

# OkCupid Project Report

BANA 212 Team 6: Kevin Cheung, Kathy Han, Changyeol Lee, Peggy Lee, Yuqing Wang

[**OkCupid Project Report**](#_r3ndgw5txidh)1

[**Introduction**](#_8v6okx26zm2j)2

[**Dataset description**](#_f1dm8vabb4d8)3

[**Dataset pre-processing**](#_3q7miq9g4ksa)3

[**Exploratory Analysis**](#_j6zk8d6ib5dd)4

[CountVectorizer](#_6qqp30i14nm9) 4

[TF-IDF (Term Frequency - Inverse Document Frequency) and TfidfVectorizer](#_bc7m5t8dugm) 5

[LDA (Latent Dirichlet Analysis) and NMF (Non-negative Matrix factorization)](#_ijbn6die4ut4) 5

[Sentiment Analysis](#_d7job57wmokt) 5

[Matching Algorithm with Jaccard Similarity](#_f21q072nupzo) 6

[**Analyses**](#_inm5l9yqwyog)6

[**Conclusion**](#_8xqmqihynuph)16

[**Bibliography**](#_b0d9reog9qat)17

# **Introduction**

The team chose to focus on online dating profiles specifically from the OkCupid platform. OkCupid is an online dating application that uses quizzes, multiple-choice questions and essays to find the closest match for the user. The vast majority of dating applications focus on the physical appearance of their users. The action of sending “likes” to a potential interest is solely based on the pictures displayed. However, the OkCupid platform strives to find more meaningful match suggestions by requiring a detailed questionnaire. The answers collected essentially serve as data for the OkCupid algorithm to determine compatibility amongst the users. However, OkCupid does not take into account the free response essays when finding a match for a profile (1). These free response essays give much more personal insights into a profile and can be powerfully leveraged to increase the chance of finding a match. By utilizing text analysis, OkCupid and the consumer can benefit - OkCupid’s algorithm can be improved and they can charge a premium to users who want a more personalized matching experience.

We are exploring this topic due to the significant growth of interest in the online dating industry. More people are looking to find deeper connections with the help of online dating services. Since dating profiles are the first thing users encounter before matching or meeting up with a potential significant other, we want to discover what makes a desirable profile and how to increase their chances of matching with another person. Here are some exploratory questions we attempt to answer in our analyses:

* What words or phrases a profile can use to increase their chances of matching with other profiles.
* Based on the words profiles choose, can we determine if the user has a positive or negative sentiment.
* Can we cluster profiles based on their interests and similarities.
* Can we recommend the best match based on their attributes, sentiment, and cluster.

*Related work/Inspiration*

According to research from Pew Research Center, the number of people using dating apps continue to increase. Approximately 30% of adults in the United States have participated in an online dating site or app in 2019. They argued that the popularity of the app is due to changes of social norms and behaviors around marriage (2). Furthermore, due to the pandemic, the number of dating app users is surging, according to reports from Business insider and CNBC (3, 4). About 57% of users evaluated the apps positively, because the user can estimate their partner's physical appearance and interests before the date, according to the research (2). Thus, the users could easily and efficiently find their partners. These research reports and articles inspired our interest in the online dating industry, and provoked us to think about possible questions, data exploration techniques and analytical approaches we can use to discover more about this topic.

# **Dataset description**

The OkCupid dataset, which consisted of 59,946 profiles and 31 variables, was sourced from Kaggle (5). These profiles were located around the San Francisco, California region. This dataset contains 10 free text responses with the following essay prompts: 'essay0' : 'My self summary', 'essay1' : 'What I'm doing with my life', 'essay2' : 'I'm really good at', 'essay3' : 'The first thing people notice about me', 'essay4': 'Favorite books, movies, tv, food', 'essay5' : 'The six things I could never do without', 'essay6' : 'I spend a lot of time thinking about', 'essay7' : 'On a typical Friday night I am', 'essay8' : 'The most private thing I am willing to admit', and 'essay9' : 'You should message me if'. The dataset also includes the following profile attributes: age, relationship status, gender, sexual orientation, body type, diet, drinking, drug usage, education, ethnicity, height, income, job, last time online, location, offspring, pets, religion, astrological sign, smoking, and language spoken. All of this information provides a diverse set of free text responses and detailed profile attributes that we will utilize in our analyses.

# **Dataset Pre-Processing**

We decided not to remove all rows with NA values because that would reduce our dataset from 59,946 profiles to approximately 4,300 profiles. Furthermore, these profiles purposefully left some of their responses blank, therefore we did not want to impute NA values with other values. These NA values were purposeful and replacing them would alter outcomes. We can conclude that these NA values are the topics the users are less comfortable disclosing when it comes to seeking a significant other. Instead, we replaced NA values with blank spaces. Also, we removed users who were 65 years or older, which removed 372 profiles from the dataset. We chose 65 years and older because 65 is the average retirement age and may not be ideal for the purpose of this project. This also removed the extreme outliers as there were profiles who listed themselves as 110 years old. If we were to remove age based on IQR, we would have needed to remove 2,890 rows, as those profiles were over the age of 52 (the upper bound), however, we were interested in this demographic.

For textual analysis, we had to convert all the essays to strings. Within the essay texts, we removed stopwords using nltk stopwords, which consisted of 179 words (6). Furthermore, we removed punctuation and white spaces from the essays. In order to perform textual analysis, we needed to tokenize the essays. We first tokenized each sentence into a list of sentences. Then, upon tokenizing each sentence, we split each sentence into a list of individual words or, in this case, tokens (7). We also converted any words with uppercase to lowercase characters and removed words with a length of less than 3 characters. After word tokenization, the tokens were stemmed using PorterStemmer, and the tokens were converted to stem words (8). We will compare stemmed tokens and unstemmed tokens when performing our analyses.

# **Exploratory Analysis**

We performed NLP and Topic Modeling with two different approaches: LDA (Latent Dirichlet Analysis), and NMF (Non-negative Matrix factorization). Within these approaches, we processed individual terms in each profile and clustered profiles into meaningful topic groups. There are two different approaches when performing LDA and NMF. These two approaches utilize two different feature extraction techniques: CountVectorizer and TfidfVectorizer. Furthermore, we performed sentiment analysis on the essays using TextBlob and we created a profile matching algorithm using the Jaccard Similarity Coefficient to calculate the similarity scores based on the attributes and new attributes we will create.

## *CountVectorizer*

The CountVectorizer tokenizes a collection of text documents and builds a vocabulary of known words, while also encoding new documents using that same vocabulary. This returns an encoded matrix with a length of the entire corpus and a width of word count with a frequency count under each word (9).

