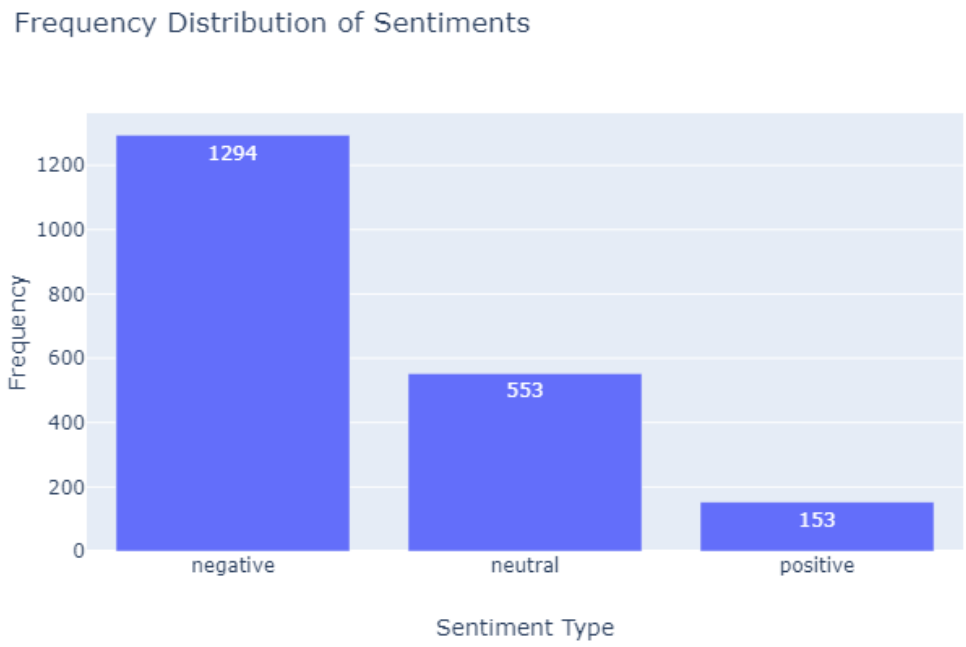
Ajay Umasankar
z5025214

Responses are not just a single paragraph but are split up for ease of reading

Metrics that are focused on include Accuracy, Macro precision and Macro recall since the micro variants are equivalent to accuracy in most of the cases. F1 score is also not used since it is just the result of using precision and recall to get another subjective number result. However, weighted average recall and precision could have been used instead but I decided to value precision and recall in each class equally as important. Hence, it is not included in order to keep accumulated data less cluttered and to come to easier conclusions.

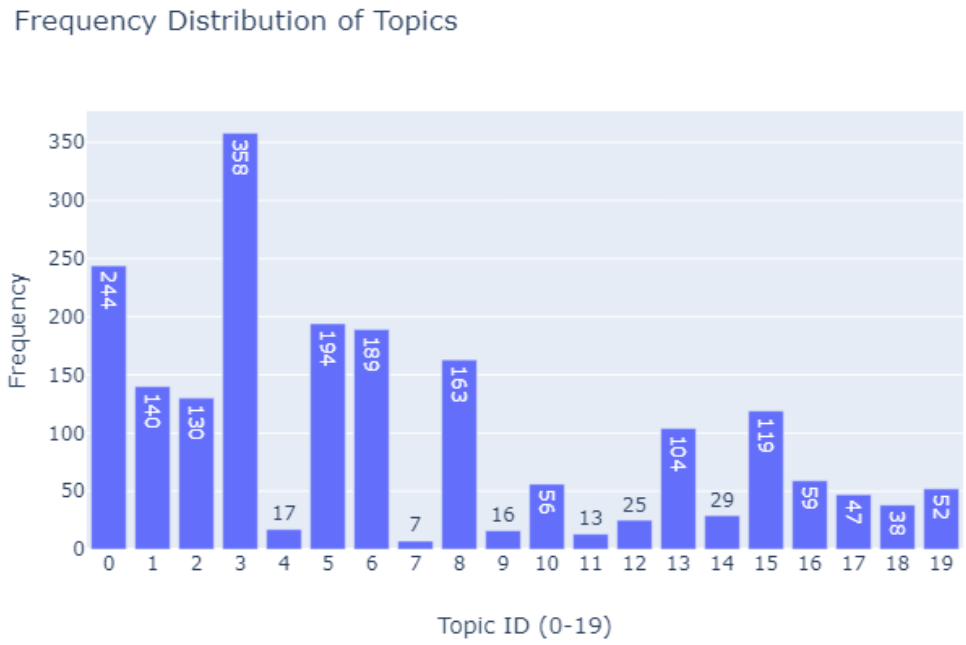
Question 1. Frequency Distribution

From the distribution graph below it is clear that the overwhelming majority of tweets carry a negative sentiment (1294 negative vs its runner up 553 neutral). Only 153 out of 2000 tweets have a positive sentiment. This means that even a DummyClassifier that predicts only a negative sentiment could have a high success rate.



