Personality Profiling with Clustering

Mobina Kargar
dymamsijhidjj@gmail.com

Abstract

The objective of this assignment is to apply various unsupervised clustering algorithms to the "Big Five Personality Test" dataset to identify distinct personality profiles among individuals. You will evaluate the performance of these clustering models using appropriate metrics and visualize the resulting clusters to gain insights into human personality traits.

1 Introduction

The dataset used in this project is derived from the IPIP-FFM (International Personality Item Pool - Five-Factor Model) questionnaire, a widely recognized tool for assessing the Big Five personality traits: Extraversion (EXT), Neuroticism (EST), Agreeableness (AGR), Conscientiousness (CSN), and Openness to Experience (OPN). The raw dataset was loaded from a tab-separated .csv file containing over 1 million responses and more than 100 personality-related variables.

2 Features

The data is composed of 100 columns and 1015340 entries (Full train dataset shape is (1015340, 100)). We can see all 100 dimensions of our dataset by printing out the first 3 entries:

Table 1: train dataset (3 rows x 100 columns)

	EXT1	EXT2	EXT3	EXT4	...	OPN7_E	OPN8_E	OPN9_E	OPN10_E
0	2216.0	1856.0	1732.0	1735.0	...	1861.0	1540.0	1484.0	1184.0
1	1351.0	2350.0	2112.0	1752.0	...	1730.0	1640.0	1872.0	2016.0

2	888.0	2941.0	1829.0	2057.0	...	1320.0	1453.0	1320.0	2355.0

We can inspect the types of feature columns:

Table 2: Data columns:

```
<class 'pandas.core.frame.DataFrame'>
Index: 1012050 entries, 0 to 1015340
Data columns (total 100 columns):
 #   Column   Non-Null Count    Dtype
---  ------   --------------    -----
 0   EXT1     1012050 non-null  float64
 1   EXT2     1012050 non-null  float64
 2   EXT3     1012050 non-null  float64
 3   EXT4     1012050 non-null  float64
 4   EXT5     1012050 non-null  float64
 5   EXT6     1012050 non-null  float64
 6   EXT7     1012050 non-null  float64
 7   EXT8     1012050 non-null  float64
 8   EXT9     1012050 non-null  float64
 9   EXT10    1012050 non-null  float64
 10  EST1     1012050 non-null  float64
 11  EST2     1012050 non-null  float64
 12  EST3     1012050 non-null  float64
 13  EST4     1012050 non-null  float64
 14  EST5     1012050 non-null  float64
 15  EST6     1012050 non-null  float64
 16  EST7     1012050 non-null  float64
 17  EST8     1012050 non-null  float64
 18  EST9     1012050 non-null  float64
 19  EST10    1012050 non-null  float64
 ...
 99  OPN10_E  1012050 non-null  float64
```

3 Distribution

3.1 Preparing Data:

_ Step 1: Remove duplicate or irrelevant observations

Duplicate columns: []

_ Step 2: Fix structural errors

… Hopefully in this code it doesn’t need to use this approach

_ Step 3: Filter unwanted outliers

Detect all rows that have outliers in at least one column and treat them

```
         EXT1  EXT2  EXT3  EXT4  EXT5  EXT6  EXT7  EXT8  EXT9  EXT10  ...  \
28        2.0   2.0   3.0   5.0   3.0   3.0   1.0   4.0   1.0    3.0  ...
29        2.0   4.0   4.0   2.0   3.0   3.0   3.0   3.0   3.0    3.0  ...
73        2.0   3.0   3.0   3.0   3.0   3.0   4.0   3.0   3.0    4.0  ...
79        3.0   2.0   5.0   3.0   4.0   2.0   4.0   3.0   3.0    3.0  ...
109       4.0   1.0   4.0   2.0   4.0   3.0   5.0   2.0   5.0    2.0  ...
...       ...   ...   ...   ...   ...   ...   ...   ...   ...    ...  ...
1015183   3.0   2.0   4.0   4.0   4.0   4.0   4.0   3.0   3.0    3.0  ...
1015203   3.0   3.0   3.0   4.0   4.0   3.0   3.0   4.0   3.0    4.0  ...
1015204   3.0   4.0   4.0   4.0   4.0   2.0   2.0   3.0   4.0    4.0  ...
1015312   2.0   2.0   2.0   4.0   3.0   2.0   1.0   5.0   1.0    3.0  ...
1015319   4.0   3.0   4.0   2.0   4.0   2.0   4.0   2.0   5.0    4.0  ...

         OPN1_E  OPN2_E  OPN3_E  OPN4_E  OPN5_E  OPN6_E  OPN7_E  OPN8_E  \
28       2216.0  1856.0  1732.0  1735.0  1092.0  1732.0  1861.0  1540.0
29       1351.0  2350.0  2112.0  1752.0  1400.0  1744.0  1730.0  1640.0
73        888.0  2941.0  1829.0  2057.0  1970.0  2576.0  1320.0  1453.0
79       2611.0  2568.0  2173.0  2516.0  3766.0  2644.0  2681.0  2917.0
109      1161.0  3416.0  1152.0  1721.0  1672.0  1400.0  1176.0  1953.0
...         ...     ...     ...     ...     ...     ...     ...     ...
1015183  1221.0  1785.0  1528.0  1967.0  2004.0  2517.0  2702.0  1411.0
1015203   659.0   618.0  1159.0  1558.0  1737.0  2026.0   718.0   608.0
1015204  1104.0  2692.0  1563.0  5466.0  1547.0  2314.0  1274.0  1970.0
1015312  1511.0  2359.0  1256.0  4943.0  2047.0  1623.0  1711.0  3352.0
1015319   631.0  3952.0  1035.0  4242.0  1260.0  2796.0  1383.0   945.0
...
1015312  1344.0   905.0
1015319   663.0  1099.0

[26240 rows x 100 columns]
```

