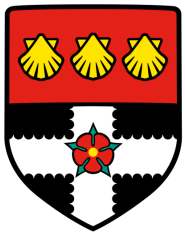


University of Reading

Department of Computer Science

Analysis and Clustering of Personality

Emma Dickie

Supervisor: Carmen Lam

A report submitted in partial fulfilment of the requirements of
the University of Reading for the degree of
Bachelor of Science in *Computer Science*

May 1, 2023

Declaration

I, Emma Dickie, of the Department of Computer Science, University of Reading, confirm that this is my own work and figures, tables, equations, code snippets, artworks, and illustrations in this report are original and have not been taken from any other person's work, except where the works of others have been explicitly acknowledged, quoted, and referenced. I understand that if failing to do so will be considered a case of plagiarism. Plagiarism is a form of academic misconduct and will be penalised accordingly.

I give consent to a copy of my report being shared with future students as an exemplar.

I give consent for my work to be made available more widely to members of UoR and public with interest in teaching, learning and research.

Emma Dickie
May 1, 2023

Abstract

In this report, the aim is to study the effect of different clustering algorithms on a dataset of personality types to see which one performs best and, therefore, what algorithm is best for future study of personality features. Clustering algorithms allow the grouping and classification of data in groups called 'clusters' which helps create separate personality types from the data. To evaluate how each these clusters work, a silhouette score is used which evaluates the distance different points have to other data points in the same cluster as well as the distance between points and points in different clusters. The end goal of this is to produce clusters that are decently separated from one another and internally dense.

To do this, four algorithms are tested – GMM, DBSCAN, Hierarchical, and K-Means. The results of the project show that DBSCAN has the best silhouette score and, therefore, created the best clusters. However, due to the nature of personality typing, more than two clusters are required and DBSCAN doesn't allow easy choosing of the number of clusters, meaning it normally produces just two. As such, this actually leaves K-Means the best algorithm since it can choose the number of desired clusters (Therefore, more than two can be chosen) and it produces the second highest silhouette scores overall. As such K-Means is the algorithm that should probably be used in personality analysis.

Keywords: Clustering, personality analysis, OCEAN

Report's total word count: 12426

Acknowledgements

I would like to thank my supervisor Carmen Lee for keeping me on track.

Contents

List of Figures

List of Tables

List of Abbreviations

SMPCS	School of Mathematical, Physical and Computational Sciences
GMM	Gaussian Mixture Modeling
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
OCEAN	Openness, Conscientiousness, Extroversion, Agreeableness, and Neuroticism

Chapter 1

Introduction

1.1 Background

Personality analysis is a field of research into finding and evaluating the core qualities of an individual such as their determination, kindness, and confidence. This can be used in a number of situations such as for therapy, defining strengths and weaknesses in a person, or hiring people with desirable qualities. (Elias (2020)) In particular, personality analysis has long been considered important by companies and the like for the purposes of marketing or targeted advertising, allowing customisation of users' personal preferences to give them a better experience with shopping and, therefore, more likely to buy more which is beneficial for both a company and its customers. Moreover, this information can also be used in the healthcare field, helping to encourage people to diet, give up smoking, or to get vaccinated. (Graves, Christopher and Matz, Sandra (2018)) As such, the numerous and various uses of personality analysis can be vastly beneficial to many people in many different ways making the improvement of personality analysis a useful field of research.

For personality analysis, there are numerous methods of classing personality types for use. The most commonly found one that most people talk about is the Myers-Briggs Type Indicator, or MBTI. However, while MBTI personality types were a popular method of classifying personality types for a while, it's become criticised over time for its lack of basis in science (Hunsley et al. (2003)). There is a tendency for people to get different answers each time they retake the test implying poor accuracy and it is not recommended for serious guidance. (Cherry (2022*a*))

Other personality classification techniques include the Five Love Languages which is, as the name implies, only used in the context of romance rather than overall personality making it poor outside of that specific situation (Gordon (2023)), Enneagram which has limited research, is considered pseudoscientific and vague, and, as such, is not widely accepted as a valid personality measure (Cherry (2022*b*)), and Cattell's 16 Personality Factors which is considered unwieldy and difficult to use as well as far too ambiguous. (Cherry (2023)) From this, the five personality trait model is considered the best for being the most scientifically accurate. The theory that there are five personality traits was first developed in 1949 by D. W. Fiske (1949) and was steadily adapted and added onto over time by researchers such as Norman (1967), Smith (1967), Goldberg (1981), and McCrae and Costa (1987). Prior to this point, there had been far more personality traits, however, due to overlaps found, these traits were steadily merged and reduced until it reached five. These five traits are most often classed as Openness, Conscientiousness, Extroversion, Agreeableness, and Neuroticism.