Ajay Umasankar
z5025214

The distributions of topics are much more sporadic but there is still a clear identifiable leader in terms of the majority class (Topic 10003 with 358 occurrences). We can already tell from this distribution that the learner needs to be quite developed in order to predict correct test set results.



ID	Topic
10000	corruption/governance
10001	employment/jobs
10002	tax/negative gearing
10003	economic management
10004	superannuation
10005	healthcare/medicare
10006	social issues/marriage equality/religion
10007	indigenous affairs
10008	asylum seekers/refugees
10009	early education and child care
10010	school education
10011	higher education
10012	innovation/science/research
10013	environment/climate change
10014	infrastructure
10015	telecommunications/nbn
10016	terrorism/national security
10017	foreign policy
10018	agriculture/irrigation/dairy industry
10019	mining and energy

Ajay Umasankar
z5025214

Question 2. Max Features

(2 marks) Vary the number of words from the vocabulary used as training features for the standard methods (e.g. the top N words for N = 100, 200, etc.). Show metrics calculated on both the training set and the test set.

Explain any difference in performance of the models between training and test set

Metrics are all generally higher when the model is run on the training set. (Subsequently, when difference in metrices is mentioned below, this implies that the model predicting training set entries always has the higher (more favourable) metrics than when predicting test set entries)

When max features is higher, the difference in performance between training and test set also increases, this is clear for BNB and MNB.

This difference can be explained by the unseen features that are in the test set (also, it is familiar with all features in training set). It might also partially be due to overfitting. Since the model did train on the training set, it will naturally do better on the training set because of these reasons.

Comment on metrics and runtimes in relation to the number of features

https://piazza.com/class/jvhnwcx8t2o5qg?cid=277 – runtime here refers to training time as specified. The bag of words vector transformation time taken changes only minimally taking 0.0488977s when max_features = 1000 and 0.042890s when max_features = 100

The difference between Training set and Test set accuracies, macro-precision and macro-recall for BNB and MNB generally increase at an increasing rate as max features increases, with topic gaining considerably more accuracy as we go with more features.

Although training set still has a very slight edge over test set for DT in terms of favourable metrics, increasing max features barely has any effect on any of the metrics. E.g. Macro precision and macro recall is the same for when max features is 300, 500, 700

Runtimes almost quadruple when going from 100 to 700 features for BNB.
Runtimes triple when going from 100 to 700 features for MNB
Runtimes is atleast 6x slower when going from 100 to 700 features for DT

Although runtimes are slower, this doesn't necessarily correspond to an increase in favourable metrices (as is obvious by DT and previous explanations) and hence we should be aware of maximizing this cost-benefit problem

Ajay Umasankar
z5025214

Decimal places were omitted for this 'item' only to easily pick out metrics such as largest accuracy by quickly analysing a column.

Max Features = 100						
	Model	**Type**	**Accuracy (%)**	**Macro-Precision (%)**	**Macro-Recall (%)**	**Runtime (s)**
Training Set	BNB	Sentiment	72	63	55	0.00396
		Topic	40	39	26	0.00299
	MNB	Sentiment	72	66	55	0.00201
		Topic	41	40	26	0.00301
	DT	Sentiment	70	58	46	0.00898
		Topic	35	20	18	0.01097
	Model	**Type**	**Accuracy (%)**	**Macro-Precision (%)**	**Macro-Recall (%)**	**Runtime (s)**
Test Set	BNB	Sentiment	73	58	51	0.00299
		Topic	26	17	15	0.00296
	MNB	Sentiment	73	61	51	0.00200
		Topic	26	15	14	0.00198
	DT	Sentiment	68	46	41	0.00898
		Topic	27	16	14	0.1097

Max Features = 200									
	Model	**Type**	**Accuracy (%)**	**Macro-Precision (%)**	**Macro-Recall (%)**	**Positive Precision (%)**	**Negative Precision (%)**	**Neutral Precision (%)**	**Runtime (s)**
Training Set	BNB	Sentiment	75.8	70.62	64.57	65.85	80.47	65.55	0.00499
		Topic	50.53	47.27	32.66				0.00498
	MNB	Sentiment	76.13	71.22	63.70	67.11	80.34	66.22	0.00296
		Topic	52.53	51.98	38.48				0.00399
	DT	Sentiment	69.87	57.53	46.58	42.31	73.94	56.33	0.01795
		Topic	38.53	24.14	21.86				0.02509
	Model	**Type**							
Test Set	BNB	Sentiment	72.4	61.43	53.66	50.0	79.28	55.0	0.00496
		Topic	32.8	21.12	18.12				0.00499
	MNB	Sentiment	73.4	66.49	53.82	64.29	79.03	56.14	0.00297
		Topic	32.4	21.29	19.14				0.00399
	DT	Sentiment	68.8	47.00	41.98	20.00	74.32	46.67	0.05195
		Topic	30.0	18.69	16.33				0.02493

