# Kanadech Jirapongtanavech z5176970

|  |  |
| --- | --- |
| Figure 1 Topic Frequency Chart | Figure 2 Sentiment Frequency Chart |

1. (1 mark) Give simple descriptive statistics showing the frequency distributions for the sentiment and topic classes across the full dataset. What do you notice about the distribution?

Topic statistics

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Variable | N | Mean | SE Mean | Standard Dev | Minimum | Q1 | Median | Q3 | Maximum |
| Frequency | 20 | 100.0 | 20.8 | 93.0 | 7.0 | 26.0 | 57.5 | 157.3 | 358.0 |

Sentiment Statistics

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Variable | N | Mean | SE Mean | Standard Dev | Minimum | Q1 | Median | Q3 | Maximum |
| Count | 3 | 667 | 334 | 579 | 153 | 153 | 553 | 1294 | 1294 |

From the figures and tables above, many tweets were talking about topic 10003, economic management and were negative tweets. The class distribution for both is skewed. More than half the tweets were negative tweets. The skewed topics distribution is, in my opinion, the expected distribution. Voters tend to care about things that are closed to them or things that could have significant impact on their livelihood such as economic management which accounted for 17.9% of the tweets.

1. (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, and comment on metrics and runtimes in relation to the number of features.

Top N – Top number of feature words  
Dataset – Data used as test sets [train, test]  
P\_MI – Precision score micro average  
P\_MA – Precision score macro average  
P\_W – Precision score weighted average  
R\_MI – Recall score micro average  
R\_MA – Recall score macro average  
R\_W – Recall score weighted average  
F1\_MI – F1 score micro average  
F1\_MA – F1 score macro average  
F1\_W – F1 score weighted average  
R – Runtime in seconds (training time + prediction time)

