COMP9414: ASSIGNMENT-2

Kavya Musty (z5240707)

QUESTION 1: The frequency distribution of 'dataset' file in terms of sentiment and topic id is given as

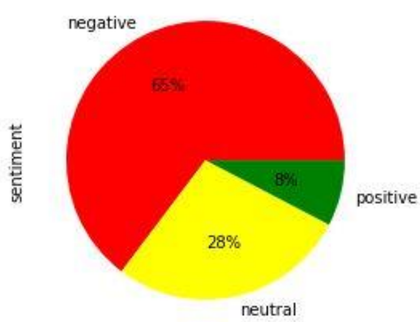
	Counts	Percentage
negative	1294	64.70
neutral	553	27.65
positive	153	7.65

Sentiment Frequency Distribution

	Counts	Percentage
10003	358	17.90
10000	244	12.20
10005	194	9.70
10006	189	9.45
10008	163	8.15
10001	140	7.00
10002	130	6.50
10015	119	5.95
10013	104	5.20
10016	59	2.95
10010	56	2.80
10019	52	2.60
10017	47	2.35
10018	38	1.90
10014	29	1.45
10012	25	1.25
10004	17	0.85
10009	16	0.80
10011	13	0.65
10007	7	0.35

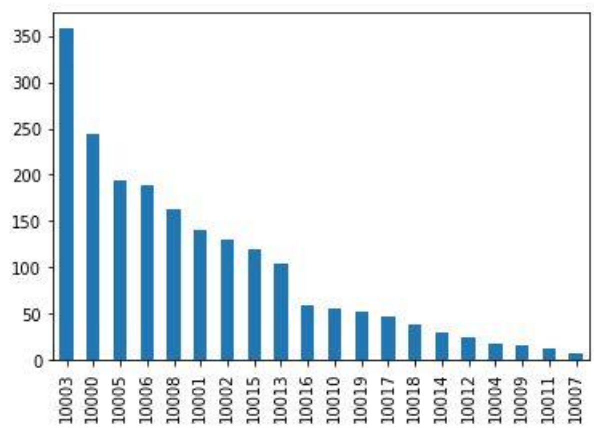
Topic id Frequency Distribution

The sentiment data is also represented in a pie chart form as:



Sentiment Frequency

The topic id data is also represented in a bar chart form as:



Topic id Frequency

- It can be deduced from **sentiment** frequency, that there are more 'negative' tweets accounting up to 65% of the dataset, followed by 'neutral' tweets (28%) and then 'positive' tweets (8%).
- From the **topic id** frequency, it can be observed that 'economic management' (10003) is one of the most discussed topics among the tweets, followed by 'corruption/governance' and so on. The least discussed topic is 'indigenous affairs' (10007).

QUESTION 2: The comparison of metrics between train data and test data of both sentiment and topics for standard models are:

Bernoulli Naïve Bayes (Sentiment) between train and test data after limiting the vocabulary to 100 words by using **'max_features=100'** is:

	precision	recall	f1-score	support
negative	0.77	0.88	0.82	967
neutral	0.61	0.47	0.53	420
positive	0.56	0.31	0.40	113
accuracy			0.72	1500
macro avg	0.64	0.55	0.58	1500
weighted avg	0.71	0.72	0.71	1500

	precision	recall	f1-score	support
negative	0.75	0.85	0.80	327
neutral	0.55	0.46	0.50	133
positive	0.38	0.15	0.21	40
accuracy			0.69	500
macro avg	0.56	0.49	0.50	500
weighted avg	0.67	0.69	0.67	500

Train data vs Test data for BNB Sentiment with max_features=100

Bernoulli Naïve Bayes (Sentiment) model when trained with a vocabulary of 200 words using **'max_features=200'** is:

	precision	recall	f1-score	support
negative	0.80	0.87	0.83	967
neutral	0.63	0.56	0.59	420
positive	0.68	0.43	0.53	113
accuracy			0.75	1500
macro avg	0.70	0.62	0.65	1500
weighted avg	0.74	0.75	0.74	1500

	precision	recall	f1-score	support
negative	0.76	0.83	0.79	327
neutral	0.53	0.49	0.51	133
positive	0.43	0.23	0.30	40
accuracy			0.69	500
macro avg	0.57	0.51	0.53	500
weighted avg	0.67	0.69	0.68	500

Train data vs Test data for BNB Sentiment with max_features=200

BNB Sentiment with max_features=100	BNB Sentiment with max_features=200
• For this model, the run time has been 0.1675 *seconds*.	• The run time is comparatively longer for this case amounting around 0.2194 *seconds*.
• It can be observed that the accuracy for training data is 72% while the accuracy for the test data is 69%. The model is not training with enough accuracy due to the limiting of the vocabulary which in turn affects the test data prediction accuracy, resulting in a pretty low accuracy compared to the standard model which has an accuracy of 70%.	• The train data accuracy has been slightly improved than the previous case by 2 percent. But the accuracy of the test data is constant. It can be deduced that; the model needs more vocabulary in order to improve its accuracy for the test data.

Bernoulli Naïve Bayes (Topic) model classification report with a vocabulary of 100 words whose **'max_features=100'** is:

	precision	recall	f1-score	support
10000	0.38	0.65	0.48	178
10001	0.47	0.36	0.41	101
10002	0.50	0.45	0.47	103
10003	0.29	0.50	0.36	267
10004	0.00	0.00	0.00	14
10005	0.59	0.50	0.54	154
10006	0.32	0.29	0.30	143
10007	0.00	0.00	0.00	6
10008	0.46	0.35	0.40	118
10009	0.17	0.08	0.11	13
10010	0.47	0.19	0.27	43
10011	0.00	0.00	0.00	10
10012	1.00	0.06	0.11	17
10013	0.39	0.31	0.35	77
10014	0.71	0.23	0.34	22
10015	0.82	0.70	0.75	92
10016	0.53	0.19	0.28	42
10017	0.58	0.21	0.30	34
10018	0.33	0.08	0.13	25
10019	0.43	0.07	0.12	41
accuracy			0.41	1500
macro avg	0.42	0.26	0.29	1500
weighted avg	0.44	0.41	0.40	1500

	precision	recall	f1-score	support
10000	0.32	0.56	0.41	66
10001	0.21	0.13	0.16	39
10002	0.40	0.44	0.42	27
10003	0.20	0.30	0.24	91
10004	0.00	0.00	0.00	3
10005	0.39	0.50	0.44	40
10006	0.13	0.13	0.13	46
10007	0.00	0.00	0.00	1
10008	0.33	0.20	0.25	45
10009	0.00	0.00	0.00	3
10010	0.00	0.00	0.00	13
10011	0.00	0.00	0.00	3
10012	0.00	0.00	0.00	8
10013	0.41	0.33	0.37	27
10014	0.33	0.14	0.20	7
10015	0.65	0.63	0.64	27
10016	0.00	0.00	0.00	17
10017	0.00	0.00	0.00	13
10018	0.00	0.00	0.00	13
10019	0.00	0.00	0.00	11
accuracy			0.29	500
macro avg	0.17	0.17	0.16	500
weighted avg	0.25	0.29	0.26	500

Train data vs Test data for BNB Topic with max_features=100

Bernoulli Naïve Bayes (Topic) model classification report with a vocabulary of 200 words whose **'max_features=200'** is:

	precision	recall	f1-score	support
10000	0.41	0.74	0.53	178
10001	0.60	0.48	0.53	101
10002	0.64	0.59	0.62	103
10003	0.39	0.57	0.46	267
10004	0.00	0.00	0.00	14
10005	0.53	0.57	0.55	154
10006	0.44	0.40	0.42	143
10007	0.00	0.00	0.00	6
10008	0.65	0.53	0.59	118
10009	0.50	0.08	0.13	13
10010	0.48	0.33	0.39	43
10011	0.00	0.00	0.00	10
10012	0.00	0.00	0.00	17
10013	0.41	0.35	0.38	77
10014	1.00	0.18	0.31	22
10015	0.81	0.74	0.77	92
10016	0.67	0.19	0.30	42
10017	0.77	0.29	0.43	34
10018	0.67	0.16	0.26	25
10019	0.55	0.15	0.23	41
accuracy			0.49	1500
macro avg	0.48	0.32	0.34	1500
weighted avg	0.52	0.49	0.48	1500

	precision	recall	f1-score	support
10000	0.36	0.61	0.45	66
10001	0.22	0.13	0.16	39
10002	0.33	0.41	0.37	27
10003	0.31	0.43	0.36	91
10004	0.00	0.00	0.00	3
10005	0.38	0.50	0.43	40
10006	0.32	0.33	0.32	46
10007	0.00	0.00	0.00	1
10008	0.43	0.40	0.41	45
10009	0.00	0.00	0.00	3
10010	0.29	0.15	0.20	13
10011	0.00	0.00	0.00	3
10012	0.00	0.00	0.00	8
10013	0.44	0.26	0.33	27
10014	0.00	0.00	0.00	7
10015	0.56	0.70	0.62	27
10016	0.00	0.00	0.00	17
10017	0.00	0.00	0.00	13
10018	0.00	0.00	0.00	13
10019	0.00	0.00	0.00	11
accuracy			0.35	500
macro avg	0.18	0.20	0.18	500
weighted avg	0.30	0.35	0.32	500

Train data vs Test data for BNB Topic with max_features=200

BNB Topic with max_features=100	**BNB Topic with max_features=200**
• The run time for this model is 0.162 *seconds*.	• The run time for this model is comparatively low with 0.153 *seconds*.
• The accuracy for this model is low for both train data (41%) and test data (29%). We can observe that the least discussed topic (10000) is being predicted with more accuracy as compared to other topics and the most occurring topic (10007) is not being predicted at all. This tells us that the model is not being trained well with just 100 words in the vocabulary.	• The accuracy has been improved significantly for this model for both train data (49%) and test data (35%). The topic (10000) prediction has also increased significantly. But the most occurring topic (10007) has still the prediction of zero. Even 200 words in the vocabulary is not enough to increase the prediction.

