COMP30027 Project 2 Report

1. Introduction

The goal of this Project is to build and critically analyse Naïve Bayes classifiers with the aim of automatically identifying the location from which a textual message (tweet) was sent, as one of four Australian cities.

2. Related Work

Previous work [1] has shown that Bernoulli NB classifier (BNB) and Multinomial NB (MNB) classifier have different effectiveness and performance on text classification. More importantly, it verifies that MNB outperforms BNB in cases where the training set is relatively large. Hence, in this paper, the effectiveness and performance of both MNB and BNB will be tested on a different text classification problem, that is the geolocability of tweets.

3. Feature Engineering

The datasets provided are extracted, pre-processed, and collected from twitters. They are in the format of "<instance id> <space> <text> <space> <location>".

There are several popular feature engineering methods used in this project. Firstly, count vectorizer is used to break the string of text down to individual words (tokenization), which then the frequency of each individual word is counted and recorded to be use as the features.

Secondly, tf-idf (term frequency – inverse document frequency) vectorizer is used, similarly it extracts individual words from the text, and counts the frequencies of words, however it normalizes the overall words frequencies to minimize the weight of common words.

These two methods are used to transform the training set to produce two individual training sets, and also to transform development set to produce two test sets.

4. Classifications

This project will be looking at two classification techniques, Multi-variate Bernoulli Naïve Bayes Classifiers (BNB) and Multinomial Naïve Bayes Classifier (MNB).

4.1. Multi-variate Bernoulli Naïve Bayes Classifier

The model is based on Bernoulli probabilistic models where every outcome only has two scenarios. In the case of the project, every token (word) in the feature vector of a document (instance) is associated with a value of 0 or 1. The value of 1 means that the token occurs in that particular document; the value of 0 indicates the token does not occurs in the document [2].

4.2. Multinomial Naïve Bayes Classifier

This model takes an alternative approach to represent the documents, that is unlike binary values, it associates each token with a term frequency. For a given term, its number of its occurrence in a document is counted and assigned to the term. [3]

5. Comparison on the classifiers' behaviours

5.1. Dataset Size

BNB only classify tokens in a document as occurs or non-occurs, it does not gain more information when it encounters the same word multiple times in the training phase. Hence, assuming the documents are randomly distributed among the training set such that an arbitrary group of documents selected from the training set will have similar distribution as the whole training set, BNB requires a relatively small dataset for it to capture the distribution of words required in order for it to predict relatively well and accurate.

On the contrary, MNB requires a large dataset for it learn about the tokens' distribution in the dataset well enough to make a relatively accurate prediction.

5.2. Information capture

Consider the following case for BNB, for example, for a word that has high frequency in a document, the posterior probability for the particular word will be the same as a word that has low frequency in the document as they both occur in the document. Hence, a high-frequency word is just as informative as a low-frequency word to the classifier.

On the contrary, MNB is able to gain more information by using the term frequency to adjust to a more accurate prediction.

6. Results

As a baseline, one-layer decision tree is implemented that acts as one-r classifier which it achieved an accuracy value of 26.2%. This result is unsurprising as all the classes have fairly distributed equally, with 4 classes, each class consists of roughly 25% of the total labels. The figure 1 shows the distribution.

Labels	Number of labels
Brisbane	25841
Sydney	25841
Perth	25840
Melbourne	25838

Figure 1.

Each classifier is trained on two feature sets, that one is count-vectorized while another is tf-idf-vectorized, they are then put to be evaluated iteratively to find out their optimum hyperparameters that yield the highest accuracy.
The following figure 2 shows the how different values of smoothing value affect the accuracies of the classifiers. Note: Since BNB have the exact same metrics scores in both feature sets (figure 3), this figure 2 will only show BNB.