_ Step 4: Handle missing data by inplace them with true or zero

```
EXT1                       1783
EXT2                       1783
EXT3                       1783
EXT4                       1783
EXT5                       1783
                           ...
endelapse                     0
IPC                           0
country                      77
lat_appx_lots_of_err          0
long_appx_lots_of_err         0
Length: 110, dtype: int64
```

3.2 Distribution for Numerical Data

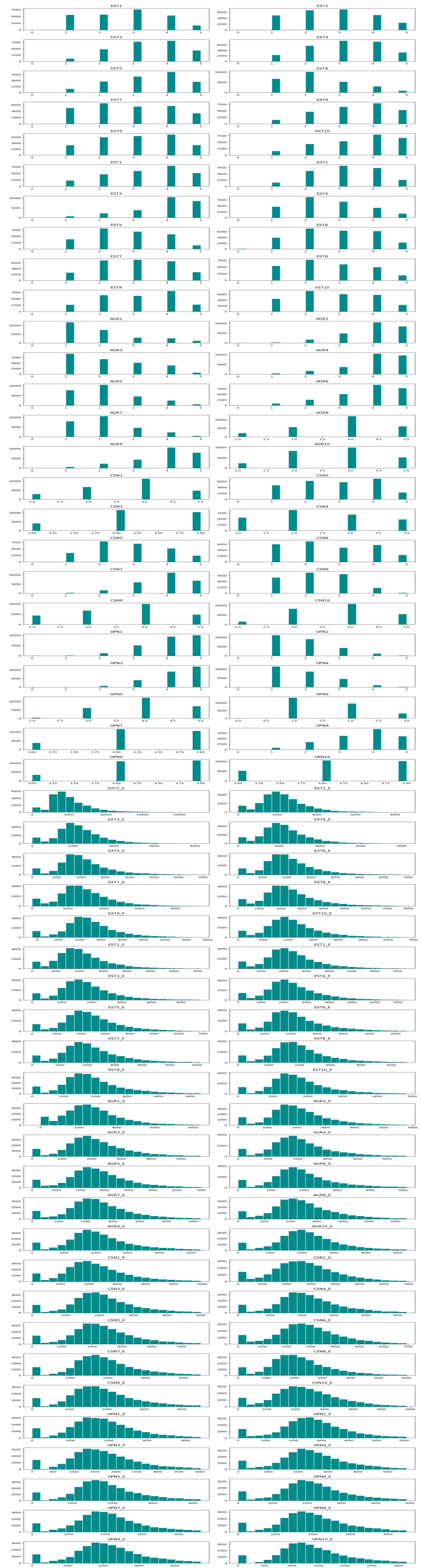

Figure1: [Grid of histograms] each representing the distribution of values in one of the numeric columns from dataset

