User Profiling in Social Media

Krysia Llull Cespedes*
Université de Montréal
krysia.llull.cespedes@umontreal.ca

Swetha Srikari Maganti*
Université de Montréal
swetha.srikari.maganti@umontreal.ca

ABSTRACT

In this document, we present the results for our work on the User Profiling in Social Media project using different machine learning techniques. We conducted datatset analysis via data exploration and visualization. We performed feature and model selection using K-fold cross validation and we obtained results that outperformed the baselines scores on each of the tasks of predicting age, gender and OCEAN traits of the users.

CCS CONCEPTS

• **Computing methodologies** → Machine learning.

KEYWORDS

datasets, age and gender classification, personality prediction

ACM Reference Format:
Krysia Llull Cespedes and Swetha Srikari Maganti. 2018. User Profiling in Social Media. In *IFT 6758: Data Science.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

In this document, we describe and report the work done in the User Profiling in Social Media using a Facebook dataset. We were tasked to predict from 9500 users' entries the gender (binary classification), age group (multi-class classification), and OCEAN personality traits (regression). Each user entry contained profile information (labels), NRC Emoticon Lexicon for Nuances of Emotion features, Linguistic Inquiry and Word Count (LIWC) features and Oxford features extracted from profile picture of some users.

After going through the initial phases of exploring and visualizing the data, we proceeded to select the subset of features and algorithms that displayed better performance for each of the tasks. The following sections describe the work done.

2 DATASET AND METRICS

The dataset contained profile information of 9500 Facebook users, collected with the *MyPersonality* application, that we utilized as train data. Also, a remote test set was used to evaluate our submissions in a remote server. This test data was divided into public test –with 334 users entries– and hidden test containing 1334 users.

*Both authors contributed equally to this work.

IFT6758, 2019, Montreal

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/1122445.1122456

2.1 Dataset Analysis

We analyzed the train dataset composition. For the task of predicting users genders, we found that the train data provided was gender-imbalanced with larger percent of user being female (Fig. 2). Also, for the age prediction task, we found that the group of users in the range "xx-24" represented a larger portion of the dataset (Fig. 1). Furthermore, we also investigated a different aspect of the OCEAN trait's label. As a regression task, we analyzed the mean and variance of each trait in the dataset and estimated the number of outliers for each trait, as shown in Fig. 3.

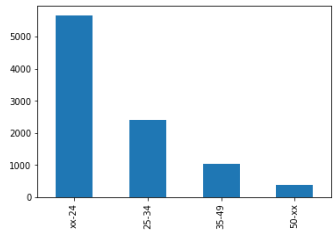

Figure 1: Distribution of Facebook user's (train data) Age group

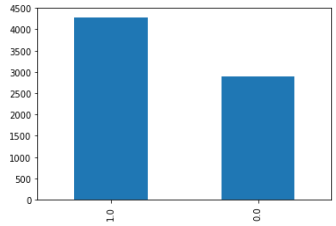

Figure 2: Distribution of Facebook user's (train data) gender

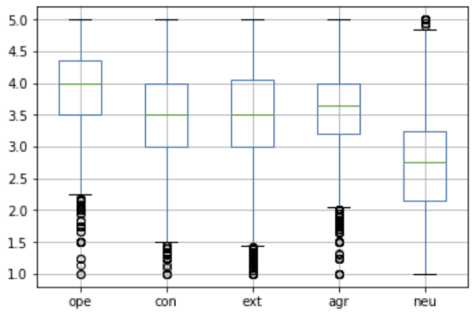

Figure 3: Distribution of the user's OCEAN traits

2.1.1 Feature Analysis. For each of the task described above, we had access to a set of features represented by Linguistic Inquiry Word Count, NRC Emotion Lexicon, Oxford features and Relations. We also used each user profile information as it contains the user's id, the target classes and real values for each traits that comprised each of the tasks.

The Linguistic Inquiry Word Count (LIWC) per user was provided as a summary of the user's linguistic traits. This was required as a privacy constraint related to the idea of not having access to the user's messages. Instead, LIWC gives us a total word count (WC), mean amount of words per sentence (WPS), and a rich set of summary variables like analytical thinking, clout, authenticity, and emotional tone scores [1]. In total, LIWC is comprised of 81 numerical features related to each user's id.

We also used the NRC Emotion Lexicon [2] that provides information that can be used to identify personality traits. It contains the user's perceived score [0,1] from 10 emotion categories: positive, negative, anger, anticipation, disgust, fear, joy, sadness, surprise, trust.

