For feature engineering, we tried to select our features from three different perspectives which are Stylometry and also Bag of Words. We used specific methods such as TF-IDF and Doc2Vec to allow the machine to extract features itself for the method Bag of Words(BOW), while we manually selected features for Stylometry. Since there are two types of feature selection for our project, we decided to use two different preprocessing methods to filter the dataset respectively. First, for the BOW, how we preprocessed the dataset is to turning the sentences into a list of words then eliminate unnecessary punctuations such as apostrophe, slash, etc. However, when we studied the dataset, we found that some people particularly like using multiple periods and exclamation marks, etc., so we kept those important punctuations but just kept one of them if they are repetitions. We also turned all the words into lowercase except those who only consist of capital words because we found some people always use this kind of words which can be a great feature when identifying these authors. Besides these steps, we also eliminated all the stop words that are common in English and removed some unnecessary but common words such as “http”. Finally, we implemented lemmatization to remove interferences further.

Doc2Vec is a method based on Word2Vec but the difference is that this method turning the paragraphs or documents into numeric representation as the features. For this model, we tried with Distributed Bag of Words algorithm which the document vectors are gained from the result of trained neural network. The task of the neural network is to predict a probability distribution of words in a document with a given random word from the document. Hence the features are the inferred vectors from the DBOW model.

TF-IDF is a method in information retrieval that reflects how important a word is in a given document. Before applying TF-IDF to generate vectors, we also used N-gram to create more meaningful features based on the sequence of words. The first step we took when implementing TF-IDF method is to include every feature that TF-DF generated. However, we quickly found that this is the wrong way because there are more than 400millions of features, which is unrealistic to compute. Then we started from very small amount of features by filtering the document frequency, and lowered the filter parameter gradually. In this way, we could have different number of feature per iteration and compare which feature set are more important to the model. Similarly, we also tried to filter the features by specifying the max number of features which ranked by the normalized term frequency.

This multi-class text classification problem is similar to supervised machine learning problem in which we have an author as the label for each tweet. We first compared the three categories of features with a baseline classifier logistic regression. We found that TF-IDF performed best, and the dataset as well as the number of classes is huge, so we decided to compare four conventional learners with default hyperparameters: Logistic Regression, SVM, Naïve Bayes, and Decision Trees.

After comparing the results of the classifiers with default hyperparameters above, we decided to focus on Logistic Regression and SVM because they require less computational resources while having higher accuracy. With default hyperparameters, we observed that there is overfitting for logistic regression. When we added penalties to the model, there is a significant decrease in accuracy from 40% to 16%. On the other hand, we also tested with SVM with linear, polynomial, and kernel methods. The results showed that linear SVM has the highest accuracy.

The reason why Doc2Vec performed bad might because the feature vectors are affected by the small single document size. For example, each tweet has 140 words limitation. For such small corpus, the vectors learned from Doc2Vec are small and the same words can be learned for multiple times with different meanings. What’s more, rare words are treated as noise in this model, so the preprocessing procedure should be redesigned in the future when using Doc2Vec.

From the results of models, SVM with linear kernel performs better than Logistic Regression. The reason can be that the logistic regression tries to find the best solution while the SVM tries to give optimal solution. To be more specific, with the large margins, the SVM has lower variance than logistic regression which means outliers does not affect the results much. The logistic loss function diverges faster, so it is more sensitive for outliers. From the results of logistic regression, we can find that the dataset is non-linear separable, which means that more outliners are there for the models. Thus, SVM did better than Logistic Regression in this project.

As for the TF-IDF, the reason why the features that extracted from TF-IDF give us the best results can be concluded in three aspects.

First, with TF-IDF, the uniqueness of feature are guaranteed. When identifying authors for tweets, the goal is to find the uniqueness of the author for a given tweet. However, one big challenge is that the content is not fixed, and, with fixed length limitation, it is hard to identify a pattern of structures of a tweet. By using TF-IDF, even the structures of tweets may not be so different from each other, we can get a large number of unique features as these features reflects how important the they are for the tweets.

Second, some people like to tweet about same topic. In this way, the term frequency will be large for this author. TF-IDF is invented to dealing with this requirement.

As for the third reason, the combination of stylometry and TF-IDF covers the feature diversity. For example, while TF-IDF handled the contextual part, the stylometry covered structural part. Although there may not many structural patterns as addressed above, using stylometry features such as the length of the tweet can still help. For example, some people like short tweets while others like longer tweets.