# Classification Using Various Machine Learning Methods to Predict Tweet Locations

Patrick Tjahjadi
Maurice Bernardo Edwin

## 1. Introduction

Over the past few years, the number of social media users have increased drastically, leading to an awash of available online data. With the ever-increasing amount of content available online, it becomes more valuable to be able to make sense of such data. In the context of Twitter, machine learning methods allow us to receive data about a user with regards to their tweets.

This report attempts to assess supervised machine learning methods that are able to identify a user's location based on their tweets. The aim is to identify the location from which a textual message was sent – a simplification of geotagging. In addition, this report uses a variety of machine learning methods and assess the accuracy of each of these classifiers during its validation process. This report also investigates the effectiveness of an ensemble learning method to create a combined stacked classifier and assess its capabilities in this simplified approach to geotagging.

## 2. Related Literature

There are numerous studies of the effectiveness of classifiers with regards to social media, and different literatures and articles provide generally different approaches towards classification. It considers the work of Pappas, Azab and Mihalcea in 2018 that attempts to identify the location of a Twitter user based on their tweets.

### 2.1. Underlying Concepts

In this report, several supervised classification methods are used. Almeida and Alberto in 2013 provide different approaches towards detecting undesired online comments using various machine learning methods.

### 2.1.1. Naïve Bayes

The Naïve Bayes classifier is used to predict the class of available instances through assessing its probabilities but can also be used for other machine learning applications such as clustering (Lowd, 2005).

This report provides multiple types of Naïve Bayes, in which multinomial (with Laplace smoothing), Gaussian and Bernoulli classifiers were used, respectively with the assumption that the probability of each word is independent of the word's context and position (McCallum and Nigam, 1998).

### 2.1.2. Decision Trees/Random Forests

Decision trees are more useful than other conditional probability classifiers since they allow conditional independence and don't need to store a separate probability for every combination of parent variable (Lowd, 2005). The model variance of using decision trees have a twofold impact towards evaluation but would usually tend towards a relatively higher accuracy than other classifiers. This report is also inspired by Almeida and Alberto in 2013 where they use random forests to determine the majority vote from the combination of decision trees. The overall model variance can be reduced compared to using the decision of individual trees, which leads to more accurate decisions without the need for pruning (Biau, 2012).

### 2.1.3. Ensemble Learning Methods

This report uses various ensemble learning methods such as bagging and stacking. Bagging is used for decision trees to create random forests to reduce model variance (DeFilippi, 2018). Stacking allows the creation of a meta-classifier that benefits from learning the behavior of other classifiers (DeFilippi, 2018).

### 2.2. Results Discussion

The Naïve Bayes classifier has been fortunately proven and agreed to be one of the most used classifiers due to its simplicity and performance (Almeida and Alberto, 2013) (Hoang, Moriceau and Mothe, 2017). Random forests have also been used as a great alternative due to its high feature space and low variance, providing higher accuracy than other classifiers in predicting locations of tweets (Biau and Scornet, 2015) (Hoang, Moriceau and Mothe, 2017). This report also uses two stacked learners that benefit from bagging. Although it is initially expected that the stacked learner would perform better than individual learners because it benefits from the strengths of each individual classifier, a study by Dzeroski and Zenko in 2004 reaffirms that stacked learners perform comparably as the best selected classifier used for the stacking process.

## 3. Methodology

### 3.1 Data and Feature Selection

Initially, the data for this report is received from Twitter using the Twitter API. It contains the training, development and test datasets of raw tweets, and the top 10, 50 and 100 features used to classify tweets into 4 Australian cities: Sydney, Melbourne, Brisbane and Perth.

The intuition of this simplified geotagging procedure derives from the notion that the lexicon contained in each tweet has a correlation with its location. This has been proven by Pappas, Azab and Mihalcea's research in 2018, where they predicted the US state in which the tweet was made by assessing its lexical model using various classifiers.