Openness is the ability to try new things, conscientiousness is the ability to be thoughtful and goal-orientated such as thinking ahead or being organised, extroversion is the ability to enjoy the company of others, agreeableness is the level of trust and kindness, and neuroticism is the level of emotional instability such as the ability to get upset quickly. (Thomas (2022))

There are various methods of analysing data to gain information from it which could be used for this project. For example, cohort analysis uses historical data to compare a part of a person's behaviour which can then be grouped with similar characteristics, regression analysis studies how one variable is affected when other variables are changed, neural networks can form patterns and make predictions from learning the data, and conjoint analysis how different attributes are valued by a person. However, cohort analysis requires a history such as their usage of an app or website which makes it incompatible with personality types. Likewise, regression analysis studies the changes that certain variables make on the data rather than classifying that data, neural networks are mainly for making a prediction which is not yet known (and personality typing methods such as OCEAN generally have strict boundaries that determine where a specific person falls), and conjoint analysis is specifically for finding what features a user likes which is unrelated to the methodology of OCEAN personality results. (Calzon (2023))

As such, the best kind of analysis for this job is clustering. This is a way of grouping data in such a way that similar data points are classified together, finding patterns in the data. For example, this can be used in the context of this report to classify different OCEAN personality features into different personality types. Since clustering is often used for this purpose, it makes sense to try to perfect the method for future personality analysis.

In this project, the aim is to test which clustering algorithm is the best in the context of clustering OCEAN personality types. This will involve acquiring a dataset of OCEAN results and applying each clustering algorithm to it individually to see what personality classifications each one identifies, how many there are, and how well the results work as clusters.

1.2 Problem statement

Over the years, a lot of research has been done into trying to figure out personality types. Clustering is one of the most frequent methods of doing this, however there are a range of different algorithms that can be used for this task, each with their own pros and cons, making them better against some forms of data and worse against others. As such, knowing what algorithm works best on personality data would be helpful for future research into the topic.

Since one can't try every clustering algorithm, the pool for this project has been narrowed it down to 4 using a mixture of popular clustering algorithms and clustering algorithms which other papers that cluster OCEAN personality types have used in the past. The choices for this paper are K-means and DBSCAN because they're popular methods and Gaussian Mixture Modelling and Hierarchical clustering because they were used in multiple papers on the subject of personality types and therefore were, presumably, already chosen as the best algorithm for the problem before.

In this project, the intention is to test this selection of algorithms against each other in the use of clustering OCEAN personality types (Openness, Conscientiousness, Extroversion,

Agreeableness, and Neuroticism) to find which one works best.

1.3 Aims and objectives

The aim of this project are to test at least two algorithms against OCEAN personality types so they can be compared by factors such as how good their clusters are, the number of clusters they work best with, and how well they deal with noise.

Objectives:

1. Obtain a dataset of results of OCEAN personality types
2. Clean and pre-process the dataset until it's in a state useable for clustering (Two dimensions with no duplicates or NaN values)
3. Individually find the optimal number of clusters for each algorithm and note this value
4. Plot the clusters visually for each algorithm
5. Calculate silhouette score for each clustering algorithm
6. Compare the result of each algorithm to figure out which algorithm has the highest overall silhouette score, which one has the highest score when outliers are within the dataset, and which one has the highest score when outliers are removed from the dataset.

From this, the hope is to determine the best algorithm out of the selection being tested.

1.4 Solution approach

1.4.1 Data collection

For this project, the results of a survey of OCEAN personality results was found from Kaggle. This dataset was obtained from an online test. Most of the data is recorded as a number from a drop down menu which is associated with an answer.

This dataset was chosen because it has all the possible categories this project might need and more. It has 19719 entries which is enough data to get a pretty accurate result even once outliers and such are removed. It was also rated as one of the most popular datasets of this type on the website.

1.4.2 Data Understanding and Pre-processing

To understand and pre-process the data, the initial plans would be to figure out the features of the dataset. This would be the number of records, attributes, attribute types and the quality of the data. This dataset consists of a series of personal information made up of each participant's race, age, whether English is their first language, gender, which hand they write with, where they found the test, and their country as well as their answers with ten for each category of OCEAN, all measured as an answer between 1-5 of how well each question applies to them with 0 as a missed question. There are, overall, 19719 records and 57 attributes.

