Professor He
MATH 656
Landon, Roger, Calvin

Final Social Media Project

Introduction

People go on Facebook to connect with friends, post their opinions through the status function, or share pictures of a recent trip that they took part in. These are forms of unstructured data. If scientists at Facebook explore and understand users' behaviors, they certainly can attract more users and build social media to become a better place. The dataset utilized in this project is taken from the larger study conducted by the Psychometrics Center at the University of Cambridge. The objective of the study is to cluster these users and explore the relationship between the Big5 Personality Traits and each cluster. From psychological trait theory, psychologists categorize human behavior into five different personality traits. They are Extraversion (sociable vs shy), Neuroticism (neurotic vs calm), Agreeableness (friendly vs uncooperative), Conscientiousness (organized vs careless), and Openness (insightful vs unimaginative). After creating and selecting our own features for this project separating the 250 users in this data set into different clusters, we plan to match these clusters to the Big5 personalities.

Methods

A series of methods were required to properly understand how the Facebook statuses and network features could best generate clusters on the 250 users and the relationship with the Big 5 personalities. First, as with most data science projects, the data was cleaned. We noted that the dataset contained information on 250 Facebook users with unique IDs, possessed 15 variables, and 9917 records. A simple EDA (exploratory data analysis) exercise showed that we had extraneous columns ('Unnamed: 2', 'Unnamed: 3', 'Unnamed: 4','Unnamed: 5', 'Unnamed: 6'), so we dropped these columns using 'drop' python function. Next, we observed that there was a row missing values and as we couldn't have an accurate estimate of its missing value, the row was dropped using 'dropna'. Similarly, we noted that user ID '5c73047a8292131a9aa261cfa07932fd' on 12/25 had a status of ' #VALUE!', thus we dropped this row as well. In some instances inputting a mean or median value would work, but as we were dealing with unstructured textual data and nearly independent users, we decided inputting a "best guess" or median/mean replacement wouldn't work and thus opted to remove the two rows in this unique instance. Next, we discovered in our research the NLTK Python package had a TweetTokenizer() method that was ideal for creating social media unigrams, as it allowed the unigrams to maintain emojis and hashtags. This would be helpful later when we looked for personality traits.

Two features were generated: (1) total number of unigrams per post & (2) total number of posts in the data sample by user. We hypothesized that one who posts more often and/or with more words per status, is more likely to be gregarious and socially outgoing, whereas those who post less often & with fewer word counts are likely to be more shy. To create these features, we used python group-by transformation count as well as len() functions. Three more additional features were then created. Three sets of words were created. Although logical hypotheses, these sets for topics such as egoism, weren't pulled from other research. The first set was for egoism including words like 'i','me','myself'. The second set was for capturing extraversion, namely sociable people who we hypothesized would post exclamation points, heart emojis, smiling emojis, or say things like 'friend' or 'party' for example. The last set was testing agreeableness with words like 'compromise', 'agree', 'agreed', or 'agreement'. By creating a function called 'term_count' and testing if these special set words were present in the Facebook social media post unigrams, the method allowed us to determine the frequency of these words and perhaps shed light on the user's personality. A function was created to achieve this, defined as term_count(input_list, comparison_set), where comparison sets are the aforementioned sets like 'i','me', or 'myself' for egoism for example. The function, when provided these parameters, returns len([1 for word in input_list if str(word).lower() in comparison_set]), where 1 is a placeholder value that is arbitrary. It should be noted that these values were later scaled to ensure one feature didn't overshadow another feature due to possessing a greater value magnitude. It should also be noted that stop words from the nltk.corpus package were considered, but it was recognized that some of the stop words were actually helpful, such as 'i','we' et cetera, and that the total count of words used in a status could be predictive of gregarious individuals, thus removing stop words in fact removed insight rather than help the analysis.

