AUTHORSHIP PROFILING

Viet Tai Le
Pattranit Chaiyabud
Gabriella Martini

JUNE 14, 2020

Table of Contents

1 Introduction

This project is based on authorship analysis, specifically gender classification of author profiling of Twitter posts. Authorship analysis is a process that classifies texts into specific classes based on the characteristics of the author, whereas author profiling is a method that tries to discover the hidden elements of the author such as gender, age, language or personality. Our aim here is to find the gender of the authors of Twitter posts as Twitter is currently one of the biggest social media platforms that allows its users to communicate with each other through the use of text. There are about 330 million monthly active users of Twitter, and as many as 500 million tweets (short texts with a maximum of 280 characters) are sent out every single day (Lin, 2019).

The dataset (train_labels.csv) given to us consists of Twitter posts with training ids and genders of 3,100 authors. Using this dataset, we want to develop best model among machine learning algorithm namely Logistic Regression, Random Forest and Neural Networks. After that we tested out the best model performance using the test dataset (test.csv) with twitter posts from 500 authors to predict the authors' genders.

2 Pre-processing and feature generation

Before creating the models to predict the gender, we first have to pre-process the data since the raw data could be filled with irrelevant information and have various inconsistencies, however, we only want to extract important data to enhance the prediction. After pre-processing the data, we generate features to train our models. This step is crucial as it transforms the data into numerical value, to be precise, vector so that the algorithm is able to understand and learn, in order to accurately classify the twitter posts.

2.1 Pre-processing

To begin with, from raw data to a perfect feature, the following pre-process-steps are required:

1. **Removing Twitter context labels like Emoji, human labelling and tweet signs**:
   Raw tweet data contain many icons and signs which express the feeling of a writer, for example, smile, cry, and laugh and hashtag, mention sign etc. However, they are unnecessary to include as a key feature, hence they should be removed. The following shows what are removed:
   Emoji (Baqapuri, n.d.) - remove the possible emoji that tweet users usually apply could be categorized into 2 groups namely positive and negative.

   Positive: happy/excited/joyful emoticons or the emoticons with positive vibes

   ':-)', ':)', ';)', ':o)', ':]', ':3', ':c)', ':>', '=]', '8)', '=)', ':}',

   ':^)', ':-D', ':D', '8-D', '8D', 'x-D', 'xD', 'X-D', 'XD', '=-D', '=D',

   '=-3', '=3', ':-))', "':-)", "':)", ':*', ':^*', '>:P', ':-P', ':P', 'X-P',

   'x-p', 'xp', 'XP', ':-p', ':p', '=p', ':-b', ':b', '>:)', '>;)', '>:-)','<3'

   Negative: sad/disappointed/cry or the emoticons with negative vibes

   ':L', ':-/', '>:/', ':S', '>:[', ':@', ':-(', ':[', ':-||', '=L', ':<',

   ':-[', ':-<', '=\\', '=/', '>:(', ':(', '>.<', "':-(", "':(", ':\\', ':-c',

   ':c', ':{', '>:\\', ';('

Mention sign: remove the mention sign '@' followed by name appears as when tweet users mention some other account.

Hashtag: remove hashtag '#' sign as tweet users use to help them relate to relevant keywords or phrases in the Tweet.

Retweet sign: when a tweet message is shared, usually retweet starts with 'RT'. Hence, remove the retweet sign using regular expression

Hyperlink: remove all hyperlink 'https'' as they couldn't tell anything

The special string of html: remove special string related to html as they are not understandable.

2. **Case normalization**: the process of normalizing text into upper or lowercase. Here, we normalized all the letters into lowercase letters.