In our attempt to increase the accuracy, the data is pre-processed with custom tokens. This report excludes non-alphabetic characters, non-Latin characters, URLs and other features where their frequency are less than 10 tweets. This method takes insight from the work of Pappas, Azab and Mihalcea in 2018 and Mahmud, Nichols and Drews in 2012. They affirm that these feature selection methods work well in predicting locations because most of these exclusions do not contribute to the classifier's evaluation. This report extends further by lowercasing all words. Moreover, shortened versions of city names are generalized (e.g. 'melb' is categorized as 'melbourne').

| City | Lexicon |
|---|---|
| Sydney | sydney, nsw, ovoawl |
| Melbourne | melbourne, victoria, temperature |
| Brisbane | brisbane, queensland, qld |
| Perth | perth, annemarie, wa |

Table 1: Sample words in the city lexicons

### 3.2 Classifier Assemblage

The concepts discussed in the literatures will be the classifiers used for this report and its accuracies will be assessed and compared. The results will then be used as a principal constituent to perform error analysis.

Naïve Bayes and decision trees were chosen because various literatures have confirmed its effectiveness (Almeida and Alberto, 2013) (Biau and Scornet, 2015) (Hoang, Moriceau and Mothe, 2017). The Naïve Bayes classifier has three variations, albeit being categorized as one classifier. In this report, the multinomial Naïve Bayes has a hyperparameter of alpha set to 1, which implements Laplace smoothing. The decision trees are given an entropy criterion with no maximum depth, so it considers information gain.

Moreover, this report attempts to combine the Naïve Bayes and random forests classifiers into a stacked classifier using stacked ensemble learning. Then, the meta-classifier that learns from these classifiers is a decision tree with no maximum depth using entropy criterion. This stacked classifier's accuracy will then be assessed and is initially expected to perform better than other classifiers. All classifiers are built using scikit-learn and implemented into this report using some default and custom hyperparameters.

## 4. Results and Analysis

### 4.1. Evaluation

The classifiers were evaluated using the evaluation dataset with both default and pre-processed tokens. The pre-processing method involves retrieving the top 34, 185 and 2000 combined features. The first top 34 and 185 features were based on the number of features of the top 10 and 50 default tokens for each city. The top 2000 features were taken as it is deemed to be the most appropriate in classifying the location of the tweets because it allows the classifier to predict from a wider range of features, or considerations.

| **Classifier** | **Top10** | **Top50** | **Top100** |
|---|---|---|---|
| Binomial NB | 29.483 | 30.07 | 30.702 |
| Multinomial NB | 29.491 | 30.135 | 30.783 |
| Gaussian NB | 29.481 | 29.842 | 30.145 |
| Decision Tree | 29.553 | 30.086 | 30.719 |
| Stacked Learner | 29.556 | 29.805 | 30.164 |
| Bagged Multinomial NB | 28.952 | 29.395 | 30.389 |

Table 2: Accuracy of the classifiers in the evaluation dataset in percentage with default tokens.

| Classifier | Top34 | Top185 | Top2000 |
|---|---|---|---|
| Binomial NB | 31.525 | 31.63 | 32.324 |
| Multinomial NB | 31.549 | 31.691 | 32.8 |
| Gaussian NB | 28.746 | 29.223 | 28.47 |
| Decision Tree | 31.705 | 32.037 | 31.34 |
| Stacked Learner | 31.686 | 31.799 | 31.252 |
| Bagged Multinomial NB | 31.713 | 31.694 | 32.838 |

Table 3: Accuracy of the classifiers with preprocessed datasets in percentage

## 4.2. Interpretability and Analysis

Firstly, it is to note that this report does not consider several factors that might influence the decisions of the classifiers, such as the time when the tweet was sent and the distribution of Twitter users present in the dataset, as a small number of Twitter users would not indicate a fair representation of the Twitter population. Rather than only using lexical assessments, extensions to provide more contextual knowledge of a tweet would certainly contribute to the accurate decisions of the classifiers. The related literatures provided in this report have more attributes and context to help train the classifiers to evaluate their datasets better.

It can be seen from Tables 2 and 3 that the more features added for consideration, the more accurate the classifiers are in predicting the correct location. There are, however, some exceptions as shown in Table 3, where one plausible reason is that the difference in information gain from the top 34, 185 and 2000 datasets negatively affect the decision tree in predicting the correct location. Whereas the top 185 features heavily affect the decision of the classifier, adding extra features into consideration only adds noise and overfits the decision tree, which in turn, reduces the validation accuracy.