Ajay Umasankar
z5025214

Max Features = 300						
	Model	**Type**	**Accuracy (%)**	**Macro-Precision (%)**	**Macro-Recall (%)**	**Runtime (s)**
Training Set	BNB	Sentiment	77	71	66	0.00596
		Topic	54	43	33	0.00695
	MNB	Sentiment	78	73	67	0.00397
		Topic	57	60	42	0.00396
	DT	Sentiment	70	58	47	0.02693
		Topic	39	24	22	0.03490
	Model	**Type**	**Accuracy (%)**	**Macro-Precision (%)**	**Macro-Recall (%)**	**Runtime (s)**
Test Set	BNB	Sentiment	73	68	55	0.00596
		Topic	34	21	19	0.00598
	MNB	Sentiment	73	65	56	0.00396
		Topic	33	20	19	0.00396
	DT	Sentiment	69	47	42	0.02493
		Topic	30	19	16	0.03792

Max Features = 500						
	Model	**Type**	**Accuracy (%)**	**Macro-Precision (%)**	**Macro-Recall (%)**	**Runtime (s)**
Training Set	BNB	Sentiment	81	79	71	0.00898
		Topic	59	55	34	0.00897
	MNB	Sentiment	81	78	71	0.00495
		Topic	65	67	47	0.00495
	DT	Sentiment	70	58	47	0.04088
		Topic	39	24	22	0.05282
Test Set	BNB	Sentiment	72	59	52	0.00997
		Topic	35	20	18	0.00896
	MNB	Sentiment	72	66	56	0.00495
		Topic	37	22	21	0.00488
	DT	Sentiment	69	47	42	0.03988
		Topic	30	19	16	0.05574

Ajay Umasankar
z5025214

Max Features = 700						
	Model	**Type**	**Accuracy (%)**	**Macro-Precision (%)**	**Macro-Recall (%)**	**Runtime (s)**
Training Set	BNB	Sentiment	83	80	73	0.00598
		Topic	68	69	50	0.00698
	MNB	Sentiment	83	81	72	0.01193
		Topic	58	52	31	0.01196
	DT	Sentiment	70	58	47	0.05384
		Topic	39	24	22	0.07878
	Model	**Type**	**Accuracy (%)**	**Macro-Precision (%)**	**Macro-Recall (%)**	**Runtime (s)**
Test Set	BNB	Sentiment	73	64	52	0.01297
		Topic	34	19	17	0.01394
	MNB	Sentiment	73.8	69	59	0.00594
		Topic	38	23	21	0.00696
	DT	Sentiment	69	37	42	0.05274
		Topic	30	19	16	0.07782

Ajay Umasankar
z5025214

Question 3. Baseline Predictors

- Using Vader as the baseline predictor, our standard models perform very well with about a 30% higher accuracy and higher neutral precision and macro precision. (possibly due to issues with VADER discussed in assignment spec).
- Using majority class as a baseline for Sentiment, we get very similar accuracy and lower macro precision and macro recall compared to our standard models.
- Using majority class as a baseline for Topic, we get lower accuracy (except for BNB) and lower macro precision and macro recall compared to our standard models.
- This low macro recall and macro precision on Majority class baselines is because we are using averages between individual precisions/recalls, not weighted averages.

Standard Models, Sentiment Analysis								
Test set	**Model**	**Type**	**Accuracy (%)**	**Macro-Precision (%)**	**Macro-Recall (%)**	**Positive Precision (%)**	**Negative Precision (%)**	**Neutral Precision (%)**
	BNB	Sentiment	71.6	47.12	40.97	0	71.81	69.57
	MNB	Sentiment	74.0	64.31	52.35	55.56	80.11	57.26
	DT	Sentiment	68.8	47.00	41.98	20.00	74.32	46.67
			Accuracy (%)	**Macro-Precision (%)**	**Macro-Recall (%)**	**Positive Precision (%)**	**Negative Precision (%)**	**Neutral Precision (%)**
Baseline (VADER)			43.2	39.98	44.96	13.56	76.96	29.41
Baseline (Majority Class = Negative)			67	22.33	33.33	0	67.00	0

Standard Models, Topic Analysis					
Test set	**Model**	**Type**	**Accuracy (%)**	**Macro-Precision (%)**	**Macro-Recall (%)**
	BNB	Topic	18.0	2.76	5.27
	MNB	Topic	28.8	17.29	12.53
	DT	Topic	30.0	18.69	16.33
Baseline (Majority Class = Topic 10003)			**Accuracy (%)**	**Macro-Precision (%)**	**Macro-Recall (%)**
			17.4	0.87	5

Ajay Umasankar
z5025214

Question 4. Porter Stems and Stop words

- Stemming and stop word removal generally increases all metrics for Topic and reduces all metrics for Sentiment amongst both training and test sets.
- However, BNB Sentiment and topic are mostly unaffected in relation to metrics after the additional pre-processing.

