NLP_Topic_Modeling

April 24, 2024

1 Discovering Common Themes and Language Patterns Between Dating Profiles on OKCupid Using TF-IDF and Semantic Similarity

by: Phuong Thao Nguyen

Research Focus: The aim is to determine whether certain subjects are more prevalently mentioned in dating profiles among groups differentiated by their alcohol consumption habits.

Goal: This notebook is dedicated to examining how various profiles, particularly those that participate in recreational drinking, articulate their interests. We will also investigate the feasibility of grouping these profiles by the sentiment expressed within their self-descriptions.

1.1 Introduction

As the digital dating landscape becomes ever more integral to the social fabric of modern romance, individuals take to online platforms to construct profiles that broadcast their personal hobbies, lifestyle choices, and more. The language they adopt serves not just to communicate information but also to project distinct social and personal identities. Despite the variety of expressions used across profiles, there are often linguistic commonalities among users, particularly when they align with specific interest groups, such as those related to drinking habits. This phenomenon raises the question: even as individuals strive to stand out, are there discernible patterns in language that correlate with their attitudes towards drinking?

In their comprehensive study, "Dating Apps and Their Sociodemographic and Psychosocial Correlates: A Systematic Review," Di Blasi et al. explore the notion that the behaviors and motivations driving individuals to use dating apps, like seeking sexual partnerships, may be intertwined with their drinking practices. This insight prompts us to further investigate into whether the language used by app users distinctly reflects their drinking behavior—essentially, does the verbiage in dating profiles echo the users' stance on alcohol consumption, despite the overarching intent to present a unique persona?

1.2 Citation

CHATGPT

1.3 Research Question

Research Question: Can we differentiate between frequent vs non-frequent drinkers based on their dating descriptions and how do they differ when compare to each other?

Secondary Question: What are the topics that the two groups talk about and how do they differ? How do these topics vary across each group?

Hypothesis: Based on the self-descriptions of users on OKCupid, it is possible to differentiate between them. Users who frequently drink are likely to have descriptions that reflect social outings and activities, just as those who drink less or not at all will exhibit similar traits in their profiles. Whereas infrequent drinkers or non-drinkers might emphasize interests in lifestyle or hobbies. It is expected that the content of these descriptions will gravitate towards themes reflective of their respective drinking habits.

In addition, infrequent drinkers will have more variety in their interests compare to frequent drinkers.

Secondary Hypothesis : The topics between each group will be vastly different with social activities topics in frequent groups while there will be more family or introverted activities as topics in infrequent group.

1.4 About the Data

Source: https://github.com/rudeboybert/JSE_OkCupid/blob/master/okcupid_codebook_revised.txt
https://www.kaggle.com/datasets/bryanteh/profiles-dating-app

The data consist of profiles from OKCupid that record a descriptions of user's profiles. The profiles contains the following variables:

Age: The user's age.

Body Type: A descriptive term for the user's physical shape or build.

Diet: The user's eating habits or dietary preferences.

Drinks: Frequency or preference regarding alcohol consumption by the user.

Drugs: Information regarding the use of recreational drugs by the user.

Education: The highest level of schooling or academic achievement the user has completed.

Essay0 - Essay9: The User's descriptions of themselves

Ethnicity: The user's cultural background or ethnicity.

Height: The user's height, likely in inches or centimeters.

Income: The user's annual income or salary range.

Job: The user's occupation or type of work they do.

Last Online: The last time the user was online or active on the platform.

Location: The user's current city or place of residence within California

Offspring: Information regarding whether the user has children or not, and if they want to have kids in the future

Orientation: The user's sexual orientation.

Pets: Information about whether the user has pets and what kind.

Religion: The user's religious beliefs or affiliation.

Sex: The user's biological sex or gender identity.

Sign: The user's astrological sign.

Smokes: Information regarding whether the user smokes cigarettes or tobacco.

Speaks: The languages the user speaks.

Status: The user's relationship status (e.g., single, seeing someone, married).

The main focus is the smokes and essay portion. In addition to this, the 9 essays will be combined into 1 singles variable called "combined_essay"

Separating the data into two groups There are 2 focus groups: group who drinks frequently and those who drinks infrequently. In order to analyze the semantic similarity within groups and cross groups, I will separate the profiles into 2 groups, frequent_drinking and infrequent_drinking

The dataset contains the column "drinks" which users will indicate their drinking preference. The selections are the following: often, very often, desperately, not at all, and rarely. These words are indicators of which group the profile belongs to.

frequent_drinking: ['often', 'very often', 'desperately']

infrequent_drinking: ['not at all', 'rarely']

1.5 Approaches

To examine the relationship between OKCupid users' drinking habits and the content of their self-introductions, our study will employ a three-phased analytical approach.

Logistic Regression Analysis In the initial phase, we will perform logistic regression analysis. This statistical method will enable us to determine whether the self-descriptions of users who frequently drink are significantly different from those who do not drink or drink rarely, based on their profile texts.

MNF We then move on to topic modeling to investigate if there are topics within the frequent and infrequent often talk about. Instead of choosing all of the essays and combined into one like we did for logistic regresion, in this case we will look at the text for each essay. Each essay itself is a prompt that user can answer, we only chose users who have answered all of the essay prompts. In addition, we have intentionally selected seven of the ten essays from the dataset, focusing on those that delve into personal behaviors and characteristics rather than just interests. This is the chosen essays:

essay0- My self summary

essay2- I'm really good at

essay3- The first thing people usually notice about me

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

This approach allows us to concentrate on essays that are more likely to yield insights into lifestyle choices which are crucial to our study. By excluding essays primarily centered around occupational and leisure activities (essay1, essay4, and essay9) such as "What I'm doing with my life," "Favorite books, movies, shows, music, and food," and "You should message me if…", we aim to narrow down our analysis to topics that directly involve behavioral patterns. This refined focus enhances our ability to understand how these behaviors interconnect with broader personal traits, thereby providing a deeper, more coherent analysis of the data with Matrix Factorization (MNF) and the Silhouette score.

The dataset will be refined to include only relevant columns: the 'drinks' column, which indicates the user's drinking habit, and the essay columns ('essay0' to 'essay9'), which contain the users' self-descriptive narratives. This selection was intended to capture the essence of how users present themselves in relation to their drinking habits.

[1]:
```
# Load the Drive helper
from google.colab import drive

# Below will prompt for authorization but it will make your google drive␣
 ↪available (i.e., mount your drive).
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

[ ]:
```
# !pip install spacy
```

[ ]:
```
# !pip install beautifulsoup4 spacy
```

[ ]:
```
# !pip install wordcloud
```

[ ]:
```
# !pip install gensim
```

[ ]:
```
# # !pip install spacy
# !python -m spacy download en_core_web_sm
```

[ ]:
```
#find out where you are and move to correct location
import os #package for figuring out operating system
import pandas as pd
import spacy
import numpy as np
import plotly.graph_objects as go
from bs4 import BeautifulSoup
from wordcloud import WordCloud
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
```

```python
from sklearn.decomposition import NMF
from sklearn.metrics import silhouette_score
import matplotlib.pyplot as plt
import numpy as np

# nlp = spacy.load("en_core_web_sm")

#load model
spacy.cli.download("en_core_web_lg")
nlp = spacy.load('en_core_web_lg')
```

[ ]:
```python
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
# from gensim import corpora, models
```

[ ]:
```python
import spacy
import numpy as np
from scipy.spatial.distance import cosine
from scipy.stats import ttest_ind
```

[ ]:
```python
df = pd.read_csv(
    "profiles.csv",
    encoding="ISO-8859-1",
    )
df.head(5)
```

[ ]:
```
   age        body_type               diet    drinks      drugs  \
0   22  a little extra  strictly anything  socially      never
1   35         average       mostly other     often  sometimes
2   38            thin           anything  socially        NaN
3   23            thin         vegetarian  socially        NaN
4   29        athletic                NaN  socially      never

                          education  \
0      working on college/university
1              working on space camp
2     graduated from masters program
3      working on college/university
4  graduated from college/university

                                              essay0  \
0  about me:<br />\n<br />\ni would love to think…
1  i am a chef: this is what that means.<br />\n1…
2  i'm not ashamed of much, but writing public te…
3          i work in a library and go to school. . .
4  hey how's it going? currently vague on the pro…
```

```
                                             essay1  \
0  currently working as an international agent fo…
1  dedicating everyday to being an unbelievable b…
2  i make nerdy software for musicians, artists, …
3          reading things written by old dead people
4                          work work work work + play

                                             essay2  \
0  making people laugh.<br />\nranting about a go…
1  being silly. having ridiculous amonts of fun w…
2  improvising in different contexts. alternating…
3  playing synthesizers and organizing books acco…
4  creating imagery to look at:<br />\nhttp://bag…

                                             essay3  …  \
0  the way i look. i am a six foot half asian, ha…  …
1                                                NaN  …
2  my large jaw and large glasses are the physica…  …
3                socially awkward but i do my best  …
4           i smile a lot and my inquisitive nature  …

                          location  \
0  south san francisco, california
1              oakland, california
2        san francisco, california
3             berkeley, california
4        san francisco, california

                                        offspring orientation  \
0  doesn&rsquo;t have kids, but might want them    straight
1  doesn&rsquo;t have kids, but might want them    straight
2                                            NaN    straight
3                          doesn&rsquo;t want kids    straight
4                                            NaN    straight

                        pets                                 religion sex  \
0  likes dogs and likes cats     agnosticism and very serious about it   m
1  likes dogs and likes cats  agnosticism but not too serious about it   m
2                   has cats                                       NaN   m
3                 likes cats                                       NaN   m
4  likes dogs and likes cats                                       NaN   m

                                 sign     smokes  \
0                              gemini  sometimes
1                              cancer         no
2  pisces but it doesn&rsquo;t matter         no
3                              pisces         no
```

```
4                               aquarius        no

                                                  speaks     status
0                                                english     single
1  english (fluently), spanish (poorly), french (…     single
2                                  english, french, c++   available
3                              english, german (poorly)     single
4                                                english     single

[5 rows x 31 columns]
```

2 Data Wrangling

2.0.1 Examining each of the essays

according to Kim, A. Y., & Escobedo-Land, A. (2015). OkCupid Data for Introductory Statistics and Data Science Courses. Journal of Statistics Education, 23(2). https://doi.org/10.1080/10691898.2015.11889737, the desciptions for each essay is :

"essay0- My self summary essay1- What I'm doing with my life essay2- I'm really good at essay3- The first thing people usually notice about me essay4- Favorite books, movies, show, music, and 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 essay9- You should message me if…"

Since each essay represent the promp, we want to see if all users answer the the prompt. Step includes examining whether there is a prompt that all of the users answer and keeping only that one, or if every single users need to answer that specifc prompt

[ ]:
```
#dropna
# df.dropna(subset=['drinks'], inplace=True)
df = df[['drinks',  'essay1', 'essay2', 'essay3', 'essay0', 'essay4', 'essay5',␣
 ↪'essay6', 'essay7', 'essay8', 'essay9']]
df.head(2)
```

2.0.2 Examine whether we should keep all essays or not. In this case, let's look at how many users kept their essays

In addition, there are NaN values in the essay columns, which indicate that some users prefer to answer certain questions over others. For this reason, we decided to analyze whether there is a preference for specific questions. This analysis will help us understand which questions are most popular among users. If we find that some questions are not commonly answered, we will consider removing users who only respond to these less popular questions to maintain data consistency. Conversely, if we observe a consistent proportion of responses across all questions or similar response rates, we will combine all the answers into a single string to get a comprehensive profile description.

2.0.3 Removing NAs

Here we see that there is not one single essay that we should completely removed. Hence, to avoid missing data, we shall kept only the users that answer all the essays

[ ]:
```
df = df[['drinks', 'essay0', 'essay1', 'essay2', 'essay3', 'essay4', 'essay5',␣
↪'essay6', 'essay7', 'essay8', 'essay9']].dropna()
```

2.0.4 Creating a uniform self-description for each user from all 9 different essays

In our first approach to preparing the OKCupid dataset for analysis of users' self-descriptions in relation to their drinking habits, we began a detailed cleaning process. Our goal was to clean the essays by eliminating hyperlinks and stopwords, and then standardizing the text for uniform syntax.

We observed that the essay texts contained HTML tags, remnants of user interactions via web interfaces, URLs, and other elements that could compromise data quality. To address this, we developed a function to remove these HTML tags and URLs, ensuring only authentic user narratives remained. This step was crucial for purifying the content and removing web formatting clutter.

After removing these elements, we applied text normalization to extract only alphanumeric word sequences from the essays, eliminating punctuation, special characters, and symbols. We also removed NaN values from both the essays and drinking status columns to ensure data consistency.

Some users left certain essay columns blank, indicating a preference for specific questions. We will analyze this pattern to understand which questions are most popular.

Next, we will create a 'combined essay' column, as our data includes nine different essays per user. This will provide a summary of how users describe themselves in their profiles."

[ ]:
```
#remove html, stop words, lemmantize words and keep only alphabet words
def clean_text(text):
    soup = BeautifulSoup(text, "html.parser")
    text = soup.get_text()
    doc = nlp(text)
    cleaned_text = ' '.join([token.lemma_.lower() for token in doc if not token.
↪is_stop and token.is_alpha])
    return cleaned_text

# List of essay columns
essay_columns = ['essay1', 'essay2', 'essay3', 'essay0', 'essay4', 'essay5',␣
↪'essay6', 'essay7', 'essay8', 'essay9']

# Apply the clean_text function to each essay column
for column in essay_columns:
    df[column] = df[column].apply(clean_text)

#combined essay
essay_columns = ['essay1', 'essay2', 'essay3', 'essay0', 'essay4', 'essay5',␣
↪'essay6', 'essay7', 'essay8', 'essay9']
```

