Final Report

Li Liu, Abhishek Pandit, Adam Shelton

12/7/2019

Contents

Contributions

Li	**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
new_index	0	1	9414.01	5435.85	0.0	4707.00	9413.00	14121.00	18830
orig_index	0	1	29722.38	17241.83	0.0	14986.00	29545.00	44303.00	60550
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

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

Table 2: Other Variables

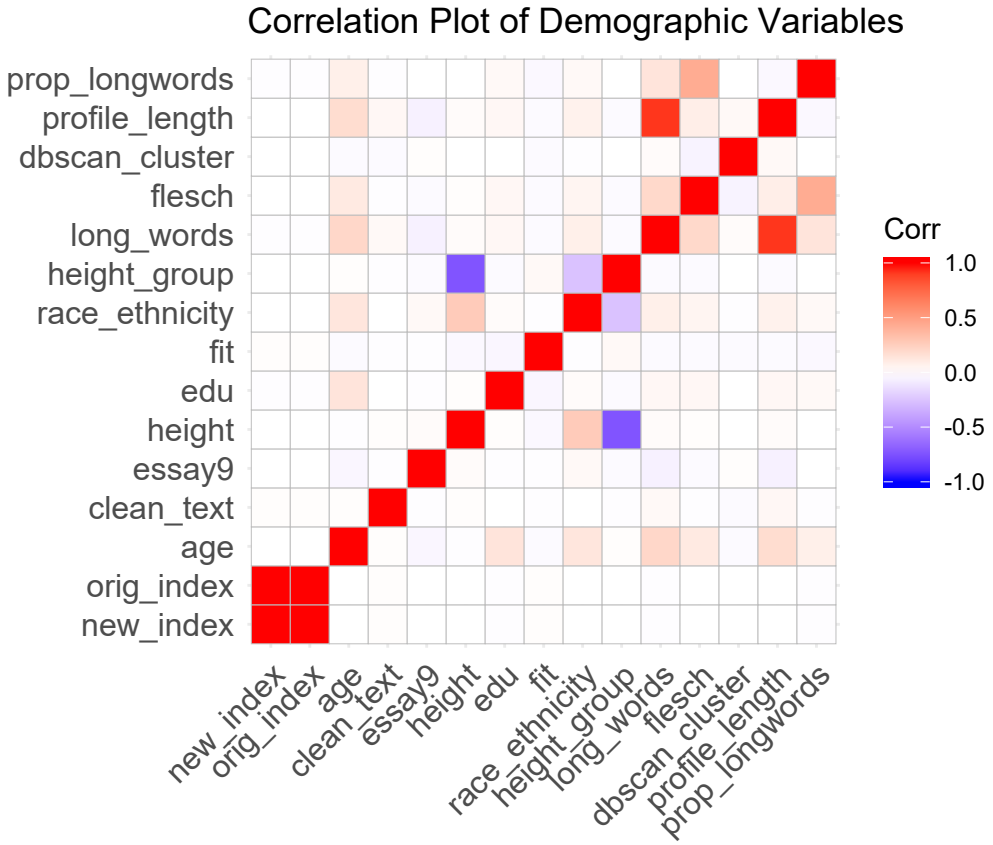
skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
clean_text	0	1	1	16952	0	18790	0
essay9	0	1	1	10849	0	18125	0
edu	0	1	7	21	0	3	0
fit	0	1	3	7	0	3	0
race_ethnicity	0	1	5	8	0	6	0
height_group	0	1	5	9	0	2	0

