Big Data, Small Personas: How Algorithms Shape the Demographic Representation of Data-Driven User Segments

Joni Salminen,1,* Kamal Chhirang,2 Soon-Gyo Jung,3 Saravanan Thirumuruganathan,3 Kathleen W. Guan,4 and Bernard J. Jansen3

Abstract
Derived from the notion of algorithmic bias, it is possible that creating user segments such as personas from data results in over- or under-representing certain segments (FAIRNESS), does not properly represent the diversity of the user populations (DIVERSITY), or produces inconsistent results when hyperparameters are changed (CONSISTENCY). Collecting user data on 363M video views from a global news and media organization, we compare personas created from this data using different algorithms. Results indicate that the algorithms fall into two groups: those that generate personas with low diversity–high fairness and those that generate personas with high diversity–low fairness. The algorithms that rank high on diversity tend to rank low on fairness (Spearman's correlation: −0.83). The algorithm that best balances diversity, fairness, and consistency is Spectral Embedding. The results imply that the choice of algorithm is a crucial step in data-driven user segmentation, because the algorithm fundamentally impacts the demographic attributes of the generated personas and thus influences how decision makers view the user population. The results have implications for algorithmic bias in user segmentation and creating user segments that not only consider commercial segmentation criteria but also consider criteria derived from ethical discussions in the computing community.

Keywords: algorithms; fairness; personas; user segmentation

Introduction
Conceptual underpinnings
Personas, introduced to computer science and human–computer interaction (HCI) by Cooper,1 are defined as fictitious people who represent real user and customer types. Researchers believe that personas evoke a sense of empathy2,3 that directs product and user experience designers, software developers, marketers, and other stakeholders to make more user-centric decisions regarding products, services, and other outputs offered to end-users and customers. Personas are, therefore, a personified form of user segmentation, that is, dividing the overall user or customer population into demographically or behaviorally defined segments.4

Algorithmic personas (APs) are developed from quantitative data to represent demographic and behavioral characteristics of the user base.5 Consequently, AP generation is the employment of algorithms and big data to create personas.6 The promise of algorithms for persona generation was observed first by Aoyama in two articles from 20057 and 2007.8 McGinn and Kotamraju applied an algorithmic approach for personas in their seminal article in 2008.6 Since then, AP generation has become increasingly common in HCI, marketing,9 health informatics,10,11 cybersecurity,12 video game studies,13,14 and many other domains that use personified user segmentation for understanding users or customers.

1School of Marketing and Communication, University of Vaasa, Finland.
2Fulda University of Applied Sciences, Fulda, Germany.
3Qatar Computing Research Institute, Hamad Bin Khalifa University, Doha, Qatar.
4Division of Psychology and Language Sciences, Faculty of Brain Sciences, University College London, London, United Kingdom.

*Address correspondence to: Joni Salminen, School of Marketing and Communication, Wolffintie 34, 65200 Vaasa, Finland, E-mail: jonisalm@uwasa.fi
The shift from manually created personas to APs has been characterized as transformative,\(^1\) as AP generation can address the challenge of analyzing large collections of personified big data\(^6\) from social media and online analytics platforms, making sense of this big user data. Manual methods are ill-equipped to analyze such amounts of user data for user segmentation and persona generation. Rapid changes in online user behavior exacerbate the challenge, as these changes require APs to be constantly updated to keep up with the user behaviors and characteristics.\(^7\) Thus, scalable and efficient user segmentation algorithms are beneficial for transforming this data into persona profiles (Fig. 1) or other forms of user segments that describe the user populations’ key behaviors and demographics.

Since the introduction of data-driven personas,\(^6\) researchers have applied a wide range of algorithms for persona generation,\(^1,3,18-23\) the most common being clustering (grouping users in a way that users in the same group are more similar to each other than to users in other groups), principal component analysis (PCA) (summarizing the user information into smaller summary indices that aim at capturing most user information in a computationally efficient format), non-negative matrix factorization (NMF) (similar to PCA, NMF reduces user data from a higher dimensionality to a lower dimensionality that captures essential behavioral and/or demographic information), and latent semantic analysis (LSA) (identifying associations between a set of users and the content they are interested in by producing a set of latent concepts related to the users and content).\(^2\)

In their review of literature, Zhu et al\(^23\) report the use of decision trees, exploratory factor analysis, hierarchical clustering, k-means clustering, LSA, multidimensional scaling analysis, and weighted graphs for persona development. Minichiello et al\(^21\) provide a similar list of methods: cluster analysis (CA), factor analysis, PCA, and LSA. A shared trait of these methods is the attempt to simplify the user population

---

**FIG. 1.** Creating a persona from data using DR. The left-side figure shows DR using the UMAP algorithm on the dataset collected for this study. The right-side figure shows Siim, an algorithmic persona that corresponds to the pattern circled in the left-side figure. Different algorithms may identify different patterns in the data, but previous research has not examined how this affects the composition of the generated personas. DR, dimensionality reduction; UMAP, Uniform Manifold Approximation and Projection.
into segments that are then transformed (“enriched”), one way or another, into finalized persona profiles for end-users (see Figs. 1 and 2).

Research problem
This collaborative process of persona generation between humans and algorithms involves multiple challenges. Although the algorithmic process is opaque to humans, they need to trust that the algorithm performs in a desirable way—that is, not exacerbating demographic biases in the data or favor one group over another when selecting traits for the personas or segments. In the following, we detail three important research problems (RPs) and our approaches to address these problems in the current study.

- First, there is a lack of studies comparing algorithms for persona generation (RP1). Apart from a brief comparison by Brickey et al.,18 none of the studies address what kind of personas different algorithms produce. The lack of studies poses a major hindrance for understanding how the choice of algorithm affects persona generation and whether there is a risk for algorithmic bias in persona generation (and user segmentation in general).

- Second, there is a lack of using non-accuracy metrics for persona evaluation (RP2). What we mean by this is that apart from25 that evaluated inclusivity, quantitative persona studies tend to focus on evaluating personas’ technical accuracy26,27 as opposed to what kind of personas are generated. However, the characteristics of the generated personas matter because persona traits (e.g., age, gender, ethnicity) risk aggravating stereotypical thinking about user populations.28,29

- Third, there is a general lack of sharing resources for persona generation (RP3). The lack of publicly available resources—code, data, algorithms, computational notebooks—poses a major hindrance to research on data-driven personas24 and the broader field of user segmentation. Even though making results are resources available for others is a basic principle for scientific progress,30 a few data-driven persona studies have made their resources available for others, thus limiting their contribution to the field.

To address RP1, we conduct an experimental study where we fix the (a) dataset, (b) the number of personas generated, and (c) the method of enriching the personas, only varying the algorithm that processes the baseline user data. Although several effects of algorithms on persona generation (and more broadly on customer segmentation) could be examined, including, for example, accuracy, computational efficiency, or run-time, we focus on studying the effect of algorithm on model outputs, that is, the generated personas. We specifically address the effect of algorithm on the demographics of the persona set—we choose demographics as the unit of analysis because of their widespread use in both persona generation and customer segmentation overall.

To address RP2, we examine three aspects vital to decision makers interested in personas for representing their user, audience, or customer base. These aspects
are diversity (i.e., the personas cover many unique characteristics found in the user base), fairness (i.e., the personas truthfully reflect the underlying data on the users), and consistency (i.e., the algorithm retains central persona traits when changing the hyperparameters). We refer to these design goals for data-driven personas and customer segments as the diversity, fairness, consistency (DFC) criteria.

To address RP3, we make the resources (code, data, algorithms) available in an online repository\(^1\) to further advance empirical persona generation and customer segmentation experiments. To protect business sensitive information, the data are made available in a scrambled format.

Building on the earlier reasonings, our research question is: How do diversity, fairness, and consistency of the generated personas’ demographic attributes vary by algorithm? The results have implications for creating more diverse, fair, and consistent personas (and other forms of user segments) from digital user data. Our results inform persona developers (and marketers) of the strengths and weaknesses of different algorithms when applied to persona generation (or customer segmentation).

They also provide wider implications for the potential of algorithmic bias when using algorithms for user or customer segmentation and draw attention to the use of non-commercial segmentation criteria influenced by the fairness/bias discussion going on in the computer science and HCI communities,\(^31\) which has not thus far been addressed in the persona generation and customer segmentation domains. The main contributions include the following: (a) Providing the most extensive comparative study to date, comparing six different algorithms for persona generation; (b) introducing and applying novel evaluation metrics in the persona generation context, while also providing suggestions for applying different algorithms for different objectives; and (c) sharing computational resources to further persona generation research and development.

