Age and Gender Classification of Tweets using Support Vector Machines

Roy Khristopher Bayot
Teresa Gonçalves, PhD

Universidade de Évora, Department of Informatics,
Rua Romão Ramalho n°59, 7000-671 Évora, Portugal
d11668@alunos.uevora.pt

Abstract. Keywords: author profiling, twitter, tfidf, normalized term frequency, SVM, POS, ngrams

1 Introduction

Author profiling has been of importance in the recent years. From a forensic standpoint for example, it could be used to determine potential suspects by getting linguistic profiles and identifying characteristics. From a business intelligence perspective, companies could target specific people through online advertising. By knowing the profile of the authors, companies would easily find what a specific group of people talk about online and device strategies to advertise to these people. They could also analyze product reviews and know what types of products are liked or disliked by certain people.

Part of the reason why the interest in author profiling grows is because the growth of the internet where text is one of the main forms of communication. Through this growth, various corpora could be extracted, curated, assembled from different sources such as blogs, websites, customer reviews, and even twitter posts. Of course, this presents some problems. For example, people from different countries who use the same online platform such as Twitter or Blogger could behave differently in terms of text usage. This presents a difficulty in profiling. This work tries to take this difficulty into account by studying which kind of features are useful for different languages.

The aim of this work is to investigate age and gender classification of tweets using Support Vector Machines using the dataset given in PAN 2015 [13]. The dataset contains twitter data from 4 different languages - English, Dutch, Italian, and Spanish. These tweets are used to profile an author based on age, gender, and 5 personality traits - agreeability, conscientiousness, extrovertedness, openness, and stability. This paper limits the investigation to age and gender classification for English tweets.

This works starts off with an investigation of the different features that could be useful in classification in English. We investigate three different sets of features. The first set is the bag of words features in the form of term frequency and term frequency inverse document frequency. The second set are text ngrams.

And finally, we also study part of speech ngrams to see if information extracted from these features are useful for characterizing twitter users. After determining which features are useful in classification, this work then proceeds to evaluating different kernels as well as different parameters for classification.

2 Related Literature

One of the first few works on author profiling is that of Argamon et al. in [1] where texts are categorized base on gender, age, native language, and personality. For personality, only neuroticism was checked. The corpus comes from different sources. The age and gender have the same corpus taken from blog postings. The native language corpus was taken from International Corpus of Learner English. Personality was taken from essays of pyschology students from University of Texas in Austin. Two types of features were obtained: content-based features and style-based features and Bayesian Multinomial Regression was used as a classifier. Bayesian Multinomial Regression was used because it was shown to be effective for text classification problems. It's a variant of logistic regression that is used instead of naive bayes classifiers because it doesnt assume independence between features.

Argamon et al. had some interesting results where from the gender task, they were able to achieve 76.1% accuracy using style and content features. For age task with 3 classes, the accuracy was at 77.7% also using style and content features. For the native language task, the classifiers were able to achieve 82.3% using only content features. And finally, in checking for neuroticism, the highest obtained was 65.7% using only style features.

There has also been some research that uses datasets collected from social media. A particular example is that of Schler et al. in [15] where writing styles in blogs are related to age and gender. Stylistic and content features were extracted from 71,000 different blogs and a Multi-Class Real Winnow was used to learn the models to classify the blogs. The winnow algorithm is a supervised learning algorithm that learns a linear classifier. It is similar to a perceptron, that updates the weight in every training example. The difference lies in the fact that for perceptron, the weight update is additive while for Winnow, the update is multiplicative. Considering scale, Multi-Class Real Winnow becomes much more efficient than SVM. A possible drawback could happen when the decision bounderies are non-linear.

Stylistic features included parts-of-speech tags, function words, hyperlinks, and non-dictionary words. Content features included word unigrams with high information gain. The accuracy achieved was around 80% for gender classification and 75% for age identification.

