Final Report

Li Liu, Abhishek Pandit, Adam Shelton

12/7/2019

Contents

Contributions

Liu	**Pandit**	**Shelton**
Item 1	Item 1	EDA
Item 2	Item 2	AGNES
Item 3	Item 3	DBSCAN

Introduction

"So tell me about yourself!" This seemingly straightforward question in day-to-day interactions is usually met with silence and hesitation. That can no longer be the case for the 1.67 trillion online dating industry, which has grown exponentially in popularity over the last decade. The dating apps, such as OkCupid and Coffee Meets Bagel, are designed to help the singles 'get to know' other people for short or long-term romantic relationships. In order to be popular and memorable, users usually have to write a short introduction to advertise themselves. Such activity could be regarded as self-marketing. As the users of dating apps come from diverse backgrounds, we are interested in how users from distinct backgrounds take different approaches to make themselves more memorable. Moreover, we design the framework of scoring users' self-introduction and algorithm for providing writing tips (such as words for being memorable). Although our project is still preliminary, it has gained a lot of interest among our friends who struggle to find a date online. Also, our methods and analysis have the potential to be adopted by the dating website to improve the users' experience and better achieve their mission as matchmakers.

Literature Review

Self-concept and self-representation have long served as grounds of debate in cognitive and positive psychology (Bruning, Schraw, and Ronning 1999) as well as social anthropology (Goffman 1975). The recent spread of social networking and its specific affordances have allowed individuals to build different online 'selves' (Papacharissi 2010). One such critical scenario may be that of mate selection, which several economists and sociologists have likened this to 'marriage marketplace' (Hitsch, Hortaçsu, and Ariely 2010). Several online dating service providers in developed countries may facilitate the expansion of potential mates beyond the limits of even extended offline social networks Cacioppo et al. (2013) assert that as many as one in three marriages in the United States is facilitated through these portals. Heino, Ellison, and Gibbs (2010) argue that these avenues further entrench the economic dimension through an acute, implicit awareness of 'relationshopping'. Herein, potential partners are reduced to entries in a catalog to be scrolled through. In this sense, they suggest an emerging conscientiousness of 'marketing', with the product being themselves, and the potential mate assuming the role of a buyer (ibid). This perception thus links the private worlds of romantic intimacy with those of mass consumption and broader perceived appeal to the opposite sex.

Potentially, we will also use some marketing theories to understand our findings. Selling themselves and finding a mate on OkCupid is not very different from selling a product on eBay. Economists have been

Table 1: Continuous Variables

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
age	0	1	32.02	9.09	18.0	26.00	30.00	36.00	69
height	0	1	70.51	3.03	3.0	69.00	70.00	72.00	95
long_words	0	1	11.33	13.28	0.0	3.00	8.00	15.00	446
flesch	0	1	7.30	4.75	-3.6	4.86	6.73	8.96	268
profile_length	0	1	117.84	122.21	1.0	43.00	85.00	153.00	2973
prop_longwords	0	1	0.10	0.08	0.0	0.06	0.09	0.12	3

Table 2: Other Variables

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
clean_text	0	1	1	16952	0	18790	0
essay9	0	1	1	10849	0	18125	0
edu	0	1	7	21	0	3	0
fit	0	1	3	7	0	3	0
race_ethnicity	0	1	5	8	0	6	0
height_group	0	1	5	9	0	2	0

interested in the matching problem of demand and supply, such as Hitsch, Hortaçsu, and Ariely (2010). Since we do not have data on users' interactions, we will focus primarily on understanding how people brand themselves to stand out in a crowd. For example, brand awareness is a key metric in marketing to quantify the degree to which people recall or recognize a brand. A high level of brand awareness helps a product stand out and get chosen when consumers face many alternatives.

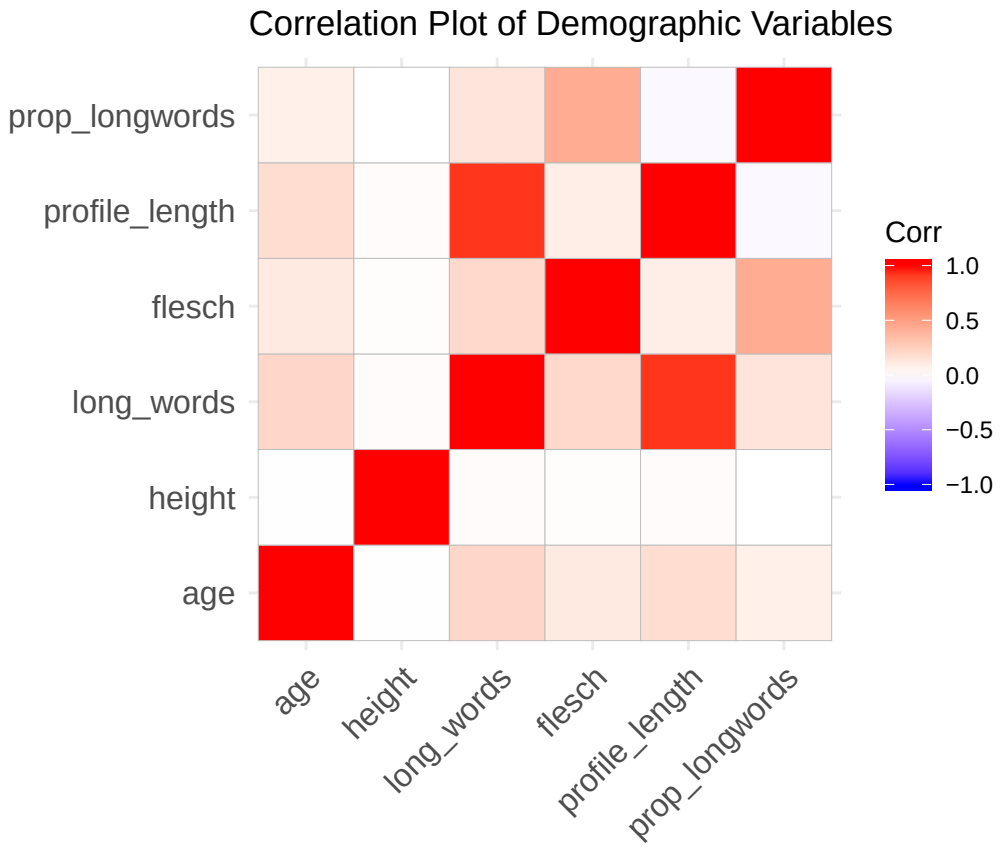
This could be applied to understand online dating. Let us imagine your future mate uses the filter to narrow down the consideration sets. He/She might still face many similar choices with high matching scores to choose from. If you want to stand out from the pool, you must make yourself memorable by highlighting the uniqueness. Thus, one possible idea in this project is to explore and understand how users could increase their brand awareness and differentiate themselves in their segments