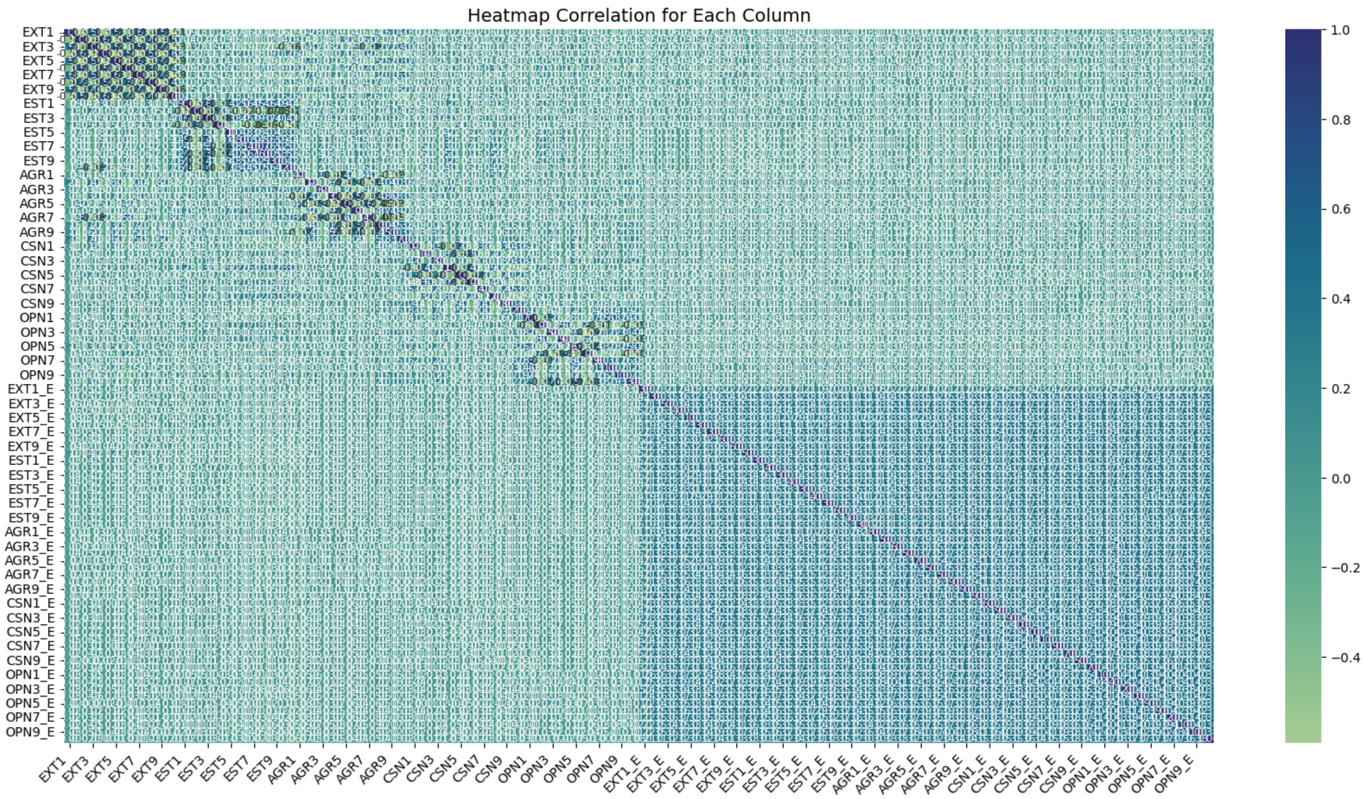

Figure2: [Heatmap] Correlation For Each Column to have more insight about data

look at how this data is distributed.

	# mean	# std	# min	# 25%	# 50%	# 75%	# max
EXT1	2.6478919025739835	1.264476090670833	0.0	1.0	3.0	4.0	5.0
EXT2	2.773242428733758	1.323982013105767	0.0	2.0	3.0	4.0	5.0
EXT3	3.288202163924707	1.2150827851864319	0.0	2.0	3.0	4.0	5.0
EXT4	3.1406985820858653	1.2374725335424257	0.0	2.0	3.0	4.0	5.0
EXT5	3.2768054937997135	1.2776361753051793	0.0	2.0	3.0	4.0	5.0
EXT6	2.4011185218121636	1.2258116234182765	0.0	1.0	2.0	3.0	5.0
EXT7	2.7715270984635145	1.4003482899197013	0.0	2.0	3.0	4.0	5.0
EXT8	3.4147986759547453	1.2719536515500225	0.0	2.0	4.0	4.0	5.0
EXT9	2.963760683760684	1.3460534121085221	0.0	2.0	3.0	4.0	5.0
EXT10	3.556569339459513	1.3052487415732186	0.0	3.0	4.0	5.0	5.0

100 rows x 7 cols 10 per page Page 1 of 10

4 PCA & T-SNE

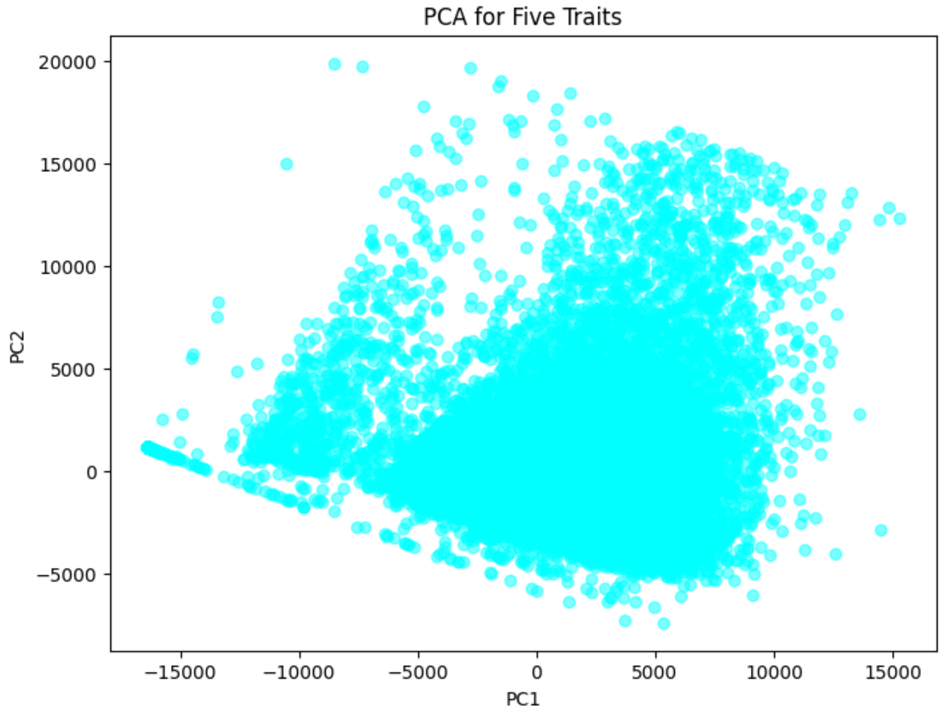

Figure3: Indicating that the five traits are correlated and can be well represented by the first two principal components and the distribution of data points in a 2-dimensional space defined by the first two principal components (PC1 and PC2). Each data point represents an individual with five traits. The x-axis represents PC1, and the y-axis represents PC2.