## *TF-IDF (Term Frequency - Inverse Document Frequency) and TfidfVectorizer*

The TF part of TF-IDF, summarizes how often a given term appears within a document and the IDF part downscales terms that appear more frequently across the corpus, or collection of documents. TF-IDF quantifies a word in documents, which computes a weight to each word, signifying the importance of the word in the document and corpus.

TfidfVectorizer will tokenize documents, learn the vocabulary and IDF weightings, and returns a matrix of TF-IDF features. The TfidfVectorizer first applies the CountVectorizer, then applies the TfidfTransformer, which takes the count matrix and normalizes it into vector formats with the corpus as length and features (tokens) as width with the weight under each token (10).

## *LDA (Latent Dirichlet Analysis) and NMF (Non-negative Matrix factorization)*

LDA and NMF are two different unsupervised topic modeling techniques. LDA is based on probabilistic graphing modeling, which will return the documents that belong to a topic in a corpus and the words that belong to a topic. LDA uses the CountVectorizer as a document word matrix input. NMF is based on linear algebra, which uses the original matrix (A) from the TfidfVectorizer and gives two matrices (W and H). W is the topics found and H is the weights for those topics. In other words, A is documented by words, H is documented by topics, and W is topics by words. Both of these models cluster documents to discover topics based on their contents (11).

## *Sentiment Analysis*

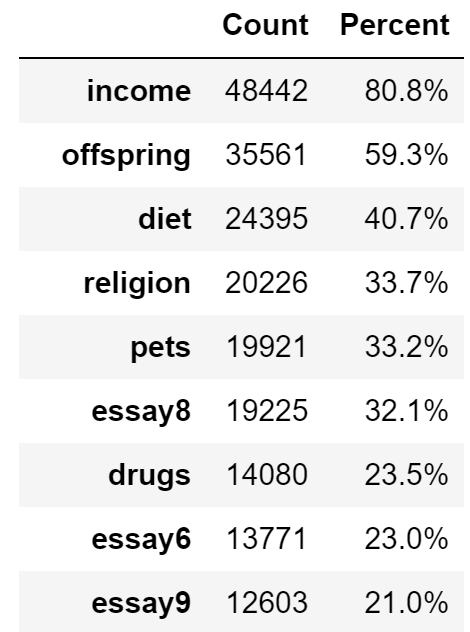
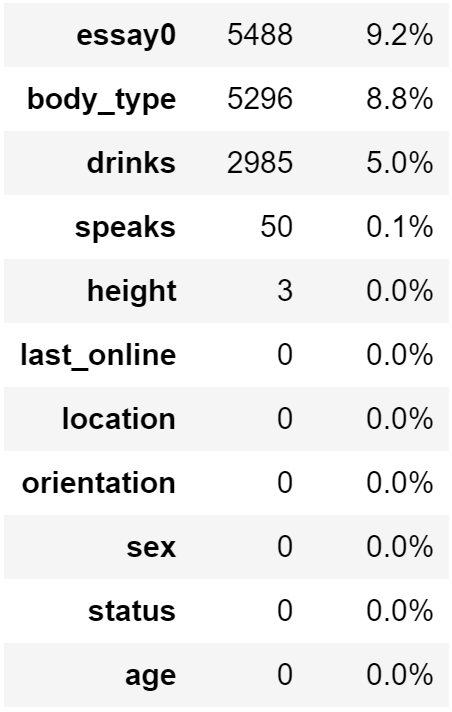
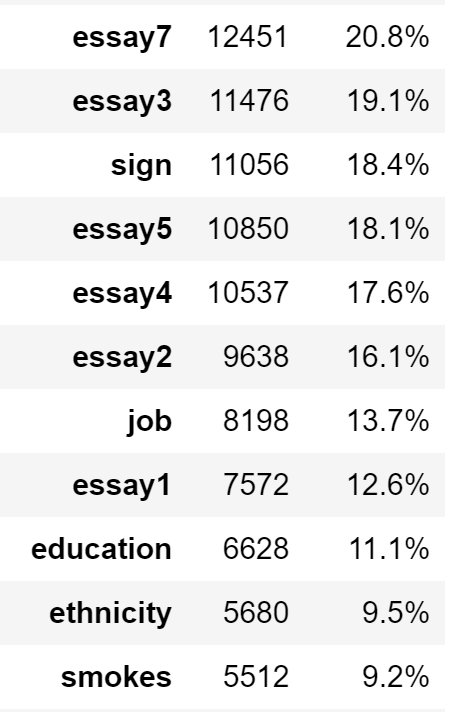
Sentiment Analysis can help us understand the sentiment and gather insightful information in text. We used TextBlob under NLTK to perform sentiment analysis. TextBlob returns a polarity and subjectivity score in a text. Polarity lies between -1 and 1, inclusive, where -1 has a negative sentiment and +1 has a positive sentiment. Subjectivity lies between 0 and 1, inclusive, where subjectivity measures the amount of personal opinion and factual information in the text. The higher the subjective sentiment score means the text contains more personal opinion rather than factual information (13).

## *Matching Algorithm with Jaccard Similarity*

The matching algorithm function calculates the Jaccard Similarity score between one profile and all other profiles. The Jaccard Similarity is the ratio of the size of the intersection of two sets to the size of their union. The Jaccard Similarity coefficient is between 0 and 1, where the higher the score, the more similar the two sets are. After all the calculations, the algorithm selects the highest score, which is the best match for the profile (14).