3. **Tokenization:** the process of splitting sentences into individual words. Here, we used a regular expression tokenizer (RegexpTokenizer) to tokenize the words.
4. **Stop words**: common words that carry little lexical content. We used the nltk stop words in English to remove all the stop words in the posts.
5. **Removing punctuation**: we removed punctuations like the period, question mark, exclamation point, comma, semicolon, colon, dash, hyphen, parentheses, brackets, braces, apostrophe, quotation marks, and ellipsis.
6. **Removing the most/least frequent word**: we removed words appearing in more than 95% of the document and less than 1% of the document as there is a high probability that the words are not important and could lead to the inaccurate result.

2.2 Feature generation

After having clean data from pre-processing, we extracted two feature sets to perform in this assignment.

- First feature set: we used Porter Stemmer function to stem the words after pre-processing data, basically transforming each word into its root, and then used TF-IDF vectorizer to make the feature by extracting in the form of unigram and bigrams of the token from the previous section.
- Second feature set: we used TF-IDF to vectorize the words in unigram and bigram just like in the first feature, however multiple the Glove Vector (Word2Vec) word embedding to each word and sum by total TF-IDF (Singh, 2019).

Therefore, the major difference between the two sets is that one uses stemmer to obtain the root of a word in the document, whereas the other one includes the word embedding to capture the context of a word in the document (Karani, 2018).

Clarification of important terms

N-grams are a set of co-occurring words where N signifies the number. So for unigrams, it will be separated word for word, whereas for bigrams it will be separated for every two word.

Porter Stemmer is a stemming algorithm that removes affixes from a word such as (-ed,-ize, -s,-de, mis-, -ing) to obtain the root part of a word. For example, the stemmer converts the word "running" to "run".

TF-IDF which is short for Term Frequency-Inverse Document Frequency measures how important a word/n-gram is to a document in a collection. It is a statistic that represents the relevance of a word, and not the frequency. This feature is very useful as it does not give much weightage to common words like "the, of,

and" that appear in all documents, and hence give other distinctive and unique words more weightage(Nicholson, n.d.).

Word Embedding is a set of feature learning techniques that learns the context of a word in a document. For example, "have a good day" and "have a nice day", the "good" and "nice" here have a very similar meaning, so word embedding is able to classify them together (Karani, 2018). This is similar yet different to stemming, as stemming is only able to find the root word, for example from "great" to "good". Word2Vec is one of the word embedding methods that can be used to do this. It is a form of neural network that vectorizes words into numbers, for the algorithm to understand the words (Nicholson, n.d.).

The main advantage of word embedding is that it allows for more expressive words. Like the example mentioned above, it was able to maintain the contextual similarity of the words like "good" and "nice". However, a disadvantage of word embedding is that it cannot understand out-of-vocabulary words.

3. Models

In this step, we introduced four different models that we are going to use to generate the prediction. These include Random Forest, Neural Network, Logistic Regression and Soft Voting Classifier. As our project requires us to predict the gender related to the given Twitter post, we chose models that were able classify and predict categorical values. We will discuss each model in more detail in this section.

3.1 Random Forest

Random forest is the model that is made up of a great number of decision trees which operate as an ensemble (Yiu, 2019). The training data sets in random forest have been randomly sampled with replacement from the original train data set, in other words, bagging, leading to varied structures of the decision tree. Furthermore, training features in random forest have been randomly selected from the same training data set of each tree. Random forest yields a class prediction and the class with the major votes is the model's prediction. Assumption: A large number of decision trees in the random forest is relatively uncorrelated or has a low correlation as it produces ensemble predictions that are more accurate than individual predictions.

In this assignment, we apply sklearn.ensemble.RandomForestClassifier to construct the random forest model with the following important parameters: 'n_estimate' which is the total number of decision trees in random forest, 'max_feature' which is the number of features to consider when looking for the best split, and 'criterion' which decide the split using default setting gini.

The main advantage of random forest is it reduces overfitting problems in decision trees and reduces variance resulting in accuracy improvement(The Professionals Point: Advantages and Disadvantages of Random Forest Algorithm in Machine Learning, n.d.). On the other hand, the disadvantage is that random forest requires a long time for training as it builds several trees and combines the output.

