Mining Facebook Data via Text, Images, and Likes*

Extended Abstract†

Wenfei Yin‡
Institute of Technology
Univ. of Washington: Tacoma
Tacoma, WA 98402
yinwf@uw.edu

Shreya Yembarwar§
Institute of Technology
Univ. of Washington: Tacoma
Tacoma, WA 98402
shreyay@uw.edu

Jonathan McFadden¶
Institute of Technology
Univ. of Washington: Tacoma
Tacoma, WA 98402
mcfaddja@uw.edu

ABSTRACT

There is a growing interest in mining social media data for predictive applications in recommender systems, tailored advertisements, personalization in various domains etc. End applications include e-commerce, digital text forensics etc. Data from Facebook profiles was mined to predict the age group, gender, and personality information of unseen social media users. The data consisted of profile pictures of the users, text (and a digest of that text) from users' posts, and 'likes' of a user. For simplicity, the ages were grouped into four ranges. Meanwhile, the personality information consisted of scores on a 5-part (score) personality test/analysis.

KEYWORDS

ACM proceedings, LaTeX, text tagging

ACM Reference format:
Wenfei Yin, Shreya Yembarwar, and Jonathan McFadden. 1997. Mining Facebook Data via Text, Images, and Likes. In *Proceedings of TCSS 555: Machine Learning Term Project, Tacoma, WA USA, June 2018 (TCSS 555: Spring 2018),* **??** pages.
https://doi.org/10.475/123_4

1 INTRODUCTION

With the explosion in data collection/production by social media sites and their associated accounts, it has become desirable to predict attributes of many users based an analysis of a small subset of users. The goal of this project was to accomplish this in a small way by predicting a user?s

- Age Group
- Gender
- Scores on a 5-part personality test

The ages were grouped using the following ranges

- 24 & under
- 25-34
- 35-49
- 50 & over

The personality scores measured five components of a user?s personality. The users had their personality traits of

- Openness (OPE)
- Extraversion (EXT)
- Conscientiousness (CON)
- Neuroticism (NEU)
- Agreeableness (AGR)

scored on a scale from 1 to 5, with a resolution to the hundredths place.

The goals were to predict the described user attributes using models which performed better than a baseline model. These models were compared using accuracy for the age and gender attributes and root mean squared error for the personality attributes. Baseline scores to beat were 0.59 for both age & gender, 0.65 for openness, 0.79 for extroversion, 0.80 for neuroticism, 0.73 for conscientiousness, and 0.66 for agreeableness. Compared to these, we obtained the following results against our own test data are given in **Table ??**.

Against unseen test data, the results we obtained are given in **Table ??**.

Table 1: RESULTS Against Seen Data

	Likes	Images	Text
Age	0.65	-	-
Gender	0.81	0.73	-
OPE	0.60	-	-
EXT	0.78	-	-
CON	0.65	-	-
NEU	0.75	-	-
AGR	0.64	-	-

2 METHODOLOGY

Since there were three different types of data, we have divided describing our methodology into three parts. Each data type/source (*text, images, & likes*) will have its own part (*section*).

*Produces the permission block, and copyright information
†The full version of the author's guide is available as `acmart.pdf` document
‡
§
¶

TCSS 555: Spring 2018, June 2018, Tacoma, WA USA

ACM ISBN 123-4567-24-567/08/06…$15.00
https://doi.org/10.475/123_4

Table 2: RESULTS Against Hidden Data

	Likes	Images	Text
Age	0.67	-	0.61
Gender	0.83	0.73	0.71
OPE	-	-	0.65
EXT	-	-	0.79
CON	-	-	0.79
NEU	-	-	0.65
AGR	-	-	0.72

2.1 Text

For the Text part, we separate it into three parts. The first one is using the text to predict the accuracy of age and gender, then we use the 82- LIWC features to predict the big five personalities. For each part, we also use different methods to build the basic model to test accuracy.

We tried four different methods in text part, Linear regression, Random forest, Naïve Bayes and Logistical regression. Finally, we find the best result from the method which uses logistical regression and Naïve regression.

The first step of the text part is data preprocessing. In the files, we find that they are separated, thus we merge the profile file and text file according to the common primary key in each take. We add each text part into profile.csv. At the meanwhile, we also transfer the age to age groups, and we use 1 and 0 to replace the female and male in order to more effective.