### Literature Review

Table 1 shows a list of commonly applied persona generation algorithms based on a sample of 63 research papers. The sample of articles was obtained as follows. First, we searched two major academic services: Google Scholar and ACM Digital Library. We used search phrases relating to persona generation (“algorithmic personas,” “data-driven personas,” “quantitative personas,” “procedural personas”). The found articles were manually screened by reading the abstracts. We excluded articles if they were not (a) full research articles written in English, (b) published in a peer-reviewed journal or conference, and (c) developed personas using data-driven approaches. We also carried out snowball sampling to fully account for relevant literature.\(^32\)

In total, the database searches and snowballing yielded 163 articles for full-text screening. For the screening, we applied the same criteria (a–c). After the full-text review, a set of 63 final articles remained. We then reviewed these articles to identify the algorithm(s) applied.

From the reviewed articles, we identified five commonly used algorithms for persona generation: (1) CA, (2) PCA, (3) LSA, (4) NMF, and (5) Latent Dirichlet allocation (LDA). Table 1 provides an overview of these algorithms and the studies adopting them. The use of the five algorithms for persona generation is discussed in the following subsections. A technical description of each algorithm and their implementation for this study is given in Appendix A1. The focus of this study is not on the technical traits of these algorithms but on their implications for APs and user segmentation in general.

#### Clustering

Tanenbaum et al\(^33\) utilized CA (k-means) to develop personas of diabetes patients and gauge patients’ readiness for adopting different medical interventions. Validation was done by calculating the Euclidean distance between the different variables and conducting Chi-squared tests. Wang et al\(^34\) also calculated Euclidean distances of different medical and demographic variables for their analysis of regional health data. A few articles qualitatively validated clusters by engaging subject experts as well as users themselves in reviewing the cluster results.\(^23,35,36\)

These individuals were tasked with assessing how representative the generated clusters were of real-life scenarios. An et al\(^37,38\) applied CA (k-means) for persona generation, and they observed that using individual-level data is expensive to collect and has concerns regarding privacy. However, the researchers did not address what kind of personas different algorithms would produce.

Kwak et al\(^39\) using CA (k-means), found the limitation that a single demographic group must fall into one

---

\(^1\)https://github.com/joolsa/Persona-Generation
persona. In contrast, various personas can be found from one demographic group, as people in the same demographic group often behave differently. A potential issue of CA is the “need for specialists to use expert judgment during clustering [to define hyperparameters].” However, this issue concerns other algorithms, too, as persona developers typically need to set the number of personas as part of the process.

Miaskiewicz et al. and Mesgari et al. applied hierarchical CA to develop clusters (and ultimately personas) of university members’ experiences with learning management and institutional knowledge systems. Both studies validated their results by looking at the relations between variables within clusters. The former calculated cosine similarity (of the angles between pairs of non-zero vectors), whereas the latter calculated Pearson correlation (the extent of a linear relationship between two variables).

Holden et al. used hierarchical CA to analyze the medical and psycho-social variables of older adults with heart failure. Results were validated using the Kruskal–Wallis test and Welch’s analysis of variance (ANOVA) to determine statistical significance between the variables as well as testing for variance, respectively. However, the researchers did not address what kind of personas different algorithms would produce.

Principal component analysis
The PCA was used for persona generation by Sinha, who created personas based on users’ characteristics. Wang et al. used PCA in combination with CA to develop health personas of regional groups. Similarly, Brickey et al. used PCA in combination with LSA and CA to develop personas for users of an army knowledge management system. As such, the combination of this method with others is prevalent. In fact, all of the found studies that applied PCA complemented it with at least one other quantitative method.

As a result, validation metrics also varied and included Cohen’s kappa (a statistical measure of inter-rater agreement of generated vs. expert clusters), Euclidean distances of different variables, Spearman’s $\rho$ (direct association between two ranked variables), and a qualitative review with survey participants. However, the researchers did not address what kind of personas different algorithms would produce.

Latent semantic analysis
Apart from the aforementioned studies, Dupree et al. used LSA to investigate attitudes toward digital privacy and cybersecurity among university students, and they recruited a separate population (different from initial survey participants) to review the developed personas. The evaluation focused on tasking individuals with self-identifying with one of the five personas and providing feedback on how realistic they were. However, the researchers did not address what kind of personas different algorithms would produce.

Non-negative matrix factorization
An et al. used NMF to decompose an interaction matrix constructed from the view counts of an organization’s social media content. They obtained latent content consumption patterns, associating each distinctive behavior pattern with demographics of users (i.e., age, gender, country) by a weight assessment that encodes the strength of the relationship between the demographic groups and the underlying pattern. The demographic group with the highest NMF weight

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
<th>Frequency, N (%)</th>
<th>Examples (Refs.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CA</td>
<td>Groups a dataset using a predetermined number of clusters. Popular approaches are partitioning-based approaches such as k-means and agglomerative such as hierarchical clustering</td>
<td>22 (34.9)</td>
<td>33,37,38,41,42</td>
</tr>
<tr>
<td>PCA</td>
<td>Linear dimension-reduction algorithm used to extract information by removing non-essential elements with a relatively small variation</td>
<td>5 (7.9)</td>
<td>18,34,44–46</td>
</tr>
<tr>
<td>LSA</td>
<td>Data analysis algorithm that uses singular value decomposition to detect hidden semantic relationships between words</td>
<td>5 (7.9)</td>
<td>18,40,46,67,83</td>
</tr>
<tr>
<td>NMF</td>
<td>Method in which data matrices are constrained as non-negative and decomposed to extract sparse and meaningful features</td>
<td>4 (6.4)</td>
<td>48–50,54</td>
</tr>
<tr>
<td>LDA</td>
<td>A generative statistical model that models each item of a collection (typically text) as a finite mixture over an underlying set of patterns</td>
<td>6 (9.5)</td>
<td>49,50,55,56,83,84</td>
</tr>
</tbody>
</table>

Percentages are of total reviewed articles. The rest (N = 21, 33.3%) are classified as “Other.” CA, cluster analysis; LDA, Latent Dirichlet allocation; LSA, latent semantic analysis; NMF, non-negative matrix factorization; PCA, principal component analysis.
was chosen as the representative persona demographics for its corresponding behavioral video-viewing pattern.

The demographic group was then augmented with a name, occupation, photo, and other characteristics, yielding complete persona profiles. Similar approach was applied in other studies from the same research team, with the general goal of developing personas from social media user statistics. However, the researchers did not address what kind of personas different algorithms would produce. For evaluation, rankings of the generated demographics groups in these studies were compared with true rankings based on real content engagements using Kendall rank correlation coefficient.

Researchers identified discriminative content for personas (i.e., content that a persona has a higher chance of engaging with compared to other personas) using a Chi-square test. Also, cosine similarity has been applied to calculate among pairs of personas until the closest pairs are determined. Further, Salmi nen et al. used qualitative data of social media users in a geographical region in the forms of Instagram profiles and semi-structured interviews to create what they term as “hybrid personas” (the core algorithm being NMF).

Qualitative research was used to enrich further and improve their hybrid personas. However, the researchers did not address what kind of personas different algorithms would produce.

Latent Dirichlet allocation
The LDA was utilized in two studies to understand the viewing behavior of different demographic groups and develop user personas for a YouTube channel. In the two studies, the authors built LDA topic models to construct matrices (in combination with NMF). Dhakad et al. developed buyer personas from click logs on an e-commerce portal. They employed LDA to model persona preferences for different occasions by sampling fashion styles and relevant fashion items across online shoppers’ activity. Further, Smith and Nayar used LDA to develop gaming personas based on controller input data from video game analytics systems. However, the researchers did not address what kind of personas different algorithms would produce.

Methodology
Research design
Conceptually, our research methodology uses different algorithms on the same dataset to produce sets of 5, 10, and 15 segments that are the bases for the personas (common numbers used in persona research). We complete the persona generation using a standardized approach to fully generate persona profiles with a name, picture, age, gender, country, interests, and other information. We then compare these sets of personas using three quantitative metrics (see the Evaluation Metrics section) to examine how the generated personas differ by algorithm. The process is as follows, with a technical approach mentioned in parentheses:

The experiment set-up is as follows: each algorithm processes the same data using the same number of segments \((N = \{5, 10, 15\})\), and we then enrich these segments using the same algorithmic process of assigning demographics and other information. In total, there are \(3 \times 6 = 18\) persona sets (3 number levels and 6 algorithms). Using three different levels for the number of personas enables us to analyze the consistency of the results.