```
df['cleaned_essay'] = df[essay_columns].fillna('').apply(lambda x: ' '.join(x),␣
↪axis=1)
```

[ ]:
```
df.head(10)
```

[ ]:
```
df.to_csv('profiles_cleaned.csv')
```

[ ]:
```
dataframe = pd.read_csv('profiles_cleaned.csv')
dataframe
```

[ ]:
```
       Unnamed: 0      drinks  \
0               0    socially
1               5    socially
2               9  not at all
3              10    socially
4              11    socially
…             …         …
28907       59941    socially
28908       59942       often
28909       59943  not at all
28910       59944    socially
28911       59945    socially

                                                  essay0  \
0      want sweep foot tired norm want catch coffee b…
1                                                awesome
2                                              rock bell
3      complex woman healthy self esteem intelligent …
4      know want life genuine guy like send message p…
…                                                    …
28907                seek long term connection share joy
28908                                           meh far
28909                                   similar interest
28910                         interested interesting
28911         bone opinion sense humor want meet face face

                                                  essay1  \
0      currently work international agent freight for…
1      build awesome stuff figure important have adve…
2      apartment like explore check thing like good j…
3      job sound lighting event make new friend keep …
4      currently young member internal strategy team …
…                                                    …
28907  happy time life come run ahead sound cliche li…
28908  currently finish school film production emphas…
28909  civil engineer enjoy help citizen san san fran…
28910  follow dream get dream get to protect people w…
```

```
28911  work elderly people psychotherapy case managem…

                                                 essay2  \
0      make people laugh rant good salting find simpl…
1      imagine random shit laugh aforementioned rando…
2      good find creative solution problem organize l…
3      hugging kiss laugh motivate people massage coo…
4      good little bit truly excel average good area …
…                                                   …
28907  outstanding osso bucco creative thrive enjoyme…
28908  filmmake photography graphic design web design…
28909  look thing objectively get thing disagree p re…
28910                                            listen
28911  great bullshitter know people plain believe cr…

                                                 essay3  \
0      way look foot half asian half caucasian mutt m…
1                big smile ask wear blue colour contact
2                                                 short
3                                      huge goofy smile
4      way dress day hat day different tie day shoe j…
…                                                   …
28907            tell people notice smile eye way dress
28908                                         dude know
28909                           quiet environment normal
28910  hair mow dimple remember indent cheek think sm…
28911  funny sarcastic totally insulting follow reali…

                                                 essay4  \
0      book absurdistan republic mouse man book want …
1      book kill mockingbird lord ring farseer trilog…
2      like tv love summer height high angry boy love…
3      constantly read read friend describe incredibl…
4      book yes avid reader move eternal sunshine van…
…                                                   …
28907  avid movie watcher follow broadway season movi…
28908  movie hook great adventure gladiator fight clu…
28909  book game change movie bourne series action sm…
28910  begin musically right listen lot mgmt mike pos…
28911  read help kathryn stockett sooooooo gooooood f…

                                                 essay5  \
0                          food water cell phone shelter
1      like love friend family need hug human contact…
2      music guitar contrast good food bike paintbrus…
3                family friend food woman music reading
4      guitar play time get available outlet correcti…
```

```
…                                                   …
28907                       family dog italy word music
28908  iphone contact lense headphone camera tv remot…
28909  iphone friend family internet bay area sport h…
28910           music family friend basketball hoop read
28911  family friend human interaction music movie bo…

                                                  essay6  \
0                             duality humorous thing
1         contribution world go breakfast love breakfast
2                                                    NaN
3      snowboarding food woman goofy nerd stuff archi…
4      little bit social influence everybody connect …
…                                                   …
28907                                         write book
28908  thinking bus work usually seat smell like urin…
28909                            aside work improve home
28910                                            chuckle
28911  sex people amazing fuck damn song stick head s…

                                                  essay7  \
0                                   try find hang club
1                                                 friend
2                                           send message
3                          have dinner drink friend work
4      hang small group friend stay go enjoy collecti…
…                                                   …
28907  run dog finish work week look forward great we…
28908               bringin home bacon drinking shakin
28909               enjoy friendly conversation dinner
28910                                day everyday friday
28911  happy hour friend run friend rant hardly worth…

                                                  essay8  \
0                     new california look wisper secret
1                cry day school bird shat head true story
2                                                     hi
3      wish jetpack blow candle birthday cake wish ar…
4               picky come date know look will waste time
…                                                   …
28907  dream sing alconquin nyc live italy cinque ter…
28908                               get tattoo waldo body
28909                                           let think
28910  like walk people house naked seriously body be…
28911          wish cry like holly hunter broadcast news

                                                  essay9  \
```

```
0      want sweep foot tired norm want catch coffee b…
1                                              awesome
2                                            rock bell
3      complex woman healthy self esteem intelligent …
4      know want life go genuine guy go like send mes…
…                                                   …
28907                seek long term connection share joy
28908                                          meh far
28909                                  similar interest
28910                            interested interesting
28911        bone opinion sense humor want meet face face

                                           cleaned_essay
0      want sweep foot tired norm want catch coffee b…
1      awesome build awesome stuff figure important h…
2      rock bell apartment like explore check thing l…
3      complex woman healthy self esteem intelligent …
4      know want life genuine guy like send message p…
…                                                   …
28907  seek long term connection share joy happy time…
28908  meh far currently finish school film productio…
28909  similar interest civil engineer enjoy help cit…
28910  interested interesting follow dream get dream …
28911  bone opinion sense humor want meet face face w…

[28912 rows x 13 columns]
```

[ ]:
```
dataframe.drinks.isna().sum()
```

[ ]:
```
0
```

[ ]:
```
dataframe = dataframe.dropna()
```

2.0.5 Process of Seprating Profiles into 2 Observation Groups

Following the consolidation of individual essays into a unified 'combined_essay' column, which streamlined the dataset and captured the essence of each user's self-description, we transitioned to a nuanced analysis of their social habits. Specifically, we segmented users based on their self-reported drinking behaviors, distinguishing between **'frequent'** and **'infrequent'** drinkers through terms like 'often' and 'rarely'.

Our objective is to curate a dataset focused exclusively on users' drinking habits and their self-descriptions, ensuring a streamlined analysis of how lifestyle choices are reflected in personal narratives.

[ ]:
```
print(f"These are the possible drinking options in OKCupid: {np.
    ↪unique(list(df['drinks']))}")
```

These are the possible drinking options in OKCupid: ['desperately' 'nan' 'not at all' 'often' 'rarely' 'socially' 'very often']

Let's look at the distribution of the of each of these options.

[ ]:
```
import plotly.express as px
```

[ ]:
```
# Assuming 'df' is your DataFrame and 'drinks' is the column of interest
item_counts = dataframe['drinks'].value_counts()

# # Convert the Series to a DataFrame
# item_counts_df = item_counts.reset_index()
# item_counts_df.columns = ['Drinking Options', 'Count']

# # Create the bar chart using Plotly Express

# fig = px.bar(item_counts_df, x='Drinking Options', y='Count',
#              title='Number of Users per Drinking Option',
#              labels={'Drinking Options': 'Drinking Options', 'Count':
 ↪'Count'})

# # Show the figure
# fig.show()
```

[ ]:
```
item_counts
```

[ ]:
```
drinks
socially       20665
rarely          3233
often           2717
not at all      1861
very often       241
desperately      195
Name: count, dtype: int64
```

The majority of the data comes from the 'socially' group where it is difficult to determine whether the drinking frequency is 'frequent' or 'infrequent' due to the lack of information about the users' social outings. Since the statement appears neutral and we aim to categorize drinking based on frequency, it is best to exclude these users.

After excluding these users, We categorized users from a dataset into two groups based on their self-reported drinking habits, identifying some as 'frequent' drinkers using terms such as 'often' and 'very often', and others as 'infrequent' drinkers using terms such as 'rarely' and 'not at all'. Subsequently, we created two distinct data subsets for these categories. We also added a new column to each subset to explicitly label users as 'frequent' or 'infrequent' drinkers, laying the groundwork for a future merge of these subsets while maintaining a clear distinction between the two groups.

However, keep in mind that when we separate the options into two groups, we lose a lot of our data. That is why we decided to review the distribution beforehand to consider splitting the data into two groups and then balancing them.

```
[ ]: import pandas as pd

# Define the categories for frequent and infrequent drinking
frequent_drinking = ['often', 'very often', 'desperately']
infrequent_drinking = ['not at all', 'rarely']

# Create two separate datasets
df_frequent = dataframe[dataframe['drinks'].isin(frequent_drinking)]
df_infrequent = dataframe[dataframe['drinks'].isin(infrequent_drinking)]

#create a new column that marks how frequent they drink, we will be combining␣
↪these data together later so we would want to categorize them
df_frequent['drink_status'] = 'frequently'
df_infrequent['drink_status'] = 'infrequently'

df_frequent = df_frequent[['drink_status','cleaned_essay', 'essay1', 'essay2',␣
↪'essay3', 'essay0', 'essay4', 'essay5', 'essay6', 'essay7', 'essay8',␣
↪'essay9'] ]
df_infrequent = df_infrequent[['drink_status','cleaned_essay', 'essay1',␣
↪'essay2', 'essay3', 'essay0', 'essay4', 'essay5', 'essay6', 'essay7',␣
↪'essay8', 'essay9'] ]
```

```
/tmp/ipykernel_11547/1733373772.py:12: SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

/tmp/ipykernel_11547/1733373772.py:13: SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
```

This segmentation into two distinct data subsets, each marked by a new column to reflect drinking frequency, laid the foundation for a comprehensive comparison. It allowed us to maintain a clear distinction between user groups, setting the stage for a deeper exploration of the intersection between personal narratives and lifestyle choices.

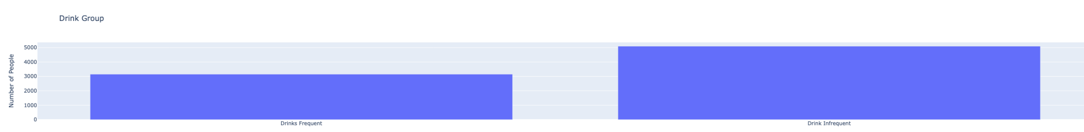
2.0.6 Balancing the Data

After dividing the two groups into those who drinks frequently and those who drinks infrequently, we observe the length of the two groups to see if there are more users in one of the two groups.

[ ]:
```
print("Length of frequent drinkers group before balance:", len(df_frequent))
print("Length of infrequent drinkers group before balance:", len(df_infrequent))
```

```
Length of frequent drinkers group before balance: 3153
Length of infrequent drinkers group before balance: 5094
```

[ ]:
```
drink_categories = ['Drinks Frequent', 'Drink Infrequent']

drink_len = [len(df_frequent), len(df_infrequent)]
# Create the bar graph
fig = go.Figure(data=[go.Bar(x=drink_categories, y=drink_len)])

# Customize the layout
fig.update_layout(
    title='Drink Group',
    xaxis_title='Drink Categories',
    yaxis_title='Number of People',
)

# Show the figure
```



Infrequent drinkers group is about 42% larger than the frequent drinkers group. Following this evaluation, we will implement a balancing technique to equalize the sizes of the two groups. This step is crucial for our subsequent analysis, as it ensures that our logistic regression model can accurately assess how self-descriptions vary between frequent and infrequent drinkers without bias toward the larger group.

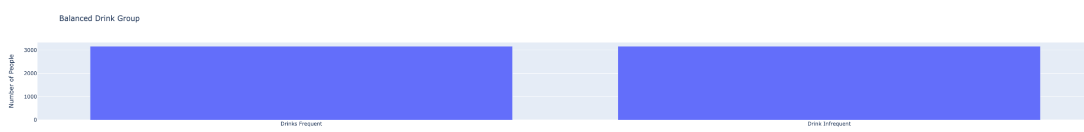
[ ]:
```
from sklearn.utils import resample

# Undersample the majority class
drinks_undersampled = resample(df_infrequent,
                                    replace=False,     # sample without replacement
                                    n_samples=len(df_frequent),     # to match minority class
                                    random_state=123) # reproducible results
```

This is our result after balancing the two groups.

```
[ ]: print("Length of frequent drinkers group after balance:", len(df_frequent))
     print("Length of infrequent drinkers groupafter balance",␣
      ↪len(drinks_undersampled))
```

```
Length of frequent drinkers group after balance: 3153
Length of infrequent drinkers groupafter balance 3153
```

```
[ ]: drink_categories = ['Drinks Frequent', 'Drink Infrequent']

     drink_len = [len(df_frequent), len(drinks_undersampled)]
     # Create the bar graph
     fig = go.Figure(data=[go.Bar(x=drink_categories, y=drink_len)])

     # Customize the layout
     fig.update_layout(
         title='Balanced Drink Group',
         xaxis_title='Drink Categories',
         yaxis_title='Number of People',

     )

     # Show the figure
```



Finally, we combined the undersampled infrequent drinkers group with the frequent drinkers group. By undersampling the larger group (infrequent drinkers) to match the size of the smaller group (frequent drinkers) and then concatenating the two, we ensured that both categories are equally represented. This balanced dataframe, now containing an equal number of users from each drinking category, sets the stage for a more unbiased analysis in our subsequent logistic regression model.