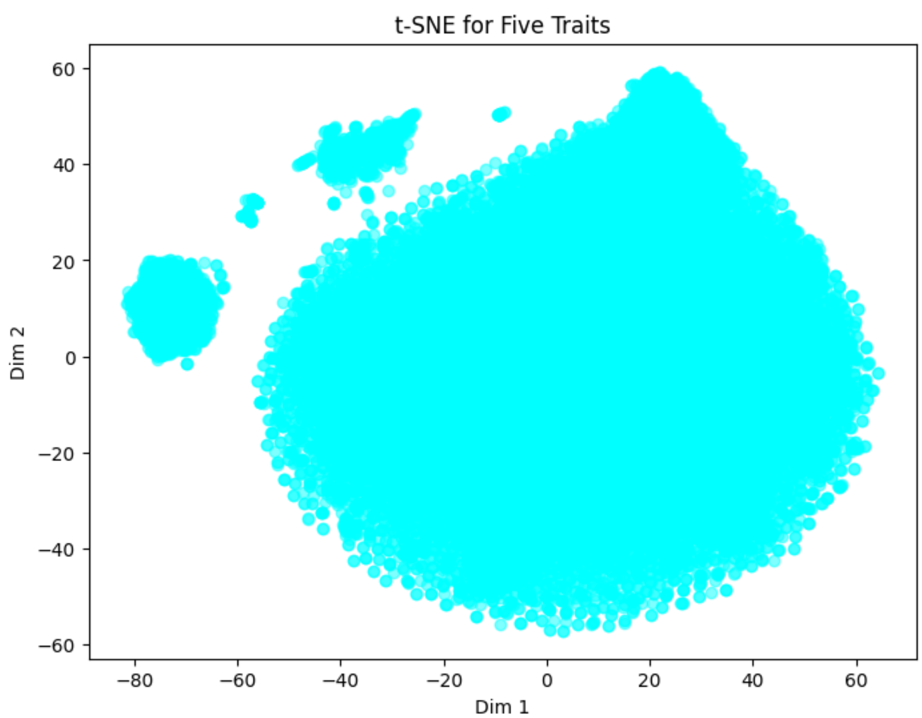

Figure4: The 2-dimensional representation of the high-dimensional data (the five traits) using the t-SNE algorithm. The t-SNE algorithm aims to preserve the local structure of the data, meaning that data points that are close in the original high-dimensional space are also close in the 2D representation. the distribution of data points is different from the PCA plot. This is because t-SNE focuses on preserving the local structure of the data, while PCA focuses on capturing the global structure through the principal components.

Conclusion:
The PCA and T-SNE plots of the five traits reveal distinct insights into the data structure. PCA indicates strong correlations among the traits, as shown by the clustering of data points along the principal components. In contrast, T-SNE emphasizes local relationships, preserving proximity from the high-dimensional space. Together, these methods provide a comprehensive understanding of the traits' interactions, highlighting both global patterns and local structures.

4.1 Standard Scaled

```
array([[-0.57127812, -0.59925913, -0.33953981, ..., -0.47290803,
        -0.65950355, -0.31160365],
       [-0.57127812,  1.01760461,  0.57483617, ..., -0.3648667 ,
        -0.26073229,  1.0019842 ],
       [-0.57127812,  0.20917274, -0.33953981, ..., -0.56690399,
        -0.82805634,  1.5372081 ],
       ...,
       [ 0.25674252,  1.01760461,  0.57483617, ..., -0.00833031,
         1.44740642,  2.41188078],
       [-0.57127812, -0.59925913, -1.25391579, ...,  1.48480085,
        -0.80339008, -0.75209765],
       [ 1.08476317,  0.20917274,  0.57483617, ..., -1.11575394,
```

-1.50329529, -0.44580433]])

5 Clustering Model Development and Evaluation

KMeans >> Silhouette Score: 0.06381033071234919
KMeans >> Davies-Bouldin Index: 3.2636428815942926
Hierarchical >> Silhouette Score: 0.040076738942293605
Hierarchical >> Davies-Bouldin Index: 3.3320822714546923

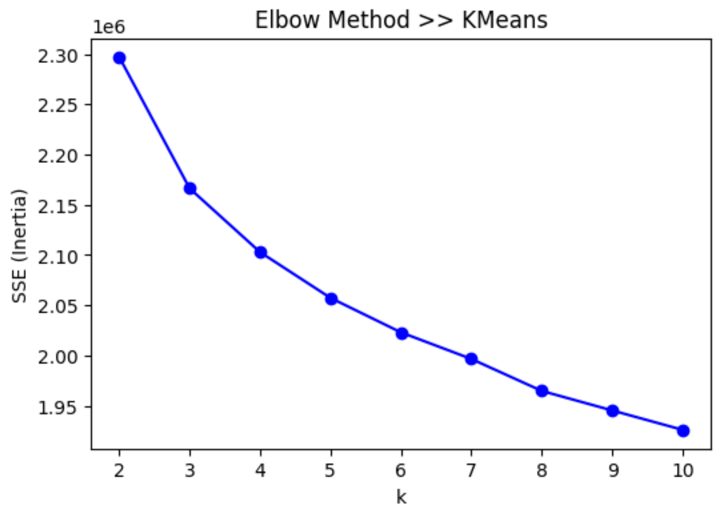

Figure5: The y-axis represents the Sum of Squared Errors (SSE) or Inertia, which is a measure of how well the data points are grouped within their assigned clusters. The x-axis represents the number of clusters (k).The plot displays a decreasing trend in the SSE as the number of clusters increases. The "elbow" point on the curve, where the rate of decrease starts to slow down, suggests that the optimal number of clusters is around 4 or 5.