Data quality can be defined as several factors: Accuracy, completeness, consistency, validity, uniqueness, and timeliness. (AI (2022)) Accuracy is the measure of how much the data matches real life scenarios. Due to being from an online test, the accuracy is a little shaky since people can easily lie or choose incorrect responses to mess around. However, for the most part, people will pick correct options out of curiosity of where they lie in terms of personality types. Moreover, unlike some online tests where they get assigned a category at the end such as MBTI tests with categories such as ISFJ and ENTJ, there's less reason to lie to deliberately try to get one category or another.

Consistency is the uniformity of the data which mostly applies to data stored in different locations which can be different from location to location. Since the data comes from one test, the uniformity is fine. Validity, meanwhile, is whether the data falls into the expected parameters. While most of the personal information is listed as numbers such as gender being a scale of 0-3 which wouldn't be valid, the only data needed for the project are the test results themselves which are a scale of 1-5. The test results themselves are valid so no changes need to be made.

Any duplicate rows are removed to make sure all the data is unique and rows with NaN values are dropped to ensure completeness. Finally, timeliness is whether the data can be obtained when needed. Since the data is not being updated in real time and is, instead, downloaded, the timeliness is also fine.

For ensuring the data quality, data visualisation will also need to be performed. This can allow outliers to be seen and to ensure the data is in the correct format for all the algorithms – for example, some algorithms only take data with a normal distribution so the data needs to be examined for this. Three visualisation techniques will be used for this: a heatmap, boxplots, and a KDE graph. The heatmap can show correlations between variables to see any initial links that will eventually be clustered together. The boxplots show the distribution of data including where the outliers lie. Finally, the KDE graph shows the distribution of data in a different way which helps show the shape of the data – this would show whether or not the data is in a normal distribution or not.

1.4.3 Algorithm Selection

For this project, four algorithms were chosen: K-Means, DBSCAN, GMM, and Agglomerative Hierarchical clustering. K-Means and DBSCAN are two of the most popular density based clustering while GMM and Hierarchical clustering were used in past reports on the subject.

K-Means was the first and the simplest. To start, the algorithm establishes several random points to act as the 'centroids' of each cluster, the number of which is established by the programmer. After that, points are classified under each cluster depending on which centre they are closest to. When this is done, the mean distance of each point to its respective cluster is calculated and this is used to calculate a new position of the centroid so it's closer to the other points within its cluster. After that, the calculation is repeated. This continues to repeat until the centroids no more changes are being made. (Jagota, Arun (2020)) (Garg, Sanjay and Jain, Ramesh Chandra (2006))

Being simplistic as it is, K-Means comes with a few limitations. For one, since the cen-

troids are selected randomly, the final clusters depend on where the centroids were placed. This means a poor initial selection can worsen the results and clusters can even be empty depending on where the centroids are and how many points there are to cluster. The fact that the number of clusters needs to be determined beforehand can also be considered a disadvantage since you need to know already what the best number of clusters is. K-Means tends to be poorer with outliers and has trouble with varying sizes and density as well as problems scaling with different numbers of dimensions.

However, despite these limitations, the simplicity of its implementation, ability to scale to large datasets, and adaptability means there are benefits to it as well. (Google Developers (2022))

The second algorithm was DBSCAN, Density-Based Spatial Clustering of Applications with Noise. Instead of determining the number of clusters beforehand, DBSCAN takes two parameters, Epsilon and the Min Points. Min Points is the minimum number of points that need to be clustered together to be considered "dense" while Epsilon is the distance measure. The algorithm repeatedly finds points of data arbitrarily until all have been found. Then, it calculates if the number of Min Points are within the epsilon radius. If this is true, they are clustered together. (Singh Chauhan, Nagesh (2022))

DBSCAN is good with arbitrarily shaped clusters as well as clusters surrounded by another different cluster. They're also good with outliers and great at separating clusters of high density from clusters of low density. However, on the downside, DBSCAN struggles with datasets with varying densities, high dimensionality datasets, and is sensitive to its variables: Epsilon and Min Points. (Shilpa, Dang (2015)) (Engati (2021))

Then there's Gaussian Mixture Modeling (GMM). Each point is initially assigned to a random cluster. To do this, Gaussian distribution is used which consists of a mean and a variance. Each cluster is given its own mean, covariance, and density, generated at random. Using an Expectation-maximization algorithm (EM), the probability is calculated that each point really does belong to the cluster it was assigned to. From there, parameters are updated to get closer to the actual clusters. The new density is calculated using the ratio of the number of points in the cluster and the total number of points. Meanwhile, the mean and covariance matrix are updated using values assigned to the distribution in proportion to the probability values for each data point. As such, the higher the probability that a point is part of a distribution, the more it contributes to the new mean and covariance. Like in K-Means, this process is repeated over and over again until the likelihood is maximised. (Singh, Aishwarya (2022)) (Maklin, Cory (2019*b*))