The work or Argamon et al. [1] became the basis for the work in PAN. It is an ongoing project from CLEF with author profiling as one of its tasks. It currently has three editions. In the first edition of PAN [12] in 2013, the task was age and gender profiling for English and Spanish blogs. There were a variety of methods used. One set includes content-based features such as bag of words, named entities, dictionary words, slang words, contractions, sentiment words, and emotion words. Another would be stylistic features such as frequencies, punctuations, POS, HTML use, readability measures, and other various statistics. There are also features that are n-grams based, IR-based, and collocations-based. Named entities, sentiment words, emotion words, and slang, contractions and words with character flooding were also considered. After extracting the features, the classifiers that were used were the following - decision trees, Support Vector Machines, logistic regression, Naïve Bayes, Maximum Entropy, Stochastic Gradient Descent, and random forests. Most of the submissions for this edition used decision trees, wherein a classification tree is learned by splitting the data based on information gain or gini index of the features from which the split is based on. The idea is to keep on splitting until the instances in the leaves come from only one class. Three used Support Vector Machines wherein the features are mapped into another space and a hyperplane is fitted on the new space. The fitted hyperplane is made such that the gap between the different classes in the new space is as wide as possible. Two approaches used logistic regression, where it measures the relationship between the category and it's features by estimating the probabilities using a logistic function. This is then used to predict. One used Naïve Bayes where a probabilistic classifier is constructed using Bayes' theorem but with independence assumptions of features. Another used Maximum Entropy, which is the multivariate form of logistic regression. Another used Stochastic Gradient Descent, which is an iterative method for sum-minimizations such as that in least squares. And finally, one used Random Forest, which is an ensemble method described later in the text, where a multitude of trees is used instead of using one tree for classification.

The work of Lopez-Monroy in [6] was considered the winner for the task although they placed second for both English and Spanish with an accuracy of 38.13% and 41.58% respectively. They used second order representation based on relationships between documents and profiles. They used liblinear but did not specify which classifier was used.

The work of Meina et al. [9] used collocations and placed first for English with a total accuracy of 38.94%. They experimented on different classifiers but Random Forests gave the best result. Their final Random Forest classifier was trained on a 12-core machine with 30GB of RAM, and parameters were obtained through trial and error. The parameters include minimum samples per leaf to be equal to 5, the size of the feature set for each tree is equal to $\sqrt{n\_features}$, and the number of trees was lowered to 666 although it converges at values higher than 1000.

On the other hand, the work of Santosh et al. in [14] gave a total accuracy of 42.08% after using POS features for Spanish. They used three different kinds of features - content based, style based, and topic based. Each of the features has a classifier. Content based features in the form of ngrams and style based features in the form of ngrams of POS tags both used Support Vector Machines. Although the kernel and parameters used were not mentioned. Topic based features in the form of LDA topic model was used as features for a Maximum Entropy classifier. The results from each of the three classifiers are fed finally into a decision tree.

In PAN 2014 [11], the task was profiling authors with text from four different sources - social media, twitter, blogs, and hotel reviews. Most of the approaches used in this edition are similar to the previous year. In [5], the method used to represent terms in a space of profiles and then represent the documents in the space of profiles and subprofiles were built using expectation maximization clustering. This is the same method as in the previous year in [6]. In [7], ngrams were used with stopwords, punctuations, and emoticons retained, and then idf count was also used before placed into a classifier. Liblinear logistic regression returned with the best result. In [17], different features were used that were related to length (number of characters, words, sentences), information retrieval (cosine similarity, okapi BM25), and readability (Flesch-Kincaid readability, correctness, style). Another approach is to use term vector model representation as in [16]. For the work of Marquardt et al. in [8], they used a combination of content-based features (MRC, LIWC, sentiments) and stylistic features (readability, html tags, spelling and grammatical error, emoticons, total number of posts, number of capitalized letters number of capitalized words). Classifiers also varied for this edition. There was the use of logistic regression, multinomial Naïve Bayes, liblinear, random forests, Support Vector Machines, and decision tables. The method of Lopez-Monroy in [5] gave the best result with an average accuracy of 28.95% on all corpus-types and languages.

3 Dataset and Tools

The dataset for the problem at hand is composed of a set of tweets for 4 different languages - English, Dutch, Italian, and Spanish. It was decided to build one model for age classification and another model for gender classification. There were 4 categories for the age classification - 18-24, 25-34, 35-49, and 50 and above but for Dutch and Italian, age classification is not possible because there was no data given.

The number of given users varies for each language. There were 152 users for English, 34 for Dutch, 38 for Italian, and 100 for Spanish. Each user has different number of tweets. The dataset is balanced based on gender.

Looking at English tweets, the table 1 gives the characteristics of this particular section of the dataset.

	min	max	average	std	median
number of tweets of each user	32	100	93.20	16.82	100
total length of tweets of each user	1979	12485	7445.30	2389.61	7438.5
average tweet length of each user	29.46	124.85	79.61	20.28	80.35

Table 1. Various statistics on twitter data.

