TCSS 555 Machine Learning
User Profiling in Social Media

Yuyang Yu
sgyyu3@uw.edu

Jiaming Hu
huj22@uw.edu

Bin Zhang
binzhang@uw.edu

Abstract

As more and more users are creating their own content on the web, there is a growing interest to mine this data for use in personalized information access services, recommender systems, tailored advertisements, and other applications that can benefit from personalization. Research in psychology has suggested that behavior and preferences of individuals can be explained to a great extent by age, gender and underlying psychological constructs (or so called personality traits). In addition to a myriad of applications in e-commerce, there is a growing interest of user profiling in digital text forensics as well [6].

The goal of this report is to build a system for automatic recognition of the age, gender, and personality of social media users. When given as input the text status updates, profile picture and "likes" of a social media user (relation), this system should return as output the age, gender and personality trait scores of that user.

1 Introduction

A variety of approaches have been recently proposed to automatically infer users' age, gender and personality from their user generated content, profiles and pictures in social media. Approaches differ in terms of the machine learning algorithms and feature sets used. In this report, our goal is when given as input the text status updates, profile picture and "likes" of a social media user (relation), this system should return as output the age, gender and personality trait scores of that user.

In order to get this goal, we create different models based on the different inputs dataset. They are (1) Using the text to predict the gender, age and big five personality, (2) Using the relation dataset to predict the gender, age and big five personality and (3) Using the pictures to predict the age and gender. Then we picked out the models which have the high accuracy to the final ensemble.

2 Methodology

2.1 Text

For the text part, we separate it into three part. They are (1) Using the text to predict the gender and age, (2) Using the LIWC table to predict the gender and age and (3) Using the text to predict the personality. In these three part, we use different methods to select the features and train the model in order to get the best result.

2.1.1 Using text to predict the gender and age

Text analysis is a major application for machine learning algorithm. However the raw data, a sequence of symbols cannot be fed directly to the algorithms themselves as most of them expect numerical feature vectors with a fixed size rather than the raw text documents with variable length. In order to solve this, we use the following two methods:

(1) Common Vectorizer usage

We use the common vectorizer to tokenization and occurrence counting in the single class. Tokenizing strings and giving an integer id for each possible token, for instance by using white-space and punctuation as token separators. Counting the occurrences of tokens in each documents [4].

(2) Tf-idf term weighting

In the text corpus, some words will be present (such as: "the", "a", "are" in English) carrying very little meaningful information about the actual contents of the document. If we were to feed the direct count data directly to a classifier those very frequent terms would shadow the frequencies of rarer yet more interesting terms. In order to reweighting the count features into floating point values suitable for usage by a classifier it is very common to use the tf-idf transform [4].

After extracting the text's features, we compare the performance of 2 learning algorithms trained on these features, namely Naive Bayes(NB) and Support Vector Machine with a rbf kernel(SVM). The following two figures (figure 13, figure 14 and figure 15)show how the two methods predict for the age.

2.1.2 Using LIWC table to predict the gender and age

The linguistic Inquiry and Word Count tool, known as LIWC, is well known text analysis software which is widely used in psychology studies. Thus the big five personality, we use the LIWC table to predict. In the table, there are about 81 features. Then we use the stepwise in R to select the best 20 features from the 81 features. At last, we compare the result of 4 training methods, they are Linear Regression, Support Vector Machine (SVR), K-Nearest-Neighbour (k=5) and Lasso Regression. The following Table 6 shows the result of these methods. According the result of the four methods, we decide to leave the Linear Regression and Lasso Regression for the later ensemble.

2.1.3 Using the text to predict the personality

Cause we have already used the LIWC table to predict the personality, then we try to use the text content to predict the personality, in order to get the best method to get the best accuracy.

We use the tf-idf term weighting to extract the features. We extract the unigram, bigram and trigram for the further training model. However, when it adds the trigram, it leads to the training time increase and the result is not very good. Thus we decide to use only the unigram and bigram.

It is adaptable for the continuous data to use the regression model. We try the several different regression models. At last, we decide to use the Huber Regression model to train the data. The result of this method is showing on the Table 1.

Table 1. Using text to predict the big five personality

	Ope	Neu	Ext	Agr	Con
Baseline	0.65	0.73	0.79	0.66	0.80
Text Model	0.63	0.80	0.82	0.67	0.72

2.2 Relation

For relation dataset, we tried several machine learning methods, such as Page-User-Page model, k Nearest Neighbors (kNN), applying singular-value-decomposition (SVD) on User-Like matrix, and Perceptron neural network. The most successful practice is the Page-User-Page and the Perceptron neural network.

(1) Page-User Page model

This approach can be divided into three steps. Firstly, based on the user's like in the training set, calculate the average gender, age, personality scores of each page into a Page-Score matrix. Then, for each user in the testing set, calculate the average gender, age, and personality scores based on previous calculated Page-Score matrix. Lastly, round up their gender and their age-group, together with the average big five scores as their final predictions.

Furthermore, upon closer examination, we found out that the original training dataset is highly unbalanced in age and gender. For example, most training examples are under 24 years old. It results in the trained classifier is more likely to classify unseen instances into the age group of xx-24. Similarly, gender is also facing the same situation. Thus, in order to improve the accuracy, we decided to set a better threshold for the classifier. Thus, we took the brute-force searching approach together with 10-fold cross validation to track down the best threshold. Finally, as shown in Fig. 1 and Fig. 2, we got the best thresholds for age and gender. According to these results, we relatively improved the accuracy by setting 25.6, 32.4, 51.5 for the three age group thresholds and 0.60 for the gender criterion as illustrated in Fig. 3.

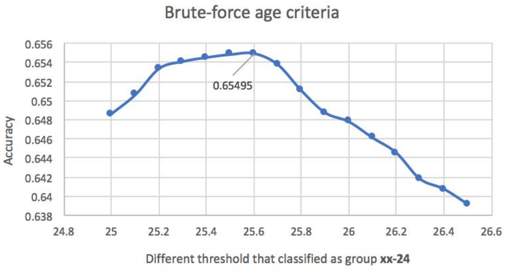


Fig. 1. Brute-force to track down the best thresholds for age.