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Classifier** | **Top N** | **Dataset** | **P\_MI** | **P\_MA** | **P\_W** | **R\_MI** | **R\_MA** | **R\_W** | **F1\_MI** | **F1\_MA** | **F1\_W** | **Runtime** |
| DT\_topics | 100 | test | 0.282 | 0.178 | 0.26 | 0.282 | 0.161 | 0.282 | 0.282 | 0.162 | 0.26 | 0.009 |
| DT\_topics | 100 | train | 0.351 | 0.189 | 0.306 | 0.351 | 0.193 | 0.351 | 0.351 | 0.186 | 0.318 | 0.009 |
| DT\_topics | 200 | test | 0.3 | 0.187 | 0.279 | 0.3 | 0.163 | 0.3 | 0.3 | 0.164 | 0.27 | 0.021 |
| DT\_topics | 200 | train | 0.385 | 0.242 | 0.367 | 0.385 | 0.219 | 0.385 | 0.385 | 0.221 | 0.357 | 0.022 |
| DT\_topics | 400 | test | 0.3 | 0.187 | 0.279 | 0.3 | 0.163 | 0.3 | 0.3 | 0.164 | 0.27 | 0.034 |
| DT\_topics | 400 | train | 0.385 | 0.242 | 0.367 | 0.385 | 0.219 | 0.385 | 0.385 | 0.221 | 0.357 | 0.035 |
| DT\_sentiment | 100 | test | 0.708 | 0.433 | 0.646 | 0.708 | 0.419 | 0.708 | 0.708 | 0.412 | 0.661 | 0.008 |
| DT\_sentiment | 100 | train | 0.69 | 0.423 | 0.612 | 0.69 | 0.423 | 0.69 | 0.69 | 0.41 | 0.636 | 0.009 |
| DT\_sentiment | 200 | test | 0.672 | 0.396 | 0.59 | 0.672 | 0.406 | 0.672 | 0.672 | 0.391 | 0.619 | 0.014 |
| DT\_sentiment | 200 | train | 0.705 | 0.44 | 0.638 | 0.705 | 0.44 | 0.705 | 0.705 | 0.431 | 0.66 | 0.014 |
| DT\_sentiment | 500 | test | 0.65 | 0.392 | 0.573 | 0.65 | 0.395 | 0.65 | 0.65 | 0.377 | 0.591 | 0.028 |
| DT\_sentiment | 500 | train | 0.703 | 0.435 | 0.632 | 0.703 | 0.427 | 0.703 | 0.703 | 0.418 | 0.652 | 0.029 |
| BNB\_topics | 100 | test | 0.27 | 0.175 | 0.27 | 0.27 | 0.15 | 0.27 | 0.27 | 0.155 | 0.257 | 0.012 |
| BNB\_topics | 100 | train | 0.407 | 0.367 | 0.421 | 0.407 | 0.263 | 0.407 | 0.407 | 0.286 | 0.396 | 0.013 |
| BNB\_topics | 200 | test | 0.328 | 0.211 | 0.312 | 0.328 | 0.181 | 0.328 | 0.328 | 0.184 | 0.305 | 0.013 |
| BNB\_topics | 200 | train | 0.506 | 0.473 | 0.521 | 0.506 | 0.326 | 0.506 | 0.506 | 0.353 | 0.489 | 0.014 |
| BNB\_topics | 600 | test | 0.35 | 0.195 | 0.332 | 0.35 | 0.176 | 0.35 | 0.35 | 0.17 | 0.313 | 0.032 |
| BNB\_topics | 600 | train | 0.581 | 0.487 | 0.596 | 0.581 | 0.324 | 0.581 | 0.581 | 0.332 | 0.542 | 0.038 |
| BNB\_topics | 900 | test | 0.336 | 0.204 | 0.345 | 0.336 | 0.159 | 0.336 | 0.336 | 0.156 | 0.292 | 0.027 |
| BNB\_topics | 900 | train | 0.569 | 0.383 | 0.553 | 0.569 | 0.29 | 0.569 | 0.569 | 0.287 | 0.516 | 0.0403 |
| BNB\_sentiment | 100 | test | 0.72 | 0.554 | 0.688 | 0.72 | 0.489 | 0.72 | 0.72 | 0.505 | 0.695 | 0.008 |
| BNB\_sentiment | 100 | train | 0.719 | 0.641 | 0.701 | 0.719 | 0.547 | 0.719 | 0.719 | 0.576 | 0.7 | 0.01 |
| BNB\_sentiment | 200 | test | 0.724 | 0.615 | 0.708 | 0.724 | 0.537 | 0.724 | 0.724 | 0.558 | 0.711 | 0.0098 |