Empirical Strategy

Analysis & Results

Exploratory Data Analysis

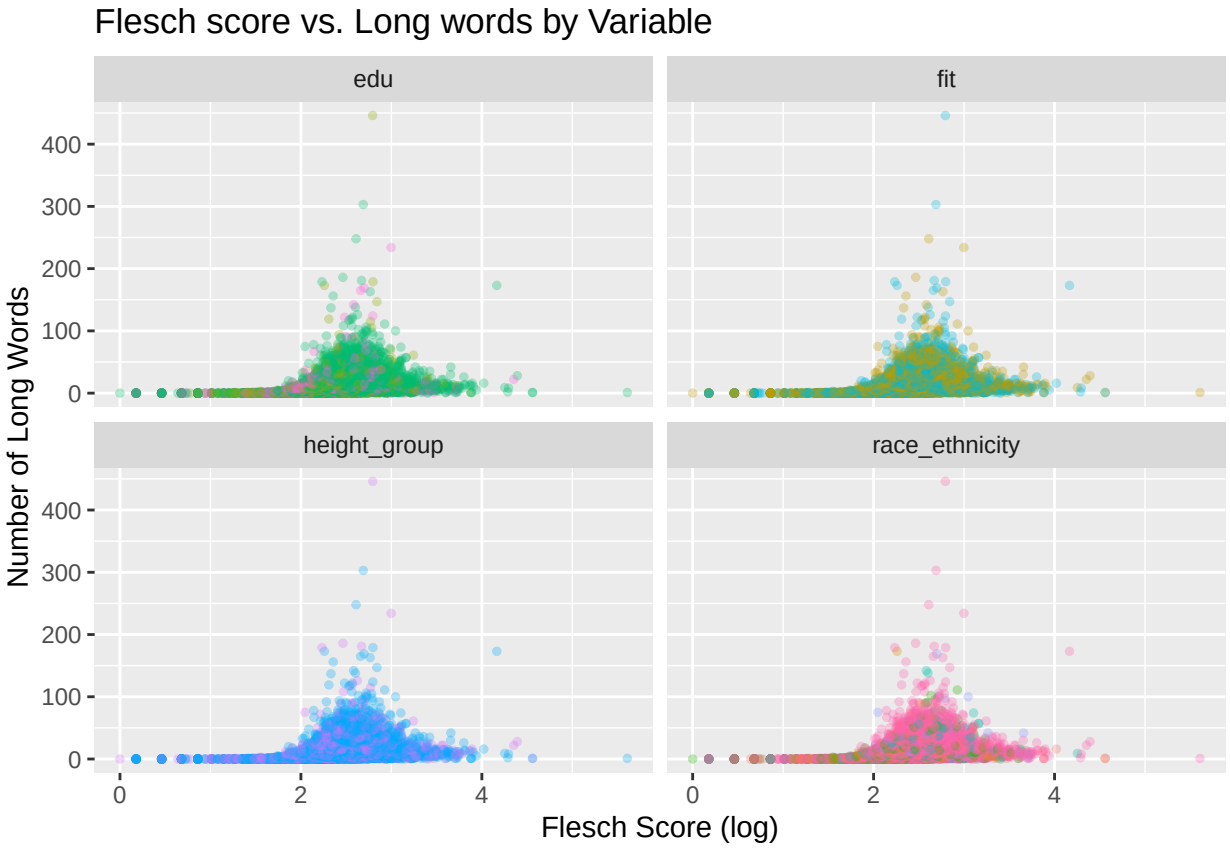
Descriptive Statistics



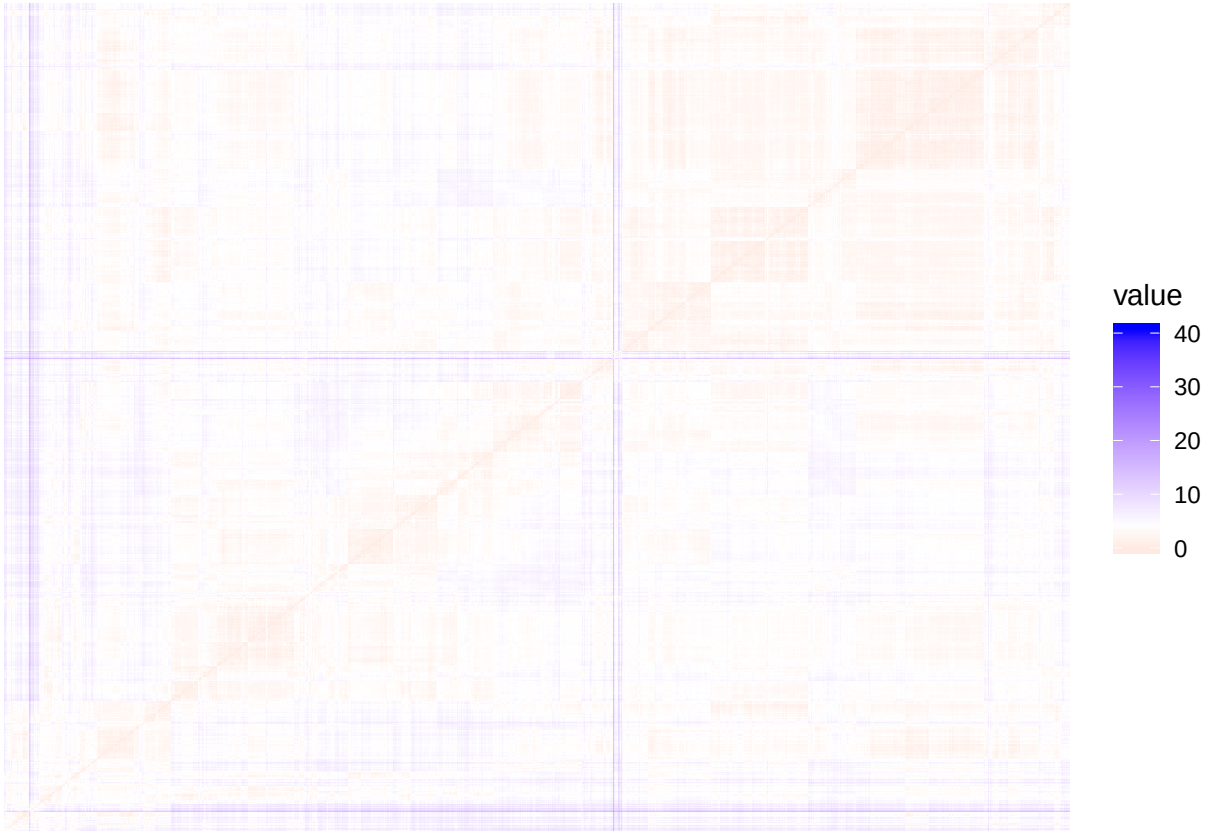
The data we used was approximately 60,000 anonymous OkCupid profiles from 2012 that were gathered with consent from users in the San Francisco area (Kim and Escobedo-Land 2015). This data was downloaded from the GitHub page for Kim and Escobedo-Land (2015), https://github.com/rudeboybert/JSE_OkCupid. The data contains demographic attributes of users that were submitted to their profile, including variables like age, height, race, and education, in addition to a selection of ten short essays that users have written in response to different prompts to display on their profiles. We subsetted this data to 18817 profiles of men, and generated additional features for the numbers and proportions of long words and Flesch–Kincaid readability scores of the main profile essay. The majority of male users in our sample or white, fit, not-short,

and have more than a high school education. The mean reported age is 32 and the mean reported height is 70.5 inches (approximately 5 foot 9 inches).

