RESEARCH

Introduction to the profile areas of data sciences: project 9

Eva Aßmann* and Paul Vogler

*Correspondence:
eva.assmann@fu-berlin.de
Institute of Mathematics and Informatics, Berlin, Takustr. 9, Berlin, Germany
Full list of author information is available at the end of the article

Abstract

Goal of the project: The goal was to extract topics or personality groups from user data depicting their Facebook likes.

Main results of the project: The analysis showed that age, gender, but also political orientation and openness defined the classification. Five groups could be identified from the dataset, with two of them being mostly similar.

Personal key learnings: We learned how to efficiently analize large userbound sparse datasets and how to get personality or topic clusters out of them.

Estimated working hours: 10

Project evaluation: 1

Number of words: 2306

Suggestions for improvements of the project:

1 Data

For reproducing the Kosinski study, the original data was used [1]. The data set comprised three tables storing psychodemographic information on 110.728 Facebook users from the U.S. and their likes for specific Facebook objects.
users.csv stored the user profiles which comprised an anonymized user ID, gender ('0'=male, '1'=female), age (from 18 to 80), political view ('0'=Democrat, '1'=Republican) and five scores representing the user's personality based on the five-factor model of personality (i.e. Openness, Conscientiousness, Extroversion, Agreeableness and Neuroticism). The five scores overall ranged from -5.49 to 2.85.
likes.csv stored the anonymized IDs and the string-formatted content for 1,580,284 Facebook objects in two columns.
users-likes.csv stored associations between users and Facebook objects. For every Facebook object that a user liked, there was a row with the user's ID in one column and the respective like ID in the second column. Every user in the data set liked at least one object. Every object can only be liked once by a user.

2 Sample Big Data Set

The three tables described above were loaded into three pandas Data Frames. For some users in *users.csv*-DataFrame, information on their political views were missing. The missing values were not imputed, because of the correlation analysis with users metadata conducted in section 6. Kosinski et al. performed correlation analysis in R using only pairwise complete objects, meaning missing entries in the metadata were ignored. Thus, missing values were allowed to stay in the data set. Missing names of some liked Facebook objects were imputed with 'no data'.

3 Constructing a User-Like Matrix

The digital footprints of the users were represented by constructing a User-Like Matrix. The rows represented the users, columns represented the Facebook objects, each cell stored the information whether there is an association (i.e. like) between a user and a Facebook object. In case a user liked a specific object, the User-Like Matrix stored the value 1, else it stored 0. Because in most cases when these kind of matrices are used, each row item is associated with only a small fraction of all variables and most cells store zero entries, a sparse coordinate format matrix is used as data structure. The coordinate format sparse matrix from the scipy.sparse library was used, storing only non-zero entries. The User-Like-Matrix was created as follows:

First, for every user and Facebook object in the *users-likes.csv*-DataFrame, the corresponding indices of the user ID and like ID were matched in the *users.csv*-DataFrame and *likes.csv*-DataFrame, respectively. After mapping the indices, the user-like associations were stored in a sparse matrix M, such that the ith row of M represented the likes of the user in the ith row of the *users.csv*-DataFrame and the jth column of M represented the users that liked the object in the jth row of the *likes.csv*-DataFrame. The descriptive statistics of the resulting *Raw Matrix* are shown in Table 1.

Descriptive Statistic	Raw Matrix	Trimmed Matrix
# of users	110.728	19742
# of unique likes	1.580.284	8523
# of user-like pairs	10.612.326	3817840
Matrix density	0.006%	2.27%
Likes per User	-	-
Mean	96	193
Median	22	106
Minimum	1	50
Maximum	7973	2487
Users per Like	-	-
Mean	7	448
Median	1	290
Minimum	1	150
Maximum	19.998	8445

Table 1: Descriptive Statistics of the raw and trimmed User-Like Matrix M. Matrix density is defined as the number of non-zero entries divided by the matrix size

It took us quite some time to explore different ways of matching indices between *users-likes.csv*-DataFrame and *users.csv*-DataFrame until we came to the idea to use the pandas map() function. The first tries were computationally very inefficient and either did not even finish within 2 hours or caused a runtime error. This is why in some of the following analysis sections we had to leave out some steps due to time limitations.