| BNB\_sentiment | 200 | train | 0.757 | 0.704 | 0.75 | 0.757 | 0.647 | 0.757 | 0.757 | 0.671 | 0.751 | 0.0118 |
| BNB\_sentiment | 400 | test | 0.716 | 0.651 | 0.709 | 0.716 | 0.544 | 0.716 | 0.716 | 0.572 | 0.706 | 0.017 |
| BNB\_sentiment | 400 | train | 0.785 | 0.752 | 0.781 | 0.785 | 0.668 | 0.785 | 0.785 | 0.699 | 0.78 | 0.022 |
| BNB\_sentiment | 1000 | test | 0.732 | 0.732 | 0.738 | 0.732 | 0.525 | 0.732 | 0.732 | 0.545 | 0.714 | 0.04 |
| BNB\_sentiment | 1000 | train | 0.845 | 0.854 | 0.848 | 0.845 | 0.707 | 0.845 | 0.845 | 0.747 | 0.839 | 0.052 |
| MNB\_topics | 100 | test | 0.254 | 0.177 | 0.257 | 0.254 | 0.141 | 0.254 | 0.254 | 0.147 | 0.241 | 0.005 |
| MNB\_topics | 100 | train | 0.407 | 0.4 | 0.431 | 0.407 | 0.262 | 0.407 | 0.407 | 0.289 | 0.394 | 0.007 |
| MNB\_topics | 200 | test | 0.328 | 0.219 | 0.314 | 0.328 | 0.194 | 0.328 | 0.328 | 0.199 | 0.31 | 0.01 |
| MNB\_topics | 200 | train | 0.525 | 0.516 | 0.542 | 0.525 | 0.385 | 0.525 | 0.525 | 0.418 | 0.517 | 0.011 |
| MNB\_topics | 400 | test | 0.354 | 0.225 | 0.341 | 0.354 | 0.211 | 0.354 | 0.354 | 0.213 | 0.34 | 0.0096 |
| MNB\_topics | 400 | train | 0.622 | 0.655 | 0.646 | 0.622 | 0.456 | 0.622 | 0.622 | 0.497 | 0.61 | 0.0116 |
| MNB\_topics | 1000 | test | 0.362 | 0.216 | 0.355 | 0.362 | 0.196 | 0.362 | 0.362 | 0.196 | 0.342 | 0.024 |
| MNB\_topics | 1000 | train | 0.707 | 0.724 | 0.728 | 0.707 | 0.501 | 0.707 | 0.707 | 0.547 | 0.692 | 0.029 |
| MNB\_topics | 1500 | test | 0.348 | 0.205 | 0.344 | 0.348 | 0.18 | 0.348 | 0.348 | 0.181 | 0.324 | 0.027 |
| MNB\_topics | 1500 | train | 0.715 | 0.736 | 0.74 | 0.715 | 0.497 | 0.715 | 0.715 | 0.542 | 0.699 | 0.035 |
| MNB\_sentiment | 100 | test | 0.724 | 0.56 | 0.689 | 0.724 | 0.475 | 0.724 | 0.724 | 0.491 | 0.691 | 0.007 |
| MNB\_sentiment | 100 | train | 0.723 | 0.663 | 0.707 | 0.723 | 0.548 | 0.723 | 0.723 | 0.581 | 0.702 | 0.007 |
| MNB\_sentiment | 200 | test | 0.736 | 0.646 | 0.719 | 0.736 | 0.534 | 0.736 | 0.736 | 0.559 | 0.719 | 0.0065 |
| MNB\_sentiment | 200 | train | 0.757 | 0.702 | 0.748 | 0.757 | 0.629 | 0.757 | 0.757 | 0.657 | 0.749 | 0.0085 |
| MNB\_sentiment | 400 | test | 0.734 | 0.689 | 0.727 | 0.734 | 0.584 | 0.734 | 0.734 | 0.618 | 0.725 | 0.018 |
| MNB\_sentiment | 400 | train | 0.795 | 0.763 | 0.791 | 0.795 | 0.695 | 0.795 | 0.795 | 0.723 | 0.791 | 0.018 |
| MNB\_sentiment | 1000 | test | 0.746 | 0.7 | 0.741 | 0.746 | 0.565 | 0.746 | 0.746 | 0.595 | 0.734 | 0.017 |
| MNB\_sentiment | 1000 | train | 0.858 | 0.851 | 0.858 | 0.858 | 0.782 | 0.858 | 0.858 | 0.811 | 0.856 | 0.019 |
| MNB\_sentiment | 1500 | test | 0.734 | 0.677 | 0.737 | 0.734 | 0.571 | 0.734 | 0.734 | 0.591 | 0.727 | 0.0225 |
| MNB\_sentiment | 1500 | train | 0.883 | 0.883 | 0.882 | 0.883 | 0.801 | 0.883 | 0.883 | 0.834 | 0.881 | 0.0295 |

