OKCupid - Date-a-Scientist
ML portfolio project

Dr Delphine Vincent
(data scientist)

Google Scholar ORCID LinkedIn GitHub Tableau Public Website email

1/ Project description

This is a portfolio project for **CodeCademy Data Scientist / Machine Learning Specialization**.

In recent years, there has been a massive rise in the usage of dating apps to find love. Many of these apps use sophisticated data science techniques to recommend possible matches to users and to optimize the user experience. These apps give us access to a wealth of information that we've never had before about how different people experience romance.

I've decided to develop an unsupervised machine learning (ML) model that would identify individuals with similar features so that the app can suggest relevant matches. The results will then be displayed via an interactive interface.

2/ Experimental design

Context:

The dating app dataset provided by CodeCademy doesn't include a single feature to predict individual closeness. I must resort to unsupervised methods to structure the data and then find close matches.

Rationale:

Using exploratory data analysis (EDA) to inspect the data, Natural Language Processing (NLP) to process textual features, and machine learning (ML) models eliminate noise and identify hidden patterns to group individuals based on feature similarities, and ultimately determine potential matches between grouped individuals.

Tools:

python, python pandas, python NumPy, python scikit-learn, python NLTK, python wordcloud, python GENSIM, python spaCY, python SciPy, python UMAP, python squarify, python RegEx, python seaborn, python matplotlib, python gradio, Jupyter Notebook, Anaconda Navigator, MS Excel, Tableau Public, GitHub

3/ Data preparation

Steps:

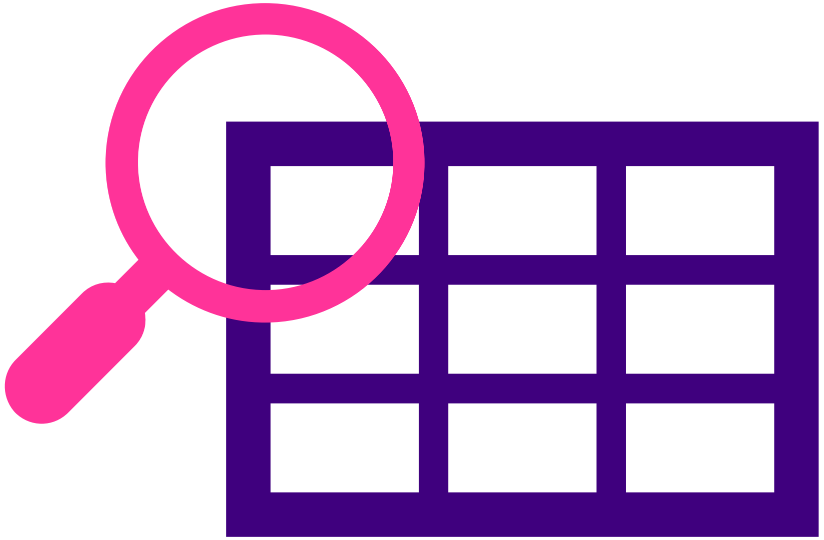
1. Loading and inspecting data
2. Filtering/cleaning/transforming/encoding features for EDA, as well as NLP and ML models

3.1/ Data Inspection

Following loading of the data, we must inspect it to determine dataset size, data types, unique labels, occurrences of missing values (NAs), outliers, aberrant values, etc…

Steps:

1. Dataset overview
2. Feature summary



3.1.1/ Data Inspection - Overview

The data file has 59946 rows (individuals) and 31 columns (features); it doesn't contain an index.

The dataset consists of three main types of features: numerical, categorical, and textual.

- **Numerical Features**: These represent continuous numerical values, such as age, height, and income.
- **Categorical Features**: These represent discrete categories, including both nominal (e.g., gender, language, ethnicity, location) and ordinal (e.g., drinking and smoking habits, morphology).
- **Textual Features**: These are open-ended responses where users describe various aspects of their life, preferences, and expectations. These include the essay fields (essay0–essay9), which contain free-text descriptions of users' summaries, lifestyles, qualities, and thoughts. Unlike categorical features, which have a fixed set of possible values, textual features exhibit high variability in content and require natural language processing (NLP) techniques for analysis.

There are 59946 missing values in total; some variables have none (e.g., age, status, sex, income...), some have a few NAs (height 3, speaks 50), while others have many (e.g. offsprings contains almost 60% of NAs).

3.1.2/ Data Inspection – Feature summary

Features provided capture demographics, lifestyle, and preferences. The table below summarises the variables by type, subtype, labels, missing values (NAs) and outliers.

Name	Description	Type	Subtype	Unique Labels	NAs	Outliers
body_type	Individual's body type	Categorical	Ordinal	12	5296	
diet	Individual's diet (with strictness scale)	Categorical	Nominal(Ordinal)	18	24396	
drinks	Individual's alcohol intake	Categorical	Ordinal	6	2985	
drugs	Individual's drug intake	Categorical	Ordinal	3	14080	
education	Individual's highest education (with achievement scale)	Categorical	Ordinal(Ordinal)	32	6628	
ethnicity	Individual's ethnicity	Categorical	Nominal	217	5680	
job	Individual's job	Categorical	Nominal	21	8198	
location	Individual's location (town, state)	Categorical	Nominal	199	0	
offspring	Parental status and desire for kids	Categorical	Ordinal	15	35561	
orientation	Individual's sexual orientation	Categorical	Nominal	3	0	
pets	Individual's pet preferences (for dogs and cats)	Categorical	Nominal/Ordinal	15	19921	
religion	Individual's religion (with relevancy scale)	Categorical	Nominal(Ordinal)	45	20226	
sex	Individual's gender	Categorical	Binary	2	0	
sign	Individual's astrological sign (with relevancy scale)	Categorical	Nominal(Ordinal)	48	11056	
smokes	Individual's smoking status	Categorical	Ordinal	5	5512	
speaks	Languages spoken (with fluency scale, may include multiple)	Categorical	Nominal	7647	50	
status	Individual's relationship status	Categorical	Nominal	5	0	
last_online	Individual's last login datetime	Datetime	yyyy-mm-dd-hh-mm	30123	0	
age	Individual's age	Numerical	Continuous	N/A	0	> 100
height	Individual's height	Numerical	Continuous	N/A	3	< 21 inches
income	Individual's income	Numerical	Continuous	N/A	0	-1, >500K
essay0	Individual's summary	Textual	Natural Language	54350	5488	
essay1	Individual's life	Textual	Natural Language	51516	7572	
essay2	Individual's qualities	Textual	Natural Language	48635	9638	
essay3	Individual's obvious traits	Textual	Natural Language	43533	11476	
essay4	Favorite books, movies, shows, music, and food	Textual	Natural Language	49260	10537	
essay5	Six most important things	Textual	Natural Language	48963	10850	
essay6	Main thoughts	Textual	Natural Language	43603	13771	
essay7	Friday night habits	Textual	Natural Language	45554	12451	
essay8	Most private admissions	Textual	Natural Language	39324	19225	
essay9	Expectations from others	Textual	Natural Language	45443	12603	

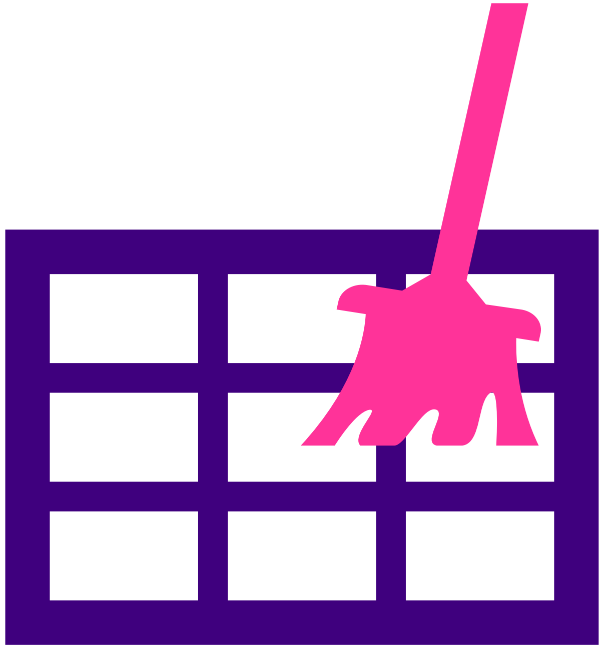
3.2/ Data wrangling

Data filtering, cleaning, transforming and encoding depends on the data type.

Data types:

1. *Numerical variables*: clean and transform if skewed
2. *Categorical variables*: clean and encode
3. *Textual variables*: tokenise, lemmatize, categorise, and encode
4. *Date variable*: not used (not informative enough)

Note: index added (**profile_id**) to keep track
of individuals following ML modeling.



3.2.1/ Numerical features

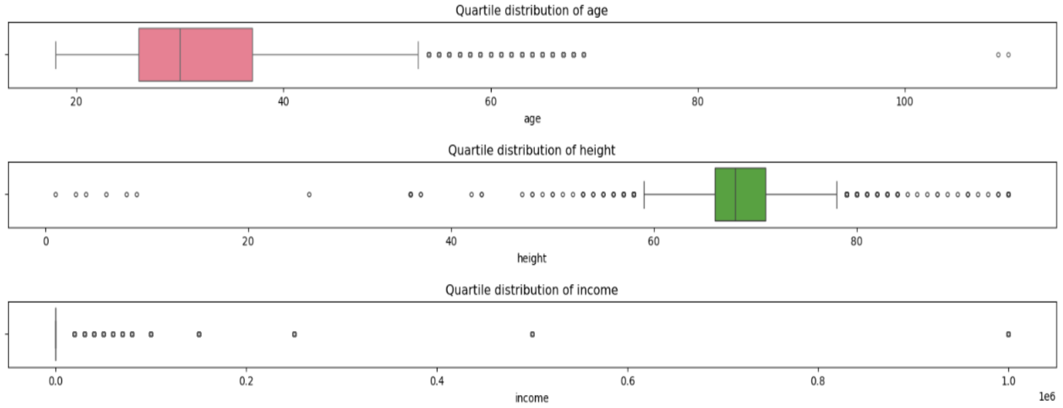
There are 3 numerical variables:

- **age**: min=18, max=110, Q3=37, 0 NA. Right skew. Continuous age range to 70, with 2 outliers at 110, right skew. Filtering out age>80 and log-transformation.
- **height**: min=1, max=95, Q1=66, 3 NAs. Left skew. Most data points are > 20 inches, with 6 outliers. Filtering out height<20 which takes care of the left skew. No need to transform.
- **income**: min=-1, max=1 million, Q3=-1, 0 NAs. Severe right skew. Most data points are $-1 which is not valid. High income values whilst valid don't make sense for most jobs (e.g. students, unemployed). Filtering 0<income>=500,000. Cube-root-transformation. **There is a positive relationship between income and job; I'll use the income median to encode job.**

Raw data

'Summary statistics of numerical variables:'

	age	height	income
count	59946.000000	59943.000000	59946.000000
mean	32.340290	68.295281	20033.222534
std	9.452779	3.994803	97346.192104
min	18.000000	1.000000	-1.000000
25%	26.000000	66.000000	-1.000000
50%	30.000000	68.000000	-1.000000
75%	37.000000	71.000000	-1.000000
max	110.000000	95.000000	1000000.000000

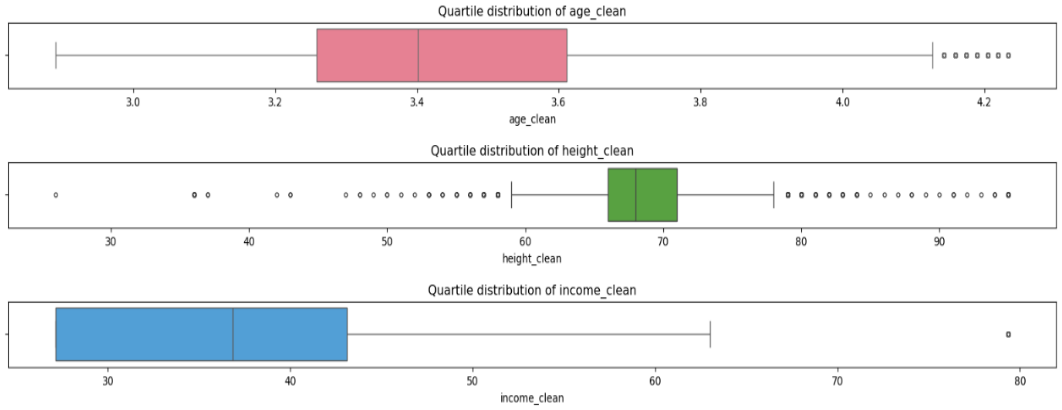


Cleaned / transformed data

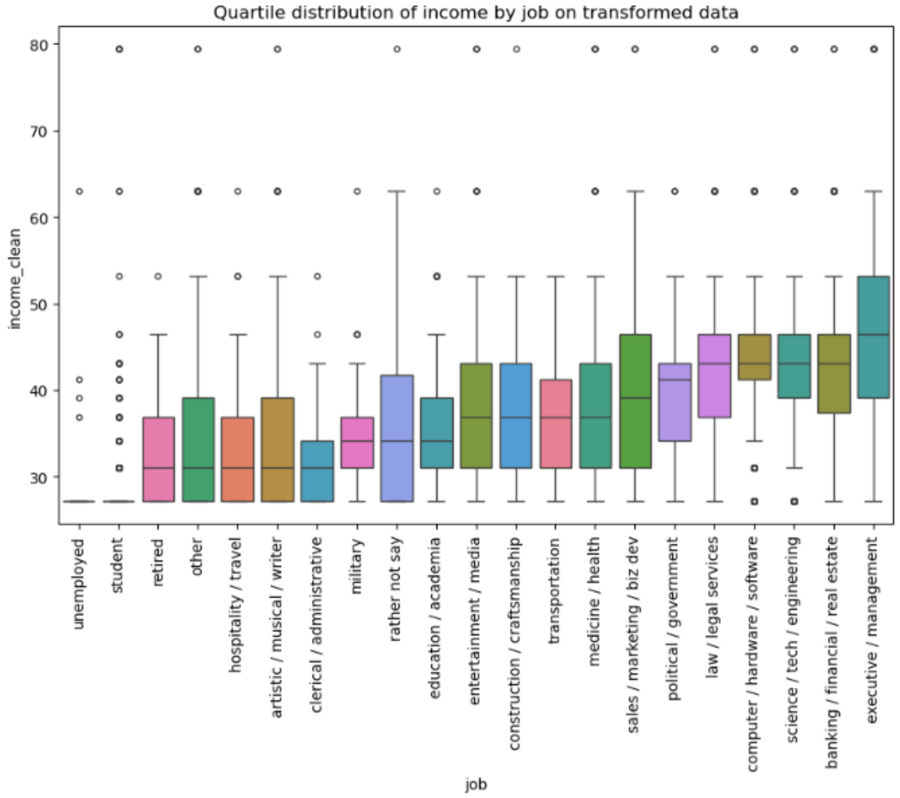
age <80
LOG10

height >20

0 < income <250k
cube-root



Income per job



3.2.2/ Ordinal and nominal features

There are 17 categorical variables (not including “essay”). Features with a main type and a scale (e.g. religion, sign) are split. Categorical features were wrangled and encoded as follows:

Mere encoding:

- **body_type:** ignore "NAN" and scale the categories from -5 to 5 (fit to overweight) → **body_type_level**
- **diet:** group by main diet categories (**diet_code**) and add level of strictness (**diet_level**)
- **drinks:** ignore "NAN" and scale the categories from 0-5 (**drinks_level**)
- **drugs:** ignore "NAN" and scale the categories from 0-2 (**drugs_level**)
- **ethnicity:** lots of ethnic categories but only a few that are prominent, agglomerate the minor categories as others → **ethnicity_code**
- **job:** group by main job categories compute the median income by job, and use it as a level → **job_code**
- **orientation:** scale from 1-3 → **orientation_code**
- **sex:** convert to binary 0/1 → **sex_code**
- **smokes:** ignore "NAN" and scale the categories from 0 to 4 → **smokes_code**
- **status:** exclude unknown and create ordinal variable → **status_code**

Splitting and encoding:

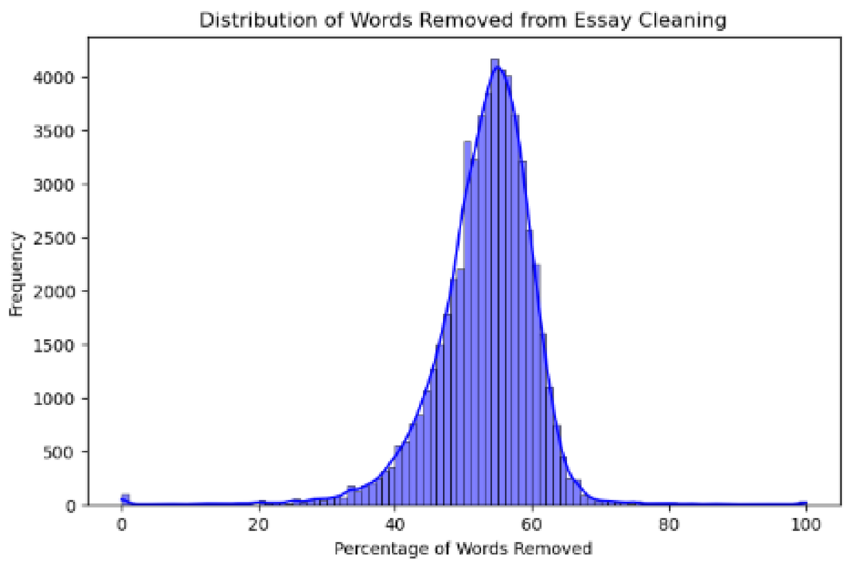
- **education:** group by highest level of education reached (**education_code**) and add whether graduated/working on/dropout (**education_level**)
- **location:** lots of location categories but only a few that are prominent, agglomerating by state shows that 99% is in CA and mostly in the San Francisco area. Subset country (**location_country_code**), state (**location_state_code**), and town (**location_town_code**)
- **offspring:** ignore "NAN" and subdivide as **has_offspring** from 0 to 2, and **wants_offspring** from 0 to 1
- **pets:** subdivide by dogs (**pets_dog**) and cats (**pets_cat**) encoding levels of affinity (0 to 2)
- **religion:** ignore "NAN" and group by main faith types (**religion_code**) and add a practice scale 1 to 4 (**religion_level**)
- **sign:** ignore "NAN" and group by main sign categories (**sign_code**) and add a relevance scale(**sign_level**)
- **speaks:** establish frequency of language and use it as an encoder, sum all encoded languages listed as a proxy for multilingual ability (**speaks_counter**), consign C++ skills to **speaks_program**

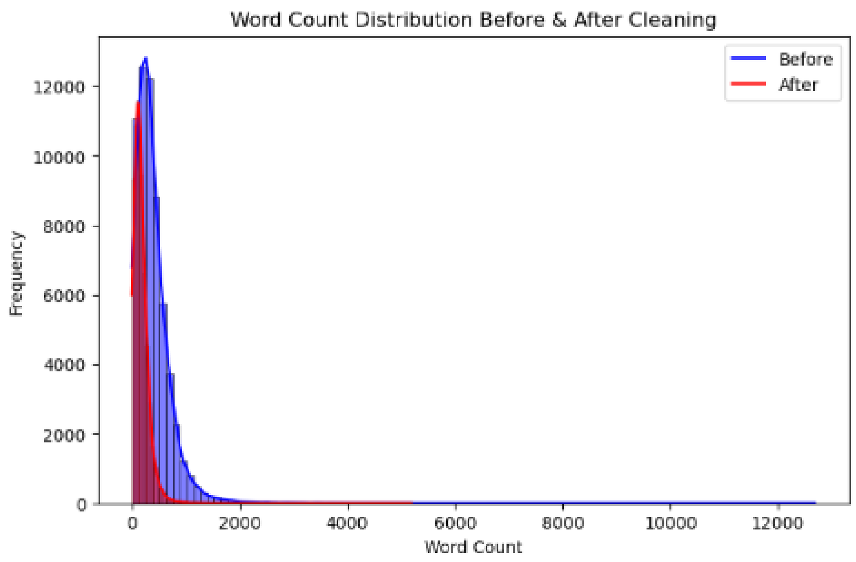
3.2.3/ Textual features

There are 10 text variables:

Name	Description	Type	Subtype	Unique Labels	NAs
essay0	Individual's summary	Textual	Natural Language	54350	5488
essay1	Individual's life	Textual	Natural Language	51516	7572
essay2	Individual's qualities	Textual	Natural Language	48635	9638
essay3	Individual's obvious traits	Textual	Natural Language	43533	11476
essay4	Favorite books, movies, shows, music, and food	Textual	Natural Language	49260	10537
essay5	Six most important things	Textual	Natural Language	48963	10850
essay6	Main thoughts	Textual	Natural Language	43603	13771
essay7	Friday night habits	Textual	Natural Language	45554	12451
essay8	Most private admissions	Textual	Natural Language	39324	19225
essay9	Expectations from others	Textual	Natural Language	45443	12603

For simplicity, all 10 "essay' features are concatenated into one column (**essay_combined**).
HTML tags, punctuation, extra spaces, line breaks, numbers and common words (stopwords) are removed.
The remaining words are tokenised and lemmatized and consigned to a new variable (**essay_clean**) which lists only meaningful words that will be used for topic extraction.





Total words before cleaning:
22,808,851
Total words after cleaning:
10,360,533
Proportion of words retained:
45%
Proportion of words eliminated:
55%

4/ Exploratory Data Analysis (EDA)

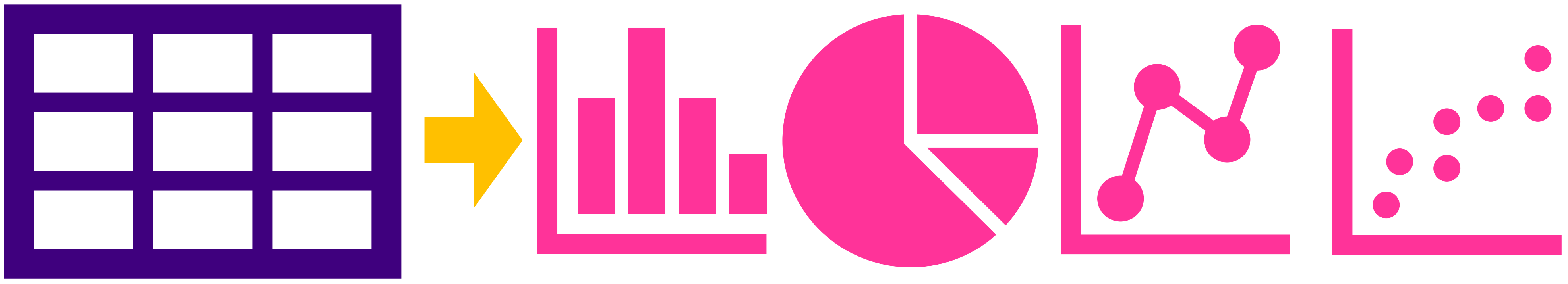
Steps:

1. Distribution of clean features reveals what types of individuals register their interest with Okcupid dating app
2. Correlation analysis informs on the relationships between encoded features

4.1/ EDA – summarizing the data

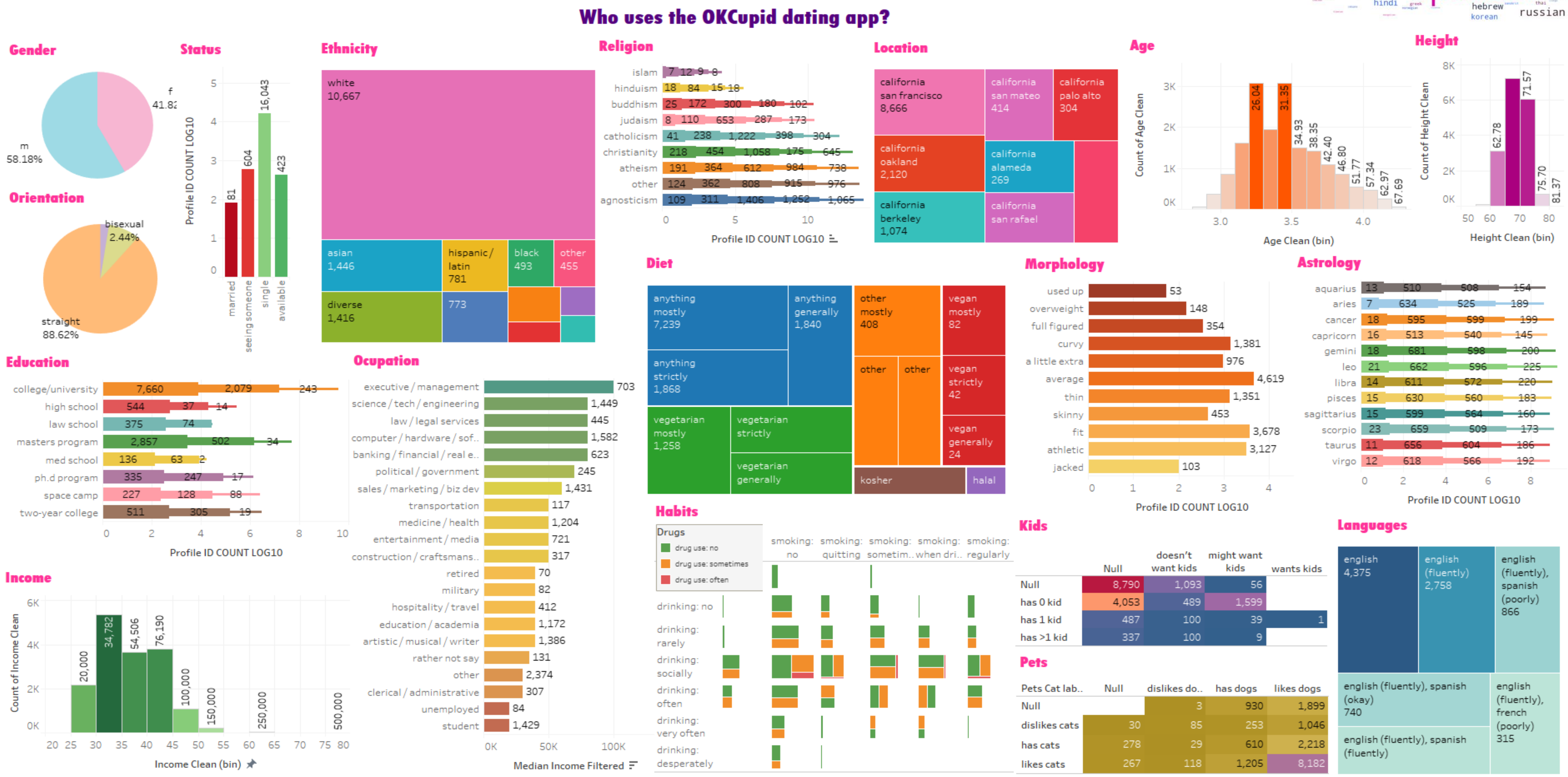
Steps:

1. Plotting the data distributions as histograms, pie charts, treemaps, wordclouds, heatmaps and bar plots helps summarise the features and reveal the main trends (data, charts and dashboard available from Tableau Public)
2. From those main trends can be inferred which types of individuals (archetypes) are interested in using the dating app



4.1.1/ EDA – Variable distributions

Tableau Public dashboard allows for a juxtaposition of all relevant EDA charts (link).