In addition to manipulating "status" for aforementioned features, the team utilized the provided variables including network size (# of direct connections), betweenness (measure of bridge spanning role), brokerage (indicates number of connected neighbors' pairs that individual does not directly connect with), density (density of connections between individual & friends/ friends of friends), and transitivity (notion that friends of my friends are also my friends). In the validation phase for homogenous characters of members in each cluster, the team also compared the cluster results (ex. Cluster 0,1, n) against the 5 personality trait gold standards aforementioned (e.g. cEXT for extraversion, cNEU for neuroticism, cAGR for agreeableness, cCON for conscientiousness, cOPN for openness). This concludes the discussion of features employed in this study.

The next methodological step required using our two cluster methods. Our first cluster method was the k-means approach, which merits a definition. Hailed as one of the most popular techniques for clustering, Stanford defines it as finding the "best centroids by alternating between (1) assigning data points to clusters based on the current centroids (2) choosing centroids (points which are the center of a cluster) based on the current assignment of data points to clusters." [1] Namely, this signifies that a point is appointed to the nearest cluster centroid. The

[1] Piech, C. (n.d.). *CS221*. Lecture, Stanford University; Stanford University. Retrieved from https://stanford.edu/~cpiech/cs221/handouts/kmeans.html.

second method was fuzzy c-means, which can be defined as “a data clustering technique in which a data set is grouped into N clusters with every data point in the dataset belonging to every cluster to a certain degree”, [2] hence the “fuzzy” adjective because as a data point gets closer to the centroid its “membership” to that cluster increases. Conversely, the farther away a point is to the centroid, the lower the membership of the data point to the cluster.

Next the optimal number of clusters were determined. For the KMeans clustering, the team used the elbow method as a tool to select the most optimal count of clusters and the ideal model fit demarcated with a vertical black dotted line at the “elbow” point of inflection at the ideal value of k clusters.[3] We used the following code where ‘//’ stands for a new line: “visualizer = KElbowVisualizer (kmeans, k=(2,15)) // visualizer.fit(df)// visualizer.poof()”, see code for full details. As for the Fuzzy c-means, a paper called “*A Method to Find Optimum Number of Clusters Based on Fuzzy Silhouette on Dynamic Data Set*” [4] was particularly useful as it highlighted that the Silhouette cluster validity criterion index could be applied to determine the optimal cluster size of the Fuzzy c-means, where a higher silhouette value is better. To visualize in two dimensions, a PCA analysis was performed and then a scatterplot using the seaborn visualization package utilized with the best two principal components (PCA1 and PCA2), the results of which are discussed later in this report.

In terms of how we validated results, two techniques were used. For the K-means, SSE & SSB metrics were utilized to validate the results. These stand for the sum of squared error (SSE) and sum of squares between (SSB) clusters, such that we want to minimize the SSE as this is the intra-cluster variance within the cluster (smaller is better here) and maximize the SSB as it is the inter-cluster variance. For Fuzzy c-means, we used the silhouette width. Namely, the highest silhouette index per cluster group (Ex. 2,3,4,5,...n clusters) validated the best clustering.

The last notable method to discuss is how we compared the personality gold standards. Indeed, once we obtained our clustering results (ex. Each of the 250 users had its associated cluster grouping value such as “cluster 2” for example in its own column), we did a comparison to how many of each cluster were in each gold standard personality. We then drew conclusions from our results about our feature predictive power and overall usefulness of our models using Facebook statuses/network features in exploring the relationship between each cluster and the Big 5 personalities introduced in the introduction.

[2] *Fuzzy C-Means Clustering*. MATLAB & Simulink. (n.d.). Retrieved December 10, 2022, from https://www.mathworks.com/help/fuzzy/fuzzy-c-means-clustering.html

[3] *Clustering Visualizers - Elbow method*. Elbow Method - Yellowbrick v1.5 documentation. (n.d.). Retrieved December 13, 2022, from https://www.scikit-yb.org/en/latest/api/cluster/elbow.html

[4] Subbalakshmia, C., Krishnab, G. R., Raoc, K. M., & Raod, P. V. (n.d.). *A Method to Find Optimum Number of Clusters Based on Fuzzy Silhouette on Dynamic Data Set*. Retrieved December 11, 2022, from https://core.ac.uk/download/pdf/82373216.pdf

Results & Discussions

In this project, we introduced two clustering methods, K-means clustering and Fuzzy c-means clustering (FCM). As shown in Figure 1, we use the Elbow method to determine that K=7 is the optimal clustering number for K-means clustering.