```
[ ]: # Combine the undersampled class with the minority class
     balanced_df = pd.concat([drinks_undersampled, df_frequent])
     balanced_df
```

```
[ ]:        drink_status                                     cleaned_essay  \
     8770   infrequently  want work hard industrial engineer have time l…
     21434  infrequently  deal fact open nonmonogamous relationship find…
     3045   infrequently  kind fun like think consciousness expansive fa…
     23721  infrequently  like park outside like yoga music event dance …
     20445  infrequently  like read open meet person day elementary scho…
```

```
…             …                                                …
28858    frequently  want summer madmen discussion friend workout a…
28860    frequently  chill good time journalist filmmaker interacti…
28870    frequently  think great friend love write delete try somed…
28884    frequently  like dinosaur personal shopping mean shopping …
28908    frequently  meh far currently finish school film productio…

                                                  essay1  \
8770   work hard industrial engineer have time like s…
21434  try follow heart thing happy personally profes…
3045   favorite part work train people turning point …
23721            work coffee month decide think make art
20445  day elementary school counselor aid kid good b…
…                                                  …
28858                      prepare bar drink like lawyer
28860  journalist filmmaker interactive medium produc…
28870                      love write delete try someday
28884  personal shopping mean shopping live ci tay tr…
28908  currently finish school film production emphas…

                                                  essay2  \
8770                                   maybe photography
21434  good computer bake lift heavy thing juggle fas…
3045                    validate people try good listen
23721              drawing painting talk music go upside
20445            play sport share conversation read book
…                                                  …
28858  make bad leftover delicious eat rice hand herd…
28860  make people smile read emotion competitive lol…
28870  see good people situation get excited natural …
28884           make joke watching lose tell people suck
28908  filmmake photography graphic design web design…

                                                  essay3  \
8770                  come meet new people assume unusual
21434  people notice size hard room usually eye humor…
3045                                       calm demeanor
23721                        curly hair nice friendly lay
20445     chinese eye funny notice easily visible people
…                                                  …
28858                                         height eye
28860     mannerism voice height eye color unique style
28870  mexican people hear usually assume white meet …
28884                     laugh butt expensive shoe order
28908                                          dude know

                                                  essay0  \
```

```
8770                                                    want
21434  deal fact open nonmonogamous relationship find…
3045         kind fun like think consciousness expansive
23721  like park outside like yoga music event dance …
20445                          like read open meet person
…                                                    …
28858  want summer madmen discussion friend workout a…
28860                                     chill good time
28870                                  think great friend
28884                                       like dinosaur
28908                                             meh far

                                                  essay4  \
8770   read lot list favorite author book long time m…
21434  favorite book voracious reader love read hard …
3045   anatomy epidemic robert whitaker howl move cas…
23721  music indie pop indie electronic psychedelic p…
20445  book jackie robinson story redemption song aut…
…                                                    …
28858  madmen steinbeck steak tartare good show louie…
28860  power glory native son great gatsby barry lynd…
28870  book city salt truck professor madman music bl…
28884                like tv movie music different food
28908  movie hook great adventure gladiator fight clu…

                                                  essay5  \
8770   ok silly mean like son computer ipad internet …
21434  internet access computer live kitty yes large …
3045   curiosity love potential growth hope stick ble…
23721           food water sleep caffeine sunshine music
20445        friend music book computer laughter dancing
…                                                    …
28858  live middle prefer family sunshine peace quiet…
28860                        baseball beer book laughter
28870  parent sibling friend pen paper laughter music…
28884        tv sister electricity netflix work day beer
28908  iphone contact lense headphone camera tv remot…

                                                  essay6  \
8770   life general live presence negative influence …
21434  spend great deal time think want life matter r…
3045    self improvement friend throw party spirituality
23721                                        future hold
20445                                 family life travel
…                                                    …
28858                escape timbuktu law student budget
28860                             hypothetical situation
```

```
28870                                empower oakland youth
28884                     bff dog day go wear later obama win
28908   thinking bus work usually seat smell like urin…

                                                  essay7  \
8770                                          typical life
21434   usually spend quiet night home partner recover…
3045                                     host dinner party
23721                                           try switch
20445          generally have good time friend regardless
…                                                       …
28858                                            spend cab
28860   like go till sun come narcoleptic tendency figure
28870   have drink friend catch sleep lose week readin…
28884             celebrate day rest fam puzzle like party
28908                  bringin home bacon drinking shakin

                                        essay8  \
8770                           answer honestly
21434                          huge teddy bear
3045                            know ride bike
23721                              pretty open
20445   recently cut hair year take new picture
…                                             …
28858                                site year
28860                            true love die
28870                                     sure
28884                           scared antique
28908                     get tattoo waldo body

                                                  essay9
8770                                                 want
21434   deal fact open nonmonogamous relationship find…
3045          kind fun like think consciousness expansive
23721   like go park outside like yoga go music event …
20445                             like read open meet person
…                                                       …
28858   want summer madmen discussion friend workout a…
28860                                      chill good time
28870                                    think great friend
28884                                       like dinosaur
28908                                             meh far

[6306 rows x 12 columns]
```

3 Using LogisticRegression to predict frequent vs infrequent drinker

After balancing the data, the next step we will use is the application of logistic regression with TF-IDF vectorization to discern potential variations in self-descriptions between the two groups. This step is pivotal as it complements our analysis by extracting words that are thematically specific to each group, highlighting meaningful distinctions. The successful classification of the groups by logistic regression indicates the presence of unique words in their self-descriptions, allowing us to identify and extract these words to uncover the differing thematic elements within each group's narrative.

How Logistic Regression Work Logistic regression in this context would analyze the TF-IDF transformed textual features (representing self-descriptions) alongside the corresponding labels (indicating drinking frequency) to classify users into frequent or infrequent drinking groups. By learning patterns from the TF-IDF features, logistic regression would then predict the likelihood of a user belonging to a particular drinking category based on their self-description, aiding in the identification of significant linguistic patterns that differentiate between the two groups.

Assign Creating Labels and Features First, We are extracting the 'combined_essay' column to represent users' self-descriptions assigning it to documents and 'drink_status' column to label for classification labeling. Below is one example of our label and document that will be use in classification

[ ]:
```
# Assign 'combined_essay' to documents and 'drink_status' to labels
documents = balanced_df['cleaned_essay']
labels = balanced_df['drink_status']

# Print one example of labels and description
print("Example description:", documents.iloc[0])
print("Example label:", labels.iloc[0])
```

```
Example description: want work hard industrial engineer have time like start
spend lot time read go gym day simply enjoy feel guilty able great maybe
photography come meet new people assume unusual bear prague czech republic grow
communism parent bitter regime apparently rich communist take happy child friend
fun thing rich kid poor difference recall base smart united state mental midget
respect lot money judge intelectual level like well original country actually
advantage different major attach specific class have freedom accomplish set mind
come usa seek well life come visit family russians take country means favor
communism glad spend childhood regime provide insight normally need instance
learn live horrible system victim mean find way rule law consider beneficial
help situation country good thing weird want people tolerate culture weird
standard instance freeway stop speeding audacity tell cop think number mean
speed limit think believe ticket read lot list favorite author book long time
music like kind music good movie etc food tent limit meat vegetarian ok silly
mean like son computer ipad internet camera condo renaissance health club
mention sign life general live presence negative influence past typical life
```

```
answer honestly want
Example label: infrequently
```

[ ]:
```
documents
```

[ ]:
```
8770     want work hard industrial engineer have time l…
21434    deal fact open nonmonogamous relationship find…
3045     kind fun like think consciousness expansive fa…
23721    like park outside like yoga music event dance …
20445    like read open meet person day elementary scho…
                                    …
28858    want summer madmen discussion friend workout a…
28860    chill good time journalist filmmaker interacti…
28870    think great friend love write delete try somed…
28884    like dinosaur personal shopping mean shopping …
28908    meh far currently finish school film productio…
Name: cleaned_essay, Length: 6306, dtype: object
```

3.0.1 Creating a sparse matrix to using tfidf_vectorizer

After assigning our combined_essay to documents and drink_status to label, we create a TF-IDF matrix, which is a mathematical representation of textual data, like dating profiles.

This matrix assigns scores to words based on how important they are in each profile and how unique they are across all profiles. We achieve this by using a TfidfVectorizer tool, which also removes common words that don't help us understand the differences between profiles, like "I" or "the". After fitting this vectorizer to the profile descriptions, we obtain the TF-IDF matrix, which we can then use for various analyses, such as predicting drinking habits based on profile descriptions.

Min_df and max_df are parameters used to define the vocabulary based on document frequency (DF) thresholds. These thresholds can help in filtering out terms that are too rare or too common.

If min_df is a float (between 0.0 and 1.0), it represents a proportion of documents. For example, min_df = 0.01 means "ignore terms that appear in less than 1% of the documents.

If min_df is an integer, it represents the minimum number of documents a term must appear in to be included in the vocabulary.

If max_df is a float, it represents a proportion of documents. For instance, max_df = 0.95 means "ignore terms that appear in more than 95% of the documents.

We used TF-IDF to calculating the average importance of each word (TF-IDF score) for each drinking frequency group, it identifies the top words that stand out the most for each category. This allows us to see which terms are more commonly used by users who drink frequently versus those who drink infrequently, providing insights into the language associated with different drinking behaviors on dating profiles.

[ ]:
```python
from sklearn.feature_extraction.text import TfidfVectorizer

# Initialize a TfidfVectorizer
tfidf_vectorizer = TfidfVectorizer(max_df=0.95, min_df=2, stop_words='english')
```

```
# Fit and transform the documents
tfidf_matrix = tfidf_vectorizer.fit_transform(documents)
tfidf_matrix
```

```
[ ]: <6306x26641 sparse matrix of type '<class 'numpy.float64'>'
        with 1104494 stored elements in Compressed Sparse Row format>
```

We now have our tfidf_matrix. Each row represents a document (e.g., a dating profile), and each column represents a unique word.The values in the matrix indicate the importance of each word in each document, relative to the entire collection of documents.Words that appear frequently in a document but are rare across all documents receive higher scores.

Common words like 'I' or 'the' are removed, as they don't contribute much to understanding the differences between documents.The TF-IDF matrix is useful for various analyses, such as predicting characteristics or behaviors based on textual data.

3.1 Training Model

```
[ ]: from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn import metrics

# TF-IDF features from previous steps and 'labels' is target variable
X = tfidf_matrix
y = labels

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
 ↪random_state=42)
```

```
[ ]: #define our chosen classifier
classifier = LogisticRegression(max_iter=1000)  # Increasing max_iter to ensure␣
 ↪convergence
classifier.fit(X_train, y_train)
```

```
[ ]: LogisticRegression(max_iter=1000)
```

```
[ ]: from sklearn.metrics import accuracy_score, classification_report

# Predicting the labels for the test set
y_pred = classifier.predict(X_test)

# Evaluating the classifier
print("Accuracy:", accuracy_score(y_test, y_pred))
print("Classification Report:", classification_report(y_test, y_pred))
```

```
Accuracy: 0.7630744849445324
Classification Report:               precision    recall  f1-score   support
```

```
  frequently       0.76      0.76      0.76       631
infrequently       0.76      0.76      0.76       631

    accuracy                           0.76      1262
   macro avg       0.76      0.76      0.76      1262
weighted avg       0.76      0.76      0.76      1262
```

The logistic regression model achieved an accuracy of approximately 77%, indicating its ability to correctly classify users' drinking habits based on their self-descriptions from dating profiles. The precision, recall, and F1-score metrics further support the model's effectiveness, with both frequent and infrequent drinking categories exhibiting balanced performance.

Notably, the model demonstrates slightly higher precision for infrequent drinkers but slightly higher recall for frequent drinkers, suggesting a nuanced understanding of the distinctions between the two groups. Overall, these results indicate promising predictive capabilities in discerning between frequent and infrequent drinkers using textual data from dating profiles.

3.1.1 Confusion Matrix

Next, we will create a confusion matrix to evaluate the true positives, true negatives, false positives, and false negatives. This is useful because it provides a detailed breakdown of the model's performance, allowing us to identify where the model is making correct classifications and where it may be misclassifying instances. By understanding these metrics, we can gain insights into the strengths and weaknesses of the model and make informed decisions for refining or optimizing its performance.

[ ]:
```
y_pred
```

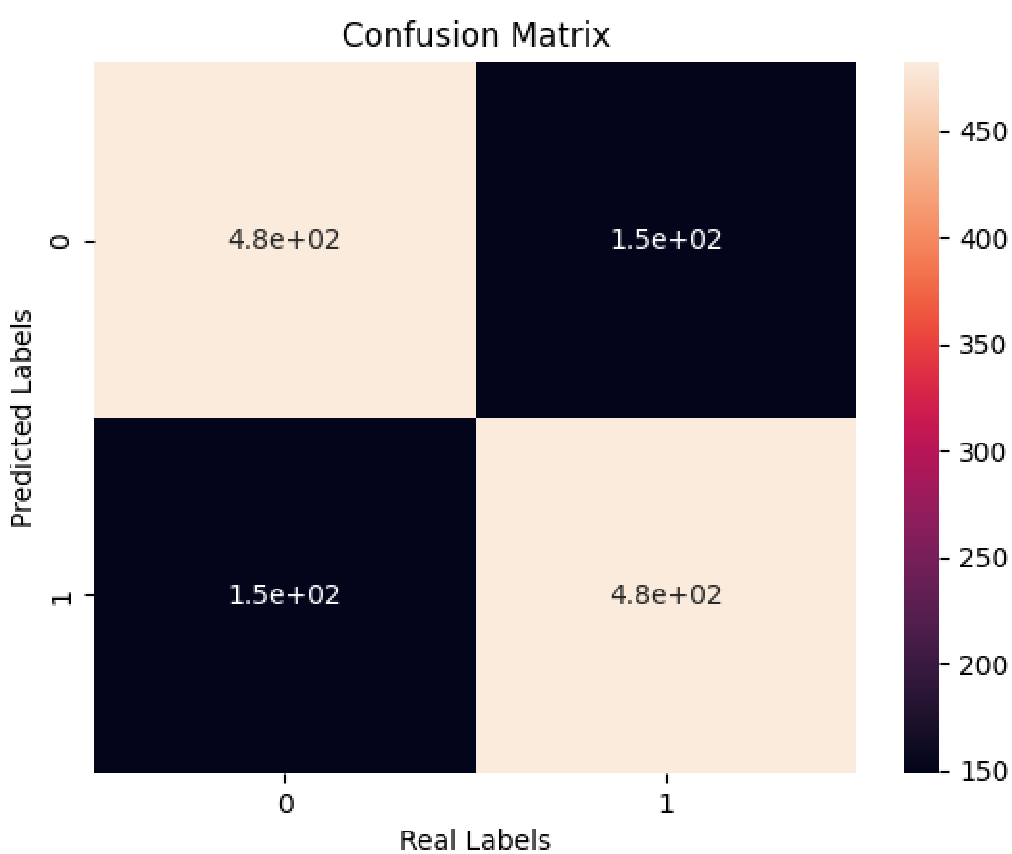
[ ]:
```
array(['infrequently', 'infrequently', 'infrequently', …,
       'infrequently', 'infrequently', 'frequently'], dtype=object)
```

[ ]:
```
y_test
```

[ ]:
```
14924    infrequently
27905    infrequently
207      infrequently
24087      frequently
24909    infrequently
             …
15585      frequently
26712      frequently
21809      frequently
177      infrequently
26593      frequently
Name: drink_status, Length: 1262, dtype: object
```