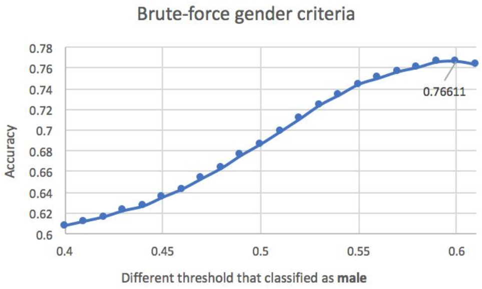


Fig. 2. Brute-force to track down the best threshold for gender.

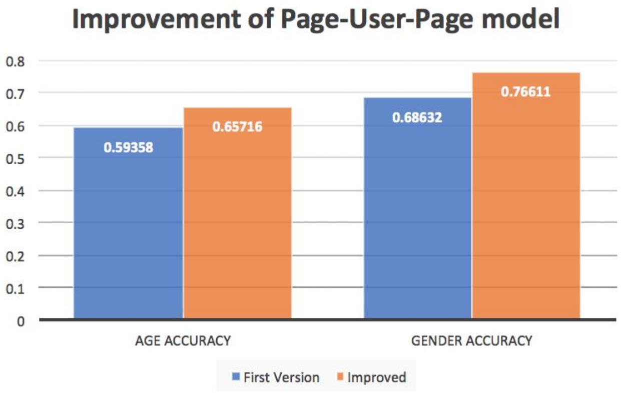


Fig. 3. The comparison chart of Page-User-Page model with original criteria and revised criteria.

(2) *k* Nearest Neighbor

Both weighted *k* Nearest Neighbor and the traditional *k* Nearest Neighbor algorithm have been applied on the relation dataset. The first thing to do is to transform the relation data into a User-Like matrix. Then, for each unseen instance, we predict it based on its *k* nearest neighbors in the training set. For this purpose, we tried not only to use unweighted *k* nearest neighbor regression, but also weight the distance according to the information gain. However, we abandoned this classifier due to the lack of satisfactory results. The biggest reason the model gets into low accuracy problems is the training data is very unbalanced. This is further illustrated in Fig. 4, which shows the liked page occurrence distribution is highly concentrated in 1 to 5. Thus, neither could we simply drop pages with small number of likes, nor could we easily apply *k*NN based on these low occurrence pages.

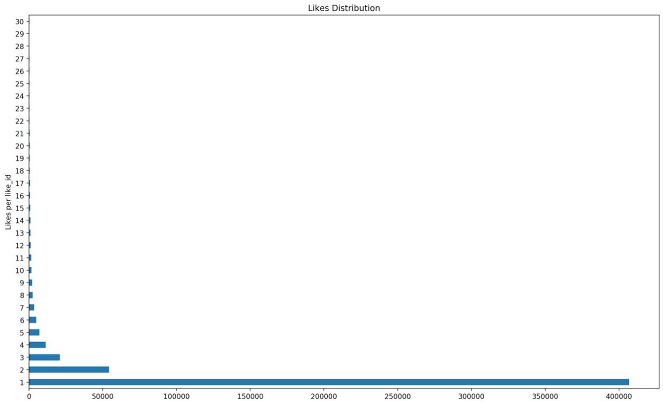


Fig. 4. The distribution of the number of likes over each identical page, i.e, like_id.

(3) Regression with SVD

This classifier is implemented by following the approach recommended by Kosinski [1]. Since the users and their likes were represented as a sparse User-Like matrix, we firstly applied singular-value-decomposition (SVD) on the matrix to reduce its dimensionality to 100 components. Then, numeric variables were predicted by linear regression, while dichotomous variables were predicted using logistic regression model.

e.g. $age = \alpha + \beta_1 Component_1 + ... + \beta_n Component_{100}$

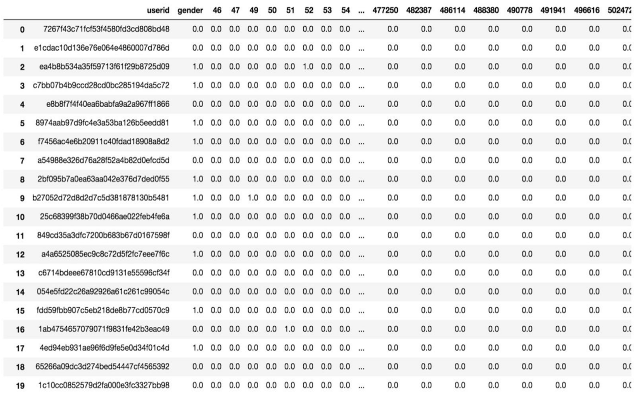
However, there were several components tends to be infinite when applying SVD in the first step. Thus, we had to preprocess the training data. This process may lose too much information, which results the final result was not as good as random guessing.

(4) Perceptron neural network

After the failure of *k*NN and SVD approaches, we has turned to reprocess the User-Like matrix in order to feed it to the perceptron neural network. In the User-Like matrix, the like_id as columns which treated as features as shown in Fig. 5. Obviously, the matrix is sparse due to the large number of like_id and its un-continuity. Hence, we remapped the userid and like_id to continuous sequence, reduced the number of matrix columns from 1671353 to 536204. Since the liked page occurrence distribution is highly concentrated in 1 to 5, we selected the pages that liked by more than 3 people. After that, our final matrix is 9500 x 53948 which has reasonable size and not loose too much details. Then we built a perceptron model use this matrix with Keras, Table 2 is the definition of network:

Table 2. Network definition of ANN

Layer	
Input	53948 like id vector
FC	512 neurons, activation=ReLU, dropout=0.5
Softmax	4 for age, 2 for gender



	userid	gender	46	47	49	50	51	52	53	54	...	477250	482387	486114	488380	490778	491941	496616	50247
0	7267f43c71fcf53f4580fd3cd808bd48	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	e1cdac10d136e76e064e4860007d786d	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	ea4b8b534a35f59713f61f29b8725d09	1.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	c7bb07b4b9ccd28cd0bc285194da5c72	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	e8b8f7f4f40ea6babfa9a2a967ff1866	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	8974aab97d9fc4e3a53ba126b5eedd81	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	f7456ac4e6b20911c40fdad18908a8d2	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	a54988e326d76a28f52a4b82d0efcd5d	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	2bf095b7a0ea63aa042e376d7ded0f55	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	b27052d72d8d2d7c5d381878130b5481	1.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	25c68399f38b70d0466ae022feb4fe6a	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	849cd35a3dfc7200b683b67d0167598f	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	a4a6525085ec9c8c72d5f2fc7eee7f6c	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
13	c6714bdeee67810cd9131e55596cf34f	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
14	054e5fd22c26a92926a61c261c99054c	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
15	fdd59fbb907c5eb218de8b77cd0570c9	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
16	1ab4754657079071f9831fe42b3eac49	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
17	4ed94eb931ae96f6d9fe5e0d34f01c4d	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
18	65266a09dc3d274bed54447cf4565392	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
19	1c10cc0852579d2fa000e3fc3327bb98	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Fig. 5. The remapped User-Like matrix, the entries of which were set to 1 if there existed an association between a user and a Like and 0 otherwise. The first column is the userid, while the second one is the target variable.