4.1.2/ EDA – User Archetype

Demographic Information

•**Location** → A **treemap** showing the most common locations of users, highlighting **California cities** such as **San Francisco, Palo Alto, and Berkeley**.
•**Languages** → A **treemap** highlighting the most common **languages spoken with fluency scaling**, and multilingual abilities. *Note*: A **wordcloud** better shows the main languages spoken, namely: **English, Spanish, French, Chinese, German, Italian, and Japanese**.
•**Ethnicity** → A **treemap** visualizing ethnic distribution, showing **white (10,667)** as the largest group, followed by **Asian, Hispanic/Latino, and diverse**.
•**Religion** → A **bar chart** depicting the religious affiliations of users, with **Atheism and Agnosticism**, being the most prominent.

Physical Attributes

•**Gender** → A **pie chart** showing the proportion of males (58.18%) and females (41.82%) in the dataset.
•**Age** → A **histogram** displaying the user **age distribution**, which is bimodal peaking at 26 and 31 years-old.
•**Height** → A **bar chart** showing the distribution of **height** among users, peaking at 67 inches.
•**Morphology** → A **bar chart** describing the user's body type, mostly featuring between from **athletic** to **curvy**.

Education, Occupation & Income

•**Education** → A **bar chart** displaying educational background, showing the highest proportion of users have a **college/university degree**, followed by **law school, master's programs,** and **two-year college**.
•**Occupation** → A **horizontal bar chart** listing various **job categories** ranked by median income. Best paying jobs are **executive/management, science/tech/engineering, law/legal services, computer/hardware/software,** as well as **banking/financial/real estate**.
•**Income** → A **histogram** visualizing the income distribution, with most users earning around **$20,000 to $100,000**.

Lifestyle & Personal Preferences

•**Orientation** → A pie chart displaying sexual orientation, with the majority being **straight (88.6%),** a small percentage identifying as **gay (9%)**, and the rest as **bisexual (2.4%)**.
•**Habits** → A **stacked bar chart** categorizing users based on **smoking, drinking,** and **drug use habits**. Many individuals drink socially, occasionally smoking and using drugs.
•**Diet** → A **treemap** classifying users by diet preference and exclusivity scaling. Most individuals eat **anything** or are **vegetarian**.
•**Astrology sign** → A **bar chart** displaying the proportion of users under each **zodiac sign** with a relevance scale. The distribution is uniform across signs.

Family life

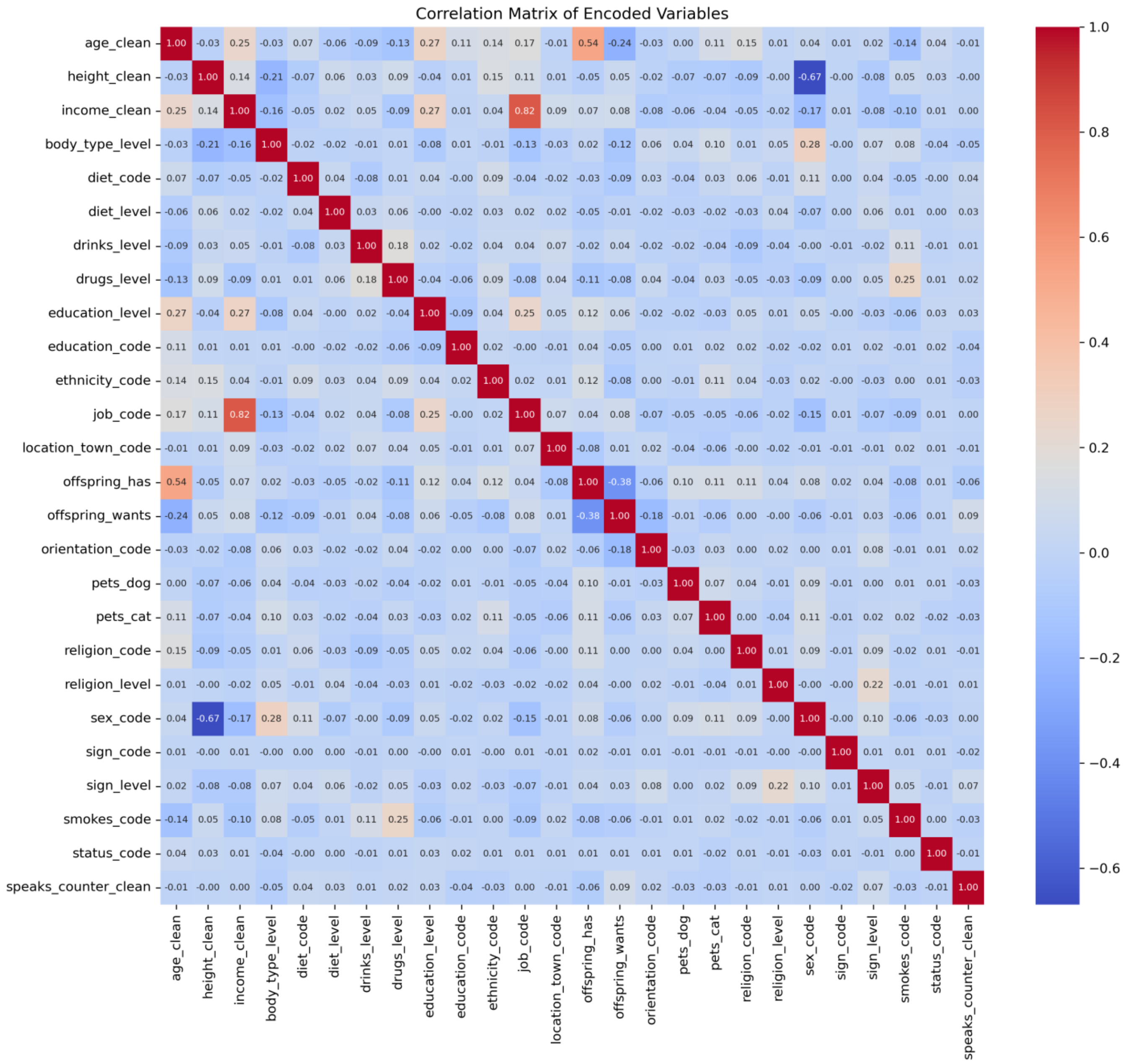
•**Relationship Status** → A bar chart representing relationship status, with categories like **married, single, seeing someone, available, etc.** Most users are **single.**
•**Progeny** → A **heatmap** showing the number of kids per user and whether they **want kids, don't want kids, or are unsure**. Most users have **no kid**.
•**Pets** → A **treemap** displaying pet preferences, including **likes cats, dislikes cats, has dogs, etc.**. Most users **like both dogs and cats**.

4.2/ EDA – Variable correlations

Steps:

1. Computing the correlations between all paired features helps assess the strength and direction of their relationships.
2. Strongly correlated features (whether positively or negatively) can provide valuable insights for feature selection, redundancy reduction, and missing value imputation.
3. High correlations may indicate multicollinearity, which can affect certain machine learning models, while weak or no correlation suggests independent variables.
4. Understanding these relationships helps refine data preprocessing and improve model interpretability.

4.2.1/ Correlations – Matrix



Strongest Positive Correlations (Red)

1. **offspring_has & age_clean (0.56)** → Younger individuals are less likely to have children
2. **income_clean & job_code (0.53)** → certain occupations are associated with higher income
3. **age_clean & income_clean (0.39)** → the older, the greater the wealth
4. **smokes_code & drugs_level (0.35)** → people who smoke are more likely to use drugs
5. **income_clean & education_level (0.29)** → Higher education achived correlates with higher income
6. **offspring_has & offspring_wants (0.29)** → Those with children are more likely to want more children
7. **religion_level & sign_level (0.26)** → highly spiritual people also believe in astrology

Strongest Negative Correlations (Blue)

1. **Sex_Code & Height_Clean (-0.66)** → Males are taller than women
2. **age_clean & offspring_wants (-0.22)** → Younger person don't want children
3. **body_type_level & height_clean (-0.21)** → shorter people tend to be less fit
4. **smokes_code & income_clean (-0.20)** → people who smoke earn less money

Weak Correlations

1. **smokes_code & age_clean (-0.17)** → smokers are younger
2. **smokes_code & drinks_level (0.16)** → people who smoke are more likely to drink alcohol
3. **sex_sode & job_code(-0.15)** → gender differences in occupation

NOTE: none of those features are strongly correlated ($R2 > 0.8$) and therefore are not collinear

4.2.2/ Correlations – Correlated Features

The strongest correlations (R2>|0.5|) aid missing value (NAs) imputations.

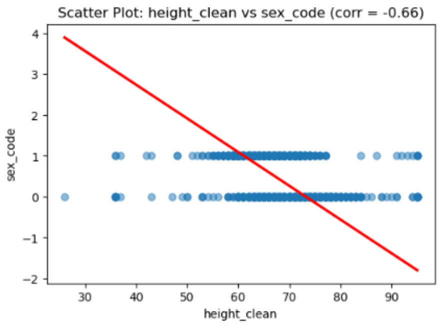
Gender & Height: Women are generally smaller than men. Normal distribution per sex. No missing values; otherwise, the average height per sex could be used to impute missing values.

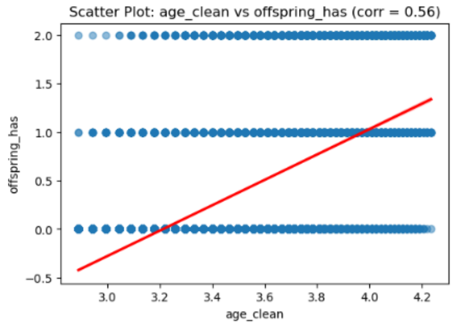
Age & Children: Older individuals have more kids than younger ones. Normal distributions with little overlap for no kid (0) or kid(s) (1 or 2). offspring_has Nas (59%) can be imputed based on an age_clean threshold (age_clean<3.6→no kid, age_clean>=3.6→ kid(s)).

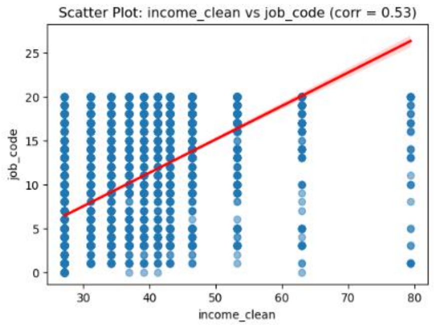
Job & Income: Some occupations pay more than others. Income median is used to impute income missing or invalid (-1) values.

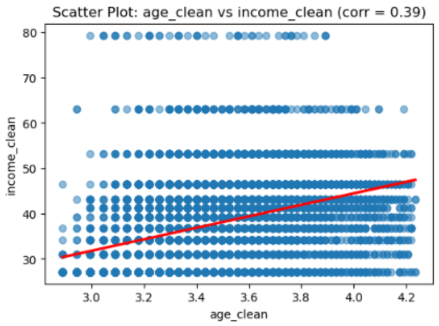
Age & Income: Older individuals earn more money. Normal distributions with little overlap for low incomes. Weaker relationship than job/income but could have been used for income NAs imputation as well.

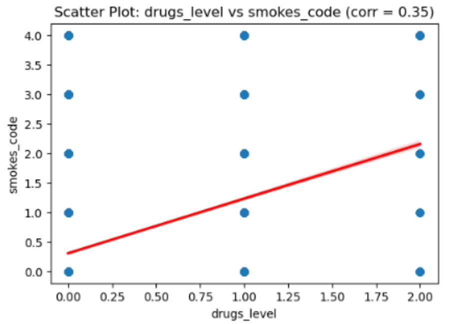
Smoking & Drug use: Smokers also use drugs. Relationship is not strong enough to impute missing values.