```
[ ]: import matplotlib.pyplot as plt
import seaborn as sns
# Check out a classification report
print(metrics.classification_report(y_test, y_pred))

# We can also look at incorrect predictions in a confusion matrix heatmap
cm = metrics.confusion_matrix(y_test, y_pred)
sns.heatmap(cm, annot=True)
plt.title('Confusion Matrix')
plt.xlabel('Real Labels')
plt.ylabel('Predicted Labels')
plt.show()
```

```
              precision    recall  f1-score   support

  frequently       0.76      0.76      0.76       631
infrequently       0.76      0.76      0.76       631

    accuracy                           0.76      1262
   macro avg       0.76      0.76      0.76      1262
weighted avg       0.76      0.76      0.76      1262
```



With an overall accuracy of 76%, the model demonstrates balanced precision, recall, and F1-score for both groups, indicating its effectiveness in correctly classifying instances. The macro and weighted averages further reinforce the model's consistent performance across both classes, supporting its reliability in predicting drinking habits from textual data.

This shows that the model perform relative well when classifying between frequent vs infrequent profiles, suggesting that there might be words that can identify the two groups from one another. Let's examine what are the top words within each categories

[ ]:
```
coef_df
```

[ ]:
```
       coefficients   vocabulary
0          0.161462           aa
1          0.128479          aaa
2         -0.037585      aaaaand
3         -0.052856       aaaand
4          0.004346  aaaannndddd
…               …            …
26636      0.186606           zz
26637      0.101117          zzz
26638     -0.015899         zzzz
26639      0.096221        zzzzz
26640      0.026281       zzzzzz

[26641 rows x 2 columns]
```

[ ]:
```
coef_df = pd.DataFrame({'coefficients':list(classifier.coef_.flatten()),␣
↪'vocabulary': list(pd.DataFrame(tfidf_vectorizer.vocabulary_, index=[0]).T.
↪sort_values(0).index)})
# take the lowest coefficients
lowest = coef_df.sort_values(by='coefficients').head(1000).
↪reset_index(drop=True)
lowest.columns = [col+'_1' for col in lowest.columns]

# take the highest coefficients
highest = coef_df.sort_values(by='coefficients').tail(1000).
↪sort_values(by='coefficients', ascending=False).reset_index(drop=True)
highest.columns = [col+'_2' for col in highest.columns]

# put them together to compare
pd.concat([lowest, highest], axis=1)
```

[ ]:
```
     coefficients_1 vocabulary_1  coefficients_2 vocabulary_2
0         -7.657880         beer        2.154272      healthy
1         -7.253350         wine        2.037080     computer
```

```
2          -6.678105        drink         2.013511        learn
3          -4.733220          bar         1.655301         yoga
4          -3.101497      whiskey         1.597597         drug
..               …            …             …         …
995        -0.236180          les         0.256276      anatomy
996        -0.236120  translation         0.256163        bunny
997        -0.236110    tenenbaum         0.255694    emergency
998        -0.235991     cynicism         0.255187      ireland
999        -0.235979         amon         0.255180       ethnic

[1000 rows x 4 columns]
```

[ ]:
```python
frequent_top = {word: abs(coef) for word, coef in zip(lowest['vocabulary_1'],
 ↪lowest['coefficients_1'])}
infrequent_top = {word: coef for word, coef in zip(highest['vocabulary_2'],
 ↪highest['coefficients_2'])}

# Create the word clouds
wordcloud_lowest = WordCloud(width=800, height=400, background_color='white').
 ↪generate_from_frequencies(frequent_top)
wordcloud_highest = WordCloud(width=800, height=400, background_color='white').
 ↪generate_from_frequencies(infrequent_top)

# Plot the word clouds
plt.figure(figsize=(15, 7))

plt.subplot(1, 2, 1)
plt.imshow(wordcloud_lowest)
plt.title('Frequent Words')
plt.axis('off')

plt.subplot(1, 2, 2)
plt.imshow(wordcloud_highest)
plt.title('Inrequent Words')
plt.axis('off')

plt.show()
```

Comparing the top words associated with each group provides valuable insights into the thematic differences between frequent and infrequent drinkers. The words most closely linked with the "frequently" group predominantly revolve around social interactions, drinking contexts, and, notably, contain a higher frequency of profanity. This suggests a lifestyle characterized by social engagement, recreational activities, and perhaps a more casual or expressive communication style. On the other hand, the top words associated with the "infrequently" group tend to focus more on personal well-being and relational aspects, indicating a greater emphasis on individual experiences and interpersonal relationships among infrequent drinkers. This distinction highlights the diverse self-descriptive narratives within the dataset and underscores the unique lifestyle preferences and priorities associated with each drinking frequency category.

4 Topic Modeling

essay0- My self summary

essay1- What I'm doing with my life

essay2- I'm really good at

essay3- The first thing people usually notice about me

essay4- Favorite books, movies, show, music, and 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

essay9- You should message me if…

Description here is taken from: https://github.com/rudeboybert/JSE_OkCupid/blob/master/okcupid_codebook_

In this case, I will remove

essay1- What I'm doing with my life

essay4- Favorite books, movies, show, music, and food,

essay9- You should message me if…,

as these reflect interests that is either occupational, leisure activities that might not reflect behaviors

[ ]:
```
#essays and Prompt
essay_columns = ['essay0', 'essay1', 'essay2', 'essay3', 'essay5', 'essay6',␣
 ↪'essay7', 'essay8']
prompt_titles = [
    "My self summary", "What I'm doing with my life", "I'm really good at",
    "The first thing people usually notice about me",
```

```
    "The six things I could never do without", "I spend a lot of time thinking␣
↪about",
    "On a typical Friday night I am", "The most private thing I am willing to␣
↪admit"
]
```

[ ]:
```

```

4.1 Silhouette Score

After obtaining the necessary essays, we want to understand the best number of topics for each of the essays. Traditionally, we would utilize the elbow test, which involves plotting the variation explained as a function of the number of clusters (or topics) and selecting the point where the increase in the number of clusters does not significantly improve the explained variation—the 'elbow point'. In adapting this test for Non-negative Matrix Factorization (NMF), which is typically used for topic modeling rather than clustering, we modified the approach to focus on reconstruction error, how much the model's output differs from the original data; a smaller error means the model is doing a good job of capturing the important information. In our process, we tested for a maximum of 15 topics to determine if there is an optimum point for the elbow test. However, as demonstrated by the attached reconstruction error graph, the elbow test did not yield a clear 'elbow', suggesting that continually increasing the number of topics would result in a model that is too generalized and not particularly insightful, by continuously increasing the topics, the construction error would continuous drop. This occurs for both frequent and infrequent drinkers.

[ ]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.decomposition import NMF
from sklearn.feature_extraction.text import TfidfVectorizer

def nmf_reconstruction_error(data, max_topics=15):
    tfidf_vectorizer = TfidfVectorizer(max_df=0.95, min_df=2,␣
↪stop_words='english')
    tfidf_matrix = tfidf_vectorizer.fit_transform(data.dropna())  # Handle␣
↪missing data

    reconstruction_errors = []
    for k in range(1, max_topics + 1):
        nmf_model = NMF(n_components=k, init='nndsvd', random_state=0)
        nmf_model.fit(tfidf_matrix)
        reconstruction_error = nmf_model.reconstruction_err_
        reconstruction_errors.append(reconstruction_error)
    return reconstruction_errors
```

```
essay_columns = ['essay0', 'essay2', 'essay3', 'essay5', 'essay6', 'essay7',␣
↪'essay8']
```

[ ]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.decomposition import NMF
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics import silhouette_score

def nmf_silhouette_scores(data, max_topics=15):
    tfidf_vectorizer = TfidfVectorizer(max_df=0.95, min_df=2,␣
↪stop_words='english')
    tfidf_matrix = tfidf_vectorizer.fit_transform(data.dropna())  # Handle␣
↪missing data

    silhouette_scores = []
    for k in range(2, max_topics + 1):  # Silhouette score requires at least 2␣
↪topics
        nmf_model = NMF(n_components=k, init='nndsvd', random_state=0)
        W = nmf_model.fit_transform(tfidf_matrix)  # Document-topic matrix
        pseudo_labels = np.argmax(W, axis=1)
        score = silhouette_score(tfidf_matrix, pseudo_labels,␣
↪metric='euclidean')  # Use euclidean as an example
        silhouette_scores.append(score)
    return silhouette_scores
```

In response to this, we turn to the Silhouette Score, which offers a more refined measure of how similar each object (in this case, words or phrases from the essays) is to its own cluster compared to other clusters. The Silhouette Score is calculated for each instance and can take values from -1 to +1. A high silhouette score indicates that an instance is well matched to its own cluster and poorly matched to neighboring clusters. If most objects have a high value, the clustering configuration is appropriate. If many points have a low or negative value, the clustering configuration may have too many or too few clusters. It's important to note that the Silhouette Score may yield different numbers of topics for each essay. This flexibility is advantageous as it allows for the unique content and thematic spread of each essay to determine the appropriate number of topics, rather than forcing a uniform number of topics across potentially diverse essays. This approach helps avoid overgeneralization and allows each essay's nuanced content to emerge more naturally.

[ ]:
```
import matplotlib.pyplot as plt

# Assuming essay_columns, nmf_reconstruction_error, and the dataframes are␣
↪defined

# Create a figure with two subplots (side by side)
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(20, 10))  # Adjusted for two␣
↪subplots
```

```python
# First subplot for frequent drinkers
for column in essay_columns:
    print(f"Analyzing {column}...")
    errors = nmf_reconstruction_error(df_frequent[column], max_topics=15)  #␣
↪Assuming df_frequent is defined
    ax1.plot(range(1, 16), errors, marker='o', label=column)

ax1.set_title('Elbow test of Each Essay Frequent Drinkers')
ax1.set_xlabel('Number of Topics')
ax1.set_ylabel('Reconstruction Error')
ax1.grid(True)
ax1.legend()

# First subplot for frequent drinkers
for column in essay_columns:
    if df_frequent[column].notnull().sum() > 0:  # Ensure there is data to plot
        scores = nmf_silhouette_scores(df_frequent[column], max_topics=15)
        ax2.plot(range(2, 16), scores, marker='o', label=f'{column}')  #␣
↪Starting from 2 topics

ax2.set_title('Silhouette Scores for Frequent Drinkers')
ax2.set_xlabel('Number of Topics')
ax2.set_ylabel('Silhouette Score')
ax2.legend()
ax2.grid(True)


# Display the complete figure with both subplots
plt.show()
```

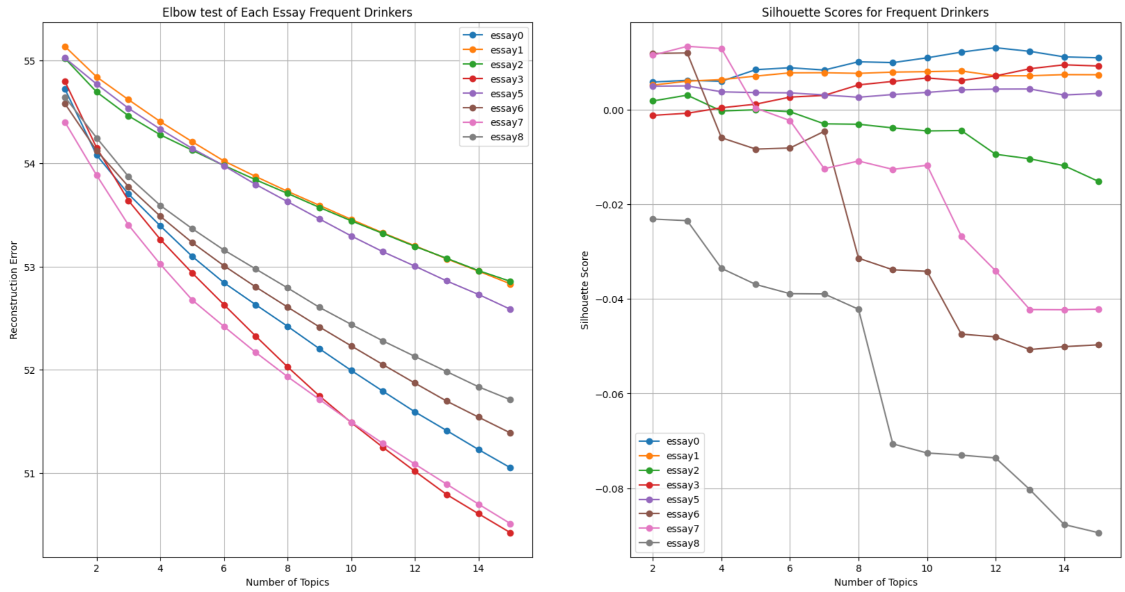
```
Analyzing essay0…
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/decomposition/_nmf.py:1770:
ConvergenceWarning:

Maximum number of iterations 200 reached. Increase it to improve convergence.