Multinomial Naïve Bayes (Sentiment) between train and test data after limiting the vocabulary to 100 words by using **'max_features=100'** is:

	precision	recall	f1-score	support
negative	0.76	0.90	0.83	967
neutral	0.64	0.45	0.53	420
positive	0.58	0.27	0.37	113
accuracy			0.73	1500
macro avg	0.66	0.54	0.58	1500
weighted avg	0.71	0.73	0.71	1500

	precision	recall	f1-score	support
negative	0.74	0.86	0.80	327
neutral	0.53	0.42	0.47	133
positive	0.47	0.17	0.25	40
accuracy			0.69	500
macro avg	0.58	0.49	0.51	500
weighted avg	0.66	0.69	0.67	500

Train data vs Test data for MNB Sentiment with max_features=100

Multinomial Naïve Bayes (Sentiment) between train and test data after limiting the vocabulary to 200 most occurring words by using **'max_features=200'** is:

	precision	recall	f1-score	support
negative	0.81	0.89	0.85	967
neutral	0.66	0.57	0.61	420
positive	0.73	0.47	0.57	113
accuracy			0.77	1500
macro avg	0.73	0.64	0.68	1500
weighted avg	0.76	0.77	0.76	1500

	precision	recall	f1-score	support
negative	0.75	0.80	0.77	327
neutral	0.50	0.45	0.47	133
positive	0.37	0.25	0.30	40
accuracy			0.67	500
macro avg	0.54	0.50	0.52	500
weighted avg	0.65	0.67	0.66	500

Train data vs Test data for MNB Sentiment with max_features=200

MNB Sentiment with max_features=100	**MNB Sentiment with max_features=200**
• The run time of this model is 0.153 *seconds.*	• The run time for this model is 0.174 *seconds* which indicates that, it takes more time than with less vocabulary and also has accuracy in its prediction.
• The accuracy for this model is 69% which is low than the standard model which predicts with an accuracy of 75%. The train data reads the sentiment with more accuracy as compared to the test set. Overall, limiting the vocabulary is not helping the model to be trained precisely.	• The accuracy for this model is 67% for its test set. It seems to be decreasing in its accuracy for test set. But the accuracy of training set seems to improve significantly. So, increasing the vocabulary is decreasing the prediction score for test data.

Multinomial Naïve Bayes (Topic) between train and test data after limiting the vocabulary to 100 words by using **'max_features=100'** is:

	precision	recall	f1-score	support
10000	0.46	0.61	0.53	178
10001	0.44	0.31	0.36	101
10002	0.51	0.45	0.47	103
10003	0.27	0.53	0.35	267
10004	1.00	0.07	0.13	14
10005	0.58	0.53	0.55	154
10006	0.33	0.31	0.32	143
10007	0.00	0.00	0.00	6
10008	0.50	0.33	0.40	118
10009	0.57	0.31	0.40	13
10010	0.43	0.21	0.28	43
10011	0.00	0.00	0.00	10
10012	0.50	0.06	0.11	17
10013	0.42	0.34	0.37	77
10014	0.71	0.23	0.34	22
10015	0.79	0.71	0.75	92
10016	0.50	0.14	0.22	42
10017	0.50	0.18	0.26	34
10018	0.33	0.12	0.18	25
10019	0.40	0.05	0.09	41
accuracy			0.41	1500
macro avg	0.46	0.27	0.31	1500
weighted avg	0.45	0.41	0.40	1500

	precision	recall	f1-score	support
10000	0.39	0.56	0.46	66
10001	0.15	0.10	0.12	39
10002	0.45	0.48	0.46	27
10003	0.22	0.38	0.28	91
10004	0.00	0.00	0.00	3
10005	0.38	0.45	0.41	40
10006	0.14	0.15	0.14	46
10007	0.00	0.00	0.00	1
10008	0.33	0.18	0.23	45
10009	0.00	0.00	0.00	3
10010	0.14	0.08	0.10	13
10011	0.00	0.00	0.00	3
10012	0.00	0.00	0.00	8
10013	0.32	0.26	0.29	27
10014	0.50	0.14	0.22	7
10015	0.62	0.59	0.60	27
10016	0.00	0.00	0.00	17
10017	0.00	0.00	0.00	13
10018	0.00	0.00	0.00	13
10019	0.00	0.00	0.00	11
accuracy			0.29	500
macro avg	0.18	0.17	0.17	500
weighted avg	0.26	0.29	0.27	500

Train data vs Test data for MNB Topic with max_features=100

Multinomial Naïve Bayes (Topic) between train and test data after limiting the vocabulary to 200 words by using **'max_features=200'** is:

	precision	recall	f1-score	support
10000	0.48	0.66	0.56	178
10001	0.57	0.47	0.51	101
10002	0.63	0.59	0.61	103
10003	0.39	0.60	0.47	267
10004	0.50	0.07	0.12	14
10005	0.54	0.63	0.58	154
10006	0.47	0.43	0.45	143
10007	0.00	0.00	0.00	6
10008	0.69	0.53	0.60	118
10009	0.67	0.46	0.55	13
10010	0.56	0.47	0.51	43
10011	0.50	0.10	0.17	10
10012	0.50	0.06	0.11	17
10013	0.45	0.38	0.41	77
10014	1.00	0.27	0.43	22
10015	0.78	0.75	0.76	92
10016	0.82	0.21	0.34	42
10017	0.72	0.38	0.50	34
10018	0.62	0.40	0.49	25
10019	0.67	0.15	0.24	41
accuracy			0.52	1500
macro avg	0.58	0.38	0.42	1500
weighted avg	0.55	0.52	0.51	1500

	precision	recall	f1-score	support
10000	0.44	0.59	0.50	66
10001	0.22	0.13	0.16	39
10002	0.39	0.48	0.43	27
10003	0.28	0.41	0.33	91
10004	0.00	0.00	0.00	3
10005	0.34	0.42	0.38	40
10006	0.31	0.33	0.32	46
10007	0.00	0.00	0.00	1
10008	0.42	0.40	0.41	45
10009	0.00	0.00	0.00	3
10010	0.38	0.23	0.29	13
10011	0.00	0.00	0.00	3
10012	0.00	0.00	0.00	8
10013	0.35	0.26	0.30	27
10014	0.00	0.00	0.00	7
10015	0.53	0.67	0.59	27
10016	0.17	0.06	0.09	17
10017	0.00	0.00	0.00	13
10018	0.50	0.08	0.13	13
10019	0.00	0.00	0.00	11
accuracy			0.35	500
macro avg	0.22	0.20	0.20	500
weighted avg	0.32	0.35	0.32	500

Train data vs Test data for MNB Topic with max_features=200

MNB Topic with max_features=100	MNB Topic with max_features=200
• The run time for this model is 0.138 *seconds*.	• The run time for this model is 0.185 *seconds* but predicts with more accuracy than the 100-word limited model.
• The accuracy for this model is 29% which is less than the standard model which has an accuracy around 36%. The train data has more precision than the	• The accuracy for this model's test data is around 35% which is nearer to the standard model with an accuracy of 36%. This model reads both the train

test data but the model requires more vocabulary to increase its accuracy.	and test data with more accuracy and predicts with good precision.

Decision Tree Classifier (Sentiment) between train and test data after limiting the vocabulary to 100 most occurring words by using **'max_features=100'** is:

	precision	recall	f1-score	support
negative	0.64	1.00	0.78	967
neutral	0.00	0.00	0.00	420
positive	0.00	0.00	0.00	113
accuracy			0.64	1500
macro avg	0.21	0.33	0.26	1500
weighted avg	0.42	0.64	0.51	1500

	precision	recall	f1-score	support
negative	0.65	1.00	0.79	327
neutral	0.00	0.00	0.00	133
positive	0.00	0.00	0.00	40
accuracy			0.65	500
macro avg	0.22	0.33	0.26	500
weighted avg	0.43	0.65	0.52	500

Train data vs Test data for DT Sentiment with max_features=100

Decision Tree Classifier (Sentiment) between train and test data after limiting the vocabulary to 200 most occurring words by using **'max_features=200'** is:

	precision	recall	f1-score	support
negative	0.67	0.96	0.79	967
neutral	0.50	0.14	0.22	420
positive	0.00	0.00	0.00	113
accuracy			0.66	1500
macro avg	0.39	0.37	0.34	1500
weighted avg	0.57	0.66	0.57	1500

	precision	recall	f1-score	support
negative	0.66	0.95	0.78	327
neutral	0.34	0.08	0.13	133
positive	0.00	0.00	0.00	40
accuracy			0.64	500
macro avg	0.34	0.34	0.31	500
weighted avg	0.53	0.64	0.55	500

Train data vs Test data for DT Sentiment with max_features=200

DT Sentiment with max_features=100	DT Sentiment with max_features=200
• The run time for this model is 0.496 *seconds*.	• The run time for this model is 0.65 *seconds* which is higher than the other model and also predicts with less accuracy than the other model.
• The precision for this model is 65% for the test data which is more than the standard model which has an accuracy of 64%. This indicates that by giving more features to the model, the decision tree is getting over pruned which might lead to a decrease in the accuracy.	• The precision for this model is 64% which is same as the standard model. By giving more feature words, the decision tree might get over pruned and produce defective results for some cases which might lead the to low precision than the previous model.

Decision Tree Classifier (Topic) between train and test data after limiting the vocabulary to 100 most occurring words by using **'max_features=100'** is:

	precision	recall	f1-score	support
10000	0.62	0.10	0.17	178
10001	0.00	0.00	0.00	101
10002	0.00	0.00	0.00	103
10003	0.18	0.96	0.30	267
10004	0.00	0.00	0.00	14
10005	0.40	0.11	0.17	154
10006	0.00	0.00	0.00	143
10007	0.00	0.00	0.00	6
10008	0.00	0.00	0.00	118
10009	0.00	0.00	0.00	13
10010	0.00	0.00	0.00	43
10011	0.00	0.00	0.00	10
10012	0.00	0.00	0.00	17
10013	0.00	0.00	0.00	77
10014	0.00	0.00	0.00	22
10015	0.00	0.00	0.00	92
10016	0.00	0.00	0.00	42
10017	0.00	0.00	0.00	34
10018	0.00	0.00	0.00	25
10019	0.00	0.00	0.00	41
accuracy			0.19	1500
macro avg	0.06	0.06	0.03	1500
weighted avg	0.15	0.19	0.09	1500

	precision	recall	f1-score	support
10000	0.64	0.14	0.23	66
10001	0.00	0.00	0.00	39
10002	0.00	0.00	0.00	27
10003	0.18	0.96	0.31	91
10004	0.00	0.00	0.00	3
10005	0.64	0.17	0.27	40
10006	0.00	0.00	0.00	46
10007	0.00	0.00	0.00	1
10008	0.00	0.00	0.00	45
10009	0.00	0.00	0.00	3
10010	0.00	0.00	0.00	13
10011	0.00	0.00	0.00	3
10012	0.00	0.00	0.00	8
10013	0.00	0.00	0.00	27
10014	0.00	0.00	0.00	7
10015	0.00	0.00	0.00	27
10016	0.00	0.00	0.00	17
10017	0.00	0.00	0.00	13
10018	0.00	0.00	0.00	13
10019	0.00	0.00	0.00	11
accuracy			0.21	500
macro avg	0.07	0.06	0.04	500
weighted avg	0.17	0.21	0.11	500