2.3 Image

It is well known that convolutional neural network is very good at image recognition. A Convolutional Neural Network works by moving small filters across the input image. The filters are re-used for recognizing patterns throughout the entire image. This makes the Convolutional Networks much more powerful than perceptron networks with the same number of neurons.

In order to avoid overfitting, we distorted the input images for training, with randomly cropped images, randomly adjusted hue, contrast, saturation and brightness and randomly flipped the image.

We experimented three different convolutional neural networks.

(1) Google Inception V3 network

The original Inception V3 network [12] was trained for visual recognition which tried to classify 1000 classes. Unfortunately the Inception v3 model is unable to classify images of people, like age and gender. However, Inception network is pretty well on extracting useful information from image, so we can retain its final softmax layer with our own dataset to get the ability of classifying gender or age.

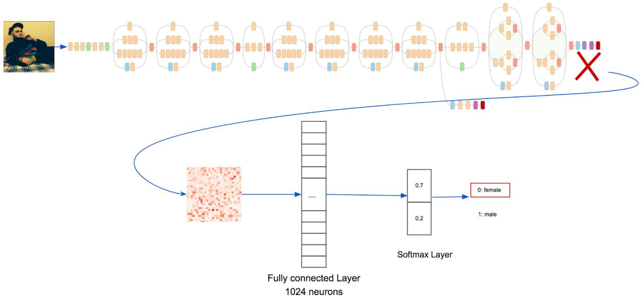


Fig. 6. The place where we retrained google's Inception v3 network. Feeded with our images, the inception network can do prediction on gender.

(2) Simple two convolutional layers network

The parameters of this network was extracted from CIFAR-10 [11]. It have two convolutional layers both with 5x5 kernel size, applied 64 filters, rectified linear units(ReLU) are used as activation functions. Then used two 2x2 max pooling layers for downsampling, connected with two fully-connected layers(256 and 128 neurons respectively).

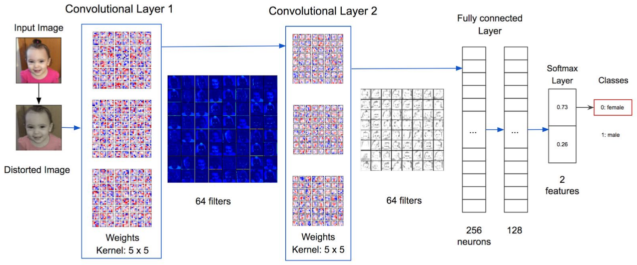


Fig. 7. A simple neural network contains two convolutional layers and two fully-connected layers

(3) Sophisticated convolutional neural network

This approach was followed the caffe model by Eran Eidinger, Roee Enbar, and Tal Hassne [10]. We implemented it by native Tensorflow. This CNN model is referred as "CNN-ETR" blow.

Table 3. The details of CNN-ERT network:

Layer	
Input	227x227 (we oversampling the image to 256 and cropped to 227)
Conv2D	7x7 filters = 96
Maxpool	3x2
Conv2D	5x5 filters = 256
Maxpool	3x2
Conv2D	3x3 filters = 384
Maxpool	3x2
FC	512 dropout=0.5
FC	512 dropout=0.5
Softmax	2 for gender, 4 for age

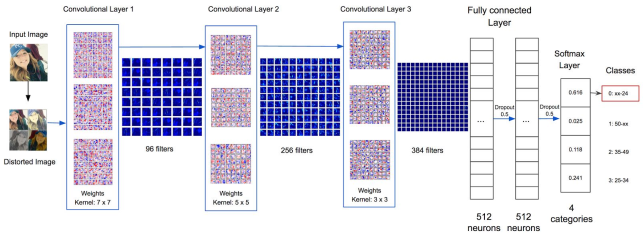


Fig. 8. A more sophisticated neural network contains three convolutional layers and dropout rate at 0.5.

2.4 Ensemble

As we described above, we have 5 classifiers for gender: SVM-Text, Page-User-Page, ANN-Relation, CNN-Inception and CNN-ERT; 4 classifiers for age: SVM-Text, Page-User-Page, ANN-Relation and CNN-ERT; 3 estimators for personality: Linear Regression-LIWC, LASSO-LIWC and Page-User-Page. So we designed a unified interface for all of this models, the Main module hold a series of this uniform objects, collect predicted values or classes from them. We used the simplest way for ensemble that for classification problem voting for majority, for regression problem we took average from all estimators. There is one thing need to be careful that we have 4 classifiers for age which have 4 categories, in some cases, there is no majority exists. We picked predicted class from the one has best accuracy under this circumstances.

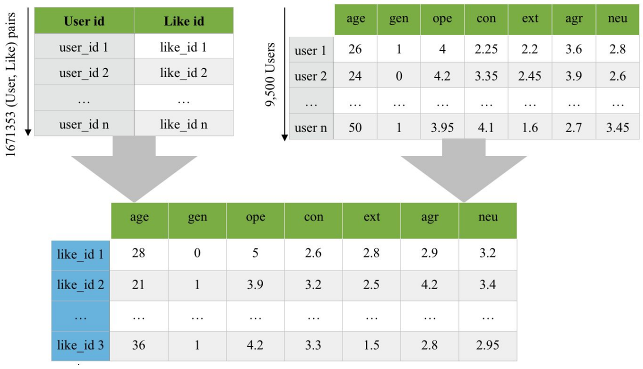
3 Dataset and metrics

3.1 Text

For the text data. It's content and results distributed in two different csv file. Before using the data, we need to merge them into one file according to the user-id. Then use this new combined file to process.