Standard Models, Stop word removal + Porter stemming					
Model		**Type**	**Accuracy (%)**	**Macro-Precision (%)**	**Macro-Recall (%)**
Training	BNB	Sentiment	82.33	58.32	54.83
		Topic	25.53	12.78	8.01
	MNB	Sentiment	94.2	95.89	83.63
		Topic	77.47	85.16	55.10
	DT	Sentiment	68.8	42.25	41.74
		Topic	37.87	22.41	20.57
Test	BNB	Sentiment	70.6	43.68	41.31
		Topic	18.6	3.39	5.55
	MNB	Sentiment	70.8	53.44	50.70
		Topic	34.6	20.14	17.00
	DT	Sentiment	68.4	38.33	39.04
		Topic	31.6	20.12	17.88

Standard Models					
Model		**Type**	**Accuracy (%)**	**Macro-Precision (%)**	**Macro-Recall (%)**
Training	BNB	Sentiment	83.2	59.01	55.84
		Topic	24.4	5.41	7.53
	MNB	Sentiment	93.73	96.02	81.87
		Topic	71.93	79.88	45.32
	DT	Sentiment	69.87	57.53	46.58
		Topic	38.53	24.14	21.86
Test	BNB	Sentiment	71.6	47.12	40.97
		Topic	18.0	2.76	5.27
	MNB	Sentiment	74.0	64.31	52.35
		Topic	28.8	17.29	12.53
	DT	Sentiment	68.8	47.00	41.98
		Topic	30.0	18.69	16.33

Ajay Umasankar
z5025214

Question 5. Neutral Tweet Deletion

Question 2 Repeat

- We can see a huge improvement in accuracies across the board with neutral removed (going from around 75 average to consistent 90s).
- Macro precision and Macro recall also increase about 15 compared to when neutral statements are included.
- Negative precisions is also even higher than accuracy nearing 95% for most models. Positive precisions sometimes increase and sometimes decrease, possibly due to the lack of sample cases with positive sentiment.
- Runtimes are also faster for all models, but DT has an especially noticeable reduction in runtime (about 6x faster for the test set)

Standard Models, Max Features = 200, Neutral removed							
	Model	**Accuracy (%)**	**Macro-Precision (%)**	**Macro-Recall (%)**	**Positive Precision (%)**	**Negative Precision (%)**	**Runtime (s)**
Training	BNB Sentiment	91.42	77.71	73.73	61.05	94.37	0.00454
	MNB Sentiment	91.88	79.31	74.39	64.13	94.49	0.00200
	DT Sentiment	90.67	78.43	61.61	65.12	91.74	0.00898
Test	BNB Sentiment	90.93	77.69	67.41	62.50	92.88	0.00399
	MNB Sentiment	90.67	77.14	65.06	61.90	92.37	0.00200
	DT Sentiment	90.13	76.35	59.25	61.54	91.16	0.00798

Standard Models, Max Features = 200								
	Model	**Accuracy (%)**	**Macro-Precision (%)**	**Macro-Recall (%)**	**Positive Precision (%)**	**Negative Precision (%)**	**Neutral Precision (%)**	**Runtime (s)**
Training Set	BNB Sentiment	75.8	70.62	64.57	65.85	80.47	65.55	0.00499
	MNB Sentiment	76.13	71.22	63.70	67.11	80.34	66.22	0.00296
	DT Sentiment	69.87	57.53	46.58	42.31	73.94	56.33	0.01795
Test Set	BNB Sentiment	72.4	61.43	53.66	50.0	79.28	55.0	0.00496
	MNB Sentiment	73.4	66.49	53.82	64.29	79.03	56.14	0.00297
	DT Sentiment	68.8	47.00	41.98	20.00	74.32	46.67	0.05195

Ajay Umasankar
z5025214

Question 3 Repeat

- Like Q2, accuracies increase greatly across the board (except for VADER). In this case however, macro precision increases greatly for BNB and DT.
- In general, positive precision decreases and negative precision increases greatly when neutral cases are removed.

Sentiment Analysis, Neutral Removed							
Test set	**Model**	**Type**	**Accuracy (%)**	**Macro-Precision (%)**	**Macro-Recall (%)**	**Positive Precision (%)**	**Negative Precision (%)**
	BNB	Sentiment	89.60	94.79	51.25	100	89.57
	MNB	Sentiment	89.07	69.57	60.86	47.62	91.53
	DT	Sentiment	90.13	76.35	59.25	61.54	91.16
			Accuracy (%)	**Macro-Precision (%)**	**Macro-Recall (%)**	**Positive Precision (%)**	**Negative Precision (%)**
Baseline (VADER)			48.27	37.93	35.62	19.20	94.58
Baseline (Majority Class = Negative)			89.33	44.67	50.00	0	89.33