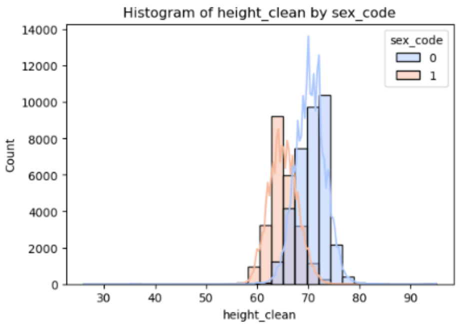


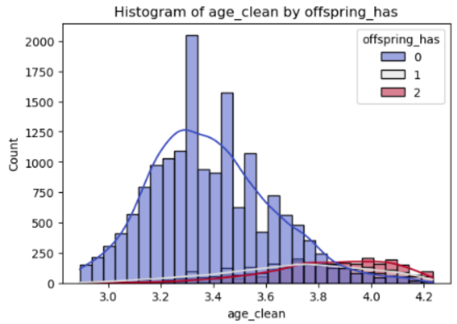


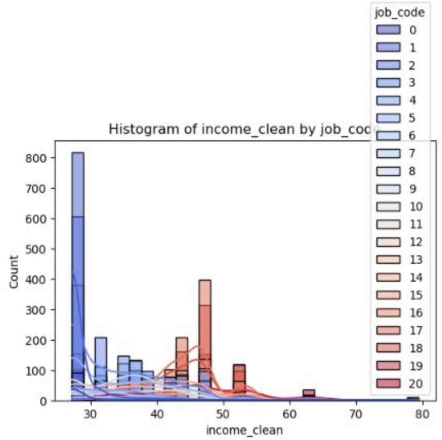


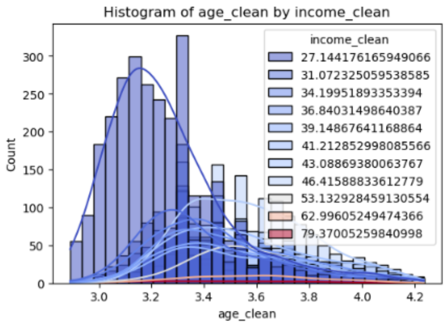


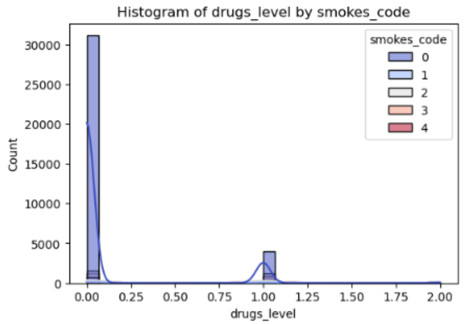








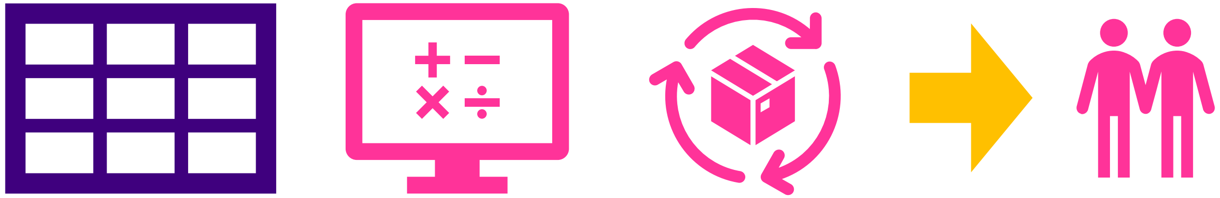




5/ Unsupervised ML

Steps:

1. Classification of lemmatized essay using LDA
2. Weighting of features and row selection to minimise NAs
3. Identification and elimination of outliers to reduce noise using Isolation Forest
4. Clustering of individuals to decomplexify data using HDBSCAN
5. Pair-matching of individuals within clusters using Cosine similarity score



5.1/ Classification of essay_clean

Method:

Latent Dirichlet Allocation (LDA) for Topic Modeling is a generative probabilistic model used to discover hidden topics in a collection of text documents. It extracts underlying themes from user essays on the dating app. **Essay_clean** variable results from tokenization, stopword removal, and lemmatization so that Only meaningful words remain for topic extraction.

Strategy:

"**essay_clean**" is assumed to be composed of multiple **topics** in different proportions. Each topic is represented by a **probability distribution over words** (i.e., some words are more likely in certain topics).

By assuming a Dirichlet distribution for both topic distribution within a variable as well as word distribution within a topic, LDA ensures sparsity. The textual variable is reduced to only a few dominant topics, and each topic is represented by a subset of relevant words.

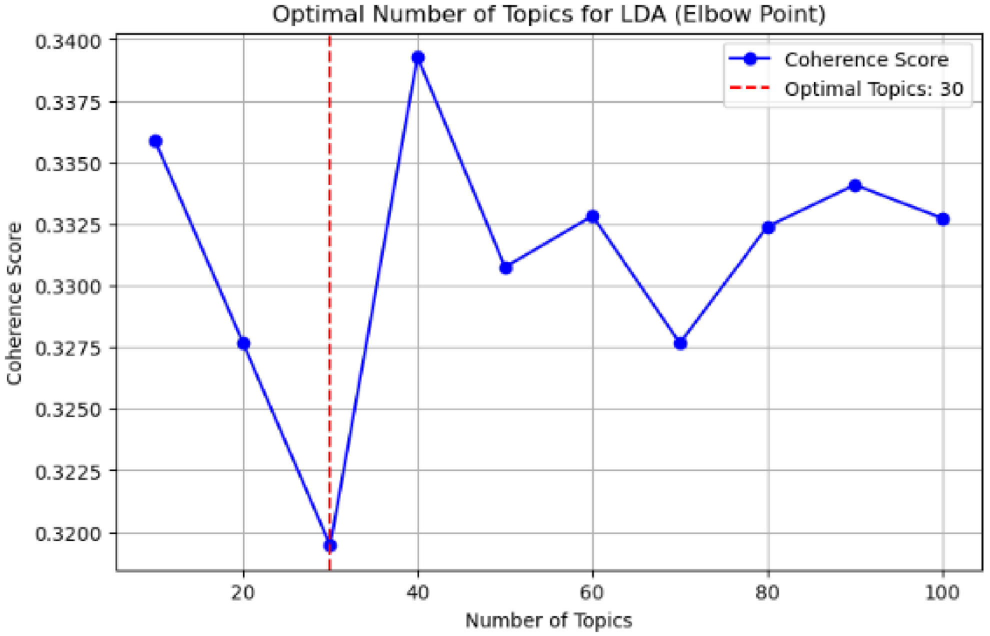
Steps:

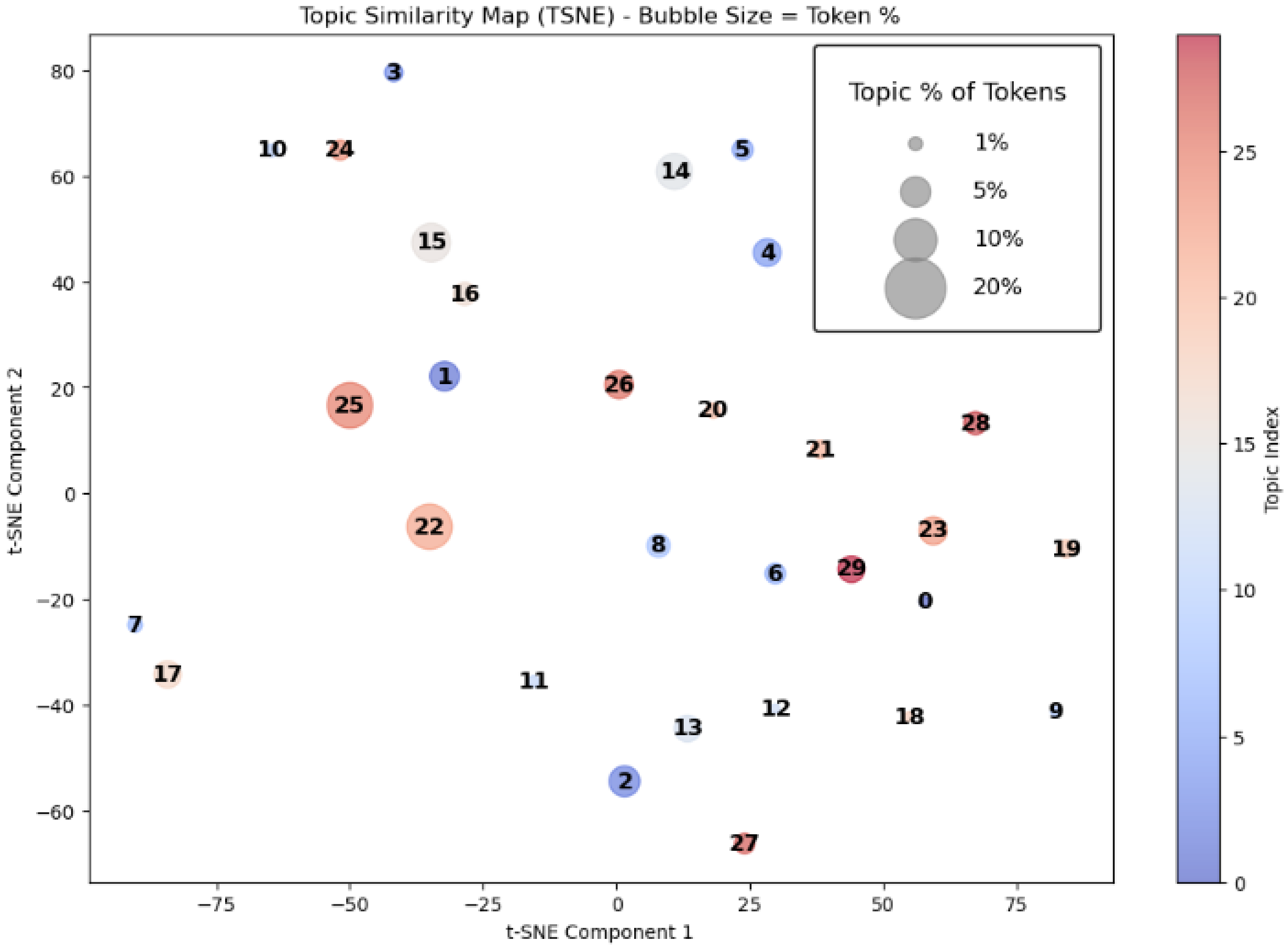
1. **essay_clean** is converted into a bag-of-words (BoW) or Term Frequency-Inverse Document Frequency (TF-IDF) matrix which represents word counts.
2. LDA Model Training assigns each word in "essay_clean" a probabilistic topic and iteratively updates topic distributions using Gibbs Sampling or Variational Inference.
3. The model outputs a set of topics (clusters of words that frequently appear together).

Optimisations:

- Choosing the Number of Topics (k) using coherence scores.
- Assessing Word Relevance in Topics and eliminating most frequent terms

5.1.1/ LDA – k optimisation





Results:

Dictionary created with **26534 unique words**

Optimal Number of Topics: **k = 30**

The plot on the right shows that those 30 topics are well separated meaning distinct from each other, and that their predominance varies.

5.1.2/ LDA – eliminating nondescript terms

Strategy to reduce word list:

1. Filter Out Highly Frequent, Generic Words (Low Discriminative Power)
   - remove top 1% most frequent words.
2. Use TF-IDF Weighting
   - Instead of selecting the most frequent words, we identify words that are unique to each topic.
3. POS (Part-of-Speech) Filtering
   - Keep revealing terms.
   - Remove nondescript words (e.g., "want", "think", "make")
4. Semantic Filtering via Word Embeddings
   - Check word similarity between topic words
   - Keep words that are semantically distinct.

Impact:

On average, 95% of the words were eliminated, maintaining on average around 600 terms.

These words provided great insights to categorise a topic.

For instance, wordclouds show that topic 1 mostly deals with startup, engineer and IT whilst topic 2 is about yoga, healthy habits and spirituality.

Dominant_ Topic	Total word count	Unique word count	unique word count post-TF-IDF filtering	Unique words removed (%)
12	5485	2562	341	87
18	6107	2751	508	82
9	7001	2853	508	82
11	17136	4637	631	86
0	15934	5564	560	90
10	22236	6175	592	90
7	41395	7659	614	92
3	40755	7831	574	93
27	74782	10461	588	94
19	54875	10573	502	95
20	68295	10927	631	94
21	81019	13459	597	96
5	144866	14694	596	96
24	101992	15544	599	96
28	200192	18195	556	97
17	321682	23948	620	97
16	318727	26343	622	98
6	236056	27515	632	98
8	381035	30411	579	98
26	447340	31194	611	98
4	462986	32256	611	98
14	694521	32870	606	98
1	638880	33631	625	98
2	579807	35165	657	98
25	847149	38645	614	98
22	1048613	40470	644	98
13	559200	40857	635	98
23	574235	43689	551	99
29	715371	52682	583	99
15	1652861	73892	636	99

5.1.3/ LDA – Naming topics

Dominant_ Topic	Topic Prevalence (%)	Topic_Label post TF-IDF filtering
25	11.3	Phone / Chill / Hip
22	11.2	Active / Positive / Relax
15	8.1	Answer / Use / Turn
14	7.1	Active / Summer / Hiking
2	5.1	Yoga / Spiritual / Healthy
1	4.7	Startup / Software / Engineer
26	4.3	Wear / Ice / Chocolate
23	4.2	King / Beatle / Dead
4	4.1	Month / Language / Build
17	4.0	Student / Science / Graduate
29	3.9	Cat / Dead / Boy
13	3.8	Create / Passionate / Beautiful
8	2.9	Video / Dead / War
28	2.8	Sunshine / Development / Arrest
16	2.8	Animal / Cat / Weird
5	2.3	Development / Arrest / Mad
27	2.2	Design / Artist / Paint
6	2.2	Queer / Gender / Project
24	2.1	Face / Hate / Money
3	1.8	Money / Single / Smoke
21	1.7	Motorcycle / Bicycle / Fix
19	1.7	Author / Wes / Tom
20	1.2	Dancing / Yoga / Shoe
7	1.1	Genre / Phone / Simple
10	0.8	Mom / Boy / Baby
11	0.7	Site / Consider / Intelligent
0	0.6	Brother / Young / Red
9	0.4	Chocolate / Yoga / Shoe
18	0.4	Mad / Burn / Startup
12	0.4	Soul / Massage / Curious