3.2 Relation

The dataset used for the relation based predictions was a huge set of association between user_id and like_id . There were two ways to handle this training data. For the Page-User-Page approach, we constructed a new page score matrix as shown in Fig. 9 which reads data from user-like relations and user profiles. It can then be used to predict unseen instances by taking the average over their favored pages' score.



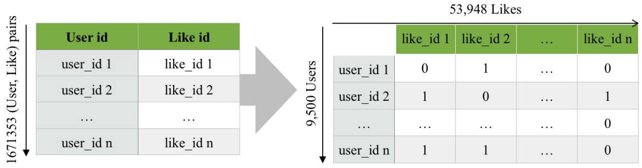
User id	Like id
user_id 1	like_id 1
user_id 2	like_id 2
…	…
user_id n	like_id n

	age	gen	ope	con	ext	agr	neu
user 1	26	1	4	2.25	2.2	3.6	2.8
user 2	24	0	4.2	3.35	2.45	3.9	2.6
…	…	…	…	…	…	…	…
user n	50	1	3.95	4.1	1.6	2.7	3.45

	age	gen	ope	con	ext	agr	neu
like_id 1	28	0	5	2.6	2.8	2.9	3.2
like_id 2	21	1	3.9	3.2	2.5	4.2	3.4
…	…	…	…	…	…	…	…
like_id 3	36	1	4.2	3.3	1.5	2.8	2.95

Fig. 9. The page-score matrix extracted from the relation dataset and user profiles.

Another way to handle the relation dataset is to transform them into a User-Like matrix as illustrated in Fig. 10. By expanding the original flatten dataset, the extracted User-Like matrix can then be processed by the *k* nearest neighbor algorithm or the artificial neural network classifier.



User id	Like id
user_id 1	like_id 1
user_id 2	like_id 2
…	…
user_id n	like_id n

	like_id 1	like_id 2	…	like_id n
user_id 1	0	1	…	0
user_id 2	1	0	…	1
…	…	…	…	0
user_id n	1	1	…	0

Fig. 10. Expand the flatten relation data into a sparse User-Like matrix, while 1 represents the existence of association between a user and a like_id and 0 otherwise.

3.3 LIWC

For the LIWC, there are 81 features in the table. In order to get the best result, we use the stepwise in R to choose the best different 20 features for the big five personality.

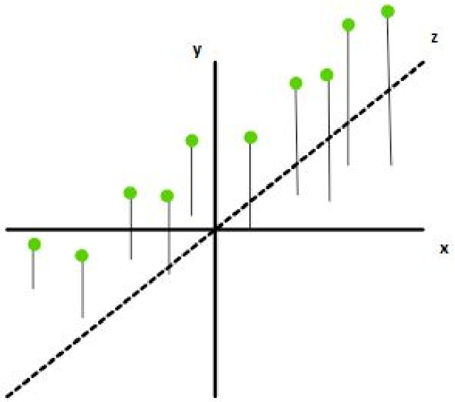


Fig. 11. The stepwise to get the best features

3.4 Image

Table 4. Number of image files for each category

Category	Number of Images
xx-24	5669
25-34	2400
35-49	1044
50-xx	385
-	-
female	5398
male	3927

The images were settled in a single folder, we regrouped them according to the labels in profile.csv and count them into Table 4. As the numbers shown in Table 4, the images belong to each category are unbalanced. For age classification, more than 50% of images reside to group xx-24 which will obviously cause CNN model more likely to misclassify people into group xx-24. It is a little better on gender classification since it's only have two classes, although the images between female and male also unbalanced.

We also noticed other critical issues:

a) some of images are obviously not human.

b) some of images have multiple people inside.

c) some of images didn't reflect the true label. A mother used her daughter's picture, for instance.

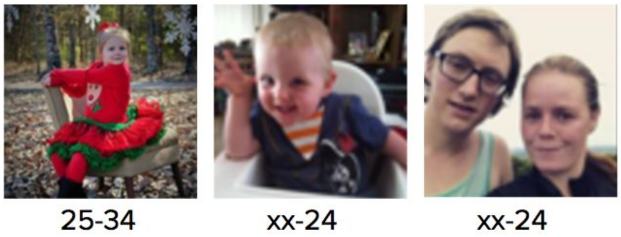


Fig. 12. Example pictures that didn't reflect the TRUE labels

3.2 metrics

In this project, we are facing two types of machine learning problems: classification and regression. For classification problem, we measure our classifiers by its accuracy while we use RMSE to measure our regression models.

Table 5. Metrics for each models

Model	Metrics	
SVM-Text	Accuracy	10-fold cross validation
Page-User-Page (Age & Gender)	Accuracy	10-fold cross validation
Page-User-Page (Personality)	RMSE	10-fold cross validation
LIWC-Linear	RMSE	10-fold cross validation
LIWC-LASSO	RMSE	10-fold cross validation
ANN-Relation	Accuracy	10-fold cross validation
CNN-Inception	Accuracy	Training, validation accuracy and the accuracy on a test-set
CNN-ETR		

4 Results

4.1 Text

(1) The result of the two models(Naive bayes and Supported Vector Machine) in using text to predict the gender and age is as following figures. Finally we decide use the SVM model to predict the gender and age, when we use the text input.

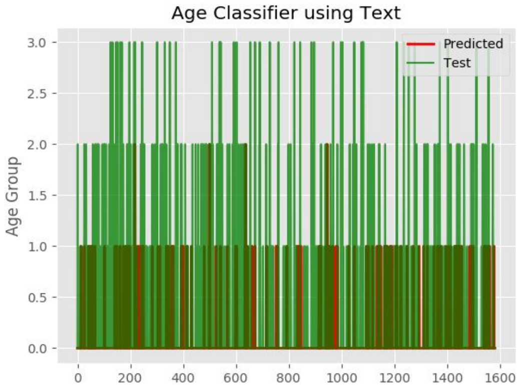


Fig. 13. Age classifier use Naive Bayes

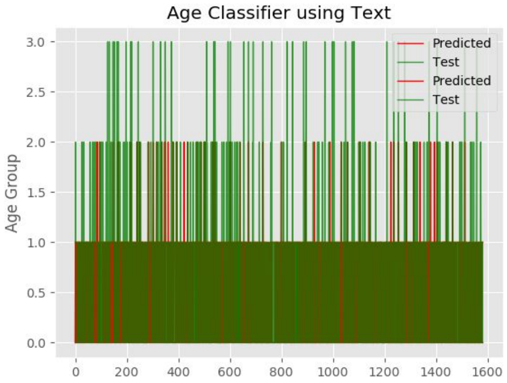


Fig. 14. Age classifier use Support Vector Machine(SVC)

