Twitter Author Profiling

Neeraj Lad*1, Ritesh Tawde*2

Abstract
Authorship profiling deals with study of analyzing different classes of authors by learning how they use their language, words, etc. Authorship profiling helps in identifying demographics of unknown author such as age, gender, native language, etc. It is of growing importance in forensics, security. Also, it help in market analysis by knowing how people react to different products, campaigns, reviews and to understand which group of people like the product and which group do not.

Keywords
author profiling — social media mining — machine learning

* *Computer Science, School of Informatics and Computing, Indiana University, Bloomington, IN, USA*
1**Corresponding author**: nlad@iu.edu
2**Corresponding author**: rtawde@iu.edu

Contents

1 Introduction 1
2 Related Word 1
2.1 Author Profiling . . . 1
2.2 Current work . . . 1
3 Data 1
3.1 Description . . . 2
3.2 Data Collection . . . 2
3.3 Data Summary Statistics . . . 2
4 Algorithm and Methodology 2
4.1 Data Preprocessing . . . 2
4.2 Algorithms for classification . . . 2
4.3 Libraries . . . 3
5 Results and Evaluation 3
5.1 Results . . . 3
Gender classification • Age classification (imbalanced class) • Age classification (balancing class)
5.2 Evaluation . . . 3
6 Discussion and Conclusion 4
References 4

1. Introduction

Authorship profiling deals with analyzing a collection of texts written by an unknown author and classifying that author's attributes such as gender and age. Authorship profiling is a very useful task in today's age with a large amount of textual data being generated on different platforms such as e-commerce websites, blogs, social networks, etc. It would be profitable for online sellers to infer the attributes of their users based on the reviews. Thus knowing the profile of customers, the seller can modify the products to target the majority demographic and come up with targeted advertisements to attract more people from the same group. The same use-case can be extrapolated to social media such as Twitter. This can enable diverse movements active on such social networks to know more deeply about their followers and critics, and thus evolve their manifestos based on which demographics they wish to target. We take a look at a few Machine Learning algorithms such as Linear SVC, Multinomial Naive Bayes among others to complete this task.

2. Related Word

2.1 Author Profiling

Author Profiling is the task of identifying details about the author just by analyzing the text at hand. Here the idea is to identify characteristics of unknown author by their writing style, use of English language, Mother Tongue Influence (MTI), gender, etc[1]. The particular author profile dimensions which we considered in this paper are gender, and age-group.

2.2 Current work

Argamon et al.[1] discuss the author profiling task pretty extensively. They have mentioned the techniques of classification using stylistic and content based features of text to get the most information out of it and build the model. They also paper discusses the outcome of classification task by providing distribution of words' usage by different gender people with different age groups.

We use similar approach by identifying the distribution of word's usage by different kind of people and use this as our training parameter along with few other parameters.

3. Data

3.1 Description

PAN-16[2] is a labeled dataset. It consists of a subset of tweets taken from *Twitter* [3] Twitter is an online social network where users publish posts called *'tweets'*, which are limited to 280 characters (previously 140 characters). The dataset consist of tweets in three languages: English, Spanish, Dutch. This paper only uses data from the English corpus. It consists of 436 xml files. Each xml file represents one twitter user and is named as *'twitter_user_id.xml'*. Each file has a collection of Tweet IDs and the corresponding URLs of tweets created by that specific user. There are 263,031 tweets in the entire dataset. There is a file *'truth.txt'* which maps a *twitter_user_id* to their Gender and Age group. The categories for Gender are Male and Female. The categories for Age group are 18-24, 25-34, 35-49, 50-64, 65-xx.

3.2 Data Collection

We used *'twitter-text-python'*[4] which is a tweet parser written in Python. For every *twitter_user_id* in *'truth.txt'*, it opens the corresponding file in the English corpus directory. It uses *'ElementTree'* in the Python module *'xml'* to parse the xml file. For every document in the file, it extracts the URL and uses Selenium[5] to extract the tweet text.

3.3 Data Summary Statistics

There are 436 users in the English corpora, with 263,031 tweets overall. The 'Gender' categories (Male and Female) have equal number of users at 218 each, but the number of tweets are different for each category. The 'Age' category consists of age groups '18-24', '25-34', '35-49', '50-64' and '65-above' with categories '35-49' and '25-34' having the highest number of users at 140 and 182 respectively.