4 Trimming the User-Like Matrix

The descriptive statistics of the *Raw Matrix* showed that in comparison with the size of the data set (i.e. number of users and likes), on average only a small number of users have an association with a Facebook object and vice versa (Tab. 1). In order to reduce memory requirements and computation time and retain enough information

for following analyses, the *Raw Matrix* was trimmed by removing the least frequent data points. First, the *Raw Matrix* was converted from coordinate format into compressed row storage format, in order to enable the trimming operations. Then, rows and columns that had fewer non-zero entries than the respective thresholds were removed. The row threshold C_r=50 and column threshold C_c=150 from the Kosinski study were applied. Since removing users could push the number of non-zero entries for some like-columns below the C_c threshold, this procedure was run repeatedly until the dimensions of the *Trimmed Matrix* did not change any more. The descriptive statistics for the resulting Trimmed Matrix are shown in Table 1.

5 Reducing the Dimensionality of the User-Like Matrix Using SVD and LDA

5.1 Background and goal

The goal of this step was to group Facebook users based on their likes and at the same time pool the Facebook objects into topics based on the likes they got from the users. Latent Dirichlet Analysis (LDA) was applied to the Trimmed Matrix, in order to get a grouping based on the User-Like-Associations.

5.2 Method

The basic assumption of LDA states that each user and each Facebook object belong, with a specific probability, to a set of k clusters (or topics). The LatentDirichletAllocation function of the scikit-learn library was used. As input, the function requires the number of topics k, the concentration parameter for the prior distribution of topics for a user α and the concentration parameter for the prior distribution of objects within a topic δ. Following the recommendations of Kosinski et al., α was set to 10 and δ was set to 0.1 in order to generate a distinct clustering that was easy to interpret and still contain as much information of the original data as possible. Additionally, the random_state parameter of LatentDirichletAllocation was set to the seed=68 used by Kosinski et al..

Before training the actual LDA model, a commonly used approach to select the optimal number of topics k was used. Six different values for k, in the range of 2 to 7, were used to train six different LDA models, leaving the remaining parameters unchanged. The resulting estimated log-likelihoods were plotted against the k values. The k value at the end of a rapid growth of the log-likelihood usually offers good interpretability, while larger k values usually offer better predictive power. The log-likelihood can be used to compare the fit of different coefficients for a model. The higher the log-likelihood, the better the employed coefficients. Looking at the plot, k=4 marks the end of a rapid growth (Tab. 1). Thus, Kosinski et al.'s choice of k=5 represents a good compromise between interpretability and predictive power. With all parameters defined, LatentDirichletAllocation was performed on the Trimmed Matrix.

5.3 Results

The resulting matrix of size 19742x5 described the probabilities of each of the users from the Trimmed Matrix to belong to each of the clusters. Cluster probabilities for the first ten users are shown in Fig. 2.