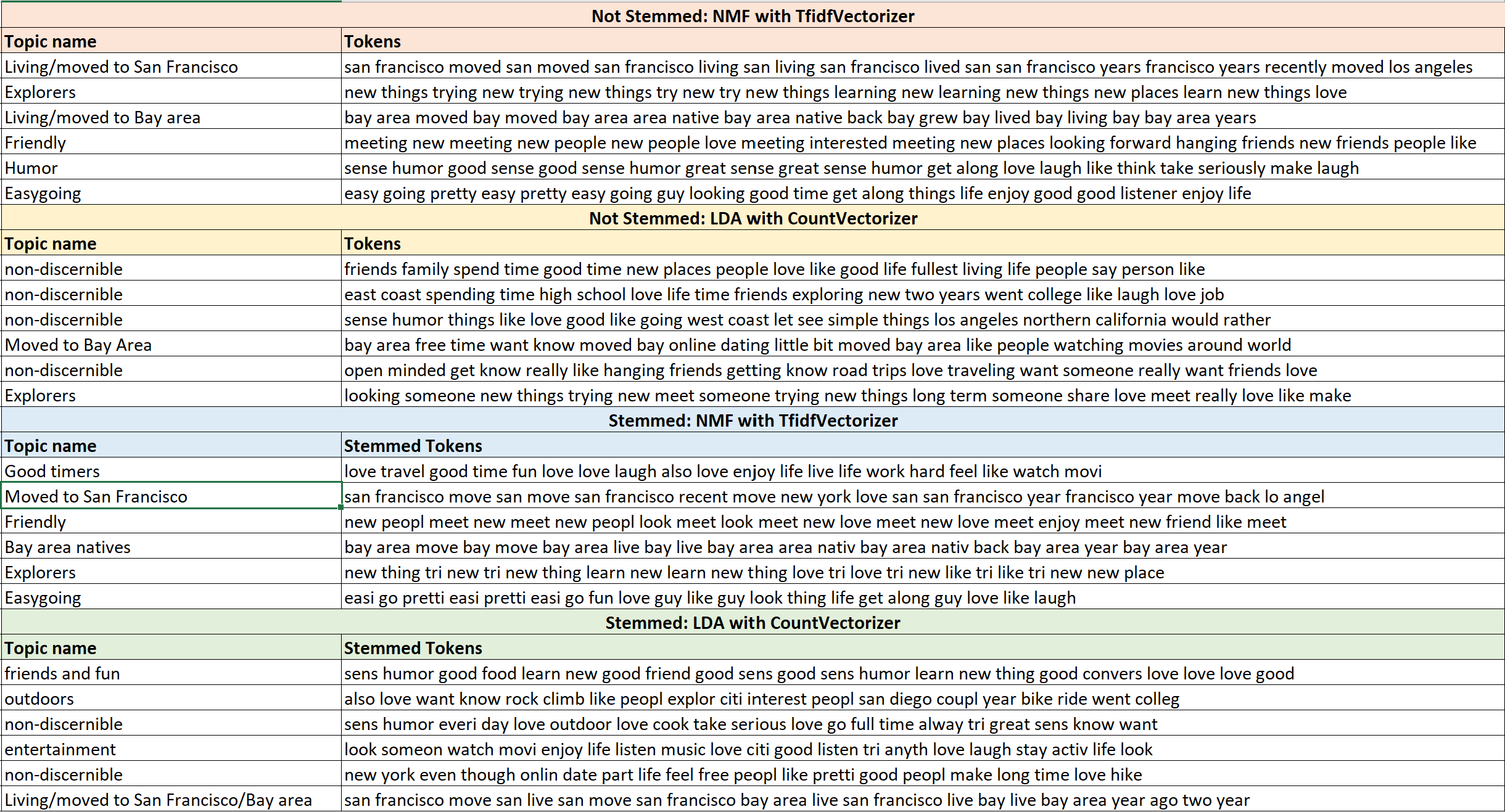
# **Analyses**

Upon removing 65 and older profiles, the dataset included 35,657 males (59.85%) and 23,917 females (40.15%). The mean age was approximately 32 for males and 33 for females. To discover which variables were of most use to us, we analyzed the count and percent of NA values in each column. Below are the variables with NA value counts and percentages. It is interesting to note that 'essay8' : 'The most private thing I am willing to admit' and 'essay6' : 'I spend a lot of time thinking about' were the most ignored essays. These two essays are very personal questions and were omitted by 32% of users for essay8 and 23% of users for essay6.

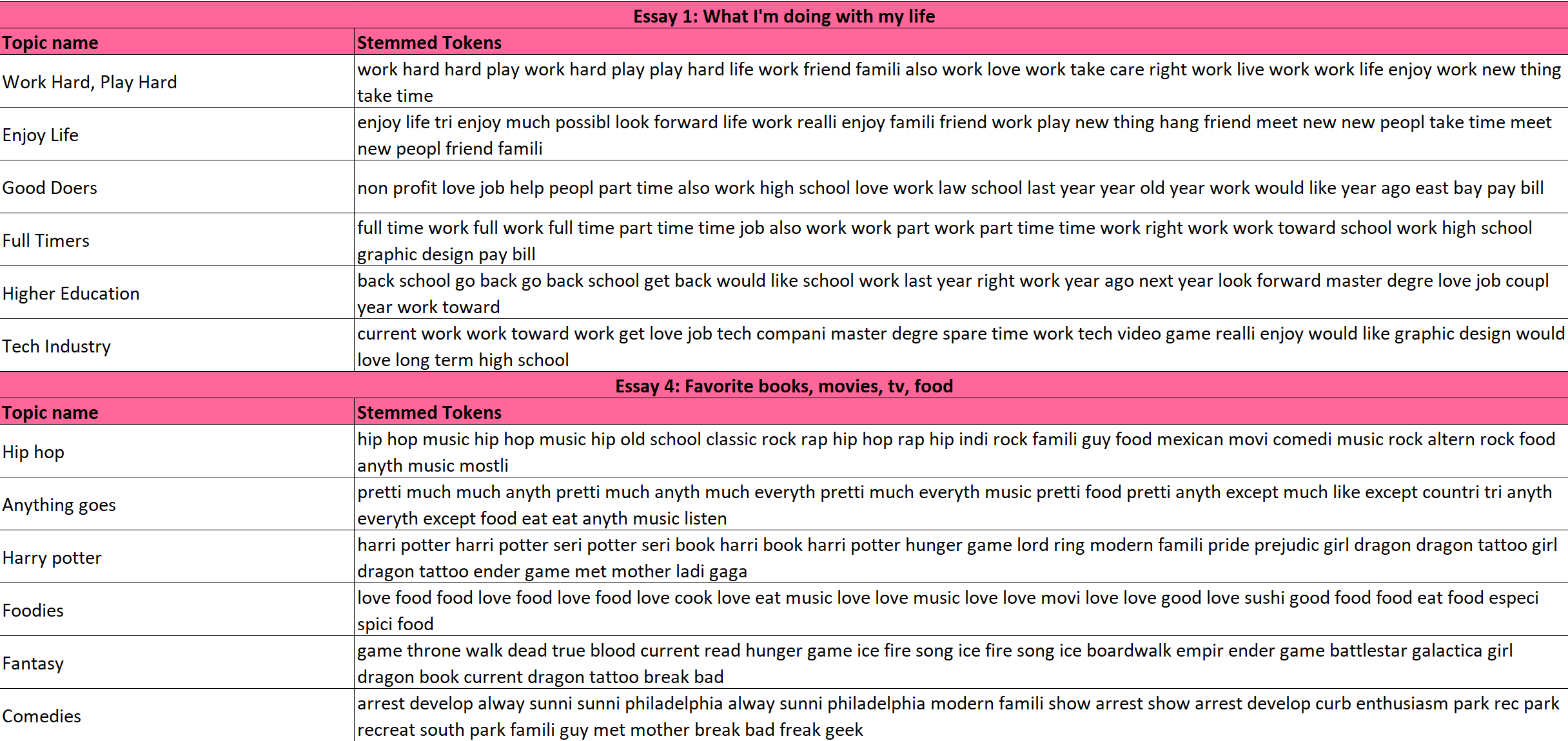
 

We performed exploratory visual analysis to look at demographics and the descriptive characteristics of all the profiles. 86 percent of profiles identified themselves as straight, 9.4 percent identified as gay, and 4.6 percent identified as bisexual. The top four ethnicities included White, Asian, Hispanic/Latin, and Black. Most males identified themselves as having an average, fit, or athletic body type, whereas women labeled themselves as having an average, fit, curvy body type, or left this characteristic blank. When looking at education level, women and men tend to be more evenly distributed in higher education, such as having a masters degree or a law degree. 73 percent of users do not smoke and 69.7 percent of users drink socially. Since this dataset focuses on profiles located in the San Francisco region, it is not surprising to see that there is a high proportion of men in the STEM field, however, women had a high proportion in the health and education fields.