We also had access to the the Oxford features[3] for each user. The Oxford features provide facial point information from each user's profile picture. It contains 65 metrics of the spatial location, measurements and appearance of the physiognomy of detected human faces in the images. Particularly, for this datatset, each user had either none or multiple Oxford entries.

Lastly, the Relation features provided us with the relations of users with Facebook pages. The dataset contains the user's liked pages in an occurrence scenario, and because of that, we transformed it to a matrix-based representation –*bag of words*– of pages likes per user. We filtered out this features and included only pages with at least 15 connections. The transformed dataset resulted in 1871 feature columns per user entry.

3 METHODOLOGY

We started by exploring and visualizing the data. For this, we performed supervised feature selection using an exhaustive search, and model selection using K-fold cross-validation.

3.1 Methods

In this section, we will explain the primary concepts and models applied in this project. We used **Supervised feature selection:** to find relevant features that separate samples from different classes (classification) or approximate target variables (regression). As our exhaustive search methods, we used **wrapper methods** to find an optimal subset of features leveraging an underlying model. As model validation and selection technique, we used **K-fold cross validation** to asses how the results obtained during training would generalize to an independent test set.

We also evaluated a set of different models. The first model we evaluated was **Linear Regression**, a linear approach to model the relationship between a response variable (or dependent variable) and one or more features (or independent variables). We also explored **Logistic regression**, a statistical model that uses a logistic function to model a binary response variable (or dependent variable). It is also used for multi-class classification tasks. We also implemented a **Multi Layer Perceptron**, an artificial neural network that consists of multiple layers connected in a directed acyclic graph (Feed-Forward architecture) that is usually trained trough back-propagation and iterative methods (Stochastic Gradient Descent) as learning techniques.

We also explored boosting methods. We used **AdaBoost**, a classifier that is a meta-estimator which begins by fitting a classifier on the original dataset, and then sequentially fits additional copies of the classifier on the same dataset but on incorrectly classified instances whose weights are adjusted such that subsequent classifiers focus more on difficult cases. Also, we used **Gradient Boosting**, a method that builds an additive model in a forward stage-wise fashion; it allows for the optimization of arbitrarily differentiable loss functions.

Lastly, we tried a simple **K Nearest Neighbors classifier**, a flavour of non parametric methods that used for classification based on the vote of neighbor examples.

In general, the framework that guided this project is described as in Fig. **??** below.

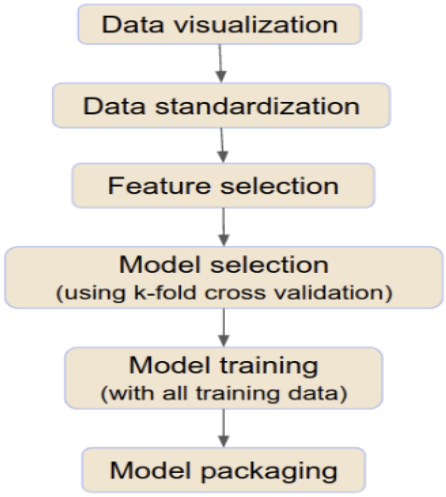

Figure 4: Methodology used in the project

3.2 Metrics

For evaluating the model performance, we used accuracy for the classification tasks of gender prediction and age prediction. For the regression task of prediction the users OCEAN traits, we used Root Mean Squared Error (RMSE).

4 EVALUATION

For each one of the tasks we evaluated the candidate methods over the available training sets and their combinations.

4.1 Age Classification

For classifying the age of the facebook users into either one of the 4 groups : "$xx-24$", "$25-34$", "$35-49$" or "$50-xx$", we trained several

[1] http://liwc.wpengine.com/interpreting-liwc-output
[2] https://pdfs.semanticscholar.org/45d0/660b7bbf60f53be75e4c263bd7c135b66a1d.pdf
[3] https://azure.microsoft.com/en-ca/services/cognitive-services/face

classifiers like K-nearest neighbors, Logistic Regression, Adaboost classifier, etc and tried different possible combination of features (LIWC, Oxford, NRC and Relation).

Initially, we started by setting a baseline model which predicts the most frequent age group in the train dataset provided which is $xx - 24$. This model gave an accuracy of 0.594. Further, we tried different models to beat the baseline model.