3.2 Neural Networks

Neural networks are a set of algorithms that are designed after the structure of the human brain (A Beginner's Guide to Neural Networks and Deep Learning, n.d.) in order to make predictions as any other model. The process neural networks work is it takes in data, trains themselves to recognise the pattern in the data and predict the output for the new set of similar data. In order to follow the process mentioned above, at least 3-layer networks of neurons (core processing units) are built consisting of an input layer for receiving data, hidden layers for performing most of the computation of networks and an output layer for predicting. It seems like a simple structure, yet powerful enough to handle most of the complex problems such as image processing, character recognition and forecasting (Yiu, 2019).

In this assignment, we apply keras.wrappers.scikit_learn.KerasClassifier to construct 3-layers neural networks with the following important parameters: 'n' which is number of neurons/nodes in each layer and 'a' which is an activation function, in this case, we use sigmoid activation function. Activation function is a function that determines whether the particular neurons get activated and if it is activated then it passes through the networks for prediction (Keras for Beginners, n.d.).

The main advantage of neural networks is that they can learn and model both linear, nonlinear and complex relationships yet generalize models (Mahanta, 2017) and predict unseen data with high accuracy. However, the main disadvantage of neural networks is that they are related to a large number of parameters (Cross Validation - Disadvantages of Neural Network Method, n.d.) which is difficult to find a good setting, also hyperparameters can strongly interact with each other (Cross Validation - Disadvantages of Neural Network Method, n.d.).

3.3 Logistic Regression

Logistic regression is the model that is usually applied in classification problems as its target variable is categorical variables (Agrawal, 2017). It is an extension of linear regression that can return possessing probability and based on decision boundary, the probability can be classified into class. Moreover, we can apply regularization to avoid overfitting. Assumption: Firstly, there is enough training sample size. Secondly, there is absence of multicollinearity (intercorrelation among the features) and finally, no outliers (Statistics Solutions, n.d.).

In this assignment, we apply sklearn.linear_model.LogisticRegression to construct the logistic regression model with the following important parameters: 'C' which is the inverse of regularization strength, a penalty measure that increases with increasing number of features(What Is the Inverse of Regularization Strength in Logistic Regression? How Should It Affect My Code?, n.d.), 'max_iter' which is the maximum number of iterations needed for it to converge, and 'penalty' which specifies the method used for penalization.

The advantage of using logistic regression is that it is simple, efficient, and has low variance. However, a disadvantage of logistic regression is that the model needs to be transformed into non-linear (Agrawal, 2017), usually done using the sigmoid function.

3.4 Soft Voting Classifier

Voting classifier is an ensemble machine learning method that combines the prediction of different models improving a chance of better performance than having a single model. Two major types of voting classifier are 'soft voting' and 'hard voting'. Hard voting involves predicting the class with the most votes, but soft voting takes into account the probability of each class predicting the class label with the largest probability (Brownlee, 2020). Therefore, soft voting considers each classifier's uncertainty in the final decision leading to a lower variance of accuracy and high stability of the model (Brownlee, 2020). Assumption: all models in the ensemble have generally the same good performance (Brownlee, 2020)

In this assignment, we apply soft voting classifier using sklearn.ensemble.VotingClassifier to construct soft voting classifier model with the following important parameters: 'estimators' which specify all ensemble models, 'voting' which identify soft or hard method, and 'weights' which is the sequence of weights of class probabilities.

The advantage of using the voting classifier is that it offers lower variance in the predictions over single models leading to a lower mean performance of the ensemble meaning higher stability or confidence of the model (Mangale, 2019). On the other hand, the disadvantage of voting classifiers or the limitation is that it treats all the models equally which might be good only for some situations and poor for others (Brownlee, 2020).

3.5 Discussion of model differences

- Random forest and neural networks support non-linear solutions, whereas logistic regression does not.
- Random forest deals with collinearity better than logistic regression as logistic regression's assumption states that there must be non-collinearity among variables.

