Predicting User Features from Social Media data

François Mercier
francois.mercier.4@umontreal.ca
20167322

Nicolas Sauthier
nicolas.sauthier@umontreal.ca
932337

Zicong Mo
zicong.mo@umontreal.ca
20141760

Andrew Kristensen
drew.kristensen@umontreal.ca
20119706

Yifan (Andy) Bai
yifan.bai@umontreal.ca
20153885

Figure 1: The Maze: User08

ABSTRACT

Using various models throughout the duration of the project, we were able to beat the baselines set by the mean values and majority votes for the social media data we were given to work with. We used graph learning, neural networks, decision trees, and other methods in order to obtain these results.

KEYWORDS

datasets, social media, regression, classification, personality prediction

1 INTRODUCTION

With the increase in both number of social media platforms and amount of data being shared on them, predicting users' habits and preferences is becoming an increasingly important task for both marketing firms and online retailers. For this project, we were given information from users taken from Facebook.com and asked to predict their age, their gender, and 5 personality traits from this data. This meant we had to combine both regression methods and classification methods in order to achieve sufficient results, and we applied several different techniques during the course of the quarter in order to investigate the efficacy of each approach. We found that with the combined model of Gradient Boosting and Neural networks, we were able to achieve scores better than the baselines by a minimum of just under 1 percent and a maximum of just over 30 percent.

2 DATASETS AND LABELS

For the tasks described we had four datasets with information on every user relative to its profile picture, its textual information and the pages he or she liked.

2.1 Page-User Relationships

We were given the connection between users and pages in the form of edges between user nodes and page nodes in the bipartite graph. By parsing the csv file, we were able to build this graph and use it to make inferences. In total, there were 9500 users in the training set, with just over 536k unique pages, and just over 1.67 million edges in the total training graph.

From these, we can compute various metrics for the pages, and the distribution of standard deviations in the values derived from the pages' liked users can be found in figure **??** in section A.2.

2.2 Oxford

In order to use the profile picture information, and Microsoft®API called Faces (formerly Project Oxford) was used. This API transform an image into numerical information about the faces in the images. For each face detected in the image it gives:

- Coordinates of the edges of the rectangle containing the face,
- coordinates of specific point in the face (eyes, nose, mouth etc. See figure 10),
- Information about the angulation of the face,
- Information about the facial hair (beard, mustache and sideburns)

As we can see in figure 11, most of the profile pictures contains only one face. A minority contains up to three distinct faces and less than a quarter contains no faces in their profile picture.

2.3 NRC

NRC dataset is a set of features created using scores for 10 features, representing nuances of sentiments, from the NRC Word-Emotion Association (aka NRC Emotion Lexicon).

In the latter, the scores were created with the point-wise mutual information between each word and the presence of the feature (as an hashtag). In other words, higher the score is present, stronger the association between the word and the feature is supposed to be.

As we can see in the figure 7, scores distributions are skewed and there are some presences of outliers like the 0 and 1 value for the positive feature. Interestingly, using correlation matrix, figure **??**, we can cluster these 10 scores into two groups:

- one "positive" group with positive correlation between them and negative correlation with the other groups
- the "negative" group

This indicates the NRC features represent 5 scores for positive emotions and 5 for negative emotions.

2.4 LIWC

Linguistic Inquiry and Word Count, called LIWC, is a feature extraction method to capture structural component from text data. It contains several features such as word count, WC, or word per sentence, WPS.

These features are highly skewed for some of them and are positive real numbers, figure 9.

3 LABELS AND METRICS

A description of the label to predict and the metrics used to assess the prediction performance.

3.1 Labels

3.1.1 Age labels. For this task, the goal is to predict which age group the person self identifies. It would be impossible to classify the exact ages as there are 81 distinct age values with a minimum of 1 year and a maximum of 112, as seen in figure 3. Instead, we group them into four different classes (0: 24 and under, 1: 25 to 34 inclusive, 2: 35 to 49 inclusive, 3: 50 and over). Since all age values are positive integers with aforementioned extrema, all data corresponds to the four classes. This label set is highly imbalanced (0: 5669, 1: 2401, 2: 1045, 3:385).