Assign labels to Dominant_Topic:

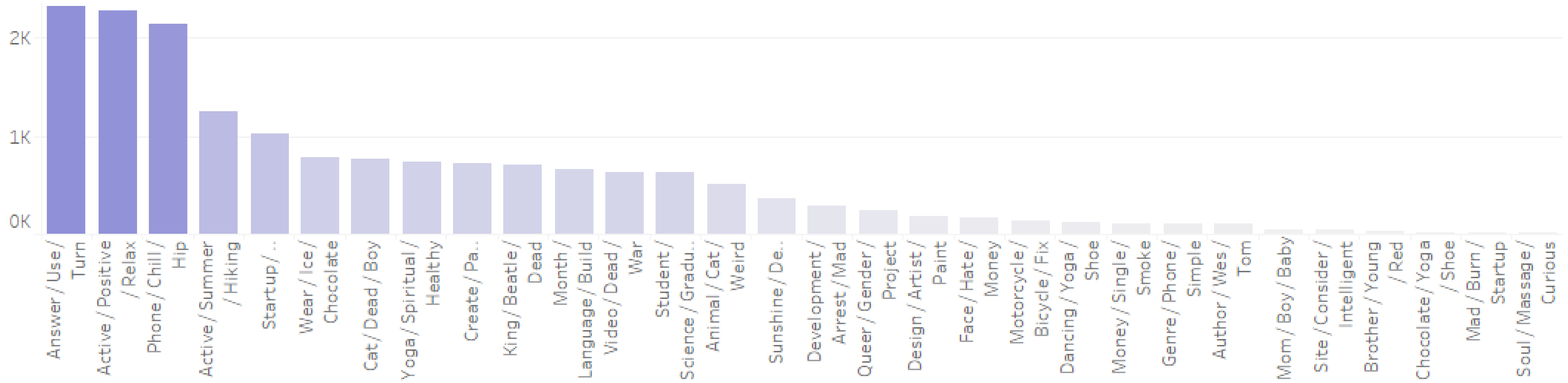
The 3 most frequent words were retrieved and combined to create a meaningful name for Dominant_Topic (**Topic_Label**).
Those names highlights the diversity of Dominant Topics (e.g. "Soul / Massage / Curious", "Motorcycle / Bicycle / Fix", "Design / Artist / Paint").

Computing the prevalence of Dominant_Topic:

A **topic_prevalence** (%) was derived from the topic probabilities assigned to each row. It represents the proportion of the total probability mass assigned to each topic across all row, rather than just the number of words. This means that a topic could have a high probability across essay_clean values with few words or low probability in essay_clean values with many words .
Most prevalent topics are "Lol / Phone / Chill" (11.3%), "Active / Positive / Relax" (11.2%), "Answer / Use / Turn" (8.1%), and "Active / Summer / Hiking" (7.1%).

Distribution of Dominant_Topic:

Topic frequency and prevalences mostly coincide. The 4 ost frequent topics are also the most prevalent ones (see above). Rare topics are "Soul / Massage / Curious", "Mad / Burn / Startup" , and "Chocolate / Yoga / Shoe"; they display the list prevalence.



5.2/ Variable weighting and row selection

Rationale for feature weighting:

Not all features bear the same impact on how should individuals be paired, therefore I have weighted them according to what I assume is important.

Variable weighting is indicated in the table.

For instance, I decided that people residing in the same town would be more likely to be willing meet than if they are far apart. Furthermore, it seemed obvious to me that sexual orientation should match in a dating app.

NOTE: This decision might introduce a bias.

Elimination of rows with too many NAs:

A completeness score is defined based on the frequency of NAs across all features and taking into account their weights. The maximum completeness score was $3 \times 4 + 2 \times 9 = 30$ and only rows achieving ≥ 90% (27) are selected. This ensured that only rows with enough data are considered.

Output:

59896 total rows

38454 (64%) < 90% incomplete rows → filtered out

21442 (36%) >= 90% complete rows → kept for future matching model

NOTE: This decision might also introduce a bias as the majority of rows are eliminated.

Name	Type	Weight
location_town	object	3
orientation_code	int64	3
status_code	int64	3
Dominant_Topic	int64	3
diet_code	float64	2
drinks_level	float64	2
drugs_level	float64	2
offspring_has	float64	2
offspring_wants	float64	2
pets_dog	float64	2
pets_cat	float64	2
religion_code	float64	2
religion_level	float64	2
sign_level	float64	2
smokes_code	float64	2
age_clean	float64	1
height_clean	float64	1
income_clean	float64	1
location_state	object	1
location_country	object	1
body_type_level	float64	1
diet_level	float64	1
education_level	float64	1
education_type	object	1
education_code	int64	1
ethnicity_clean	object	1
ethnicity_code	int64	1
job_code	float64	1
location_town_code	int64	1
sex_code	int64	1
sign_code	float64	1
speaks_program	int64	1
Topic_Label	object	1
hdbscan_clusters	int64	1

5.3/ Outlier detection and filtering using Isolation Forest

Rationale:

Removing outliers to retain most relevant data for future matching model

Method:

IsolationForest algorithm

Principle:

- Detects extreme outliers (= anomalies)
- Converts an unsupervised anomaly detection problem into a supervised classification problem by assigning labels (-1 = inlier, 1 = outlier)

Steps:

1. Optimisation of parameters
2. Application of Isolation Forest algorithm with optimum parameters to detect outliers
3. Evaluate model's performance

5.3.1/ Isolation Forest – parameters optimisation

Isolation Forest Parameters tested:

n_estimators: [50, 100, 200, 400]
contamination: [0.01, 0.05, 0.1, 0.2]
random_state: [42, 2023, None]

Best Isolation Forest Parameters:

bootstrap: False,
contamination: 0.2,
max_features: 1.0,
max_samples: 'auto',
n_estimators: 50,
n_jobs: -1,
random_state: 42,
verbose: 0,
warm_start: False

Isolation Forest Model Performance Using Best Parameters:

Silhouette Score: 0.1104
Accuracy: 0.9063
Precision: 0.8654
Recall: 0.6294
F1 Score: 0.7287

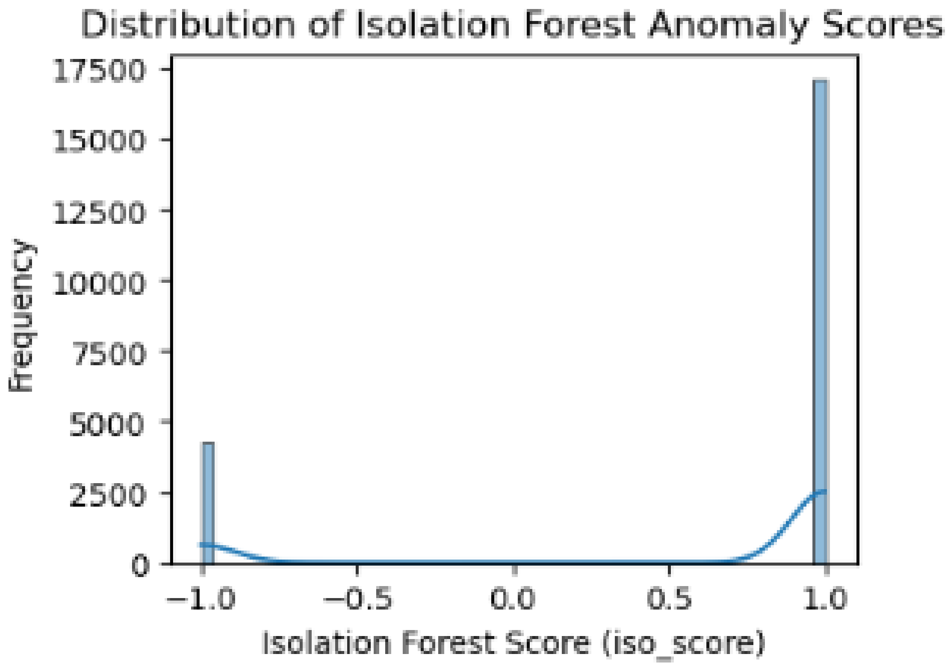
5.3.1/ Isolation Forest – outlier detection

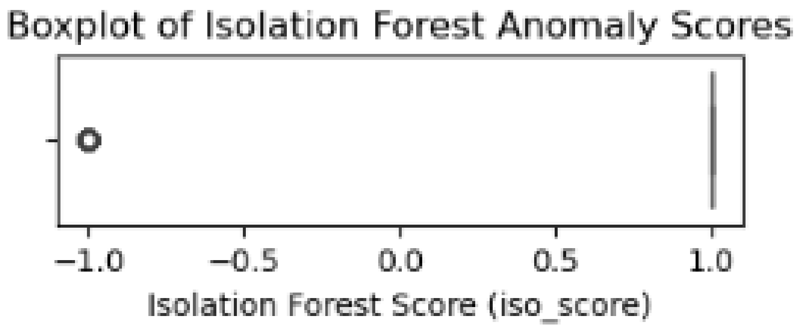
Detection of outliers using Isolation Forest optimized method:

21442 total rows

4289 (20%) outliers (-1) → filtered out

17153 (80%) inliers (1) → kept for future matching model





NOTE: Removing 20% of users could disproportionately affect underrepresented groups.

5.4/ Clustering of rows using HDBSCAN

Rationale:

Discover natural groupings in the dataset to structure and decomplexify data for future matching model

Method:

Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) clustering

Principle:

- Unsupervised algorithm
- Group similar data points together based on density.
- Identify noise (outliers) as separate from clusters, which complements IsolationForest.
- No need to specifying the number of clusters
- Suited to high-dimensional and noisy data.

Steps:

1. Optimisation of parameters
2. Application of clustering algorithm with optimum parameters to detect groups
3. Evaluate model’s performance

5.4.1/ HDBSCAN clustering – parameters optimisation

HDBSCAN Clsutering Parameters tested:

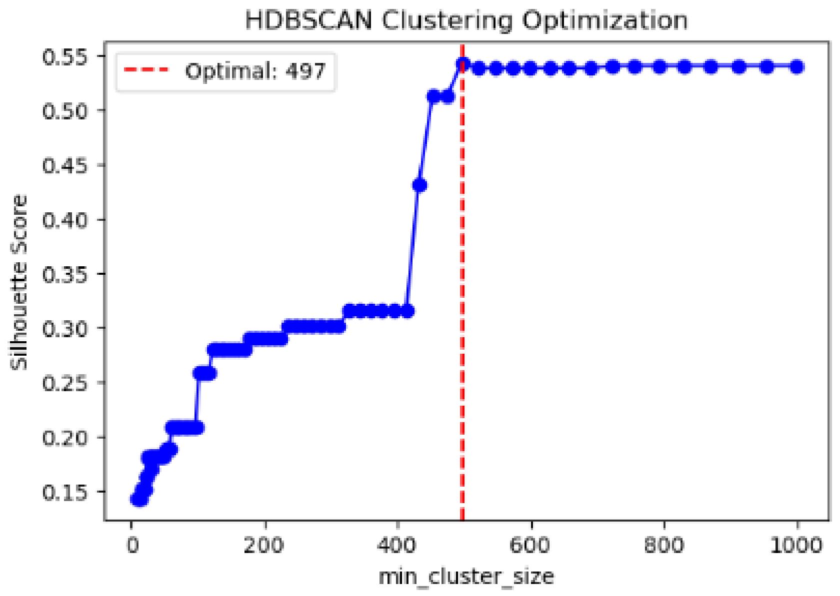
min_cluster_size: range(100, 1001, 9)
min_samples: range(10, 21, 6)
cluster_selection_method: ["eom"]
metric: ["euclidean"]

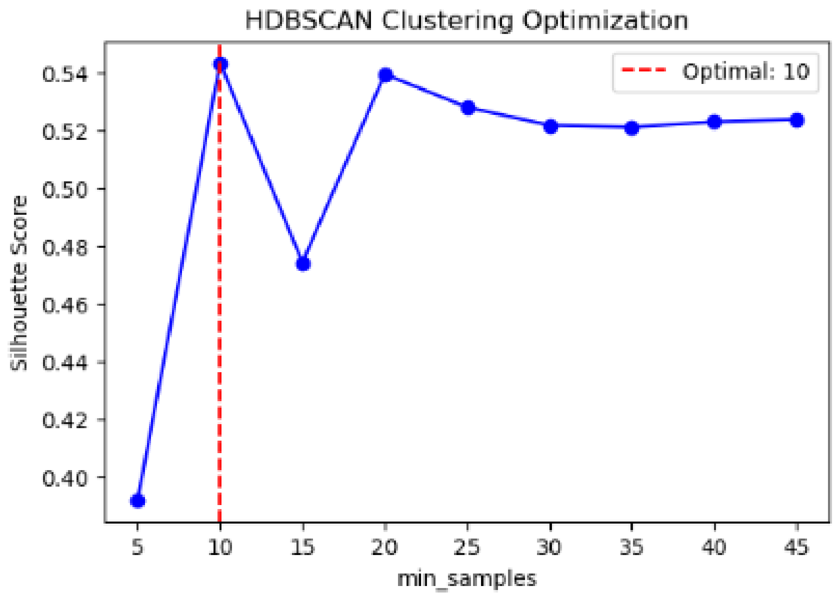
Best HDBSCAN Parameters:

min_cluster_size: 500
min_samples: 10
cluster_selection_method: eom
metric: euclidean
algorithm': 'best'
allow_single_cluster: False
alpha: 1.0
cluster_selection_epsilon: 0.0
leaf_size: 40,
prediction_data: False