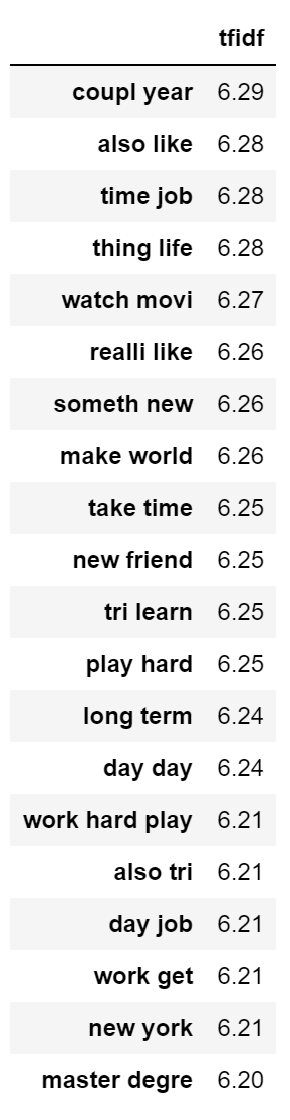
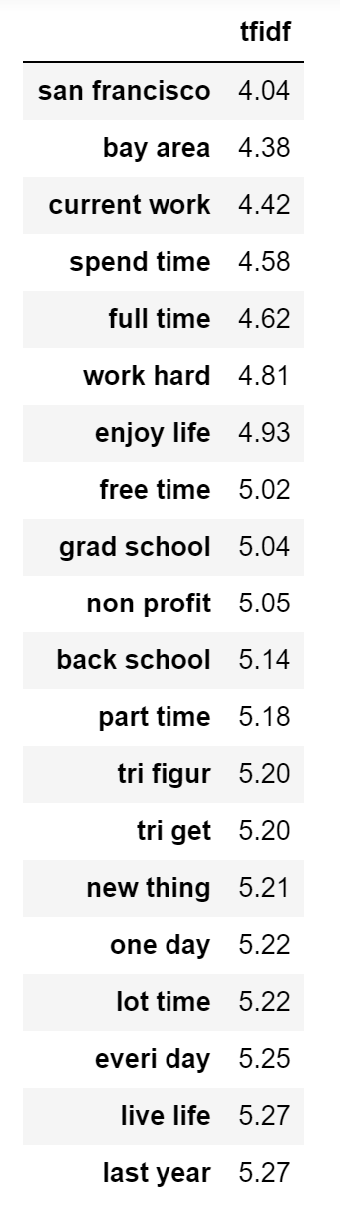
Our analyses primarily focused on the essay portion of the data as these were free text responses by individuals. We first wanted to test our different techniques, so we decided to use essay0 ('My self summary') as that was the essay with the least amount of NA values (9.2%). Once we have applied our two different tokenization functions (one with stemming and the other without stemming) on essay0, we wanted to compare our two topic modelling techniques to see which one provided better topics based on the features. We first set our CountVectorizer to ignore terms that have a document frequency lower than 0.5% and to only extract bigrams and trigrams (a sequence of two and three adjacent elements from a string of tokens). Then we implemented CountVectorizer with LDA to extract 20 topics from the combined essays corpus. The number of topics was arbitrarily chosen. Once we extracted the topics from LDA, we wanted to compare it to the NMF topic modeling approach. We set our TfidfVectorizer with the same conditions as the CountVectorizer and implemented it with the NMF model. Thus we have four models to explore, stemmed essay0 tokens and unstemmed essay0 tokens on LDA and NMF, shown below.



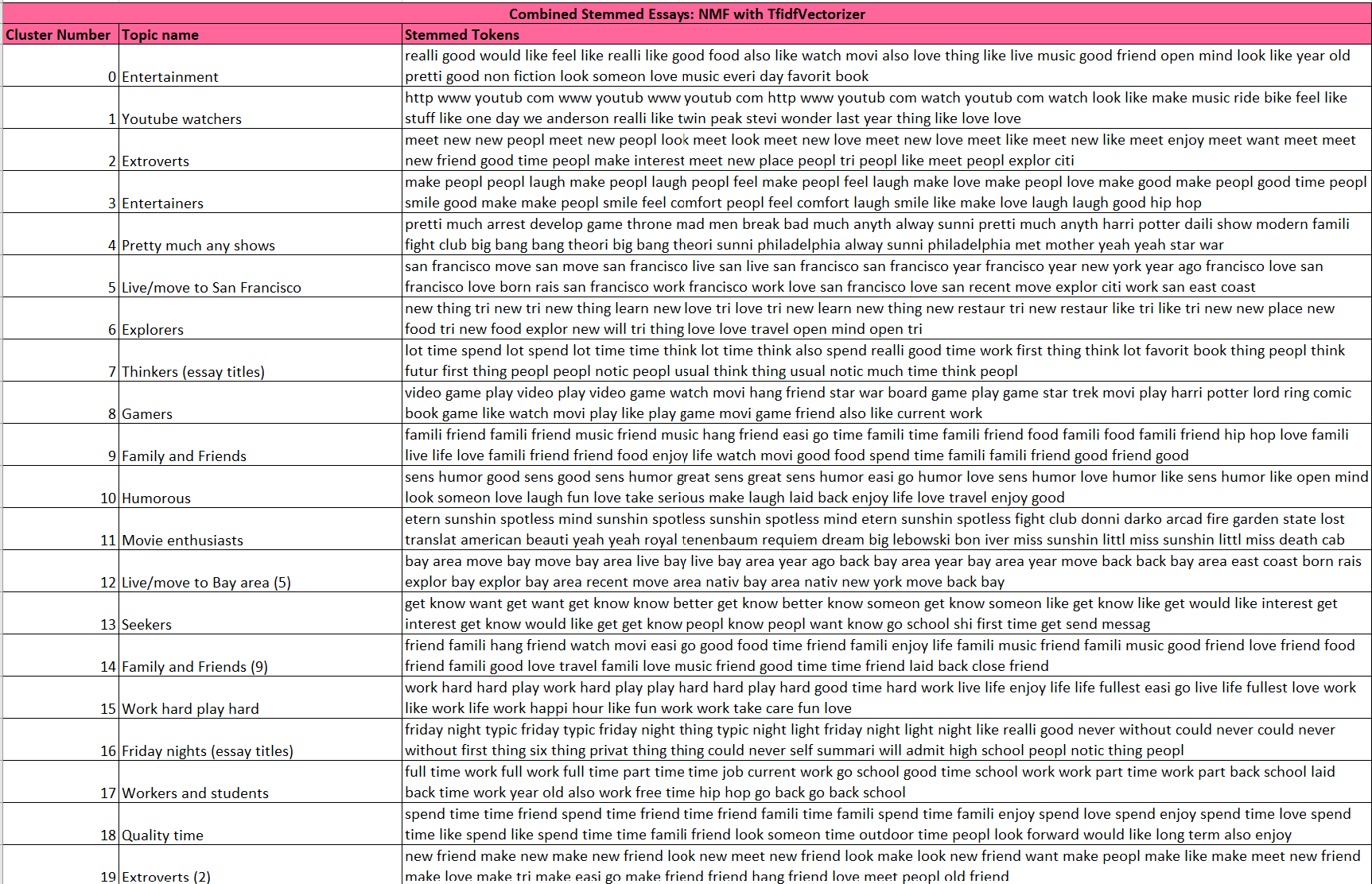
After reviewing each topic from both models, we extracted the first 6 topics from each model and attempted to give each topic a topic name. We were unable to find a clear topic for most of LDA models so we labeled them as “non-discernible” and decided LDA would not be an effective model for our analyses. Among the NMF models, we concluded that stemmed tokens with NMF with TfidfVectorizer produced the most clear and cohesive topics and would provide a better analysis since we are using stemmed tokens.