Note that the same enrichment process is employed, producing 5, 10, and 15 personas for each algorithm. Fifteen personas is more than the conventional number of “less than 10 personas” widely cited in the HCI literature. In this study, we consider the higher number of personas sensible, as many organizations deal with large and diverse online audiences that cannot be captured in a handful of personas. The outcome of the data collection and application of the algorithms is a collection of datasets representing the main segments or dimensions in the data.

Data collection and pre-processing
We partner with an international news and media organization with 5.08 M subscribers at the time of writing (April 2020) to collect a dataset that contains both behavioral (i.e., what videos were viewed and how many times) and demographic statistics about users. We collect the data from the organization’s YouTube Channel by leveraging the YouTube Analytics Application Programming Interface. The dataset contains 363M views for 12.3K videos published between March 2007 and December 2019.

The justification for the dataset in terms of helping us achieve our research goals relies on the following rationale: (a) the dataset is large (typical for online user data), (b) its structure is typical for Web analytics platforms (e.g., Google Analytics and Facebook

\(^1\)https://developers.google.com/youtube/analytics
Insights provide the same output), and (c) its analysis extends beyond what can be done manually, requiring algorithmic processing to build robust personas. Also, (d) this dataset is typical for many large online businesses that generate much content or have many products/pages to offer.

The distributions of view counts by gender and age are shown in Figure 3. Geographic distribution is shown in Figure 3c.

After collecting the user data from an online analytics platform, the data are transformed into an interaction matrix that captures the engagement between user groups (rows) and the content items (columns). In this dataset, the content indicates online videos, but depending on the dataset, the content can be webpages, e-commerce products, flight destinations, or other entities of interest.

The cell values in the interaction matrix indicate the number of interactions that a given demographic group (row) has for a given content (column). The demographic groups come from the online analytics and social media analytics platforms that use this grouping to aggregate user data and to protect the privacy of individual users. The age buckets used by these platforms—for example, YouTube Analytics, Google Analytics, and Facebook Insights—include 13–17, 18–24, 25–34, 35–44, 45–54, 55–64, and 65+. These groups are used to set up the interaction matrix.

For example, the demographic group “Finnish, 35–44, Male” can have 1200 views for “Video ABC.” Thus, the values of the matrix are counts (15, 4000, 55,867, ...), always either positive or zero. Many demographic groups typically have zero values for a given content, but this sparsity depends on the dataset. In our data matrix, 98.764% of the values are zeros, indicating high sparsity (the more content and demographic groups there are, the higher the sparsity tends to be because not all groups would be interested in all content).

Figure 4 provides a more formal explanation using the example of the NMF algorithm. In Figure 4, \( V \) indicates the \( g \times c \) matrix of \( g \) user groups and \( c \) online contents. The element of \( V \), \( V_{ij} \), is any number that reflects the user group \( G_i \)'s interest or engagement or intent toward content \( C_j \). In the case of Google Analytics, \( V_{ij} \) is typically a session for a given webpage, \( C_j \) from user group \( G_i \). The NMF decomposes \( V \) to two lower dimensionality matrices, \( W \) (demographics) and \( H \) (contents), both of which are defined by \( p \) latent patterns \(^49\); \( p \) being a hyperparameter that indicates the number of personas generated. More details about NMF can be found in the seminal work by Lee and Seung.\(^59\)

Each algorithm independently processes this interaction matrix, finding \( p \) patterns (clusters, segments, components), where \( p = \{5, 10, 15\} \). In this case, because the content is online videos, the segmentation aims at preserving information on the video viewing behaviors of different demographic groups. These \( p \) segments become \( p \) personas when they are enriched by personified information (e.g., name, picture, topics of interest) to create the final personas.

Personification and enrichment are standard procedures for persona generation\(^60\); without it, the segments would remain as nameless and faceless user representations—the general benefit of personification is that human attributes increase stakeholders’ empathy toward the segment that the persona represents,\(^61\) whereas enrichment provides a more rounded, detailed information about the persona.\(^2\)

**Persona generation**

For the enrichment, we use automatic persona generation\(^5\) (APG), a system for APG. This system has numerous advantages, including standardization. What this means in our experiments is that all outputs by the algorithms undergo the same enrichment process that involves no manual intervention. For example, each piece of content is topically classified (explained in An et al\(^49\)), providing topics of interest for each persona.

Based on the outputs, APG chooses a representative demographic group for each latent pattern and enriches this demographic group with personified and other information (picture, name, topics of interest, quotes, etc.) to create a complete persona profile (Fig. 4). The result is a distinct set of personas based on behavioral and demographic attributes of the user population. Note that APG’s procedure for assigning the demographic group is identical and deterministic—with the given data and algorithm, a set of \( p \) personas will always have the same age, gender, and country when using APG for persona generation. As the only variable that changes in the process of our experiment is the algorithm, the differences in the generated personas stem from the algorithms. For a more detailed explanation and validation of APG, see An et al.\(^48\,\,49\)

\(^1\)https://persona.qcri.org
FIG. 3. Distribution of video view counts by (a) gender, (b) age, and country (c) in the baseline data. Distribution of video view counts by country and region (the smaller plot shows Top-20). The distribution indicates a highly imbalanced dataset, typical for online user data. This is also observed from descriptive statistics: there are 185 countries and regions; on average, a country or region has 1.97M views but the standard deviation is 9.06M (4.6 times the average).
Choice and implementation of algorithms

The algorithms identified for persona generation in this study are described in Table 2. Out of the chosen algorithms, CA, PCA, NMF, and LDA have previously been applied for persona generation. The LSA was not selected because it is a type of matrix factorization algorithm (using singular value decomposition), and we already selected NMF for testing. Because we also wanted to test new methods for persona generation, we selected two novel (as in previously not applied for APs) algorithms: Uniform Manifold Approximation and Projection (UMAP) and Spectral Embedding (SE).

Given our analysis of existing algorithms, these two seem like next logical algorithms for persona generation to take on. Thus, we ended up with six algorithms to test. The chosen algorithms support the goals of this study for multiple reasons: they (a) represent the most often used algorithms in persona research, so comparing them is relevant; (b) are standard approaches in computer science, which affords implementation and replicability in future studies; (c) involve variability (Table 2), which means that the comparison is likely to result in meaningful differences (e.g., in terms of linear and non-linear algorithms); and (d) are readily available in software packages and data science libraries, which facilitates their deployment in persona generation projects in practice. The specific implementation we use for each algorithm is explained in Appendix A1.

Evaluation metrics

Reasoning. The generated persona sets are compared by three metrics, explained in the following subsections. The metrics that we propose for this problem are vital for data-driven personas and customer segmentation in general, because they address non-commercial and ethical aspects of segmentation efforts, areas that are lacking attention in these problem domains but that are broadly acknowledged as important within the computer science community.62,63,64

- Diversity matters as a user segmentation goal, because the segments should represent demographically diverse groups of people. If they do not, decision makers may end up receiving information about only select groups and thus ignore the needs and wants of other groups, posing disadvantage to those groups. As put by Drosou et al.31(p.73) “diversity [is] an important component of a data-responsible society.”
Table 2. Properties of the chosen algorithms

<table>
<thead>
<tr>
<th></th>
<th>Sensitive to imbalanced data</th>
<th>Linearity</th>
<th>Compute complexity</th>
<th>Interpretablity</th>
<th>No. of hyper parameters</th>
<th>Topology preservation</th>
<th>Impact of sparsity</th>
<th>Global/Local</th>
<th>Impact of dimensionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>CA</td>
<td>No</td>
<td>L</td>
<td>High</td>
<td>High</td>
<td>3</td>
<td>No</td>
<td>Medium</td>
<td>Global</td>
<td>High</td>
</tr>
<tr>
<td>PCA</td>
<td>Yes</td>
<td>L</td>
<td>Low</td>
<td>Low</td>
<td>1</td>
<td>Yes</td>
<td>Low</td>
<td>Global</td>
<td>Low</td>
</tr>
<tr>
<td>NMF</td>
<td>Yes</td>
<td>L</td>
<td>Medium</td>
<td>Medium</td>
<td>1</td>
<td>No</td>
<td>Medium</td>
<td>Local</td>
<td>Low</td>
</tr>
<tr>
<td>LDA</td>
<td>No</td>
<td>L</td>
<td>Medium</td>
<td>Low</td>
<td>1</td>
<td>No</td>
<td>High</td>
<td>Global</td>
<td>Medium</td>
</tr>
<tr>
<td>UMAP</td>
<td>No</td>
<td>N</td>
<td>Medium</td>
<td>High</td>
<td>1</td>
<td>Yes</td>
<td>Low</td>
<td>Local</td>
<td>Low</td>
</tr>
<tr>
<td>SE</td>
<td>Yes</td>
<td>N</td>
<td>High</td>
<td>Medium</td>
<td>2</td>
<td>Yes</td>
<td>Low</td>
<td>Local</td>
<td>Low</td>
</tr>
</tbody>
</table>

Topology Preservation means whether the algorithm preserves the structure of the data: if two items are neighbors in the high dimension, will they also be neighbors in the low dimension. Global/Local: whether the algorithm tries to preserve global structure or local interactions. Usually, linear algorithms focus on global and non-linear ones on local. Impact of dimensionality: what the impact of the original dimension size is. For example, PCA and NMF work well for millions of dimensions, whereas CA quickly becomes inoperable.