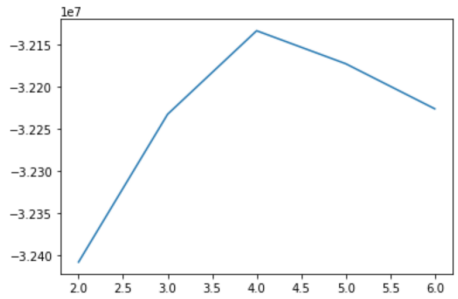


Figure 1: Finding the optimal k by plotting log-likelihoods of trained LDA models against the used values for k.

```
[[0.117135  , 0.11760453, 0.50607297, 0.15867398, 0.10051352],
 [0.13104847, 0.1208406 , 0.34029965, 0.3090692 , 0.09874208],
 [0.21163727, 0.12836589, 0.26093792, 0.32244258, 0.07661634],
 [0.19460897, 0.08475805, 0.40351513, 0.25884564, 0.05827221],
 [0.11961076, 0.1169092 , 0.24832052, 0.41527467, 0.09988485],
 [0.13744016, 0.12221499, 0.11311782, 0.5351601 , 0.09206693],
 [0.10307004, 0.10382989, 0.11956902, 0.58537771, 0.08815335],
 [0.08346499, 0.07457899, 0.07811625, 0.69746346, 0.06637631],
 [0.14519788, 0.11332814, 0.3705406 , 0.29698684, 0.07394653],
 [0.12638206, 0.12820065, 0.30076538, 0.35993769, 0.08471422]]
```

Figure 2: Topic probabilities for the first ten users.

5.4 Discussion

In order to assign the clusters to interpretable topic groups, the relationship between cluster probabilities and psychodemographic information for the users had to be analyzed in the following section. If there was more time, it would have been interesting to run LDA analysis with different prior distribution parameters to see how the basic assumptions about the topics and objects influences the clustering of users.

LDA is an easy to interpret dimensionality reduction and clustering technique that extracts patterns from language data. Applying LDA reduces the ratio between users and objects, which is required by most statistical analyses. It removes multicollinearity and redundancy in the data by grouping correlated objects into the same dimension or cluster. This decreases computational runtime and the risk of overfitting, improving the statistical power of analyses and predictive models. Additionally, data is easier to interpret when represented by less dimensions.

In addition to LDA, Kosinski et al. also applied Singular Value Decomposition (SVD) for dimensionality reduction and pattern extraction from the Trimmed User-Liked Matrix. SVD decomposes a matrix into the product of three smaller matrices that are of size k or have k many columns, where k is the selected number of dimensions, the input matrix should be reduced to. Although, SVD decomposes the Trimmed Matrix into k dimensions and LDA clusters the data in the Trimmed Matrix into k cluster, the outputs of both methods describe the same: the probabilty for each user to belong to each of the k extracted topics.

Due to the aforementioned time limitation, we only focused on LDA analysis within this project.

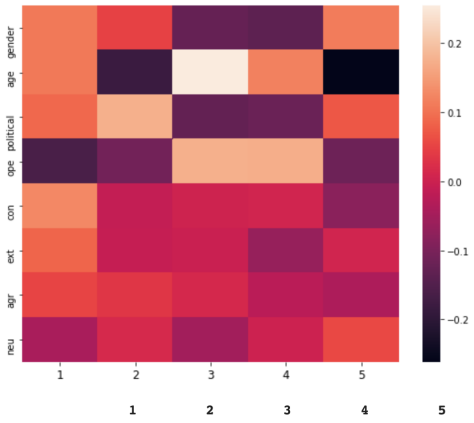
6 Interpreting Clusters and Dimensions

6.1 Background and goal

In order to understand, what topic each of the resulting clusters was describing and to gain insights into what psychodemographic properties were most responsable for the grouping of a user into a specific topic cluster, the LDA output was further explored.

6.2 Method

Correlation analysis was performed for the LDA output User-cluster-probabilities Matrix and the psychodemographic user information from the *user.csv*-DataFrame which was updated in the dimensionality-reduction-step (i.e. gender, age, political view, and five scores user personality profile). The correlation function corr() from the pandas library was employed using the default pearson correlation method. Missing values in the political views of a user were ignored in the analysis. The resulting pairwise correlation matrix was visualized in a heat map (Fig. 3).