Gaussian Mixture is the fastest algorithm for mixture models and, moreover, doesn't bias the mean towards zero or bias the clusters towards any particular size or structure that might not apply, making it good for clusters of varying sizes or non-spherical clusters. It's also less sensitive to scale. However, there are quite a few drawbacks too. For one, this algorithm will always use all the components it has access to, regardless of how well they may or may not apply. It also assumes a normal distribution for features which makes it poor on categorical data and non-normally distributed numeric values. While it can handle different cluster shapes, it also assumes an elliptic shape which can cause it to perform badly on irregularly shaped clusters. Like other clustering algorithms, it also requires the number of clusters to be defined beforehand and is sensitive towards outliers. Similarly to K-Means, GMM is also sensitive

to its initial conditions. The initial mean, covariance, and density randomly generated can influence how well the clusters score meaning running it multiple times can achieve different scores. On top of this, it's a slower algorithm to run. (Ellis, Christina (2021)) (Sckit Learn (2023))

Finally, Hierarchical Clustering is attempted. Specifically, an agglomerative approach, a 'bottom up' approach. While a divisive approach tends to produce more accurate hierarchies and is more efficient, it is far more complex, making it difficult to program. (GeeksForGeeks (2022))

First, it classes every single point of data as its own group, then starts merging points together to create larger and larger groups until there is only one cluster left. The number of desired clusters can be calculated by visualizing all the different merges as a dendrogram. Then, the dendrogram is cut where the lines are longest to obtain the optimal number of clusters – the number of clusters will be the number of dendrogram lines intersected. This means no assumptions need to be made about the number of clusters to use. (Bock (2022))

There are several methods in Hierarchical to determine how to group together points. For example, one can use the minimum distance between clustering points. For this report, Ward's method is used. This method uses the calculation of the sum of the square of the distances which makes it good against noise. It is biased towards globular clusters however. (Reddy Patlolla, Chaitanya (2018))

Agglomerative hierarchical clustering is simple to implement and there is no need to pre-specify the number of clusters – looking at the dendrogram can easily find the required number. However, if points are grouped incorrectly at an earlier stage in the clustering process, this can't be undone. Moreover, due to the different distance metrics that can be used, a lot of different results can be produced. Moreover, hierarchical clustering is overall poor on vast amounts of data and all different measures that can be used come with their own disadvantage. Irfana Sultana, Shaik (2020)

1.4.4 Result Evaluation

To evaluate each algorithm, a silhouette score will be used to evaluate how good each result is. Since there aren't any pre-determined categories for the data to be clustered into, this project is focused on unsupervised learning, meaning only unsupervised methods of measuring accuracy can be used. (Delua, Julianna (2021)) Silhouette scores combines the inter-cluster distances between points with the intra-cluster distances between clusters to calculate how good a clustering algorithm is. For a good algorithm, clusters should be clearly distinguished – as in, far apart from one another to be clearly their own cluster – and the distances between points within a cluster should be close enough together to justify them being categorised the same. While some other techniques will test one or the other of these factors, a silhouette score tests both factors making it the best candidate for measuring how well each algorithm worked. The closer the score is to 1, the better it is. (Bhardwaj, Ashutosh (2020))

Several types of distance equations can be used to calculate the silhouette score such as the Minkowski distance, the Manhattan distance, and the Euclidean distance. Whenever a distance matrix is needed for this paper, the Euclidean distance is used. The Euclidean

distance is the most commonly used distance matrix, using Pythagoras theorem to calculate a straight line distance. Since no particular complexity is needed and using any more obscure distance measures such as Manhattan which uses a grid for distance would probably negatively affect results, using Euclidean is best. (Gohrani, Kunal (2019))

For each algorithm being tested, data pre-processing techniques were used, removing rows with missing values and removing duplicate rows. Moreover, the data set used for this project was in the form of different answers for a test with ten questions per personality type, all rated from 1-5. As such, the mean was calculated for each category to combine them all into one column scoring each personality type. From there, since nothing else was required for the goal of the project, all other columns except the means were dropped.