Empirical Strategy

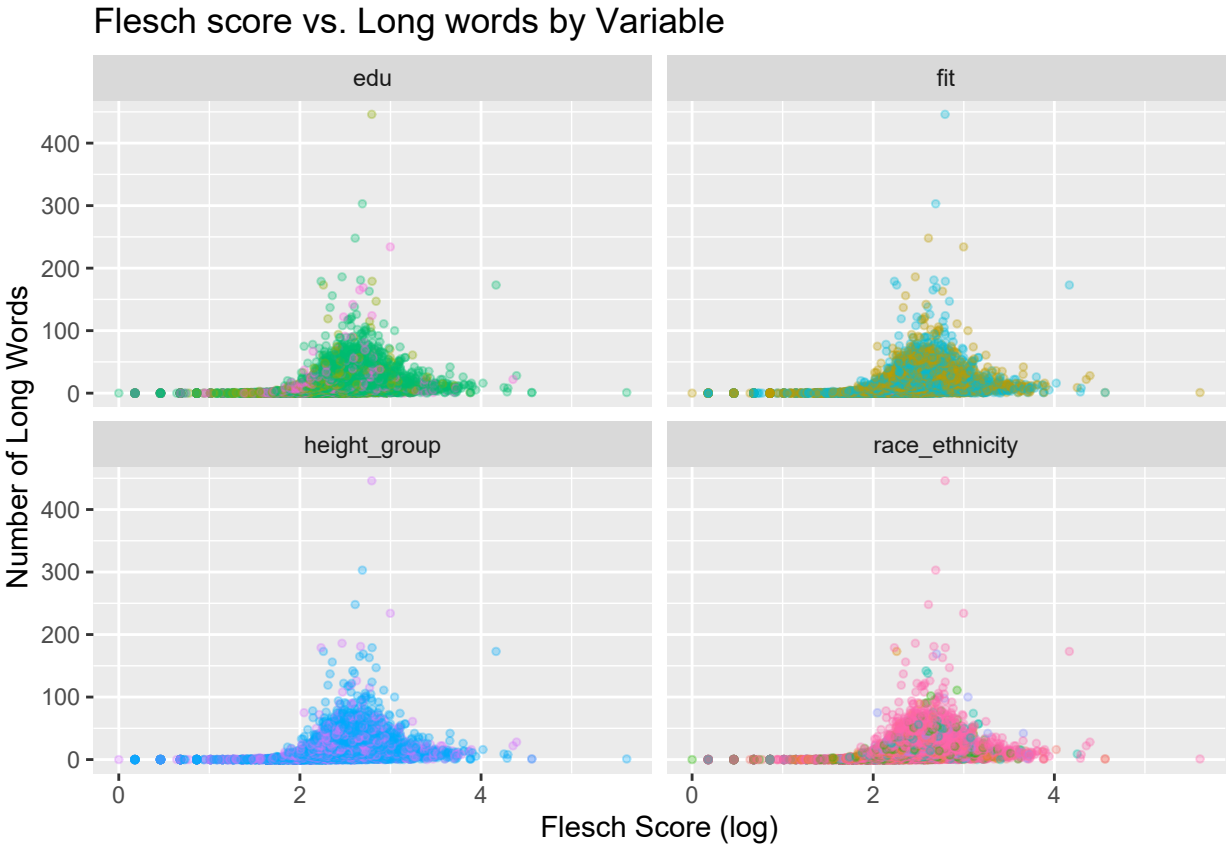
Analysis & Results

Exploratory Data Analysis

Descriptive Statistics



Visual Analysis



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 subset of 2000 observations was used instead.

Text Analysis

Word2Vec

Topic Modeling

DBSCAN

A DBSCAN model was used to detect outliers within a data-set of 50 vectors calculated by Doc2Vec.

Combining Text and Demographic Data

Discussion

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)

# drops variables in a dataset with a certain
# percentage or number of missing values (a lower
# value means less missing values are allowed
# before a variable is dropped)
```

```r
drop_high_na = function(data_set, cutoff = 0.5, rows = FALSE) {
    library(tidyverse)
    if (cutoff < 0) {
        stop("Cutoff cannot be less than zero!")
    }
    # returns a logical value if a row/column at a
    # specifed index should be dropped
    drop_index = function(index, ds) {
        items = NULL
        if (rows) {
            items = ds[index, ] %>% unlist()
        } else {
            items = ds[, index] %>% unlist()
        }
        num_nas = items %>% is.na() %>% sum()
        if (cutoff < 1) {
            perc_nas = num_nas/length(items)
            return(perc_nas < cutoff)
        } else {
            return(num_nas < cutoff)
        }
        return(NULL)
    }

    if (rows) {
        keep_row = sapply(1:nrow(data_set), drop_index,
            data_set)
        return(data_set[keep_row, ])
    }
    keep_col = sapply(1:ncol(data_set), drop_index,
        data_set)
    return(data_set[, keep_col])
```

```
}



## ----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--------------------------------------------------------------
```

```r
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-----------------------------------------------------
sampled_data = original_data %>% sample_n(2000)
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 = 3,
    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"))
```

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
}

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(-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")
```

```r
## ----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.

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