# Machine Learning Engineer Nanodegree

## Capstone Proposal

Sharon Zollner
September 6, 2017

## Domain Background

New Zealand is approaching what has become an unexpectedly close general election. The main opposition party changed its leader on 1 August and polling between the two main parties has become much closer since that time. Using machine learning I would like to analyse the texts of the press statements from the six parties most likely to be represented in Parliament following the election based on current polling (National, Labour, NZ First, the Greens, the Maori Party, and ACT).

Potential analysis falls into two broad sections: general text analysis to solve a classification problem and sentiment analysis. The classification problem should be able to be addressed with Multinomial Naïve Bayes (possibly with extensions such as TF-IDF conversion as outlined in Multinomial Naïve Bayes for text categorisation revisited, which led to better results in this study that is fairly analogous to my problem). I also intend to explore other potential avenues; eg support vector machines and convolutional neural network have been successfully used for textual analysis, but these are not part of the core project as currently planned. Sentiment analysis would require both labelled data and additional tools such as pre-trained NLTK classifiers and is beyond the scope of this project. However, some interesting qualitative conclusions may come out of investigation of which words characterise the different parties' statements.

## Problem Statement

Among themes I would like to explore are:

- Are the parties releases predictable enough that I can build a Naïve Bayes model that correctly identifies which party a given press release came from based on the words within it?
- Do the most significant features / simple word counts reflect my general expectation of the major topics that the parties fixate on?
- If the above prove to be dead ends, an interesting alternative would be unsupervised learning to cluster the press statements and see which parties appear most aligned by this measure. However, this is Plan B and not part of my proposal as it stands.

## Datasets and Inputs

The press releases for the parties are available on their website. The data is intended to be public, so there are no concerns on this front. I have already written and executed Python code using Beautiful Soup to download the texts. As at the time of writing there are the following number of releases since 1 July: National: 535, Labour: 123, NZ First: 169, Greens: 70, Maori: 36, ACT: 91 (total:

1024). This is an unbalanced dataset, which will require special techniques (see, for example An overview of classification algorithms for unbalanced datasets by Vaishali Ganganwar and Addressing the Problem of Unbalanced Data Sets in Sentiment Analysis by Asmaa Mountassir, Houda Benbrahim and Ilham Berrada, and Practical guide to deal with imbalanced classification problems in R by Analytics Vidhya). If results are not promising I may drop the three smaller parties. The data will need standard pre-processing (eg removing stop words) and also checking that non-meaningful identifying patterns are removed (eg one party's releases as web-scraped all start with "Posted by").

Some other patterns will be distinctly unsurprising and thus uninteresting, eg the Maori Party almost certainly uses a lot more Maori language words in their press releases – if I want to focus on extracting words as "themes" I may need to actually remove those words (say by using min_df slightly larger than the proportion of total that are Maori Party releases in my CountVectorizer, which would only consider words that are found in at least one document authored by another party) and re-run analysis as a separate exercise to get the most important English words in Maori Party releases. Feature selection is very important in text analysis but will need to be done carefully given the unbalanced data set.

Given the unbalanced classes, one option I may end up exploring is simply collecting more data releases from the smaller parties, but these would then cover a different time period and associated events, which gives the algorithm an "unfair" advantage in solving the classification problem – "unfair" in that the resulting key feature words would not be particularly informative as they would likely relate to events that occurred outside the time frame of the press releases of the the other parties. An alternative would be over-sampling, including multiple copies of the releases from the smaller parties. I could also randomly under-sample from the National Party but since my dataset is not large to start with, this would not be high up my list of options to explore. Another option would be splitting up the National Party releases by author, but it would be an oddly asymmetric treatment of the parties that puts the goal of achieving a high classification accuracy above all else, and since I am at least as interested in the words that prove to be key features, I am unlikely to go down this route. Generating synthetic samples is beyond the scope of this project.

## Solution Statement

In terms of techniques, to solve this problem I intend to start with Multinomial Naïve Bayes, the most common algorithm for this type of problem, because it copes well with large feature spaces and is straightforward to implement. I intend to consider extensions selected from options including TF-IDF conversion to upweight words that are rarer in the corpus, Bernoulli Naïve Bayes (which takes into account not only the words that are included but also the words that are *excluded)*, including bi-grams and tri-grams (2- or 3-word phrases) with ferocious feature selection necessitated, or even a totally different kind of model such as a support vector machine or neural network. I may combine different types of models using for example Adaboost or Community methods to let different models 'vote'. However, Naïve Bayes is an appealing approach in that it is very easy to extract which features are most relevant (unlike a black box neural network, for example), which in this exercise is arguably potentially the most interesting finding (after all, press releases always have the author clearly labelled!). If I hit a total dead end, I may change tack entirely and undertake an unsupervised clustering exercise to see which parties appear most aligned with each other, ie address a completely different question, but I hope this will not prove necessary!

## Benchmark Model

This is not the kind of project for which one can find an obvious benchmark model to assess performance. The dataset is new ("hot off the press" in fact) and has not – to my knowledge – been analysed before. However, this kind of text classification problem is a well-established methodology. This thesis, Word-based classification of political text based on speaker and topic labels is a fairly similar kind of analysis to what I intend to do, albeit with a more varied and larger dataset (nearly 5500 documents). The Naïve Bayes models achieved accuracy of around 65-70% when identifying the speaker, depending on the specification.

A low but data-specific benchmark for my model is simply whether it is more accurate than a random choice (but knowing the odds, eg picking the National party was the author 535/1024 of the time) would be on average.

I would consider this project as described a success if I were able to fairly accurately identify the (party) author of a given press statement from a held-out test set, and also if some of the high-ranking words that characterised each party's releases had some intuitive meaning that could be interpreted in an interesting way. If I find my results are unsatisfactory, I can easily gather much more data by not only including the most recent releases as I work through the project, but also older press releases. However, I have initially limited it to the most recent data in hopes of capturing "the mood of the nation" in terms of what themes are dominating in the run-up to the election.

## Evaluation Metrics

The most objective performance metric will be whether my model can correctly identify the party that authored a press statement – beating a random guess by say 10% would in my view be a reasonable outcome given the degree of overlap that is likely to be present and the number of parties. Results are likely to be more robust if obtained through k-fold cross validation (though the commonly used k=10 may be spreading my data too thinly). I will conveniently have a constant stream of potential test data arriving on a daily basis!

As noted, the sample is unbalanced, in that the National party (currently in Government) puts out many more press releases than the other parties. Therefore simply using an accuracy metric is not appropriate. A fuller analysis of a confusion matrix including false negatives and false positives, and consideration of metrics such as precision, recall, and the combined F1 measure, or possibly Cohen's kappa or ROC curves is warranted in this situation, as described in the article 8 tactics to combat unbalanced classes in your machine learning dataset. There is no a priori reason to weight one kind of error more heavily than another in this case.

## Project Design

The first stage, to gather the text data, is well underway. I have already scraped the press releases from the websites using Beautiful Soup. Next is preprocessing, eg extracting the author and date as well as the main text, as well as the standard preprocessing steps for text analysis such as stripping punctuation, eliminating stop words and stemming.

I will then undertake Multinomial Naïve Bayes text analysis to see if a) I can predict the author of a test press release and b) whether the useful features are interesting in their own right. As described above I then may investigate one more kind of text analysis model such as a support vector machine or convolutional neural network. I think it is also import to do at least some basic research into the best way of visualising my results, as that is a key component of this kind of analysis, eg word clouds from http://www.wordle.net/ using the frequency tables that drop out of my analysis.