Next, several graphs were used to look at the data in more detail: A heatmap, a boxplot, and a KDE. First was the heatmap which showed correlations between data. From this, it can be seen that correlations between results were mostly weak. The strongest, with a value of 0.3 was between Openness and Conscientiousness. Following that up with a correlation of 0.2 was Extroversion and Agreeableness, Neuroticism and Conscientiousness, and Agreeableness and Neuroticism. The weakest correlation with a value of -0.02 was Neuroticism and Extroversion.

The next graph was a box plot to see the distribution of results and the potential outliers in the results. Most of the personality types were quite tightly distributed except for neuroticism which had the largest spread and the fewest outliers.

Finally, was the KDE plot to see the probability density of the results. Like the box plot, this showed most of the personality types being tightly distributed around 3 except for neuroticism which was more widely spread.

1.4.5 Dealing with Outliers

Some algorithms do better with outliers than others. In terms of testing different algorithms, this could prove an issue since it means some algorithms won't be working at peak performance with the outliers in and taking the outliers out might affect the algorithms that already have ways of handling outliers such as DBSCAN. Moreover, the initial problem of testing what algorithms do best on clustering personality types includes nothing about whether outliers should remain in or out.

As such, the decision made was to make two datasets. One with the outliers in and one without and then testing both on each algorithm. That way, one would also be able to see whether or not outliers would be a problem when deciding what algorithm to use.

1.5 Summary of contributions and achievements

In this report, four algorithms are tested – K-Means, DBSCAN, GMM, and Agglomerative Hierarchical. Each is ran through different methods to obtain the optimal number of clusters and then tested on a dataset with outliers and a dataset without outliers.

DBSCAN is the algorithm which does the best, both with outliers, without outliers, and just overall, although it does best on a dataset with the outliers left in. As such, this would imply that, in clustering personality types, DBSCAN is the best option to choose because it obtains the most distinguished clusters – different clusters are well separated from one another and the points within each cluster are decently close together.

However, based on past literature, most other papers require a specific number of clusters to test a hypothesis and DBSCAN is a problem in this sense because the number of clusters can't be easily established by the user. As such, the second best algorithm is K-Means. K-Means is shown to work best with outliers removed and allows the user to establish the number of clusters for themselves. Meanwhile, GMM and Hierarchical work less well, implying that these are poor choices for the task.

It is also discovered that all of the algorithms work best with two clusters. However, this number is often rejected in past literature since both personality types classified by this method would simply be reflections of each other. As such, K-Means did best with three clusters while GMM and Hierarchical did best with four, implying that there is most likely three or four main personality types that can be found within the data. Adding more means the clusters become less well defined in terms of silhouette score.

1.6 Organization of the report

This report is organised into six chapters: Chapter 1 the introduction, chapter 2 the literature review, chapter 3 the methodology, chapter 6 the conclusion, chapter 4 the results, and chapter 7 the reflection.

In the introduction, in section 1.1, the background context of the project is explained. Section 1.2 lists what the problem this project attempts to solve is and why this project is needed while section 1.3 lists the aims and objectives that this project aims to achieve to solve this problem. Section 1.4 is how this problem will be approached and section 1.5 lists what this project manages to accomplish in the end. Finally, 1.6 summarises the organisation of the report and what to expect in each chapter and section.

For the literature review, section 2.1 summarises past papers into the subject of clustering algorithms and personality clustering. Section 2.2 lists the similarities between these papers and, as such, section 2.3 lists what this project should do to as influenced by these papers. Finally, section 2.4 summarises everything the chapter covered.

In methodology, section 3.1 covers how different algorithms will be implemented and section 3.2 summarises this in a shorter way.

Next is the results. Section 4.1 covers the outcome of each algorithm and how they compare to one another while section 4.2 summarises this.

Then comes the discussion and analysis. Section 5.1 covers the results in more detail such as what the clusters looked like. Section 5.2 is about the significance of these results - how much they can be taken at face value and what they imply about the work. Section 5.3 covers

the limitations of the project and section 5.4 summarises everything.

In conclusions and future work, section 6.1 reiterates what was done in this project and what was found from the results while section 6.2 covers what could be done in future papers to improve and expand on the work done in this project.

Finally, the entirety of chapter 7 covers what the author has learnt through this experience and what changes were made to the project as things progressed.

Chapter 2

Literature Review

2.1 Summarization of prior work

For this project, multiple past papers were looked through, all on the topic of clustering personality types to find out where gaps lay in research and what this project could do to bridge those gaps. In total, five papers were looked at. Three were in the topic of clustering personality types, two were papers on comparing algorithms.