Tables 2 and 3 suggest that among all Naïve Bayes types, the multinomial Naïve Bayes is the most accurate. This is because multinomial Naïve Bayes takes account of attributes which are based on natural numbers corresponding to frequencies, which suits the most in this context (Gandhi, 2018). The accuracy of the multinomial Naïve Bayes classifier can further be improved by implementing bagging with the hyperparameters of taking 35% of the maximum samples and 90% of the maximum features, because the dataset involves a lot of tweets that do not match any selected features and taking a small portion of the samples allows the removal of such noise, allowing the Naïve Bayes classifier to predict more accurately. However, there are still some cases where the predicted accuracy may be lower than without bagging, since the samples and features are taken randomly, and might skip instances or features that are helpful in training the classifier.

It should be noted that the analysis of the Naïve Bayes classifiers concerns a discrete frequency count of how many tweets have a certain number of features present within them, as shown in Figures 1 and 2. As the frequencies are not normally distributed, Gaussian Naïve Bayes, which consider the feature to be normally distributed and frequencies to be continuous, is not a good fit (Gandhi, 2018).

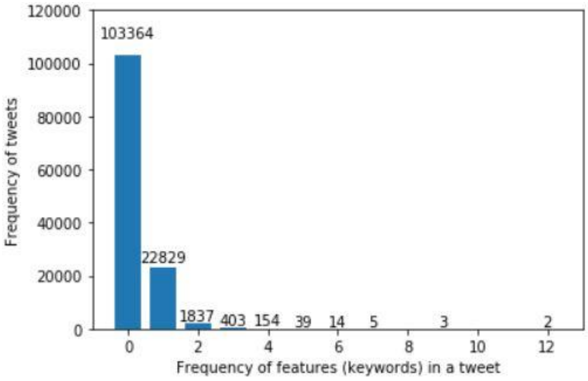

Figure 1: Detailed plot of the frequency of keywords occurring in a tweet using default tokens in the evaluation dataset.

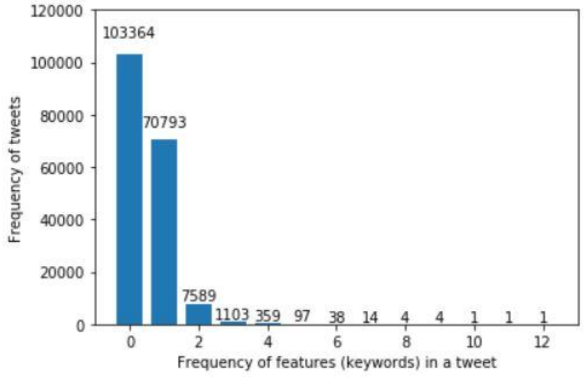

Figure 2: Detailed plot of the frequency of keywords occurring in a tweet using pre-processed tokens in the evaluation dataset.

Note that there are more features included when using pre-processed tokens as shown in Figure 2 compared to using default tokens as shown in Figure 1, which allows more features to be considered by the classifiers during evaluation. This ultimately increases the accuracy, since it prevents the classifier from degrading into 0-R performance, which predicts solely based on the majority class when no selected features are present in a tweet.

Moreover, there are many tweets that have more than one frequency of keywords occurring. The frequency of keywords, should it be more than 1, should affect more to the location of the tweet, which Binomial Naïve Bayes ignores (Gandhi, 2018). Figures 1 and 2 both suggest that Binomial Naïve Bayes disregards many features that may influence the decision of the classifier to make more accurate predictions. Therefore, Binomial Naïve Bayes does not fit well in this context.

Uniquely, the accuracy of the decision trees and stacked learner varies for every attempt in evaluating its accuracy. The declared decision tree algorithm from the sci-kit library randomizes the order of each decision tree level, and this difference consequently randomizes the sub-sample which causes the structure of the decision tree to change for every attempt, leading to different predictions and accuracies. Decision trees also suffer from overfitting, and the accuracy of the training and evaluation datasets differ greatly (Trivedi, 2017).