- Logistic regression is the most interpretable one out of all models as random forest, neural networks and soft voting are more difficult to visualize.
- Logistic regression usually outperforms neural networks as they require a large amount of training data (Varghese, 2019). Therefore, when training data is small and there is a large number of features, logistic regression is more compatible.
- Voting classifier or ensemble methods help improve machine learning performance by combining several models allowing better prediction(What Are the Advantages of the Ensemble Method? - Quora, n.d.) compared to single model so this is probably the main difference between voting classifier and others.

3.6 Discussion of parameter selection and algorithm selection

- Hyperparameters are the parameter we specify before fitting a model, while model parameters are parameters arising from the model fitting result. An example to show this difference is, in a logistic regression model, the regularization strength is a hyper parameter which should be set out before fitting, model parameters, on the other hand, is the coefficients of the fitted model.
- We want to select the best learning algorithm (including its optimal hyperparameter) from a group of models. In the following section, we will refer to this as algorithm selection.
- Given the classification problem we are facing, we wonder which algorithm is the most suitable, yielding the best performance, hence we will show how to perform algorithm selection and hyper-parameter selection among models mentioned above.

4 Experiment setups

In this section, we try to find the best algorithm and then build a model by fine tuning the hyper-parameter and estimate test accuracy. Cross-validation can either do one of them but not both at the same time. Here, we want to use nested cross-validation. In nested cross-validation, we have an outer k-fold cross-validation loop to split the data into training and test folds, and an inner loop is used to select the model via k-fold cross-validation on the training fold. After model selection, the test fold is then used to evaluate the model performance. After we have identified our best performing algorithm, we can use regular k-fold cross-validation approach (on the complete training set) to find its “optimal” hyperparameters. As nested cross validation is quite computationally expensive, in order to save running time, we will set an inner loop with k=5 in k fold cross validation and an outer loop with k=3 in k fold cross validation.

The experiments are carried out using the following two feature sets: first, feature vector represented by TF-IDF, and second, feature vector represented by TF-IDF weighted with Word2Vec.

Hyper-parameter sets

In this section, we set up some hyper-parameters that will be used in 5-cross validation in the inner loop. For models with a small search space (Logistic regression), the tuning method we used is Grid Search which prepares the sets of candidate hyperparameters. Whereas for models with a large search space (Random Forest, Neural network,Voting classifier), we will use Random Search to reduce the processing time.

Random forest

For the Random Forest Classifier, there are several different hyperparameters that can be adjusted. In this assignment, we will take into consideration the following four parameters:

- n_estimators: number of trees in the forest. Here we used fine tuning from 100 to 1000 with step of 100

- max_features: maximum number of features considered for splitting a node. We choose fine tuning between 'sqrt' and 'log2'
- max_depth : maximum number of levels in each decision tree. (Fine tuning from 10 to 100 with step of 10)
- min_samples_split : minimum number of data points placed in a node before the node is split. (Fine tuning between 2,5,10)

Neural network

- Number of epochs: the number of epochs is the number of times that the entire training dataset is shown to the network during training.(Fine tuning between [15,30,50])
- Batch Size: The batch size in iterative gradient descent is the number of patterns shown to the network before the weights are updated. [10,20,40,80]
- Activation function: The activation function controls the non-linearity of individual neurons and when to fire. (Fine tuning between [relu,tanh] in the hidden layer)
- Number of hidden neurons: The number of neurons in a layer is an important parameter to tune. (Fine tuning between [50,100,150,200] in the hidden layer)

Logistic Regression

- Penalty: This hyper-parameter is used to specify the type of normalization used. (Fine tuning between [l1,l2])
- Inverse of regularization: This hyper-parameter is denoted as C. Smaller values of this hyper-parameter indicate a stronger regularization. (Fine tuning between [0.5,1,1.5,2,2.5,3,3.5,4,4.5,5])
- Max iter: Represents maximum number of iterations taken for the solvers to converge a training process. Fine tuning between [50,100,150])

Voting Classifier

In this report, we built an ensemble algorithm by soft voting the base classifiers. The parameters of voting class is a combination of parameters set of base models. Also, we need to fine tune weights of the base classifier.