Table 1

From the table above, all classifiers seem perform better when predicting the classes of training sets because these data is used to train the models. When N word is 100, all sentiment classifiers have similar performance. When the word is doubled, the performance for decision tree sentiment classification is remain rather similar but is marginally increased for the two Naïve Bayes methods. Overall, for sentiment classification, the three models perform equally well. This is likely because, there are only three predictable classes for sentiment, namely *negative, neutral and positive.*

Regarding runtimes, decision tree classifiers are the slowest of all classifiers. This is as expected because DT models are known to be more complex for certain domains than NB models. When comparing the performance of topics classifiers, F1\_MI (micro) metric is considered because it produces a high result when precision and recalled is balanced. This is true for the table above. Micro average is considered because there is class imbalance. The F1 scores show that in general NB models are better at classifying topics that DT models.

In a multiclass problem such as voting topics classification, each class is not equally important because I believe that voters did not concern themselves with every topic being discussed. Thus, in my opinion, micro setting is more suited because it gives each observation an equal weight rather than macro setting which gives each class an equal weight which may not necessarily be true in Federal Election because some matters were more pressing than others.

1. (2 marks) Evaluate the standard models with respect to baseline predictors (**VADER** for sentiment analysis, majority class for both classifiers). Comment on the performance of the baselines and of the methods relative to the baselines.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Baseline | Accuracy | F1 Micro Avg | F1 Macro Avg | F1 Weighted | Runtime (sec) |
| Majority class topics | 0.174 | 0.17 | 0.01 | 0.05 | 0.0130 |
| Majority class sentiment | 0.648 | 0.65 | 0.26 | 0.51 | 0.0110 |
| VADER sentiment | 0.430 | 0.43 | 0.37 | 0.48 | 0.2200 |

Table 2

Table 2 above shows the performance of the baseline classifiers. Both majority class classifiers were trained using first 1500 tweets and tests against last 500 tweets. VADER was also used to predict the last 500 tweets. The results show that the performance majority class topics classifier was very poor, and that majority class sentiment classifier performed better than VADER.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Standard model | Vocab size | Accuracy | F1 Micro | F1 Macro | F1 Weighted |
| DT\_topics | 200 | 0.296 | 0.30 | 0.17 | 0.27 |
| BNB\_topics | All (6907) | 0.178 | 0.18 | 0.02 | 0.06 |
| MNB\_topics | All (6907) | 0.290 | 0.29 | 0.12 | 0.25 |
| DT\_sentiment | 200 | 0.672 | 0.67 | 0.40 | 0.62 |
| BNB\_sentiment | All (6907) | 0.716 | 0.72 | 0.40 | 0.65 |
| MNB\_sentiment | All (6907) | 0.73 | 0.73 | 0.52 | 0.71 |

Table 3

Table 3 shows the performance of six standard models. These six models clearly outperformed the baseline models in all metrics. BNB\_topics and MNB\_topics would perform marginally better than majority class baseline when the vocab size is smaller (refer to table 1).

1. (2 marks) Evaluate the effect that preprocessing the input features, in particular stop word removal plus Porter stemming as implemented in **NLTK**, has on classifier performance, for the three standard methods for both sentiment and topic classification. Compare results with and without preprocessing on training and test sets and comment on any similarities and differences.



Figure Preprocessing performance comparison on testing set

The figure above was plotted using the data in Appendix A with 200 words vocabulary size. The figure shows every topic classifier benefitted from stop word removal and Porter stemming especially BNB\_topics which benefited the most from preprocessing. Without preprocessing, BNB\_topics performed slightly worse than MNB\_topics and slightly better. But with preprocessing, BNB\_topics became the best performer among the three topics classifiers. Sentiment classifiers seem to suffer from preprocessing although not by much. Preprocessing had no significant effect on MNB\_sentiment.

/\*INSERT TRAINING DATA\*/

1. (2 marks) Sentiment classification of neutral tweets is notoriously difficult. Repeat the experiments of items 2 (with N = 200), 3 and 4 for sentiment analysis with the standard models using only the positive and negative tweets (i.e. removing neutral tweets from both training and test sets). Compare these results to the previous results. Is there any difference in the metrics for either of the classes (i.e. consider positive and negative classes individually)?
2. (6 marks) Describe your best method for sentiment analysis and your best method for topic classification. Give some experimental results showing how you arrived at your methods. Now provide a brief comparison of your methods in relation to the standard methods and the baselines.

# Appendix A

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Classifier | Stem & Stop | Accuracy | F1\_Micro | F1\_Macro | F1\_Weighted |
| DT\_topics | yes | 0.348 | 0.35 | 0.22 | 0.33 |
| DT\_sentiment | yes | 0.654 | 0.65 | 0.39 | 0.62 |
| BNB\_topics | yes | 0.458 | 0.46 | 0.27 | 0.43 |
| BNB\_sentiment | yes | 0.708 | 0.71 | 0.42 | 0.66 |
| MNB\_topics | yes | 0.44 | 0.44 | 0.27 | 0.42 |
| MNB\_sentiment | yes | 0.728 | 0.73 | 0.57 | 0.72 |
| DT\_topics | no | 0.3 | 0.3 | 0.17 | 0.27 |
| DT\_sentiment | no | 0.69 | 0.69 | 0.45 | 0.65 |
| BNB\_topics | no | 0.33 | 0.33 | 0.18 | 0.3 |
| BNB\_sentiment | no | 0.72 | 0.72 | 0.56 | 0.71 |
| MNB\_topics | no | 0.35 | 0.35 | 0.2 | 0.33 |
| MNB\_sentiment | no | 0.736 | 0.74 | 0.56 | 0.72 |