- Fairness is understood in terms of equity (“equity is defined as the quality of being fair and impartial”) so that the demographic segment’s probability of being included in a finite number of segments shown to decision makers should reflect the segment’s share of voice (i.e., representativeness, size, importance) in the baseline dataset, meaning the dataset that the user segments are created from. If a segment is highly prevalent in the baseline data but hardly visible among the segments, that would not be a fair (or equitable) representation of the data.

- Consistency matters because one would expect that if making changes to the algorithm’s parameters, such as the number of segments created, on different runs of the algorithm the same or similar segments should be identified by the algorithm. If the algorithm instead identifies very different segments at each run, it is behaving in an unstable or random way and its outputs should be less trusted than those of a more consistent algorithm.

Regarding the interpretation of the obtained scores, high diversity indicates that a lot of different demographic groups are represented in the generated personas (segments). High fairness indicates that the generated personas (segments) correspond well with the most engaged users in the source data. High consistency indicates that the personas (segments) generated using a smaller number of personas (segments) appear also when changing the hyperparameter to higher number of personas (segments), thereby indicating higher reliability that these particular segments are important—or otherwise put, that the algorithm is not randomly selecting the segments.

---

Diversity. We use the count of unique attributes for comparing diversity. The count of unique attributes (D) is the number of persona attribute values (age groups, genders, countries) present in each persona set. For 15 personas, a value such as D_{age} = 16 would indicate that the algorithm designed a set of personas with 16 unique demographic attributes, for example, 2 genders, 5 age groups, and 9 countries (D = 2 + 5 + 9 = 16). Note that D can be computed by considering all three demographic attributes and each demographic attribute separately.

For example, D_{age} = 5 means the persona set contains personas from five unique age groups. D affords a straightforward interpretation of diversity between the persona sets. For example, if Persona Set A has personas from three age groups and Persona Set B has personas from six age groups, the latter is considered to be (6–3)/3 = 100% more diverse (in terms of age) than the former.

Fairness. We use statistical parity (SP) for measuring fairness. Fairness assessments in machine learning tend to focus on prediction or classification. However, persona generation is not a classification task, but an unsupervised learning task. However, we can apply the existing principles from computing studies developing tools for fairness assessment. For example, Dwork et al propose that individual fairness is defined such that similar individuals are treated similarly. In its more elementary interpretation, this implies that a member i in Group A that has the same characteristics (e.g., race, gender) as a member j in Group B will have an equal probability of succeeding (e.g., being chosen for a job).

A classic example of analyzing fairness is using personal attributes such as gender or race to predict whether a person is rich or not (e.g., 50k+ annual salary...
or not). In the case of personas or customer segmentation in general, the analogy of demographic groups being significant is the question, “what is the probability of this demographic class being selected by the algorithm among the generated personas?”

There are (at least) two ways to approach this issue. Equality and equity. Equality would translate to any demographic group having the same expected probability of being included in the generated persona set. An equity-based approach would translate to some groups having a higher expected probability of being included in the personas, because of their special needs or other factors. In our setting, we apply the equity-based approach, and the “other factor” is the demographic group’s share of the engagements in the baseline data.

This means that demographic segments with more views are expected to have a higher chance of appearing in the persona set—this is fair because it corresponds to a truthful representation of the user population. Although fairness criteria are always subject to some degree of relativism, for the objective of finding the bias of algorithms, an essential question is “If 30% of the total views come from the US, then is it fair to say that 30% of the personas in the persona set should be from the US?” If this statement is fair, then SP is an appropriate fairness indicator.

Therefore, we calculate SP as the difference between two values:

\[
SP_i = \frac{pi}{P} - \frac{n_i}{N},
\]

where \(SP\) for a given demographic attribute \(i\) (e.g., \(p = “Male”\)) is its fraction in the persona set \(P\) divided by the corresponding fraction of that demographic group’s engagement counts \(n\) from the total \(N\) engagements. For example, if five personas are from the United States in a persona set of 10, then 50% of the personas are from the United States. Given that 30% of the views in the original data are from the United States, the value will be \(SP = 0.5 - 0.3 = 0.2\). The total \(SP\) is calculated by taking an average across all demographic attribute values.

Consistency. The notion of consistency matters, because some researchers have found personas to be abstract and inconsistent. For example, if the personas included in a 5-persona set would be very different from the 10-persona set, this would call into question the validity of the method. To evaluate how consistent the generated personas are, we computed a Consistency Score (CS) for each algorithm. For this, we take the demographic groups an algorithm generates in the 5 persona set, and we compare how many are the same as in the 10 persona set; then, comparing the 10-persona set with 15 personas.

For example, if all five demographic groups are in the 10 persona set, the score is 1.00. We carry out this calculation three times: comparing 5 personas with 10 personas, 10 personas with 15 personas, and 5 personas with 15 personas. In other words, we also calculate how many demographic groups from the 10 persona set are in the 15 persona set, and how many from the 5 persona set are in the 15 persona set. Thus, we end up with three fractions (e.g., 1.00, 0.80, 0.20)—their average is the final \(CS\).

Using the number of personas is appropriate here since the number of segments is the major hyperparameter shared by all the unsupervised algorithms tested here (hyperparameter refers to an external value of a given parameter that is set by the research as opposed to being internally optimized by the algorithm itself).

Formally, the \(CS\) used in our study can be expressed as follows. Let us have three persona sets, denoted Set A, Set B, and Set C. The number of personas in Set A is \(n_A\). The number of personas in Set B is \(n_B\). The number of personas in Set C is \(n_C\). Without loss of generality, we set \(n_A < n_B < n_C\). Comparing the three sets will result in a sub-\(CS\) for three combinations: Set A—Set B, Set A—Set C, and Set B—Set C. That means, we need to choose 2 in 3 sets each time for comparison of consistency. Thus, there are \(C(3,2) = 3\) combinations.

Then, the \(CS\) for each algorithm is calculated as follows:

\[
\frac{1}{3} \left( \frac{\sum_{i=1}^{n_A} P_{p_i \in \{P_A\}}} {n_A} + \frac{\sum_{i=1}^{n_B} P_{p_i \in \{P_B\}}} {n_A} + \frac{\sum_{i=1}^{n_C} P_{p_i \in \{P_C\}}} {n_B} \right),
\]

where \(P_{Ai}\) denotes the Persona \(i\) of Set \(A\) and \(\{P_B\}\) denotes all the personas of Set \(B\). \(P_{Ai \in \{P_B\}}\) is the indicator function, which is:

\[
P_{p_i \in \{P_A\}} = \begin{cases} 1 & \text{if } P_{Ai} \in \{P_B\} \\ 0 & \text{if } P_{Ai} \notin \{P_B\} \end{cases}
\]
Results

Diversity

The diversity results are shown in Table 3. Results from a two-factor repeated-measures ANOVA show that the algorithms significantly differ by their $D$ values, $F(5, 10) = 12.49, p < 0.001$. The algorithms with the highest $D$ values tend to be LDA, UMAP, and SE. A post hoc analysis (Welch’s $t$-test) indicates that algorithms in Group 1: CA, PCA, and NMF generate significantly more unique persona attributes ($M=16$) than algorithms in Group 2: CA, PCA, and NMF ($M=11$), $t(13.79) = -2.45, p = 0.028$. The observed effect size ($d=1.16$) indicates that the magnitude of the difference between the groups is large.