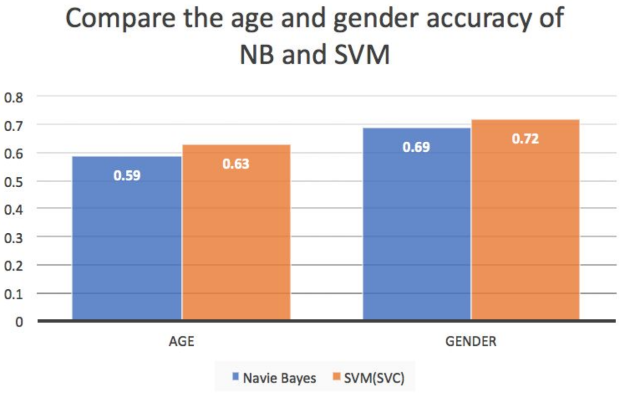


Fig. 15. Compare the age and gender accuracy of NB and SVM

(2) following table is the result of using the LIWC to predict the big five personality. According the result we decide to use the Linear Regression and the Lasso Regression for the further ensemble.

Table 6. Using LIWC predict the big five personality

	Ope	Neu	Ext	Agr	Con
Linear Regression	**0.62**	**0.78**	**0.80**	**0.65**	**0.70**
SVM(SVR)	0.63	0.81	0.82	0.67	0.73
KNN	0.63	0.79	0.80	0.66	0.71
Lasso Regression	**0.62**	**0.79**	**0.80**	**0.65**	**0.71**

4.2 Relation

(1) Page-User Page model

Fig. 16 presents the accuracy of predicting user age. The average accuracy over the 10-fold cross validation is 65.72%, which is highly improved from the baseline i.e., 59%. Hence, it is certainly a valuable classifier for the later ensemble on age predictions.

Similarly, Page-User-Page model also worked well on classifying social media user gender. Fig. 17 demonstrates its performance on gender prediction. The average accuracy is 76.61%, which is also enhanced from the baseline of 59%.

As for personality identification, the average Root Mean Squared Error (RMSE) over five personality traits as shown in Fig. 18. The average RMSE of openness, conscientiousness, extroversion, agreeableness, and neuroticism are 0.614, 0.716, 0.805, 0.665, and 0.799. All of them are relatively improved from the baseline.

The overall accuracy of this model may influenced by the users who have not liked many pages. To get rid of this influence, we can add a threshold based on the number of likes of each instance as the model confidence in the future.

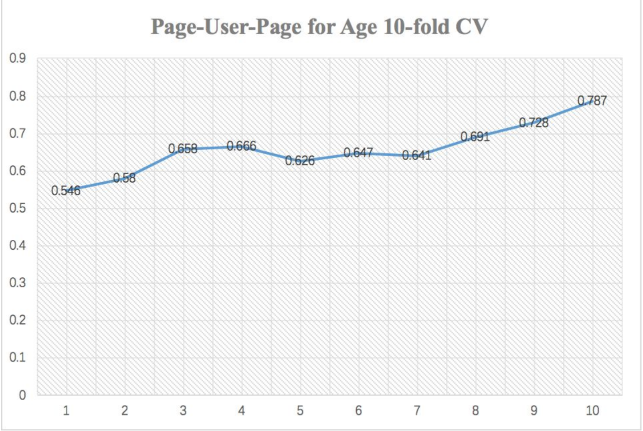


Fig. 16. Prediction accuracy of classification for user age.

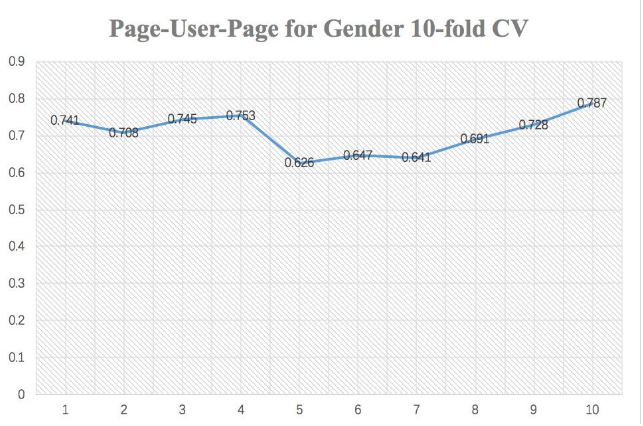


Fig. 17. Prediction accuracy of classification for user gender.

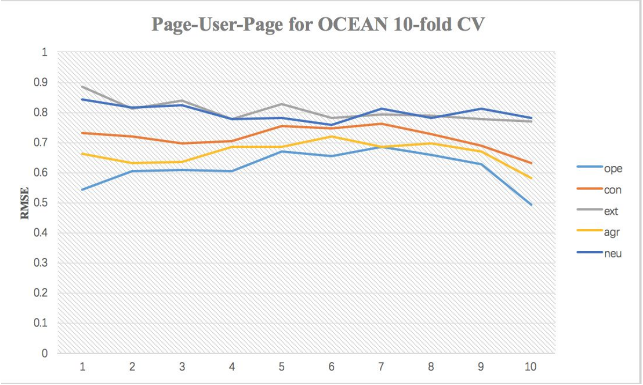


Fig. 18. Prediction accuracy of classification for personalities.

(2) Perceptron neural network

Fig. 19 and Fig. 20 show the prediction accuracy of perceptron neural network on gender and age in terms of 10-fold cross validation. The average accuracy of gender prediction is 82.56%, while that of age is 67.51%. Also, as shown in these two figures, the prediction accuracy of gender is relatively more stable than that of age. This is because there are only two classifications in gender, while there are four age groups. Meanwhile, the accuracy of this approach is also suffers from the same drawback in relation based classification, that is the accuracy may be dragged down by some users who did not like many pages. However, it could also be optimized in later ensemble phase.