HDBSCAN Model Performance Using Best Parameters:

Silhouette Score: 0.54
Davies-Bouldin Index: 1.95
Calinski-Harabasz Index: 42133.1

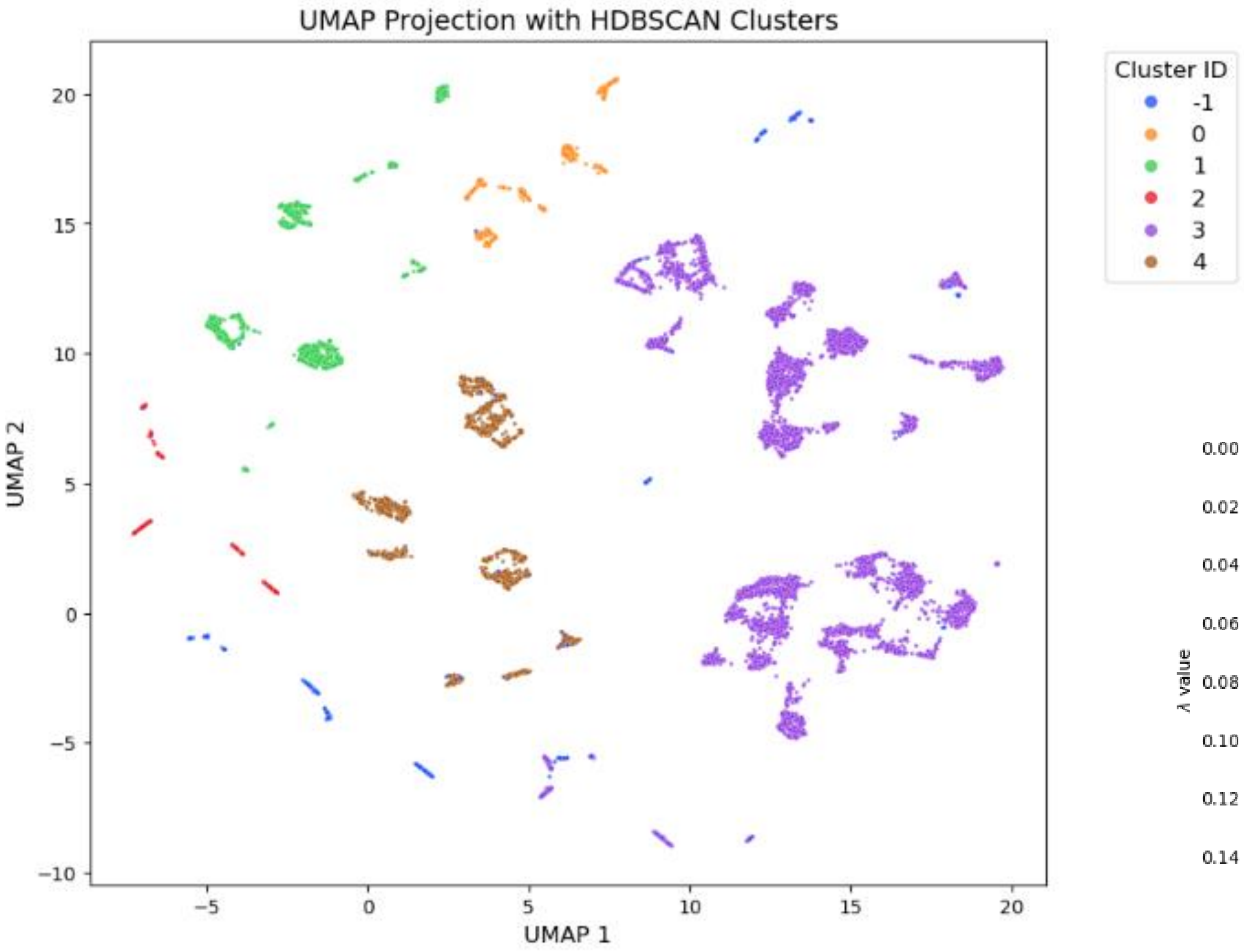




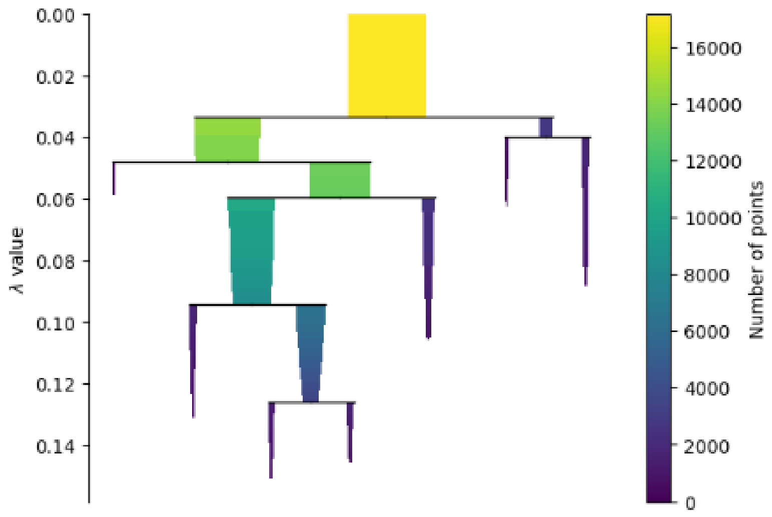
5.4.2/ HDBSCAN clustering – structuring dataset

Detection of groups using HDBSCAN clustering algorithm:

6 clusters are defined of various sizes (cluster 3 with 10292 individuals or cluster 2 with 572 individuals), including outliers (cluster -1 with 862 individuals). I couldn't associate the HDBSCAN clusters to specific features. UMAP projection plot shows greater granularity than 6 groups.



hdbscan_clusters	Individuals (%)
3	10292 (60%)
4	2658 (15%)
1	1834 (11%)
0	935 (6%)
-1 (outliers)	862 (5%)
2	572 (3)



5.5/ Pair matching of individuals using Cosine metrics

Rationale:

Pair individuals based on their similarities with each HDBSCAN cluster.

Method:

Cosine similarity algorithm: suited to numerical and encoded categorical data, simple and efficient as there is no need for labels or interactions; it computes the similarity directly.

Principle:

Cosine similarity calculates the angle between two feature vectors:

- smaller angle → higher similarity (closer match)
- larger angle → lower similarity (less of a match)

Steps:

1. Add a condition about orientation (Straight: Male matches Female and vice versa, Gay: Male matches Male and Female matches Female; Bisexual: male and female can match both male and female)
2. Median impute NAs and normalize the data (so that no single feature dominates).
3. Compute cosine similarity between all individuals in the same HDBSCAN cluster.
4. Recommend the most similar matches based on the highest cosine similarity scores.
5. Evaluate the model's performance

5.5.1/ Cosine pairing – Model performance

Regression Performance (Predicting Cosine Similarity):

- *Train*: $R^2 = 0.9689$ → The model explains 97% of the variance in similarity scores (excellent).
- *Validation*: $R^2 = 0.8496$ → 85% allows for a strong generalization.
- *Test*: $R^2 = 0.8790$ → 88% of the test data variance is explained (very robust).

→ Higher intra-cluster similarity (closer to 1) and lower inter-cluster similarity (closer to 0) → The model is grouping similar people correctly.

Classification Performance (Predicting Good Matches) :

- *Train*: Accuracy=0.9576, Precision=0.9528, Recall=0.9628, F1=0.9578 → Nearly perfect match classification.
- *Validation*: Accuracy=0.8936, Precision=0.8832, Recall=0.9116, F1=0.8972 → Great generalization with a high recall of 90.68% (few missed good matches).
- *Test*: Accuracy=0.8948, Precision=0.8840, Recall=0.9132, F1=0.8984 → Consistently high across unseen data.

→ The model retrieves relevant similar profiles for a given user.

5.5.2/ Cosine pairing – Network analysis

Method:

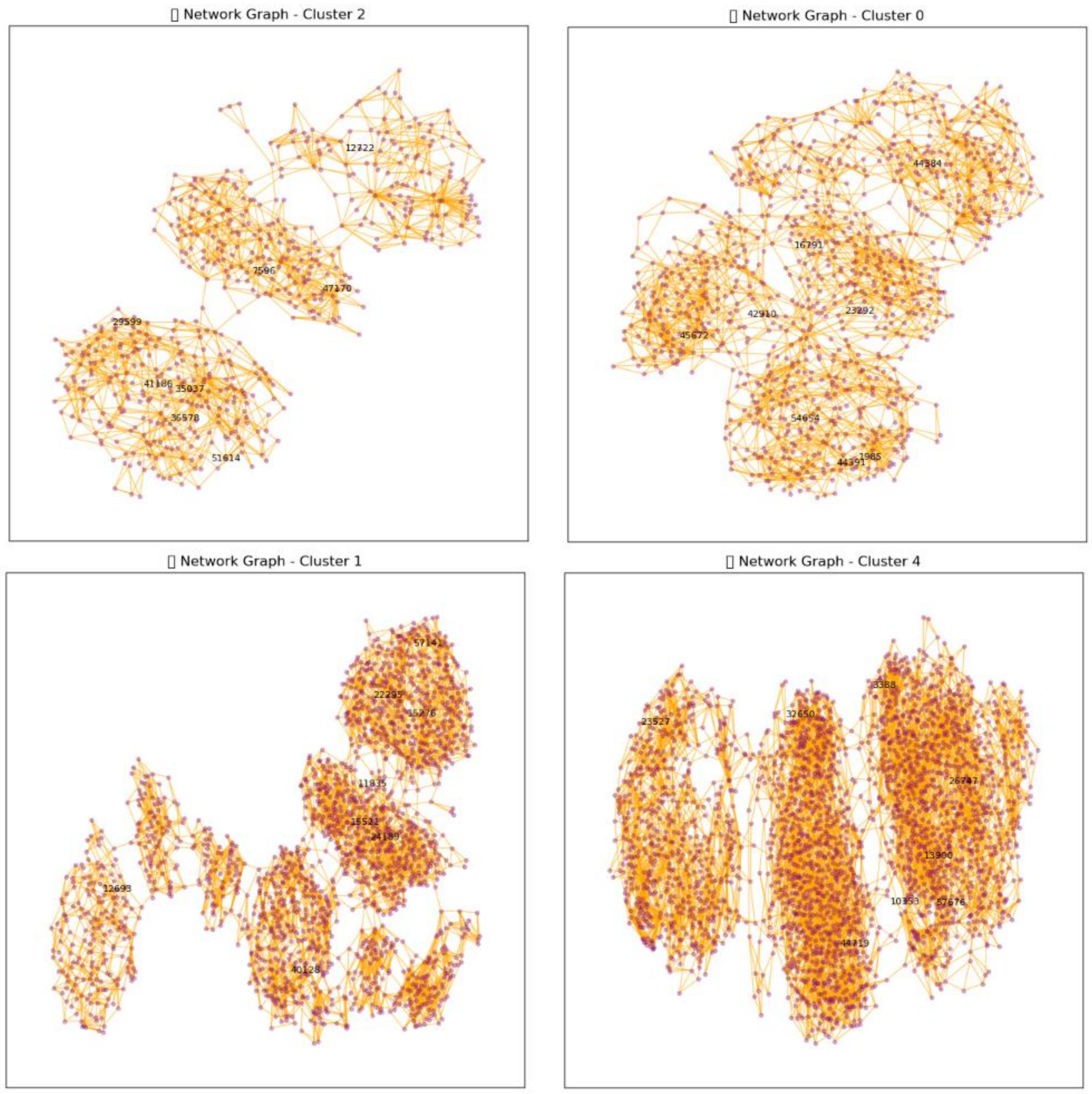
A network graph is well suited to display matched profiles based on cosine similarity.

- Nodes = Profile IDs
- Edges (Links) = Cosine similarity scores (thicker edges = stronger match)
- density and structure of the graphs provide insights into the connectivity and clustering of profiles

Results:

The network from each HDBSCAN cluster shows several dense sub-clusters. These tightly connected groups represent profiles sharing many similarities.