- Weight: Sequence of weights to weight class probabilities before averaging voting (Fine tuning between [1,2,3],[[1,3,2],[1,1,1],[3,2,1])

Evaluation Metric:

In this assignment, we will mainly be using the accuracy measure to evaluate the performance of the models. However, in addition to the accuracy rate, we can expand more about True Positive rate (TPR) and True Negative rate (TNR), to evaluate the performance on the test set. Basically, expecting the models to perform well in all these metrics, the accuracy, TPR and TNR. In this report, we would treat the male label as positive and the female label as negative.

Accuracy = (True Positive+TrueNegative)/(TruePositive+False Positive+False Negative+True Negative)

True Positive Rate (TPR) (Sensitivity) = TruePositive/(TruePositive+FalseNegative)

True Negative Rate (TNR) (Specificity) = TrueNegative/(TrueNegative+FalsePositive)

5 Experimental results

The table below shows the comparison of the mean and standard deviation of nested cross validation accuracy of all models that are run on different sets of feature vectors.

Feature	Random Forest	Neural network	Logistic regression	Voting Classifier
TFIDF	77.3	78.4	79.0	79.2
TFIDF+ Word2vec	77.0	78.6	78.7	79.4

The table compares the performance of our four classification algorithms. We used a mean of 3-fold outer loop accuracy. The result in logistic regression and random forest are almost the same. The reason is in those algorithm, we set up parameters for doing feature selection (max_features and penalty). Although dimension reduction and feature selection are different concepts, they both reduce the number of features in the dataset. As base algorithms perform similarly between different feature sets, this also leads to the result for the voting classifier almost the same in 2 feature vectors. However, by doing dimension reduction, building the model is much faster. Hence, in developing the last model, we would use TF-IDF + Word2vec feature set.

In addition, when comparing between algorithms, the performance of neural network and logistic regression is almost the same using this dataset. The model which ensembles our three base classifiers perform slightly better than stand-alone classifier (79.4%). Hence, we will choose our final model to be the Voting classifier which is trained on TF-IDF + Word2Vec feature vector

Now, we can fine tune parameters on the Voting Classifier to develop the final model. Although the GridSearch method guarantees that the search will produce the perfect solution, it is very computationally expensive. On the other hand, RandomSearch does not guarantee that we will find the best suit for the build model. Hence, our approach here is to try to find the best hyper-parameter of each base algorithm and then using those parameters to build the final model. Once again, we use the GridSearchCV tuning method for logistic regression and RandomSearchCv for RandomForest and Neural Network. Additionally, we will use GridSearch to fine tune the weights between those algorithms.

Evaluate test accuracy

Once we have developed the final model, we want to evaluate model performance on the test set. As we are assuming that the test data has the same distribution as the train data, we expect the test accuracy and estimated test accuracy (nested cross validation accuracy) to be almost the same. Therefore, we can conclude our final model is good and can be said that it has generalized to a fair level.

	Accuracy	TPR	TNR
Voting classifier	0.82 ± 0.002	0.806	0.837

The table above shows the performance of our final model on the test set of 500 (252 female+248 male) tweets. From the observation, it can be seen that the test accuracy and validation accuracy are not much different. Hence, it's true to say our model is not overfitting. There are two terms, TPR and TNR, which are also used to evaluate the performance of the final model on each class. The TPR is the accuracy percentage of our

final model when it predicts the male class, which is 80.6%. The TNR is the accuracy percentage of our final model when it predicts the female class, which is 83.7%. As shown, the TPR and TNR is quite high, and hence our final model is doing a good job at the prediction of the 2 classes.