After data preprocessing, we use the Naïve Bayes to predict the age and gender. As is known to us who focus on machine learning. Naïve Bayes classifiers are a family of simple "probabilistic classifiers" based on applying Bayes theorem with strong independence assumptions between the features. However, we find the model only working well on gender part. The accuracy of age in Naïve Bayes model is 0.49 which is lower than baseline (0.59). Therefore, we have to change our method. After researching, we guess the Logistical Regression will work well on age part. At the beginning, we split the training data into two part, the one is training data which are 8000 rows, the other is test data which is 1500 rows. To our surprise, it does not work well. After trying different methods, like linear regression, random forest. We find the model cannot work well when we use text to predict the age, thus we try to use 82-LIWC features to build model.

The LIWC, The linguistic Inquiry and Word Count tool, is known text analysis software which is widely used in psychology studies, In the file, for each user, it has 82 features. We have to merge the LIWC and profile file. When we create the model, we delete the big five personalities which need to be predicted in the table. After testing in local machine, we find it work well in age predict. The accuracy of age group reach to 0.62.

For Five Big Personality, we have to used 82-LIWC features. Because of big data, we decide to use Linear regression to set the model. In statistics, Linear regression is a linear approach to modeling the relationship between a scalar response and one or more explanatory variables. Fortunately, Linear regression work well in the model, the results almost similar with baseline. Although we try to decrease the RMSE of five big personality, the methods we tried are almost same.

Table 3: IMAGE MODEL-1: Basic CNN Model

Layer	Filters	Neurons	Fltr. Sz	Act.	Dropout
Conv [a]	32	-	3×3	relu	-
Conv [b]	32	-	3×3	relu	-
Conv [c]	64	-	3×3	relu	-
Full-Con[d]	-	64	-	relu	0.5
Output	-	1	-	sigmoid	-

Note: Fltr. Sz = Filter Size
Note: Act. = Activation Function

[a]Convolutional
[b]Convolutional
[c]Convolutional
[d]Fully Connected

2.2 Images

Convolutional Neural Networks was used to handle the image source of this experiment. CNN is a class of deep, feed forward ANN having applications in image/video processing, NLP etc. CNNs draw inspiration the animal vision cortex system. CNNs requires minimal preprocessing, i.e. the network learns the features as opposed to them being engineered. CNNs work by extracting features and recognizing patterns from the images, by convoluting the image with filters. CNNs have proved to be a robust and low error approach for the image classification tasks. The overlap between Machine Learning and Computer Vision has seen many recent technological developments, particularly Deep Learning. CNNs with deeper/wider layers and larger training datasets tend to perform better.

In order to avoid over-fitting, data augmentation was performed on the image database. The augmentation operations included rescaling each pixel, flipping horizontally, rotating the image by variable degrees, randomly zooming into the image etc.

A series of experiments was conducted, with different CNN models, before and after performing any filtering operations through Haar cascades. Only gender prediction was implemented with images, since experiments with age prediction did not give good results.

2.2.1 Image Model-1: Basic CNN Model. Reference for this model was taken from the Keras blog[1]. The model had an input size of 224x224 pixels, with 3 convolutional layers and and one fully connected layer. The optimizer used was 'adam' and the the evaluation metric was 'accuracy'. This model is described in **Table ??**.

2.2.2 Image Model-2: RESNET50. Bottleneck features were extracted from the RESNET50 model. Final layer of the model was not included. The pre-trained weights were used to find the bottleneck values, i.e. features were extracted. Fully connected layers were

[1]**Reference:** https://blog.keras.io/building-powerful-image-classification-models-using-very-little-data.html

Table 4: IMAGE MODEL-1: Basic CNN Model

Layer	Filters	Neurons	Fltr. Sz	Act.	Dropout
Full-Con[a]	-	256	-	relu	0.5
Output	-	1	-	sigmoid	-

Note: Fltr. Sz = Filter Size
Note: Act. = Activation Function

[a]Fully Connected

Table 5: IMAGE MODEL-1: Basic CNN Model

Layer	Filters	Neurons	Fltr. Sz	Act.	Dropout
Conv [a]	32	-	3×3	relu	-
Conv [b]	32	-	3×3	relu	-
Conv [c]	64	-	3×3	relu	-
Conv [d]	64	-	3×3	relu	-
Conv [e]	64	-	3×3	relu	-
Full-Con[f]	-	256	-	relu	-
Full-Con[g]	-	128	-	relu	-
Full-Con[h]	-	32	-	relu	0.5
Output	-	1	-	sigmoid	-

Note: Fltr. Sz = Filter Size
Note: Act. = Activation Function

[a]Convolutional
[b]Convolutional
[c]Convolutional
[d]Convolutional
[e]Convolutional
[f]Fully Connected
[g]Fully Connected
[h]Fully Connected

added to run with the training data. This model is described in **Table ??**.