Looking at the number of tweets per user, we can see that the minimum is 32 tweets, the maximum is 100 tweets per user, with an average of 93.20, a standard deviation of 16.82, and a median of 100. Looking at the total length of all tweets for each user, the minimum is 1979 characters, the maximum is 12485 characters, an average of 7445.30 with a standard devation of 2389.61, and a median of 7438.5. Finally, looking into the average tweet length per user, the minimum is 29.46, the maximum is 124.85, average is 79.61, with a standard deviation of 20.28, and a median of 80.35.

Processing the data was done through Python using the scikits-learn [10] library. For POS tags, TreeTagger as described in [**?**] was used with a python wrapper.

4 Methodology

For this study there were three sets of features that were placed under study - bag-of-words features, text ngrams, and part of speech tags. The same approach was used for all the tasks which was more or less straightforward: feature extraction, then use the features for classification using a Support Vector Machine with a linear classifier and a default c value of 1, and then evaluate this through a 10 fold cross validation. Cross validation accuracies are compared to one another using the Wilcoxon signed rank test [18] to determine if the differences between the accuracies are statistically significant.

The Wilcoxon signed rank test [18] allows the comparison between two experiments done on the same data set without making any assumptions on the distributions from which the set is drawn. A confidence interval of 95% was used. Therefore, if there was an experiment between two settings and the returned p-value after a Wilcoxon signed rank test yields less than 0.05, the null hypothesis is rejected. This means that the values are statistically different. If it yields a value greater than 0.05, it means that values between the two experiments are not statistically different.

After determining which features are the most useful, we then reduce the number of features by ranking the importance through information gain. We determine the number the most number of features that gives an accuracy that is not statistically different from the accuracy given by the full number of features using Wilcoxon signed rank test [18]. We then use that reduced number of features to test the different kernels and different parameters.

4.1 Preprocessing

For each language, xml files from each user are read. Then the tweets taken from each user are extracted and concatenated into one line to form one training example. The examples are then transformed by putting them all in lower case. No stop words are removed. Hashtags, numbers, mentions, shares, and retweets were not processed or transformed to anything else. They were retained as is and therefore will correspond to another item in the dictionary of features. The resulting file is used for feature extraction. The resulting file is used for feature extraction.

4.2 Feature Extraction

The following set of features were extracted after preprocessing and the number of features extracted from each set is given in the table 2:

Bag-of-Words Two different bag-of-words features were extracted. The first is normalized term frequency. Terms are normalized by the number of terms in a training example. No terms were discarded. The second one is *tfidf* where terms were normalized by the inverse document frequency. No terms were also discarded.

Text Ngrams Text ngrams were the final set of features that was examined. Counts of bigrams and trigrams were taken after preprocessing. No normalization was done and no terms were discarded.

POS Ngrams Part of speech tags were also another set of features that was examined in this study. It was extracted by using a python wrapper to Schmid's TreeTagger program as detailed in [**?**]. Counts for unigrams, bigrams, and the combination of the two were used as features. The English parameter file for TreeTagger has 36 different tags, Italian has 38, 41 for Dutch, and 75 for Spanish. No terms were discarded.

	tf	tfidf	Text bi	Text tri	POS uni	POS bi
English	26264	26264	104435	147577	35	895
Dutch	8569	8569	29648	39286	39	792
Italian	12590	12590	36002	44051	38	443
Spanish	24688	24688	87313	122130	68	1999

Table 2. Number of features extracted

4.3 Feature Reduction

After extracting *tfidf* features, we want to reduce the number of features in the text. To do that, classification was done using Support Vector Machines [3] with a linear kernel, choosing C=1 as the default. But instead of using all the *tfidf* features, different number of features were tested. The number of features were ranked by either information gain or gain ratio. The results for the accuracies by varying the feature set is given in table 5 in the succeeding chapter. However, it is suffice to say that for the succeeding experiment on polynomial kernels and radial basis function kernels, the number of features selected will be 9000 and 7000 for age and gender classification respectively. The discussion on selecting the number of features is given in the succeeding chapter.