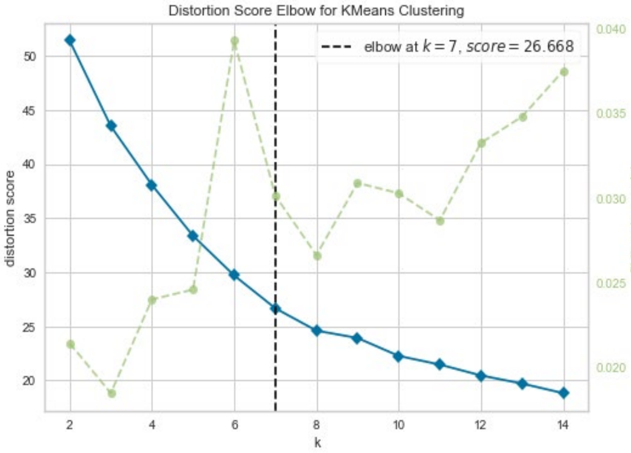


Figure 1

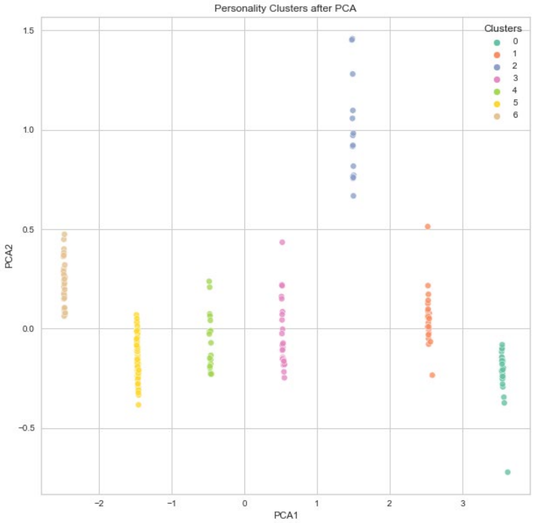


Figure 2

Using Principal Component Analysis (PCA), we were able to visualize our clustering results. As Figure 2 shows, 250 persons clearly separated into 7 clusters. Secondly, the SSE (within cluster sum of squares) divided by TSS (Total sum of squares) is equal to 0.39, this indicates objects (data points) in cluster are related to each other and are likely aligned to the correct clustering. Indeed, the SSB (between cluster sum of squares) divided by TSS is 0.61, which indicates cluster is well separated from other clusters. The results for the FCM (0.24) are not as ideal (value not as high) as K-means (0.39), as determined by comparing the highest average silhouette width in Figure 3, which further details FCM silhouette plot.