The majority of the variables included in the demographic data are independent, but some weaker correlations do exist. As expected there are positive correlations between all the features generated from the essay text. Age is also positively correlated with our text-generated features, perhaps suggesting that older people are more educated and write with more complexity. While Flesch scores and the amount of long words are correlated, there do not appear to be any demographic interactions with that relationship.


Flesch score vs. Long words by Variable
edu
fit
height_group
race_ethnicity
Number of Long Words
0
100
200
300
400
0
2
4
Flesch Score (log)

Clusterability



The demographic data is highly clusterable with a Hopkins Statistic of 0.904 on a random subset of 2,000 observations. Unfortuantely, clusterability could not be determined for the whole dataset due to performance limitations. However, 2,000 observations should be sufficient for determining clusterability.

Clustering of Demographic Data

K-means

AGNES

Agglomerative Nesting was used to cluster the demographic data with a bottom-up approach to contrast the K-means clustering. As an AGNES model would not complete on the full data-set, a random subset of 5,000 observations was used instead. This AGNES model used a Gower's dissimilarity matrix of the data to aid in the clustering of categorical variables, since most of the demographic variables are categorical. Using the `NbClust` package in R, we determined that the optimal number of clusters was two, when using the `ward.D2` method to match our AGNES model. The model gives us two well-defined clusters along the first

two principal components. The defining factor between these two clusters, majorly appears to be height, which, per the Wilcox Test, has a statistically significant difference between the means of the two clusters by about five inches.

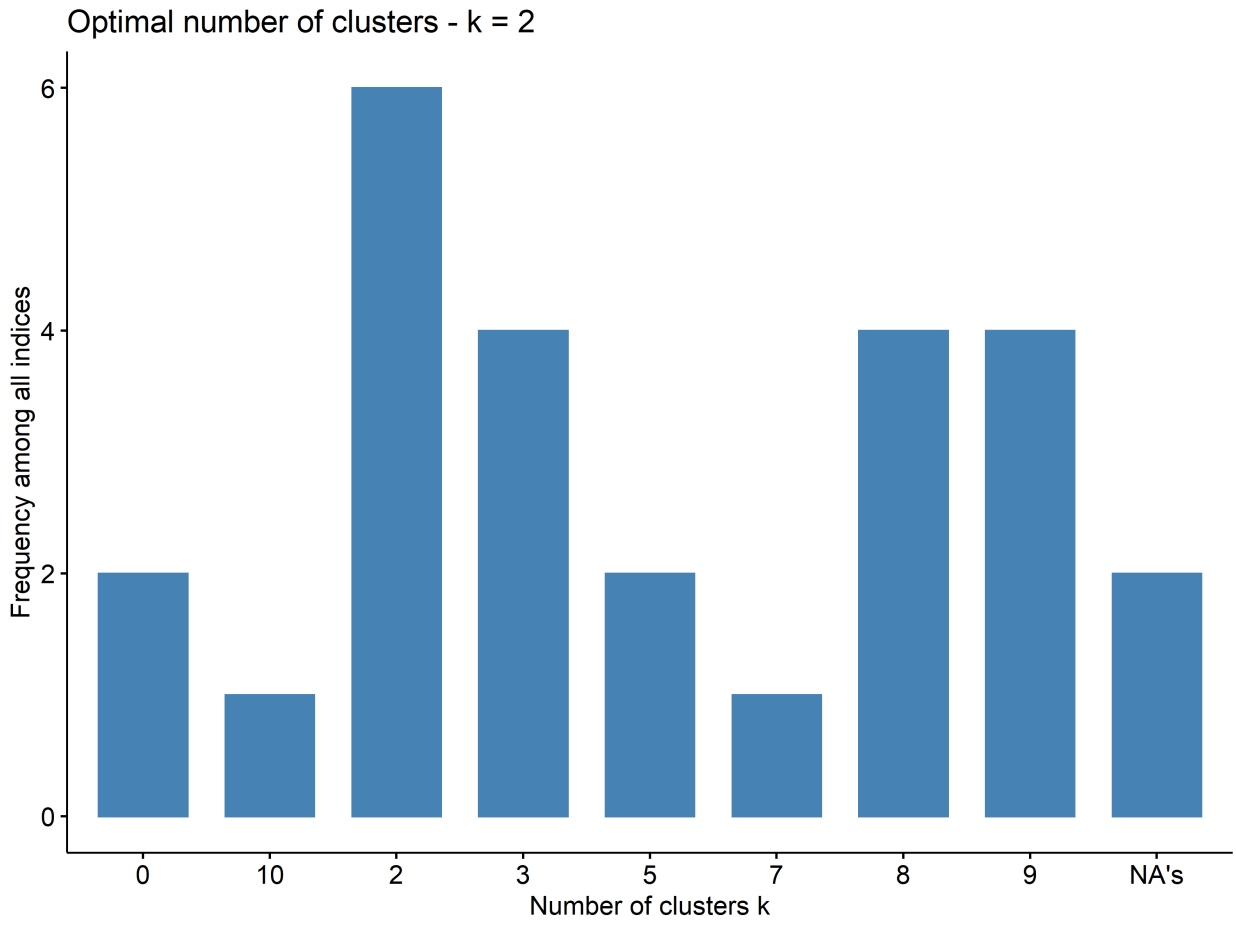


Figure 1: NbClust Results

Text Analysis

Word2Vec

Topic Modeling

DBSCAN

As we wanted to determine factors behind why a profile might stick out, we decided to use a DBSCAN model was used to detect outliers within a data-set of 50 vectors calculated by Doc2Vec. As Doc2Vec