2.1.1 Paper 1

Sava and Popa (2011)

1073 participants from Romania within the age group of 16 to 60. Categorised by whether they were rural or urban, whether they were 16-25 or 26-40 or 41-60 in terms of age, ethnographically significant regions, and whether they were male or female. When the percentages were tested, the results were 49.3 percent (urban) and 50.7 percent (rural); 26.5 percent (16-25 years), 34.2 percent (26-40 years), and 39.3 percent (41-60 years), 50.1 percent males and 49.9 percent females. This categorisation allowed for random samples to be taken from each category.

The objectives of this paper were finding out how many clusters are the most appropriate, testing replicability of results, and testing how different traits affected the results.

Outliers were removed by removing social desirability scores over two standard deviations above the average and unusual cases outliers at a p ¡ .00 level based on Mahalanobis distance statistics. The remaining cases (1039 participants) were randomly allocated to two sub samples for cross-validation purposes, comparing results between the remaining 1039 and the initial group.

The paper used Ward's algorithm followed up by K-means non-hierarchical method to validate how many clusters should be used. The result was two, three, and five clusters were the best numbers. However, less faith was placed in the two cluster solution since both clustered personality types would simply be mirror images of one another. From there, each number of clusters were analysed to see what personality types could be found in each cluster.

The results found were that the two cluster solution was found to be less accurate than the other solutions but, from a five cluster solution, five personality types were determined: the

Undercontrolled, the Strain, the Resilient, the Overcontrolled, and the Passive. Moreover, the Resilient type occurred more frequently amongst university graduates and post-graduates and less frequently amongst those with less years of education. On the reverse side, the Overcontrolled type occurred more amongst those with less than 12 years of education. Postgraduates also have the highest frequency of Passive personality types.

There was also found to be a higher percentage of Resilient and Undercontrolled in the urban milieu while Overcontrolled and Strain occur more in the rural milieu. Passive types are also more frequent in the centre of cities while Strain occurs more in the suburbs.

Besides only testing the Romanian population, only two algorithms were used, creating a possible limitation in the experiment. After all, besides assumptions made about the data, it can't be certain that K-Means is the best clustering algorithm for the experiment.

2.1.2 Paper 2

Reece (2009)

680 university students were sampled for this paper. From this sample, 497 (73 percent) were females and 183 (27 percent) males, all between 15 to 64 years old. 395 were Caucasian (58 percent), 185 African-American (27 percent), 61 Asian American (9 percent), 39 Hispanic, mixed race or other (6 percent), all derived from survey results. This sample was used to prove or disprove the hypothesis that there are three discernable personality types and that they fit the Block and Block's (1980) three empirically derived personality types – Resilients who are generally well adjusted both socially and cognitively, Undercontrollers, who are often impulsive and antisocial, and Overcontrollers who tend to be shyer and require a more rigid control.

To test this, the results were split randomly into two groups. The first group was analysed using a hierarchical clustering method followed by a non-hierarchical clustering method to fine-tune results. Specifically, the average-linkage hierarchical method was used because it avoids the bias in Ward's method towards clusters of equal sizes. The analysis was conducted twice, once for a three-cluster solution and a second time for a four-cluster solution.

It was found that there were three discernible personality types and it was also found that they were not Block's types (Block and Block, 1980) since none of them properly matched the Undercontroller type, though the other two clusters could have been fitted to Block's types.

Like the last paper, only two algorithms were used. Moreover, both papers had a conflict over whether Ward's method or the average linkage method should be used.

2.1.3 Paper 3

Ligato (2021)

The method and datasets from Gerlach et al. (2018) were used and replicated to prove or disprove several hypothesis' to find what personality characteristics have in common.

- Hypothesis 1: Following R equivalent code of the Gerlach study will show the same clusters as the prior study. This provides first pass support for the equivalence of data,

allowing for the further study into controlling for social desirability and central tendency biases.

- Hypothesis 2: Accurate preprocessing will get rid of the social desirability bias in the data sets, which will make the Role Model cluster disappear.
- Hypothesis 3a: Accurate preprocessing will get rid of the central tendency bias in the data sets, which will make the Average cluster disappear.
- Hypothesis 3b: Accurate preprocessing will get rid of the central tendency bias in the data sets, which will make the Self-Centered cluster disappear.
- Hypothesis 3c: Accurate preprocessing will get rid of the central tendency bias in the data sets, which will make the Reserved cluster disappear.
- Hypothesis 4: All cluster analytic results will be compared and no clusters will appear as anything but noise

For the algorithm, Gaussian Mixture Modelling was used with a Bayesian Information Criterion (BIC). BIC is stricter about false positives and, since multi-cluster analytic techniques were used, certainty needed to be made that all clusters were representative of genuine results rather than noise.
Meanwhile, GMM was used because it accounts for potential unequal sizes of clusters and allows for covariance between the clusters which gives it an edge over K-Means in terms of nuanced predictions.