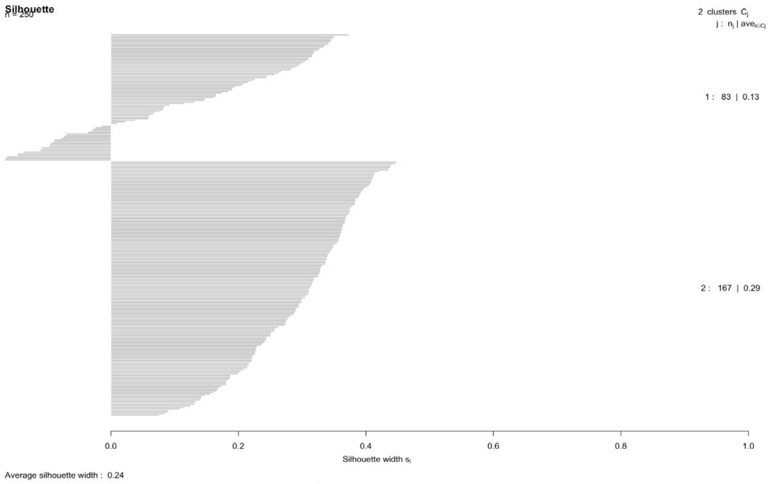
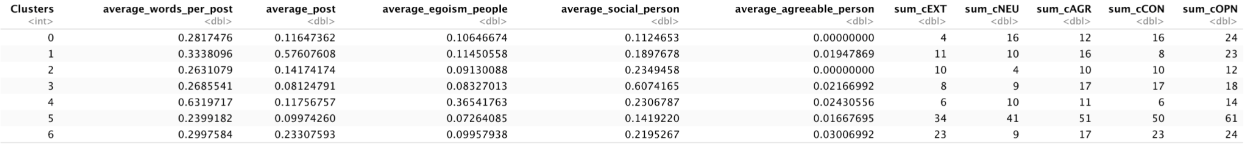
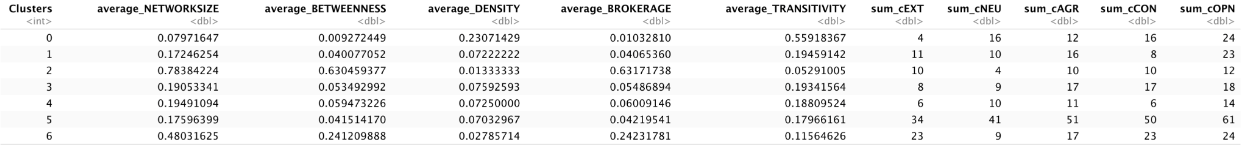


Figure 3



Clusters	average_words_per_post	average_post	average_egoism_people	average_social_person	average_agreeable_person	sum_cEXT	sum_cNEU	sum_cAGR	sum_cCON	sum_cOPN
<int>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
0	0.2817476	0.11647362	0.10646674	0.1124653	0.00000000	4	16	12	16	24
1	0.3338096	0.57607608	0.11450558	0.1897678	0.01947869	11	10	16	8	23
2	0.2631079	0.14174174	0.09130088	0.2349458	0.00000000	10	4	10	10	12
3	0.2685541	0.08124791	0.08327013	0.6074165	0.02166992	8	9	17	17	18
4	0.6319717	0.11756757	0.36541763	0.2306787	0.02430556	6	10	11	6	14
5	0.2399182	0.09974260	0.07264085	0.1419220	0.01667695	34	41	51	50	61
6	0.2997584	0.23307593	0.09957938	0.2195267	0.03006992	23	9	17	23	24

Table 1



Clusters	average_NETWORKSIZE	average_BETWEENNESS	average_DENSITY	average_BROKERAGE	average_TRANSITIVITY	sum_cEXT	sum_cNEU	sum_cAGR	sum_cCON	sum_cOPN
<int>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
0	0.07971647	0.009272449	0.23071429	0.01032810	0.55918367	4	16	12	16	24
1	0.17246254	0.040077052	0.07222222	0.04065360	0.19459142	11	10	16	8	23
2	0.78384224	0.630459377	0.01333333	0.63171738	0.05291005	10	4	10	10	12
3	0.19053341	0.053492992	0.07592593	0.05486894	0.19341564	8	9	17	17	18
4	0.19491094	0.059473226	0.07250000	0.06009146	0.18809524	6	10	11	6	14
5	0.17596399	0.041514170	0.07032967	0.04219541	0.17966161	34	41	51	50	61
6	0.48031625	0.241209888	0.02785714	0.24231781	0.11564626	23	9	17	23	24