We began with feature selection and chose one feature at a time to train the classifiers and found that the classifier could not extract more information when any one of the 4 features is used. Out of all the 4 features, classifier trained using LIWC features outperformed the classifiers trained with other features. The reason for this is that there are some linguistic features such as word count, words per sentence, etc which are similar for users of certain age group.

Other method we used is the combination of all the 4 features which increased the dimensionality making the training and validation process slow. It had so many irrelevant and redundant features that did not improve the accuracy.

We tried using 2 features together and out of all the possible combinations, LIWC and Oxford features gave better accuracy. Though this combination gave slightly better results than using LIWC features alone. It is clear from this result that oxford features are not very informative compared to LIWC features for age classification.

To further improve the accuracy, we created a **Bag of Page Likes** which is a sparse matrix with columns having most popular like ids and rows with all the user ids. It is filled with 1's if the user has liked a particular page else it is filled with 0's. Even though, the number of features is very large, it gave much improved results.

As the Bag of page likes feature gave better results than other features for every classifier we tried, we chose Bag of page likes as our best feature and began model selection.

Though KNN classifier gave good results with LIWC features, it became very slow during testing and performed badly when Bag of Page Likes are used which could be because of the curse of dimensionality.

The Logistic regression classifier gave very good results with accuracy of 64.46% with Bag of Page likes. This could be due to the fact that linear classifiers work well with sparse data [9].

While Adaboost Classifier performed better than KNN classifier, it's accuracy was less than Logistic Regression classifier.

We selected the best feature and classifier out of all these features and classifiers for age classification by 10 fold cross validation.

Fig 5 shows the accuracies obtained with Logistic Regression classifier trained on different features.

4.2 Gender Classification

Similar to Age classification task, to classify the gender of a facebook user to either male or female, we trained several classifiers and tried different possible combination of features (LIWC, Oxford, NRC and Relation).

We set a baseline model which predicts the most frequent gender in the train dataset which turns out to be **female**. It gave an accuracy of 0.591.

To beat this baseline, we used hand-engineered features as well as the 4 features (Oxford, LIWC, NRC and relation) individually to train the classifiers. Some of the hand-engineered features are

BASELINE : 59.4

FEATURES	FEATURE SELECTION	NUMBER OF FEATURES	ACCURACY
Page like count	No	1	59.67
Pages liked by more than 5 users	**No**	**35562**	**64.45**
Pages liked by more than 15 users	No	14528	63.61
LIWC	No	81	61.41
Oxford	No	64	59.56

Figure 5: Accuracies obtained with Logistic Regression classifier trained on different features (10-fold cross validation)

combination of Oxford, LIWC and NRC features and combination of Oxford and LIWC features.

Oxford, LIWC and NRC features together resulted in almost same accuracy to that obtained with Oxford and LIWC features without NRC features. As joining all the 3 features slowed down the training and testing process due to increase in the number of dimensions (features), we used scikit-learn's SelectFromModel which is a meta-transformer for selecting features based on importance weights to reduce the number of dimensions.

The combination of 3 features resulted in a total of 155 features which after feature selection using scikit-learn's SelectFromModel [10] gave 37 features. These 37 features used to train the classifier did not outperform the 155 features with 0.15% accuracy.

Oxford and LIWC features together make 145 features and after feature selection it reduced to 69. These 69 feaures did not show any improvement as well and resulted in 2% less accuracy.

As Oxford + LIWC + NRC features gave similar results to that of Oxford + LIWC, we chose Oxford + LIWC features as the best out of these two as their dimension was less. We used these features without any feature selection because 76 features that were not included after feature selection improved the accuracy of the model by 2%. Fig 6 and Fig 7 show that Oxford and LIWC are important features as there is a difference between male and female's boxplots.

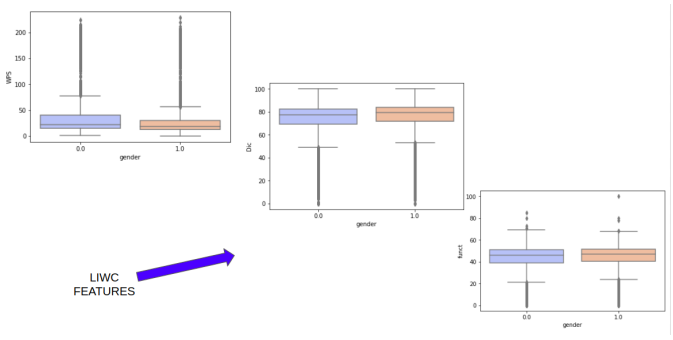

Figure 6: Boxplots of some LIWC features

We trained classifiers like K nearest neighbors classifier, Logistic Regression, Multilayer Perceptron classifier and Adaboost classifier