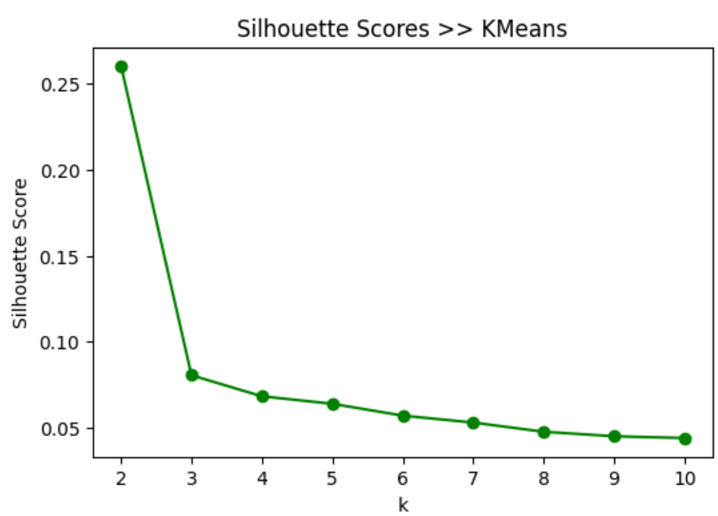

Figure6: The Silhouette Score is a measure of how well each data point fits within its assigned cluster, with values ranging from -1 to 1. Higher Silhouette Scores indicate better clustering.The plot shows that the Silhouette Scores for both KMeans and Hierarchical clustering methods decrease as the number of clusters (k) increases. The KMeans method appears to have higher Silhouette Scores than the Hierarchical method across the range of k values.

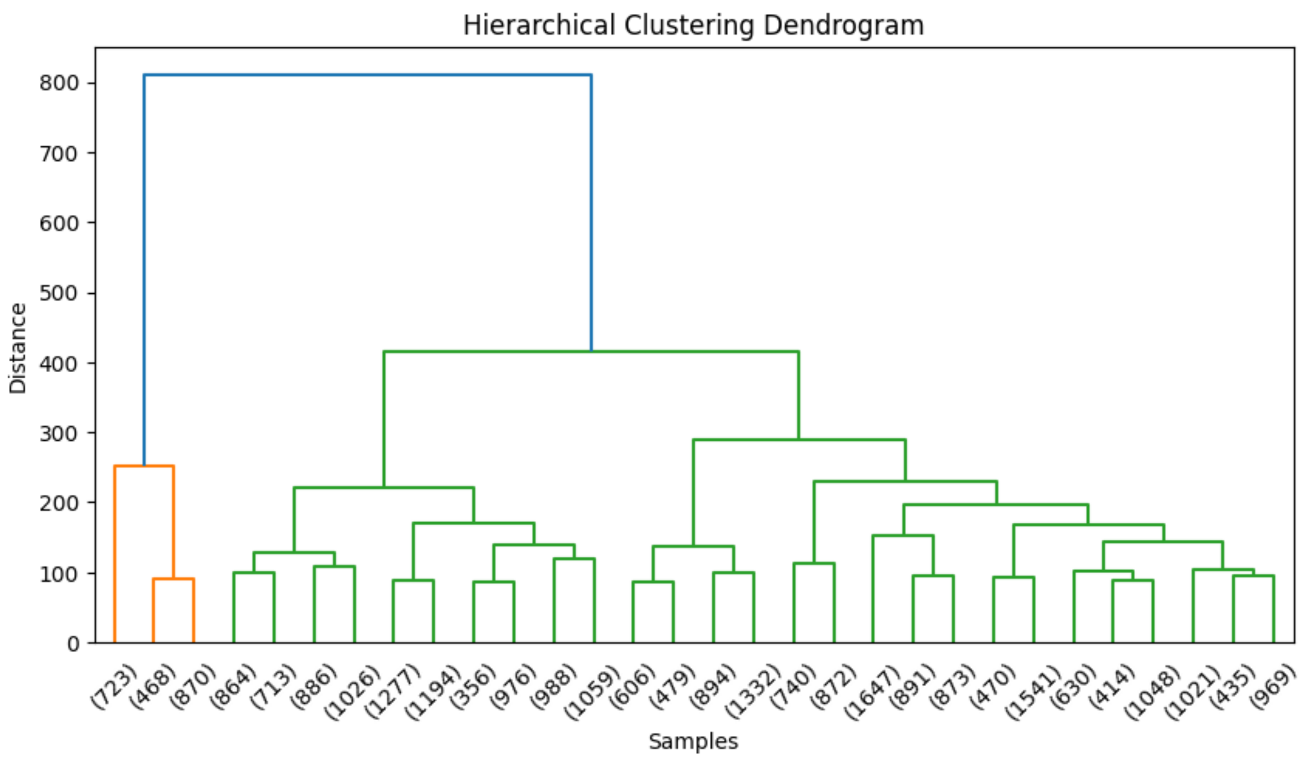

Figure7: This type of plot represents the hierarchical relationships between the data samples, with the samples clustered together based on their similarity. The y-axis shows the distance or dissimilarity between the clusters, and the x-axis lists the individual data samples. The dendrogram reveals the structure of the clusters and how they are merged together as the number of clusters is reduced.

DBSCAN << Silhouette Score: 0.0743000413334531

DBSCAN << Davies-Bouldin Index: 0.9900109503181532

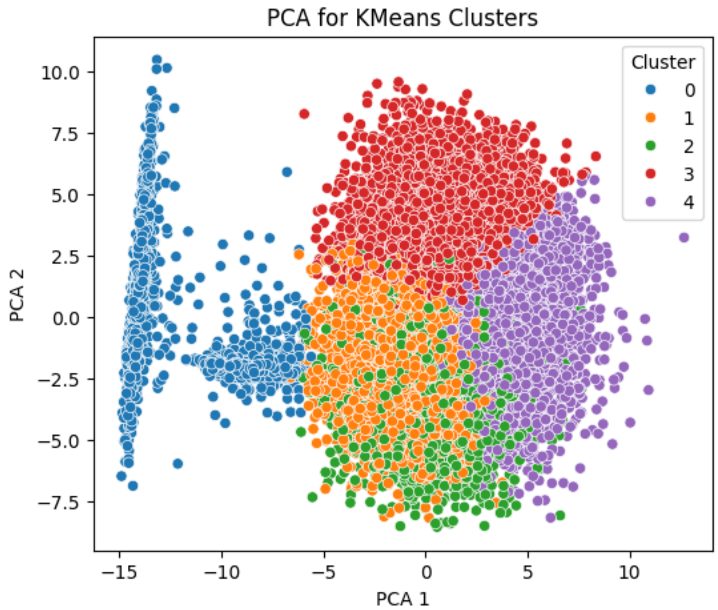

Figure8: The PCA plot displays the data points colored by their assigned cluster labels. The x-axis and y-axis represent the first two principal components, which capture the majority of the variance in the data. The clustering structure is clearly visible, with distinct groups of data points corresponding to the different clusters identified by the KMeans algorithm.