We did some further topic modeling on essay1: 'What I'm doing with my life' and essay4: 'Favorite books, movies, tv, food' as these essays were within the top 4 most filled out essays. We extracted some random clusters to show what cluster each profile is in as shown below. The tokens that show up in the beginning of each topic have the lowest weight, meaning these tokens show up the most. It is interesting to see that essay4, the topic model clustered Game of Thrones, The Walking Dead, and True Blood all together and they all belong to a “Fantasy” theme. This can also be seen in our “Comedies” cluster, where Arrested Development, It’s Always Sunny in Philadelphia, Modern Family, Curb Your Enthusiasm, Parks and Recreation, South Park, Family Guy, and How I Met Your Mother show up in the same topic. Through NMF with TfidfVectorizer, we can see that these profiles have been grouped to their respective clusters based on their interests and can be utilized by OkCupid for users to increase their chances of matching with other users based on their similarities. 

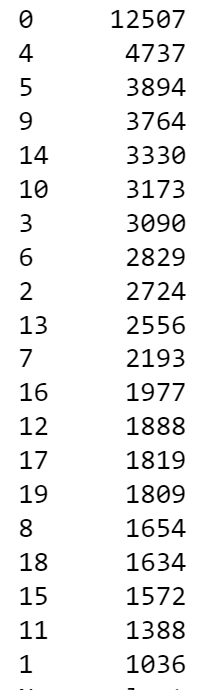
In the case of dating profiles, we discerned that the lower the TF-IDF score or the more often the word appears, the better it is for the profile. This is because users in online dating want to be matched with the most number of people to maximize their chances of finding a significant other. People can connect with one another based on their favorite movies, shows, and music similarity or they can also connect on their current situation in life. The words users use in their essays can help them increase their chances of finding a potential mate. The images shown below show the top 20 highest and lowest TF-IDF weighted n\_gram tokens for essay4 (left 2 images) and essay1 (right 2 images). For example, if a user were to use the words “Game of Thrones” in their essays often, they will be placed in a cluster where other users use that word often as well, therefore, they can be matched based on that similarity. OkCupid can offer a function for premium users, in which users can rank essays based on importance, and from there cluster profiles with the same ranking using textual analysis.