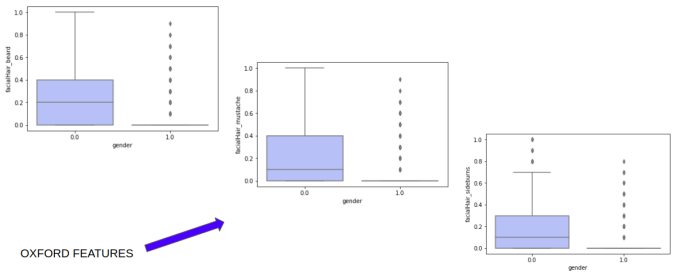

Figure 7: Boxplots of some Oxford features

with Oxford + LIWC features. Out of all these classifiers, Adaboost classifier gave highest accuracy of 81.3% with 10 fold cross validation.

Fig 8 shows the accuracies obtained with Adaboost classifier trained on different features.

BASELINE : 59.1

FEATURES	FEATURE SELECTION	NUMBER OF FEATURES	ACCURACY
Oxford	No	64	78.34
Oxford and LIWC	**No**	**145**	**81.32**
LIWC	No	81	66.6
Oxford	SelectFromModel	24	78.34
LIWC	SelectFromModel	26	66.46
Oxford and LIWC	SelectFromModel	69	80.98

Figure 8: Accuracies obtained with Adaboost classifier trained on different features (10-fold cross validation)

After selecting the best features and model for the 3 prediction tasks through k-fold cross validation, we trained the models using the entire train data with the best features and dumped the models in pickle files. We developed functions that modified the test data features to get the same features as train data so that the dataset remains independent and identically distributed.

For age prediction, we created a Bag of page likes model (a sparse matrix) similar to the training with the most popular like ids (35562 out of 536204 like ids)obtained during feature selection (used for training) and ignored any new like ids in the test data.

For gender prediction, we wrote a function that merged oxford and LIWC features and used these features to predict the gender of unknown facebook users.

4.3 Traits Prediction

After exploring and visualizing the data, we started to do some feature and hypothesis selection. For this task we evaluated the following model's families: Linear Regression, MLP and Gradient Boosting. In order to select the underlying model we would be using for feature selection, we individually tested the performance of each of the above candidate models over each dataset using a k-fold cross validation of 5 and manual hyper-parameter tuning per model. The Tables 1, 2, 3 (below) summarize the results obtained. Note that the best results are highlighted.

Table 1: Linear Regression performance per dataset

Trait	NRC	LIWC	Relation	Oxford
Ope	0.631	0.63	>1	0.634
Con	0.715	0.714	>1	0.719
Ext	0.808	0.824	>1	0.81
Agr	0.658	0.663	>1	0.661
Neu	0.792	0.792	>1	0.792

Table 2: MLP performance per dataset

Trait	NRC	LIWC	Relation	Oxford
Ope	0.643	0.714	>1	0.641
Con	0.723	0.811	>1	0.724
Ext	0.821	0.915	>1	0.817
Agr	0.668	0.757	>1	0.667
Neu	0.801	0.869	>1	0.797

Table 3: Gradient Boosting performance per dataset

Trait	NRC	LIWC	Relation	Oxford
Ope	0.63	0.624	**0.623**	0.632
Con	0.713	**0.708**	0.714	0.718
Ext	0.807	**0.8**	0.804	0.809
Agr	0.656	**0.652**	0.658	0.661
Neu	0.79	**0.788**	**0.788**	0.79

This initial set of tests showed us that the Gradient Boosting model gave us the best scores for each OCEAN traits when compared against all the other models. Therefore, we selected *Gradient Boosting* as the model used to perform feature selection. From this process, we obtained a subset of all the available features. We found that the LIWC features resulted to be, overall, the most informative features to this particular task (predicting OCEAN traits) – Table 3.

Then, we did an exhaustive search between all the possible features sets combinations. We found that by combining the features of LIWC, NRC Emotion Lexicon and Relations, we obtained the best overall performance across traits (except for Neuroticism). We present this results in Table 4.