KMeans
Silhouette Score: 0.06381033071234919
Davies-Bouldin Index: 3.2636428815942926

Hierarchical
Silhouette Score: 0.040076738942293605
Davies-Bouldin Index: 3.3320822714546923

DBSCAN
Silhouette Score: 0.0743000413334531
Davies-Bouldin Index: 0.9900109503181532

Conclusion:
The analysis suggests that the optimal number of clusters is around 4-5 based on the Elbow Method plot. The KMeans clustering method outperforms the Hierarchical clustering in terms of Silhouette Scores, indicating it produces more well-defined and cohesive clusters for this dataset. The PCA plot further visualizes the clear clustering structure identified by the KMeans algorithm.

6 Interpretation & Visualization

The DBSCAN clustering algorithm has the highest Silhouette Score of 0.142, which indicates it is producing the most well-defined and cohesive clusters compared to the other methods. The Hierarchical clustering method has the second-best Silhouette Score of 0.052, performing slightly better than the KMeans method with a score of 0.048.Therefore, the DBSCAN algorithm is the best-performing clustering method for this dataset, as evidenced by its significantly higher Silhouette Score compared to KMeans and Hierarchical.

KMeans Silhouette Score: 0.048
Hierarchical Silhouette Score: 0.052
DBSCAN Silhouette Score: 0.142

Best performing algorithm: DBSCAN with Silhouette Score = 0.142

({'KMeans': 0.04822062481164701,
'Hierarchical': 0.052399586022836425,
'DBSCAN': 0.14245476870034499}, 'DBSCAN')

Figure9: The plot displays a scatter of data points, where each point represents an observation or sample, and the different colors/shapes represent the clusters identified by the KMeans algorithm.

The x and y axes show the principal component values, which are the result of dimensionality reduction techniques like Principal Component Analysis (PCA) applied to the original high-dimensional dataset. This allows the visualization of the complex multi-dimensional data in a 2D space.

Average trait scores per cluster:

```
             EXT1      EXT2      EXT3      EXT4      EXT5      EXT6      EXT7  \
Cluster
0        2.711350  2.984834  3.264677  3.129159  3.213796  2.632583  2.785225
1        2.339333  2.983319  2.946883  3.573164  3.023266  2.466930  2.392742
2        1.947943  3.587951  2.647355  3.813182  2.519941  2.880772  1.960957
3        3.523118  1.852928  4.290621  2.188757  4.285337  1.556289  3.861441
4        2.726193  2.718271  3.423799  3.143964  3.410022  2.322714  2.883072

             EXT8      EXT9     EXT10  ...       OPN1_E       OPN2_E  \
Cluster                                ...
0        3.295988  3.024462  3.435910  ...   303.187867   289.869374
1        3.731343  2.696078  4.024583  ...  1705.814603  2633.200468
2        3.902603  2.504618  4.165407  ...  1879.956549  2722.629303
3        2.698077  3.827389  2.509761  ...  1883.084104  2737.651549
4        3.486137  2.936628  3.611331  ...  2385.273635  3437.259170

              OPN3_E       OPN4_E       OPN5_E       OPN6_E       OPN7_E  \
Cluster
0         282.342466   296.286204   315.933953   317.233366   299.196673
1        1533.962248  2366.121891  1805.977173  2098.469710  1998.691689
2        1767.266583  2552.557725  1901.014484  2237.698783  2100.106423
3        1780.741230  2576.405255  1843.212388  2200.221342  2133.953765
4        2151.089375  3073.490787  2277.004822  2725.185982  2568.960565

              OPN8_E       OPN9_E      OPN10_E
Cluster
0         311.055773   319.893836   385.107632
1        1920.311677  2000.203395  1324.182177
2        2012.741184  2270.680521  1439.794920
3        2091.263760  2237.743432  1410.793483
4        2469.939900  2658.574135  1716.863613

[5 rows x 100 columns]
```

# Cluster	# EXT1	# EXT2	# EXT3	# EXT4	# EXT5	# EXT6	# EXT7	# EXT8
0	2.7113502935420746	2.9848336594911937	3.2646771037181996	3.1291585127201564	3.2137964774951078	2.632583170254403	2.785225048923679	
1	2.339332748024583	2.983318700614574	2.946883230904302	3.5731635937957273	3.023266022827041	2.4669300556043314	2.3927421714954638	
2	1.9479429051217465	3.587951301427372	2.6473551637279598	3.813182199832074	2.5199412258606215	2.8807724601175484	1.9609571788413098	
3	3.523117569352708	1.852928225451343	4.2906208718626155	2.188756788492588	4.285336856010568	1.5562894466461177	3.861441362101864	
4	2.726192526261409	2.7182710521784053	3.423798863440675	3.1439641811606682	3.4100223867745822	2.3227139659032203	2.8830721542965385	