	1	2	3	4	5
gender	0.107990	0.047306	-0.125888	-0.135848	0.113160
age	0.109071	-0.182050	0.254510	0.117541	-0.261499
political	0.090987	0.174307	-0.126981	-0.119108	0.072591
ope	-0.162607	-0.109280	0.174087	0.172957	-0.115234
con	0.125943	-0.010253	0.002775	0.008250	-0.081482
ext	0.086093	-0.009360	-0.002054	-0.067141	0.007532
agr	0.051300	0.034413	0.014024	-0.023305	-0.039312
neu	-0.043497	0.016090	-0.051138	0.001585	0.055119

Figure 3: Correlation between the k=5 LDA clusters (LDA1 to LDA5 and the users psychodemographic metadata

6.3 Results

The resulting correlation matrix showed that gender, age, political views and the personality trait of openness are most strongly correlated with the clusters (Fig. 3). Cluster LDA1 correlated negatively with openness and positively with conscientiousness, gender (0=male, 1=female), and age. Cluster LDA2 negatively correlated

with age and openness and positively with political view (0=Democrat, 1=Republican). This suggests that LDA2 contained Cluster LDA3 was negatively correlated with political view and gender, while beeing positively correlated with age and openness. Cluster LDA4 seemed to be similar to LDA3, but showed weaker positive correlation with age. LDA5 correlated negatively with age and openness, and positively with gender.

6.4 Discussion

The correlation with the users metadata for the individual clusters differed slightly from the observations that were described by Kosinski et al. what probably originated from the LatentDirichletAllocation() function using online variational Bayes algorithm, while Kosinski et al. used the R LDA() function with Gibbs Sampling method. While according to the results of Kosinski et al. LDA1 contained young, conservative female users, our correlation result for LDA1 suggested that the cluster contained the group of older, conscientious and less open-minded female users. Also, our results showed for LDA2 to contain the group of young, less open-minded Republicans and not young, conservative females. The correlation analysis for LDA3 resulted in the same user group for us and Kosinski et al.: older, open-minded, liberal males. While for Kosinski et al., LDA4 showed weak correlation to all personality traits, our correlation analysis suggested that LDA4 represented also older, open-minded, liberal males. Our results did not show strikingly strong correlation with all personality traits for LDA5, as did the results from the Kosinski study, but represented the younger, less open-minded female users.
Corelation analysis...

7 Predicting Real-Life Outcomes with Facebook Likes

7.1 Background and goal

The goal of this section was to build prediction models based on the SVD dimensions and LDA clusters extracted from the User-Like-Matrix that could assign new users to a topic group. Due to the aforementioned time limitations, we did not manage to train and evaluate such a prediction model.

7.2 Method

7.3 Results

7.4 Discussion

8 Summary - from a social science perspective

Both the LDA and the SVD algorithm allow to assign clustering groups to user-data. Even though they both use different underlying algorithms, the results from both of them can be seen as assigning groups that match both input sets, in this case facebook-users and their likes. The user groups can be used to target specific audiences through social media, for example for advertisements or for a recommendation system.
From our results the cluster defining features were gender, age, political view and openness. The first two can often easily be taken from social media profiles of the users, because these require the date of birth for a minimal age check and the gender is also mostly provided in the profile or can be seen in profile pictures. But more

complex features like the political view and openness in this example can only be estimated, similarly to the analysis presented in this report, using the user behaviour. From there the found groups are not always easily interpretable and might often times not accurately represent the individual, but they provide a great overview over connections between demographics, political views, media trends, target groups and so on.

Additionally, because the like-user-data is automatically tracked based on the user behaviour on the website, it should be less biased than, for example, questionnaires, where the questions can be answered wrong intentionally. This is especially true for more personal information that an individual would not openly share. Also a lot more users can be analyzed through their like behaviour, than by more traditional means, because most users probably wouldn't bother to answer questionnaires at all. SVD and LDA are algorithms that allow these huge datasets to be analyzed efficiently, while still making the answers interpretable.

The major concern that remains with datasets that provide user information is, to what extend the data should be analyzed and what should remain private information. Most automatically collected user information is probably accessed without the users consent, or at least without the user noticing it. So it should be at least considered if the current information one is working with is appropriate or ethical to gather and analyze.

References

[1] Kosinski et al., Mining big data to extract patterns and predict real-life outcomes, Psychol Methods. 2016 Dec; 21(4): 493–506. doi: 10.1037/met0000105