Sentiment Analysis								
Test set	**Model**	**Type**	**Accuracy (%)**	**Macro-Precision (%)**	**Macro-Recall (%)**	**Positive Precision (%)**	**Negative Precision (%)**	**Neutral Precision (%)**
	BNB	Sentiment	71.6	47.12	40.97	0	71.81	69.57
	MNB	Sentiment	74.0	64.31	52.35	55.56	80.11	57.26
	DT	Sentiment	68.8	47.00	41.98	20.00	74.32	46.67
			Accuracy (%)	**Macro-Precision (%)**	**Macro-Recall (%)**	**Positive Precision (%)**	**Negative Precision (%)**	**Neutral Precision (%)**
Baseline (VADER)			43.2	39.98	44.96	13.56	76.96	29.41
Baseline (Majority Class = Negative)			67	22.33	33.33	0	67.00	0

Ajay Umasankar
z5025214

Q4 Repeat

- We saw metrics in general decrease with stemming for sentiments before, but it is more of a random distribution now and negative precision is even higher with all the preprocessing.
- Overall, the change seems to be minimal with metrics decreasing/increasing randomly. However, when BNB was only slightly affected in Q4 previously, there seems to be no effect on the stopword removal and stemming now on the neutral removed sets.

Standard Models, Neutral removed + Stopwords remove + Porter stemming						
	Model	**Accuracy (%)**	**Macro-Precision (%)**	**Macro-Recall (%)**	**Positive Precision (%)**	**Negative Precision (%)**
Training	BNB Sentiment	89.83	94.90	51.77	100	89.79
	MNB Sentiment	97.39	98.05	88.00	98.85	97.26
	DT Sentiment	90.21	74.50	63.30	56.90	92.11
Test	BNB Sentiment	89.60	94.79	51.25	100	89.57
	MNB Sentiment	85.87	64.14	65.67	35.56	92.73
	DT Sentiment	91.47	82.33	65.50	72.22	92.44

Standard Models, Neutral removed						
	Model	**Accuracy (%)**	**Macro-Precision (%)**	**Macro-Recall (%)**	**Positive Precision (%)**	**Negative Precision (%)**
Training	BNB Sentiment	89.83	94.90	51.77	100	89.79
	MNB Sentiment	96.92	98.34	85.40	100	96.67
	DT Sentiment	90.67	78.43	61.61	65.12	91.74
Test	BNB Sentiment	89.60	94.79	51.25	100	89.57
	MNB Sentiment	89.07	69.57	60.86	47.62	91.53
	DT Sentiment	90.13	76.35	59.25	61.54	91.16

Ajay Umasankar
z5025214

Standard Models, Stop word removal + Porter stemming					
Model		**Type**	**Accuracy (%)**	**Macro-Precision (%)**	**Macro-Recall (%)**
Training	BNB	Sentiment	82.33	58.32	54.83
		Topic	25.53	12.78	8.01
	MNB	Sentiment	94.2	95.89	83.63
		Topic	77.47	85.16	55.10
	DT	Sentiment	68.8	42.25	41.74
		Topic	37.87	22.41	20.57
Test	BNB	Sentiment	70.6	43.68	41.31
		Topic	18.6	3.39	5.55
	MNB	Sentiment	70.8	53.44	50.70
		Topic	34.6	20.14	17.00
	DT	Sentiment	68.4	38.33	39.04
		Topic	31.6	20.12	17.88

Standard Models					
Model		**Type**	**Accuracy (%)**	**Macro-Precision (%)**	**Macro-Recall (%)**
Training	BNB	Sentiment	83.2	59.01	55.84
		Topic	24.4	5.41	7.53
	MNB	Sentiment	93.73	96.02	81.87
		Topic	71.93	79.88	45.32
	DT	Sentiment	69.87	57.53	46.58
		Topic	38.53	24.14	21.86
Test	BNB	Sentiment	71.6	47.12	40.97
		Topic	18.0	2.76	5.27
	MNB	Sentiment	74.0	64.31	52.35
		Topic	28.8	17.29	12.53
	DT	Sentiment	68.8	47.00	41.98
		Topic	30.0	18.69	16.33

Ajay Umasankar
z5025214

Question 6. Own Models

Could have tested on equal amounts neutral, positive and negative but not enough data to do so.
My models were made with accuracy as the main benchmark for performance. This is because I believe that maximizing the number of correctly classified classes over other subjective metrics such as macro precision and f1 score are more important. (this is also true of the rest of the report)

The arrows on the right symbolize an improvement/decrease in accuracy from the previous test. Other metrics are shown as well but for each test (in **BOLD)** one should generally just look at the accuracy in comparison to the next test. Highlighted in RED indicates the variable that is changed

Even if some additional preprocessing made accuracy lower, they weren't completely disregarded and were indeed tested somewhat frequently, sometimes this is shown below but left excluded on purpose to reduce report size.

Note: The englishstemmer (`from nltk.stem.snowball import EnglishStemmer`) was used in the majority of tests because nltk suggested that this stemmer is an improvement over the porterstemmer (and some of the tests below when comparing to previous report results indicate this as well)

Ajay Umasankar
z5025214

Sentiment:

BNB was initially chosen due to its consistently high sentiment scores and receptiveness to changing different variables.