For the results, they were as follows.

- Hypothesis 1 Results: Hypothesis 1 was supported. Since Gerlach et al. (2018) made seemingly arbitrary distinctions in terms of showing equivalence with prior research, essentially everything could be supported as being equivalent.
- Hypothesis 2 Results: The researcher considered the hypothesis to be partially supported since the Role Model cluster was not substantiated in the more advanced data analytic techniques after strict data cleaning procedures were used.
- Hypothesis 3a Results: Upon further review and comparison with prior literature, the Average cluster was not a legitimate cluster and was not supported in enough of the analyses to say it was anything but a statistical abnormality in prior data sets. The researcher considered this hypothesis to be supported since the cluster was not substantiated in the research.
- Hypothesis 3b Results: Upon further review and comparison with prior literature, the Self-Centred cluster was not a legitimate cluster and was not supported in enough of the analyses to say it was anything but a statistical abnormality in prior data sets. The researcher considered the hypothesis to be supported since the cluster was not substantiated in the research.
- Hypothesis 3c Results: The Reserved cluster was not found to be a legitimate cluster and was not supported in enough by analyses to say it was anything but a statistical abnormality in prior data sets. The researcher considers this hypothesis to be supported since the cluster was not substantiated in our research.

- Hypothesis 4 Results: This hypothesis was partially supported. The Role Model cluster turned up in quite a few of the analyses. However, the more the data was cleaned up and the more advanced cluster analytic techniques that were used, the less likely it was to find the Role Model cluster.

Only one algorithm is used for this and, yet again, it matches up with none of the past papers in terms of what algorithm they think is best.

2.1.4 Paper 4

Abbas (2008)

For this paper, multiple datasets were found online with varying noise, size, and randomness. for the purpose of testing four clustering algorithms against each other to see which algorithms perform best. The algorithms are tested on several factors – the size of the dataset used, the number of clusters, the type of dataset, and the type of software. The specific algorithms tried were K-means, agglomerative hierarchical clustering, Self-Organisation Map (SOM), and Expectation Maximization (EM)

The results are that with the number of clusters, the k-means algorithm performance gets better while the SOM algorithm worsens. The performance of the k-means algorithm and EM algorithm are better than the hierarchical algorithm. SOM shows more accuracy classifying objects into their suitable clusters compared to other algorithms. As the value of k (number of clusters) increases, the accuracy of the hierarchical algorithm increases until it matches the SOM algorithm. K-means and EM have less accuracy than the others but their accuracy becomes very good when using huge datasets. Hierarchical clustering and SOM show good results when using small datasets. Hierarchical clustering and SOM show better results when using a random data set and the vice versa. K-means and EM are sensitive to noise in a dataset. A hierarchical clustering is more sensitive to noise than SOM

For this paper, it only tests four algorithms, none being the preferred ones used in the papers about personality clustering like what this report is about. Moreover, the author believes that other factors could have also been considered when comparing algorithms and that normalised and non-normalised data would have yielded new results. They also believe that testing the algorithms in an application of some sort such as character recognition software could have given more information to work with.

2.1.5 Paper 5

Fasulo (1999)

This paper examines four recent papers on clustering agoirthms, examining the strengths and weaknesses of each approach. For this, the following algorithms are examined: K-clustering, Hierarchical Clustering, Input to Clustering Algorithms, Banfield and Raftery: Mixture Models, Gibson, Kleinburg, and Raghaven: Dynamical Systems, Agrawal, Gehkre, et. al.: Subspace Clustering, Ben-dor and Yakhini: Clique Graphs

The result is that subspace clustering and Clique graphs handled mixed data the best while Clique graphs and Dynamical systems did best with categorical data. Mixture Models did best

with data that was numerical with clusters that were spherical or ellipsoidal. The EM clusters weren't easily scalable due to slow converging but Dynamical systems and Subspace clustering can handle large input sets (Although Subspace Clustering takes a lot more time to process as the number of dimensions of the input increases.) K-clustering, Hierarchical Clustering, and Input to Clustering Algorithms don't deal with noise but all the other algorithms do in some way or another.

Like the last paper, these results aren't tested on personality clustering.