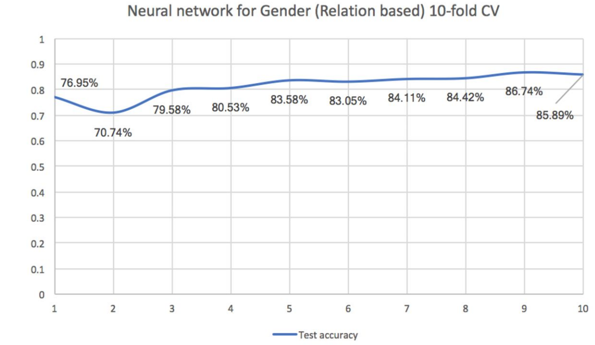


Fig. 19. 10-fold CV accuracy of perceptron neural network on gender classification.

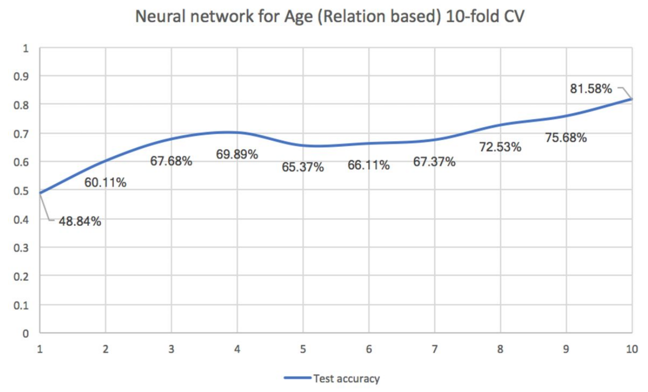


Fig. 20. 10-fold CV accuracy of perceptron neural network on age classification.

4.3 Image

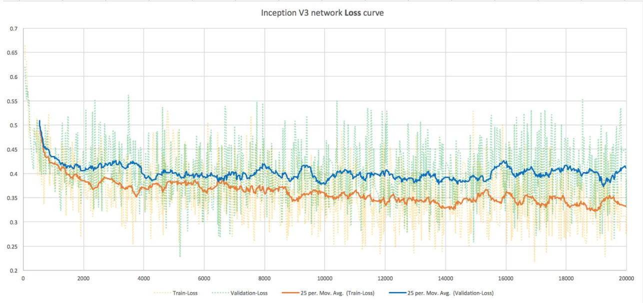


Fig. 21. The Loss curve of Inception V3 network from step 50 to step 20000. One step represent one iteration or one batch, in this case, the batch size is 100 images.

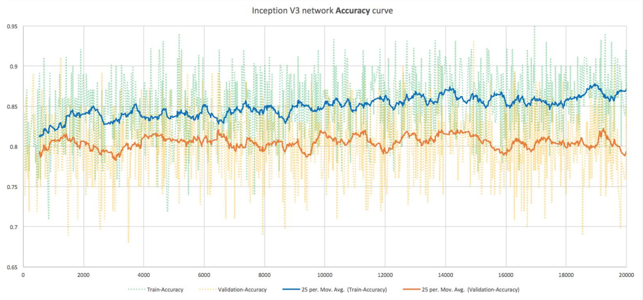


Fig. 22. The training and validation accuracy curve of Inception V3 network from step 50 to step 20000.

The performance of Google's Inception V3 network on gender classification surprised us although it is not designed for this purpose. It is because Inception V3 network is good at extracting features from image and the gender classification problem is simple enough(2 categories). Throughout the loss and accuracy curve, we noticed the best performance appear around 10,000 steps. We saved the trained model at step 11,000 which eventually got 0.82 accuracy on VM's test. Comparing to its excellent performance on gender, the Inception network has extremely poor performance on age classification. It only reached 0.36 accuracy based on our test.

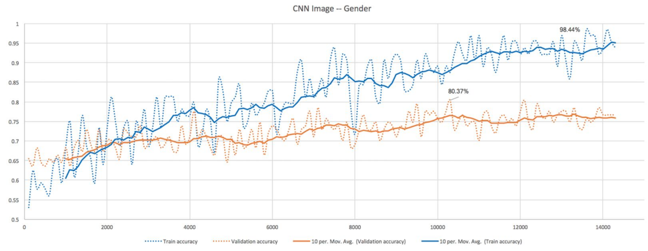


Fig. 23. Training and validation accuracy curve of CNN-ETR model on gender classification from iteration 0 to 15000.

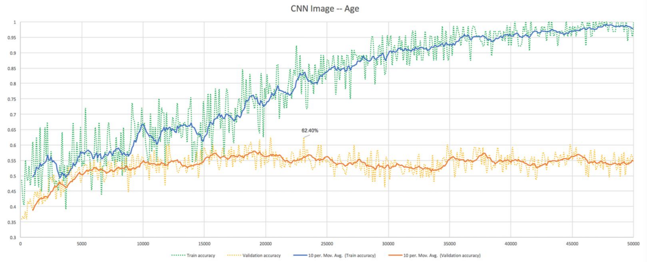


Fig. 24. Training and validation accuracy curve of CNN-ETR model on age classification from iteration 0 to 50000.

The CNN-ETR model which we trained from scratch. It's accuracy on gender classification reached 0.77 under this unbalanced image dataset while the accuracy on age was only reached 0.606 which almost the same as baseline(frequency based). We tried to train age model with much more iterations in order to observe the fluctuation of performance but the result shows us there is no improvement once we beyonded the 16,000 iterations. Hence, we save the models on 9,100 and 14,100 iterations for gender and age respectively.

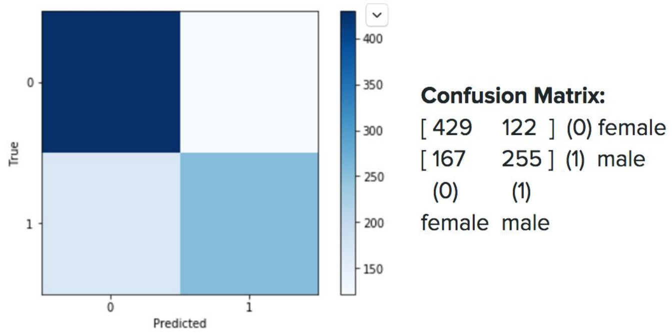
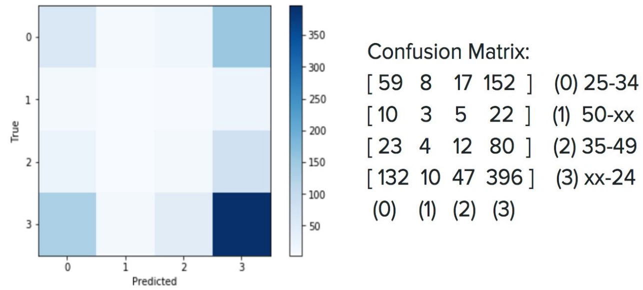


Fig. 25. The confusion matrix of CNN-ETR model when test-set contain 973 images.