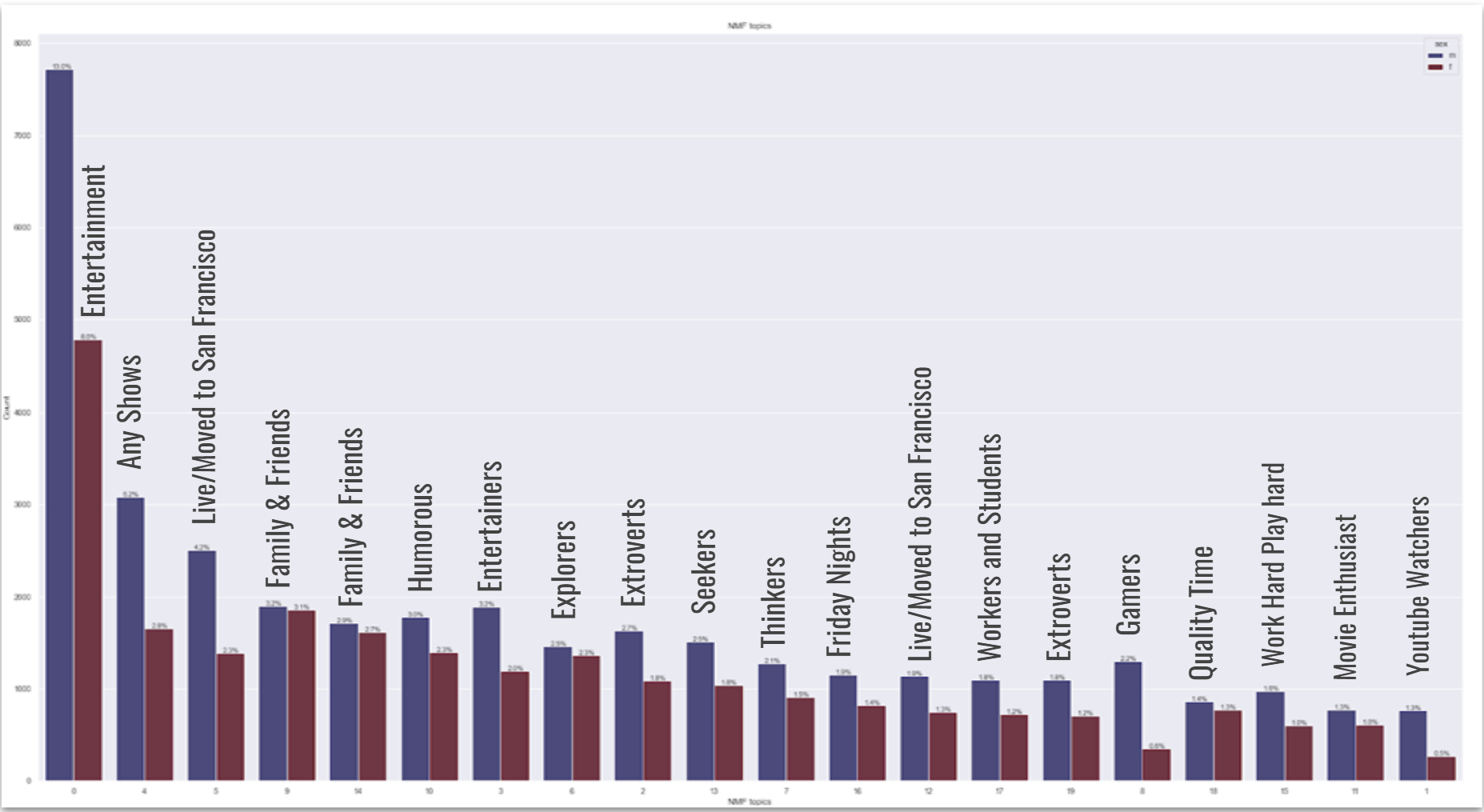


Next, we combined all the essays for each profile into one essay chunk. After we ran the NMF with TfidfVectorizer on the combined essays, we were able to cluster the profiles into their respective clusters. We will use these cluster groups further on in the analyses for our matching algorithm. As seen below, we were able to discern topic names for almost all of the topics. There were similar topics aomg the clusters, such as topics 5 and 12 (live/move to San Francisco/Bay Area), 9 and 14 (family and friends), and 2 and 19 (Extroverts).

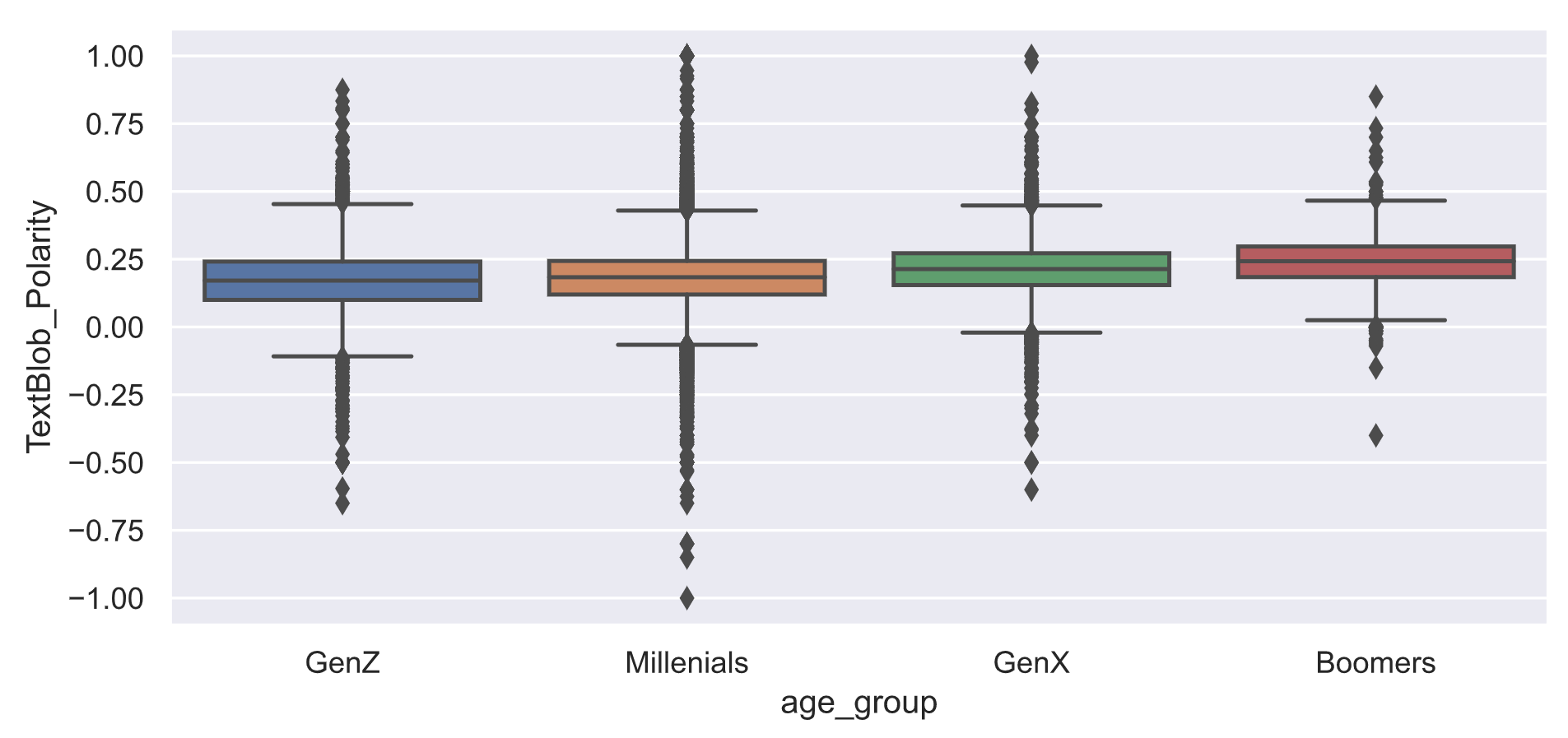


Furthermore, we grouped each profile into their cluster and plotted the clusters based on their gender. Males and females tend to have a similar distribution for most topics based on the plot below. 21 percent of profiles were clustered into the “Entertainment” group - they like good food, watching movies, music, good friends and having an open mind. It is interesting to see that the percentage of females for clusters 9 and 14 (“Family and Friends”) are very close to the percentage of males, as well as in group 6 (“Explorers”). Some of the least popular clusters were the “Youtube watchers” and “Gamers”, in which males had a much higher percentage than females.. We also see that the top words within the combined essays are: family and friends, san francisco/bay area, sense of humor, watching movies, and new things, just to name a few. We will use the clusters found in this combined essay portion, for our matching algorithm.