3.1.2 Gender labels. In this task we are trying to predict which gender the person identified himself or herself on his or her profile.

In our dataset we have two classes (1: female and 0: male). There is no missing data and all the data corresponds to those two classes. This label is somewhat balanced the proportion of females being 0.57.

3.1.3 Personality scores. Personality scores are 5 real values between 0 and 5. They are part of the taxonomy for personality traits, called Big Five personality traits.

After analyzing the personality scores data, we found that:

- No presence of orphan (missing scores for some users)
- Skewed distribution for each scores, figure 4
- Some correlations between scores, figure 2.

By looking at correlation values, the scores can be divided in two groups:

- "positive" scores: agreeableness (Agr.), open (Opn.), extrovert (Ext.) conscientious (Con.).
- "negative" scores: neuroticism (Neu.).

3.2 Metrics

3.2.1 Age. The metric adopted for age classification is accuracy, defined as the number of correctly classified users over the total number of users. It is expressed as a fraction with a baseline of 0.594. Alternatively, it could also be expressed in percentages. Note that this metric is based on classification of age groups rather than exact values of age.

3.2.2 Gender. The metric used for the gender classification task is the accuracy. It is defined as the number of correctly classified users on the total number of classified users.

The baseline we used as a benchmark was a simple prediction of the majority class for all user which had, per definition an accuracy of 57%.

As for the personality scores, we used a bootstrapping testing approach in order to validate our approaches and our improvement.

3.2.3 Personality scores. For personality score labels, the official evaluation metric for the task was root mean square error, RMSE. This metric is commonly used for regression tasks, such as predicting personality scores.

$$RMSE(Targets, Predictions) = \sqrt{\sum_{i=1}^{n}(Targets_i - Predictions_i)^2}$$

As the primary goal for the task was to beat the baselines (mean prediction for personality scores), we also used bootstrapping techniques in order to get the distribution of RMSE in on validation test (20% of training set).

Using these distributions allowed us to compute the confidence intervals of our models' RMSE to use for comparison. Finally, a Z Test, with $H_0 : RMSE_{mean} \geq baselines$, was used to compare the mean from the bootstrapped RMSEs on validation set with the baseline scores. Thus, we used p-value to estimate our confidence about our models ability to achieve the primary goal.

4 METHODOLOGY

The project ran through a wide variety of approaches during different phases on the development. Our project made use of graph learning, neural networks, decision trees, and gradient boosting.

4.1 Algorithm

A description of the various algorithms used during the project

4.1.1 Neural Network. An Artificial Neural Network is a network consisting of layers of artificial neurons. Such networks can be trained as models for both classification and regression tasks. The structure we used is the basic feed-forward network with fully connected layers, also know as Multi-layer Perception. A high capacity Neural Network model with high dimensional input and low training accuracy can be built efficiently with our resources, which is the main reason we chose this model. The neural network is used for predicting personality scores using LIWC and NRC data.

For the Neural Network, we used the Keras implementation.

4.1.2 Logistic regression. Logistic regression is a generalized linear model that applies a sigmoid function on a linear function, in order

to predict a label between 0 and 1. The linear function gives weight to each of the features, while the logistic function is of the form:

$$p = \frac{1}{1 + e^{-(w_0 + w_1 \cdot x_1 + w_2 \cdot x_2 + \ldots)}}$$

Logistic regression is used for our initial gender prediction model, which takes all the facial data as input and predict the gender (binary classification problem).

For the logistic regression, we used the Scikit implementation.

4.1.3 Decision tree. Decision tree regressors are models which learn the hierarchical splits on attributes, in the form of a tree, in order to improve the prediction. The predictions are either the majority class of samples in leaves for classification or the mean of samples for regression.

For the decision tree regressor, we used the Scikit implementation.

4.1.4 Gradient Boosting. Gradient boosting models are ensemble models, typically decision trees, which are trained sequentially on residuals. The main idea is that learning on residuals forces models to focus on the hardest part of the data. This is an ensemble technique to address bias.

For the gradient boosting regressor, we used the Scikit implementation.

4.1.5 Adaboost. Adaboost is a specific subtype of gradient boosting algorithm developed in 1999[1] which iteratively gives each weak classifier (stump in our case) an optimal weight.