Table 1. Dataset distribution per class labels

Total users	436
Total Tweets	263,031
Male	149059
Female	113972
18-24	18126
25-34	92059
35-49	105520
50-64	44896
65-above	2430

4. Algorithm and Methodology

4.1 Data Preprocessing

- Since the dataset only contains the *tweet IDs* and *tweet URLs*, we scraped the tweets from Twitter using Selenium Chrome Driver. All tweets are stored in a single corpus using JSON format. As the dataset is in raw format, tweets are pre-processed using the following techniques[6] :

  1. **Ignore case** : Converting all tweets to lowercase.
  2. **Ignore punctuation** : Removing all punctuation marks to only consider alpha-numeric data.
  3. **Remove stop/frequent words** : Removing the common English language stopwords using nltk's [7] stop words corpus.
  4. **Word stemming** : Applying word stemming techniques to get the root word using the stemming[8] function in the python library.

- We considered few more parameters as potential feature vectors. Those are whether there was any external link present in the tweet and the number of mentions in a single tweet.

- Target labels are encoded using target labeling. Target label for gender are encoded as follows :

Table 2. Target factoring for gender

Gender	Target label
Male	1
Female	0

Age groups are categorized into numerical factors as follows :

Table 3. Target factoring for age groups

Age group	Target Label
18-24	0
25-34	1
35-49	2
50-64	3
65-above	4

- The tweet text is converted to document term matrix to work well with machine learning algorithms using vectorizers

- We split the entire dataset into 90% training and 10% testing dataset.

4.2 Algorithms for classification

Since this is a binary classification task (for gender) and multi class classification (for age groups), we tried modeling using following algorithms to come up with the algorithm which provides best model for each of the classification task:

1. **DummyClassifier** : This classifier is just a baseline classifier to compare results from the other algorithms.
2. **MultinomialNB** : Multinomial Naive Bayes is based on the priors and maximum likelihood of any class to predict the posterior probability of class given the data.
3. **Linear SVC** : Linear Support Vector classification is a fast and robust algorithm for classification.

4. **SGDClassifier** : Stochastic Gradient Descent is an algorithm which tries to minimize the cost function while estimating the best model and reducing the squared loss error between the actual and the predicted value while fitting the parameters. We tried SGD with few different parameters using two different loss functions, viz., 'L2' and 'elasticnet'.

5. **GradientBoosting** : GradientBoostingClassifier makes use of a tree based model.

Also, since the age group class is highly imbalanced, we tried *RandomOverSampling* technique to balance the classes prior to training phase using all the above algorithms. The *Results* section provides detailed outcomes for each of the prediction task using each of the technique.

4.3 Libraries

We used Python's *scikit*[9] library, which provides an extensive and robust collection of algorithms for Machine Learning. Also, since we worked extensively with textual data, we used *nltk* to complete most of the pre-processing task efficiently. Lastly for plotting the graphs, python's *matplotlib*[10] library is used.

5. Results and Evaluation

5.1 Results

5.1.1 Gender classification

Table 4 summarizes the results for gender classification

Table 4. Gender classification

Classifier	Training time(*in sec*)	Accuracy
Dummy (Baseline)	0.019	0.50
Multinomial NB	0.271	0.7990
Linear SVC	47.937	0.8086
SGD (using l2)	4.129	0.7993
SGD (using elasticnet)	7.965	0.7993
GradientBoosting	65.164	0.6585

Figure 1 outputs the graph plot

5.1.2 Age classification (imbalanced class)

Table 5 summarizes the results for age classification

Table 5. Age classification (imbalanced class)

Classifier	Training time(*in sec*)	Accuracy
Dummy (Baseline)	0.026	0.2050
Multinomial NB	0.299	0.6914
Linear SVC	130.597	0.7124
SGD (using l2)	20.540	0.6946
SGD (using elasticnet)	35.892	0.6983
GradientBoosting	351.131	0.5311

