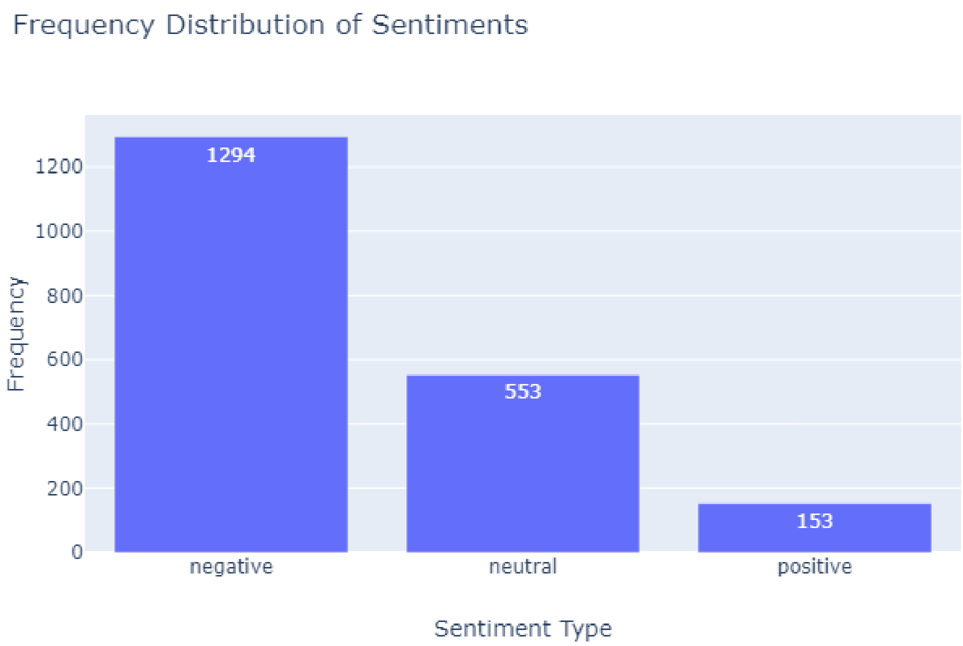
Ajay Umasankar
z5025214

Responses are small paragraphs but split up for ease of reading

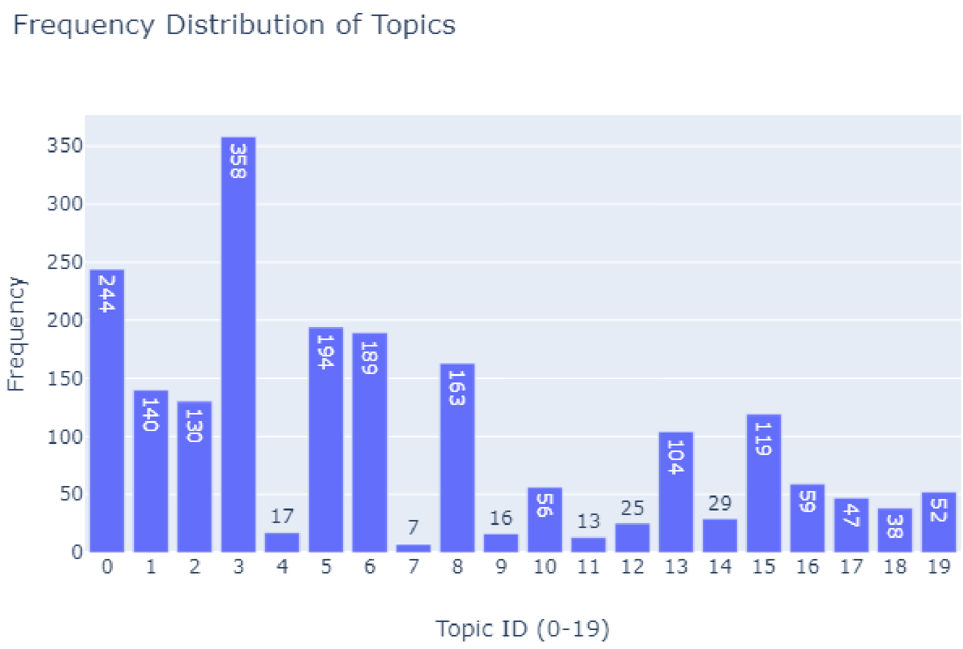
Question 1. Frequency Distribution

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						
Training Set	**Model**	**Type**	**Accuracy (%)**	**Macro-Precision (%)**	**Macro-Recall (%)**	**Runtime (s)**
	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
Test Set	**Model**	**Type**	**Accuracy (%)**	**Macro-Precision (%)**	**Macro-Recall (%)**	**Runtime (s)**
	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

Training:
BNB: 75.85, 62.94, 50.77

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

Ajay Umasankar
z5025214

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

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

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

Topic Analysis								
Training	**Model**	**Type**	**Accuracy (%)**	**Macro-Precision (%)**	**Macro-Recall (%)**	**Positive Precision (%)**	**Negative Precision (%)**	**Neutral Precision (%)**
	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 (%)**	**Positive Precision (%)**	**Negative Precision (%)**	**Neutral Precision (%)**
			17.4	0.87	5			

Ajay Umasankar
z5025214

Q4. 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.

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

Training (neg, neut, pos)
BNB: 79.13,95.83,0.00
MNB: 94.06, 93.62, 100
DT: 71.45, 55.28, 0.00

Ajay Umasankar
z5025214

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

Stemming only, Training

83.67	91.84	57.34
26.27	11.65	8.34
92.60	95.30	78.41
71.40	76.28	43.96
70.07	60.91	49.21
38.4	21.25	20.13

Array : (TRAINING)
BNB sentiment: neg – 80.69, neut – 94.84, pos – 100
MNB sentiment: neg – 91.54, neut – 94.36, pos - 100
DT sentiment: neg – 72.07, 64.77, 45.90

Ajay Umasankar
z5025214

Q5. Answer Q2 (N=200),Q3,Q4

Q2. N=200

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

Q3. VADER, Baseline

Standard Models, 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)			43.20	39.98	44.96	19.20	94.58
Baseline (Majority Class = Negative)			89.33	44.67	50.00	0	89.33

Ajay Umasankar
z5025214

Q4. Stopword + porter stemming without neutral statements

Standard Models, Neutral removed, Stopwords removed, 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