Train data vs Test data for DT Topic with max_features=100

Decision Tree Classifier (Topic) between train and test data after limiting the vocabulary to 200 most occurring words by using **'max_features=200'** is:

	precision	recall	f1-score	support
10000	0.78	0.30	0.43	178
10001	0.37	0.19	0.25	101
10002	0.48	0.31	0.38	103
10003	0.19	0.88	0.31	267
10004	0.00	0.00	0.00	14
10005	0.25	0.10	0.15	154
10006	0.00	0.00	0.00	143
10007	0.00	0.00	0.00	6
10008	0.00	0.00	0.00	118
10009	0.00	0.00	0.00	13
10010	0.38	0.21	0.27	43
10011	0.00	0.00	0.00	10
10012	0.00	0.00	0.00	17
10013	0.00	0.00	0.00	77
10014	0.00	0.00	0.00	22
10015	0.00	0.00	0.00	92
10016	0.00	0.00	0.00	42
10017	0.00	0.00	0.00	34
10018	0.00	0.00	0.00	25
10019	0.00	0.00	0.00	41
accuracy			0.24	1500
macro avg	0.12	0.10	0.09	1500
weighted avg	0.22	0.24	0.17	1500

	precision	recall	f1-score	support
10000	0.77	0.35	0.48	66
10001	0.47	0.18	0.26	39
10002	0.56	0.56	0.56	27
10003	0.20	0.87	0.32	91
10004	0.00	0.00	0.00	3
10005	0.12	0.07	0.09	40
10006	0.00	0.00	0.00	46
10007	0.00	0.00	0.00	1
10008	0.00	0.00	0.00	45
10009	0.00	0.00	0.00	3
10010	0.43	0.23	0.30	13
10011	0.00	0.00	0.00	3
10012	0.00	0.00	0.00	8
10013	0.00	0.00	0.00	27
10014	0.00	0.00	0.00	7
10015	0.00	0.00	0.00	27
10016	0.00	0.00	0.00	17
10017	0.00	0.00	0.00	13
10018	0.00	0.00	0.00	13
10019	0.00	0.00	0.00	11
accuracy			0.26	500
macro avg	0.13	0.11	0.10	500
weighted avg	0.22	0.26	0.19	500

Train data vs Test data for DT Topic with max_features=100

DT Topic with max_features=100	DT Topic with max_features=200
• The run time of this model is 0.456 *seconds*.	• The run time for this model is 0.65 seconds which is higher than the other model but predicts with more accuracy with test data.
• The accuracy for test data is 21% which is lower than the standard model which accounts up to 24%. The accuracy for test data is pretty low. As there are a greater number of topics to be classified, the model requires more	• The accuracy for the test data for this model is 26% which is same as the standard model. The classification precision is more accurate for this model as the vocabulary consists of more words and there are many topics to be classified. The model needed

words in its vocabulary to increase its accuracy.	more vocabulary which is provided in this model.

QUESTION 3: Standard models (Sentiment) when compared to **'Majority class' Sentiment** is given below:

Photos - Standard_MNB_Sentiment.JPG

	precision	recall	f1-score	support
negative	0.79	0.89	0.84	327
neutral	0.66	0.59	0.62	133
positive	0.75	0.23	0.35	40
accuracy			0.76	500
macro avg	0.73	0.57	0.60	500
weighted avg	0.75	0.76	0.74	500

Photos - Standard_BNB_Sentiment.JPG

	precision	recall	f1-score	support
negative	0.69	0.98	0.81	327
neutral	0.76	0.22	0.34	133
positive	0.00	0.00	0.00	40
accuracy			0.70	500
macro avg	0.48	0.40	0.38	500
weighted avg	0.65	0.70	0.62	500

Photos - Standard_DT_Sentiment.JPG

	precision	recall	f1-score	support
negative	0.66	0.95	0.78	327
neutral	0.34	0.08	0.13	133
positive	0.00	0.00	0.00	40
accuracy			0.64	500
macro avg	0.34	0.34	0.31	500
weighted avg	0.53	0.64	0.55	500

Photos - MajorityClass_Sentiment.JPG

	precision	recall	f1-score	support
negative	0.65	1.00	0.79	327
neutral	0.00	0.00	0.00	133
positive	0.00	0.00	0.00	40
accuracy			0.65	500
macro avg	0.22	0.33	0.26	500
weighted avg	0.43	0.65	0.52	500

Classification report--Top right: Standard Multinomial Sentiment, Top left: Standard Bernoulli Sentiment, Bottom right: Standard Decision Tree Sentiment, Bottom Left: Majority Class Sentiment

- Out of all the models above, Multinomial Naïve Bayes has more prediction accuracy with more precision in each of the sentiment classification too.
- Majority Class sentiment is predicted using DummyClassifier(): DummyClassifier(strategy='most_frequent', random_state=None, constant=None)
- The strategy attribute is set to 'most_frequent' which indicates that the model always predicts the most frequent label in the training set.
- In conclusion, the majority class classification is more accurate than the Decision Tree Classifier but not as good as Bernoulli or Multinomial Naïve Bayes Classifiers.

Standard models (Topic) when compared to **'Majority class' Topic** is given below:

Photos - Standard_MNB_Topic.JPG

	precision	recall	f1-score	support
10000	0.57	0.55	0.56	66
10001	0.00	0.00	0.00	39
10002	0.30	0.11	0.16	27
10003	0.25	0.82	0.38	91
10004	0.00	0.00	0.00	3
10005	0.32	0.33	0.32	40
10006	0.48	0.26	0.34	46
10007	0.00	0.00	0.00	1
10008	0.50	0.27	0.35	45
10009	0.00	0.00	0.00	3
10010	0.00	0.00	0.00	13
10011	0.00	0.00	0.00	3
10012	0.00	0.00	0.00	8
10013	0.40	0.15	0.22	27
10014	0.00	0.00	0.00	7
10015	0.53	0.30	0.38	27
10016	0.00	0.00	0.00	17
10017	0.00	0.00	0.00	13
10018	0.00	0.00	0.00	13
10019	0.00	0.00	0.00	11
accuracy			0.33	500
macro avg	0.17	0.14	0.14	500
weighted avg	0.30	0.33	0.27	500

Photos - Standard_BNB_Topic.JPG

	precision	recall	f1-score	support
10000	0.50	0.08	0.13	66
10001	0.00	0.00	0.00	39
10002	0.00	0.00	0.00	27
10003	0.18	0.99	0.31	91
10004	0.00	0.00	0.00	3
10005	0.00	0.00	0.00	40
10006	0.00	0.00	0.00	46
10007	0.00	0.00	0.00	1
10008	0.00	0.00	0.00	45
10009	0.00	0.00	0.00	3
10010	0.00	0.00	0.00	13
10011	0.00	0.00	0.00	3
10012	0.00	0.00	0.00	8
10013	0.00	0.00	0.00	27
10014	0.00	0.00	0.00	7
10015	0.00	0.00	0.00	27
10016	0.00	0.00	0.00	17
10017	0.00	0.00	0.00	13
10018	0.00	0.00	0.00	13
10019	0.00	0.00	0.00	11
accuracy			0.19	500
macro avg	0.03	0.05	0.02	500
weighted avg	0.10	0.19	0.07	500

Photos - Standard_DT_Topic.JPG

	precision	recall	f1-score	support
10000	0.77	0.35	0.48	66
10001	0.47	0.18	0.26	39
10002	0.56	0.56	0.56	27
10003	0.20	0.87	0.32	91
10004	0.00	0.00	0.00	3
10005	0.12	0.07	0.09	40
10006	0.00	0.00	0.00	46
10007	0.00	0.00	0.00	1
10008	0.00	0.00	0.00	45
10009	0.00	0.00	0.00	3
10010	0.43	0.23	0.30	13
10011	0.00	0.00	0.00	3
10012	0.00	0.00	0.00	8
10013	0.00	0.00	0.00	27
10014	0.00	0.00	0.00	7
10015	0.00	0.00	0.00	27
10016	0.00	0.00	0.00	17
10017	0.00	0.00	0.00	13
10018	0.00	0.00	0.00	13
10019	0.00	0.00	0.00	11
accuracy			0.26	500
macro avg	0.13	0.11	0.10	500
weighted avg	0.22	0.26	0.19	500