Analyzing essay1…
Analyzing essay2…
Analyzing essay3…
Analyzing essay5…
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/decomposition/_nmf.py:1770:
ConvergenceWarning:

Maximum number of iterations 200 reached. Increase it to improve convergence.
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/decomposition/_nmf.py:1770:
ConvergenceWarning:

Maximum number of iterations 200 reached. Increase it to improve convergence.

/usr/local/lib/python3.10/dist-packages/sklearn/decomposition/_nmf.py:1770:
ConvergenceWarning:

Maximum number of iterations 200 reached. Increase it to improve convergence.

/usr/local/lib/python3.10/dist-packages/sklearn/decomposition/_nmf.py:1770:
ConvergenceWarning:

Maximum number of iterations 200 reached. Increase it to improve convergence.


Analyzing essay6…
Analyzing essay7…

/usr/local/lib/python3.10/dist-packages/sklearn/decomposition/_nmf.py:1770:
ConvergenceWarning:

Maximum number of iterations 200 reached. Increase it to improve convergence.


Analyzing essay8…

/usr/local/lib/python3.10/dist-packages/sklearn/decomposition/_nmf.py:1770:
ConvergenceWarning:

Maximum number of iterations 200 reached. Increase it to improve convergence.

/usr/local/lib/python3.10/dist-packages/sklearn/decomposition/_nmf.py:1770:
ConvergenceWarning:

Maximum number of iterations 200 reached. Increase it to improve convergence.

/usr/local/lib/python3.10/dist-packages/sklearn/decomposition/_nmf.py:1770:
ConvergenceWarning:

Maximum number of iterations 200 reached. Increase it to improve convergence.

/usr/local/lib/python3.10/dist-packages/sklearn/decomposition/_nmf.py:1770:
ConvergenceWarning:

Maximum number of iterations 200 reached. Increase it to improve convergence.

/usr/local/lib/python3.10/dist-packages/sklearn/decomposition/_nmf.py:1770:
ConvergenceWarning:
```

```
Maximum number of iterations 200 reached. Increase it to improve convergence.

/usr/local/lib/python3.10/dist-packages/sklearn/decomposition/_nmf.py:1770:
ConvergenceWarning:

Maximum number of iterations 200 reached. Increase it to improve convergence.
```



[ ]:
```python
# Initialize subplots with 1 row and 2 columns
fig, axes = plt.subplots(1, 2, figsize=(20, 10))

# Second subplot for infrequent drinkers
for column in essay_columns:
    print(f"Analyzing {column}...")
    errors = nmf_reconstruction_error(df_infrequent[column], max_topics=15)
    axes[0].plot(range(1, 16), errors, marker='o', label=column)

axes[0].set_title('Elbow test of Each Essay Infrequent Drinkers')
axes[0].set_xlabel('Number of Topics')
axes[0].set_ylabel('Reconstruction Error')
axes[0].grid(True)
axes[0].legend()

# Second subplot for infrequent drinkers
for column in essay_columns:
    if df_infrequent[column].notnull().sum() > 0:  # Ensure there is data to␣
 ↪plot
        scores = nmf_silhouette_scores(df_infrequent[column], max_topics=15)
```

```
        axes[1].plot(range(2, 16), scores, marker='o', label=f'{column}')  #␣
↪Starting from 2 topics

axes[1].set_title('Silhouette Scores for Infrequent Drinkers')
axes[1].set_xlabel('Number of Topics')
axes[1].set_ylabel('Silhouette Score')
axes[1].legend()
axes[1].grid(True)

# Adjust layout for better spacing
plt.tight_layout()
plt.show()
```

```
Analyzing essay0…
Analyzing essay1…
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/decomposition/_nmf.py:1770:
ConvergenceWarning:

Maximum number of iterations 200 reached. Increase it to improve convergence.
```

```
Analyzing essay2…
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/decomposition/_nmf.py:1770:
ConvergenceWarning:

Maximum number of iterations 200 reached. Increase it to improve convergence.
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/decomposition/_nmf.py:1770:
ConvergenceWarning:

Maximum number of iterations 200 reached. Increase it to improve convergence.
```

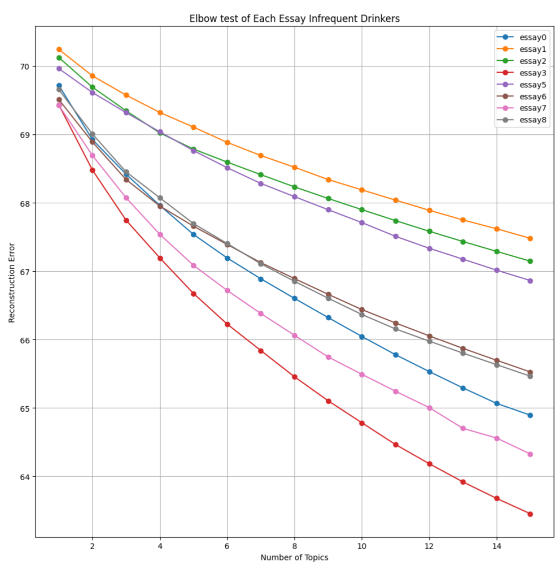
```
Analyzing essay3…
```

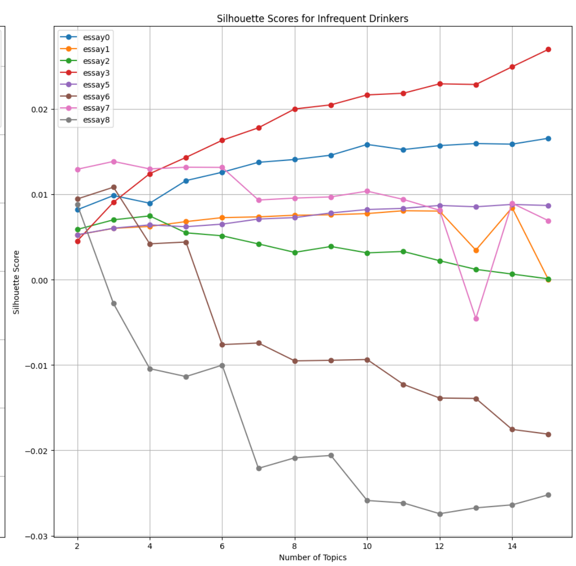
```
/usr/local/lib/python3.10/dist-packages/sklearn/decomposition/_nmf.py:1770:
ConvergenceWarning:

Maximum number of iterations 200 reached. Increase it to improve convergence.
```

```
Analyzing essay5…
Analyzing essay6…
Analyzing essay7…
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/decomposition/_nmf.py:1770:
ConvergenceWarning:
```

```
Maximum number of iterations 200 reached. Increase it to improve convergence.

Analyzing essay8…
/usr/local/lib/python3.10/dist-packages/sklearn/decomposition/_nmf.py:1770:
ConvergenceWarning:

Maximum number of iterations 200 reached. Increase it to improve convergence.

/usr/local/lib/python3.10/dist-packages/sklearn/decomposition/_nmf.py:1770:
ConvergenceWarning:

Maximum number of iterations 200 reached. Increase it to improve convergence.

/usr/local/lib/python3.10/dist-packages/sklearn/decomposition/_nmf.py:1770:
ConvergenceWarning:

Maximum number of iterations 200 reached. Increase it to improve convergence.

/usr/local/lib/python3.10/dist-packages/sklearn/decomposition/_nmf.py:1770:
ConvergenceWarning:

Maximum number of iterations 200 reached. Increase it to improve convergence.

/usr/local/lib/python3.10/dist-packages/sklearn/decomposition/_nmf.py:1770:
ConvergenceWarning:

Maximum number of iterations 200 reached. Increase it to improve convergence.
```





After determining the optimal number of topics using the Silhouette Score, we will employ Non-negative Matrix Factorization (NMF). This technique will be utilized in conjunction with the number of topics suggested by the Silhouette Score to generate the top words for each topic, thus allowing us to ascertain the themes that are prevalent within them. NMF is particularly suited for this task as it excels in decomposing high-dimensional data while maintaining the non-negativity of the data, which is inherent to word frequencies in text.

4.2 MNF

NMF is an unsupervised learning algorithm that decomposes a high-dimensional non-negative data matrix, such as a TF-IDF weighted document-term matrix from text data, into two lower-dimensional non-negative matrices. This method is especially effective in text mining and topic modeling because it helps identify patterns and topics within large collections of textual data. The process begins by transforming each essay into a vector within the TF-IDF space, ensuring each word's frequency is balanced by its commonality across all documents. NMF then factors this matrix into a document-topic matrix (W) and a topic-term matrix (H), where W illustrates how each document relates to the underlying topics, and H shows which terms are most significant for each topic. By analyzing the top terms from the H matrix, we can identify and interpret the main themes expressed by different user groups on OKCupid, providing a clear view of the prevalent topics among frequent and infrequent drinkers.

[ ]:
```
def apply_nmf(tfidf_matrix, n_topics=5):
    """Applies NMF to the given TF-IDF matrix and returns the model and the␣
↪topic-term matrix."""
    nmf_model = NMF(n_components=n_topics, random_state=42)
    W = nmf_model.fit_transform(tfidf_matrix)  # Document-topic matrix
    H = nmf_model.components_  # Topic-term matrix
    return nmf_model, W, H
```

[ ]:
```
def analyze_essay_topics_and_score(dataframe, text_column, prompt_title,␣
↪n_topics=5):
    results = {}
    print(f"\nAnalyzing {prompt_title}...")
    valid_entries = dataframe[text_column].dropna()
    if valid_entries.empty:
        print(f"No data to process in {text_column}")
        return None

    tfidf_vectorizer = TfidfVectorizer(max_df=0.95, min_df=2,␣
↪stop_words='english')
    tfidf_matrix = tfidf_vectorizer.fit_transform(valid_entries)

    nmf_model, W, H = apply_nmf(tfidf_matrix, n_topics=n_topics)

    topics = []

    feature_names = tfidf_vectorizer.get_feature_names_out()
    for i, topic_weights in enumerate(H):
```

```
        top_features_ind = topic_weights.argsort()[-10:][::-1]
        top_features = [feature_names[j] for j in top_features_ind]
        weights = topic_weights[top_features_ind]

        print(f"\nTop words for topic {i+1} in {prompt_title}:")
        print(", ".join(top_features))
        topics.append({"words": top_features, "weights": weights.tolist()})

    results['topics'] = topics
    return results
```

[ ]:
```
#essays and Prompt
essay_columns = ['essay0', 'essay1', 'essay2', 'essay3', 'essay5', 'essay6',␣
 ↪'essay7', 'essay8']
prompt_titles = [
    "My self summary", "What I’m doing with my life", "I’m really good at",
    "The first thing people usually notice about me",
    "The six things I could never do without", "I spend a lot of time thinking␣
 ↪about",
    "On a typical Friday night I am", "The most private thing I am willing to␣
 ↪admit"
]
```

4.2.1 Infrequent: TOP WORDS FOR EACH TOPIC WITHIN EACH ESSAY

[ ]:
```
analyze_essay_topics_and_score(df_infrequent, 'essay0', "My self summary",␣
 ↪n_topics=12)
analyze_essay_topics_and_score(df_infrequent, 'essay2', "I’m really good at",␣
 ↪n_topics=3)
analyze_essay_topics_and_score(df_infrequent, 'essay3', "The first thing people␣
 ↪usually notice about me", n_topics=14)
analyze_essay_topics_and_score(df_infrequent, 'essay5', "The six things I could␣
 ↪never do without", n_topics=3)
analyze_essay_topics_and_score(df_infrequent, 'essay6', "I spend a lot of time␣
 ↪thinking about", n_topics=3)
analyze_essay_topics_and_score(df_infrequent, 'essay7', "On a typical Friday␣
 ↪night I am", n_topics=3)
analyze_essay_topics_and_score(df_infrequent, 'essay8', "The most private thing␣
 ↪I am willing to admit", n_topics=2)
```