Confusion Matrix:
[59 8 17 152] (0) 25-34
[10 3 5 22] (1) 50-xx
[23 4 12 80] (2) 35-49
[132 10 47 396] (3) xx-24
(0) (1) (2) (3)

Fig. 26. The confusion matrix of CNN-ETR model when test-set contain 1277 images.

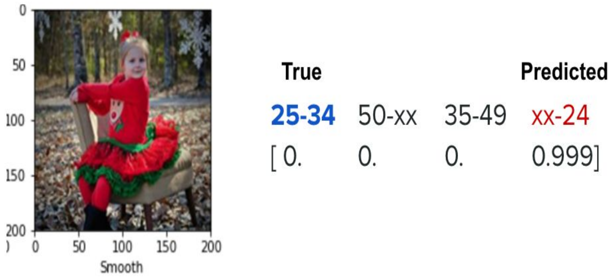


Fig. 27. A typical misclassified example, CNN-ETR did a correctly prediction but the true label on this image didn't reflect the real situation.

The result of confusion matrices in line with our expectation that the CNN model more likely to misclassify gender male, age group 50-xx and 35-49 due to the small size of training set for these categories. Also, the problem that some training data with wrong labels confused the network which we already foreseen when analyzed the dataset itself.

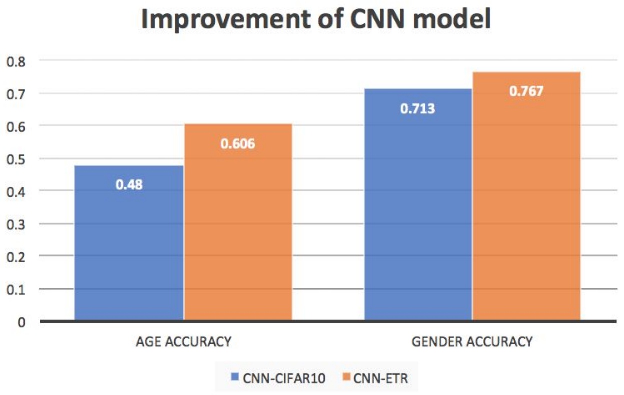


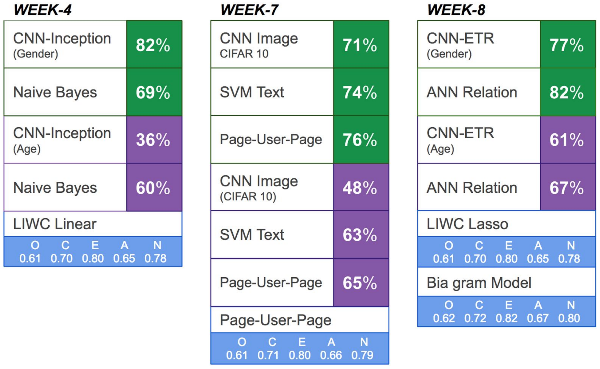
Fig. 28. Performance of CNN-CIFAR10 and CNN-ETR model.

The significantly improvement of CNN model is not only the CNN-ETR network is more sophisticated than CNN-CIFAR10. A more important fact is that our CNN-CIFAR10 network doesn't include dropout which is the key factor that contributed the improvement.

4.4 Ensemble

Table 7. Every week's result (accuracy and RMSE) of our models running on VM

	Age	Gender	Ope	Neu	Ext	Agr	Con
Baseline	0.59	0.59	0.65	0.80	0.79	0.66	0.73
Week 4	0.59	**0.82**	**0.64**	**0.79**	**0.78**	**0.65**	**0.72**
Week 7.1	**0.65**	0.68	**0.63**	0.80	0.79	0.67	0.73
Week 7.2	0.65	0.73	0.63	0.79	0.78	0.66	**0.71**
Week 8.1	**0.66**	**0.84**	0.63	0.79	0.78	0.66	0.71
Week 8.2	0.66	**0.87**	0.63	0.79	0.78	0.66	0.71
Week 10	0.63	**0.88**	0.63	0.79	0.78	0.66	0.71



WEEK-4
CNN-Inception (Gender) 82%
Naive Bayes 69%
CNN-Inception (Age) 36%
Naive Bayes 60%
LIWC Linear
O C E A N
0.61 0.70 0.80 0.65 0.78

WEEK-7
CNN Image CIFAR 10 71%
SVM Text 74%
Page-User-Page 76%
CNN Image (CIFAR 10) 48%
SVM Text 63%
Page-User-Page 65%
Page-User-Page
O C E A N
0.61 0.71 0.80 0.66 0.79

WEEK-8
CNN-ETR (Gender) 77%
ANN Relation 82%
CNN-ETR (Age) 61%
ANN Relation 67%
LIWC Lasso
O C E A N
0.61 0.70 0.80 0.65 0.78
Bia gram Model
O C E A N
0.62 0.72 0.82 0.67 0.80

Fig. 29. The timeline of adding or improving models.

In week 4, we used CNN-Inception for gender, Naive Bayes for age and LIWC-Linear Regression for Personality. In week 7, SVM-Text contributed to age and gender(week 7.2), CNN-CIFAR used for gender and Page-User-Page used for personality. From the week 8, we began to use ensemble method, the CNN-Inception, CNN-ETR and SVM-Text combined together for gender, CNN-ETR, Page-User-Page and SVM-Text voted for age. The better gender accuracy on week 8.2 came from the improvement of CNN-ETR. In week 10, we introduced ANN-Relation into the gender classifiers, and tested age accuracy only with ANN-Relation.