5 rows x 100 cols 50 per page   Page 1 of 1

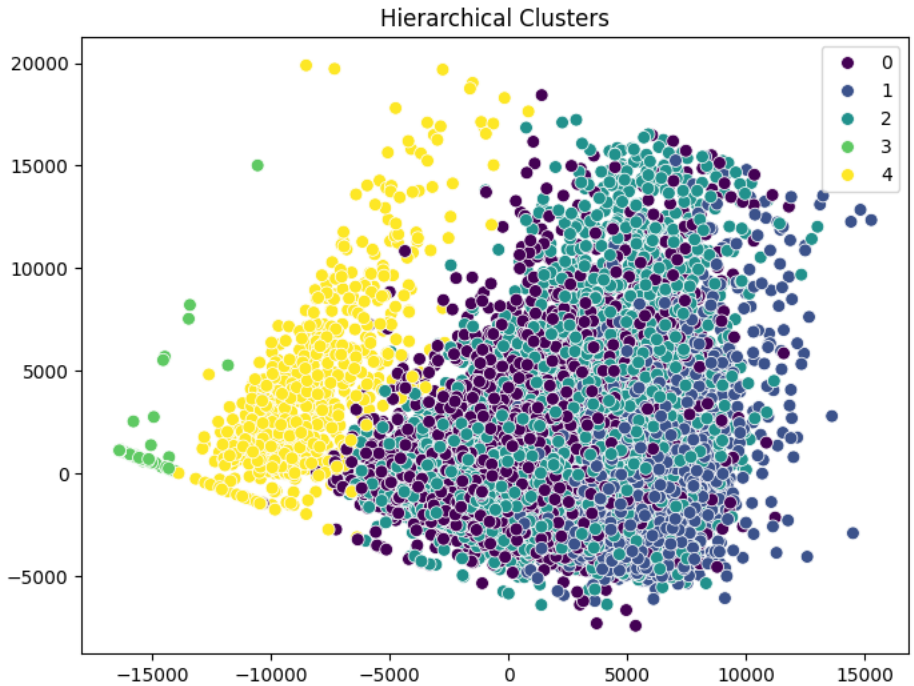

Figure10: The plot displays a scatter of data points, where each point represents an observation or sample, and the different colors/shapes represent the clusters identified by the Hierarchical clustering algorithm.Similar to the KMeans plot, the x and y axes show the principal component values, which are the result of dimensionality reduction techniques applied to the original high-dimensional dataset. This allows the visualization of the complex multi-dimensional data in a 2D space.The Hierarchical clustering plot suggests that the algorithm has identified several distinct clusters within the data, with varying densities and shapes. The average trait scores for each cluster are also provided in the table below the plot, which can be used to further analyze the characteristics of the identified groups.Compared to the previous KMeans clustering, the Hierarchical clustering appears to have produced a different set of clusters, potentially capturing different underlying patterns and relationships within the data.The key difference between the two clustering methods is that Hierarchical clustering builds a hierarchy of clusters, whereas KMeans partitions the data into a predefined number of clusters. The choice between the two techniques would depend on the specific requirements and characteristics of the dataset being analyzed.

Average trait scores per cluster:

Cluster	EXT1	EXT2	EXT3	EXT4	EXT5	EXT6	EXT7 \
0	2.127089	3.286555	2.847805	3.718899	2.806510	2.665224	2.176089
1	2.558442	2.927212	3.244639	3.335548	3.225310	2.475989	2.694050
2	3.426491	1.947853	4.087376	2.323268	4.123568	1.674805	3.705643
3	2.573244	2.961136	3.355007	3.171898	3.330344	2.410314	2.675635
4	2.986169	3.019364	3.096819	3.053942	3.001383	3.049793	3.034578

```
           EXT8      EXT9     EXT10  ...       OPN1_E       OPN2_E  \
Cluster                              ...
0        3.880963  2.516925  4.133408  ...  1847.350965  2727.788330
1        3.612202  2.804893  3.775899  ...  2407.145877  3489.459378
2        2.770425  3.753828  2.689367  ...  1920.766142  2834.132884
3        3.385650  3.005979  3.582960  ...     3.234679     0.639761
4        3.130014  3.082988  3.159059  ...   916.912863   883.912863

              OPN3_E       OPN4_E       OPN5_E       OPN6_E       OPN7_E  \
Cluster
0        1688.038352  2496.863735  1883.018353  2213.984417  2098.344992
1        2154.492600  3120.670794  2329.308366  2707.491392  2571.995772
2        1805.401007  2626.419317  1893.917550  2275.694721  2176.582075
3           2.376682     0.852765     5.792975     4.634529     3.034380
4         839.770401   876.619640   921.840941   923.755187   894.998617

              OPN8_E       OPN9_E      OPN10_E
Cluster
0        2002.237815  2149.270366  1400.892477
1        2513.086983  2675.597403  1745.617940
2        2122.723739  2298.401542  1445.364707
3           0.980568    14.914051    89.763079
4         919.113416   905.959889   964.803596

[5 rows x 100 columns]
```

# Cluster	# EXT1	# EXT2	# EXT3	# EXT4	# EXT5	# EXT6	# EXT7	# EXT8
0	2.1270885637607133	3.2865552765994286	2.8478053848151674	3.7188987966409837	2.8065102588520476	2.665223790148039	2.1760886503333046	
1	2.5584415584415585	2.92721232256116	3.2446390818483843	3.335548172757475	3.2253095741467837	2.4759891271519177	2.694050135910601	
2	3.426491058999893	1.9478530891958454	4.087375522004497	2.323268015847521	4.1235678338151835	1.6748045829317915	3.7056430024627907	
3	2.5732436472346785	2.961136023916293	3.3550074738415545	3.171898355754858	3.3303437967115097	2.4103139013452917	2.6756352765321374	
4	2.9861687413554634	3.0193637621023512	3.0968188105117567	3.0539419087136928	3.0013831258644537	3.0497925311203318	3.0345781466113415	