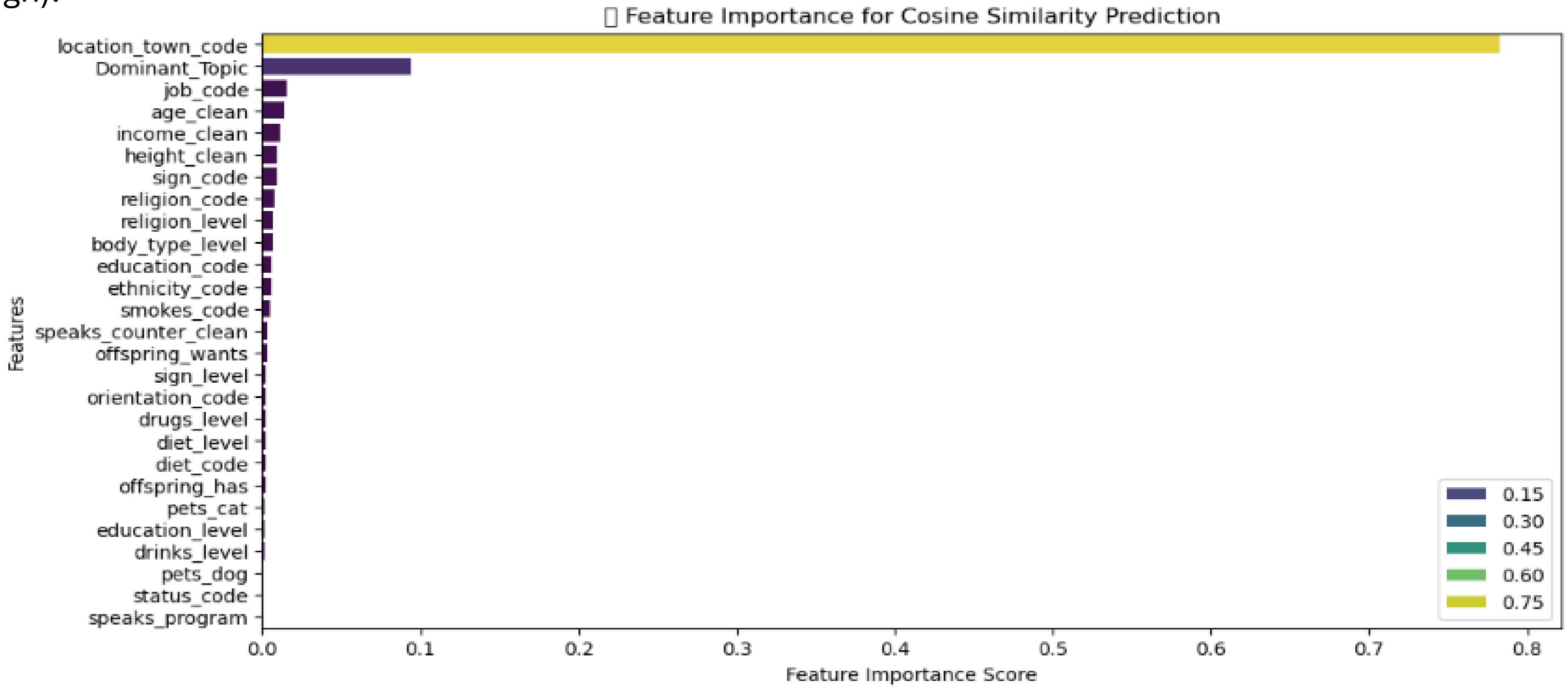
Bridging nodes (profiles with multiple connections between groups) could act as central figures in matchmaking.



5.5.3/ Cosine pairing – Feature importance

Method: Feature importance is determined using Random Forest algorithm and Gini importance by how much each feature reduces impurity across all decision trees in the forest.

Results: The most important features are by far location which contributes 78% of the model, followed by LDA topic which explains 9% of the model and a few other features contributing ~2-3% (job, age, income, height and sign).



6/ Matches visualisation

Methods:

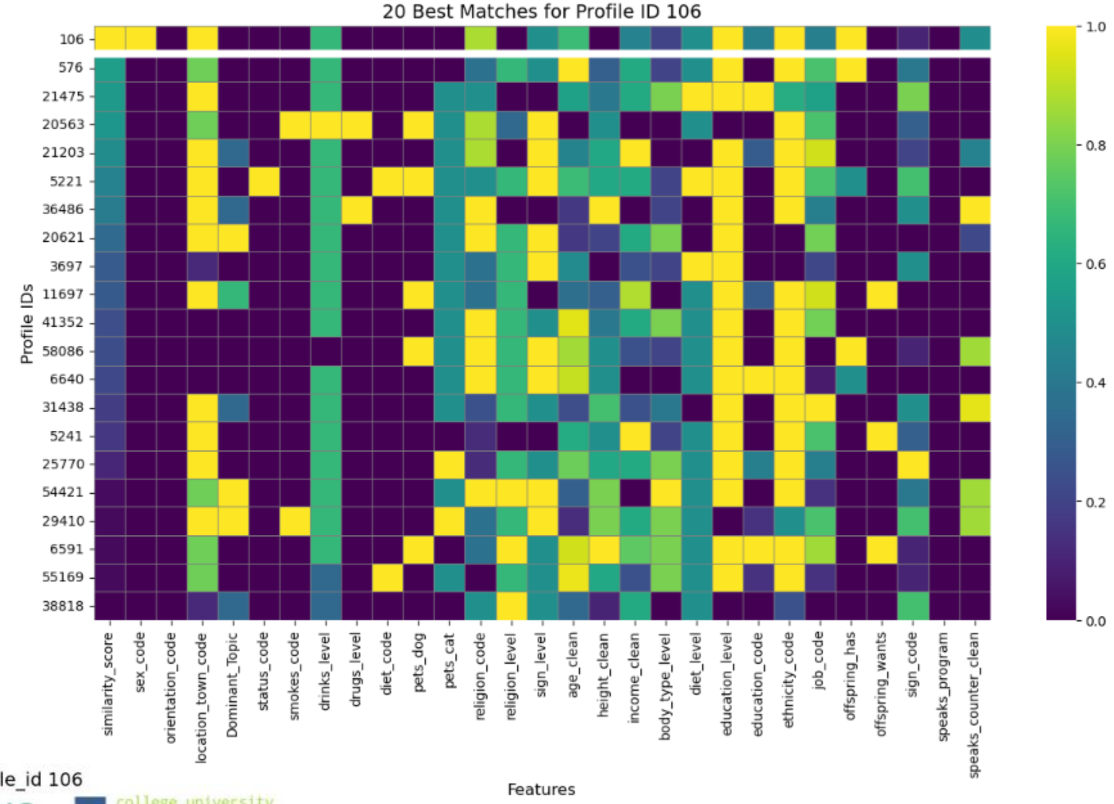
- Heatmap of encoded features
- WordCloud of categorical feature labels
- Gradio interactive interface to display up to 20 closest matches and all the relevant features
- Tableau Public dashboard to summarise the resuls

User cases:

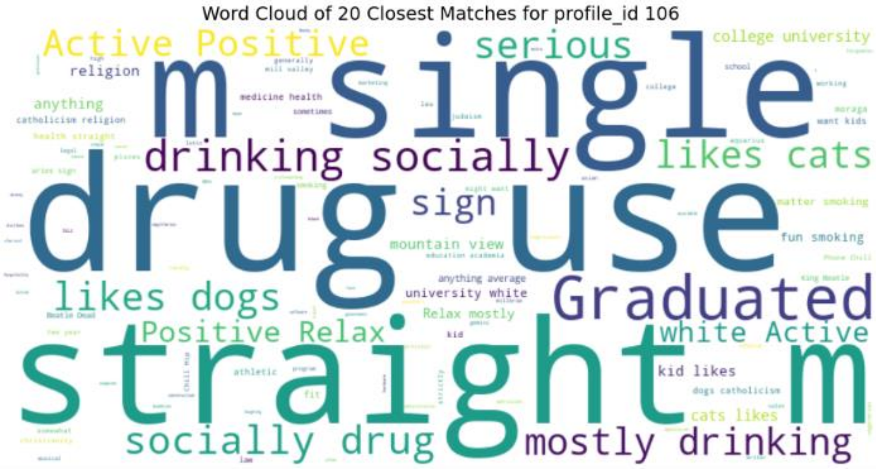
1. Profile ID 106
2. Profile ID 57749

6.1/ 20 closest matches for profile_id 106

Profile Id	106
Location Town	mountain view
Orientation	straight
Sex	female
Status	single
Topic Label	Enjoy / New / Look
Education Type	masters program
Education Level Label	Graduated
Job	education / academia
Offsprings Has Label	has >1 kid
Offspring Wants Label	
Pets Cat label	
Pets Dog Label	likes dogs
Smokes Code Label	smoking: no
Drinks Level Label	drinking: socially
Drugs Level Label	drug use: no
Diet Type	anything
Religion Code Label	judaism
Sign Code Label	aries



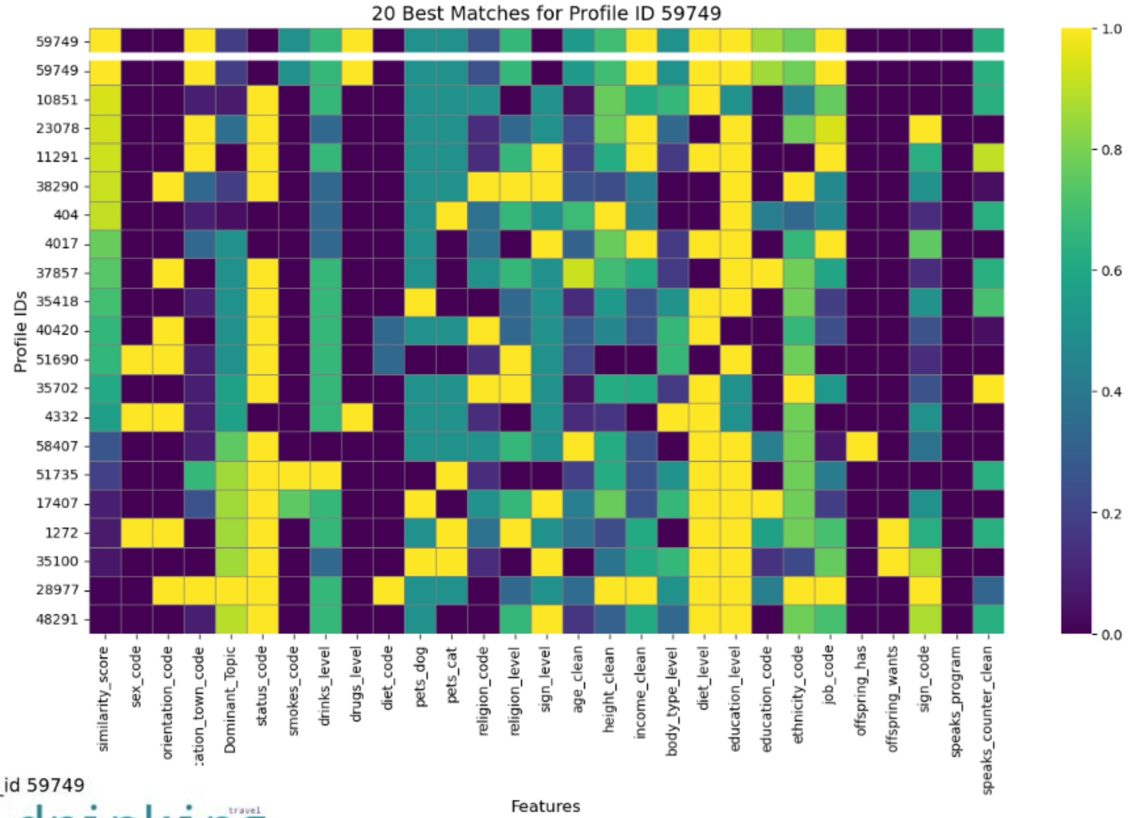
ML results: The 20 closest matches to profile_id 106 are single straight males who have graduated university, drink socially, occasionally smoke and use drugs, like cats, and mostly reside in Mountain View.



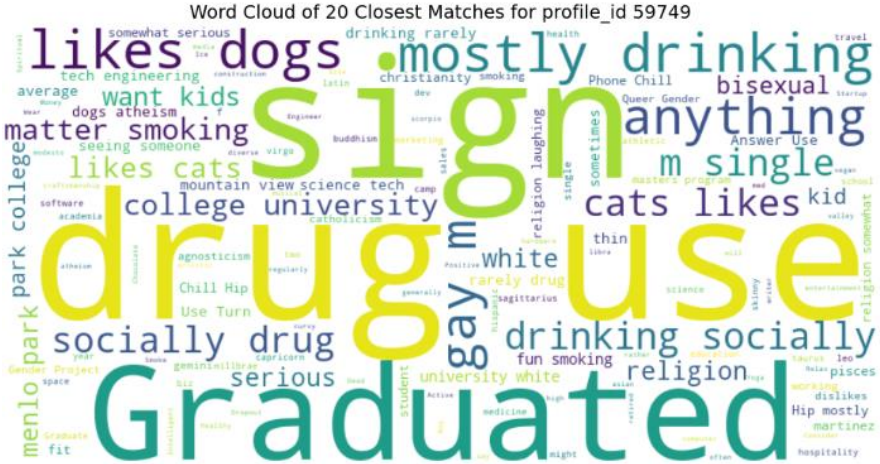
profile_id	orientation	sex	age_filtered	height_filtered	ethnicity_clean	location_town	location_state	location_country	education_typ
106	straight	f	42	65.000000	white	mountain view	california	US	masters progr
576	straight	m	61	68.000000	white	moraga	california	US	college/unive
21475	straight	m	37	69.000000	hispanic / latin, white	mountain view	california	US	two-year coll
20563	straight	m	19	70.000000	white	moraga	california	US	college/unive
21203	straight	m	32	71.000000	white	mountain view	california	US	law school
5221	straight	m	42	71.000000	white	mountain view	california	US	college/unive
36486	straight	m	23	75.000000	white	mountain view	california	US	college/unive
20621	straight	m	23	67.000000	asian	mountain view	california	US	college/unive
3697	straight	m	33	65.000000	asian	millbrae	california	US	college/unive
11697	straight	m	29	68.000000	white	mountain view	california	US	law school
41352	straight	m	58	69.000000	white	mill valley	california	US	college/unive

6.2/ 20 closest matches for profile_id 59749

Profile Id	59749
Location Town	mountain view
Orientation	gay
Sex	male
Status	seeing someone
Topic Label	Lot / Make / Read
Education Type	space camp
Education Level Label	Graduated
Job	science / tech / engineering
Offsprings Has Label	
Offspring Wants Label	doesn't want kids
Pets Cat label	likes cats
Pets Dog Label	likes dogs
Smokes Code Label	smoking: sometimes
Drinks Level Label	drinking: socially
Drugs Level Label	drug use: sometimes
Diet Type	anything
Religion Code Label	buddhism
Sign Code Label	



ML results: The 20 closest matches to profile_id 59749 are single gay males who have graduated university, drink socially, occasionally smoke and use drugs, like dogs and cats, and eat anything.