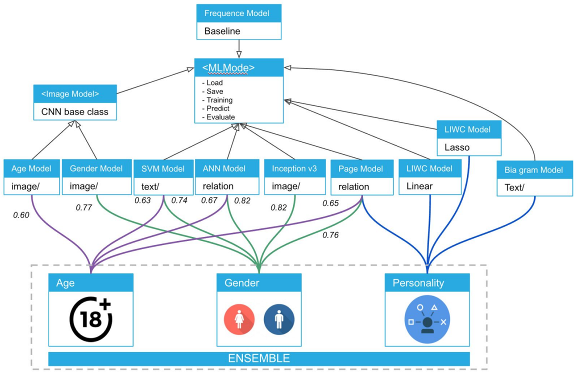


Fig. 30. The components comprise our framework, as the classifiers and estimators increasing, the framework itself starts to be similar to a neural network.

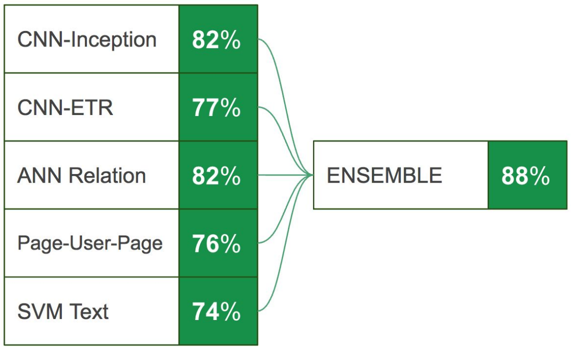


Fig. 31. The result of ensembling with 5 gender models.

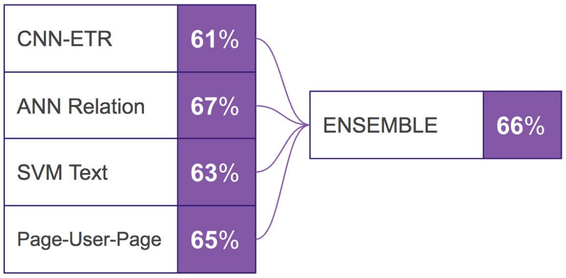


Fig. 32. The result of ensembling with 4 age models

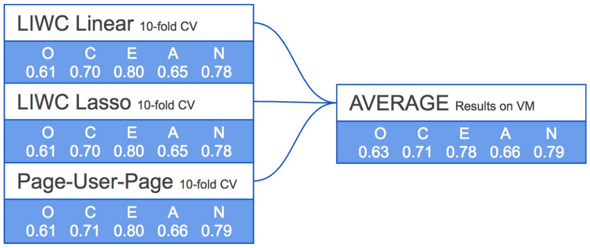


Fig. 33. The result of averaging 3 personality estimators.

The result of ensembling gender classification shows the power of ensemble method. Comparing with the best gender classifier(CNN-Inception), fortunately it had 6% improvement. However, the ensemble didn't show any improvement on age classification. It could be caused by inaccurate model evaluation, for example, the ANN-Relation performs worse on VM. Additionally, the average of three estimators for personality is even worse than LIWC-Linear model by viewing the results table of every week on VM. The performance of LIWC-Lasso is very close to LIWC-Linear, but the Page-User-Page model is the worst one among the three models for personality.

5 Conclusions and future work

Generally, we successfully built a system to automatically and accurately inferring of the age, gender, and personality of social media users. When given as input the digital records, such as status, profile pictures and Likes, this system can return as output the age, gender and personality trait scores of that user. Our work on social media user's individual attributes can be valuable to personalized information access services, recommender systems, tailored advertisements, and other applications that can benefit from personalization.

We made great progress on gender classification but the accuracy on age was stagnant. It is possible to improve our ANN-Relation model by optimizing network parameters and selecting better features. It is a little sad we didn't have time to try NLTK for text tokenization which could bring better features extracted from text than TF-IDF. It is absolutely we could make progress on Image if we introduce face detection that can solve the problem of multiple persons in one image or filter the image that is not human. Inspired by decision tree, there is a possible way that could improve the performance of CNN age model by changing 4 categories problem info several 2 categories in case of unbalanced number of images. Additionally, our ensemble code still elementary, we don't have time to implement the features such as downgrade the Page-User-Page classifier and the ANN-Relation classifier if the target user have few liked pages or ignore the Text source if there is no status update or few words for that user.

References

[1] M. Kosinski, D. Stillwell and T. Graepel. 2013. Private Traits and Attributes are Predictable from Digital Records of Human Behavior. *PNAS.* vol. 110, no. 15, 5802-5805, DOI: 10.1073/pnas.1218772110

[2] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayurci. 2002. Wireless Sensor Networks: A survey. Comm. ACM 38, 4(2002), 393-422

[3] Patricia S. Abril and RObert Plant. 2007. The patient holder's dilemma: Buy, sell, or troll? Commun. ACM 50, 1(Jan. 2007), 36-44. DOI: http://dx.doi.org/10.1145/1188913.1188915

[4]http://ogrisel.github.io/scikit-learn.org/sklearn-tutorial/modules/feature_extraction.html

[5] http://scikit-learn.org/stable/modules/svm.html

[6] TCSS 555: Machine Learning. User Profiling in Social Media

[7] Farnadi, G., Sushmita, S., Sitaraman, G., Ton, N., De Cock, M., Davalos, S.: A Multivariate Regression Approach to Personality Impression Recognition of Vloggers. In: Proceedings of the WCPR, pp. 1–6 (2014)

[8] Farnadi, G., Zoghbi, S., Moens, M., De Cock, M.: Recognising personality traits using Facebook status updates. In: Proceedings of the WCPR, pp. 14–18 (2013)

[9] Golbeck, J., Robles, C., Turner, K.: Predicting personality with social media. In: CHI'11 Extended Abstracts on Human Factors in Computing Systems, pp. 253–262. ACM (2011)

[10] Eidinger, Eran, Roee Enbar, and Tal Hassner. "Age and gender estimation of unfiltered faces." IEEE Transactions on Information Forensics and Security 9.12 (2014): 2170-2179.

[11] Convolutional Neural Networks, https://www.tensorflow.org/tutorials/deep_cnn

[12] How to Retrain Inception's Final Layer for New Categories, https://www.tensorflow.org/tutorials/image_retraining

[13] Keras: Deep Learning library for Theano and TensorFlow, https://keras.io/

[14] pandas: powerful Python data analysis toolkit, http://pandas.pydata.org/pandas-docs/stable/

[15] Documentation of scikit-learn, http://scikit-learn.org/stable/documentation.ht