Normal MNB, lowercase=True

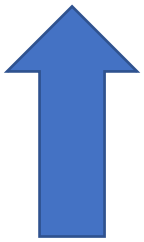
```
Accuracy:   0.736
Precision (array):  [0.79310345 0.56140351 0.55555556]
Precision (macro):  0.6366875042011159
Recall (macro):     0.5098457711442786
              precision    recall  f1-score   support

    negative       0.79      0.89      0.84       335
     neutral       0.56      0.51      0.54       125
    positive       0.56      0.12      0.20        40

    accuracy                           0.74       500
   macro avg       0.64      0.51      0.53       500
weighted avg       0.72      0.74      0.71       500
```

Normal MNB, lowercase=False

```
Accuracy:   0.74
Precision (array):  [0.80327869 0.568      0.55555556]
Precision (macro):  0.6422780813600485
Recall (macro):     0.5235373134328358
              precision    recall  f1-score   support

    negative       0.80      0.88      0.84       335
     neutral       0.57      0.57      0.57       125
    positive       0.56      0.12      0.20        40

    accuracy                           0.74       500
   macro avg       0.64      0.52      0.54       500
weighted avg       0.72      0.74      0.72       500
```

Normal MNB, lowercase=True, EnglishStemmer

```
Accuracy:   0.746
Precision (array):  [0.8032345 0.5785124 0.625    ]
Precision (macro):  0.6689156326806412
Recall (macro):     0.5248507462686568
              precision    recall  f1-score   support

    negative       0.80      0.89      0.84       335
     neutral       0.58      0.56      0.57       125
    positive       0.62      0.12      0.21        40

    accuracy                           0.75       500
   macro avg       0.67      0.52      0.54       500
weighted avg       0.73      0.75      0.72       500
```



Ajay Umasankar
z5025214

Normal MNB, lowercase=True, EnglishStemmer + stopword removal

```
Accuracy:   0.726
Precision (array):  [0.81449275 0.55639098 0.36363636]
Precision (macro):  0.5781733649010538
Recall (macro):     0.5436019900497512
              precision    recall  f1-score   support

    negative       0.81      0.84      0.83       335
     neutral       0.56      0.59      0.57       125
    positive       0.36      0.20      0.26        40

    accuracy                           0.73       500
   macro avg       0.58      0.54      0.55       500
weighted avg       0.71      0.73      0.72       500
```

Normal MNB, lowercase=False, EnglishStemmer, max_features=900

This was one of the higher results for varying max_features, more info in next test

```
Accuracy:   0.742
Precision (array):  [0.82369942 0.54609929 0.69230769]
Precision (macro):  0.687368801684384
Recall (macro):     0.5639154228855722
              precision    recall  f1-score   support

    negative       0.82      0.85      0.84       335
     neutral       0.55      0.62      0.58       125
    positive       0.69      0.23      0.34        40

    accuracy                           0.74       500
   macro avg       0.69      0.56      0.59       500
weighted avg       0.74      0.74      0.73       500

Time:  0.006981611251831055
```

Normal MNB, lowercase=True, EnglishStemmer, max_features=900

Max_features was varied between 200-3000 and there were accuracies of up to 75 but the best result so far was 75.2 with max_features = 900

```
Accuracy:   0.752
Precision (array):  [0.82471264 0.57553957 0.69230769]
Precision (macro):  0.6975199681103922
Recall (macro):     0.5739054726368159
              precision    recall  f1-score   support

    negative       0.82      0.86      0.84       335
     neutral       0.58      0.64      0.61       125
    positive       0.69      0.23      0.34        40

    accuracy                           0.75       500
   macro avg       0.70      0.57      0.60       500
weighted avg       0.75      0.75      0.74       500
```

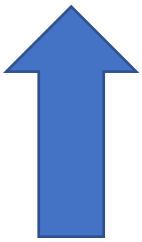
Ajay Umasankar
z5025214

Normal MNB, lowercase=True, EnglishStemmer, max_features=900, Stopwords

```
Accuracy:   0.706
Precision (array):  [0.80177515 0.51824818 0.44      ]
Precision (macro):  0.5866744410371586
Recall (macro):     0.5506517412935322
              precision    recall  f1-score   support

    negative       0.80      0.81      0.81       335
     neutral       0.52      0.57      0.54       125
    positive       0.44      0.28      0.34        40

    accuracy                           0.71       500
   macro avg       0.59      0.55      0.56       500
weighted avg       0.70      0.71      0.70       500
```

Alpha=1.1 MNB, lowercase=True, EnglishStemmer, max_features=900

```
Accuracy:   0.76
Precision (array):  [0.82670455 0.58823529 0.75      ]
Precision (macro):  0.7216466131907309
Recall (macro):     0.5778855721393036
              precision    recall  f1-score   support

    negative       0.83      0.87      0.85       335
     neutral       0.59      0.64      0.61       125
    positive       0.75      0.23      0.35        40

    accuracy                           0.76       500
   macro avg       0.72      0.58      0.60       500
weighted avg       0.76      0.76      0.75       500
```