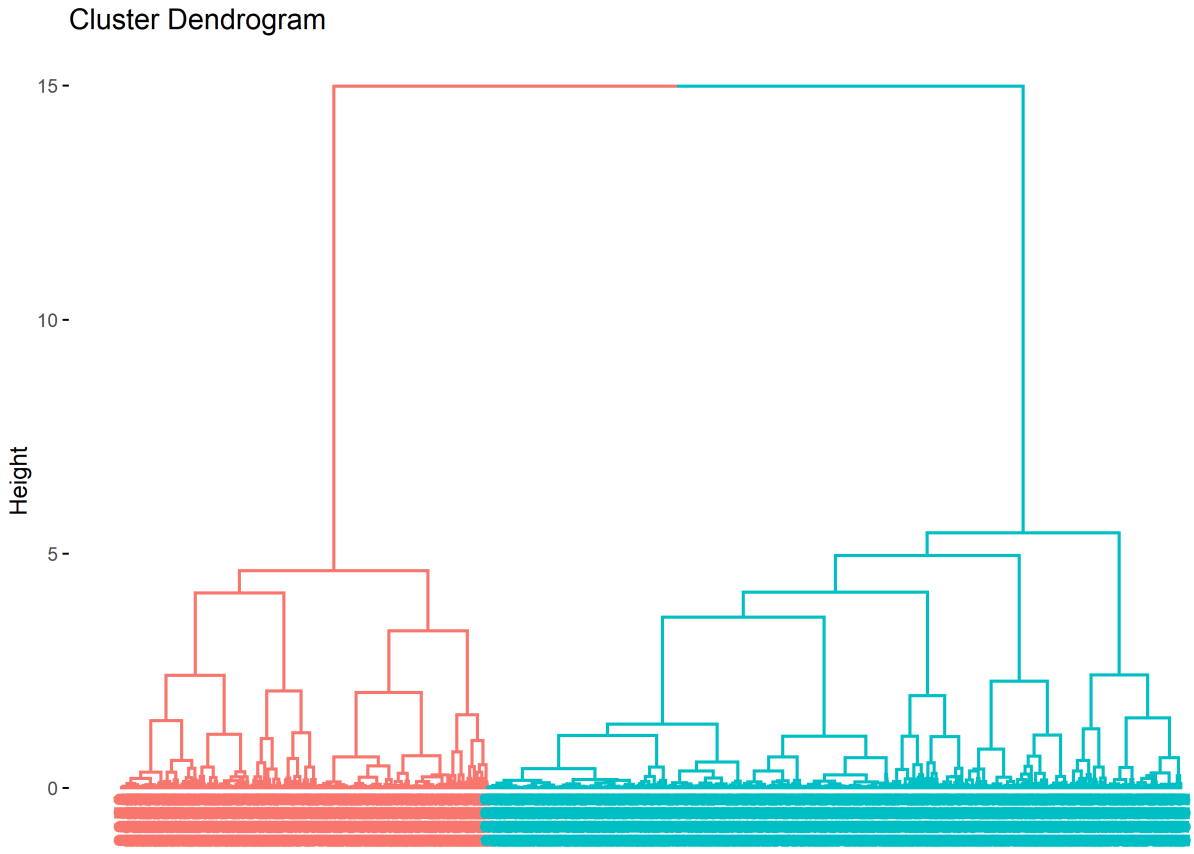


Figure 2: AGNES Dendrogram

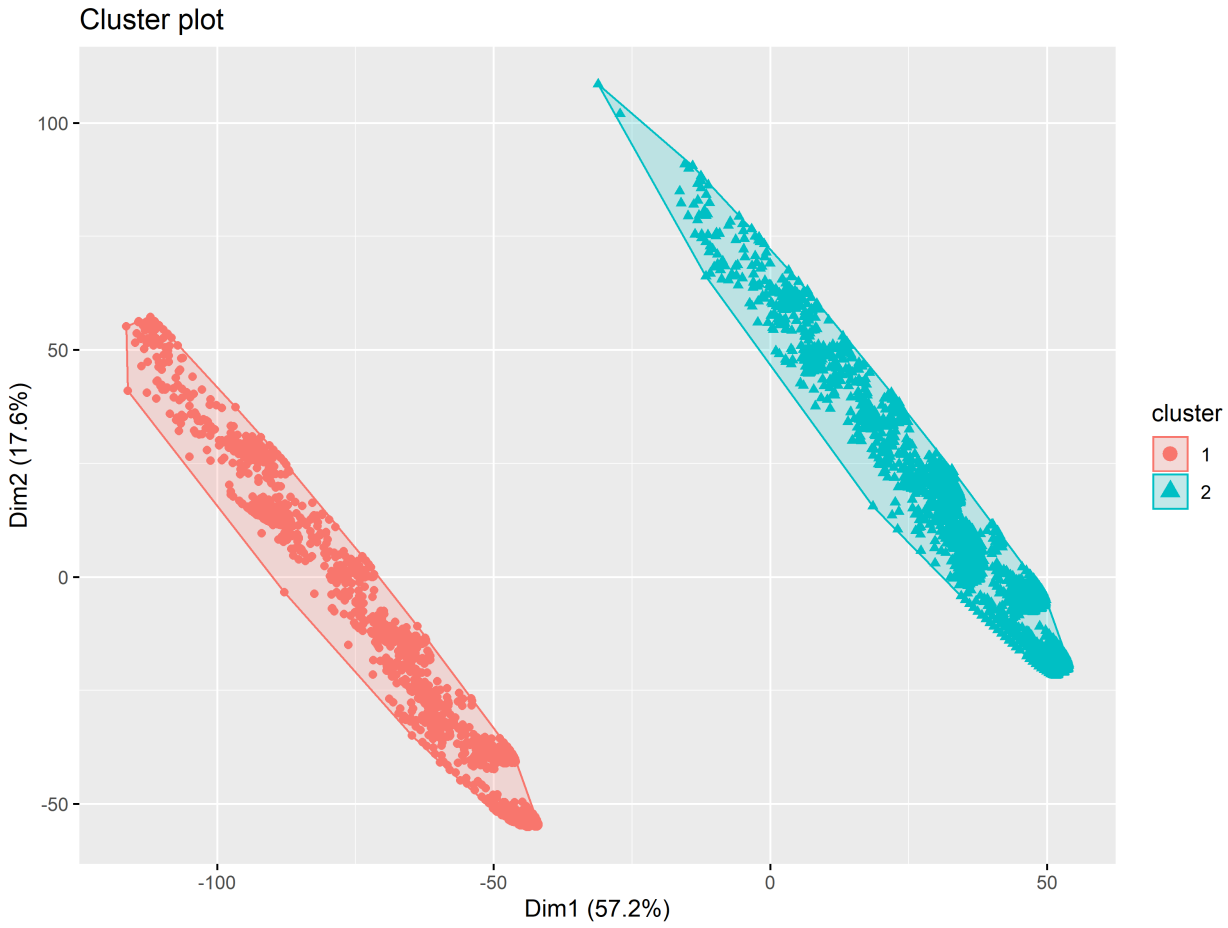


Figure 3: AGNES Cluster Results

vectors capture different characteristics about the text, any outliers in these vectors should be profiles that deviate from the norm to some degree. As shown below, these Doc2Vec vectors are highly independent with very few correlations, but also highly clusterable, with a Hopkins Statistic of 0.808.