In order to stabilize and remove the tendency of decision trees to overfit, this report uses Random Forests as a bagged classifier to reduce the model variance (DeFilippi, 2018). This random forest classifier is then stacked with a bagged multinomial Naïve Bayes classifier and has a similar performance with decision trees. It should be noted that the accuracy differs from every attempt, similar to the decision tree's behavior, possibly due to the same aforementioned reasons. Overall, they perform relatively well, but not better than the best performing individual classifier (multinomial Naïve Bayes). This is further backed by Dzeroski and Zenko in 2004, stating that a stacked learner does not necessarily perform better than individual classifiers.

Overall, the restricted training dataset and variability of the evaluation dataset impedes the performance of the classifiers. The distribution of keywords present in the training dataset varies greatly, and there are not many words that explicitly show the location where the tweet was sent from, unless the words have a direct relationship to a particular Australian city or there is a certain trend in which only one city follows as shown in Table 1.

## 5. Conclusions

There is no single classifier that can be selected to perform well on all evaluation contexts. However, this report suggests that the Naïve Bayes, decision trees and random forests classifiers have been proven from both literary texts and this report's evaluation methods to be the most effective classifiers to perform discrete categorical classification (Almeida and Alberto, 2013) (Biau and Scornet, 2015) (Hoang, Moriceau and Mothe, 2017) .

Ultimately, additional context and information of a tweet is required in order to improve the classifier's accuracy. With the given dataset, this report is restricted to only being able to improve its classifiers by finding more advanced methods of feature selection and preprocessing. Understanding the behavior of users and finding more information of a tweet, such as the demographics of Twitter users is crucial to improve the accuracy of locating tweets (Pappas, Azab and Mihalcea, 2018).

In conclusion, predicting locations of tweets using solely lexical analysis would only assist the classifier in prediction to a limited extent. There are also many considerations that need to be examined regarding the strengths and weaknesses of each classifier used in this report. Various ensemble methods aimed at reducing the bias and variance and improving the predictive force of the models do not guarantee an increase in accuracy. Improvements can be done by using more sophisticated feature selection criteria and adding more context in the given training dataset.

## References

Almeida, T., & Alberta, T. (2013). *Learning to Block Undesired Comments in the Blogosphere* (pp. 261-266). Sao Paulo: Federal University of São Carlos.

Biau, G. (2012). *Analysis of a Random Forests Model* (pp. 1064-1072). Paris: Universite Pierre et Marie Curie.

Biau, G., & Scornet, E. (2015). *A Random Forest Guided Tour* (pp. 1-35). Paris: Sorbonne University.

DeFilippi, R. (2018). Boosting, Bagging, and Stacking — Ensemble Methods with sklearn and mlens. Retrieved from https://medium.com/@rrfd/boosting-bagging-and-stacking-ensemble-methods-with-sklearn-and-mlens-a455c0c982de

Dzeroski, S., & Zenko, B. (2004). *Is Combining Classifiers with Stacking Better than Selecting the Best One?* (pp. 256-273). Ljubljana: Jozef Stefan Institute.

Gandhi, R. (2018). Naive Bayes Classifier. Retrieved from https://towardsdatascience.com/naive-bayes-classifier-81d512f50a7c

Hoang, T., Moriceau, V., & Mothe, J. (2017). *Predicting Locations in Tweets*. Toulouse: University of Toulouse.

Lowd, D. (2005). *Naive Bayes Models for Probability Estimation* (pp. 1-7). Seattle: University of Washington.

Mahmud, J., Nichols, J., & Drews, C. (2012). *Where Is This Tweet From? Inferring Home Locations of Twitter Users* (pp. 511-514). San Jose: IBM Research.

McCallum, A., & Nigam, K. (1998). *A Comparison of Event Models for Naive Bayes Text Classification* (pp. 41-48). Pittsburgh.

Pappas, K., Azab, M., & Mihalcea, R. (2018). *A Comparative Analysis of Content-based Geolocation in Blogs and Tweets*. Ann Arbor: University of Michigan.

Trivedi, J. (2017). The Indecisive Decision Tree — Story of an emotional algorithm (1/2). Retrieved from https://towardsdatascience.com/the-indecisive-decision-tree-story-of-an-emotional-algorithm-1-2-8611eea7e397

POSTERA CRESCAM LAUDE

THE UNIVERSITY OF
MELBOURNE

Patrick Tjahjadi – ptjahjadi@student.unimelb.edu.au
Maurice Bernardo Edwin – mauricee@student.unimelb.edu.au