For the five personas set, LDA produces personas with 57.1% more unique demographic attributes than CA, PCA, and NMF. For the 10 personas set, LDA and UMAP produce personas with 70.0% more unique demographic attributes than CA and 54.5% more than PCA and NMF. For the 15 personas set, LDA and UMAP produce personas with 83.3% more unique demographic attributes than CA, 46.7% more than PCA, and 37.5% more than NMF.

In terms of age, two rare age groups are the youngest (13–17) and the oldest (65+) age group. The age group 13–17 appears in five persona sets (LDA$_5$, LDA$_{10}$, UMAP$_5$, UMAP$_{10}$, and UMAP$_{15}$), whereas the age group 65+ only appears in three persona sets (UMAP$_{10}$, UMAP$_{15}$, and SE$_{15}$). An example of a persona from this age group is shown in Figure 5c. Although the age groups of these personas are less common, the countries of the generated personas tend to belong to the Top-10 countries in the baseline data, with the curious exception of Antigua and Barbuda. Interestingly, none of the persona sets contain personas from all age groups, implying that more personas beyond the number of 15 are needed to cover all age groups in the data.

All the algorithms generate personas from both genders, apart from two cases, in which none of the personas generated by the algorithm were female: CA$_5$, and LDA$_{15}$ (Table 3).

The countries of the personas show an interesting finding in that some of the algorithms generate personas that also represent fringe geographics, that is, countries that have a very low proportion of views in the baseline data. LDA, UMAP, and SE account for most of the marginalized personas, examples shown in Figure 5. For example, view counts from users from Finland are only 0.145% of the total view counts; however, UMAP$_{15}$ generates as Finnish persona (Fig. 5b).

Table 3. Diversity results

<table>
<thead>
<tr>
<th></th>
<th>CA</th>
<th>PCA</th>
<th>NMF</th>
<th>LDA</th>
<th>UMAP</th>
<th>SE</th>
</tr>
</thead>
<tbody>
<tr>
<td>5 Personas</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$D_{age}$</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>$D_{gender}$</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>$D_{country}$</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>5</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>$D_{total}$</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>11</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>10 Personas</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$D_{age}$</td>
<td>5</td>
<td>3</td>
<td>3</td>
<td>5</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>$D_{gender}$</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>$D_{country}$</td>
<td>3</td>
<td>6</td>
<td>6</td>
<td>10</td>
<td>9</td>
<td>8</td>
</tr>
<tr>
<td>$D_{total}$</td>
<td>10</td>
<td>11</td>
<td>11</td>
<td>17</td>
<td>17</td>
<td>15</td>
</tr>
<tr>
<td>15 Personas</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$D_{age}$</td>
<td>5</td>
<td>3</td>
<td>3</td>
<td>6</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>$D_{gender}$</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>$D_{country}$</td>
<td>5</td>
<td>10</td>
<td>11</td>
<td>15</td>
<td>14</td>
<td>13</td>
</tr>
<tr>
<td>$D_{total}$</td>
<td>12</td>
<td>15</td>
<td>16</td>
<td>22</td>
<td>22</td>
<td>21</td>
</tr>
</tbody>
</table>

Higher is better. For example, the value of 2 for CA, 15 personas (first column, third row from the bottom) indicates that among the 15 personas, CA generated personas from two different age groups. “Total” indicates the sum of unique age groups, genders, and countries. Highest total values of diversity bolded.

Fairness

Fairness results are shown in Table 4. A two-factor repeated-measures ANOVA shows that the algorithms significantly differ by their $SP$ scores, $F(5, 10) = 8.21, p=0.003$. The lowest $SP$ scores (reversely, the highest fairness) tend to be among CA, PCA, and NMF. Similarly, the persona sets significantly differ by their $SP$ scores, $F(2, 10) = 19.28, p < 0.001$. This indicates that the number of personas affects the diversity scores. However, unlike in the case of diversity, the best fairness values are obtained with 10 personas. As with $D$, the post hoc analysis indicates two distinct groups emerging from the fairness results.

Group 1: CA, PCA, and NMF has significantly smaller $SP$ scores ($M=0.070$) than Group 2: LDA, UMAP, and SE ($M=0.118$), $t(11.84) = -3.475$,
“Three personas you would otherwise not see.” Among the examples of personas with unique demographics, there is only one persona from Ireland (a), only one from Finland (b), and only one from the age group of 65+ (c). The rarer demographics emerge only with certain algorithms when increasing the number of persona in a set. (a) Conor—(SE13), (b) Ville (UMAP13), (c) Sarah (SE13). SE, Spectral Embedding.
Table 4. Statistical parity scores

<table>
<thead>
<tr>
<th></th>
<th>CA</th>
<th>PCA</th>
<th>NMF</th>
<th>LDA</th>
<th>UMAP</th>
<th>SE</th>
</tr>
</thead>
<tbody>
<tr>
<td>5 Personas</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>SP</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>age</td>
<td>0.04</td>
<td>0.11</td>
<td>0.05</td>
<td>0.19</td>
<td>0.12</td>
<td>0.14</td>
</tr>
<tr>
<td>gender</td>
<td>0.14</td>
<td>0.06</td>
<td>0.06</td>
<td>0.26</td>
<td>0.26</td>
<td>0.26</td>
</tr>
<tr>
<td>country</td>
<td>0.13</td>
<td>0.10</td>
<td>0.13</td>
<td>0.08</td>
<td>0.11</td>
<td>0.08</td>
</tr>
<tr>
<td>10 Personas</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>SP</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>age</td>
<td>0.03</td>
<td>0.08</td>
<td>0.08</td>
<td>0.13</td>
<td>0.11</td>
<td>0.09</td>
</tr>
<tr>
<td>gender</td>
<td>0.04</td>
<td>0.04</td>
<td>0.04</td>
<td>0.06</td>
<td>0.06</td>
<td>0.06</td>
</tr>
<tr>
<td>country</td>
<td>0.10</td>
<td>0.06</td>
<td>0.06</td>
<td>0.10</td>
<td>0.07</td>
<td>0.07</td>
</tr>
<tr>
<td>15 Personas</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>SP</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>age</td>
<td>0.05</td>
<td>0.09</td>
<td>0.10</td>
<td>0.10</td>
<td>0.07</td>
<td>0.09</td>
</tr>
<tr>
<td>gender</td>
<td>0.01</td>
<td>0.08</td>
<td>0.08</td>
<td>0.14</td>
<td>0.19</td>
<td>0.12</td>
</tr>
<tr>
<td>country</td>
<td>0.07</td>
<td>0.04</td>
<td>0.02</td>
<td>0.08</td>
<td>0.08</td>
<td>0.08</td>
</tr>
</tbody>
</table>

Lower is better (closer to the baseline data). The lowest numbers bolded. The values were obtained as follows: (a) First, we converted all the negative SP values to their absolute values. Then, (b) we calculated the mean of SP for each algorithm’s each persona set. This was repeated (c) for all the attribute values in a demographic category. For example, age values in the table are averages of all seven age groups, indicating how well the given algorithm represents the age group distribution in the baseline data.

SP: statistical parity.

\[ p < 0.005. \] The observed standardized effect size is large \((d = 1.64)\). Note that smaller SP score indicates higher fairness (the lower the value, the closer the persona attributes are to the baseline data).

Consistency

The results of CS indicate that most of the tested algorithms produce consistent personas according to our definition of the persona demographic groups not changing when changing the number of personas generated. Four algorithms achieve the perfect score of CS = 1.0: CA, PCA, NMF, and SE. The two remaining ones behave more inconsistently, with UMAP (CS = 0.40) scoring higher than LDA (CS = 0.10). A perfect consistency implies that the same personas that were part of the smaller number of personas are part of the larger number of personas as well.

The fact that all the algorithms, except LDA and UMAP, rank perfectly on CS implies that the algorithms behave consistently but differently—that is, an algorithm generates the same personas in 10 and 15 persona sets than in the 5 persona set, but the personas differ by the algorithm. Each of the algorithms tries to identify key patterns but have different definitions of importance. Thus, the generated personas tend to be different (Table 5).

We can further quantify how different the personas are by computing the Jaccard coefficient \((J)\) for each pair of algorithms. \(J\) indicates the intersection over the union of two persona sets \(A\) and \(B\) designed by two different algorithms, which can be interpreted as a similarity of persona attributes. Here, \(J\) compares the similarity of sets of personas that are defined by age-gender-country, and it indicates the intersection over the union of two persona sets \(A\) and \(B\) generated by two different algorithms. \(J\) is equal to 1 if the sets are the same and 0 if they are completely different. The results in Figure 6 show that the personas outputted by different algorithms differ substantially, showing a clear clustering among CA, PCA, and NMF.