For the Adaboost classifier, we used the Scikit implementation.

4.2 Features Engineering

A description of the approaches used on the dataset to extract meaningful features.

4.2.1 Page Profiling. The earliest approach we took, the page profiling work utilized the bipartite nature of the relations graph. We extracted this data from the relations.csv included in the data by building each edge line by line in the training data. By creating an estimate of the kinds of users which like a page, we can use multiple pages to narrow down the values for the user we are searching for. By weighting pages differently, we can create an efficient predictor of new users. The weights are given by the inverse of the standard deviation of the values for each target value across the page's users. This means that when a page has a large number of similar users, then any new user who likes this page is probably similar to this group, so the inverse of a small standard deviation is going to give us a much larger weight than a page with few users or with a wide variety of users.

4.2.2 Page profiling for gender prediction. We also used this user-page relational graph approach for the gender to aggregate the data for the prediction task. Using the relations.csv data, We constructed an adjacency matrix for the 10'000 most frequent liked pages. The adjacency matrix was weighted in order to reflect the number of common users (edges) between two pages (nodes). We then reduced this sparse matrix with a SVD approach and took the 15 first dimension as features in our model. For each user, we used the average of the embedding of his or her pages. We imputed zeros for users without pages or with pages not in the 10'000 most frequent liked pages.

4.2.3 Text data for age prediction. The two text datasets, LIWC and NRC, were jointly used for predicting age groups. Since both are composed of associated features with unique but matching IDs of 9500 users, the work started from merging them based on user IDs. However, that left a feature set of dimension of 90, a recipe for slow training and overfitting at following stages. Based on correlation matrices of LIWC and NRC, 3 features were eliminated due to high correlation with others: 'Comma', 'funct', 'QMark'.

This strategy was proven sufficient for beating the baseline, meaning further feature engineering for this task was not needed.

4.2.4 Oxford feature engineering. In order to use our information we did implemented two pre-processing steps.

First, in order to have one face per profile, for profiles with more than one face to choose from, we chose the face which had the least angulation i.e. which was the most "facing". For profiles without any faces we imputed 0 for all values.

Second, we tried to standardize the face information. In order to do so we calculated the distance from the tip of the nose to all the point in the face seen in figure 10. We corrected those distance for relative angulation and we scaled them up or down in order to keep the distance between the eyes constant and equal to one.

In order to avoid redundant information, we removed distances that were highly correlated (over 0.99).

4.2.5 L1 feature selection. Feature selection is a preprocessing step to select a subset of available features.

L1 feature selection consists of training a linear model with an L1 regularizer term. The L1 regularizer term forces sparsity in the features used. Thus, the output of this model is a set of coefficients for each feature, which includes some null coefficients due to the regularization. The features with non zero coefficients are therefore the ones selected for use in the downstream model. The main benefit of this preprocessing is to decrease the number of input features in order to decrease variance, directly ddressing the curse of dimensionality, and to increase interpretability.

For this technique, we used the Scikit implementation.

5 RESULTS AND DISCUSSION

5.1 Age predictions

Early attempts focused on using the Naive Bayes' algorithm with Laplacian smoothing on facial features. However, it was quickly abandoned due to the lack of data, which amounts to around 2200 samples for the training set when excluding duplicates, as well as a failure in successfully feature-engineering, such as taking cartesian distances.

The effort then switched to using text datasets, namely LIWC and NRC, where three algorithms were tried: Random Forest, Extra Trees and Gradient Boosting. Merging two datasets, the first step was to eliminate features highly correlated with others. 'Comma', 'funct', 'QMark' were eliminated. Then the three classifiers were run on this feature set, with results summed up below.

As shown, Gradient Boosting already beat the baseline comfortably. Subsequently, we employed grid search and random search

Table 1: Age prediction results per algorithms, no resampling

Algorithm	CV Acc. on train set
Baseline	0.594
Random Forest	0.584
Extra Trees	0.585
Gradient Boosting	0.611

trying to further improve the accuracy, using the same 5-fold cross-validation strategy. After trials, the accuracy had improved to 0.615, and given it also achieved around 0.61 on the public test set, it was adopted as the final model. Given there was a delay in confirming the results on weekly evaluation, we also experimented with Nicolas' feature set, which reported a best accuracy of 0.657 on the public test set using the following parameters found from grid search: number of estimators = 70, learning rate = 0.45. However, since we later received confirmation of success with Gradient Boosting, this model was not included in the final submission.