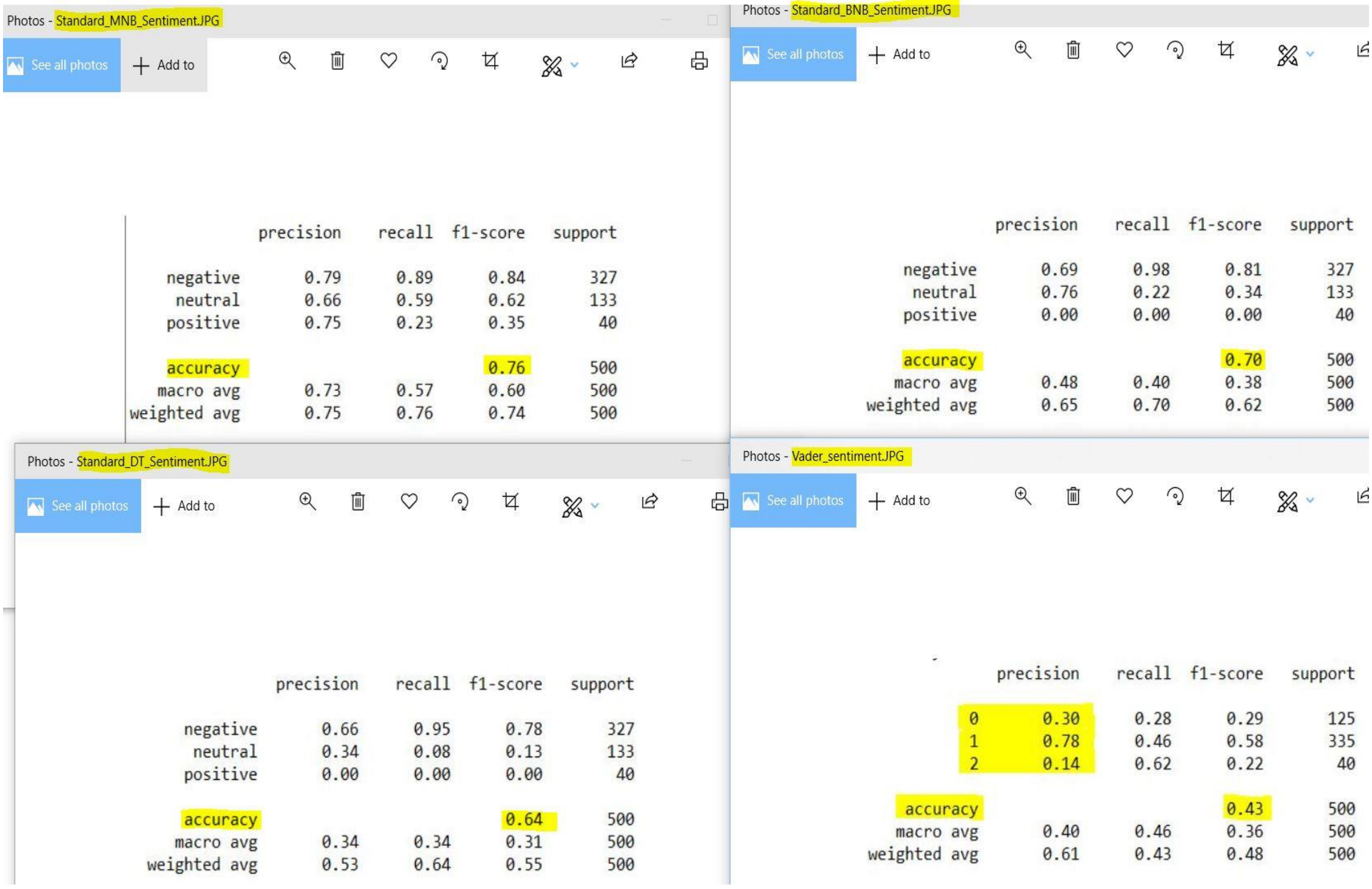
Photos - MajorityClass_Topic.JPG

	precision	recall	f1-score	support
10000	0.00	0.00	0.00	66
10001	0.00	0.00	0.00	39
10002	0.00	0.00	0.00	27
10003	0.18	1.00	0.31	91
10004	0.00	0.00	0.00	3
10005	0.00	0.00	0.00	40
10006	0.00	0.00	0.00	46
10007	0.00	0.00	0.00	1
10008	0.00	0.00	0.00	45
10009	0.00	0.00	0.00	3
10010	0.00	0.00	0.00	13
10011	0.00	0.00	0.00	3
10012	0.00	0.00	0.00	8
10013	0.00	0.00	0.00	27
10014	0.00	0.00	0.00	7
10015	0.00	0.00	0.00	27
10016	0.00	0.00	0.00	17
10017	0.00	0.00	0.00	13
10018	0.00	0.00	0.00	13
10019	0.00	0.00	0.00	11
accuracy			0.18	500
macro avg	0.01	0.05	0.02	500
weighted avg	0.03	0.18	0.06	500

Classification report--Top right: Standard Multinomial Topic, Top left: Standard Bernoulli Topic, Bottom right: Standard Decision Tree Topic, Bottom Left: Majority Class Topic

- In topic classification, it can be observed that Multinomial Classifier has the most accuracy with 33%. It is then followed by Decision Tree Classifier with 26%.
- Majority Classifier has the least accuracy than all the standard models. The precision factor is zero for almost all topics except topic 1003 which is 18% and that percent accounts to the whole model accuracy.
- From all the model classification report it is understandable that almost for all topics in the models, the prediction accuracy is not at all matching and Multinomial is the only model which considers almost every topic in the test data.

Standard models (Sentiment) when compared to **VADER Classification** is given below:



Photos - Standard_MNB_Sentiment.JPG

	precision	recall	f1-score	support
negative	0.79	0.89	0.84	327
neutral	0.66	0.59	0.62	133
positive	0.75	0.23	0.35	40
accuracy			0.76	500
macro avg	0.73	0.57	0.60	500
weighted avg	0.75	0.76	0.74	500

Photos - Standard_BNB_Sentiment.JPG

	precision	recall	f1-score	support
negative	0.69	0.98	0.81	327
neutral	0.76	0.22	0.34	133
positive	0.00	0.00	0.00	40
accuracy			0.70	500
macro avg	0.48	0.40	0.38	500
weighted avg	0.65	0.70	0.62	500

Photos - Standard_DT_Sentiment.JPG

	precision	recall	f1-score	support
negative	0.66	0.95	0.78	327
neutral	0.34	0.08	0.13	133
positive	0.00	0.00	0.00	40
accuracy			0.64	500
macro avg	0.34	0.34	0.31	500
weighted avg	0.53	0.64	0.55	500

Photos - Vader_sentiment.JPG

	precision	recall	f1-score	support
0	0.30	0.28	0.29	125
1	0.78	0.46	0.58	335
2	0.14	0.62	0.22	40
accuracy			0.43	500
macro avg	0.40	0.46	0.36	500
weighted avg	0.61	0.43	0.48	500

Classification report--Top right: Standard Multinomial Sentiment, Top left: Standard Bernoulli Sentiment, Bottom right: Standard Decision Tree Sentiment, Bottom Left: VADER Class Sentiment

- VADER (Valence Aware Dictionary for Sentiment Reasoning) is a lexical based approach. It has a dictionary of sentiment words with their corresponding intensities, which range between -4 to +4. (Positive: 2, Negative: 1, Neutral: 0)
- VADER not only classifies the review as positive or negative, but it also tells us the intensity of positiveness or negativeness in the text. But the problem with VADER is that it can only find sentiments of the words that are present in the lexicon. Any new slangs, if used in a review will not have any effect on classifying the reviews as those slang words are not part of the lexicon and hence do not have polarity.
- So, for this particular twitter data, there will always be new slangs which would not be taken into consideration for VADER classifier.
- In conclusion, VADER has the least accuracy than the other standard models due to its lexicon-based reviewing.

QUESTION 4: The difference between the standard models **before and after pre-processing (Stop word removal and Porter Stemming)** is:

Bernoulli Naïve Bayes (Sentiment):

Training data:

	precision	recall	f1-score	support
negative	0.78	1.00	0.88	967
neutral	0.95	0.60	0.74	420
positive	0.00	0.00	0.00	113
accuracy			0.81	1500
macro avg	0.58	0.53	0.54	1500
weighted avg	0.77	0.81	0.77	1500

Test data:

	precision	recall	f1-score	support
negative	0.69	0.98	0.81	327
neutral	0.76	0.22	0.34	133
positive	0.00	0.00	0.00	40
accuracy			0.70	500
macro avg	0.48	0.40	0.38	500
weighted avg	0.65	0.70	0.62	500

Training data:

	precision	recall	f1-score	support
negative	0.81	1.00	0.89	967
neutral	0.91	0.66	0.76	420
positive	0.00	0.00	0.00	113
accuracy			0.83	1500
macro avg	0.57	0.55	0.55	1500
weighted avg	0.77	0.83	0.79	1500

Test data:

	precision	recall	f1-score	support
negative	0.72	0.95	0.82	327
neutral	0.71	0.35	0.47	133
positive	0.00	0.00	0.00	40
accuracy			0.72	500
macro avg	0.48	0.43	0.43	500
weighted avg	0.66	0.72	0.66	500

*classification report--LEFT: BNB Sentiment **before** pre-processing| RIGHT: BNB Sentiment **after** pre-processing*

- In this model, the accuracy of training data and test data has increased by **two percent** after applying the stop word removal and porter stemming.
- Stemming tries to cut off details like exact form of a word and produce word bases as features for classification and stop words remove words that makes no sense to the computer. Due to these, the sentiment is being depicted with more accuracy for the training data which is then used for test data.

Bernoulli Naïve Bayes (Topic):

Training data:

	precision	recall	f1-score	support
10000	0.91	0.38	0.53	178
10001	0.00	0.00	0.00	101
10002	0.00	0.00	0.00	103
10003	0.19	1.00	0.32	267
10004	0.00	0.00	0.00	14
10005	1.00	0.01	0.03	154
10006	1.00	0.01	0.01	143
10007	0.00	0.00	0.00	6
10008	0.00	0.00	0.00	118
10009	0.00	0.00	0.00	13
10010	0.00	0.00	0.00	43
10011	0.00	0.00	0.00	10
10012	0.00	0.00	0.00	17
10013	0.00	0.00	0.00	77
10014	0.00	0.00	0.00	22
10015	0.00	0.00	0.00	92
10016	0.00	0.00	0.00	42
10017	0.00	0.00	0.00	34
10018	0.00	0.00	0.00	25
10019	0.00	0.00	0.00	41
accuracy			0.22	1500
macro avg	0.15	0.07	0.04	1500
weighted avg	0.34	0.22	0.12	1500

Test data:

	precision	recall	f1-score	support
10000	0.50	0.08	0.13	66
10001	0.00	0.00	0.00	39
10002	0.00	0.00	0.00	27
10003	0.18	0.99	0.31	91
10004	0.00	0.00	0.00	3
10005	0.00	0.00	0.00	40
10006	0.00	0.00	0.00	46
10007	0.00	0.00	0.00	1
10008	0.00	0.00	0.00	45
10009	0.00	0.00	0.00	3
10010	0.00	0.00	0.00	13
10011	0.00	0.00	0.00	3
10012	0.00	0.00	0.00	8
10013	0.00	0.00	0.00	27
10014	0.00	0.00	0.00	7
10015	0.00	0.00	0.00	27
10016	0.00	0.00	0.00	17
10017	0.00	0.00	0.00	13
10018	0.00	0.00	0.00	13
10019	0.00	0.00	0.00	11
accuracy			0.19	500
macro avg	0.03	0.05	0.02	500
weighted avg	0.10	0.19	0.07	500

*BNB Topic classification report for training and test data **before** pre processing*

Training data:

	precision	recall	f1-score	support
10000	0.85	0.65	0.74	178
10001	0.00	0.00	0.00	101
10002	1.00	0.01	0.02	103
10003	0.20	1.00	0.34	267
10004	0.00	0.00	0.00	14
10005	0.79	0.15	0.25	154
10006	1.00	0.08	0.14	143
10007	0.00	0.00	0.00	6
10008	1.00	0.07	0.13	118
10009	0.00	0.00	0.00	13
10010	0.00	0.00	0.00	43
10011	0.00	0.00	0.00	10
10012	0.00	0.00	0.00	17
10013	0.00	0.00	0.00	77
10014	0.00	0.00	0.00	22
10015	0.00	0.00	0.00	92
10016	0.00	0.00	0.00	42
10017	0.00	0.00	0.00	34
10018	0.00	0.00	0.00	25
10019	0.00	0.00	0.00	41
accuracy			0.28	1500
macro avg	0.24	0.10	0.08	1500
weighted avg	0.46	0.28	0.20	1500