The inconsistency of the UMAP and LDA can partially explain the high \(D\) scores of these algorithms. In other words, the inconsistency of LDA and UMAP is because they choose novel demographic groups when generating the persona segments. This proposition is supported by the observed strong negative relationship between CS and \(D\) \((p = -0.823)\), which implies diversity-consistency trade-off.

This trade-off is defined as follows: If optimizing for DFC decrease. Conversely, if optimizing for fairness or consistency, there will be less diversity in the personas. This trade-off implies that DFC are conflicting design goals for AP generation, at least when increasing the number of personas. We can tackle this trade-off by taking the average rank of the algorithms and the DFC metrics to assign a composite rank score for each algorithm. When doing so, SE appears as the most “balanced” algorithm (Table 6), followed by NMF. CA, while being the most commonly used algorithm in AP research, ranks the lowest on this composite comparison.

Table 5. The sets of five personas exemplifies how the personas are different, even when generating a small number

<table>
<thead>
<tr>
<th>Persona 1</th>
<th>Persona 2</th>
<th>Persona 3</th>
<th>Persona 4</th>
<th>Persona 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>CA M 25–34 United States</td>
<td>M 25–34 India</td>
<td>M 35–44 United States</td>
<td>M 18–24 India</td>
<td>M 45–54 United States</td>
</tr>
<tr>
<td>LDA M 35–44 South Africa</td>
<td>M 13–17 India</td>
<td>F 13–17 United States</td>
<td>F 55–64 Malaysia</td>
<td>M 45–54 Serbia</td>
</tr>
<tr>
<td>UMAP M 35–44 Brazil</td>
<td>M 35–44 Cote d’Ivoire</td>
<td>M 25–34 Venezuela</td>
<td>F 25–34 Venezuela</td>
<td>F 13–17 Dominican Republic</td>
</tr>
</tbody>
</table>

Personas appearing at least two times are bolded.

F, female; M, male.
Depending on the use case, persona developers may want to prioritize certain design goals, such as diversity over fairness (or vice versa). For this, the computations can be further developed by introducing three parameters: Diversity Penalty ($a$), Fairness Penalty ($b$), and Consistency Penalty ($c$). Let us say that for a particular use case, the persona developers want to increase fairness, but at the same time, they want to maintain a high diversity and consistency of the personas. Thus, they consider all metrics important, but fairness three times as important as the two others. Accordingly, they set $a$ and $c$ to 0.20, and $b$ to 0.60, such that the mean rank score $MRS_i$ is calculated as a weighted sum of rank $R$ of $i$ for each metric:

$$MRS_i = R_{D,i} \times 0.20 + R_{SP,i} \times 0.60 + R_{CS,i} \times 0.20,$$  \hspace{1cm} (4)$$

where the mean rank score $MRS$ of algorithm $i$ is calculated as a weighted sum of rank $R$ of $i$ for each metric.

Using these penalty parameters to compute the $MRS$, the ranking of the algorithms is now in favor of NMF, with SE falling a shared second position with PCA (Table 6). Note that these parameters are presented as examples only; future work should conduct a proper sensitivity analysis. Nonetheless, considering the DFC design goals as “weights” for the algorithms is intuitive.

**Discussion**

**Research contribution**

As far as we know, this is the most extensive study to experiment with different algorithms for persona generation to date. Overall, our results suggest that the tested algorithms can be categorized into two groups: (a) those with low diversity and high fairness (CA, PCA, NMF), and (b) those with high diversity and low fairness (LDA, UMAP, and SE). This relationship is supported by the strong negative correlation (Spearman’s $\rho = -0.83$) between the diversity and fairness.
rankings of the algorithms. Further, the results indicate that the highest diversity is consistently achieved with 15 personas rather than with 10 or 5 personas.

The same cannot be said for fairness; even though the number of personas has a statistically significant effect on SP, the best average performance is obtained with 10 personas. These findings have several implications. First, concerning AP generation, our findings expand the AP work by Chapman et al.\textsuperscript{26} and Chapman and Milham\textsuperscript{27} and Brickey et al.\textsuperscript{18,46} In regards to the former, we provide quantitative metrics for persona evaluation\textsuperscript{27} and evaluate demographic variables rather than coverage or prevalence.\textsuperscript{26} Our study addresses the lack of standardized metrics for persona evaluation\textsuperscript{24} by using three relevant metrics to assess persona generation outcomes.

We also confirm the findings that different algorithms tend to “disagree”\textsuperscript{18}—that is, design different personas from the same baseline data. Although Brickey et al.\textsuperscript{18} tested two algorithms (CA, PCA), our study considers these two and four additional algorithms. Also, Brickey et al.\textsuperscript{18} used inter-rater agreement (Cohen’s $\kappa$) for evaluation, whereas we used metrics specifically tailored for persona generation goals.

In terms of findings, it is interesting that the three methods that maximize diversity—LDA, UMAP, and SE—are also the three methods most rarely applied in the persona literature (Table 1). No previous study uses SE and UMAP to generate APs. The LDA has been previously used,\textsuperscript{18,49} but CA, PCA, and NMF are dominant methods in persona generation. Our findings imply that the research and practice of persona generation (and user segmentation) benefits from experimentation with novel algorithms, as these novel approaches can result in fairer or more diverse persona sets.

Technically oriented researchers often see the proclaimed objectivity of algorithms as an advantage relative to manual persona generation and user segmentation.\textsuperscript{5,6,15,36} However, our findings suggest that it is possible that algorithmically created segments might not be any more diverse, fair, or consistent than those created completely by humans using manual means. Instead, APs can also be subject to demographic biases that may originate from multiple sources, such as data distributions, the way algorithms process the data (in the mathematical sense), or from the assigned hyperparameters.\textsuperscript{63}

To this end, the use of the DFC metrics shifts persona evaluation away from the traditional technical metrics (e.g., perplexity, accuracy, loss, error, etc.) toward evaluating the outputs of persona generation in terms of the kind of personas related to the algorithm design. The metrics we use support the design goals of diverse, fair, and consistent personas, taking a step toward ethically robust APs and user segments that portray the diversity of the user base in an accurate manner.\textsuperscript{28} The connection of persona generation and user segmentation to algorithmic fairness is an important contribution that should be further expanded upon in computational studies dealing with customer segmentation using big data.

Hence, the results suggest there is a need for discussion around algorithmic bias in customer segmentation literature. As we suggest, these concerns can be addressed and awareness to them created by leveraging new metrics inspired by the ongoing fairness/bias discussion in the computing community.

Concerning the real-world impact of our findings, there are crucial observations to be made about the large impact that the choice of one algorithm over another has on the composition of the generated personas or user segments. Given that customer segmentation permeates almost every organization on the planet, there is a crucial need for awareness on how a simple change of algorithm can drastically alter the outcomes obtained from the same customer data. Because firms and decision makers are not looking to offer “everything for everyone” but instead tailor their offerings based on segments, whether using personified segments (i.e., personas) or some other types of segments, the impact of the chosen algorithm seems to be not trivial but drastic.

This observation puts pressure on organizations from two sides: first, (a) which persona generation or customer segmentation algorithm should they choose for a given situation? Second, (b) given the obscure behavior of algorithms for persona generation and customer segmentation, is their use dangerous and potentially misleading? Should new segmentation techniques be developed from scratch? Because of these fundamental questions, we expect this study not to be an isolated incidence, but part of a larger research agenda on improving AP generation and customer segmentation. Although fairness frameworks and acronyms such as Fairness, Accountability, and Transparency, Findable, Accessible, Interoperable, and Reusable,\textsuperscript{57} Fairness, Accountability, Transparency, Ethics, Safety and Security,\textsuperscript{58} and Equity, Accountability, Trust, and Explainability\textsuperscript{69} have been developed by the research
community and industry actors to scrutinize the use of algorithmic decision making in many fields, the application of these frameworks or concepts in the domain of APs (or user segmentation in general) is lacking.

Hence, our study makes an important contribution to investigating fairness in the context of algorithmic user segmentation. Focusing on what type of personas are created is essential, because AP studies often assume that the use of algorithms and quantitative data prevents persona developers from injecting their biased interpretations into the created personas. However, if the use of algorithms would involve aspects of unfair, inaccurate, or inconsistent personas, this would present a major issue for the ethics of persona generation or other type of algorithmic user segmentation.