2.2.3 Image Model-3: A bit advanced CNN model from scratch. The model has 5 convolutional layers, 3 fully connected and 1 output layer. Optimizer used was 'adam' and the evaluation metric was 'accuracy'. This model is described in **Table ??**.

2.3 Likes

When considering the Likes data, we first create a **User/LikeID Matrix** from the training data we are given. This process is described below in the Dataset and Metrics section. Based on the training data we were given, this resulted in a matrix with 536204 columns. Due to the large number of columns in this matrix, dimensionality reduction using Singular Value Decomposition or Principal Components Analysis was considered. Unfortunately, when the singular values of this matrix were computed (*up to the 9499th singular value*), there was no characteristic "cliff" drop-off in their magnitudes which would indicate a good point for truncation.

To illustrate this, the type of "cliff" in the singular value magnitudes we were seeking can be seen in the plot of singular value magnitudes, from a different dataset given in **Figure 1**.

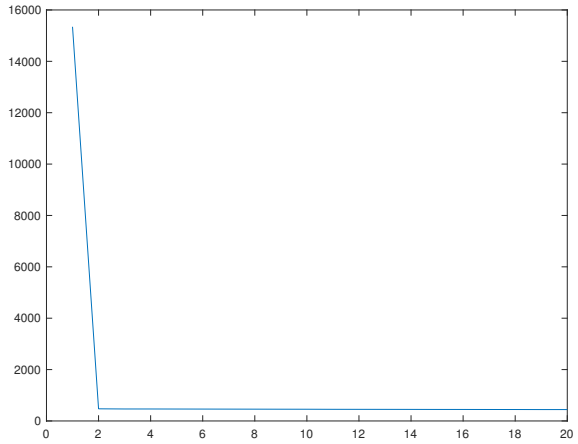

Figure 1: The type of 'cliff' drop-off in singular value magnitudes we wanted to see.

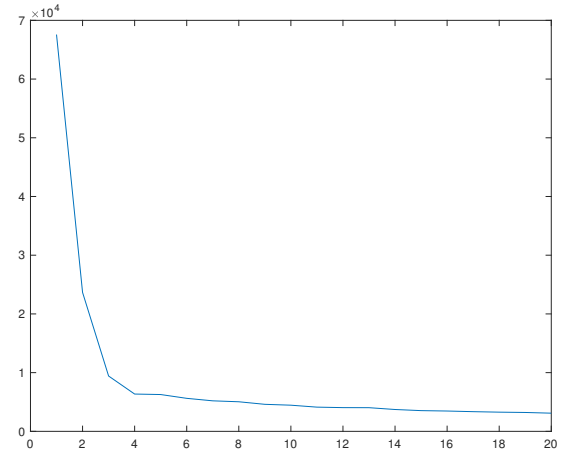

Figure 2: The singular value magnitudes we got instead of what we wanted to see.

Unfortunately, the result was got did not have the sudden drop over just a few singular values. What we got instead can be seen in **Figure 2**.

Thus, we proceeded to use the entirety of the column space of the **User/LikeID Matrix**. We constructed models using the machine learning algorithms

- Random Forest
- AdaBoost
- Bagging (*with in-bag scoring*)
- Bagging (*with out-of-bag scoring*)
- Naive Bayes (*Bernoulli*)
- K-Nearest Neighbor
- Support Vector Machine

- Linear Support Vector Machine
- Gradient Boosting
- Neural Networks (*implemented using Keras*)

Each of these algorithms was run using a variety of performance parameters, including the number of estimators, the learning rate, the tolerance, the number of neighbors, and more. Ultimately, for each of the user attributes, we chose the algorithms and settings