Test data:

	precision	recall	f1-score	support
10000	0.63	0.26	0.37	66
10001	0.00	0.00	0.00	39
10002	0.00	0.00	0.00	27
10003	0.19	0.98	0.32	91
10004	0.00	0.00	0.00	3
10005	0.20	0.03	0.04	40
10006	0.00	0.00	0.00	46
10007	0.00	0.00	0.00	1
10008	0.00	0.00	0.00	45
10009	0.00	0.00	0.00	3
10010	0.00	0.00	0.00	13
10011	0.00	0.00	0.00	3
10012	0.00	0.00	0.00	8
10013	0.00	0.00	0.00	27
10014	0.00	0.00	0.00	7
10015	0.00	0.00	0.00	27
10016	0.00	0.00	0.00	17
10017	0.00	0.00	0.00	13
10018	0.00	0.00	0.00	13
10019	0.00	0.00	0.00	11
accuracy			0.21	500
macro avg	0.05	0.06	0.04	500
weighted avg	0.13	0.21	0.11	500

*BNB Topic classification report for training and test data **after** pre processing*

- In this model, the accuracy of training data has increased by **six percent** after applying the stop word removal and porter stemming.
- The accuracy of test data has increased by **three percent** after pre-processing.
- After pre-processing it can be observed in the tables that, precision for each topic is less for training data but increases in the test data table.
- For example, take the topic 10000. For the training data before pre-processing, the precision is, 91%. After pre-processing is 85%. Consider the same for test data. The precision before pre-processing is 50%. After pre-processing is 63%. So, it follows that trend for this model.

Multinomial Naïve Bayes (Sentiment):

Training data:

	precision	recall	f1-score	support
negative	0.93	0.99	0.96	967
neutral	0.92	0.91	0.92	420
positive	1.00	0.43	0.60	113
accuracy			0.93	1500
macro avg	0.95	0.78	0.83	1500
weighted avg	0.93	0.93	0.92	1500

Test data:

	precision	recall	f1-score	support
negative	0.79	0.89	0.84	327
neutral	0.66	0.59	0.62	133
positive	0.75	0.23	0.35	40
accuracy			0.76	500
macro avg	0.73	0.57	0.60	500
weighted avg	0.75	0.76	0.74	500

Training data:

	precision	recall	f1-score	support
negative	0.91	0.98	0.94	967
neutral	0.91	0.85	0.88	420
positive	0.98	0.52	0.68	113
accuracy			0.91	1500
macro avg	0.93	0.79	0.84	1500
weighted avg	0.91	0.91	0.91	1500

Test data:

	precision	recall	f1-score	support
negative	0.80	0.87	0.83	327
neutral	0.60	0.57	0.59	133
positive	0.43	0.23	0.30	40
accuracy			0.74	500
macro avg	0.61	0.55	0.57	500
weighted avg	0.72	0.74	0.72	500

*classification report--LEFT: MNB Sentiment **before** pre-processing| RIGHT: MNB Sentiment **after** pre-processing*

- For this model, stemming and stop word removal is decreasing the accuracy for both the train data and test data.
- Stemming tries to cut off details like exact form of a word and produce word bases as features for classification. Sometimes, however, these details play important role by themselves. For example, phrase "runs today" may refer to a runner, while "long running"

may be about phone battery lifetime. In this case stemming makes classification worse, not better. So, we are decreasing the number of distinctive tokens per class, which may have a negative effect on predictive power. This is the same case with stop words. That is why maybe the accuracy is going down.

Multinomial Naïve Bayes (Topic):

Training data:

	precision	recall	f1-score	support
10000	0.92	0.93	0.92	178
10001	0.85	0.61	0.71	101
10002	0.94	0.65	0.77	103
10003	0.47	0.98	0.64	267
10004	1.00	0.14	0.25	14
10005	0.82	0.84	0.83	154
10006	0.82	0.79	0.81	143
10007	1.00	0.33	0.50	6
10008	0.85	0.84	0.84	118
10009	0.00	0.00	0.00	13
10010	0.83	0.23	0.36	43
10011	0.00	0.00	0.00	10
10012	1.00	0.06	0.11	17
10013	0.87	0.75	0.81	77
10014	1.00	0.23	0.37	22
10015	0.91	0.82	0.86	92
10016	1.00	0.31	0.47	42
10017	1.00	0.18	0.30	34
10018	1.00	0.36	0.53	25
10019	1.00	0.27	0.42	41
accuracy			0.73	1500
macro avg	0.81	0.47	0.53	1500
weighted avg	0.80	0.73	0.71	1500

Test data:

	precision	recall	f1-score	support
10000	0.62	0.56	0.59	66
10001	0.12	0.03	0.04	39
10002	0.36	0.15	0.21	27
10003	0.25	0.81	0.38	91
10004	0.00	0.00	0.00	3
10005	0.32	0.30	0.31	40
10006	0.44	0.24	0.31	46
10007	0.00	0.00	0.00	1
10008	0.58	0.31	0.41	45
10009	0.00	0.00	0.00	3
10010	0.00	0.00	0.00	13
10011	0.00	0.00	0.00	3
10012	0.00	0.00	0.00	8
10013	0.30	0.11	0.16	27
10014	0.00	0.00	0.00	7
10015	0.47	0.33	0.39	27
10016	0.00	0.00	0.00	17
10017	0.00	0.00	0.00	13
10018	0.00	0.00	0.00	13
10019	0.00	0.00	0.00	11
accuracy			0.33	500
macro avg	0.17	0.14	0.14	500
weighted avg	0.32	0.33	0.28	500

MNB Topic classification report for training and test data ***before*** *pre processing*

Training data:

	precision	recall	f1-score	support
10000	0.89	0.93	0.91	178
10001	0.88	0.60	0.72	101
10002	0.86	0.76	0.80	103
10003	0.56	0.93	0.70	267
10004	1.00	0.21	0.35	14
10005	0.77	0.89	0.82	154
10006	0.86	0.83	0.84	143
10007	1.00	0.17	0.29	6
10008	0.83	0.92	0.87	118
10009	0.00	0.00	0.00	13
10010	0.89	0.37	0.52	43
10011	1.00	0.10	0.18	10
10012	1.00	0.18	0.30	17
10013	0.89	0.88	0.89	77
10014	1.00	0.50	0.67	22
10015	0.88	0.87	0.87	92
10016	0.94	0.38	0.54	42
10017	1.00	0.32	0.49	34
10018	1.00	0.64	0.78	25
10019	0.94	0.41	0.58	41
accuracy			0.77	1500
macro avg	0.86	0.55	0.61	1500
weighted avg	0.81	0.77	0.76	1500

Test data:

	precision	recall	f1-score	support
10000	0.61	0.62	0.62	66
10001	0.18	0.05	0.08	39
10002	0.39	0.26	0.31	27
10003	0.31	0.78	0.45	91
10004	0.00	0.00	0.00	3
10005	0.42	0.53	0.47	40
10006	0.49	0.39	0.43	46
10007	0.00	0.00	0.00	1
10008	0.58	0.49	0.53	45
10009	0.00	0.00	0.00	3
10010	0.00	0.00	0.00	13
10011	0.00	0.00	0.00	3
10012	0.00	0.00	0.00	8
10013	0.41	0.26	0.32	27
10014	0.00	0.00	0.00	7
10015	0.62	0.56	0.59	27
10016	0.25	0.06	0.10	17
10017	0.00	0.00	0.00	13
10018	1.00	0.08	0.14	13
10019	0.00	0.00	0.00	11
accuracy			0.41	500
macro avg	0.26	0.20	0.20	500
weighted avg	0.39	0.41	0.37	500

MNB Topic classification report for training and test data ***after*** *pre processing*

- The accuracy of the training data is increasing by **four percent** and by **eight percent** for test data after pre-processing.
- As there are many numbers of topics to classify it would be advantageous to have more stemming and removing of stop words to help the model to train properly.
- So overall, the increase of precision is understandable.

Decision Tree Classifier (Sentiment):

Training data:

	precision	recall	f1-score	support
negative	0.74	0.90	0.81	967
neutral	0.53	0.42	0.47	420
positive	0.00	0.00	0.00	113
accuracy			0.70	1500
macro avg	0.43	0.44	0.43	1500
weighted avg	0.63	0.70	0.65	1500

Test data:

	precision	recall	f1-score	support
negative	0.72	0.87	0.79	327
neutral	0.49	0.38	0.43	133
positive	0.00	0.00	0.00	40
accuracy			0.67	500
macro avg	0.40	0.42	0.40	500
weighted avg	0.60	0.67	0.63	500

Training data:

	precision	recall	f1-score	support
negative	0.72	0.92	0.81	967
neutral	0.49	0.30	0.37	420
positive	0.00	0.00	0.00	113
accuracy			0.68	1500
macro avg	0.40	0.41	0.39	1500
weighted avg	0.60	0.68	0.62	1500

Test data:

	precision	recall	f1-score	support
negative	0.69	0.91	0.79	327
neutral	0.42	0.23	0.29	133
positive	0.00	0.00	0.00	40
accuracy			0.65	500
macro avg	0.37	0.38	0.36	500
weighted avg	0.57	0.65	0.59	500

*classification report--LEFT: DT Sentiment **before** pre-processing| RIGHT: DT Sentiment **after** pre-processing*

- For this model, stemming and stop word removal is decreasing the accuracy for both the train data and test data.
- Stemming tries to cut off details like exact form of a word and produce word bases as features for classification. Also, the stop words are being removed and there might not be much featured words for the model to be trained on. The decision tree maybe doesn't have enough data to work on and predicting incorrect sentiment for the test data and also maybe training data.