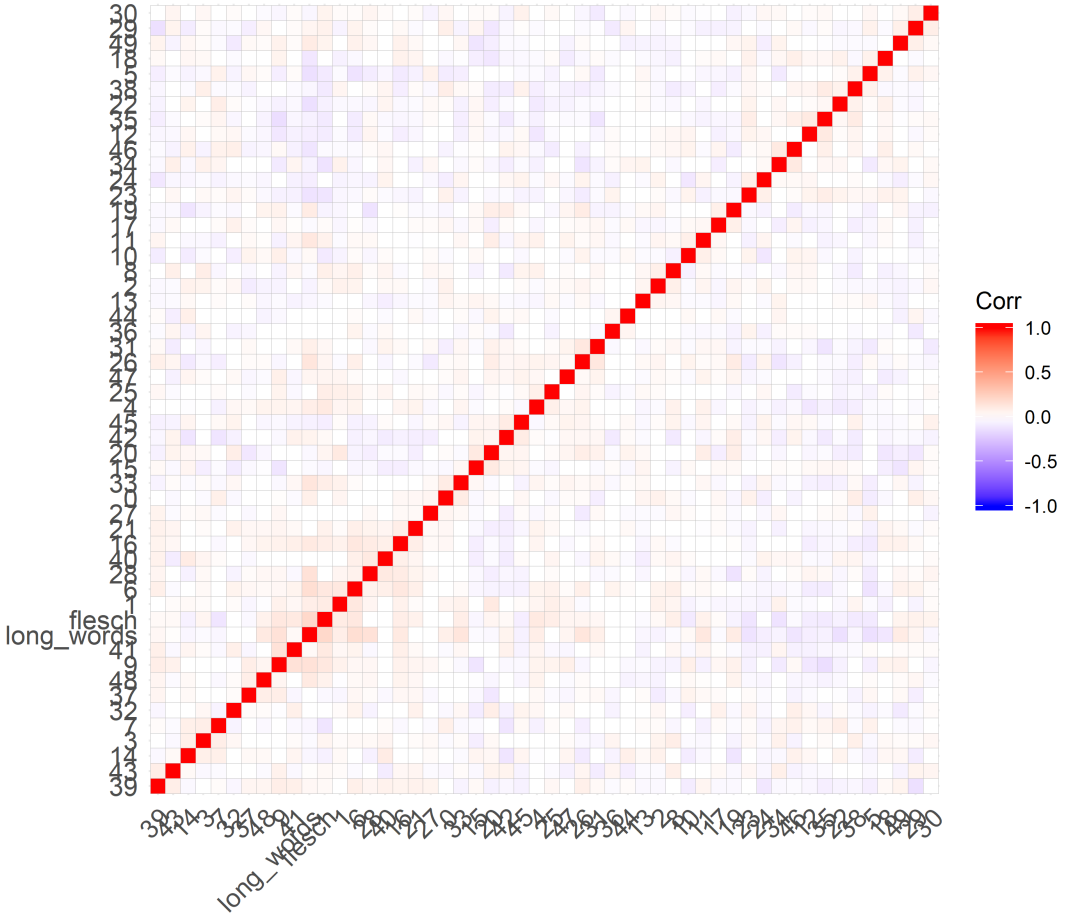


Figure 4: Doc2Vec Correlation Plot

Using a K-nearest-neighbors distribution plot, we determined that the optimal value for the epsilon neighborhood size of the DBSCAN model was 9. This was determined using 5 nearest neighbors, to match the default minimum number of points in the epsilon region, which we used for the model.

The DBSCAN model identified one cluster, and 108 outliers, accounting for 0.57% of the observations. Using a Wilcox Test, we can determine that the difference in the means between the "typical" profiles and outlier profiles are insignificant for the continuous variables for height and number of long words, but are highly significant for Flesch score and age. Mean age is higher among the outliers by about 4.5 years, while the mean Flesch score of outliers is about 8 points higher. This suggests that the outliers, being older and writing at a more advanced level, might be more educated (or at least trying to appear so) than their "typical" peers.