```
Analyzing My self summary…

Top words for topic 1 in My self summary:
want, hang, chat, meet, guy, learn, hi, play, real, awesome
```

Top words for topic 2 in My self summary:
like, read, profile, chat, meet, laugh, guy, sound, thing, far

Top words for topic 3 in My self summary:
friend, look, new, meet, relationship, people, guy, long, date, term

Top words for topic 4 in My self summary:
know, wanna, let, write, happen, question, difference, actually, common, great

Top words for topic 5 in My self summary:
interested, friendship, relationship, date, connection, profile, far, long, live, curious

Top words for topic 6 in My self summary:
think, match, read, common, click, cute, cool, connection, awesome, hit

Top words for topic 7 in My self summary:
feel, free, chat, right, curious, inclined, connection, common, need, write

Top words for topic 8 in My self summary:
interesting, profile, conversation, sound, person, chat, meet, curious, share, funny

Top words for topic 9 in My self summary:
message, profile, send, read, guy, hang, probably, respond, nice, shy

Top words for topic 10 in My self summary:
good, love, life, enjoy, laugh, sense, time, humor, thing, kind

Top words for topic 11 in My self summary:
talk, wanna, need, hang, bored, question, nice, let, meet, coffee

Top words for topic 12 in My self summary:
fun, person, guy, nice, hang, smart, curious, open, easy, laugh

Analyzing I'm really good at…

Top words for topic 1 in I'm really good at:
good, thing, friend, like, think, pretty, know, love, cook, lot

Top words for topic 2 in I'm really good at:
make, people, laugh, smile, feel, comfortable, care, love, talk, dancing

Top words for topic 3 in I'm really good at:
listen, friend, talk, cooking, help, advice, people, writing, love, read

Analyzing The first thing people usually notice about me…

Top words for topic 1 in The first thing people usually notice about me:
smile, lot, big, energy, dimple, warm, great, nice, attitude, face

Top words for topic 2 in The first thing people usually notice about me:
eye, green, lip, hazel, brown, voice, color, beautiful, body, big

Top words for topic 3 in The first thing people usually notice about me:
notice, people, thing, usually, think, guess, tend, comment, meet, like

Top words for topic 4 in The first thing people usually notice about me:
tell, sure, meet, maybe, nice, idea, like, people, hard, dunno

Top words for topic 5 in The first thing people usually notice about me:
hair, long, curly, red, maybe, blonde, usually, glass, color, tattoo

Top words for topic 6 in The first thing people usually notice about me:
humor, sense, style, great, good, intelligence, dry, fashion, sarcastic, sarcasm

Top words for topic 7 in The first thing people usually notice about me:
know, let, like, maybe, meet, guess, quiet, think, people, shy

Top words for topic 8 in The first thing people usually notice about me:
tall, asian, big, skinny, dark, handsome, pretty, think, funny, foot

Top words for topic 9 in The first thing people usually notice about me:
look, young, like, age, think, good, depend, people, old, year

Top words for topic 10 in The first thing people usually notice about me:
height, tattoo, voice, short, boob, maybe, lip, probably, size, pretty

Top words for topic 11 in The first thing people usually notice about me:
laugh, lot, like, love, big, loud, time, tattoo, usually, head

Top words for topic 12 in The first thing people usually notice about me:
ask, sure, idea, question, guess, maybe, lot, people, clue, depend

Top words for topic 13 in The first thing people usually notice about me:
blue, eye, glass, accent, big, comment, idea, gray, wear, long

Top words for topic 14 in The first thing people usually notice about me:
personality, probably, nice, person, good, friendly, easy, talk, lot, pretty

Analyzing The six things I could never do without…

Top words for topic 1 in The six things I could never do without:
friend, family, music, love, good, book, laughter, nature, art, coffee

Top words for topic 2 in The six things I could never do without:

```
food, water, air, shelter, sleep, good, oxygen, thing, fresh, internet

Top words for topic 3 in The six things I could never do without:
phone, computer, car, internet, cell, family, friend, tv, laptop, game

Analyzing I spend a lot of time thinking about…

Top words for topic 1 in I spend a lot of time thinking about:
think, time, thing, spend, lot, people, like, try, work, world

Top words for topic 2 in I spend a lot of time thinking about:
future, present, family, past, want, like, career, friend, hold, year

Top words for topic 3 in I spend a lot of time thinking about:
life, want, love, meaning, live, people, universe, friend, good, family

Analyzing On a typical Friday night I am…

Top words for topic 1 in On a typical Friday night I am:
friend, movie, hang, home, watch, dinner, relax, play, family, good

Top words for topic 2 in On a typical Friday night I am:
typical, friday, night, thing, like, try, time, life, day, saturday

Top words for topic 3 in On a typical Friday night I am:
work, week, usually, relax, home, day, saturday, project, sleep, weekend

Analyzing The most private thing I am willing to admit…

Top words for topic 1 in The most private thing I am willing to admit:
private, know, admit, tell, thing, willing, person, like, think, share

Top words for topic 2 in The most private thing I am willing to admit:
ask, open, tell, person, book, know, answer, want, question, pretty
```

[ ]:

```
{'topics': [{'words': ['private',
    'know',
    'admit',
    'tell',
    'thing',
    'willing',
    'person',
    'like',
    'think',
    'share'],
   'weights': [2.9595058102588174,
    1.0820265208101474,
```

```
    0.7767271729395782,
    0.6596830541836098,
    0.6188802851742765,
    0.36899562813954256,
    0.3614416649889932,
    0.347288636825546,
    0.2394593779282515,
    0.22091678488838812]},
  {'words': ['ask',
    'open',
    'tell',
    'person',
    'book',
    'know',
    'answer',
    'want',
    'question',
    'pretty'],
   'weights': [2.6667787100584954,
    0.3723582908231198,
    0.2901622832372687,
    0.2815721123105194,
    0.2431807923802003,
    0.22797082835413507,
    0.19819910450965372,
    0.14695642438684423,
    0.14467606368821848,
    0.1346989449003299]}]}
```


4.2.2 Frequent: TOP WORDS FOR EACH TOPIC WITHIN EACH ESSAY

[ ]:
```
analyze_essay_topics_and_score(df_frequent, 'essay0', "My self summary",␣
  ↪n_topics=12)
analyze_essay_topics_and_score(df_frequent, 'essay2', "I'm really good at",␣
  ↪n_topics=3)
analyze_essay_topics_and_score(df_frequent, 'essay3', "The first thing people␣
  ↪usually notice about me", n_topics=13)
analyze_essay_topics_and_score(df_frequent, 'essay5', "The six things I could␣
  ↪never do without", n_topics=3)
analyze_essay_topics_and_score(df_frequent, 'essay6', "I spend a lot of time␣
  ↪thinking about", n_topics=3)
analyze_essay_topics_and_score(df_frequent, 'essay7', "On a typical Friday␣
  ↪night I am", n_topics=3)
analyze_essay_topics_and_score(df_frequent, 'essay8', "The most private thing I␣
  ↪am willing to admit", n_topics=2)
```

Analyzing My self summary…

Top words for topic 1 in My self summary:
want, talk, hang, drink, play, adventure, movie, coffee, chat, watch

Top words for topic 2 in My self summary:
like, read, talk, people, drink, profile, laugh, beer, music, girl

Top words for topic 3 in My self summary:
good, time, love, look, life, enjoy, laugh, thing, sense, humor

Top words for topic 4 in My self summary:
know, far, tell, girl, difference, read, life, honest, mean, enjoy

Top words for topic 5 in My self summary:
think, cute, cool, handle, hit, weird, hang, friend, common, funny

Top words for topic 6 in My self summary:
fun, guy, look, smart, nice, cool, hang, happy, ready, ur

Top words for topic 7 in My self summary:
meet, new, friend, people, look, drink, try, date, person, cool

Top words for topic 8 in My self summary:
wanna, drink, hang, talk, grab, chat, coffee, cool, kick, eat

Top words for topic 9 in My self summary:
message, send, read, profile, reason, write, way, actually, tell, far

Top words for topic 10 in My self summary:
feel, right, common, like, free, compel, bite, chemistry, ya, fuck

Top words for topic 11 in My self summary:
interested, read, profile, hang, talk, awesome, relationship, coffee, maybe, date

Top words for topic 12 in My self summary:
interesting, sound, talk, cute, attractive, way, conversation, intelligent, nice, willing

Analyzing I'm really good at…

Top words for topic 1 in I'm really good at:
make, people, laugh, smile, fun, cooking, joke, feel, listen, time

Top words for topic 2 in I'm really good at:
good, friend, pretty, cook, like, time, love, think, listen, know

Top words for topic 3 in I’m really good at:
thing, fix, lot, new, learn, stuff, break, work, figure, talk

Analyzing The first thing people usually notice about me…

Top words for topic 1 in The first thing people usually notice about me:
eye, blue, green, big, guess, tattoo, leg, personality, compliment, maybe

Top words for topic 2 in The first thing people usually notice about me:
smile, lot, big, face, think, tattoo, positive, attitude, style, laughter

Top words for topic 3 in The first thing people usually notice about me:
people, notice, usually, thing, think, probably, talk, meet, pretty, good

Top words for topic 4 in The first thing people usually notice about me:
hair, red, curly, long, facial, beard, style, short, probably, guess

Top words for topic 5 in The first thing people usually notice about me:
tell, meet, sure, idea, nice, maybe, beard, voice, tattoo, hmm

Top words for topic 6 in The first thing people usually notice about me:
laugh, loud, lot, hear, love, probably, make, big, usually, time

Top words for topic 7 in The first thing people usually notice about me:
height, maybe, beard, probably, accent, body, style, voice, average, miss

Top words for topic 8 in The first thing people usually notice about me:
tall, loud, pretty, asian, handsome, dark, awesome, wear, blonde, friendly

Top words for topic 9 in The first thing people usually notice about me:
know, let, hell, maybe, care, talk, oh, personality, quiet, wish

Top words for topic 10 in The first thing people usually notice about me:
like, look, lot, think, young, guy, face, age, time, person

Top words for topic 11 in The first thing people usually notice about me:
humor, sense, style, good, personality, blue, love, talk, fashion, dry

Top words for topic 12 in The first thing people usually notice about me:
glass, wear, probably, maybe, hat, tattoo, ass, face, beard, freckle

Top words for topic 13 in The first thing people usually notice about me:
ask, sure, question, guess, probably, friend, people, say, accent, tattoo

Analyzing The six things I could never do without…

Top words for topic 1 in The six things I could never do without:
family, friend, music, internet, phone, laughter, coffee, dog, travel, computer

```
Top words for topic 2 in The six things I could never do without:
good, book, beer, wine, coffee, food, friend, thing, time, conversation

Top words for topic 3 in The six things I could never do without:
food, water, air, sex, love, shelter, music, oxygen, laughter, people

Analyzing I spend a lot of time thinking about…

Top words for topic 1 in I spend a lot of time thinking about:
think, thing, time, lot, spend, people, like, work, stuff, world

Top words for topic 2 in I spend a lot of time thinking about:
future, past, present, friend, family, plan, music, travel, sex, food

Top words for topic 3 in I spend a lot of time thinking about:
life, want, love, world, live, people, friend, universe, music, family

Analyzing On a typical Friday night I am…

Top words for topic 1 in On a typical Friday night I am:
friend, drink, home, bar, movie, watch, hang, dinner, good, usually

Top words for topic 2 in On a typical Friday night I am:
work, week, day, saturday, usually, sleep, weekend, relax, try, drinking

Top words for topic 3 in On a typical Friday night I am:
night, friday, typical, week, like, thing, usually, saturday, day, try

Analyzing The most private thing I am willing to admit…

Top words for topic 1 in The most private thing I am willing to admit:
private, admit, thing, willing, person, tell, internet, duh, share, anymore

Top words for topic 2 in The most private thing I am willing to admit:
ask, like, know, want, open, tell, pretty, think, person, book
```

[ ]:

```
{'topics': [{'words': ['private',
    'admit',
    'thing',
    'willing',
    'person',
    'tell',
    'internet',
    'duh',
    'share',
    'anymore'],
```

```
  'weights': [2.6911226703423705,
   0.8932381939663437,
   0.5356176738597267,
   0.3390376442994997,
   0.2895740824713592,
   0.1568628308817984,
   0.1272671798913233,
   0.10821319240137138,
   0.10222233076140232,
   0.09778071777946289]},
 {'words': ['ask',
   'like',
   'know',
   'want',
   'open',
   'tell',
   'pretty',
   'think',
   'person',
   'book'],
  'weights': [1.5924128101814554,
   1.0148928556603238,
   0.7936512765062358,
   0.3870874051684248,
   0.3688911938482376,
   0.3660296236148263,
   0.25840590712424805,
   0.2515351208554705,
   0.24269654683700145,
   0.22915521793230592]}]}
```

5 Analysis

The output of NMF's output top words for each of the topic for the essays. In order to retrieve a comprehensive report, we inputted the result into generative AI to formulate an ideal theme for each of our topics. Here we see that the largest number of topics involve 'my self summary' with 12 for for frequent and infrequent, and 'the first thing people noticed about me' with 13 for frequent and 14 for infrequent. These are self proclaim description of the users personality trait and on the other hand their physical traits, which makes sense as they would have more vary in the number of topics due to the user's distinct characteristics. When examine the themes between all of the essays, there is a clear indication of extroversion in frequent drinkers compare to infrequent drinkers.

6 Results, Limitation, and Next Step

This study has successfully applied natural language processing and machine learning techniques to analyze user profiles on the dating app OkCupid, focusing on how alcohol consumption influences self-descriptions. Our primary research question sought to determine whether frequent and non-frequent drinkers could be differentiated based on their profile contents. The results affirmatively indicate that there are distinct linguistic patterns and topics that correlate with users' drinking behaviors. Frequent drinkers predominantly use language that highlights social activities and extroversion, such as mentions of "social outings," "bars," and "drinking." In contrast, non-frequent drinkers' profiles are characterized by more introspective and hobby-oriented terms, showing a clear preference for "cultural activities," "reading," and "nature."

Our secondary research involved analyzing how topics extracted from profiles vary between these groups. The findings suggest that frequent drinkers are more likely to discuss themes related to large social gatherings and demonstrate a preference for dynamic social environments. Meanwhile, non-frequent drinkers focus on topics that involve one-on-one interactions and interpersonal relationships, indicating a more introverted personality. These findings align with existing literature on social behavior and alcohol consumption, which suggests that alcohol consumption can be a significant factor in socialization patterns and personal presentation, especially in socially driven contexts like dating apps.

This study has several notable limitations that should be considered when interpreting the findings. Firstly, by excluding the socially drinking group and only categorizing participants into frequent and infrequent drinkers, we significantly reduced the sample size, potentially limiting the generalizability of our results. Additionally, all participants reside in California, meaning our findings may not accurately reflect the diversity of the entire dating pool and are limited in geographical scope. The cultural homogeneity of the sample could obscure how regional and cultural differences impact self-presentation on dating platforms. Furthermore, we downsampled the infrequent drinkers, which might have skewed the comparative analysis. There is also the possibility that respondents may underreport their drinking habits, preferring not to portray themselves as heavy drinkers, which could lead to a discrepancy between reported data and actual behavior.

Looking forward, it is imperative to consider other demographic variables such as age, which might influence language style and the way individuals present themselves online. This approach can help refine our understanding of how different age groups navigate the landscape of online dating in the context of alcohol consumption. Future research should aim to include a broader demographic scope by incorporating a cross-group comparison that includes social drinkers, to identify universal versus unique thematic elements across different drinking habits. Investigating language patterns related to different substances or lifestyle choices could offer further insights into how various facets of personality or interests influence self-description. Additionally, a focused analysis on regional differences within California could reveal whether geographic location correlates with semantic similarity or shared interests, shedding light on how cultural factors influence online self-presentation. These efforts would not only enhance the depth of our understanding but also improve the applicability of our findings across more diverse populations.

[ ]:
```
# %%capture
!apt-get install texlive-xetex texlive-fonts-recommended texlive-plain-generic
!jupyter nbconvert --to pdf /content/drive/MyDrive/Colab\ Notebooks/
↪NLP_Topic_Modeling..ipynb
```

[2]:
```
# %%capture
!apt-get install texlive-xetex texlive-fonts-recommended texlive-plain-generic
```