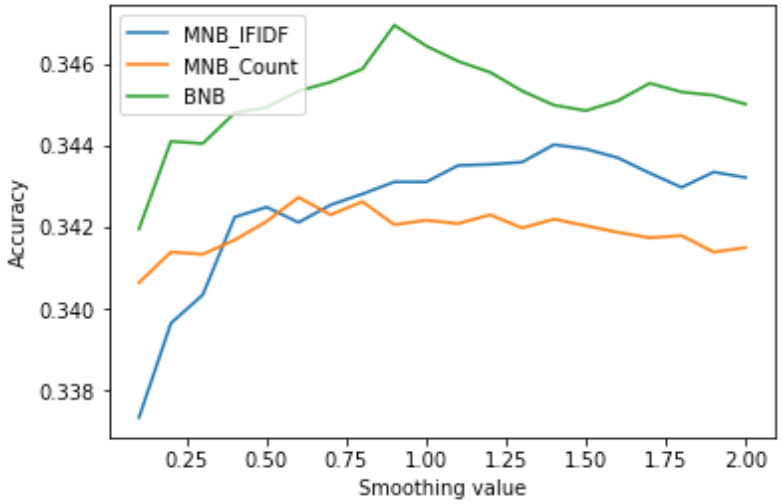

Figure 2.

In Naïve Bayes classifier, the additive smoothing value is very important as it can heavily affect the classifier's performance. Overestimate the smoothing value means overestimating the probability of an unforeseen feature, which leads to inaccurate representation of the distribution of the features in the training set, hence ultimately it affects the performance of the classifier. Similarly, underestimate the smoothing value can impact the classifier's performance as well.
From the figure 2.0, it can be seen that BNB has peak accuracy at smoothing value of around 0.9, while having any other smoothing value seems to worsen its accuracy performance. Similarly, this can be observed for both cases in MNB where optimum smoothing values produce the best accuracy value.

The following figure 3 shows the metrics for both classifiers trained and tested with optimum hyperparameters on both feature sets. More specifically, the metrics

include weighted accuracy, weighted precision, weighted recall, and weighted f-score.

Classifier	Smoothing	Accuracy	Precision	Recall	F-Score
Bernoulli NB (IF-IDF Vectorized)	0.9	0.34694	0.37249	0.34694	0.35148
Bernoulli NB (Count Vectorized)	0.9	0.34694	0.37249	0.34694	0.35148
Multinomial NB (IF-IDF Vectorized)	1.35	0.34410	0.35583	0.34410	0.34619
Multinomial NB (Count Vectorized)	0.6	0.34273	0.40121	0.34273	0.35253

Figure 3.

Firstly, BNB has the same metrics values in both feature sets, which agrees with the behaviour of the classifier that it does not recognize the term frequency, but the term occurrence instead.

Moreover, both the classifiers seem to have similar performance, with Bernoulli NB has the highest accuracy, however it is only insignificantly higher than MNB in both cases. This suggests that while the training dataset is not large enough to cover the distribution of words in the test set, BNB is able to learn enough information for its model to perform better than MNB in both cases. Hence, this result seems to agree with the comparison discussed above in section 5.

It has been stated that Naïve Bayes is extremely sensitive to features selection [4][5][6]. In order to investigate how the features selection affects the performance of both BNB and MNB, a chi-square value has been calculated for all features, different top percentiles of the features with the best scores are then used to train the classifiers. The results are as shown in the figure 4 below. Note: Since BNB have the exact same metrics scores in both feature sets (figure 3), this figure 4 will only show BNB.