6. Conclusion

In this project, we have discussed four algorithms, Random Forest, Neural Network, Logistic Regression and Soft Voting Classifier, to predict the gender of the authors. After running nested cross validation, we chose to use the Voting Classifier to be the best algorithm to solve this task. To ensure that the Voting Classifier gives us the best performance, we fine tuned its parameters. Finally, we tested our accuracy on the test dataset and obtained an accuracy score of 82%, with a fair distribution of TPR and TNR. This means that the model is not biased to predicting one gender class, and hence is a good model that can be used to predict the genders of the Twitter posts.

7. References

A Beginner's Guide to Neural Networks and Deep Learning. (n.d.). Pathmind. Retrieved 14 June 2020, from http://pathmind.com/wiki/neural-network

Agrawal, A. (2017, March 31). *Logistic Regression. Simplified.* Medium. https://medium.com/data-science-group-iitr/logistic-regression-simplified-9b4efe801389

Baqapuri, A. I. (n.d.). *Twitter Sentiment Analysis*. 53.

Brownlee, J. (2020, April 16). How to Develop Voting Ensembles With Python. *Machine Learning Mastery*. https://machinelearningmastery.com/voting-ensembles-with-python/

cross validation—Disadvantages of Neural network method. (n.d.). Cross Validated. Retrieved 14 June 2020, from https://stats.stackexchange.com/questions/116245/disadvantages-of-neural-network-method

Karani, D. (2018, September 2). *Introduction to Word Embedding and Word2Vec*. Medium. https://towardsdatascience.com/introduction-to-word-embedding-and-word2vec-652d0c2060fa

Keras for Beginners: Building Your First Neural Network - victorzhou.com. (n.d.). Retrieved 14 June 2020, from https://victorzhou.com/blog/keras-neural-network-tutorial/

Lin, Y. (2019, November 30). *10 Twitter Statistics Every Marketer Should Know in 2020 [Infographic]*. Oberlo. https://www.oberlo.com/blog/twitter-statistics

Mahanta, J. (2017, July 12). *Introduction to Neural Networks, Advantages and Applications*. Medium. https://towardsdatascience.com/introduction-to-neural-networks-advantages-and-applications-96851bd1a207

Mangale, S. (2019, May 18). *Voting Classifier*. Medium. https://medium.com/@sanchitamangale12/voting-classifier-1be10db6d7a5

Nicholson, C. (n.d.). *A Beginner's Guide to Bag of Words & TF-IDF*. Pathmind. Retrieved 14 June 2020, from http://pathmind.com/wiki/bagofwords-tf-idf

Singh, R. (2019, September 12). *Featurization of Text data (BOW, TF-IDF, AvgW2V, tfidf weighted W2V)*. Medium. https://medium.com/@ranasinghiitkgp/featurization-of-text-data-bow-tf-idf-avgw2v-tfidf-weighted-w2v-7a6c62e8b097

The Professionals Point: Advantages and Disadvantages of Random Forest Algorithm in Machine Learning. (n.d.). Retrieved 14 June 2020, from http://theprofessionalspoint.blogspot.com/2019/02/advantages-and-disadvantages-of-random.html

Varghese, D. (2019, May 10). *Comparative study on Classic Machine learning Algorithms*. Medium. https://towardsdatascience.com/comparative-study-on-classic-machine-learning-algorithms-24f9ff6ab222

What is the inverse of regularization strength in Logistic Regression? How should it affect my code? (n.d.). Retrieved 14 June 2020, from https://www.likeanswer.com/question/31529

Yiu, T. (2019, August 4). *Understanding Neural Networks*. Medium. https://towardsdatascience.com/understanding-neural-networks-19020b758230

Yiu, T. (2019, August 14). *Understanding Random Forest*. Medium. https://towardsdatascience.com/understanding-random-forest-58381e0602d2

What are the advantages of the ensemble method? - Quora. (n.d.). Retrieved 14 June 2020, from https://www.quora.com/What-are-the-advantages-of-the-ensemble-method