```
Reading package lists… Done
Building dependency tree… Done
Reading state information… Done
The following additional packages will be installed:
  dvisvgm fonts-droid-fallback fonts-lato fonts-lmodern fonts-noto-mono fonts-
texgyre
  fonts-urw-base35 libapache-pom-java libcommons-logging-java libcommons-parent-
java
  libfontbox-java libfontenc1 libgs9 libgs9-common libidn12 libijs-0.35
libjbig2dec0 libkpathsea6
  libpdfbox-java libptexenc1 libruby3.0 libsynctex2 libteckit0 libtexlua53
libtexluajit2 libwoff1
  libzzip-0-13 lmodern poppler-data preview-latex-style rake ruby ruby-net-
telnet ruby-rubygems
  ruby-webrick ruby-xmlrpc ruby3.0 rubygems-integration t1utils teckit tex-
common tex-gyre
  texlive-base texlive-binaries texlive-latex-base texlive-latex-extra texlive-
latex-recommended
  texlive-pictures tipa xfonts-encodings xfonts-utils
Suggested packages:
  fonts-noto fonts-freefont-otf | fonts-freefont-ttf libavalon-framework-java
  libcommons-logging-java-doc libexcalibur-logkit-java liblog4j1.2-java poppler-
utils ghostscript
  fonts-japanese-mincho | fonts-ipafont-mincho fonts-japanese-gothic | fonts-
ipafont-gothic
  fonts-arphic-ukai fonts-arphic-uming fonts-nanum ri ruby-dev bundler debhelper
gv
  | postscript-viewer perl-tk xpdf | pdf-viewer xzdec texlive-fonts-recommended-
doc
  texlive-latex-base-doc python3-pygments icc-profiles libfile-which-perl
  libspreadsheet-parseexcel-perl texlive-latex-extra-doc texlive-latex-
recommended-doc
  texlive-luatex texlive-pstricks dot2tex prerex texlive-pictures-doc vprerex
default-jre-headless
  tipa-doc
The following NEW packages will be installed:
  dvisvgm fonts-droid-fallback fonts-lato fonts-lmodern fonts-noto-mono fonts-
texgyre
  fonts-urw-base35 libapache-pom-java libcommons-logging-java libcommons-parent-
java
  libfontbox-java libfontenc1 libgs9 libgs9-common libidn12 libijs-0.35
libjbig2dec0 libkpathsea6
  libpdfbox-java libptexenc1 libruby3.0 libsynctex2 libteckit0 libtexlua53
libtexluajit2 libwoff1
  libzzip-0-13 lmodern poppler-data preview-latex-style rake ruby ruby-net-
telnet ruby-rubygems
  ruby-webrick ruby-xmlrpc ruby3.0 rubygems-integration t1utils teckit tex-
common tex-gyre
```

```
  texlive-base texlive-binaries texlive-fonts-recommended texlive-latex-base
texlive-latex-extra
  texlive-latex-recommended texlive-pictures texlive-plain-generic texlive-xetex
tipa
  xfonts-encodings xfonts-utils
0 upgraded, 54 newly installed, 0 to remove and 45 not upgraded.
Need to get 182 MB of archives.
After this operation, 571 MB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu jammy/main amd64 fonts-droid-fallback all
1:6.0.1r16-1.1build1 [1,805 kB]
Get:2 http://archive.ubuntu.com/ubuntu jammy/main amd64 fonts-lato all 2.0-2.1
[2,696 kB]
Get:3 http://archive.ubuntu.com/ubuntu jammy/main amd64 poppler-data all
0.4.11-1 [2,171 kB]
Get:4 http://archive.ubuntu.com/ubuntu jammy/universe amd64 tex-common all 6.17
[33.7 kB]
Get:5 http://archive.ubuntu.com/ubuntu jammy/main amd64 fonts-urw-base35 all
20200910-1 [6,367 kB]
Get:6 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libgs9-common
all 9.55.0~dfsg1-0ubuntu5.6 [751 kB]
Get:7 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libidn12 amd64
1.38-4ubuntu1 [60.0 kB]
Get:8 http://archive.ubuntu.com/ubuntu jammy/main amd64 libijs-0.35 amd64
0.35-15build2 [16.5 kB]
Get:9 http://archive.ubuntu.com/ubuntu jammy/main amd64 libjbig2dec0 amd64
0.19-3build2 [64.7 kB]
Get:10 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libgs9 amd64
9.55.0~dfsg1-0ubuntu5.6 [5,031 kB]
Get:11 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libkpathsea6
amd64 2021.20210626.59705-1ubuntu0.2 [60.4 kB]
Get:12 http://archive.ubuntu.com/ubuntu jammy/main amd64 libwoff1 amd64
1.0.2-1build4 [45.2 kB]
Get:13 http://archive.ubuntu.com/ubuntu jammy/universe amd64 dvisvgm amd64
2.13.1-1 [1,221 kB]
Get:14 http://archive.ubuntu.com/ubuntu jammy/universe amd64 fonts-lmodern all
2.004.5-6.1 [4,532 kB]
Get:15 http://archive.ubuntu.com/ubuntu jammy/main amd64 fonts-noto-mono all
20201225-1build1 [397 kB]
Get:16 http://archive.ubuntu.com/ubuntu jammy/universe amd64 fonts-texgyre all
20180621-3.1 [10.2 MB]
Get:17 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libapache-pom-java
all 18-1 [4,720 B]
Get:18 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libcommons-parent-
java all 43-1 [10.8 kB]
Get:19 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libcommons-logging-
java all 1.2-2 [60.3 kB]
Get:20 http://archive.ubuntu.com/ubuntu jammy/main amd64 libfontenc1 amd64
1:1.1.4-1build3 [14.7 kB]
```

```
Get:21 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libptexenc1
amd64 2021.20210626.59705-1ubuntu0.2 [39.1 kB]
Get:22 http://archive.ubuntu.com/ubuntu jammy/main amd64 rubygems-integration
all 1.18 [5,336 B]
Get:23 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 ruby3.0 amd64
3.0.2-7ubuntu2.4 [50.1 kB]
Get:24 http://archive.ubuntu.com/ubuntu jammy/main amd64 ruby-rubygems all
3.3.5-2 [228 kB]
Get:25 http://archive.ubuntu.com/ubuntu jammy/main amd64 ruby amd64 1:3.0~exp1
[5,100 B]
Get:26 http://archive.ubuntu.com/ubuntu jammy/main amd64 rake all 13.0.6-2 [61.7
kB]
Get:27 http://archive.ubuntu.com/ubuntu jammy/main amd64 ruby-net-telnet all
0.1.1-2 [12.6 kB]
Get:28 http://archive.ubuntu.com/ubuntu jammy/universe amd64 ruby-webrick all
1.7.0-3 [51.8 kB]
Get:29 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 ruby-xmlrpc all
0.3.2-1ubuntu0.1 [24.9 kB]
Get:30 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libruby3.0
amd64 3.0.2-7ubuntu2.4 [5,113 kB]
Get:31 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libsynctex2
amd64 2021.20210626.59705-1ubuntu0.2 [55.6 kB]
Get:32 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libteckit0 amd64
2.5.11+ds1-1 [421 kB]
Get:33 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libtexlua53
amd64 2021.20210626.59705-1ubuntu0.2 [120 kB]
Get:34 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libtexluajit2
amd64 2021.20210626.59705-1ubuntu0.2 [267 kB]
Get:35 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libzzip-0-13 amd64
0.13.72+dfsg.1-1.1 [27.0 kB]
Get:36 http://archive.ubuntu.com/ubuntu jammy/main amd64 xfonts-encodings all
1:1.0.5-0ubuntu2 [578 kB]
Get:37 http://archive.ubuntu.com/ubuntu jammy/main amd64 xfonts-utils amd64
1:7.7+6build2 [94.6 kB]
Get:38 http://archive.ubuntu.com/ubuntu jammy/universe amd64 lmodern all
2.004.5-6.1 [9,471 kB]
Get:39 http://archive.ubuntu.com/ubuntu jammy/universe amd64 preview-latex-style
all 12.2-1ubuntu1 [185 kB]
Get:40 http://archive.ubuntu.com/ubuntu jammy/main amd64 t1utils amd64
1.41-4build2 [61.3 kB]
Get:41 http://archive.ubuntu.com/ubuntu jammy/universe amd64 teckit amd64
2.5.11+ds1-1 [699 kB]
Get:42 http://archive.ubuntu.com/ubuntu jammy/universe amd64 tex-gyre all
20180621-3.1 [6,209 kB]
Get:43 http://archive.ubuntu.com/ubuntu jammy-updates/universe amd64 texlive-
binaries amd64 2021.20210626.59705-1ubuntu0.2 [9,860 kB]
Get:44 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-base all
2021.20220204-1 [21.0 MB]
```

```
Get:45 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-fonts-
recommended all 2021.20220204-1 [4,972 kB]
Get:46 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-latex-base
all 2021.20220204-1 [1,128 kB]
Get:47 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libfontbox-java all
1:1.8.16-2 [207 kB]
Get:48 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libpdfbox-java all
1:1.8.16-2 [5,199 kB]
Get:49 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-latex-
recommended all 2021.20220204-1 [14.4 MB]
Get:50 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-pictures
all 2021.20220204-1 [8,720 kB]
Get:51 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-latex-extra
all 2021.20220204-1 [13.9 MB]
Get:52 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-plain-
generic all 2021.20220204-1 [27.5 MB]
Get:53 http://archive.ubuntu.com/ubuntu jammy/universe amd64 tipa all 2:1.3-21
[2,967 kB]
Get:54 http://archive.ubuntu.com/ubuntu jammy/universe amd64 texlive-xetex all
2021.20220204-1 [12.4 MB]
Fetched 182 MB in 6s (32.4 MB/s)
Extracting templates from packages: 100%
Preconfiguring packages …
Selecting previously unselected package fonts-droid-fallback.
(Reading database … 121752 files and directories currently installed.)
Preparing to unpack …/00-fonts-droid-fallback_1%3a6.0.1r16-1.1build1_all.deb
…
Unpacking fonts-droid-fallback (1:6.0.1r16-1.1build1) …
Selecting previously unselected package fonts-lato.
Preparing to unpack …/01-fonts-lato_2.0-2.1_all.deb …
Unpacking fonts-lato (2.0-2.1) …
Selecting previously unselected package poppler-data.
Preparing to unpack …/02-poppler-data_0.4.11-1_all.deb …
Unpacking poppler-data (0.4.11-1) …
Selecting previously unselected package tex-common.
Preparing to unpack …/03-tex-common_6.17_all.deb …
Unpacking tex-common (6.17) …
Selecting previously unselected package fonts-urw-base35.
Preparing to unpack …/04-fonts-urw-base35_20200910-1_all.deb …
Unpacking fonts-urw-base35 (20200910-1) …
Selecting previously unselected package libgs9-common.
Preparing to unpack …/05-libgs9-common_9.55.0~dfsg1-0ubuntu5.6_all.deb …
Unpacking libgs9-common (9.55.0~dfsg1-0ubuntu5.6) …
Selecting previously unselected package libidn12:amd64.
Preparing to unpack …/06-libidn12_1.38-4ubuntu1_amd64.deb …
Unpacking libidn12:amd64 (1.38-4ubuntu1) …
Selecting previously unselected package libijs-0.35:amd64.
Preparing to unpack …/07-libijs-0.35_0.35-15build2_amd64.deb …
```