Decision Tree Classifier (Topic):

Training data:

	precision	recall	f1-score	support
10000	0.78	0.30	0.43	178
10001	0.37	0.19	0.25	101
10002	0.48	0.31	0.38	103
10003	0.19	0.88	0.31	267
10004	0.00	0.00	0.00	14
10005	0.25	0.10	0.15	154
10006	0.00	0.00	0.00	143
10007	0.00	0.00	0.00	6
10008	0.00	0.00	0.00	118
10009	0.00	0.00	0.00	13
10010	0.38	0.21	0.27	43
10011	0.00	0.00	0.00	10
10012	0.00	0.00	0.00	17
10013	0.00	0.00	0.00	77
10014	0.00	0.00	0.00	22
10015	0.00	0.00	0.00	92
10016	0.00	0.00	0.00	42
10017	0.00	0.00	0.00	34
10018	0.00	0.00	0.00	25
10019	0.00	0.00	0.00	41
accuracy			0.24	1500
macro avg	0.12	0.10	0.09	1500
weighted avg	0.22	0.24	0.17	1500

Test data:

	precision	recall	f1-score	support
10000	0.77	0.35	0.48	66
10001	0.47	0.18	0.26	39
10002	0.56	0.56	0.56	27
10003	0.20	0.87	0.32	91
10004	0.00	0.00	0.00	3
10005	0.12	0.07	0.09	40
10006	0.00	0.00	0.00	46
10007	0.00	0.00	0.00	1
10008	0.00	0.00	0.00	45
10009	0.00	0.00	0.00	3
10010	0.43	0.23	0.30	13
10011	0.00	0.00	0.00	3
10012	0.00	0.00	0.00	8
10013	0.00	0.00	0.00	27
10014	0.00	0.00	0.00	7
10015	0.00	0.00	0.00	27
10016	0.00	0.00	0.00	17
10017	0.00	0.00	0.00	13
10018	0.00	0.00	0.00	13
10019	0.00	0.00	0.00	11
accuracy			0.26	500
macro avg	0.13	0.11	0.10	500
weighted avg	0.22	0.26	0.19	500

*DT Topic classification report for training and test data **before** pre processing*

Training data:

	precision	recall	f1-score	support
10000	0.21	0.63	0.31	178
10001	0.38	0.24	0.29	101
10002	0.41	0.46	0.43	103
10003	0.25	0.45	0.32	267
10004	0.00	0.00	0.00	14
10005	0.75	0.45	0.57	154
10006	0.29	0.04	0.07	143
10007	0.00	0.00	0.00	6
10008	0.38	0.20	0.26	118
10009	0.00	0.00	0.00	13
10010	0.46	0.40	0.42	43
10011	0.00	0.00	0.00	10
10012	0.00	0.00	0.00	17
10013	0.19	0.06	0.10	77
10014	0.00	0.00	0.00	22
10015	0.77	0.53	0.63	92
10016	0.00	0.00	0.00	42
10017	0.00	0.00	0.00	34
10018	0.00	0.00	0.00	25
10019	0.00	0.00	0.00	41
accuracy			0.32	1500
macro avg	0.20	0.17	0.17	1500
weighted avg	0.33	0.32	0.29	1500

Test data:

	precision	recall	f1-score	support
10000	0.24	0.73	0.36	66
10001	0.35	0.18	0.24	39
10002	0.31	0.44	0.36	27
10003	0.26	0.41	0.32	91
10004	0.00	0.00	0.00	3
10005	0.71	0.42	0.53	40
10006	0.11	0.02	0.04	46
10007	0.00	0.00	0.00	1
10008	0.25	0.13	0.17	45
10009	0.00	0.00	0.00	3
10010	0.31	0.31	0.31	13
10011	0.00	0.00	0.00	3
10012	0.00	0.00	0.00	8
10013	0.10	0.04	0.05	27
10014	0.00	0.00	0.00	7
10015	0.71	0.56	0.63	27
10016	0.00	0.00	0.00	17
10017	0.00	0.00	0.00	13
10018	0.00	0.00	0.00	13
10019	0.00	0.00	0.00	11
accuracy			0.30	500
macro avg	0.17	0.16	0.15	500
weighted avg	0.26	0.30	0.25	500

*DT Topic classification report for training and test data **after** pre processing*

- The accuracy of the training data is increasing by **eight percent** and by **four percent** for test data after pre-processing.
- As there are many numbers of topics to classify it would be advantageous to have more stemming and removing of stop words to help the model to train properly.
- The decision tree is able to construct a model accordingly and increase its accuracy.

QUESTION 5: The comparison of the standard model's sentiment classification reports before and **after removing the 'neutral' tweets with max_features=200** is:

Bernoulli Naïve Bayes (Sentiment):

	precision	recall	f1-score	support
negative	0.80	0.87	0.83	967
neutral	0.63	0.56	0.59	420
positive	0.68	0.43	0.53	113
accuracy			0.75	1500
macro avg	0.70	0.62	0.65	1500
weighted avg	0.74	0.75	0.74	1500

	precision	recall	f1-score	support
negative	0.76	0.83	0.79	327
neutral	0.53	0.49	0.51	133
positive	0.43	0.23	0.30	40
accuracy			0.69	500
macro avg	0.57	0.51	0.53	500
weighted avg	0.67	0.69	0.68	500

BNB Sentiment with max_features=200 for training and test data

Training data:

	precision	recall	f1-score	support
negative	0.94	0.97	0.96	970
positive	0.67	0.49	0.56	115
accuracy			0.92	1085
macro avg	0.80	0.73	0.76	1085
weighted avg	0.91	0.92	0.91	1085

Test data:

	precision	recall	f1-score	support
negative	0.94	0.98	0.96	324
positive	0.70	0.50	0.58	38
accuracy			0.93	362
macro avg	0.82	0.74	0.77	362
weighted avg	0.92	0.93	0.92	362

BNB Sentiment after removal of 'neutral' tweets and max_features =200 for training and test data

- The test data accuracy is going from 63% to 93% which is a significant increase. The training data accuracy is increasing from 75% to 93% which is also a significant increase.
- Even the precision of negative and positive tweets is taking an exponential rise.
- The test data accuracy after removing ‘neutral’ tweets is higher than the training data accuracy which is a great effect to the model. It has a rise from 92% to 93%.

Multinomial Naïve Bayes (Sentiment):

	precision	recall	f1-score	support
negative	0.81	0.89	0.85	967
neutral	0.66	0.57	0.61	420
positive	0.73	0.47	0.57	113
accuracy			0.77	1500
macro avg	0.73	0.64	0.68	1500
weighted avg	0.76	0.77	0.76	1500

	precision	recall	f1-score	support
negative	0.75	0.80	0.77	327
neutral	0.50	0.45	0.47	133
positive	0.37	0.25	0.30	40
accuracy			0.67	500
macro avg	0.54	0.50	0.52	500
weighted avg	0.65	0.67	0.66	500

MNB Sentiment with max_features=200 for training and test data

Training data:

	precision	recall	f1-score	support
negative	0.95	0.97	0.96	970
positive	0.67	0.53	0.59	115
accuracy			0.92	1085
macro avg	0.81	0.75	0.77	1085
weighted avg	0.92	0.92	0.92	1085

Test data:

	precision	recall	f1-score	support
negative	0.94	0.98	0.96	324
positive	0.71	0.45	0.55	38
accuracy			0.92	362
macro avg	0.82	0.71	0.75	362
weighted avg	0.91	0.92	0.91	362

MNB Sentiment after removal of ‘neutral’ tweets and max_features =200 for training and test data

- The test data accuracy is going from 67% to 92% which is a significant increase. The training data accuracy is increasing from 77% to 92% which is also a significant increase.
- The positive tweets accuracy is going down from an accuracy of 73% to 67% and the negative tweets accuracy is taking a rise from 81% to 94% for training data.
- The positive tweets accuracy is increasing from a precision of 37% to 71% and the negative tweets accuracy is taking a rise from 75% to 94% for test data.

Decision Tree Classifier (Sentiment):

	precision	recall	f1-score	support
negative	0.67	0.96	0.79	967
neutral	0.50	0.14	0.22	420
positive	0.00	0.00	0.00	113
accuracy			0.66	1500
macro avg	0.39	0.37	0.34	1500
weighted avg	0.57	0.66	0.57	1500

	precision	recall	f1-score	support
negative	0.66	0.95	0.78	327
neutral	0.34	0.08	0.13	133
positive	0.00	0.00	0.00	40
accuracy			0.64	500
macro avg	0.34	0.34	0.31	500
weighted avg	0.53	0.64	0.55	500

DT Sentiment with max_features=200 for training and test data

Training data:

	precision	recall	f1-score	support
negative	0.89	1.00	0.94	970
positive	0.00	0.00	0.00	115
accuracy			0.89	1085
macro avg	0.45	0.50	0.47	1085
weighted avg	0.80	0.89	0.84	1085

Test data:

	precision	recall	f1-score	support
negative	0.90	1.00	0.94	324
positive	0.00	0.00	0.00	38
accuracy			0.90	362
macro avg	0.45	0.50	0.47	362
weighted avg	0.80	0.90	0.85	362

DT Sentiment after removal of 'neutral' tweets and max_features =200 for training and test data

- The positive tweets accuracy is not having any change and the negative tweets accuracy is taking a rise from 67% to 89% for training data.
- The positive tweets accuracy is constant and the negative tweets accuracy is taking a rise from 66% to 90% for test data.

The comparison of the VADER model sentiment classification reports before and **after removing the 'neutral' tweets with VADER Sentiment analysis** is:

	precision	recall	f1-score	support
0	0.30	0.28	0.29	125
1	0.78	0.46	0.58	335
2	0.14	0.62	0.22	40
accuracy			0.43	500
macro avg	0.40	0.46	0.36	500
weighted avg	0.61	0.43	0.48	500

	precision	recall	f1-score	support
0	0.00	0.00	0.00	0
1	0.95	0.47	0.63	327
2	0.18	0.63	0.28	35
accuracy			0.49	362
macro avg	0.38	0.37	0.30	362
weighted avg	0.88	0.49	0.60	362

*VADER Sentiment **before** and **after** removal of 'neutral' tweets*

- The accuracy of the model seems to have a significant increase of **six percent**, from 43% to 49%.
- The positive tweets (present as 2) had a precision of 14% before and after the removal of neutral the precision has an increase up to 18%.
- The negative tweets (present as 1) had a precision of 78% before and it took an exponential increase up to 95% after the removal of neutral tweets.

The comparison of **Majority Classifier** before and **after the removal of 'neutral'** tweets is:

	precision	recall	f1-score	support
negative	0.65	1.00	0.79	327
neutral	0.00	0.00	0.00	133
positive	0.00	0.00	0.00	40
accuracy			0.65	500
macro avg	0.22	0.33	0.26	500
weighted avg	0.43	0.65	0.52	500

	precision	recall	f1-score	support
negative	0.90	1.00	0.94	324
positive	0.00	0.00	0.00	38
accuracy			0.90	362
macro avg	0.45	0.50	0.47	362
weighted avg	0.80	0.90	0.85	362

*Majority Classifier Sentiment **before** and **after** removal of 'neutral' tweets*

- The accuracy of the model has an exponential rise from 65% to 90%.
- No positive tweets are being predicted for both before and after the 'neutral' tweets drop. Even the neutral tweets are also having a precision of zero before the drop.
- But, the accuracy of negative tweets had an increase if there is no neutral tweet consideration.