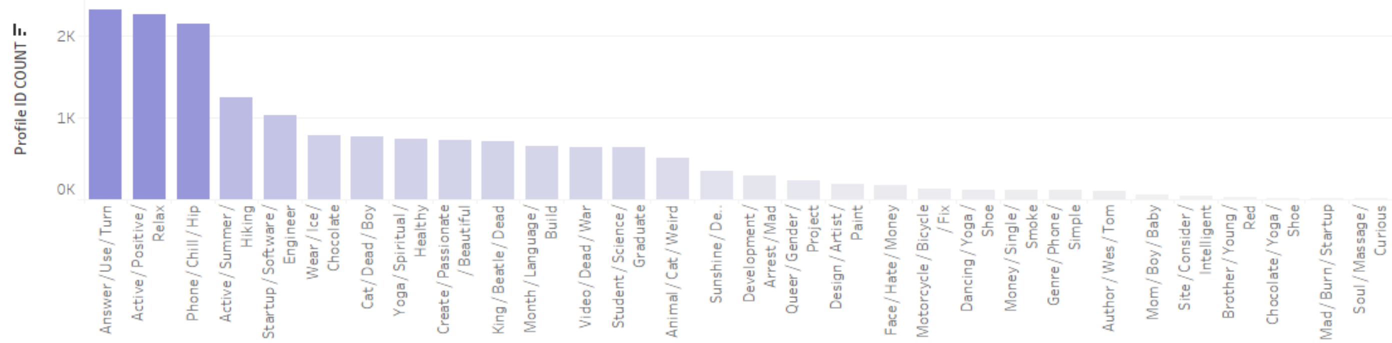
profile_id	orientation	sex	age_filtered	height_filtered	ethnicity_clean	location_town	location_state	location_country	education_typ
59749	gay	m	37	70.000000	white	mountain view	california	US	space camp
10851	gay	m	20	71.000000	hispanic / latin, white	menlo park	california	US	college/unive
23078	gay	m	25	71.000000	white	mountain view	california	US	college/unive
11291	gay	m	24	69.000000	asian, white	mountain view	california	US	college/unive
38290	bisexual	m	26	64.000000	nan	millbrae	california	US	college/unive
404	gay	m	44	74.000000	hispanic / latin	menlo park	california	US	masters progr
4017	gay	m	28	71.000000	other	millbrae	california	US	college/unive
37857	bisexual	m	59	70.000000	white	martinez	california	US	two-year coll
35418	gay	m	22	68.000000	white	menlo park	california	US	college/unive
40420	bisexual	m	27	67.000000	other	martinez	california	US	college/unive

6.3/ ML summary

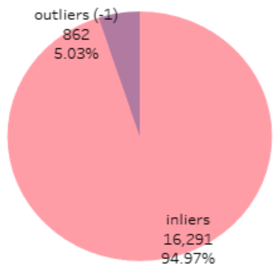
Tableau Public dashboard allows to present ML outputs in a concise manner (link).

Machine Learning Outputs

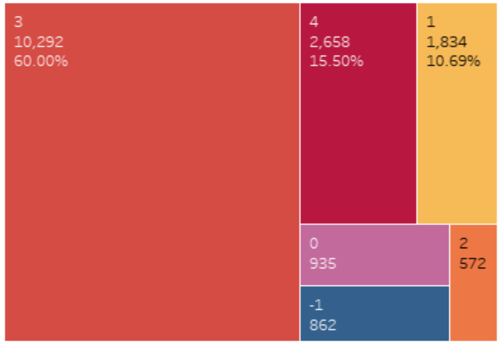
LDA Topic Distribution



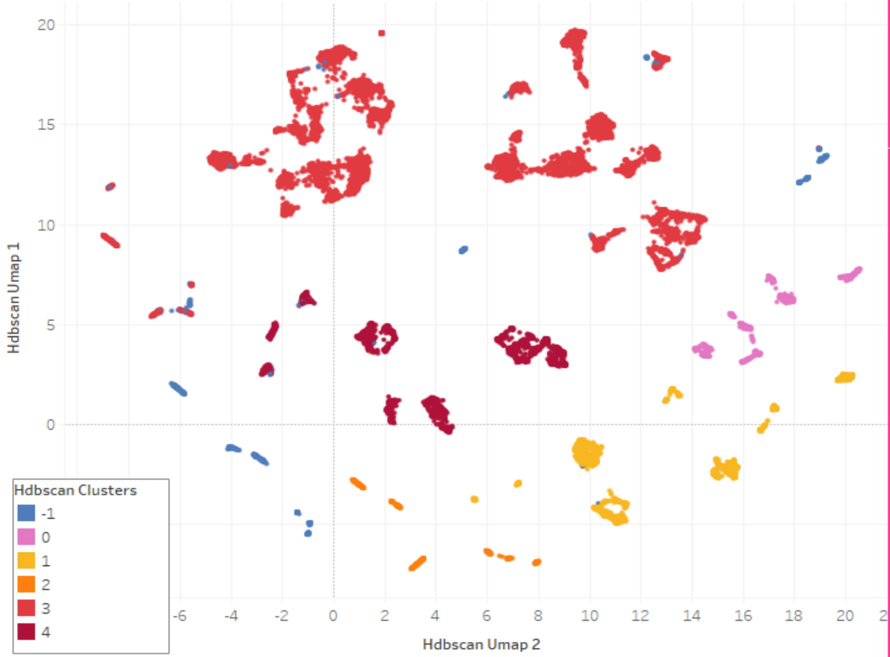
HDBSCAN Outliers



HDBSCAN Cluster Proportions



HDBSCAN Clustering of Individuals



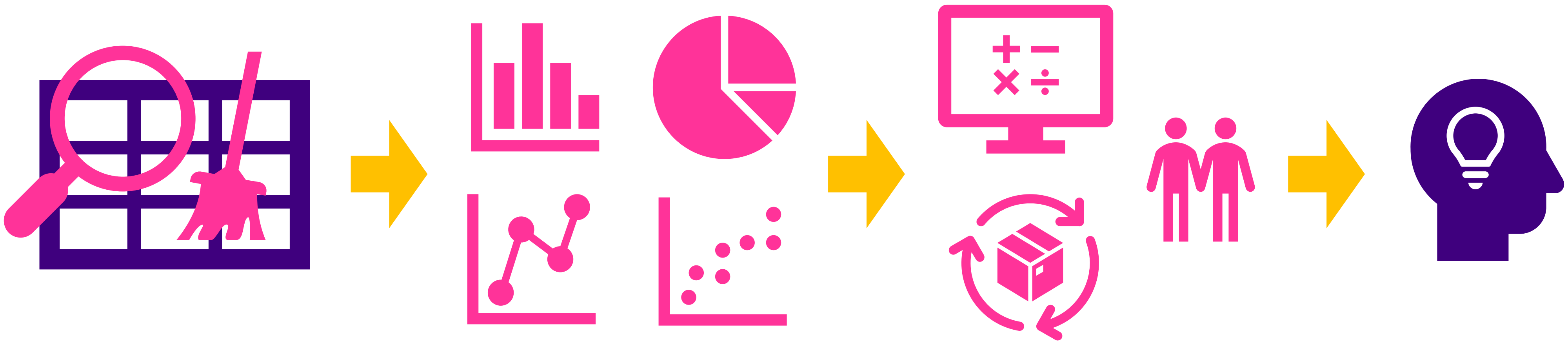
Match Table for Profile IDs 106 and 59749

Profile Id (S..	Match Profi..	Cosine S..	Location To..	Orientation	Sex	Status	Topic Label
106	576	0.9993345..	moraga	straight	m	single	Enjoy / New
	21475	0.9993118..	mountain vi..	straight	m	single	Enjoy / New
	20563	0.9992881..	moraga	straight	m	single	Enjoy / New
	21203	0.9992254..	mountain vi..	straight	m	single	New / Live /
	5221	0.9991609..	mountain vi..	straight	m	available	Enjoy / New
	36486	0.9991245..	mountain vi..	straight	m	single	New / Live /
	20621	0.9990128..	mountain vi..	straight	m	single	Family / Kn.
	3697	0.9989258..	millbrae	straight	m	single	Enjoy / New
	11697	0.9989193..	mountain vi..	straight	m	single	Want / Kno.
	41352	0.9988582..	mill valley	straight	m	single	Enjoy / New
	58086	0.9988549..	mill valley	straight	m	single	Enjoy / New
	6640	0.9988292..	mill valley	straight	m	single	Enjoy / New
	31438	0.9987700..	mountain vi..	straight	m	single	New / Live /
	5241	0.9987399..	mountain vi..	straight	m	single	Enjoy / New
	25770	0.9986647..	mountain vi..	straight	m	single	Enjoy / New
	54421	0.9985454..	moraga	straight	m	single	Family / Kn.
	29410	0.9985446..	mountain vi..	straight	m	single	Family / Kn.
	6591	0.9985435..	moraga	straight	m	single	Enjoy / New
	55169	0.9985403..	moraga	straight	m	single	Enjoy / New
	38818	0.9985044..	millbrae	straight	m	single	New / Live /
59749	59749	1	mountain vi..	gay	m	seeing som..	Lot / Make /
	10851	0.9987165..	menlo park	gay	m	single	Want / Mes
	23078	0.9984495..	mountain vi..	gay	m	single	Look / Fun /
	11291	0.9983410..	mountain vi..	gay	m	single	Try / New / .
	38290	0.9981718..	millbrae	bisexual	m	single	Lot / Make /
	404	0.9980336..	menlo park	gay	m	seeing som..	Dance / Wo.
	4017	0.9950732..	millbrae	gay	m	seeing som..	Think / Kno.
	37857	0.9943424..	martinez	bisexual	m	single	Think / Kno.
	35418	0.9935083..	menlo park	gay	m	single	Think / Kno.
	40420	0.9927069..	martinez	bisexual	m	single	Think / Kno.
	51690	0.9926849..	menlo park	bisexual	f	single	Think / Kno.
	35702	0.9915832..	menlo park	gay	m	single	Favorite / R
	4332	0.9906059..	menlo park	bisexual	f	seeing som..	Favorite / R
	58407	0.9842942..	menlo park	gay	m	single	Enjoy / New
	51735	0.9826879..	modesto	gay	m	single	Family / Kn.
	17407	0.9804648..	mill valley	gay	m	single	Family / Kn.
	1272	0.9802400..	martinez	bisexual	f	single	Family / Kn.
	35100	0.9799847..	martinez	gay	m	single	Family / Kn.
	28977	0.9788370..	mountain vi..	bisexual	m	single	Think / Get .
	48291	0.9786879..	menlo park	gay	m	single	Girl / Think

Profile ID

- [ ] (All)
- [ ] 55
- [x] 106
- [ ] 107
- [ ] 182
- [ ] 210
- [ ] 282
- [ ] 316
- [ ] 329
- [ ] 368
- [ ] 404
- [ ] 419
- [ ] 473
- [ ] 576
- [ ] 599
- [ ] 818
- [ ] 1032
- [ ] 1096
- [ ] 1187
- [ ] 1271
- [ ] 1272
- [ ] 1437
- [ ] 1462
- [ ] 1516
- [ ] 1610
- [ ] 1687
- [ ] 1758
- [ ] 1809
- [ ] 1902
- [ ] 1931
- [ ] 2003
- [ ] 2172
- [ ] 2521
- [ ] 2643
- [ ] 2891
- [ ] 3037
- [ ] 3080
- [ ] 3102
- [ ] 3176
- [ ] 3296
- [ ] 3542
- [ ] 3585
- [ ] 3596
- [ ] 3697
- [ ] 4017
- [ ] 4095
- [ ] 4219

7/ Wrapping up



7.1/ Conclusions and Perspectives

Conclusions:

- EDA revealed the archetypical dating app user.
- Several ML models were implemented to categorise textual features (LDA), eliminate outliers (Isolation Forest), structure the data (HDBSCAN clustering) and pair the most similar individuals (cosine).
- Visualisation of the closest matches using heatmap, worcloud, and interactive tables help outline the general trends of selected individuals

What's next?

Things that could be done differently:

- Different encoding of the features (e.g. agglomerate some variable labels to decrease granularity)
- No weighting of the variables (no bias) or different weighting strategy
- No row selection, instead testing different NA imputation methods (mean, KNN,...)
- No outlier elimination, instead analyse them and associate them to features
- Use UMAP projections as predictive features in cosine model for improved granularity
- Retrieve GPS data of the town for geographical mapping to then including geographical proximity to the model
- No combination of the 10 essay variables prior to NLP modeling
- For NLP analysis, compare LDA method to BERT modeling which incorporates context as well.
- Test model with new users (fictional or retrieved from public repository)
- Further develop the Gradio interface to search for traits (e.g. "wants more kids") rather than profile id.

Resources

- Python code link
- Tableau Public link

Thank you!

☺

Google Scholar ORCID LinkedIn GitHub Tableau Public Website email