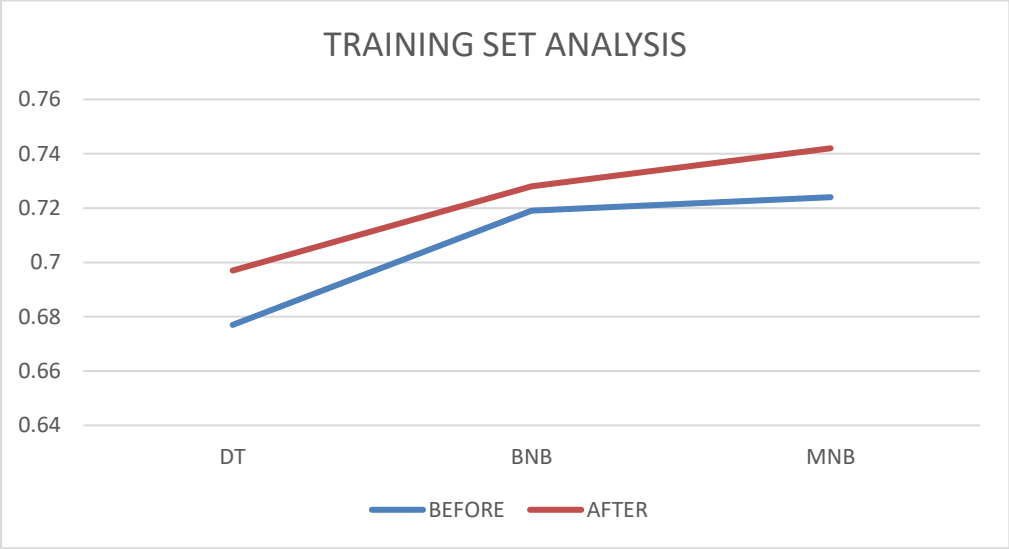
Topic	Before	after

DT	0.324	0.364
BNB	0.186	0.206
MNB	0.308	0.390



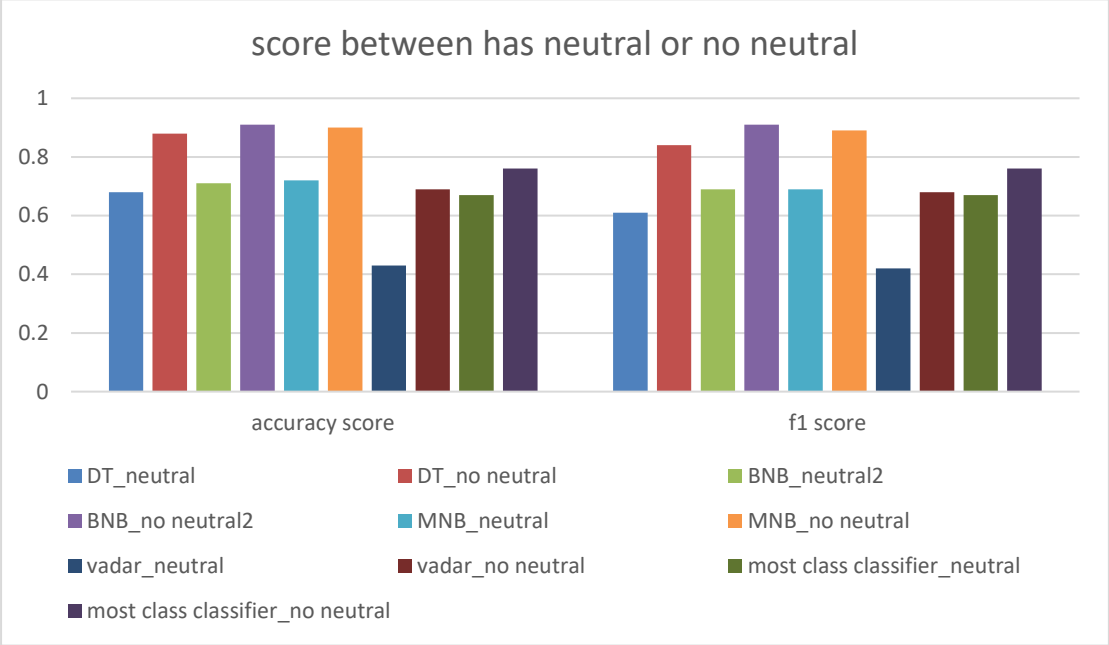
In topic analysis, the accuracy of MNB model has increased dramatically which is approximate 10 percent and the accuracy of DT model and BNB model also have been raised, which is respectively from 0.324 to 0.364 and 0.186 to 0.206.

Similarly,I compare the metrics for training set before preprocessing and after preprocessing



Item 5

max features = 200



As it is obvious in the chart removing neutral tweets has significantly improve the accuracy of each models.

Item 6

BEST METHOD

My model is based on MNB model. For sentiment and topic analysis, I think there are several methods for increasing the metrics for the models. Firstly, removing punctuation except '#, @, _, $ or %'. In my program ,it may not improve our program significantly, but as for my program , the accuracy score of sentiment analysis raises about 0.002.So it still can improve our models in some way. After '#, @, _, $ or %' these punctuation it may has something related to our topic, therefore keeping them is the best choice.

And in my topic.py and sentiment.py file, I set the max_features = 400 which could have the best accuracy.

Then, removing stopwords can raise the accuracy. as I explained in item 4,both of the training set and test set's accuracy improve. And the average raise is about 2%.

Sentiment	Before	after
DT	0.678	0.694
BNB	0.718	0.726
MNB	0.728	0.746

Thirdly, considering the sarcastic tweets problems. Because it is obvious that sarcastic tweets are the sentences which are made of some positive

words, so it is really hard for our model to distinguish these tweets. I think the punctuation(like ? and so on) or some emoji in this case is extremely important, we can tell the sarcasm by analyzing these signals.

Lastly, adding the small number tweets in our content probably is a good choice. Because the distribution of our models is not even.so there is a case that in our first 1500 tweets(also our training set),some small number topics(such as topic 10007) may not exist. Therefore, our models can not predict the right answer if some topics has never been analyzed.

How I ACHIEVE IT

First, for natural language processing I used the ntlk to preprocess the content which aims to remove the stopwords and wordstem. Then in order to avoid the uneven distribution, I added the number of tweets in training set which number less than 200 and increase them to 200. So I use a method from scikit-learn which called *train_test_split* to split my content in four parts(training set, test set, training set, test set),And I set the test_size attribute to 0.25,which 75 percent whole dataset is treated as training set.And in the end I managed to rise the accuracy score of topic analysis to 0.71 which has increased about 30 percent compared to the original one.

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.25, random_state=1,stratify = Y)
```

comparison