4.4 Parameter Tuning

Using Support Vector Machines [3] entails the use of kernels that maps the features into a different space such that separation would be done in the new space. In the earlier section, only the linear kernel was used but we further used polynomial kernels and radial basis function kernels. Polynomial kernels maps the input features to another feature space that uses the polynomial function over the similarity of the input features. Mathematically, the kernel is given by equation 1 but scikits-learn implementation has a gamma to scale the dot product as in equation 2.

$$K(x, y) = (x^T y + c)^d \tag{1}$$

$$K(x) = (\gamma(x^T x) + c)^d \tag{2}$$

For our experiments, we set the gamma to be equal to 1 but the degrees d to vary between 1, 2, and 3. For both age and gender classification, c varies between 0.0001, 0.001, 0.1, 1, 10, 1000, 10000.

Radial basis function kernel is another kernel explored. Mathematically, it is given by equation 3. It is similar the scikits implementation but with a regularization factor c.

$$K(x, x') = exp\left(-\frac{\parallel x - x' \parallel^2}{2\sigma^2}\right) \tag{3}$$

For both age and gender classification, σ and c were chosen to be one among 0.0001, 0.001, 0.1, 1, 10, 1000, and 10000.

4.5 Classification with string substitution

One of the interesting experiments is to check if the addition of some twitter specific features could affect the classification. For this experiment we only look into hashtags and links because user mentions are already handled previously since users have been anonymized. The treatment is simple and done in the preprocessing side. After taking the text from the html and converting them into lowercase, string substitution was performed on links and hashtags. All links were transformed into the text " LINK_HERE " while hashtags were turned into " HASHTAG_HERE ". The idea is that each link and hashtag wont be considered as a unique and that it would just add to the number of links or hashtags. After string substitution, *tfidf* was used and then features were ranked based on information gain. Top 9000 and 7000 words were used age and gender classification respectively. Finally, classification was done using the optimal settings given by the previous experiments and that given in the succeeding chapter.

4.6 Using Information of the Other Class as a Feature

Another interesting experiment is that of classification when using the information from the other class as a feature. For example, the results of gender classification will be used as a feature for age classification. And this will be done vice-versa with gender classification.

4.7 Exploring Ensemble Methods

Finally, a comparison with ensemble methods was also done. Two different ensemble methods were considered - Random Forests and AdaBoost. Random Forests is one of the ensemble methods wherein results are averaged in the end. The full details of the algorithm are given by the paper of Breiman in [2]. The idea is to build a multitude of decision trees and averaging their prediction results. The trees built will vary since they are built by taking a random sample with replacement from the training set. Furthermore, the features will also vary since it will be a random subset of features that are selected when the splitting is done in the training phase.

Adaptive Boosting or AdaBoost is another ensemble method formulated by Freund and Schapire in [4]. The idea is to combine different weak models to produce a powerful ensemble. It is considered adaptive since the succeeding weak learners give more weight in classifying correctly instances in the training set which was misclassified by previous weak learners.

5 Results and Discussion

5.1 Evaluating the Effect of Feature Types on Classification

	BoW		Text Ngrams		POS Ngrams		
	tf	tfidf	bigrams	trigrams	unigrams	bigrams	uni+bi
gender	0.511	0.683	0.623	0.652	0.659	0.659	0.660
age	0.395	0.691	0.638	0.553	0.579	0.611	0.593
average	0.453	**0.687**	**0.631**	0.602	0.619	**0.635**	0.626

Table 3. Accuracy results for age and gender classification using different types of features. Highest accuracy is given in bold face.

5.2 Feature Selection

The results for feature selection are given in table 5. The highest accuracy could be attained when using all the different features. However, Wilcoxon signed rank test [18] was done between the experiment with the highest accuracy and other

	BoW	Text Ngrams	POS ngrams		
	tf vs tfidf	bi vs tri	1v2	1v3	2v3
gender	**0.0019**	0.1620	0.4057	0.7624	0.7624
age	**0.0002**	0.1041	0.3258	0.9097	0.7337

Table 4. Table of p-values using Wilcoxon signed rank analysis when comparing cross validation accuracies between among bag of words features, text ngrams features, and among POS ngrams features.

subsequent experiments within each task and each ranking function. For example, in age classification, information gain features were used and the experiment that used 26263 features was compared to the other experiment that used 10000, 9000, and so on. The numbers in boldface are those which are not statistically different from each other. Therefore, for further experiments, 9000 features were used for classification while 7000 features were used for gender classfication. Information gain ranking was used instead of gain ratio because the words given by information gain were subjectively more descriptive than those given by gain ratio.