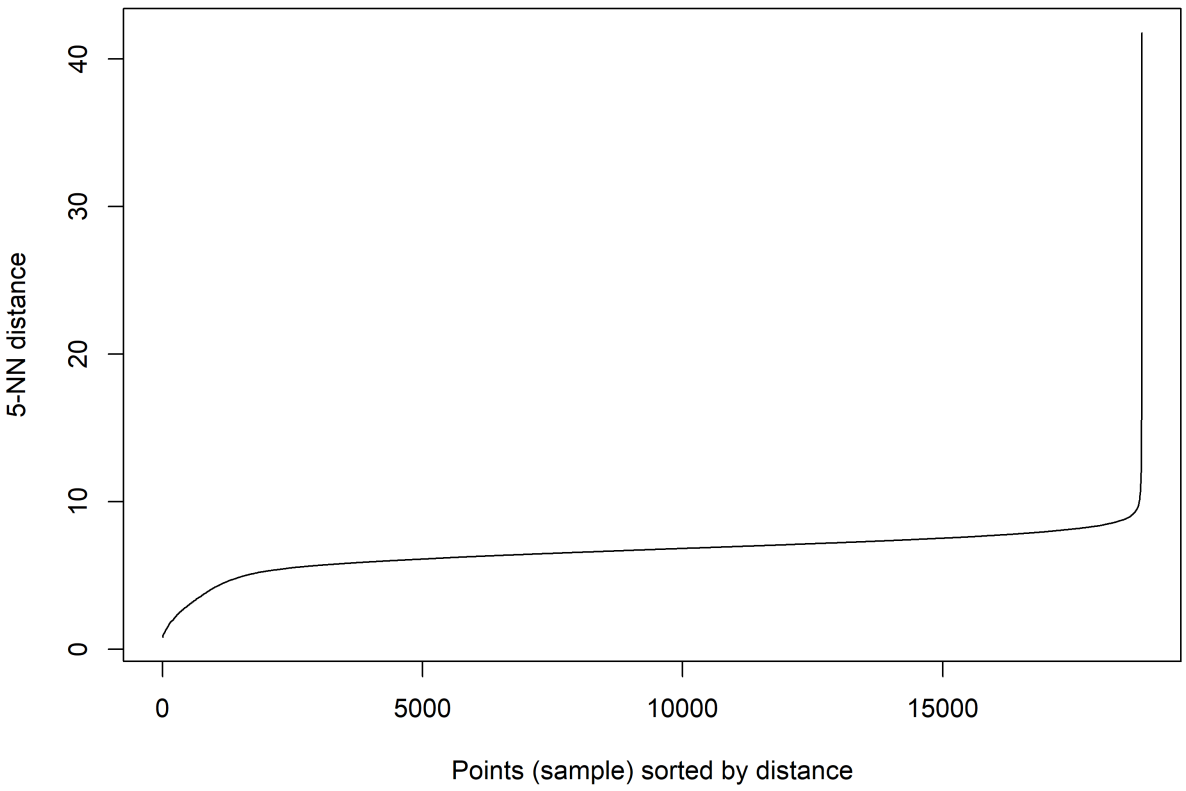


Figure 5: DBSCAN KNN Distribution Plot

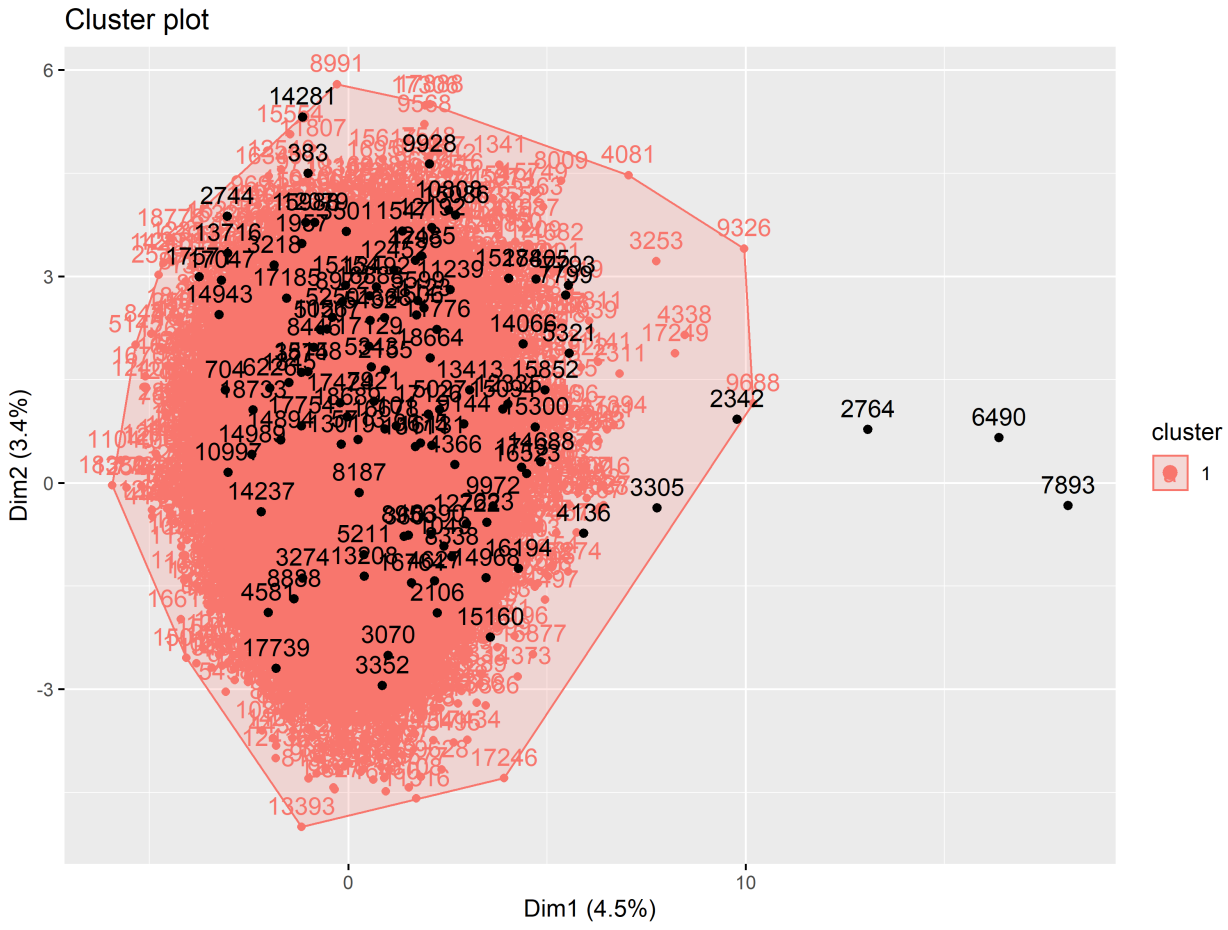


Figure 6: DBSCAN Cluster Plot

Combining Text and Demographic Data

Output

The above three pictures show the protocol of the final product that we have in mind. The protocol uses the example of Li (one of the project members), although the information is hypothetical. The first picture displays his demographic information and a short self-introduction. The second picture shows that the new algorithm highlights the common and memorable words by comparing them with the frequencies among his peers. He also gets a score of 63 and some tips on improving the writing. The third picture is the revised version of previous writing, which now Li has a memorable self-introduction with an increased score of 87. He should be confident to reach out to other users on OkCupid now.

Discussion

The project faces several limitations at this stage. Firstly, without data on the outcome (such as the number of messages), we do not know have a relative measure for the self-introductions' memorabilities. As a result, we don't have the outcomes to train the scoring function with machine learning. Secondly, without follow-up interviews, we cannot measure whether the specific choice of words was aimed at authenticity or matches with an awareness of 'relationshopping' (experienced users use certain phrases fraudulently to hook others). Thirdly, without the users' other profile images, we cannot estimate the effect of a memorable self-introduction on outcomes as the quality of images is a potential moderating variable.

Conclusion

Appendices

AGNES Code

```
## ----setup,
## include=FALSE-------------------------------------------------------
library(tidyverse)
library(knitr)
library(here)
library(cluster)
library(FactoMineR)
library(factoextra)
library(NbClust)
library(dbscan)
library(cowplot)
library(skimr)
library(ggcorrplot)
library(missMDA)
library(cluster)
library(missForest)
library(tictoc)
library(doParallel)


knitr::opts_chunk$set(echo = TRUE, fig.height = 6,
    fig.width = 8, dpi = 400)


set.seed(60615)



## ----data------------------------------------------------------------------
original_data = read_csv(here("Data", "final_okcupid.csv")) %>%
    select(-c("new_index", "orig_index", "clean_text",
```

```
        "essay9", "long_words", "flesch", "dbscan_cluster",
        "profile_length", "prop_longwords"))

names(original_data)



## ----descr-stats, cache=TRUE, fig.height = 8,
## fig.width = 11-------------------
skim_list = original_data %>% skim() %>% partition()

skim_list$numeric %>% kable()
skim_list$character %>% kable()

original_data %>% mutate_if(is.character, factor) %>%
    mutate_all(as.numeric) %>% cor(use = "pairwise.complete.obs") %>%
    ggcorrplot()

clusterability = original_data %>% mutate_if(is.character,
    factor) %>% mutate_all(as.numeric) %>% sample_n(5000) %>%
    get_clust_tendency(n = 50)
clusterability$hopkins_stat
clusterability$plot



## ----pca,
## cache=TRUE-----------------------------------------------------------
original_data %>% mutate_if(is.character, factor) %>%
    mutate_all(as.numeric) %>% mutate_all(scale) %>%
    PCA(graph = FALSE) %>% fviz_pca_biplot(label = "var",
    col.var = "red", col.ind = "grey")
ggsave2(here("Clustering", "pca_v2.png"), height = 7,
    width = 11)
```

```
## ----agg-nest, error=TRUE,
## cache=TRUE----------------------------------------
sampled_data = original_data %>% sample_n(5000)
agnes_data = sampled_data %>% mutate_if(is.character,
    factor) %>% mutate_all(as.numeric) %>% mutate_all(scale)
agnes_diss = agnes_data %>% as.matrix() %>% daisy(metric = "gower")
nb_results = NbClust(data = agnes_data, diss = agnes_diss,
    distance = NULL, min.nc = 2, max.nc = 10, method = "ward.D2")

fviz_nbclust(nb_results)

agnes_mod = agnes_diss %>% hcut(isdiss = TRUE, k = 2,
    hc_func = "agnes", hc_method = "ward.D2")
fviz_dend(agnes_mod)
sampled_data$cluster = agnes_mod$cluster
fviz_cluster(agnes_mod, data = agnes_diss, labelsize = 0)

saveRDS(sampled_data, here("Data", "Results", "agnes_results.rds"))
write_csv(sampled_data, here("Data", "Results", "agnes_results.csv"))



## ----clust-inter-----------------------------------------------------------
modal = function(vect, percent = FALSE, only_one = FALSE) {
    library(tidyverse)
    modal_val = vect %>% unlist() %>% table() %>% .[. ==
        max(.)] %>% names()
    if (only_one) {
        modal_val = modal_val[1]
    }
    if (percent) {
        return(vect %>% unlist() %>% .[. == modal_val] %>%
```

```
            (function(x) length(x)/length(vect)))
    }
    modal_val
}

cluster_significance = function(var, data, clus_var = "cluster") {
    wilcox.test(as.formula(paste(var, "~", clus_var)),
        data)$p.value
}

sampled_data %>% select_if(is.numeric) %>% group_by(cluster) %>%
    summarise_all(mean) %>% kable(caption = "Mean by Cluster")

sampled_data %>% select_if(is.numeric) %>% {
    sapply(names(select(., -cluster)), cluster_significance,
        data = .)
} %>% round(3) %>% enframe() %>% kable(caption = "Wilcox Test P-values")

sampled_data %>% select_if(is.numeric) %>% group_by(cluster) %>%
    group_by(cluster) %>% summarise_all(sd) %>% kable(caption = "Standard Deviation by Cluster")

sampled_data %>% select_if(is.numeric) %>% group_by(cluster) %>%
    group_by(cluster) %>% summarise_all(median) %>%
    kable(caption = "Median by Cluster")

sampled_data %>% mutate(cluster = factor(cluster)) %>%
    select_if((function(x) !is.numeric(x))) %>% group_by(cluster) %>%
    summarise_all(modal, only_one = TRUE) %>% kable(caption = "Mode by Cluster")
sampled_data %>% mutate(cluster = factor(cluster)) %>%
    select_if((function(x) !is.numeric(x))) %>% group_by(cluster) %>%
    summarise_all(modal, percent = TRUE, only_one = TRUE) %>%
    mutate_if(is.numeric, round, 3) %>% kable(caption = "Mode by Cluster")
```