Algorithms as conveyors of partial truths about the user base

Our results imply that the choice of an algorithm has a fundamental impact on the personas generated. This is an important discovery since APs tend to have an air of objectivity, credibility, and truthfulness in the eyes of stakeholders. Our findings imply that attributing these properties to data-driven user segmentation might not be justified at all times. More precisely, it seems impossible to argue that any of the applied algorithms captures the “truth” about the users. Instead, each algorithm focuses on certain facets of the user population.

Moreover, the complexity of the algorithms (from a mathematical point of view) typically makes it intractable to understand why a specific trait was chosen over another. This intractability concerns personas and all user segmentation efforts carried out using algorithms. The use of algorithms is always “biased” in the sense that different algorithms produce different outputs. Nevertheless, the use of algorithms is always “objective” because, given the same data and the same parameters, an algorithm always produces the same set of personas. Therefore, it is crucial to disentangle the concepts of truth and objectivity—they refer not to the same thing.

Despite this, researchers can define design goals and desiderata for APs. Perhaps even more so because there is no one perfect method for persona generation. In the absence of this perfect method, the focus should be on what kind of personas are being designed by different algorithms: are they diverse, are they fair, are they consistent?

The outcome of using the same data, but getting different results is a conundrum for the application of data science methods for persona generation and user segmentation in general. It not only stresses the “design power” that the algorithms have but also involves a more fundamental, perhaps unanswerable question of which algorithm correctly portrays the users. This question, associated with epistemological standpoints such as truthfulness and objectivity of algorithms when creating data-driven user segments, can be traced back to the discussion on the (im)possibility of scientific verifiability and falsifiability of personas and data-driven user segments.

Within the scope of this study, we are unable to provide definite answers in this regard. However, we express the concern that a precarious use of quantitative methods, coupled with stakeholders’ overconfidence in algorithmic superiority due to the mystique involved with quantitative data and mathematical formulas, can result in a disservice. In turn, broader awareness of there not necessarily being “one truth” about the user segments is likely to increase confusion among end-users of personas who the “real” personas are. Perhaps what is needed is to switch the argument for AP generation from one single objectivity to relative subjectivity: here is what an algorithm has to say about your users—but it is not the whole truth.

Practical implications

If all algorithms generate different personas, which one should a decision maker choose? An answer arises, on one hand, from preference and context for which the personas are developed (i.e., the purpose) and, on the other hand, from the intimate understanding of the nature of different algorithms when exposed to specific data (i.e., the know-how). Persona developers may choose to maximize DFC to generate persona sets that are the most applicable for their use case. Our results show that using different target metrics yields mixed results. The choice of the algorithm depends on the goal of persona generation. In particular,

- To optimize for diversity, use LDA, UMAP, or SE.
- To optimize for fairness, use CA or NMF.
- To optimize for consistency, use CA, PCA, NMF, or SE.
- When accounting for all the three criteria, use SE or NMF.

Our results show that different algorithms design different personas from the same user data. Thus, the practitioner’s choice of algorithm ultimately results in different personas. This implies that the choice of the
algorithm should not be taken lightly. More precisely, the practitioner faces two important choices: (a) the choice of the algorithm for persona generation, and (b) the choice of the hyperparameters for the selected algorithm. The ethical implication is that rather than hiding these choices under the parlance of “statistical,” “objective,” and “data,” transparency and discussion of the pros and cons of these choices should be undertaken by the wielders of the algorithms.

If one cannot explain it, one probably should not be using it. Moreover, an important guideline is to consider the goal of the customer segmentation or persona generation exercise in the first place—for example, if one seeks to get as varied understanding of the user population as possible, then using an algorithm that maximizes diversity would be beneficial. If, instead, one seeks to get a tight understanding of the most engaged segments, a fairness-based algorithm would be applicable. The paradigm of “here is our data—algorithm, please show our segments” needs to be revised to “here is our data AND our goal—algorithm, please show our segments.”

Implications for segmentation researchers
Concerning replicability and applicability, the fact that the dominant online platforms tend to output a data structure that is compatible with persona generation means that, by using this data and publicly available data science algorithms, anyone with access to data and necessary programming skills can generate personas. Thus, the practical implications of this study range across many industries and contexts, like the method of personas itself. For example, from YouTube Analytics, one can collect videos and their view counts; from Google Analytics, pages and their session counts; from Facebook Ads, the ads and their interaction metrics (views, clicks, purchases).

To advance the use of data-driven personas, we share our source code and data (with content IDs masked and view counts randomized for protecting business-sensitive information). Researchers and practitioners can obtain these resources via the code repository. Sharing data, algorithms, and code is crucial for achieving progress within the user segmentation research and practice and we hope that research contributes to setting an example of making computational materials and resources for persona generation and customer segmentation available to both research and practitioner communities.

Limitations and future research directions
There are several directions to pursue from our findings.

First, the properties of the algorithms most likely explain some of the results. For example, some algorithms may be more sensitive to imbalanced data. Although Table 2 provides descriptive information of the tested algorithms, we did not test how these properties of the algorithms affect the results. This is primarily because of the parsimonious experimental setting that focused on observing the effects of using different algorithms on the demographic composition of the generated personas. More work is required to establish explanations as to why the algorithms behave differently, but providing such explanations is beyond the scope of this work and thus left for future research.

Second, future research should also be directed toward a systematic understanding of how the dataset properties affect the results. These properties may include (a) prevalence of different user demographics (i.e., the number of rows), (b) distributions of engagement across those user attributes, (c) size of content (e.g., small organizations vs. big content producers), and (d) sparsity of the interaction matrix. The more datasets one would analyze, given they contain variation along with these properties, the better one would understand the relationship of data properties and the personas generated by the algorithms.

Third, another interesting question is whether personas generated by different algorithms could be more/less similar under a different parametrization. In this study, we kept the hyperparameters (mainly the number of personas) fixed to control the effect of number on the results, but future work could investigate how the manipulation of the algorithms’ hyperparameters affects persona generation. We limited the persona sets to three since the main focus of the study was on the algorithms and not the number of personas.

We chose the numbers of personas for the sets (5, 10, 15) based on the fact that previous research tends to favor a relatively low number of personas in a set. Nevertheless, both of these parameters could be altered by (a) comparing more sets, and (b) increasing the number of personas beyond 15. Such extended analyses would help better understand the effect of the number hyperparameter on the APs.

†https://github.com/joolsa/Persona-Generation
Fourth, there is a grave need for explainability and interpretation of unsupervised algorithms such as the ones we deployed in our study—generally, this a challenge for the whole machine learning domain (see, e.g., Hasani et al71). As our findings show, due to the unpredictable nature of most algorithms, explaining their “thought process” of choosing the specific set of demographic segments should be scrutinized in dedicated studies.

Related to explainability and to the “disagreement” among the algorithms about what segments to highlight, there is a lingering question about the design of entirely new data-driven persona and customer segmentation algorithms. Here, using interactive and intelligent system functionalities alongside with computational techniques such as top-N picking and outlier detection can, we believe, yield results that are simpler and provide more meaningful and interpretable results for stakeholders than the currently used black-box algorithms.

Fifth, other algorithms beyond the ones we tested could be experimented with. We chose the specific algorithms based on their commonness in persona generation and customer segmentation, but more advanced or differently designed algorithms in the current body of computer science could deliver complementing results. The chosen algorithms also include derivative version such as constrained NMF72 that could be explored in future studies.

Finally, the evaluation of personas and customer segments is generally considered an ongoing research area with room for contribution.27,73 Although we propose metrics to quantify “good personas” according to certain design goals, more quantitative metrics for persona evaluation could be devised, which remains an important goal for future research. User segmentation research could also investigate ways to incorporate the metrics directly into the algorithm’s objective functions, rather than focusing on a post hoc analysis of the personas. New algorithms could make it possible for creators to specify their DFC targets before persona or segment generation.

Conclusion
Persona generation via algorithms is widely considered as objective in contrast to manual persona generation, but it is largely overlooked that different algorithms actually generate very different personas. Our results indicate two groups of algorithms that produce very different outcomes for persona generation: algorithms that generate personas with high diversity/high fairness and those that generate personas with high diversity/low fairness. Most algorithms produce consistent results independent of the number of personas.

Persona developers should take care when selecting an algorithm for persona generation (or user segmentation in general), as the algorithm’s choice impacts the DFC of the personas. The fact that the algorithms create different personas from the same user data implies that algorithms have more influence in the user segmentation process than commonly understood.

Author Disclosure Statement
No competing financial interests exist.

Funding Information
No funding was received for this article.

References
tional Requirements Engineering Conference (RE 2007). Delhi, India; 2007; pp. 185–194; doi: 10.1109/RE.2007.50.