- **Age Group: Neural Network** with **750 base nodes** (*see below for full network design*
- **Gender: Neural Network** with **650 base nodes** (*see below for full network design*
- **OPE: SVM** with **tolerance of** 1×10^{-6}
- **NEU: AdaBoost** with **100 estimators and a learning rate of 1.0**
- **EXT: Random Forest** with **1500 estimators**
- **AGR: Bagging *using in-bag scoring*** with **250 estimators**
- **CON: Bagging *using in-bag scoring*** with **100 estimators**

We also strongly considered the alternative algorithms and settings

- **NEU: AdaBoost** with **250 estimators and a learning rate of 0.01**
- **EXT: K-Nearest Neighbors** with **500 neighbors**
- **AGR: K-Nearest Neighbors** with **800 neighbors**
- **CON: AdaBoost** with **1000 estimators and a learning rate of 1.0**

2.3.1 Neural Net Design. We designed three types of neural networks for this part of the project. The first type of neural net was a single-class binary classifier for predicting gender; while second type as a multi-class binary classifier for predicting age group. Finally, the last type of neural net we designed was a multi-layer linear regressor. All three networks had the same number of hidden and dropout layers, however they used different activation functions and kernel initializers. Additionally, all of these neural networks were implemented in **Keras** using its **Sequential** model. We summarize the three neural nets in **Table ??**.

Additionally, the following loss functions were used

- For **Age**: categorical_crossentropy
- For **Gender**: binary_crossentropy
- For **Personality Scores**: MSE (*mean-sqaure-error*)

Finally, it should be noted that all neural nets used the 'adam' optimizer

Table 6: Likes SUMMARY of Neural Nets

	Gender	Age Group	Regressor
Input Layer			
Neurons	536204	536204	536204
1st Hidden Layer			
Dense	975	1125	750
Kernel	*uniform*	*default*	*default*
Activation	*relu*	*relu*	*relu*
1st Dropout Layer	0.25	0.25	0.25
2nd Hidden Layer			
Dense	1300	1500	1000
Kernel	*uniform*	*default*	*default*
Activation	*relu*	*softmax*	*relu*
2nd Dropout Layer	0.375	0.375	0.375
3rd Hidden Layer			
Dense	975	1125	750
Kernel	*uniform*	*default*	*default*
Activation	*sigmoid*	*sigmoid*	*sigmoid*
3rd Dropout Layer	0.25	0.25	0.25
4th Hidden Layer			
Dense	650	750	500
Kernel	*uniform*	*default*	*default*
Activation	*relu*	*relu*	*sigmoid*
Output Layer			
Neurons	1	4	1
Kernel	*uniform*	*default*	*default*
Activation	*sigmoid*	*softmax*	*sigmoid*

3 DATASET AND METRICS

Since the dataset was divided into three classes of data, we describe the datasets provided for each data class in its own, separate section below. However, the metrics were similar for all classes of data. The age and gender attributes were categorical, so we used the accuracy of their prediction as our metric for models predicting those attributes. For the personality attributes, we used root-mean-squared-error as the metric for any model predicting those attributes.

Additionally, the one dataset common to all approaches was the profile data. That data had 9500 users and gave their

- Age (*as a double all ending in .0*)
- Gender (*with 0 for male and 1 for female, as a double all ending in .0*)
- Openness (*as a double on a scale from 1 to 5*)
- Extroversion (*as a double on a scale from 1 to 5*)
- Conscientiousness (*as a double on a scale from 1 to 5*)
- Neuroticism (*as a double on a scale from 1 to 5*)
- Agreeableness (*as a double on a scale from 1 to 5*)

3.1 Text

There are five filed in the training dataset. LIWC, profile, relation and test. For text and LIWC part, we use the text folder which contain 9500 text files, and we merge text files and a csv file named "profile.csv" because of common primary key "userid". Meanwhile, we merge the profile and the LIWC file as well. In the profile, we use 0 denote male, and 1 is female. In the age, we have classified

the age likes "xx-24", "25-34", "35-69", and "50-xx". In order to more effective, the big five personality, Openness, Conscientiousness, Extroversion, Agreeableness, Emotional Stability, being referred to as "ope", "con", "ext", "agr" and "neu".