Table 4 also summarizes the RMSE scores obtained per subset combination using Gradient Boosting (best results are highlighted). Getting the best results with Gradient Boosting[4] was not surprising as it leverages an ensemble of weak prediction models to form a stronger one, a technique known as boosting. [5].

Having selected the model and best performing features, we again performed a manual hyper-parameter search by tuning the

[4]https://en.wikipedia.org/wiki/Gradient_boosting
[5]https://en.wikipedia.org/wiki/Boosting_(machine_learning)

Table 4: Gradient Boosting scores per dataset combination

Trait	NRC_LIWC	NRC_R	LIWC_R	LIWC_NRC_R	All
Ope	0.623	0.6225	0.6203	**0.6201**	0.6202
Con	0.707	0.7105	0.7078	**0.7072**	0.7071
Ext	0.8	0.803	0.799	**0.7989**	0.799
Agr	0.652	0.6553	0.6523	**0.652**	0.652
Neu	0.787	0.7871	0.7865	0.7858	**0.7853**

learning_rate, max_depth and n_estimators of the Gradient Boosting estimator. We started with a learning rate 0f 0.01, max_depth of 4 and n_estimators of 100. We noticed that better results were obtained when adding more estimators and reducing at the same time the max_depth while keeping the initial learning rate. Table 5 shows the best scores per trait obtained by Gradient Boosting using k-fold cross validation of 5 and a configuration with learning_rate=0.01, max_depth=2, n_estimators=300.

Table 5: Gradient Boosting validation performance per OCEAN traits for the selected subset.

Ope	Con	Ext	Agr	Neu
0.620194	0.707204	0.798928	0.652045	0.785868

Finally, to prepare the model for predicting in the hidden test set, we trained a Gradient Boosting predictor with the configuration found in the model selection step on the full training set of the selected subset of features (NRC, LIWC and Relations). We deployed the predictors in the *submissions* folder where they were invoked in the *ift6758.py* script during the testing phases.

5 RESULTS

Table 6 shows the final scores obtained after the evaluation on the hidden test sets in the submission server. We also show the comparison between our best performing method and the baseline scores.

Table 6: Final results over the test set compared to the baselines.

Model	Age	Gender	Ope	Neur	Ext	Agr	Con
Baseline	0.594	0.591	0.652	0.798	0.788	0.665	0.734
Ours	**0.621**	**0.752**	**0.646**	**0.793**	**0.784**	**0.655**	**0.720**

Also, the plot in Fig 9 shows the comparison between the results obtained on the test data (scoreboard) using the selected features and model using cross validation. 'Ope' in bar plot refers to 'openness', 'Conscien' refers to 'Conscientious', 'Extro' refers to 'Extrovertness', 'Aggr' refers to 'Aggreableness' and 'Necr' refers to 'Neurotic'. Age and Gender are measured with accuracy metric while the 5 personality traits are with Root Mean Squared Error (RMSE).

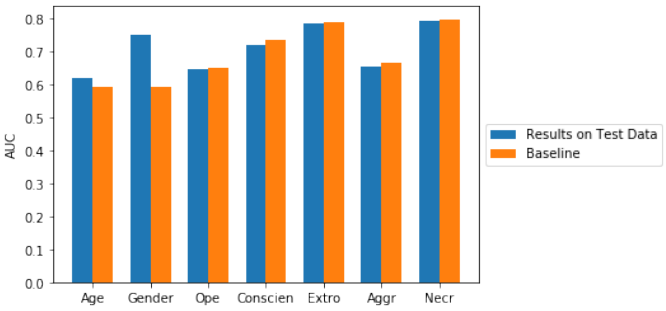

Figure 9: Comparison of the model performance with baseline

6 CONCLUSION AND FUTURE WORK

Even though our method performed well when compared to the baseline, we acknowledge that there is much more room for improvement.

- We would have loved to explore how to further filter the subset of features that ended up being used in the final algorithm and test that if using a more refined subset would give better results.
- We would have also liked to further explore the Oxford dataset, because even though we did not use it at all for predicting the OCEAN traits, we may have benefited from some of its most relevant features.
- We would also like to handle age classification as regression task initially and then grouping the predicted values to their respective age groups, although we are uncertain if this would actually improve our scores.
- Also, due to time constraints, we used in our implementation a simplified approach by that uses the same subset of datasets for predicting all traits. For future work, we aim to explore embedded methods for feature selection over the full set of features available in the dataset.
- Furthermore, we would have liked to explore better deep learning techniques as restricted our work with more traditional machine learning techniques.