Figure 2 outputs the graph plot

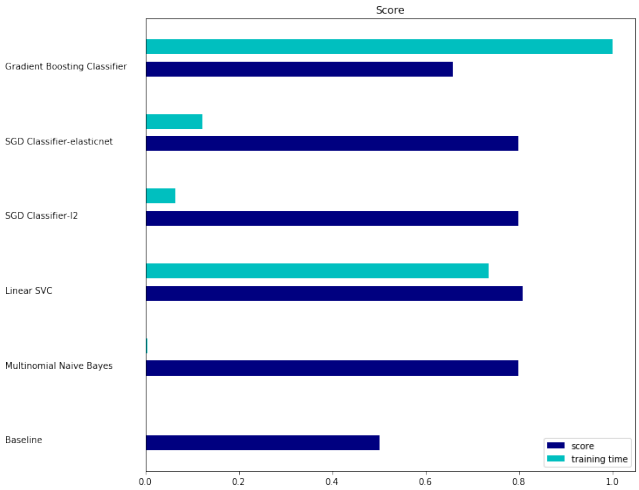

Figure 1. Gender classification

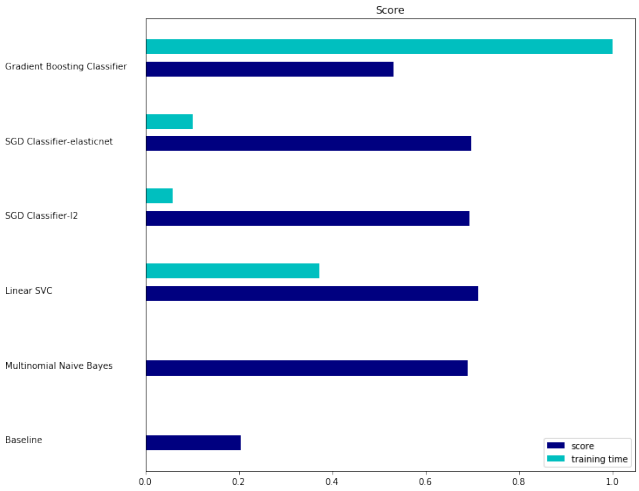

Figure 2. Age classification(imbalanced)

5.1.3 Age classification (balancing class)

Table 6 summarizes the results for age classification

Figure 3 outputs the graph plot

Based on the above observation, it can be seen that the Linear SVC algorithm gives the highest accuracy for all types of classification tasks and was chosen as the best model.

5.2 Evaluation

We requested access to the PAN-16 testing dataset by registering to access a Virtual Machine with the organizers, but we haven't heard back from the organizers as of submission of this paper. Since, we do not have an actual test set for PAN-16 author profiling competition, we could not directly compare our results with the other competitors.

But, we did run few of the algorithms along with the baseline and we can see that our model performs much better than the baseline version and the task of classification really well giving high accuracy. So we believe that we built a pretty good and comparable model for Author Profiling.

Table 6. Age classification (balanced class)

Classifier	Training time(*in sec*)	Accuracy
Dummy (Baseline)	0.033	0.2050
Multinomial NB	0.295	0.6490
Linear SVC	495.765	0.6788
SGD (using l2)	42.850	0.6689
SGD (using elasticnet)	76.001	0.6644
GradientBoosting	806.328	0.4241

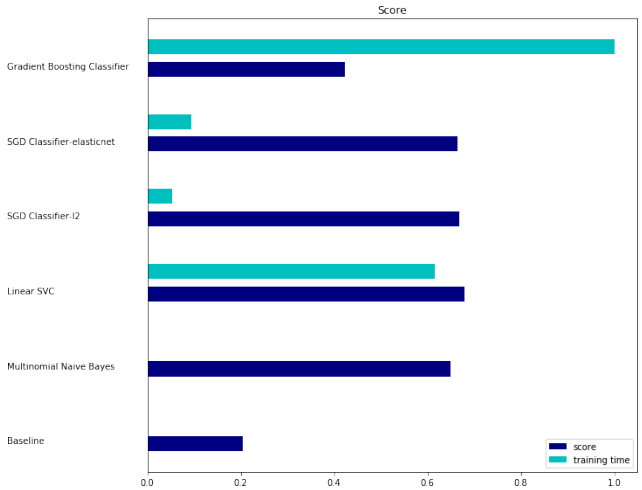

Figure 3. Age classification(balanced)

6. Discussion and Conclusion

From the above methods and results, it can be seen that the Linear SVC algorithm performed well for both kind of classification tasks giving the highest accuracy on the test dataset among all the algorithms.

Since our task was to profile author based on Gender and Age, the data pre-processing techniques coupled with good mix of algorithms gave a good accuracy on this task and the results were significant when accuracy is considered. One of the big problems for such task is to work with huge text data which is sparse and scikit's sparse matrix was handy at that time.

For further work, these techniques can be improved using more advanced data pre-processing technique and Machine Learning techniques like cross-validation, stratified k-fold splitting of data, more robust and fast algorithms such as XGBoost[11].

References

[1] *Automatically profiling the author from an anonymous text*, S. Argamon, M. Koppel, J. W. Pennebaker, J. Schler.

[2] http://pan.webis.de/clef16/pan16-web/

[3] http://twitter.com/

[4] https://github.com/edburnett/twitter-text-python

[5] http://www.seleniumhq.org/download/

[6] J. Brownlee. A Gentle Introduction to the Bag-of-Words Model.

[7] http://www.nltk.org/

[8] https://pypi.python.org/pypi/stemming/1.0

[9] http://scikit-learn.org/stable/

[10] https://matplotlib.org/

[11] *XGBoost: A Scalable Tree Boosting System*, Tianqi Chen, Carlos Guestrin *XGBoost: A Scalable Tree Boosting System*, Tianqi Chen, Carlos Guestrin