3.2 Likes

The initial likes dataset contained one column with User IDs and one column with Like IDs, where the User IDs values were repeated several times to indicate all of the posts that a given user had liked. Overall, the initial input file contained over 1.5 million different 'likes'. To make the data easier for machine learning algorithms to process this initial data was converted into a sparse matrix with 9500 rows and 536204 columns, with each row representing a user and each column representing a liked item. If an arbitrary user had liked any single item, then the element in the row for that user and the column for that liked item would have a 1 for its value. Any items not liked by a user would have 0 as the value for the element in the row for that user and the column for that liked item. This is why the sparse matrix was so useful. There were only 1.67 million likes in a matrix with 5,098,538,000 elements, so most (0.03 %) of the matrix was empty (*i.e. zero valued*). Performing a dimensional reduction on this matrix using SVD or PCA would have been useful for reducing computation load and time, as well as opening up more complex machine learning algorithms; however, as discussed in the Likes subsection of the Methodology section (2.3), no good truncation point was found.

Aside from the overall metrics discussed in the opening of this section we also employed entropy loss when examining the age and gender attributes. Additionally, categorizing the personality scores by 0.1 (*yielding 40 categories*) was considered but ultimately rejected as too inaccurate. Finally, when considering the personality scores as continuous variables between 1 and 5 (*inclusive*), we used **Mean-Sqared-Error** and the R^2 **coefficient** as additional metrics to examine model performance.

3.3 Images

The image database contained 9500 images, which was split into 8000 for training and 1500 for validation. For the purposes of gender prediction, the 8000 images were regrouped according to the labels, i.e. 3386 males and 4614 females. The regrouping was done using the profile.csv file. One critical issue to note is that it is an unbalanced distribution between the two labels. Some of the issues noticed in the database were as follows:

- Images did not contain faces
- Images contained multiple faces
- Images did not contain face belonging to the label holder

The third issue mentioned above cannot be tackled, as the system lacks that intelligent ability. Operations can be implemented to filter the images with the first two issues mentioned above. OpenCV Haar Cascades were used as the filters. Haar cascades is a method used for object detection in images. There are specialized .xml files present, for detection of various objects. These files are created by a machine learning approach where a cascade function is trained from a lot of positive and negative images. The cascade functions customized for face detection were used in this project, to filter images which had multiple faces or images which did not contain faces. The cascade functions used were for different views of a face, i.e. frontal and profile views. The images which passed through at least one cascade were retained. Manual selection was performed on the few residual images, to perform a final filtering operation.

Table 7: Text RESULTS for Age & Gender Accuracy

	Age Group	Gender
Baseline	0.59	0.59
Linear Regression	0.52	0.60
Random Forest	0.51	0.60
Naive Bayes	0.49	0.71
Logistical Regression	0.61[a] 0.59[b]	0.59

[a]Using LIWC data
[b]Using text data

Table 8: Text RESULTS for Personality Score RMSE

	OPE	NEU	EXT	AGR	CON
Baseline	0.65	0.80	0.79	0.66	0.73
Linear Regression	0.65	0.79	0.79	0.65	0.72

3.3.1 Image Metrics. This project contains usage of various machine learning algorithms, which are either classification or regression. The evaluation metric of 'accuracy' was used for the classification models, and 'root mean squared error' for the regression models.

4 RESULTS

Our overall results can be summarized by the information in **Table ??** and **Table ??**, given above. Detailed results for the three different classes of data are given in their own sections below.

4.1 Text

For age prediction model, using Logistical regression based on 82-LIWC features is best choice in the method we tried. We also use other methods, line Linear regression, random forest, Naïve Bayes. We can see the results in the chart given in **Table ??**.

From the chart in **Table ??**, we can see that the best result in age prediction is the model when it uses logistical regression based on LIWC, and the best accuracy in gender predict model is Naïve Bayes. Therefore, our choice are Naïve Bayes based on text in gender and Logistical Regression based on LIWC in age prediction.

For Big five personality, we only use linear regression on the text and LIWC part, because we tried other methods, and the results are same. The results are in the chart given in **Table ??**.

From the chart in **Table ??**, we can clearly see that the results under the predict model are similar with Baseline when we use Linear regression.

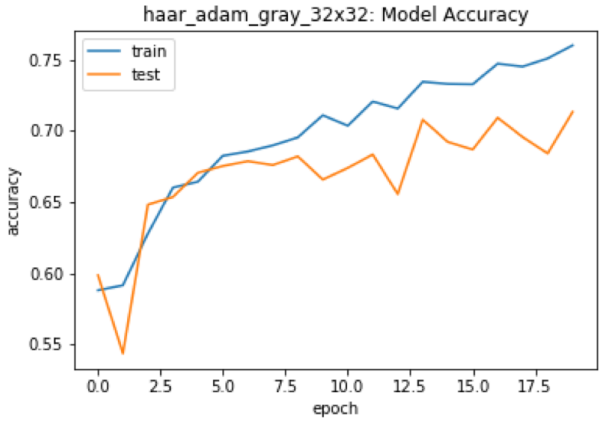

Figure 3: Accuracy curves for the final experiment with the CNN model (*created from scratch*).