DBSCAN Code

```
## ----setup,
## include=FALSE-------------------------------------------------------
library(tidyverse)
library(knitr)
library(here)
library(cluster)
library(FactoMineR)
library(factoextra)
library(NbClust)
library(dbscan)
library(cowplot)
library(skimr)
library(ggcorrplot)

knitr::opts_chunk$set(echo = TRUE, fig.height = 6,
    fig.width = 8, dpi = 400)

set.seed(60615)



## ----data------------------------------------------------------------------
demo_data = read_csv(here("Data", "compressed_okcupid.csv"))
doc2vec_data = read_csv(here("Data", "doc2vec_results.csv")) %>%
    bind_cols(select(demo_data, long_words, flesch)) %>%
    scale() %>% as_tibble()
doc2vec_data %>% skim() %>% partition() %>% .$numeric %>%
    kable()
doc2vec_data %>% select(-X1) %>% {
    ggcorrplot(cor(.), p.mat = cor_pmat(.), hc.order = TRUE,
        insig = "blank")
```

```
}


## ----clusterability-----------------------------------------------------------
clusterability = doc2vec_data %>% select(-X1) %>% get_clust_tendency(n = 15)
# clusterability$plot


## ----dbscan-------------------------------------------------------------------
doc2vec_data %>% select(-X1) %>% kNNdistplot(k = 5)
dbscan_mod = doc2vec_data %>% select(-X1) %>% dbscan(9,
    5)
doc2vec_data %>% select(-X1) %>% {
    fviz_cluster(dbscan_mod, data = .)
}

dbscan_results = doc2vec_data %>% bind_cols(enframe(dbscan_mod$cluster,
    name = NULL, value = "cluster"))
read_csv(here("Data", "compressed_with_results.csv")) %>%
    mutate(dbscan_cluster = dbscan_mod$cluster) %>%
    write_csv(here("Data", "compressed_with_results.csv"))


## ----merge--------------------------------------------------------------------
merged_demo_data = demo_data %>% select(-essay0, -essay9) %>%
    bind_cols(select(dbscan_results, cluster))
merged_doc2vec_data = demo_data %>% select(X1, essay0,
    essay9) %>% bind_cols(select(dbscan_results, -X1))


## ----demo-data----------------------------------------------------------------
modal = function(vect, percent = FALSE, only_one = FALSE) {
```

```
    library(tidyverse)
    modal_val = vect %>% unlist() %>% table() %>% .[. ==
        max(.)] %>% names()
    if (only_one) {
        modal_val = modal_val[1]
    }
    if (percent) {
        return(vect %>% unlist() %>% .[. == modal_val] %>%
            (function(x) length(x)/length(vect)))
    }
    modal_val
}

cluster_significance = function(var, data, clus_var = "cluster") {
    wilcox.test(as.formula(paste(var, "~", clus_var)),
        data)$p.value
}

merged_demo_data %>% select(-c(X1, education)) %>%
    select_if(is.numeric) %>% group_by(cluster) %>%
    summarise_all(mean) %>% kable(caption = "Mean by Cluster")

merged_demo_data %>% select(-X1) %>% select_if(is.numeric) %>%
    {
        sapply(names(select(., -cluster)), cluster_significance,
            data = .)
    } %>% round(3) %>% enframe() %>% kable(caption = "Wilcox Test P-values")

merged_demo_data %>% select(-c(X1, education)) %>%
    select_if(is.numeric) %>% group_by(cluster) %>%
    group_by(cluster) %>% summarise_all(sd) %>% kable(caption = "Standard Deviation by Cluster")
merged_demo_data %>% select(-c(X1, education)) %>%
```

```
    select_if(is.numeric) %>% group_by(cluster) %>%
    group_by(cluster) %>% summarise_all(median) %>%
    kable(caption = "Median by Cluster")

merged_demo_data %>% select(-c(X1, education)) %>%
    mutate(cluster = factor(cluster)) %>% select_if((function(x) !is.numeric(x))) %>%
    group_by(cluster) %>% summarise_all(modal, only_one = TRUE) %>%
    kable(caption = "Mode by Cluster")
merged_demo_data %>% select(-c(X1, education)) %>%
    mutate(cluster = factor(cluster)) %>% select_if((function(x) !is.numeric(x))) %>%
    group_by(cluster) %>% summarise_all(modal, percent = TRUE,
    only_one = TRUE) %>% mutate_if(is.numeric, round,
    3) %>% kable(caption = "Mode by Cluster")



## ----interpret-outliers-----------------------------------------------------
dbscan_results %>% select(-X1) %>% group_by(cluster) %>%
    summarise_all(mean) %>% kable(caption = "Mean by Cluster")
dbscan_results %>% select(-X1) %>% group_by(cluster) %>%
    summarise_all(sd) %>% kable(caption = "Standard Deviation by Cluster")
dbscan_results %>% select(-X1) %>% group_by(cluster) %>%
    summarise_all(median) %>% kable(caption = "Median by Cluster")



## ----profile-diff-----------------------------------------------------------
merged_doc2vec_data %>% select(cluster, essay0) %>%
    group_by(cluster) %>% sample_n(10) %>% kable()
```