```
Unpacking libijs-0.35:amd64 (0.35-15build2) …
Selecting previously unselected package libjbig2dec0:amd64.
Preparing to unpack …/08-libjbig2dec0_0.19-3build2_amd64.deb …
Unpacking libjbig2dec0:amd64 (0.19-3build2) …
Selecting previously unselected package libgs9:amd64.
Preparing to unpack …/09-libgs9_9.55.0~dfsg1-0ubuntu5.6_amd64.deb …
Unpacking libgs9:amd64 (9.55.0~dfsg1-0ubuntu5.6) …
Selecting previously unselected package libkpathsea6:amd64.
Preparing to unpack …/10-libkpathsea6_2021.20210626.59705-1ubuntu0.2_amd64.deb
…
Unpacking libkpathsea6:amd64 (2021.20210626.59705-1ubuntu0.2) …
Selecting previously unselected package libwoff1:amd64.
Preparing to unpack …/11-libwoff1_1.0.2-1build4_amd64.deb …
Unpacking libwoff1:amd64 (1.0.2-1build4) …
Selecting previously unselected package dvisvgm.
Preparing to unpack …/12-dvisvgm_2.13.1-1_amd64.deb …
Unpacking dvisvgm (2.13.1-1) …
Selecting previously unselected package fonts-lmodern.
Preparing to unpack …/13-fonts-lmodern_2.004.5-6.1_all.deb …
Unpacking fonts-lmodern (2.004.5-6.1) …
Selecting previously unselected package fonts-noto-mono.
Preparing to unpack …/14-fonts-noto-mono_20201225-1build1_all.deb …
Unpacking fonts-noto-mono (20201225-1build1) …
Selecting previously unselected package fonts-texgyre.
Preparing to unpack …/15-fonts-texgyre_20180621-3.1_all.deb …
Unpacking fonts-texgyre (20180621-3.1) …
Selecting previously unselected package libapache-pom-java.
Preparing to unpack …/16-libapache-pom-java_18-1_all.deb …
Unpacking libapache-pom-java (18-1) …
Selecting previously unselected package libcommons-parent-java.
Preparing to unpack …/17-libcommons-parent-java_43-1_all.deb …
Unpacking libcommons-parent-java (43-1) …
Selecting previously unselected package libcommons-logging-java.
Preparing to unpack …/18-libcommons-logging-java_1.2-2_all.deb …
Unpacking libcommons-logging-java (1.2-2) …
Selecting previously unselected package libfontenc1:amd64.
Preparing to unpack …/19-libfontenc1_1%3a1.1.4-1build3_amd64.deb …
Unpacking libfontenc1:amd64 (1:1.1.4-1build3) …
Selecting previously unselected package libptexenc1:amd64.
Preparing to unpack …/20-libptexenc1_2021.20210626.59705-1ubuntu0.2_amd64.deb
…
Unpacking libptexenc1:amd64 (2021.20210626.59705-1ubuntu0.2) …
Selecting previously unselected package rubygems-integration.
Preparing to unpack …/21-rubygems-integration_1.18_all.deb …
Unpacking rubygems-integration (1.18) …
Selecting previously unselected package ruby3.0.
Preparing to unpack …/22-ruby3.0_3.0.2-7ubuntu2.4_amd64.deb …
Unpacking ruby3.0 (3.0.2-7ubuntu2.4) …
```

```
Selecting previously unselected package ruby-rubygems.
Preparing to unpack …/23-ruby-rubygems_3.3.5-2_all.deb …
Unpacking ruby-rubygems (3.3.5-2) …
Selecting previously unselected package ruby.
Preparing to unpack …/24-ruby_1%3a3.0~exp1_amd64.deb …
Unpacking ruby (1:3.0~exp1) …
Selecting previously unselected package rake.
Preparing to unpack …/25-rake_13.0.6-2_all.deb …
Unpacking rake (13.0.6-2) …
Selecting previously unselected package ruby-net-telnet.
Preparing to unpack …/26-ruby-net-telnet_0.1.1-2_all.deb …
Unpacking ruby-net-telnet (0.1.1-2) …
Selecting previously unselected package ruby-webrick.
Preparing to unpack …/27-ruby-webrick_1.7.0-3_all.deb …
Unpacking ruby-webrick (1.7.0-3) …
Selecting previously unselected package ruby-xmlrpc.
Preparing to unpack …/28-ruby-xmlrpc_0.3.2-1ubuntu0.1_all.deb …
Unpacking ruby-xmlrpc (0.3.2-1ubuntu0.1) …
Selecting previously unselected package libruby3.0:amd64.
Preparing to unpack …/29-libruby3.0_3.0.2-7ubuntu2.4_amd64.deb …
Unpacking libruby3.0:amd64 (3.0.2-7ubuntu2.4) …
Selecting previously unselected package libsynctex2:amd64.
Preparing to unpack …/30-libsynctex2_2021.20210626.59705-1ubuntu0.2_amd64.deb
…
Unpacking libsynctex2:amd64 (2021.20210626.59705-1ubuntu0.2) …
Selecting previously unselected package libteckit0:amd64.
Preparing to unpack …/31-libteckit0_2.5.11+ds1-1_amd64.deb …
Unpacking libteckit0:amd64 (2.5.11+ds1-1) …
Selecting previously unselected package libtexlua53:amd64.
Preparing to unpack …/32-libtexlua53_2021.20210626.59705-1ubuntu0.2_amd64.deb
…
Unpacking libtexlua53:amd64 (2021.20210626.59705-1ubuntu0.2) …
Selecting previously unselected package libtexluajit2:amd64.
Preparing to unpack
…/33-libtexluajit2_2021.20210626.59705-1ubuntu0.2_amd64.deb …
Unpacking libtexluajit2:amd64 (2021.20210626.59705-1ubuntu0.2) …
Selecting previously unselected package libzzip-0-13:amd64.
Preparing to unpack …/34-libzzip-0-13_0.13.72+dfsg.1-1.1_amd64.deb …
Unpacking libzzip-0-13:amd64 (0.13.72+dfsg.1-1.1) …
Selecting previously unselected package xfonts-encodings.
Preparing to unpack …/35-xfonts-encodings_1%3a1.0.5-0ubuntu2_all.deb …
Unpacking xfonts-encodings (1:1.0.5-0ubuntu2) …
Selecting previously unselected package xfonts-utils.
Preparing to unpack …/36-xfonts-utils_1%3a7.7+6build2_amd64.deb …
Unpacking xfonts-utils (1:7.7+6build2) …
Selecting previously unselected package lmodern.
Preparing to unpack …/37-lmodern_2.004.5-6.1_all.deb …
Unpacking lmodern (2.004.5-6.1) …
```

```
Selecting previously unselected package preview-latex-style.
Preparing to unpack …/38-preview-latex-style_12.2-1ubuntu1_all.deb …
Unpacking preview-latex-style (12.2-1ubuntu1) …
Selecting previously unselected package t1utils.
Preparing to unpack …/39-t1utils_1.41-4build2_amd64.deb …
Unpacking t1utils (1.41-4build2) …
Selecting previously unselected package teckit.
Preparing to unpack …/40-teckit_2.5.11+ds1-1_amd64.deb …
Unpacking teckit (2.5.11+ds1-1) …
Selecting previously unselected package tex-gyre.
Preparing to unpack …/41-tex-gyre_20180621-3.1_all.deb …
Unpacking tex-gyre (20180621-3.1) …
Selecting previously unselected package texlive-binaries.
Preparing to unpack …/42-texlive-
binaries_2021.20210626.59705-1ubuntu0.2_amd64.deb …
Unpacking texlive-binaries (2021.20210626.59705-1ubuntu0.2) …
Selecting previously unselected package texlive-base.
Preparing to unpack …/43-texlive-base_2021.20220204-1_all.deb …
Unpacking texlive-base (2021.20220204-1) …
Selecting previously unselected package texlive-fonts-recommended.
Preparing to unpack …/44-texlive-fonts-recommended_2021.20220204-1_all.deb …
Unpacking texlive-fonts-recommended (2021.20220204-1) …
Selecting previously unselected package texlive-latex-base.
Preparing to unpack …/45-texlive-latex-base_2021.20220204-1_all.deb …
Unpacking texlive-latex-base (2021.20220204-1) …
Selecting previously unselected package libfontbox-java.
Preparing to unpack …/46-libfontbox-java_1%3a1.8.16-2_all.deb …
Unpacking libfontbox-java (1:1.8.16-2) …
Selecting previously unselected package libpdfbox-java.
Preparing to unpack …/47-libpdfbox-java_1%3a1.8.16-2_all.deb …
Unpacking libpdfbox-java (1:1.8.16-2) …
Selecting previously unselected package texlive-latex-recommended.
Preparing to unpack …/48-texlive-latex-recommended_2021.20220204-1_all.deb …
Unpacking texlive-latex-recommended (2021.20220204-1) …
Selecting previously unselected package texlive-pictures.
Preparing to unpack …/49-texlive-pictures_2021.20220204-1_all.deb …
Unpacking texlive-pictures (2021.20220204-1) …
Selecting previously unselected package texlive-latex-extra.
Preparing to unpack …/50-texlive-latex-extra_2021.20220204-1_all.deb …
Unpacking texlive-latex-extra (2021.20220204-1) …
Selecting previously unselected package texlive-plain-generic.
Preparing to unpack …/51-texlive-plain-generic_2021.20220204-1_all.deb …
Unpacking texlive-plain-generic (2021.20220204-1) …
Selecting previously unselected package tipa.
Preparing to unpack …/52-tipa_2%3a1.3-21_all.deb …
Unpacking tipa (2:1.3-21) …
Selecting previously unselected package texlive-xetex.
Preparing to unpack …/53-texlive-xetex_2021.20220204-1_all.deb …
```

```
Unpacking texlive-xetex (2021.20220204-1) …
Setting up fonts-lato (2.0-2.1) …
Setting up fonts-noto-mono (20201225-1build1) …
Setting up libwoff1:amd64 (1.0.2-1build4) …
Setting up libtexlua53:amd64 (2021.20210626.59705-1ubuntu0.2) …
Setting up libijs-0.35:amd64 (0.35-15build2) …
Setting up libtexluajit2:amd64 (2021.20210626.59705-1ubuntu0.2) …
Setting up libfontbox-java (1:1.8.16-2) …
Setting up rubygems-integration (1.18) …
Setting up libzzip-0-13:amd64 (0.13.72+dfsg.1-1.1) …
Setting up fonts-urw-base35 (20200910-1) …
Setting up poppler-data (0.4.11-1) …
Setting up tex-common (6.17) …
update-language: texlive-base not installed and configured, doing nothing!
Setting up libfontenc1:amd64 (1:1.1.4-1build3) …
Setting up libjbig2dec0:amd64 (0.19-3build2) …
Setting up libteckit0:amd64 (2.5.11+ds1-1) …
Setting up libapache-pom-java (18-1) …
Setting up ruby-net-telnet (0.1.1-2) …
Setting up xfonts-encodings (1:1.0.5-0ubuntu2) …
Setting up t1utils (1.41-4build2) …
Setting up libidn12:amd64 (1.38-4ubuntu1) …
Setting up fonts-texgyre (20180621-3.1) …
Setting up libkpathsea6:amd64 (2021.20210626.59705-1ubuntu0.2) …
Setting up ruby-webrick (1.7.0-3) …
Setting up fonts-lmodern (2.004.5-6.1) …
Setting up fonts-droid-fallback (1:6.0.1r16-1.1build1) …
Setting up ruby-xmlrpc (0.3.2-1ubuntu0.1) …
Setting up libsynctex2:amd64 (2021.20210626.59705-1ubuntu0.2) …
Setting up libgs9-common (9.55.0~dfsg1-0ubuntu5.6) …
Setting up teckit (2.5.11+ds1-1) …
Setting up libpdfbox-java (1:1.8.16-2) …
Setting up libgs9:amd64 (9.55.0~dfsg1-0ubuntu5.6) …
Setting up preview-latex-style (12.2-1ubuntu1) …
Setting up libcommons-parent-java (43-1) …
Setting up dvisvgm (2.13.1-1) …
Setting up libcommons-logging-java (1.2-2) …
Setting up xfonts-utils (1:7.7+6build2) …
Setting up libptexenc1:amd64 (2021.20210626.59705-1ubuntu0.2) …
Setting up texlive-binaries (2021.20210626.59705-1ubuntu0.2) …
update-alternatives: using /usr/bin/xdvi-xaw to provide /usr/bin/xdvi.bin
(xdvi.bin) in auto mode
update-alternatives: using /usr/bin/bibtex.original to provide /usr/bin/bibtex
(bibtex) in auto mode
Setting up lmodern (2.004.5-6.1) …
Setting up texlive-base (2021.20220204-1) …
/usr/bin/ucfr
/usr/bin/ucfr
```

```
/usr/bin/ucfr
/usr/bin/ucfr
mktexlsr: Updating /var/lib/texmf/ls-R-TEXLIVEDIST…
mktexlsr: Updating /var/lib/texmf/ls-R-TEXMFMAIN…
mktexlsr: Updating /var/lib/texmf/ls-R…
mktexlsr: Done.
tl-paper: setting paper size for dvips to a4:
/var/lib/texmf/dvips/config/config-paper.ps
tl-paper: setting paper size for dvipdfmx to a4:
/var/lib/texmf/dvipdfmx/dvipdfmx-paper.cfg
tl-paper: setting paper size for xdvi to a4: /var/lib/texmf/xdvi/XDvi-paper
tl-paper: setting paper size for pdftex to a4: /var/lib/texmf/tex/generic/tex-
ini-files/pdftexconfig.tex
Setting up tex-gyre (20180621-3.1) …
Setting up texlive-plain-generic (2021.20220204-1) …
Setting up texlive-latex-base (2021.20220204-1) …
Setting up texlive-latex-recommended (2021.20220204-1) …
Setting up texlive-pictures (2021.20220204-1) …
Setting up texlive-fonts-recommended (2021.20220204-1) …
Setting up tipa (2:1.3-21) …
Setting up texlive-latex-extra (2021.20220204-1) …
Setting up texlive-xetex (2021.20220204-1) …
Setting up rake (13.0.6-2) …
Setting up libruby3.0:amd64 (3.0.2-7ubuntu2.4) …
Setting up ruby3.0 (3.0.2-7ubuntu2.4) …
Setting up ruby (1:3.0~exp1) …
Setting up ruby-rubygems (3.3.5-2) …
Processing triggers for man-db (2.10.2-1) …
Processing triggers for fontconfig (2.13.1-4.2ubuntu5) …
Processing triggers for libc-bin (2.35-0ubuntu3.4) …
/sbin/ldconfig.real: /usr/local/lib/libtbbmalloc.so.2 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbbind_2_0.so.3 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbbind.so.3 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbb.so.12 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbbind_2_5.so.3 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbmalloc_proxy.so.2 is not a symbolic
link

Processing triggers for tex-common (6.17) …
Running updmap-sys. This may take some time… done.
Running mktexlsr /var/lib/texmf … done.
Building format(s) --all.
        This may take some time… done.
```

```
[ ]:
```