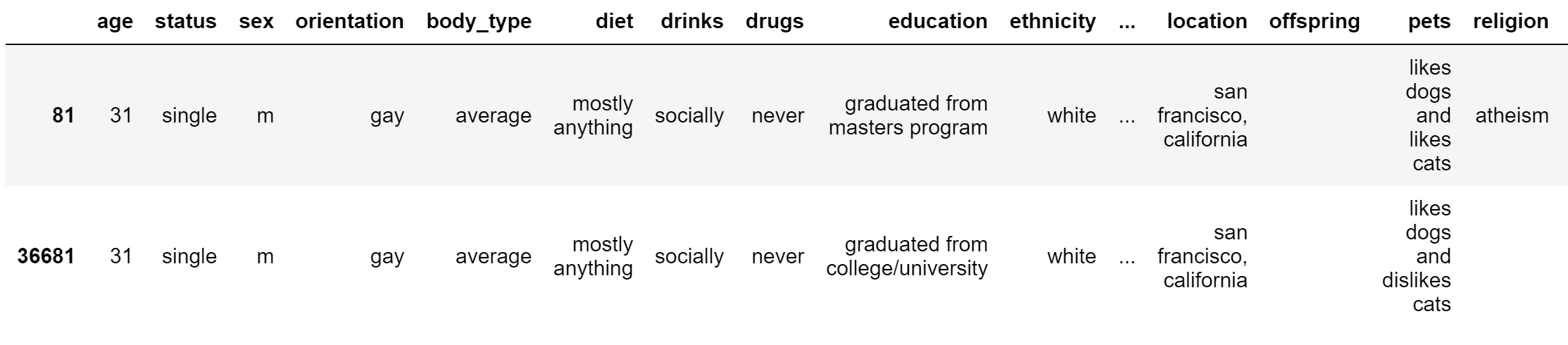


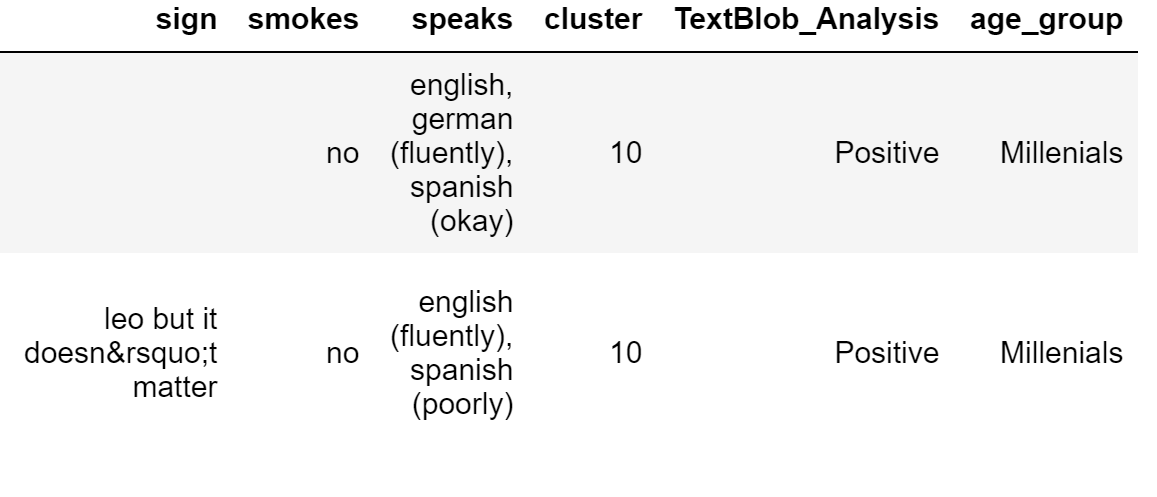


Next, we wanted to explore different demographic’s sentiment behind the combined essays. In doing so, we could determine if a profile had a positive or negative sentiment throughout their self descriptive profile. We utilized the TextBlob package, which returns a polarity and subjectivity score. We opted to use TextBlob’s default sentiment analysis implementation called PatternAnalyzer. In creating our sentiment analysis function, if the polarity score returned a positive number, then we categorized the profile as “Positive”, if the polarity score returned a negative number, then we categorized the profile as “Negative”, and if the polarity score equalled to zero, then we categorized the profile as “Neutral”. Looking at the overall results, 55,322 profiles were positive, 2,731 profiles were neutral, and 1,521 profiles were negative. This is unsurprising as most users would want to exude a positive profile in the online dating atmosphere. When comparing between male and female profiles, females tend to have a higher subjectivity and polarity score than males do. One insight that we found interesting was that the polarity score increased along with each generation. That is, the older profiles were more positive than younger profiles - GenZ and Millennials had lower polarity scores while GenX and Baby Boomers had higher polarity scores. Furthermore, Millennials had the most Negative polarity scores, while Boomers had the least Negative polarity scores.



Finally, after we collected which cluster each profile belonged to and their sentiment category, we proceeded to predict which profile would best match with one another based on the Jaccard Similarity score. We first removed the free text response essays and tokenized essays from the dataset, as well as the subjectivity and polarity scores, last online variable and income as this variable had the most NA values. We created a matching function for profiles who identify as Straight and Gay for their sexual orientation and set conditions where cluster and their sentiment must be the same. Below are examples of running the matching algorithm with the Jaccard similarity score:



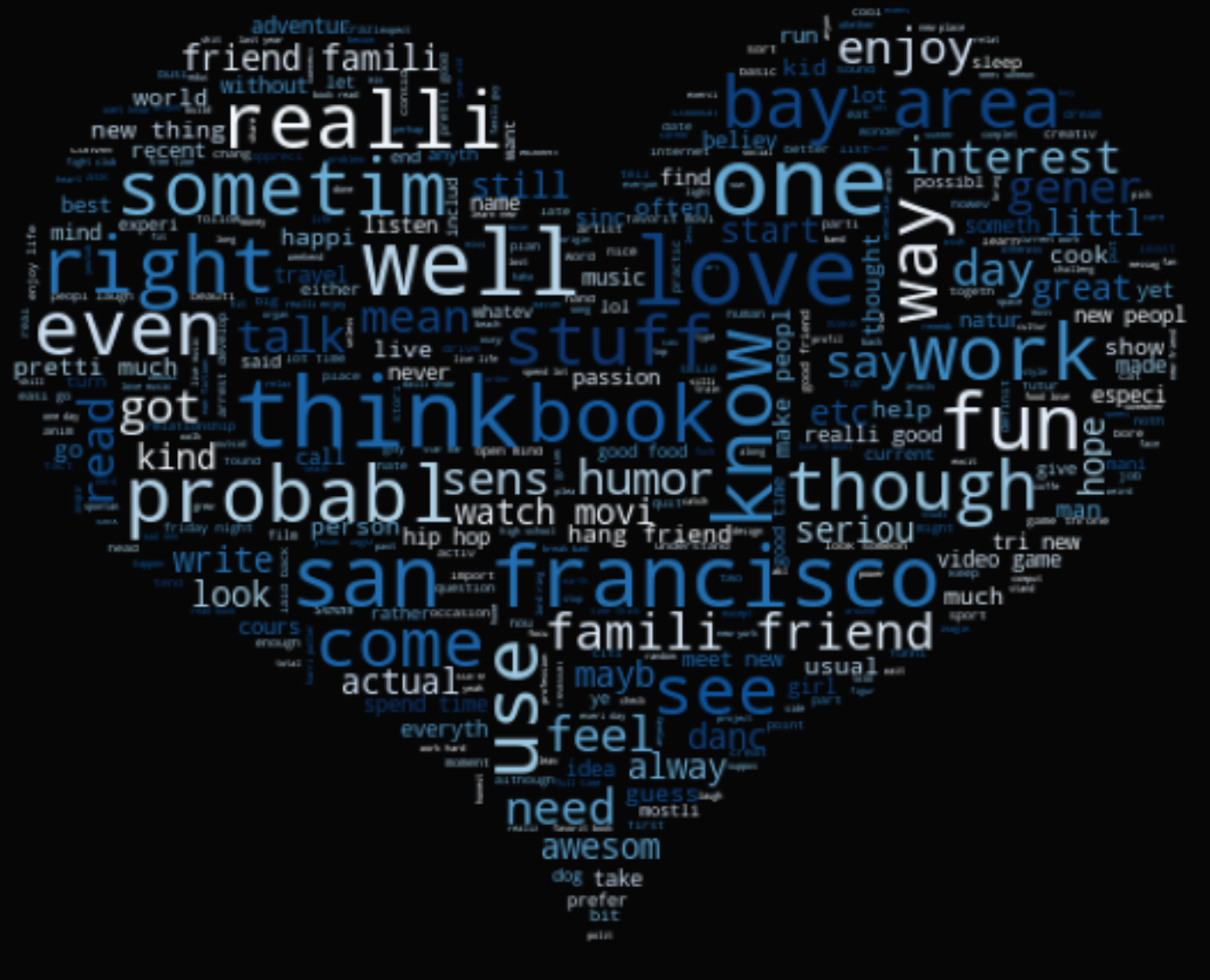






This matching algorithm took in the user id index 81 as input and since it is a gay male profile, it matched with another gay male profile with the highest Jaccard similarity score, which outputted user id index 36681. These two profiles matched with a 64% Jaccard similarity score. We also tested the matching algorithm on user id index 777 straight male and the algorithm matched with user id index 34677 straight female, with a Jaccard similarity score of 53.8%. This matching algorithm is just a simple one, however, it shows that OkCupid can include clusters from free text response essays and the overall sentiment of each user to enhance a customer's online dating experience.

We finally created 3 word clouds to explore the most frequent words found among male profiles (top left image, blue), female profiles (top right image, pink), and combined profiles (bottom image, red) as an interesting visualization (15).





# **Conclusion**

In this project, we utilized multiple NLP techniques in topic modeling to discover which model would be best for our analyses. After running LDA with CountVectorizer and NMF with TfidfVectorizer on our tokens and stemmed tokens, we concluded that NMF with TfidfVectorizer on stemmed tokens provided the most cohesive topics. From there, we were able to extract top words from essays of interest that may help users increase their chances of finding a significant other. If a user were to use these words in their essays, they may be placed into a larger cluster, giving that user a higher chance of matching with another user. On the other hand, if a user is more niche specific and is using less common words, that user will be placed in a smaller cluster similar to themself. Furthermore, within these essays of interest, we were able to cluster these profiles into meaningful topics, which showcased users’ interests, hobbies, life situations, and much more. The words that users use in their essays can play a significant part in the success of finding a significant other. In addition, we applied sentiment analysis to the profiles to explore how sentiment changes among demographic groups. Once we have obtained cluster and sentiment information, we applied our matching algorithm to include these two new attributes when finding a significant other for a specific profile.

For the purposes of the project, we only included the combined essays cluster into a matching algorithm. However, in a business sense, OkCupid can monetize this algorithm by having users pay for a premium membership. This would match users profiles based on the words they write for each essay. Users can rank the importance of each essay and from there, OkCupid can assign each essay to a cluster and provide recommendation matches to the profiles. Since the essays are free text response and go beyond descriptive characteristics, OkCupid can leverage powerful text analysis models to improve their algorithm and thus, give customers a better experience and help them find a significant other.

This dataset was only limited to profile characteristics and essays and did not include any match, like, or swipe rate. This information would be useful in creating a predictive model and finding out which attributes have the biggest impact on profile success rates.

# Bibliography

(1) (2020). Okcupid Review. Dating Sites Reviews .com. Retrieved from

<https://www.datingsitesreviews.com/staticpages/index.php?page=OkCupid-Reviews&order=DESC&mode=nested&cpage=5>

(2) Anderson. M., Vogels, E., Turner. E. (2020). Pew Research Center. Retrieved from

<https://www.pewresearch.org/internet/2020/02/06/the-virtues-and-downsides-of-online-dating/>

(3) Costa. C. (2020). CNBC. Retrieved from:

<https://www.cnbc.com/2020/03/24/how-singles-are-meeting-up-on-dating-apps-during-the-coronavirus.html>

(4) Kats. R.(2020). Business Insider. Retrieved from <https://www.businessinsider.com/dating-apps-growing-becoming-more-virtual-amid-pandemic-2020-9>

(5) Larxel. (2020). Kaggle. Retrieved from

<https://www.kaggle.com/andrewmvd/okcupid-profiles>

(6) (2020). Geeks for Geeks. Retrieved from

<https://www.geeksforgeeks.org/removing-stop-words-nltk-python/>

(7) (2020). Python Spot. Retrieved from

<https://pythonspot.com/tokenizing-words-and-sentences-with-nltk/>

(8) Aditya.B (2020). Towards Data Science. Retrieved from

<https://towardsdatascience.com/stemming-vs-lemmatization-2daddabcb221>

(9) (2020). Geeks for Geeks. Retrieved from <https://www.geeksforgeeks.org/using-countvectorizer-to-extracting-features-from-text/>

(10) (2020). Scikit Learn .org. Retrieved from <https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html>

(11) Bakharia. A. (2016). ML Review. Retrieved from

<https://medium.com/mlreview/topic-modeling-with-scikit-learn-e80d33668730>

(12) (2020). Scikit Learn .org. Retrieved from <https://scikit-learn.org/0.18/auto_examples/applications/topics_extraction_with_nmf_lda.html>

(13) Shah. P. (2020). Towards Data Science. Retrieved from

<https://towardsdatascience.com/my-absolute-go-to-for-sentiment-analysis-textblob-3ac3a11d524>

(14) Sirca. N. (2020). Towards Data Science. Retrieved from

[Finding Soul Partners Using Python | by Nurullah Sirca | Towards Data Science](https://towardsdatascience.com/finding-soul-partners-using-python-883d6017442c)

(15) Nurfikri. F. (2020). Towards Data Science. Retrieved from

<https://towardsdatascience.com/create-word-cloud-into-any-shape-you-want-using-python-d0b88834bc32>