Since the data is highly imbalanced, re-sampling was also used in an attempt to improve accuracy. Sklearn provides a handy set of packages, and the one used is Random Oversampler. The intuition is that since the data is highly imbalanced with over 5500 out of 9500 being below 24 years old while less than 1500 are age over 35, we will try to strategically over-sample those that have less data points. This time, the effort focused on improving Random Forest and Extra Trees. Both algorithms achieved over 90% in cross-validations, however, less than 50% on public test. This was due to the fact that since this resampler tried to optimize results on training set, it skewed the overall picture of the dataset and failed to generalize into other cases, such as the public test set. This is a classic example of overfitting, and one vivid lesson of what should be avoided in machine learning projects. Summarized below are the results on resampling investigations.

Table 2: Age prediction results per algorithms, with resampling

Algorithm	CV Acc. on train set
Baseline	0.594
Random Forest	0.922 (best 0.933)
Extra Trees	0.939 (best 0.946)

5.2 Gender predictions

Table 3: Gender prediction results per algorithms

Algorithm	Acc. on test set
Baseline	0.591
Logistic reg.	0.783
Adaboost	0.852

Our first approach with a linear model gave us surprisingly great results. However, when we analyzed the importance of each feature we saw that almost all of the precision came from the three facial hair features, which seamed logical. In that type of situation, a gradient boosting algorithm functions quite well because even the first iteration will use the facial hair information, but it will then boost its performance and make better use of less important features.

We also analyzed all of our features to determine which were useful by comparing results from only one type of information (oxford facial information, textual information or page information), combining two of them, or all three of them. With a bootstrapping approach, we could prove that combining the three types of information gave us the best results, as we can see in table 4. We can also infer from those results that the oxford face data was the most informative on gender prediction, followed by pages information and then text information.

Table 4: Statistics of bootstrapped accuracy for gender with different types of features

Data	Mean acc. on valid set	CI 95%
All three	0.8435	[0.8397, 0.8473]
Oxf+Text	0.8092	[0.8052, 0.8132]
Oxf+Page	0.8349	[0.8313, 0.8385]
Page+Text	0.7451	[0.7398, 0.7503]
Oxford only	0.7782	[0.7743, 0.7821]
Page only	0.7267	[0.7227, 0.7306]
Text only	0.6646	[0.6590, 0.6703]

5.3 Personality scores predictions

The models using Gradient Boosting and Neural Network were able to achieve the primary goals for the personality scores task.

All results are consolidated in the table 5.

The main takeaway are :

5.3.1 Page Profiling.

- Page Profiling worked well on the train set, but when predicting user values based on pages unseen in the training data, the performance took a noticeable hit. This comes from how we dealt with such pages, and may be reconcilable with an improved method.
- The larger the variance any score in the total population, the worse our model would be in accurately predicting that score. (See EXT and NEU predictions)

5.3.2 Decision Tree.

- Using L1 feature selection for decision trees helps to decrease the variance without hurting the bias. However, these models are not able to beat the baselines, especially the extrovert score.
- Using gradient boosting, we were able to beat the baselines but running several trials, each with a different seed, indicated a high variance of results.

- In order to control the variance during manual hyper parameters selection, bootstrap was useful in selecting the best gradient boosting models.
  This allowed us to be highly confident to beat 3 over 5 scores, fairly confident to beat the agreeable score and not confident to beat the extrovert score.
  As the bootstrapped models were trained on 80% of the training set and as the final models were trained on 100% of the training set, these estimates were fairly conservative. The selected models using this approach were actually able to beat the baselines in the test set.
  The results from this bootstrap approach is summarized in the table 6.