Alpha=.95 MNB, lowercase=True, EnglishStemmer, max_features=900 Training/Test Split = 1600/400

Training test split was varied with the other features and the best result that I found is the one featured below

```
Accuracy:   0.75
Precision (array):  [0.82206406 0.56363636 0.77777778]
Precision (macro):  0.7211593994512144
Recall (macro):     0.567705589698566
              precision    recall  f1-score   support

    negative       0.82      0.86      0.84       268
     neutral       0.56      0.61      0.58       102
    positive       0.78      0.23      0.36        30

    accuracy                           0.75       400
   macro avg       0.72      0.57      0.60       400
weighted avg       0.75      0.75      0.74       400
```

Ajay Umasankar
z5025214

My Best Sentiment Classifier

Best: Alpha=1.1 MNB, lowercase=True, EnglishStemmer, max_features=900

```
Accuracy:   0.76
Precision (array):  [0.82670455 0.58823529 0.75]
Precision (macro):  0.7216466131907309
Recall (macro):     0.5778855721393036
              precision    recall  f1-score   support

    negative       0.83      0.87      0.85       335
     neutral       0.59      0.64      0.61       125
    positive       0.75      0.23      0.35        40

    accuracy                           0.76       500
   macro avg       0.72      0.58      0.60       500
weighted avg       0.76      0.76      0.75       500
```

My best method for sentiment makes use of some of the features discussed throughout the report and with adjusting alpha on top. The experimental results are above.

(Refer to table below for following discussion)

In relation to the standard methods and baselines, my method has the largest accuracy, macro precision, macro recall, positive precision and negative precision by a good amount for some metrics. It also makes clear that my model is lacking in the Neutral area and it would be a good guess to say that my model would perform very well and even better than the standard models when the neutral set is removed.

It also convincingly beats baselines in all metrics.

Standard Models, Sentiment Analysis								
Test set	**Model**	**Type**	**Accuracy (%)**	**Macro-Precision (%)**	**Macro-Recall (%)**	**Positive Precision (%)**	**Negative Precision (%)**	**Neutral Precision (%)**
	BNB	Sentiment	71.6	47.12	40.97	0	71.81	69.57
	MNB	Sentiment	74.0	64.31	52.35	55.56	80.11	57.26
	DT	Sentiment	68.8	47.00	41.98	20.00	74.32	46.67
	My Model	Sentiment	76.0	72.16	57.79	75.00	82.67	58.82
			Accuracy (%)	**Macro-Precision (%)**	**Macro-Recall (%)**	**Positive Precision (%)**	**Negative Precision (%)**	**Neutral Precision (%)**
Baseline (VADER)			43.2	39.98	44.96	13.56	76.96	29.41
Baseline (Majority Class = Negative)			67	22.33	33.33	0	67.00	0

Ajay Umasankar
z5025214

Topic

Initially experimented generally with all Standard Models, but once again MNB showed consistently higher accuracies for Topic as well.

Normal MNB, lowercase=True (default)

```
Accuracy:   0.332
Precision (macro):  0.18394516649956366
Recall (macro):     0.14851831798650728
              precision    recall  f1-score   support
    accuracy                           0.33       500
   macro avg       0.18      0.15      0.14       500
weighted avg       0.32      0.33      0.28       500
```

Normal MNB, lowercase=False

```
Accuracy:   0.306
Precision (macro):  0.1913735141914655
Recall (macro):     0.13524483254285694

              precision    recall  f1-score   support
    accuracy                           0.31       500
   macro avg       0.19      0.14      0.14       500
weighted avg       0.33      0.31      0.26       500
```

Normal MNB, lowercase=False, max_features=900

```
Accuracy:   0.396
Precision (macro):  0.26017550047980914
Recall (macro):     0.22441750342056235

              precision    recall  f1-score   support
    accuracy                           0.40       500
   macro avg       0.26      0.22      0.23       500
weighted avg       0.39      0.40      0.38       500
```

Normal MNB, lowercase=True, max_features=900

```
Accuracy:   0.406
Precision (macro):  0.2742801756990422
Recall (macro):     0.23633778548835133

              precision    recall  f1-score   support
    accuracy                           0.41       500
   macro avg       0.27      0.24      0.24       500
weighted avg       0.40      0.41      0.39       500
```

Ajay Umasankar
z5025214

Normal MNB, lowercase=True, max_features=900, stop_words='english'

```
Accuracy:   0.446
Precision (macro):  0.29711213692440225
Recall (macro):     0.24810867297071776

              precision    recall  f1-score   support
    accuracy                           0.45       500
   macro avg       0.30      0.25      0.26       500
weighted avg       0.44      0.45      0.42       500
Accuracy:   0.454
```