	Age		Gender	
Num Features	info gain	gain ratio	info gain	gain ratio
26263	**0.6913**	**0.6913**	**0.6825**	**0.6825**
10000	**0.6321**	**0.6321**	**0.6167**	**0.6167**
9000	**0.6321**	**0.6321**	**0.6163**	**0.6163**
8000	0.6188	0.6188	**0.6233**	**0.6233**
7000	0.6188	0.6188	**0.6300**	**0.6300**
5000	0.6188	0.6188	0.4863	0.4863
2000	0.6188	0.6188	0.4983	0.4983
1000	0.6188	0.6188	0.4983	0.4983
700	0.6188	0.6188	0.4921	0.4921
500	0.6188	0.6188	0.4921	0.4921
300	0.6188	0.6188	0.4921	0.4921
200	0.6188	0.6125	0.4921	0.4921
100	0.6188	0.3950	0.4921	0.4921

Table 5. Accuracies that result from using different number of important features ranked by information gain and gain ratio. SVM linear kernel with C=1 was used.

5.3 Different Kernels

The results for age classification using support vector machines with polynomial kernel is given by table 6. The highest accuracy obtained was 80.92%. This was obtained from three different settings, degree 3 with C to be either 10, 1000, 10000. However, checking on Wilcoxon signed rank test [18], these results are not statistically significant to 80.25% given by settings with degree 2 and C to be either 10, 1000, 10000. These results are also not statistically significant against 75% with degree 3 and C to be 1. The results shown in boldface are the highest accuracies that are not statistically different from each other.

C	degree		
	1	2	3
0.0001	0.6321	0.6192	0.5121
0.001	0.6321	0.6192	0.5121
0.1	0.6321	0.6254	0.5992
1	0.6321	0.7104	**0.7500**
10	0.6321	**0.8025**	**0.8092**
1000	0.6321	**0.8025**	**0.8092**
10000	0.6321	**0.8025**	**0.8092**

Table 6. Accuracy results of using SVM with polynomial kernel on age classication task with different C and degree parameters using top 9000 informative features ranked by information gain.

The results for age classification using support vector machines with radial basis function kernel is given by table 7. Results show that the highest achieved accuracy was 80.92%. It was obtained from the kernel with gamma to be 0.001 and with a C to be 10000. This wasnt statistically different from the settings that gave 80.25%, 69.71%, 68.46%, and 65.17% which are all shown in boldface.

C	gamma						
	0.0001	0.001	0.1	1	10	1000	10000
0.0001	0.3950	0.3950	0.3950	0.3950	0.3950	0.3950	0.3950
0.001	0.3950	0.3950	0.3950	0.3950	0.3950	0.3950	0.3950
0.1	0.3950	0.3950	0.3950	0.4746	0.3950	0.3950	0.3950
1	0.3950	0.3950	0.6054	**0.6517**	0.5933	0.3950	0.3950
10	0.3950	0.3950	**0.6846**	**0.8025**	0.6196	0.3950	0.3950
1000	0.6188	**0.6971**	**0.8025**	**0.8025**	0.6196	0.3950	0.3950
10000	0.6971	**0.8092**	**0.8025**	**0.8025**	0.6196	0.3950	0.3950

Table 7. Accuracy results of using SVM with radial basis function kernel for age classification with different C and gamma parameters using top 9000 informative features ranked by information gain.

Comparing the two different kernels for age classification including different parameters, the settings settled for would be with the polynomial kernel of degree 3 with C equal to 10.

The results for gender classification using support vector machines with polynomial function kernel is given by table 8. Results show that the highest achieved accuracy was 79.54%. This comes from six different settings. The first three were when the degree is 2 and C was chosen to be either 10, 1000, or 10000. The last three settings were when degree is 3 with C also either 10, 1000, or 10000. These results are not statistically different from that which gave 70.25% which are all written in boldface.

C	degree		
	1	2	3
0.0001	0.6300	0.4983	0.4733
0.001	0.6300	0.4983	0.4733
0.1	0.6300	0.5233	0.4733
1	0.6300	0.6367	**0.7025**
10	0.6300	**0.7954**	**0.7954**
1000	0.6300	**0.7954**	**0.7954**
10000	0.6300	**0.7954**	**0.7954**

Table 8. Accuracy results of using SVM with polynomial kernel for gender classification with different C and degree parameters using the top 7000 informative features ranked by information gain.