5.3.3 Neural Network.

- All 94 features of LIWC and NRC are used as input of the network. Log transform of the most skewed 15 features can slightly improve the results. Normalization after the log transform makes the training faster and more stable.
- Multiple models are trained, including a large and small network (based on the number of layers and neurons) with 50 epochs. We observe that both validation and training loss are similar across all models. This indicates a small model with fewer parameters should be used. Significant over fitting appears after five epochs, so dropout regularization is used on each layer. The following is the final structure we use. The performance is slightly better than the gradient boost random forest.

```
model.add(Dense(200, input_dim=91, activation="linear"))
model.add(Dropout(0.1))
model.add(Dense(200, activation="linear"))
model.add(Dropout(0.1))
model.add(Dense(100, activation="linear"))
model.add(Dropout(0.1))
model.add(Dense(50, activation="linear"))
model.add(Dropout(0.1))
model.add(Dense(20, activation="linear"))
model.add(Dropout(0.1))
model.add(Dense(5 ,activation='linear'))
model.compile(loss="mean_squared_error", optimizer="Nadam")
```

- We have tried to ensemble the result from two of our best models. The objective is to find the best combination of the output from gradient boosting and neural networks. Experiments included averaging two results, building another model to determine the weight of the results such as pairwise linear regression, MLP, and linear regression. We found that averaging is the best approach and has significantly better performance on all personality scores except Ext. RMSE.

6 CONCLUSIONS AND FUTURE WORK

- We have successfully predicted personal information with high accuracy using social media data, which raised the concern of privacy issues. We believe that user activity is as important as personal information. Both of them should be protected, and access should be reasonably restricted.
- We have spent a lot of effort into using LIWC and NRC data to predict the personality score. Although we have

Table 5: RMSE comparison between all models for personality scores

Model	Training (80%)	Validation (20%)
Mean baseline	N/A	Opn. 0.652 Neu. 0.798 Ext. 0.788 Agr. 0.665 Con. 0.734
Page Profiler *Single Model min 3 users*	Opn. 0.5390 Neu. 0.7021 Ext. 0.7130 Agr. 0.5836 Con. 0.6309	Opn. 0.6436 Neu. 0.8067 Ext. 0.7966 Agr. 0.6745 Con. 0.7156
Decision tree *91 features*	Opn. 0.6299 Neu. 0.7819 Ext. 0.799 Agr. 0.6487 Con. 0.7078	Opn. 0.6203 Neu. 0.8001 Ext. 0.8170 Agr. 0.6688 Con. 0.7119
Decision tree with L1 features selection *6 features*	Opn. 0.6284 Neu. 0.7917 Ext. 0.8030 Agr. 0.6618 Con. 0.7161	Opn. 0.6175 Neu. 0.7947 Ext. 0.8053 Agr. 0.6417 Con. 0.7143
Gradient Boosting *91 features*	Opn. 0.6072 Neu. 0.7770 Ext. 0.7994 Agr. 0.6526 Con. 0.6906	Opn. 0.6266 Neu. 0.7810 Ext. 0.8256 Agr. 0.6663 Con. 0.7012
Neural Network *91 features*	Opn. 0.6053 Neu. 0.7739 Ext. 0.7836 Agr. 0.6683 Con. 0.6947	Opn. 0.6198 Neu. 0.7907 Ext. 0.8041 Agr. 0.6578 Con. 0.7068
Averaging: Gradient Boosting and Neural network *91 features*	Opn. 0.6013 Neu. 0.7439 Ext. 0.7836 Agr. 0.6541 Con. 0.6985	Opn. 0.6206 Neu. 0.78911 Ext. 0.7987 Agr. 0.6551 Con. 0.7035

Table 6: Statistics of bootstrapped RMSEs for the Gradient Boosting model using 80/20 split

Scores	Mean on valid set	CI $\pm 2 * stddev$	Z test 1 tail	p-value $RMSE \geq baseline$
Opn.	0.620502	[0.597, 0.643]	-14.955	0.00%
Neu.	0.792591	[0.775, 0.809]	-3.444	0.03%
Ext.	0.810185	[0.790, 0.830]	12.206	100%
Agr.	0.662342	[0.639, 0.685]	-1.282	9.98%
Con.	0.706747	[0.684, 0.729]	-13.401	0.00%

used different kinds of models and even the model ensemble, we can only beat the baseline by a small margin. It could be evidence of insufficient input features. The model can memorize the average of personality scores and beat the baseline by chance. A possible future exploration could be building a graph model using a page like data and combine the result with the current model.