Doc2Vec Code

```
#!/usr/bin/env python
# coding: utf-8

# In[1]:


from gensim.models.doc2vec import Doc2Vec, TaggedDocument
from nltk.tokenize import word_tokenize
import nltk
nltk.download('punkt')
import numpy as np
import pandas as pd
import warnings
warnings.filterwarnings('ignore')
import seaborn as sns


# In[2]:


import matplotlib.pyplot as plt


# In[3]:


df = pd.read_csv('../Data/compressed_okcupid.csv').dropna(subset=['essay0'])
df.info()


# In[4]:
```

```
data = df['essay0']


# In[5]:


tagged_data = [TaggedDocument(words=word_tokenize(_d.lower()),
                              tags=[str(i)]) for i, _d in enumerate(df['essay0'])]


# In[6]:


from gensim.models.doc2vec import Doc2Vec

max_epochs = 50
vec_size = 50
alpha = 0.025

model = Doc2Vec(size=vec_size,
                alpha=alpha,
                min_alpha=0.00025,
                min_count=1,
                dm =1)

model.build_vocab(tagged_data)

for epoch in range(max_epochs):
    print('iteration {0}'.format(epoch))
    model.train(tagged_data,
                total_examples=model.corpus_count,
                epochs=model.iter)
```

```
    # decrease the learning rate
    model.alpha -= 0.0002
    # fix the learning rate, no decay
    model.min_alpha = model.alpha

model.save("dv_50.model")
print("Model Saved")



# In[7]:



#from gensim.models.doc2vec import Doc2Vec

model= Doc2Vec.load("dv_50.model")
#to find the vector of a document which is not in training data
#test_data = word_tokenize("I love chatbots".lower())
#v1 = model.infer_vector(test_data)
#print("V1_infer", v1)

# to find most similar doc using tags
similar_doc = model.docvecs.most_similar('0')
print(similar_doc)



# In[8]:



model.docvecs.vectors_docs.shape
type(model.docvecs.vectors_docs)
```

```
# In[9]:



#Visualize TSNE with doc2vec
from sklearn.manifold import TSNE
def doc2vec_tsne_plot(doc_model, labels):

    tokens = []
    for i in range(len(doc_model.docvecs.vectors_docs)):
        tokens.append(doc_model.docvecs.vectors_docs[i])

    # Reduce 100 dimensional vectors down into 2-dimensional space so that we can see them
    tsne_model = TSNE(perplexity=40, n_components=2, init='pca', n_iter=2500, random_state=23)
    new_values = tsne_model.fit_transform(tokens)

    X = [doc[0] for doc in new_values]
    y = [doc[1] for doc in new_values]

    # Combine data into DataFrame, so that we plot it easily using Seaborn
    df = pd.DataFrame({'X':X, 'y':y, 'Cuisine':labels})
    plt.figure(figsize=(16, 16))
    sns.scatterplot(x="X", y="y", hue="Cuisine", data=df)
    return

doc2vec_tsne_plot(model, df['height'])



# In[10]:



from sklearn.cluster import KMeans
from sklearn.decomposition import PCA
```

```
import matplotlib.pyplot as plt


X = model.docvecs.vectors_docs
Sum_of_squared_distances = []
K = range(2,15)
for k in K:
    km = KMeans(n_clusters=k)
    km = km.fit(X)
    Sum_of_squared_distances.append(km.inertia_)



# In[11]:



plt.plot(K, Sum_of_squared_distances, 'bx-')
plt.xlabel('Number of Clusters')
plt.ylabel('Sum of Squared Distances from Cluster Centres')
plt.title('Elbow Method For Optimal Number of Clusters', fontsize=15)
plt.savefig('Elbow_Plot.png', bbox_inches='tight')



# In[12]:



num_clusters = 4
num_seeds = 4
max_iterations = 300
labels_color_map = {
    0: '#20b2aa', 1: '#ff7373', 2: '#ffe4e1', 3: '#005073', 4: '#4d0404'
}
pca_num_components = 2
#texts_list = df['essay0']
```

```
# calculate tf-idf of texts
#tf_idf_vectorizer = TfidfVectorizer(analyzer="word", use_idf=True,
                                    #smooth_idf=True, ngram_range=(2, 3))
#tf_idf_matrix = tf_idf_vectorizer.fit_transform(texts_list)

# create k-means model with custom config
clustering_model = KMeans(
    n_clusters=num_clusters,
    max_iter=max_iterations,
    precompute_distances="auto",
    n_jobs=-1
)

X = model.docvecs.vectors_docs
reduced_data = PCA(n_components=pca_num_components).fit_transform(X)
labels = clustering_model.fit_predict(reduced_data)



# In[13]:



pca_num_components = 2
X = model.docvecs.vectors_docs
reduced_data = PCA(n_components=pca_num_components).fit_transform(X)



# In[16]:



# there appears to be no column named 'clust_label' in df
#df[df['clust_label']==1]['essay0']
```

```
# In[17]:



from sklearn.manifold import TSNE
# Creating and fitting the tsne model to the document embeddings
tsne_model = TSNE(early_exaggeration=4,
                  n_components=2,
                  verbose=1,
                  random_state=2018,
                  n_iter=300)
tsne_d2v = tsne_model.fit_transform(model.docvecs.vectors_docs)



# In[18]:



plt.scatter(tsne_d2v [:, 0], tsne_d2v [:, 1])
plt.show()
plt.savefig('TSNE_blob.png', bbox_inches='tight')



# In[30]:



output = pd.DataFrame(model.docvecs.vectors_docs)
output.to_csv("../Data/doc2vec_results.csv")
output
```

References

Bruning, Roger H, Gregory J Schraw, and Royce R Ronning. 1999. *Cognitive Psychology and Instruction*. ERIC.

Cacioppo, John T, Stephanie Cacioppo, Gian C Gonzaga, Elizabeth L Ogburn, and Tyler J VanderWeele. 2013. "Marital Satisfaction and Break-Ups Differ Across on-Line and Off-Line Meeting Venues." *Proceedings of the National Academy of Sciences* 110 (25): 10135–40.

Goffman, E. 1975. *The Presentation of Self in Everyday Life*. Pelican Book. Penguin Books. https://books.google.com/books?id=tYvNnQEACAAJ.

Heino, Rebecca D, Nicole B Ellison, and Jennifer L Gibbs. 2010. "Relationshopping: Investigating the Market Metaphor in Online Dating." *Journal of Social and Personal Relationships* 27 (4): 427–47.

Hitsch, Gunter J, Ali Hortaçsu, and Dan Ariely. 2010. "Matching and Sorting in Online Dating." *American Economic Review* 100 (1): 130–63.

Kim, Albert Y, and Adriana Escobedo-Land. 2015. "OkCupid Data for Introductory Statistics and Data Science Courses." *Journal of Statistics Education* 23 (2).

Papacharissi, Zizi. 2010. *A Networked Self: Identity, Community, and Culture on Social Network Sites*. Routledge.