The results for gender classification using support vector machines with radial basis function kernel is given by table 9. Results show that the highest achieved accuracy was 80.79%. This comes from the settings with gamma to be 1 and C to be 10. However, these arent statistically different from settings which gave 80.21%, 80.13%, and 79.54% accuracy, all of which are written in boldface in the table.

Comparing the two different kernels for gender classification including different parameters, the settings settled for would be with the radial basis function kernel with gamma equal to 1 and C equal to 10.

5.4 String Substitution

Comparisons between the the best from previous parameter tuning experiments and that where links and hashtags were substituted with a different string tag is shown in table 10. We can see that using the same parameters for the classifier, the performance drops when there's string substitution but the results are not statistically different.

5.5 Using Information from Other Class

Comparisons between the the best from previous parameter tuning experiments and another method where the training set features is augmented by the information of the other class is given by table 11. We can see that age classification with features augmented by gender classification results has 2% increased accuracy but it is not statistically different. For gender classification, the results are almost the same and are also not statistically different.

C	gamma						
	0.0001	0.001	0.1	1	10	1000	10000
0.0001	0.5233	0.5233	0.5233	0.5171	0.4921	0.5171	0.4733
0.001	0.5233	0.5233	0.5233	0.5171	0.4921	0.5171	0.4733
0.1	0.5233	0.5233	0.5233	0.5171	0.4921	0.5171	0.4733
1	0.5233	0.5233	0.5233	0.6167	0.6629	0.4796	0.4733
10	0.5233	0.5233	0.6238	**0.8079**	0.7025	0.4796	0.4733
1000	0.5233	0.6367	**0.8021**	**0.8013**	0.7025	0.4796	0.4733
10000	0.6367	**0.7954**	**0.8021**	**0.8013**	0.7025	0.4796	0.4733

Table 9. Accuracy results of using SVM with radial basis function kernel for gender classification with different C and gamma parameters using the top 7000 informative features ranked by information gain.

Task	Accuracy			Settings			
	With String Sub	Best from Previous Experiments	P-value	kernel	gamma	degree	C
Age	0.7892	0.8092	0.8206	poly	N/A	3	10
Gender	0.7442	0.8079	0.4963	rbf	1	N/A	10

Table 10. Comparison between the best accuracy for age and gender classification from previous experiments against a classifier trained with links and hashtags string substitution.

Task	Accuracy			Settings			
	With String Sub	Best from Previous Experiments	P-value	kernel	gamma	degree	C
Age	0.8292	0.8092	0.5453	poly	N/A	3	10
Gender	0.8021	0.8079	0.8206	rbf	1	N/A	10

Table 11. Comparison between the best accuracy for age and gender classification from previous experiments against a classifier trained with a feature set added with the information of the other class.

5.6 Ensemble Methods

Accuracy results for random forests are given in table 12. The highest accuracy obtained for age and gender classification using random forests are 61.12% and 70.29% respectively. However, after performing Wilcoxon signed rank test [18] towards the other results within each task, the p-values show that the values are not statistically different from each other.

n-estimators	Age	Gender
10	0.6058	0.5983
100	0.6046	0.6500
1000	0.6112	0.7029
2000	0.6046	0.6762
5000	0.6108	0.6963
10000	0.6175	0.6967

Table 12. Accuracy results for age and gender classification using forests of randomized trees.

Accuracy results for AdaBoost are given in table 13. The highest accuracy obtained for age and gender classification using AdaBoost are 54.79% and 75.00% respectively. However, after performing Wilcoxon signed rank test [18] towards the other results within each task, the p-values show that the values are also not statistically different from each other.

n-estimators	Age	Gender
50	0.5479	0.6787
100	0.5217	0.7187
150	0.5475	0.7183
200	0.5146	0.7500
250	0.5275	0.7308

Table 13. Accuracy results for age and gender classification using AdaBoost.