- We would like to investigate a better method of imputation in the page profiling code when a page lacks sufficient users or information. This may lead to better results, and especially better generalization, as we could then more reliably use the information we already have from the train set in our new predictions.
- The high-level strategy of our project is to build three separate models, each of them predict age, gender, and personality scores. However, we found that there is a correlation between the target features. For example, based on the personality scores, we are able to predict the gender. One idea of improving the score is that we could route the output of a model to the input of another model, or implement hybrid fusion between models.

REFERENCES

[1] Robert Schapire and Yoram Singer. [n. d.]. Improved Boosting Algorithms Using Confidence-rated Predictions. ([n. d.]).

A PERSONALITY LABELS

A.1 Correlation matrix

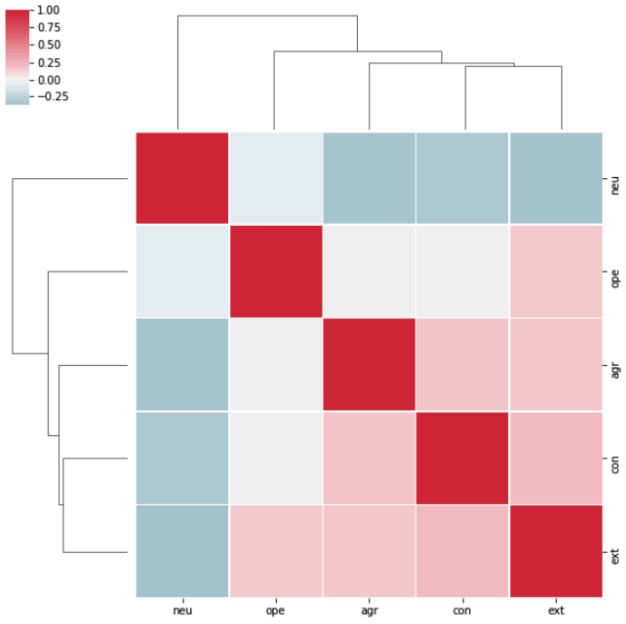

Figure 2: Personality correlation matrix

A.2 Distributions

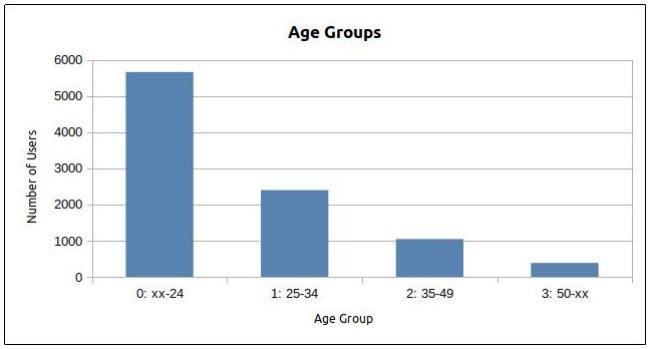

Figure 3: Age Group Distribution

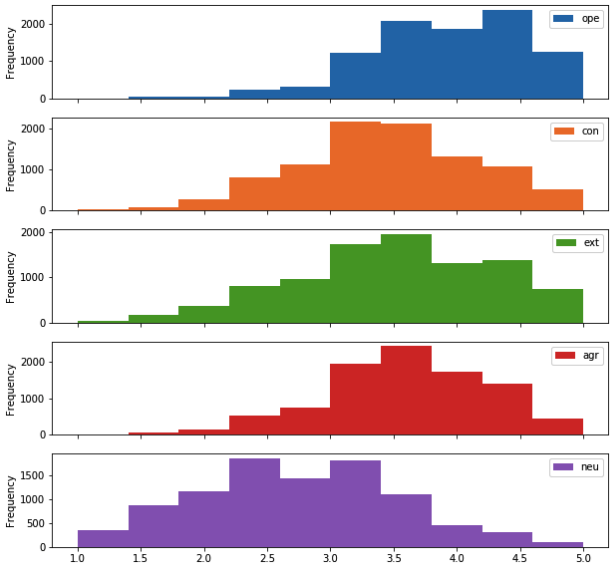