The comparison of **Standard models** before and **after the removal of 'neutral'** tweets and also **pre-processing the tweets by stemming and stop word removal** is:

Bernoulli Naïve Bayes (Sentiment):

Training data:

	precision	recall	f1-score	support
negative	0.81	1.00	0.89	967
neutral	0.91	0.66	0.76	420
positive	0.00	0.00	0.00	113
accuracy			0.83	1500
macro avg	0.57	0.55	0.55	1500
weighted avg	0.77	0.83	0.79	1500

Test data:

	precision	recall	f1-score	support
negative	0.72	0.95	0.82	327
neutral	0.71	0.35	0.47	133
positive	0.00	0.00	0.00	40
accuracy			0.72	500
macro avg	0.48	0.43	0.43	500
weighted avg	0.66	0.72	0.66	500

Training data:

	precision	recall	f1-score	support
negative	0.90	1.00	0.95	970
positive	1.00	0.03	0.07	115
accuracy			0.90	1085
macro avg	0.95	0.52	0.51	1085
weighted avg	0.91	0.90	0.85	1085

Test data:

	precision	recall	f1-score	support
negative	0.90	1.00	0.95	324
positive	1.00	0.03	0.05	38
accuracy			0.90	362
macro avg	0.95	0.51	0.50	362
weighted avg	0.91	0.90	0.85	362

*Classification report—BNB Sentiment **after** removal of 'neutral' tweets and **after** pre-processing*

- The accuracy of the training data has increased from 83% to 90%. The positive tweets have the precision of 100% after the removal of 'neutral' tweets where as before the removal, they had a precision of 0.
- The accuracy of the test data also increased from 72% to 90%. Similar to the training data, the precision of positive tweets is 100% after the removal of 'neutral' tweets.

Multinomial Naïve Bayes (Sentiment):

Training data:

	precision	recall	f1-score	support
negative	0.91	0.98	0.94	967
neutral	0.91	0.85	0.88	420
positive	0.98	0.52	0.68	113
accuracy			0.91	1500
macro avg	0.93	0.79	0.84	1500
weighted avg	0.91	0.91	0.91	1500

Test data:

	precision	recall	f1-score	support
negative	0.80	0.87	0.83	327
neutral	0.60	0.57	0.59	133
positive	0.43	0.23	0.30	40
accuracy			0.74	500
macro avg	0.61	0.55	0.57	500
weighted avg	0.72	0.74	0.72	500

Training data:

	precision	recall	f1-score	support
negative	0.98	1.00	0.99	970
positive	0.98	0.83	0.90	115
accuracy			0.98	1085
macro avg	0.98	0.91	0.94	1085
weighted avg	0.98	0.98	0.98	1085

Test data:

	precision	recall	f1-score	support
negative	0.94	0.93	0.93	324
positive	0.44	0.50	0.47	38
accuracy			0.88	362
macro avg	0.69	0.71	0.70	362
weighted avg	0.89	0.88	0.88	362

*Classification report—MNB Sentiment **after** removal of 'neutral' tweets and **after** pre-processing*

- The accuracy of the training data has increased from 90% to 98%. The positive tweets precision remains the same after the removal of 'neutral' tweets whereas the negative tweets precision increased from 91% to 98%.
- The accuracy of the test data also increased from 72% to 88%. Similar to the training data, the precision of positive tweets is 44% after the removal of 'neutral' tweets, slight increase. Whereas the negative tweets precision increased 80% to 94%.

Decision Tree Classifier (Sentiment):

Training data:

	precision	recall	f1-score	support
negative	0.72	0.92	0.81	967
neutral	0.49	0.30	0.37	420
positive	0.00	0.00	0.00	113
accuracy			0.68	1500
macro avg	0.40	0.41	0.39	1500
weighted avg	0.60	0.68	0.62	1500

Test data:

	precision	recall	f1-score	support
negative	0.69	0.91	0.79	327
neutral	0.42	0.23	0.29	133
positive	0.00	0.00	0.00	40
accuracy			0.65	500
macro avg	0.37	0.38	0.36	500
weighted avg	0.57	0.65	0.59	500

Training data:

	precision	recall	f1-score	support
negative	0.89	1.00	0.94	970
positive	0.00	0.00	0.00	115
accuracy			0.89	1085
macro avg	0.45	0.50	0.47	1085
weighted avg	0.80	0.89	0.84	1085

Test data:

	precision	recall	f1-score	support
negative	0.90	1.00	0.94	324
positive	0.00	0.00	0.00	38
accuracy			0.90	362
macro avg	0.45	0.50	0.47	362
weighted avg	0.80	0.90	0.85	362

*Classification report—DT Sentiment **after** removal of 'neutral' tweets and **after** pre-processing*

- The accuracy of the training data has increased from 68% to 89%. The positive tweets precision remains the same after the removal of 'neutral' tweets whereas the negative tweets precision increased from 72% to 89%.
- The accuracy of the test data also increased from 65% to 90%. Similar to the training data, the precision of positive is constant after the removal of 'neutral' tweets. Whereas the negative tweets precision increased 69% to 90%.

QUESTION 6: The best method for **Sentiment Analysis** for me is my model – which is taking the best accurate model out of the standard model and applying some pre-processing techniques like 'PorterStemming' and 'max_features' to achieve an **accuracy of 77%.**

- Initially, the best standard model (with more accuracy) that I got was **'Multinomial Naïve Bayes' with an accuracy of 75.8%.**
- The standard model only had 'lowercase=False' and a 'token_pattern' set to read all characters which would contain @, $, %, #, _ and numbers and words. The URL's was removed and replaced with space.
- So, taking that model as a reference, I tried to do some pre-processing techniques like 'Lemmatization' using the 'WordnetLemmetizer'. The accuracy went down to 69% just by doing that. So, I combined it with 'Stemming' using the 'PorterStemmer'. The accuracy went down further to 67%.
- As we are analysing the sentiment, maybe lemmatizing wouldn't be more appropriate as it would prune the entire dataset and the model might not have more data to train itself. So, I replaced Lemmatizing with Stemming.
- Just by doing stemming, I could get the accuracy up to 74%. I combined that with various other features like, max_features which I initially set to 500. But it was not helping the accuracy at all. So, I used a bigger figure like 2500. It went up to 77%.
- I also combined them with features like ng_gram, min_df, max_df. They just decreased the accuracy than what the standard model is.

The classification report of My_Model is:

```
Accuracy Score: 0.77
Traning data:
              precision    recall  f1-score   support

    negative       0.91      0.97      0.94       967
     neutral       0.88      0.84      0.86       420
    positive       0.97      0.58      0.73       113

    accuracy                           0.90      1500
   macro avg       0.92      0.80      0.84      1500
weighted avg       0.90      0.90      0.90      1500

Test data:
              precision    recall  f1-score   support

    negative       0.81      0.89      0.85       327
     neutral       0.66      0.61      0.64       133
    positive       0.70      0.35      0.47        40

    accuracy                           0.77       500
   macro avg       0.72      0.62      0.65       500
weighted avg       0.76      0.77      0.76       500
```

My_Model Sentiment Classification report

- The training data precisions are not as good as the standard model though. But in the test data the precision of the negative tweets is 81% which is three percent more than the standard model whose negative precision for test data is 79%.

- As the negative tweets are higher in the dataset, the model is well trained when it encounters a 'negative' sentiment tweet. It can be analysed more.

The comparison of classification reports for the standard Multinomial Classification vs My_Model is:

Accuracy score: 0.758

Training data:

	precision	recall	f1-score	support
negative	0.93	0.99	0.96	967
neutral	0.92	0.91	0.92	420
positive	1.00	0.43	0.60	113
accuracy			0.93	1500
macro avg	0.95	0.78	0.83	1500
weighted avg	0.93	0.93	0.92	1500

Test data:

	precision	recall	f1-score	support
negative	0.79	0.89	0.84	327
neutral	0.66	0.59	0.62	133
positive	0.75	0.23	0.35	40
accuracy			0.76	500
macro avg	0.73	0.57	0.60	500
weighted avg	0.75	0.76	0.74	500

Accuracy Score: 0.77

Traning data:

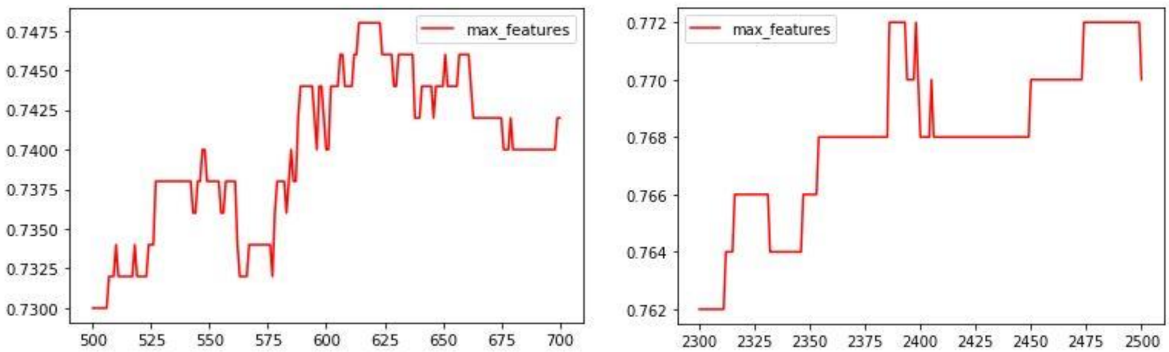
	precision	recall	f1-score	support
negative	0.91	0.97	0.94	967
neutral	0.88	0.84	0.86	420
positive	0.97	0.58	0.73	113
accuracy			0.90	1500
macro avg	0.92	0.80	0.84	1500
weighted avg	0.90	0.90	0.90	1500

Test data:

	precision	recall	f1-score	support
negative	0.81	0.89	0.85	327
neutral	0.66	0.61	0.64	133
positive	0.70	0.35	0.47	40
accuracy			0.77	500
macro avg	0.72	0.62	0.65	500
weighted avg	0.76	0.77	0.76	500

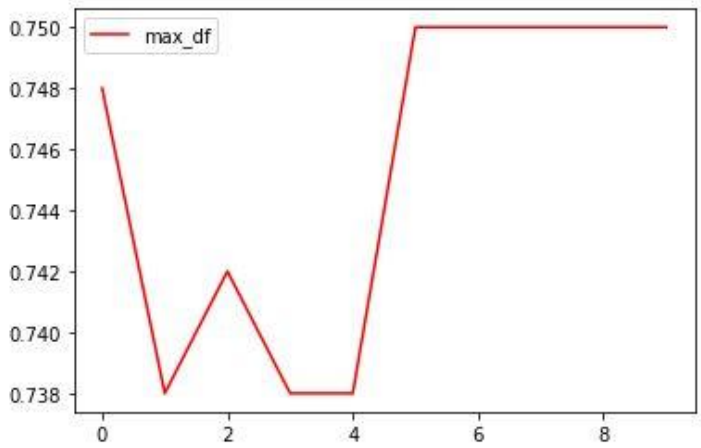
Multinomial Naïve Bayes Classification vs My_Model for Sentiment Analysis