Table 2

As you can see in Table2, we isolated the variables furnished from the original dataset, we cluster them and calculated their average, namely, average_NETWORKSIZE, average_BETWEENNESS, average_DENSITY, average_BROKERAGE, and average_TRANSITIVITY where clusters (in respective order) 2, 2, 0, 2, 0 had the highest values, and clusters (in respective order) 0, 0, 2, 0, 2 had the lowest values. This makes intuitive sense as someone that has many friends has a large network size and logically has an important bridging role, thus a high betweenness value. As individuals in this group are important bridges to many groups, it also makes sense that they would not know *every* single friend to one of their connections. This insight took a while to understand but helps explain the high brokerage value for cluster 2, which again is due to them being spread out so much as the individuals in this group play key bridging spanning roles. This logic also explains how, conversely, cluster 2 has the lowest density (low percent of friends' friends being connected to cluster members) and transitivity (friends of my friends are also my friends). Interestingly, this group shows the lowest level of neuroticism, suggesting that they are calm, which perhaps explains why they were naturally or perhaps chosen to be in such key bridging positions (e.g. Leader of a company, social media influencer, or another pivotal bridging position). Our favorite insight from this table, however, was that cluster 0, the group with the lowest gold standard extraversion in our model, had the smallest network size, which makes total sense, as one would expect shy individuals to have a small network as they are less likely to go out to make an effort in meeting people. Please reference the table to draw out further conclusions as the reader wishes, however, we believe these initial insights already provided immense value to these features, and thus now move to discuss our second set of hand made features.

As one can see in Table1, we created our own scaled features, namely, average_words_per_post, average_post (scaled average number of posts per person in cluster), average_egoism_people, average_social_person, average_agreeable_person where clusters (in respective order) show the highest values align to clusters 4, 1, 4, 3 & 6. Conversely, we see the lowest values align (in respective order to these column names previously mentioned) to clusters

5, 3, 5, 0, 0/2. This signifies that cluster detected uncooperativeness in clusters 0 and 2, based on our agreeableness set. This means for our unique set, the individuals did not use the words 'compromise', 'agree', 'agreement', or 'agreed'. To discuss these results, one can see that we accurately predicted 2 of the 3 least agreeable clusters, which was a success in our team's goal to compare our features to the gold standard personality feature traits. To improve our prediction, we would likely want to expand our agreeableness set beyond our small set of words. We were also successful in our egoism testing. Namely, cluster 5 which had the lowest egoistic feature score (0.07), showed to be the most (1) extroverted (highest gold standard score of 34) and (2) agreeable (highest gold standard score of 51). This suggests that those who are less egotistical are more likely to be extroverted and agreeable people, thus others want to engage with them socially and be friends with them.

Conclusion

In conclusion, our K-means was the best model as it had the highest silhouette value (closer to 1 signifies more well clustered). Both our network features and our hand-crafted features were correlated with personality traits and could be used to draw insights. Indeed, the egoism feature the team built showed that a low egoism score was aligned with the most extroverted and agreeable cluster of individuals in our model. The agreeableness link was not a surprise, but the extroversion relationship with less egoism was a fascinating finding. Secondly, our agreeable set was predictive of agreeableness in 2 of the 3 most agreeable clusters. Likewise, we noted that the smallest network size, as one would hypothesize, was correlated with being shy. This makes sense as people with small network sizes may not be as outgoing and willing to make new friends easily. However, in addition to these great findings, more work could and should be done to further enhance the findings. There were limitations such as time constraints in this accelerated two-week project or the focus on using only two models, which could be improved by increasing funding, research time, and available models. For further study, the team would suggest integrating a parts of speech component and stemming/lemmatization, which could improve accuracy. Indeed, lemmatization helps "reduce inflectional forms" giving one back the root word accounting for the part of speech for the unigram and thus could improve insights generated in our models in a proper implementation.[5]

[5] Stemming and lemmatization. (n.d.). Retrieved December 13, 2022, from https://nlp.stanford.edu/IR-book/html/htmledition/stemming-and-lemmatization-1.html