Figure 4: Personality Distribution

B NRC DATASET

B.1 Correlation matrix

B.2 Features distributions

C LIWC DATASET

C.1 Correlation matrix

C.2 Features distributions

D OXFORD DATASET

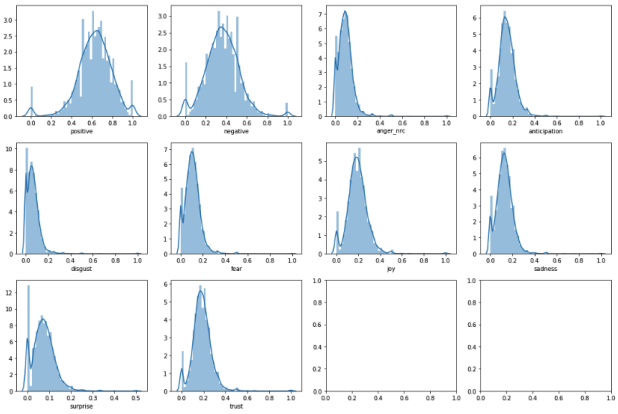

Figure 7: NRC features distribution

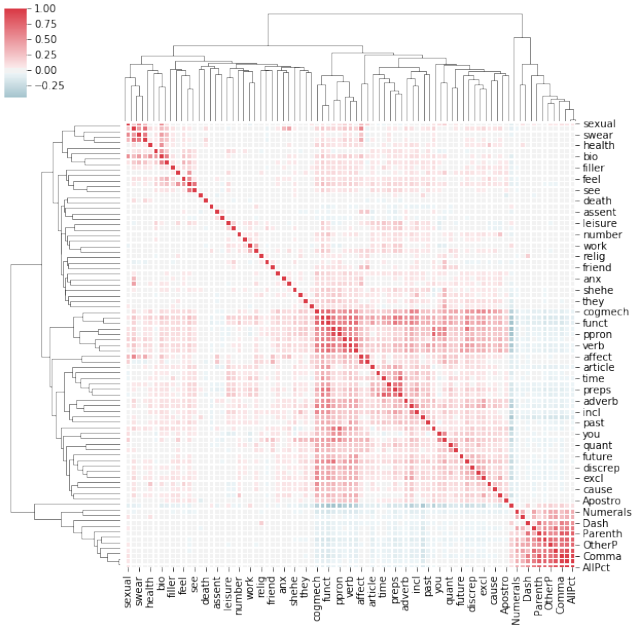

Figure 8: LIWC features correlation matrix

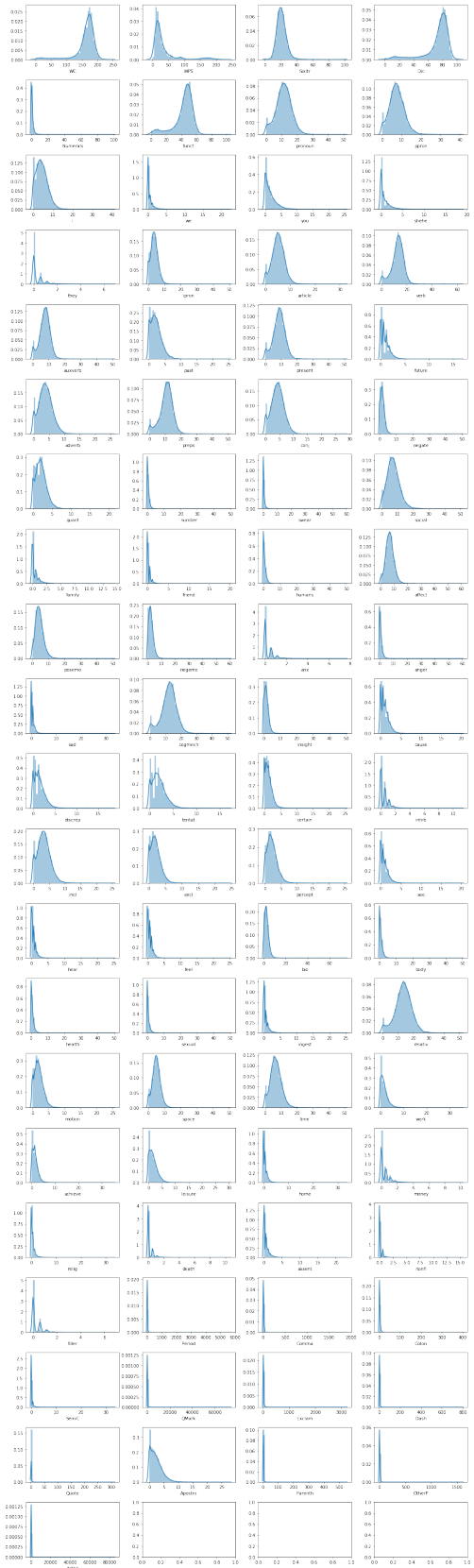

Figure 9: LIWC features distribution

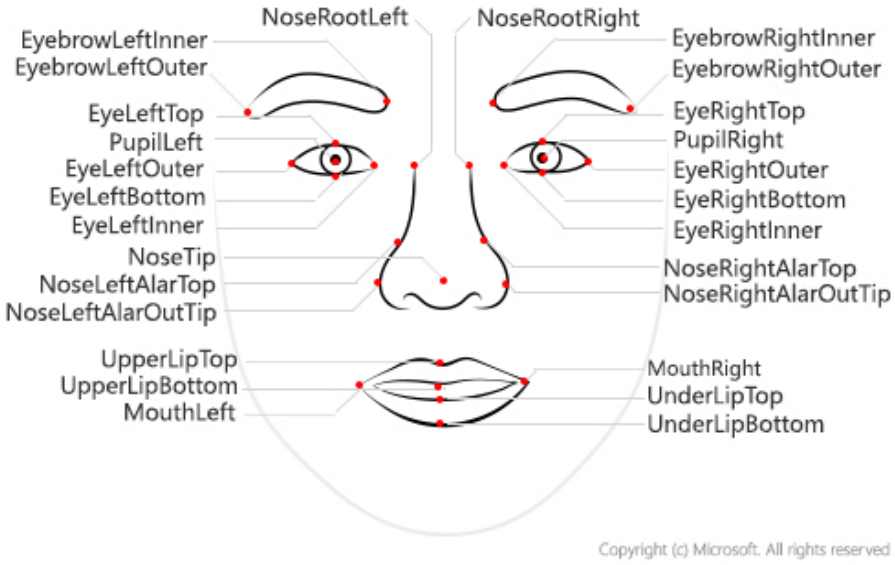

Figure 10: Oxford faces distances

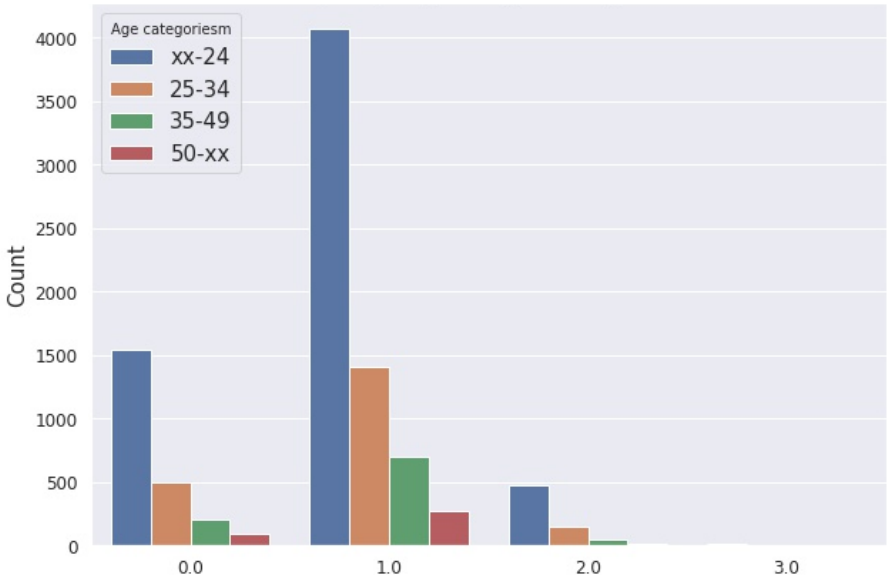

Figure 11: Number of distinct faces per age group