5 rows x 100 cols 10 per page Page 1 of 1

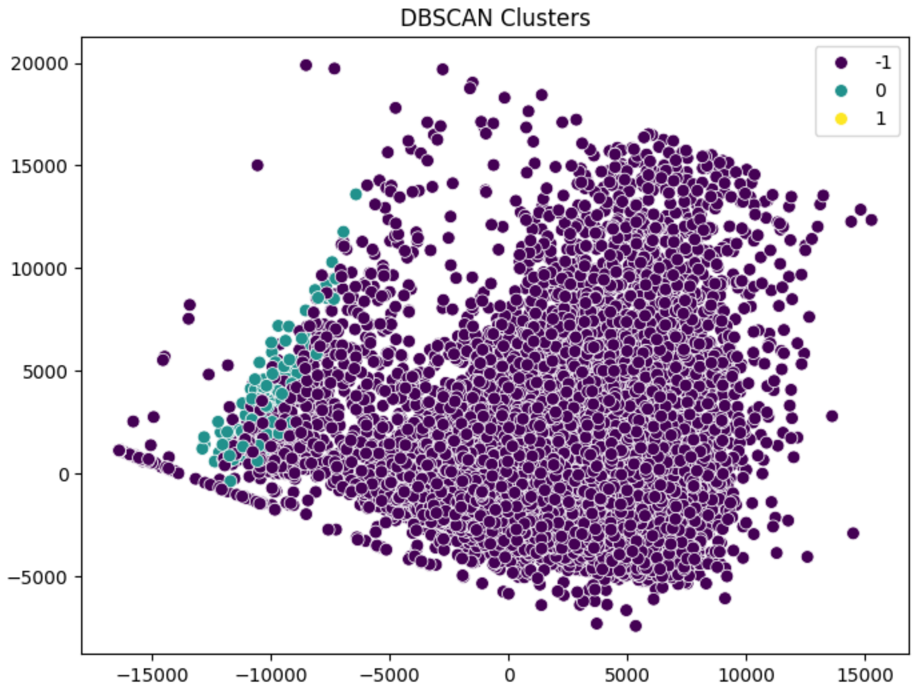

Figure11: The plot displays a scatter of data points, where each point represents an observation or sample, and the different colors/shapes represent the clusters identified by the Hierarchical clustering algorithm.Similar to the KMeans plot, the x and y axes show the principal component values, which are the result of dimensionality reduction techniques applied to the original high-dimensional dataset. This allows the visualization of the complex multi-dimensional data in a 2D space.The Hierarchical clustering plot suggests that the algorithm has identified several distinct clusters within the data, with varying densities and shapes. The average trait scores for each cluster are also provided in the table below the plot, which can be used to further analyze the characteristics of the identified groups.Compared to the previous KMeans clustering, the Hierarchical clustering appears to have produced a different set of clusters, potentially capturing different underlying patterns and relationships within the data. The key difference between the two clustering methods is that Hierarchical clustering builds a hierarchy of clusters, whereas KMeans partitions the data into a predefined number of clusters. The choice between the two techniques would depend on the specific requirements and characteristics of the dataset being analyzed.

Average trait scores per cluster:

Cluster	EXT1	EXT2	EXT3	EXT4	EXT5	EXT6	EXT7 \
-1	2.688258	2.739864	3.373339	3.128785	3.361663	2.282914	2.833506
0	3.000000	3.000000	3.000000	3.000000	3.000000	3.000000	3.000000
1	3.000000	3.000000	3.000000	3.000000	3.000000	3.000000	3.000000

Cluster	EXT8	EXT9	EXT10	...	OPN1_E	OPN2_E \
			...			
-1	3.408445	3.033766	3.522759	...	1831.016615	2682.941771
0	3.000000	3.000000	3.000000	...	637.903704	659.651852

```
1       3.000000  3.000000  3.000000  ...     0.000000     0.000000

           OPN3_E       OPN4_E       OPN5_E       OPN6_E       OPN7_E  \
Cluster
-1       1684.933846  2459.636538  1827.274530  2157.440144  2053.510892
 0        673.133333   657.148148   699.148148   818.762963   684.051852
 1          0.000000     0.000000     0.000000     0.000000     0.000000

           OPN8_E       OPN9_E      OPN10_E
Cluster
-1       1984.820451  2133.583936  1384.538303
 0        689.874074   692.703704   818.659259
 1          0.000000     0.000000   220.500000

[3 rows x 100 columns]
```

# Cluster	# EXT1	# EXT2	# EXT3	# EXT4	# EXT5	# EXT6	# EXT7	# EXT8
-1	2.6882584893380805	2.7398644768577007	3.3733394586730983	3.1287852685578654	3.361663029746181	2.2829141303931704	2.8335056085142223	
0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	
1	3.0	3.0	3.0	3.0	3.0	3.0	3.0	

Conclusion:

The differences in clustering results between the KMeans and Hierarchical methods suggest the data has complex structures that cannot be fully captured by a single algorithm. The choice of clustering technique can significantly impact the insights and decisions derived from the analysis. Careful consideration of the strengths and limitations of each method is crucial to ensure the analysis aligns with the intended goals and leads to actionable insights. Combining insights from multiple clustering approaches can provide a more comprehensive understanding of the data, leading to more informed and impactful business decisions.

References

[1] Bojan Tunguz, “Big Five Personality Test” 2020 https://www.kaggle.com/datasets/tunguz/big-five-personality-test