Normal MNB, lowercase=True, max_features=900, stop_words='english', stemmer=EnglishStemmer

```
Accuracy:   0.454
Precision (macro):  0.3015489394874101
Recall (macro):     0.26130656676897673

              precision    recall  f1-score   support
    accuracy                           0.45       500
   macro avg       0.30      0.26      0.27       500
weighted avg       0.44      0.45      0.43       500
```

Normal MNB, lowercase=True, max_features=900, stop_words='english', stemmer=PorterStemmer

```
Accuracy:   0.444
Precision (macro):  0.29065065968083537
Recall (macro):     0.2578763908313045

              precision    recall  f1-score   support
    accuracy                           0.44       500
   macro avg       0.29      0.26      0.26       500
weighted avg       0.43      0.44      0.42       500
```

Alpha=0.78 MNB, lowercase=True, max_features=900, stop_words='english', stemmer=EnglishStemmer

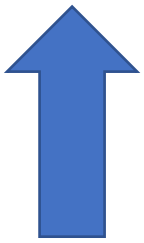
Alpha was varied with lowercase, stop_words and stemmer and this was the best result, other results omitted for the sake of report length

```
Accuracy:   0.458
Precision (macro):  0.31382615269122016
Recall (macro):     0.27384427132559114

              precision    recall  f1-score   support
    accuracy                           0.46       500
   macro avg       0.31      0.27      0.28       500
weighted avg       0.44      0.46      0.44       500
```

Ajay Umasankar
z5025214

```
Time:  0.00898599624633789
```

Alpha=0.77 MNB, lowercase=True, max_features=1000, stop_words='english', stemmer=EnglishStemmer



Alpha was varied with lowercase, stop_words and stemmer and this was the best result, other results omitted for the sake of report length

```
Accuracy:   0.466
Precision (macro):  0.32536129852632323
Recall (macro):     0.28173814087867144

              precision    recall  f1-score   support
    accuracy                           0.47       500
   macro avg       0.33      0.28      0.29       500
weighted avg       0.45      0.47      0.45       500
```

Alpha=0.99 MNB, lowercase=True, max_features=700, stop_words='english', stemmer=EnglishStemmer

```
Accuracy:   0.466
Precision (macro):  0.3547282722424256
Recall (macro):     0.2740316189773316
              precision    recall  f1-score   support
    accuracy                           0.47       500
   macro avg       0.35      0.27      0.28       500
weighted avg       0.46      0.47      0.44       500
```

Ajay Umasankar
z5025214

My Best Topic Classifier

Best: Alpha=0.77 MNB, lowercase=True, max_features=1000, stop_words='english', stemmer=EnglishStemmer

```
Accuracy:   0.466
Precision (macro):  0.32536129852632323
Recall (macro):     0.28173814087867144

              precision    recall  f1-score   support

    accuracy                           0.47       500
   macro avg       0.33      0.28      0.29       500
weighted avg       0.45      0.47      0.45       500
```

My topic classifier has very similar preprocessing options in comparison to my sentiment classifier (probably due to them both being MNB), but both perform very well in comparison to the standards.
My topic classifier has much larger metrics in all fields in comparison to standard models. It has 16.6% more accuracy than the runner up, with 30%.
It also convincingly beats the Baseline as seen in the table.

Standard Models, Topic Analysis					
Test set	**Model**	**Type**	**Accuracy (%)**	**Macro-Precision (%)**	**Macro-Recall (%)**
	BNB	Topic	18.0	2.76	5.27
	MNB	Topic	28.8	17.29	12.53
	DT	Topic	30.0	18.69	16.33
	My model	Topic	46.6	32.54	28.17
Baseline (Majority Class = Topic 10003)			**Accuracy (%)**	**Macro-Precision (%)**	**Macro-Recall (%)**
			17.4	0.87	5

Ajay Umasankar
z5025214

Appendix

Question 4. **Explanation on how stemming and stop word removal was done (READ IF METRICS VARY GREATLY, OTHERWISE NOT RELEVANT)**

- Results are from manually removing stopwords using NLTK 'english' corpus and manually stemming using Porter stemmer. **Upper/Lowercase is retained after stemming**
- Could be done by changing CountVectorizer arguments also but the order of application of stopword removal and stemming means that stopwords aren't properly recognized.
    - e.g. if stemming using analyser=, then stop_words doesn't work as specified in CountVectorizer documentation
    - e.g. if stemming is manually done outside of CountVectorizer, then stop_words='english' as CountVectorizer argument wouldn't be completely accurate on the stemmed set of words. (since original stopwords could have been stemmed)
- There are some workarounds for this as well, but then there is the issue of the stemmer automatically converting words to lowercase.
    - If we attempt to keep the case of the words in the tweets manually, then stop_words='english' cant recognize uppercase stopwords like 'Too' unless we use lowercase=True, which is against the spec.
- Hence, the sentences and words within them are scanned and stemmed manually in my method.