Figure 4: Error/Loss curve for the final experiment with the CNN model (*created from scratch*).

4.2 Images

The basic CNN model improves the gender prediction from 0.59 to 0.66. The CNN model is trained from scratch. Further improvement was shown by cleaning the data and augmenting it, where images were filtered with Haar cascades, and augmentation was performed by internal libraries of Keras. Work with RESNET50 was attempted, however the status of the experiment is in progress.

Using Haar cascades improves the performance, as it removes a lot of bias from the dataset. The accuracy curves of a few of the experiments are given in **Figure 3** and **Figure 4**. Additionally, the results of several models predictions for gender are given in **Table ??**.

4.3 Likes

When considering only the data visible to us, the performance of various machine learning algorithms and models on the Likes data is summarized in **Table ??** and **Table ??**.

Table 9: Text RESULTS for Personality Score RMSE

Model	**Accuracy**
Baseline	0.59
Basic CNN Model	0.66
Basic CNN Model with Data Cleaning	0.73

Table 10: Text RESULTS for Personality Score RMSE

	Sex	Age	OPE	NEU	EXT	AGR	CON
Baseline	0.59	0.56	0.65	0.80	0.79	0.66	0.73
N. Bayes	0.66	0.65	0.78	0.82	0.81	0.70	0.75
Rand. For.	0.75	0.61	0.62	0.81	0.77	0.65	0.70
AdaBoost	0.61	0.73	0.70	0.75	0.80	0.70	0.74
Bag IN	0.73	0.62	0.65	0.80	0.77	0.64	0.65

Table 11: Text RESULTS for Personality Score RMSE

	Sex	Age	OPE	NEU	EXT	AGR	CON
Baseline	0.59	0.56	0.65	0.80	0.79	0.66	0.73
Bag OUT	-	-	0.60	0.79	0.77	0.67	0.68
SVM	0.57	0.67	0.60	0.81	0.80	0.66	0.74
Lin. SVM	0.77	0.66	-	-	-	-	-
Neural Net	0.80	0.66	> 1	> 1	> 1	> 1	> 1

5 CONCLUSION

We drew three different conclusions, one from each class of data.

5.1 Text

For text part, we have tried many methods when we set up the model, but we do not have do more specific work on data preprocessing, such as feature selection, word choice. When we predict the age and gender, we can find more significant words, like she, he which will affect our results. Although we get a good result in the age prediction when we use Logistical regression based on 82 LIWC features, it is not perfect. In our last assignment, we find the neural network may be good at decrease RMSE in big five personality, thus we can try it in the future work.

5.2 Images

Further work on Images includes experimenting with a few more pre-trained models, like Inception, VGG19 etc. Experiments towards age prediction with an extensive data cleaning are in the pipeline.

5.3 Likes

For binary, single-class categorical data (*Gender*), the chosen model performed above baseline and ultimately exceeded the baseline score considerably. Similarly, the chosen model for binary, multi-class categorical data (*Age Groups*), preformed above baseline; however, it did not out perform the baseline to the same extent as the model for the binary, single-class categorical data.

When considering the continuous data, model performance was disparate. Some model performed above baseline (*although not to the extent of either the gender or age models*), while others were at or barely above baseline. Given this disparate performance, it is possible that the loss of accuracy from categorizing the personality attributes using a conversion to a categorical system using increments of 0.1 from 1 to 5 may not produce less accurate results than treating the variables as continuous.

For models of all attributes, recasting the models with a dimensional reduction my produce better results. Examining the frequency counts of liked items may yield insight into a good truncation point. After dimensional reduction, the resulting data should be run in models both before and after it has been scaled to some standard interval. Furthermore, for models of the personality attributes, categorizing using increments of 0.1 or 0.05 from 1 to 5 should be considered, both with and without dimensional reduction. We believe that the key to improving the quality of results from models relying on likes data lies in some sort of dimensional reduction.