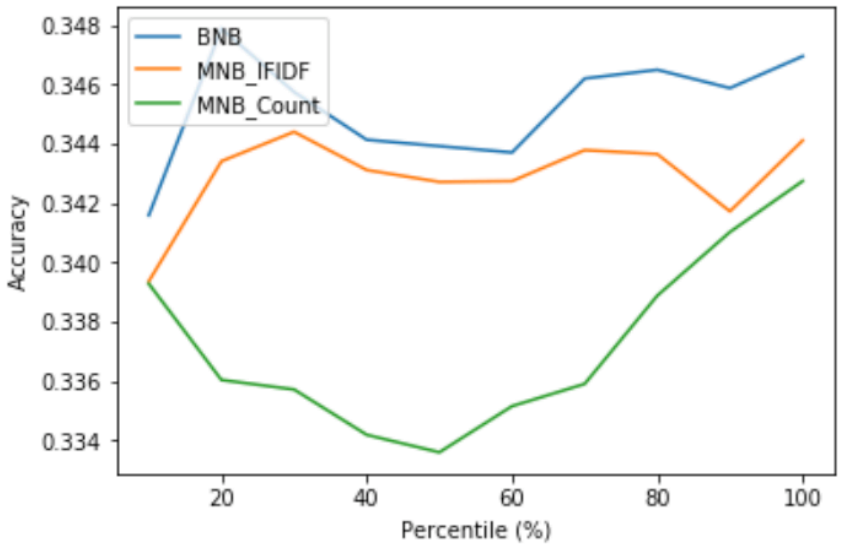

Figure 4.

Firstly, it can be noticed that BNB had a peak accuracy of 34.79% at around 20 percentiles. This suggests that BNB could in fact perform quite well with small datasets that contains good features while both MNB have been outperformed.

7. Error analysis

Upon closer inspection of the data, there are username, hashtag, and hyperlinks embedded in the features. It can be argued that usernames are unique and can be associated with multiple locations which if included in the feature sets could skew the classifier's behaviour. Likewise, hashtag and hyperlinks could also affect the classifiers in the similar way.

Hence, several tests were made to investigate the effects of usernames, hashtag, and hyperlinks that are embedded in the features. The results are shown in the figure 5 below.

Classifiers	Accuracy (w/o usernames)	Accuracy (w/o hashtags)	Accuracy (w/o hyperlinks)
BNB	0.33289	0.33370	0.34767
MNB_TFIDF	0.33353	0.33155	0.34327
MNB_COUNT	0.32860	0.33182	0.34166

Figure 5.

It can be seen that by removing usernames, hashtags, or hyperlinks individually in the features, it reduces the accuracies for the classifiers in almost all cases, except for BNB with hyperlinks excluded in the features had its accuracy improved to 34.76%, however the improvement is relatively small and one can argue that it is caused by overfitting. Hence, this suggests that the use of usernames, hashtags, and hyperlinks in the features helps to improve the classifiers' performances generally.

A more thorough error analysis is therefore needed to identify the particular traits in the features that might lead to the misclassification.

8. Conclusion

In this paper, we examined the effectiveness of both BNB and MNB classifiers on the issue of geolocating tweets. Contrary to the results shown in the previous work, BNB has slightly outperformed MNB in both count vectorized and tf_idf vectorized feature sets in terms of accuracy, and that the limited distribution of words in the training sets is suspected to be the reason. Furthermore, error analysis has shown that unique tokens such as username, hashtags, and hyperlinks in the feature set helps to improve the performance of both classifiers.

References

[1] A. McCallum, K. Nigam, A Comparison of Event models for Naïve Bayes Text Classification, AAAI-98 Workshop on Learning for Text Categorization, 1998.

[2] W. Zhang, G. Feng, An Improvement to Naïve Bayes for Text Classification, SciVerse ScienceDirect, 2011.

[3] S. Raschka, Naïve Bayes and Text Classification, Introduction and Theory, 2014.

[4] Y. Yang and J. O. Pedersen. A comparative study on feature selection in text categorization. In International Conference on Machine Learning, pages 412–420, 1997.

[5] Rogati, M.; Yang, Y. High-performing feature selection for text classification. CIKM'02, 2002, pp 659-661.

[6] J. Chen, H. Huang, S. Tian, and Y. Qu, "Feature selection for text classification with Naïve Bayes," Expert Systems with Applications, vol. 36, no. 3, pp. 5432-5435, April 2009.