The experimental data statistics are shown below:



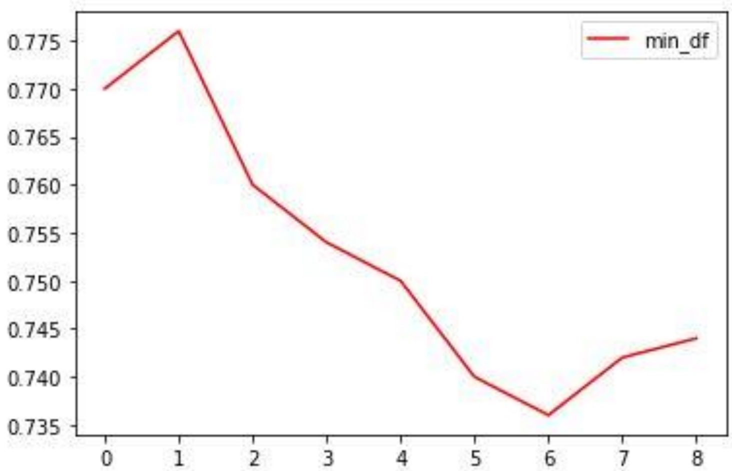
LEFT: max_features ranging from 500 to 700, RIGHT: max_features ranging from 2300-2500

- This graph shows the distribution of the accuracy of My_Model when the max_features is changed from 500-700.
- The max accuracy that could be obtained was 74.75% according to the graph.
- In the right graph, the value around 2500 seems to stabilise and also giving us more accuracy for the model.
- Then the graph seems to be taking a downhill again. So, 2500 seems to be the right fit for this particular dataset for sentiment analysis.



Varying the max_df from 0.1 to 1.0

- This graph depicts the accuracy of the model whilst varying the 'max_df' value from 0.1-1.0.
- It starts from 74.8 accuracy and then plummets at 0.1. It takes a slow rise there, but the graph stabilises at 75% accuracy.
- So, using max_df is not helping 'My_Model'.



Varying the min_df from 1 to 9

- This graph depicts the accuracy of the model whilst varying the 'min_df' value from 1-9.
- The graph initially starts with a 77% percent accuracy but gradually declines and goes to 73% percent at 'min_df=6'.
- If min_df is clubbed with max_features, it is decreasing the accuracy of the model. So, it is better to choose one. I choose 'max_features=2500'.

So, after applying **Porter Stemming** and **'max_features=2500'** and **'lowercase=True'**, My_Model is predicting with an accuracy of 77%.

The best method for **Topic Analysis** for me is my model – which is taking the best accurate model out of the standard model and applying some pre-processing techniques like 'PorterStemming' and 'max_features' and 'stop_words' to achieve an **accuracy of 43%.**

- Initially, the best standard model (with more accuracy) that I got was **'Multinomial Naïve Bayes' with an accuracy of 33%.**
- The standard model only had 'lowercase=False' and a 'token_pattern' set to read all characters which would contain @, $, %, #, _ and numbers and words. The URL's was removed and replaced with space.
- So, taking that model as a reference, I tried to do some pre-processing techniques like 'Lemmatization' using the 'WordnetLemmetizer'. The accuracy went down to 24% just by doing that. So, I combined it with 'Stemming' using the 'PorterStemmer'. The accuracy went down further to 32%.
- So, overall, I applied 'Porter Stemming', 'max_features=2500' and 'stop_words=stop_words('english')' to improve the standard model by almost 10%.

The classification report of My_Model_Topic is:

Accuracy Score: 0.432
Traning data:

	precision	recall	f1-score	support
10000	0.90	0.92	0.91	178
10001	0.84	0.65	0.73	101
10002	0.81	0.79	0.80	103
10003	0.59	0.87	0.71	267
10004	1.00	0.14	0.25	14
10005	0.72	0.88	0.80	154
10006	0.82	0.82	0.82	143
10007	1.00	0.17	0.29	6
10008	0.83	0.90	0.87	118
10009	1.00	0.15	0.27	13
10010	0.77	0.40	0.52	43
10011	0.00	0.00	0.00	10
10012	0.83	0.29	0.43	17
10013	0.88	0.84	0.86	77
10014	1.00	0.55	0.71	22
10015	0.83	0.91	0.87	92
10016	0.94	0.38	0.54	42
10017	1.00	0.26	0.42	34
10018	0.79	0.76	0.78	25
10019	0.95	0.46	0.62	41
accuracy			0.77	1500
macro avg	0.83	0.56	0.61	1500
weighted avg	0.79	0.77	0.75	1500

Testing data:

	precision	recall	f1-score	support
10000	0.59	0.62	0.60	66
10001	0.35	0.15	0.21	39
10002	0.35	0.30	0.32	27
10003	0.34	0.71	0.46	91
10004	0.00	0.00	0.00	3
10005	0.44	0.57	0.50	40
10006	0.50	0.43	0.47	46
10007	0.00	0.00	0.00	1
10008	0.57	0.51	0.54	45
10009	0.00	0.00	0.00	3
10010	0.00	0.00	0.00	13
10011	0.00	0.00	0.00	3
10012	0.00	0.00	0.00	8
10013	0.42	0.30	0.35	27
10014	0.00	0.00	0.00	7
10015	0.64	0.67	0.65	27
10016	0.17	0.06	0.09	17
10017	0.00	0.00	0.00	13
10018	1.00	0.15	0.27	13
10019	0.20	0.09	0.13	11
accuracy			0.43	500
macro avg	0.28	0.23	0.23	500
weighted avg	0.41	0.43	0.40	500

My_Model_Topic Classification report

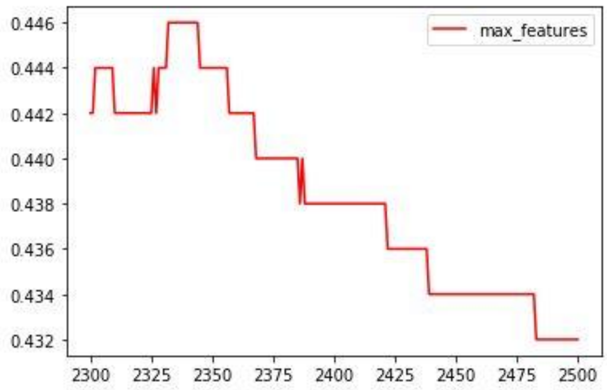
This when compared to the standard Multinomial Naïve Bayes Classification for Topic Analysis is:

	precision	recall	f1-score	support
10000	0.57	0.55	0.56	66
10001	0.00	0.00	0.00	39
10002	0.30	0.11	0.16	27
10003	0.25	0.82	0.38	91
10004	0.00	0.00	0.00	3
10005	0.32	0.33	0.32	40
10006	0.48	0.26	0.34	46
10007	0.00	0.00	0.00	1
10008	0.50	0.27	0.35	45
10009	0.00	0.00	0.00	3
10010	0.00	0.00	0.00	13
10011	0.00	0.00	0.00	3
10012	0.00	0.00	0.00	8
10013	0.40	0.15	0.22	27
10014	0.00	0.00	0.00	7
10015	0.53	0.30	0.38	27
10016	0.00	0.00	0.00	17
10017	0.00	0.00	0.00	13
10018	0.00	0.00	0.00	13
10019	0.00	0.00	0.00	11
accuracy			0.33	500
macro avg	0.17	0.14	0.14	500
weighted avg	0.30	0.33	0.27	500

Standard Multinomial Naïve Bayes for Topic Analysis for test data

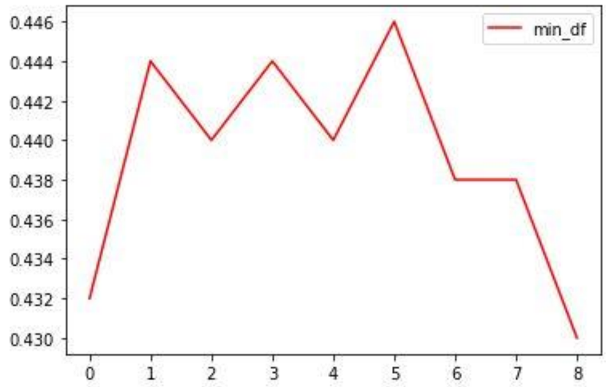
- Each topic precision seems to be comparatively higher in ‘My_Model_topic’ and the accuracy is also increasing by 10%.

The experimental data statistics are shown below:



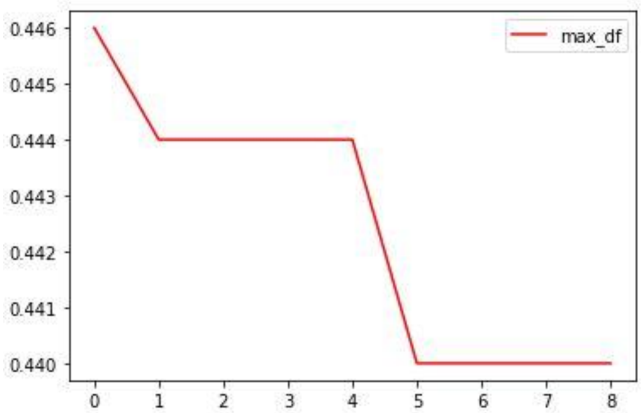
Max_features ranging from 2300 to 2500

- It can the observed that the model has the highest accuracy at 44.6% at around 2350 max_features but when combines with stemming and stop words, the accuracy goes down.
- So, at 2500 max_features, the model seems to be predicting at a higher accuracy than the standard model.



Min_df varying from 1 to 9

- Min_df when varied from 1 to 9 produces a model with accuracy coming up to 43% itself.
- But when mixed with other features, the accuracy is going down. So, its better if these are not chosen and max_features are given a preference.



Max_df varying from 0.1 to 0.9

- Max_df when varied from 0.1 to 0.9 produces a model with an accuracy coming up to 43%.
- We have already achieved that with max_features. It would just be redundant to add these features into our pre-processing.

So, after applying **Porter Stemming** and **'max_features=2500'** and **'lowercase=True'** and **stop_words=stopwords.words('english')**, My_Model_Topic is predicting with an accuracy of 43%.