6 Conclusion and Recommendation

There are various conclusions that could be found from the results. First, we can conclude that using minimal preprocessing, using the top 9000 features as ranked by information gain for age classification would yield the same results as using all the different features, from a statistical standpoint. This also goes true for gender classification but with the top 7000 features. Second, looking into the differences in kernel functions, the results show that the highest accuracy that could be achieved were 80.92% and 80.79% for age and gender respectively. These accuracies could be attained could from either polynomial kernels or radial basis function kernels. For instance, for age classification, the highest accuracy could either be obtained with a polynomial kernel with degree 3 and C to be 10 or it could also be a through a radial basis function with gamma to be 0.001 and C to be 10000. For gender classification, the similarity exists as well. A polynomial kernel could be used with the second degree and C to be 10000 and it will yield the same accuracy as the one with a radial basis function with gamma to be 0.01 and and C to be 10000. For the purposes of our next experiments, we fixed the parameters such that for age classification, we an SVM with polynomial kernel with degree to be 3 and C to be 10, while for gender classification, we use a radial basis function kernel with gamma to be 1 and C to be 10. Third, looking into features that are specific to twitter data, we see that substituting a hyperlinks and hashtags lower the accuracy as compared to the best among the previous experiments, although they arent statistically different. Fourth, we can also see that using information given by another class does not improve the classification results. Finally, results from ensemble methods are not statistically different from each other and accuracy results are lower than the others.

For future work, it should be noted that the current features are very minimal. The preprocessing is minimal. The features are *tfidf* ranked by information gain. For twitter specific features, only the hashtags, user mentions, and hyperlinks were identified and checked. It would be better if other features specific to twitter could be incorporated. Examples of such would be emoticons and character flooding. Experiments with character ngrams is also another direction to check.

References

1. Shlomo Argamon, Moshe Koppel, James W Pennebaker, and Jonathan Schler. Automatically profiling the author of an anonymous text. *Communications of the ACM*, 52(2):119–123, 2009.
2. Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.
3. Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.
4. Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.
5. A Pastor López-Monroy, Manuel Montes-y Gómez, Hugo Jair Escalante, and Luis Villaseñor-Pineda. Using intra-profile information for author profiling.
6. Adrian Pastor Lopez-Monroy, Manuel Montes-y Gomez, Hugo Jair Escalante, Luis Villasenor-Pineda, and Esaú Villatoro-Tello. Inaoe's participation at pan'13: Author profiling task. In *CLEF 2013 Evaluation Labs and Workshop*, 2013.
7. Suraj Maharjan, Prasha Shrestha, and Thamar Solorio. A simple approach to author profiling in mapreduce.
8. James Marquardt, Golnoosh Farnadi, Gayathri Vasudevan, Marie-Francine Moens, Sergio Davalos, Ankur Teredesai, and Martine De Cock. Age and gender identification in social media. *Proceedings of CLEF 2014 Evaluation Labs*, 2014.
9. Michał Meina, Karolina Brodzinska, Bartosz Celmer, Maja Czoków, Martyna Patera, Jakub Pezacki, and Mateusz Wilk. Ensemble-based classification for author profiling using various features. *Notebook Papers of CLEF*, 2013.

10. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
11. Francisco Rangel, Paolo Rosso, Irina Chugur, Martin Potthast, Martin Trenkmann, Benno Stein, Ben Verhoeven, and Walter Daelemans. Overview of the 2nd author profiling task at pan 2014. In *Proceedings of the Conference and Labs of the Evaluation Forum (Working Notes)*, 2014.
12. Francisco Rangel, Paolo Rosso, Moshe Moshe Koppel, Efstathios Stamatatos, and Giacomo Inches. Overview of the author profiling task at pan 2013. In *CLEF Conference on Multilingual and Multimodal Information Access Evaluation*, pages 352–365. CELCT, 2013.
13. Francisco Rangel, Paolo Rosso, Martin Potthast, Benno Stein, and Walter Daelemans. Overview of the 3rd Author Profiling Task at PAN 2015. In L Cappellato, N Ferro, J Gareth, and E San Juan, editors, *CLEF 2015 Labs and Workshops, Notebook Papers*. CEUR-WS.org, 2015.
14. K Santosh, Romil Bansal, Mihir Shekhar, and Vasudeva Varma. Author profiling: Predicting age and gender from blogs. *Notebook Papers of CLEF*, 2013.
15. Jonathan Schler, Moshe Koppel, Shlomo Argamon, and James W Pennebaker. Effects of age and gender on blogging. In *AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs*, volume 6, pages 199–205, 2006.
16. Julio Villena-Román and José Carlos González-Cristóbal. Daedalus at pan 2014: Guessing tweet author's gender and age.
17. Edson RD Weren, Viviane P Moreira, and José PM de Oliveira. Exploring information retrieval features for author profiling—notebook for pan at clef 2014. *Cappellato et al.[6]*.
18. Frank Wilcoxon. Individual comparisons by ranking methods. *Biometrics bulletin*, pages 80–83, 1945.