Abbreviations Used

APs = algorithmic personas
ANOVA = analysis of variance
APG = automatic persona generation
API = application programming interface
CA = cluster analysis
CS = Consistency Score
DPC = diversity, fairness, consistency
DR = dimensionality reduction
HC = hierarchical clustering
HCl = human–computer interaction
LDA = Latent Dirichlet allocation
LSA = latent semantic analysis
NMF = non-negative matrix factorization
PA = partitional clustering
PCA = principal component analysis
PCs = principal components
RPs = research problems
SE = Spectral Embedding
SP = statistical parity
UMAP = Uniform Manifold Approximation and Projection

(Appendix follows →)
Appendix

Appendix A1. The Implementation of the Algorithms

Cluster Analysis
Cluster analysis (CA) is often considered as the default technique for exploring similarity within multidimensional data. CA is a collection of unsupervised learning methods that aim to group objects depending on how similar they are. There are several CA techniques that can be divided into hierarchical clustering (HC) and partitional clustering (PA).

The HC groups individual items into larger groups by computing distances between different elements to produce clusters in a hierarchical order based on similarity. The clustering procedure continues until the data are reduced to the number of clusters set as the hyperparameter.

The PA breaks down larger objects into smaller ones. In contrast to HC, no hierarchy is followed when forming the clusters. K-means is a typical PC method, where the user defines the number of clusters into which the data are divided. These clusters reveal common patterns within distinct groups and can be built upon to form personas.

For our implementation of clustering, we use the Feature Agglomeration from the Python library of Scikit-learn, which corresponds to the clustering type of HC. The parameters of the algorithm are as follows:

- **n_clusters:** Number of clusters to create. We used 5, 10, and 15.
- **affinity:** Metric used to compute the association between the clusters. We used “euclidean” (default). Other options are “l1,” “l2,” “manhattan,” “cosine,” and “precomputed.”
- **linkage:** The linkage parameter decides which distance to use between sets of features. The algorithm tries to merge the pairs that minimize this criterion. We used the “Ward” (default option) that minimizes the variance of the clusters. Other options include “Average” (averaging the distance of each feature), “Complete” (using the maximum distances between all the features), and “Single” (using the minimum distance between all observations).

Principal Component Analysis
Principal component analysis (PCA) is one of the most widely used dimensionality reduction (DR) techniques aimed at dealing with a large number of numerical values. The PCA is considered easy to use; it works for the vast majority of datasets and is offered in most data science packages alongside with supervised methods (partial least square discriminate analyses). The PCA uses a matrix of numerical values as an input, denoted by N rows (objects) and K columns (variables).

Based on this input, PCA performs eigenvalue decomposition over the covariance matrix. The eigenvector matrix is then used to project the samples into a new subspace, denoted by l vectors named principal components (PCs), where l is any number less than the original dimension, K. The PCA also estimates how much variance of the data is captured by its PCs, using Q2 that represents the residual variance within the data left out of the model.

The disadvantage of PCA is that it is hard to interpret, and in many instances, it requires manual fine-tuning. For instance, PCA is greatly affected by the type of scaling used for the analysis and can yield completely different results using different scaling parameters. In general, PCA does not perform well with imbalanced datasets, as these can introduce artifacts in the analysis.

For our study, we use the Scikit-learn implementation of PCA. The parameters of the algorithm are as follows:

- **n_components:** Number of segments to keep. We used 5, 10, and 15.

Non-Negative Matrix Factorization
Non-negative matrix factorization (NMF) is an unsupervised method employing matrix factorization, that is, decomposition of a matrix into two or more matrices in which all values are equal to or greater than zero. The calculations behind NMF are similar to PCA but aim at explaining all of the contents in the data matrix, not just their correlations. However, when PCA is used without the non-negativity constraints, the two methods are equal in performance. The NMF is particularly useful when dealing with
real-world datasets that are often non-negative. For example, user interaction data and user characteristics are almost always equal or greater than zero. The NMF, similar to PCA, has challenges with using imbalanced data or data with a large number of features.77

For our study, we use the Scikit-learn implementation of NMF.*** The parameters of the algorithm are as follows:

- **n_components**: Number of segments to keep. We used 5, 10, and 15.

**Latent Dirichlet Allocation**

Latent Dirichlet allocation (LDA) is an unsupervised probabilistic topic model applied to process large amounts of digital data. The LDA applies a generative probabilistic model commonly used for inferring latent patterns (topics), typically from text. When processing a dataset, LDA returns a vector representing each instance’s probability associated with each latent component. The LDA deals particularly well with imbalanced datasets, and the re-balancing of datasets has been explored, concluding that imbalanced data do not affect LDA performance.78

For our study, we use the Scikit-learn implementation of LDA.***††† The parameters of the algorithm are as follows:

- **n_components**: Number of segments to keep. We used 5, 10, and 15.

**Spectral Embedding**

Spectral Embedding (SE) uses a similarity graph to perform non-linear DR.79 It forms an affinity matrix, and using spectral decomposition, returns a transformation based on the eigenvectors for each data point. The SE is particularly sensitive to imbalanced datasets, which can lead to poor performance on graphs.80

For our study, we use the Scikit-learn implementation of SE.††† The parameters of the algorithm are as follows:

- **n_components**: Number of segments to keep. We used 5, 10, and 15.
- **affinity**: There are 5 types to construct an affinity matrix. We use the “nearest_neighbors” that computes a nearest neighbors graph and constructs the affinity matrix. Other options are “Rbf” (uses Radial Basis function to construct affinity matrix), “Precomputed” (interprets X as a precomputed affinity matrix), and “precomputed_nearest_neighbors” (first interprets X as a sparse graph of precomputed nearest neighbors and then constructs the affinity matrix by selecting the nearest neighbors).

**Uniform Manifold Approximation and Projection**

Uniform Manifold Approximation and Projection (UMAP) is a non-linear DR technique.81 It is very close to t-distributed stochastic neighbor embedding, developed for effective two or three-dimensional data visualizations.82 The UMAP, similar to SE, is sensitive to imbalanced datasets. This is because both SE and UMAP use the n neighbors parameter for locating the amount of nearest neighbors used to construct the high-dimensional graph. Setting the value of n neighbors is a balancing act between preserving local versus global structure in the data, fine-grained versus large-scale features.81 A low value will push UMAP toward focusing on the local structure, whereas a high value will preserve global structure in the projection, but losing the fine detail.

For our study, we use the Umap-learn library.¶¶¶ The parameters of the algorithm are as follows:

- **n_components**: Number of segments to keep. We used 5, 10, and 15.

**Appendix A2. Consistency Score: General Case**

Let us consider the general case of CS, where both the number of the persona sets and the number of personas in each set are hyperparameters.

For example, in the case, we have four sets of personas, we need to choose two out of four for each comparison of consistency. Thus, there are \( \binom{4}{2} = \frac{4!}{2!} = 6 \) combinations. We will have six CS values, and then we take the average to compute the final CS of an algorithm.

---


¶¶¶https://github.com/lmcinnes/umap

(Appendix continues →)
In the case we have $M$ sets of personas, we need to choose two in $M$ for each time for comparison of consistency. Thus, there are $\binom{M}{2}$ combinations $= \frac{M!}{2!(M-2)!} = \frac{M(M-1)}{2}$ combinations. We will have $\frac{M(M-1)}{2}$ $CS$ values and then we take their average.

Suppose that we have $m$ set of personas denoted $M_1$ to $M_m$. $n_{M_1}$ is the number of personas in set $M_1$, $n_{M_2}$ is the number of personas in set $M_2$, and ... $n_{M_m}$ is the number of personas in set $M_m$.

Without loss of generality, we set $n_{M_1} < n_{M_2} < \ldots < n_{M_m}$.

Then, the $CS$ is calculated as:

$$\frac{1}{M} \sum_{i=1}^{m-1} \left( \sum_{k=i+1}^{m} \frac{\sum_{j=1}^{n_{M_k}} P_{M_k} \in \{P_{M_k}\}}{n_{M_k}} \right)$$

$$= \frac{1}{M \cdot (M-1)/2} \sum_{i=1}^{m-1} \left( \sum_{k=i+1}^{m} \frac{\sum_{j=1}^{n_{M_k}} P_{M_k} \in \{P_{M_k}\}}{n_{M_k}} \right)$$

where $P_{M_j}$ denotes the persona $j$ of set $M_j$ and $\{P_{M_k}\}$ denote all the personas of set $M_k$. 