2.2 Overall comparison

Out of the five papers, several things could be determined.

1. Papers 1-3, the ones testing personality clusters, used large and varied datasets
2. Paper 1 used Ward's Algorithm and K-Means, Paper 2 used Hierarchical clustering and Average Linkage Method, and Paper 3 used GMM
3. 3-5 Clusters tended to be used most commonly for personality clustering
4. Paper 4, the one comparing clustering algorithms, used various different datasets
5. Paper 4 tested Hierarchical Clustering, K-means, SOM, and EM
6. Most of the papers only use a limited number of algorithms. Papers 1-3 only tested one each while 4 and 5 didn't test any algorithms in an actual application such as personality clustering. Neither do 4 and 5 take into account the difference between normalised and non-normalised data.

2.3 Effect on this paper

From this, it can also be determined that, although there are plenty of papers comparing personality types, none of them agree on which algorithm to use. Moreover, from looking at papers comparing clustering algorithms, none of them tested the algorithms in any specific scenario – they just clustered random data. As such, this project can cover both bases. Better information on clustering algorithms and which ones to use in which situation can improve future data analysis since people can know what option is best for the job in advance.

For this project, these past papers seem to imply a large dataset should be used and, therefore, clustering algorithms that work better on that type should be tested. Out of paper 4, the optimal algorithms for this seem to be implied to be K-means and EM. There are also only a few clusters used in most of the papers so the algorithm will only need to deal well with a small number of clusters. Finally, since personality tests are usually measured numerically, the algorithms only need to work well on numerical data rather than categorical.

It can also be determined that this project should have several factors to gauge how well each algorithm does. Ignoring factors like size of the dataset because these will remain consistent, paper 4 and 5 use these as measurements:

1. Ability to deal with noise

2. How many clusters it does best with
3. Accuracy/Quality

These factors will be taken into account when designing this project.

2.4 Summary

In summary, there are numerous papers on the subject of personality clustering and numerous comparing clustering algorithms. For this project, five of them were examined, three on personality clustering and two on comparing clustering algorithms. However, none of them compare clustering algorithms in the context of personality clustering. Neither can any papers on personality clustering agree on a single algorithm or method that works best for the job. As such, the goal of this project will be to correct the gap in this research by finding which algorithm works best, using factors determined by past papers to decide which algorithm is best.

Chapter 3

Methodology

3.1 Methodology

3.1.1 Algorithm Pre-processing

Due to the differences between clustering algorithms, different forms of pre-processing are needed for matters such as cleaning the data to deal with any weaknesses the algorithm has or using different methods and techniques to obtain the best variables for the algorithm to get the best result in turn.

Hierarchical Clustering

For hierarchical clustering, not a lot of pre-processing is needed. However, to calculate the number of clusters required, the hierarchical clustering process needs to be plotted as a series of dendrograms. This plot would show each time two points are merged into one group and, therefore, displays how many clusters exist at each stage of clustering. To get the optimal number of clusters, the goal is to draw an intersecting line across the tallest vertical line. The number of lines intersected by this new line is the number of clusters that should be used. (Sharma, Pulkit (2022))

GMM

For Gaussian Mixture Modelling or GMM, there are several methods for determining the optimal number of clusters such as calculating the distance between GMMs or using Bayesian information criterion. However, one of the methods is to calculate the silhouette score of each number of clusters and find the optimal number. Since the goal of this project is to find the optimal silhouette scores of different clustering algorithms, this was the method chosen. (Lavorini, Vincenzo (2018))

To perform this method, the numbers of clusters tested was 2, 3, and 4. Since previous literature didn't go above five, testing any higher would be needlessly complex and a waste of time and computation. Moreover, due to this, instead of creating a plot, for this experiment, each of these numbers of clusters were tested individually. This would help also compare the results to previous pieces of literature on the subject and their choices of cluster numbers to confirm how good their choices were.

K-Means

K-Means is a simple clustering algorithm but has a few limitations. For one, the number of clusters needs to be calculated in advance. A common method for solving this this is the elbow method.

The elbow method can either be done with distortion - average of the squared distances from each cluster center – or inertia - sum of squared distances of each point to their closest cluster center. For this project, inertia is used. From there, the inertia is plotted for each value of K. From there, the aim is to find the point where the line starts decreasing in a linear fashion, aka, the 'elbow' of the curve. (GeeksforGeeks (2023))

Another limitation of K-Means is that it does poorly with outliers. As such, it is predicted to get a better score with the non-outliers dataset.

DBSCAN

DBSCAN needs two variables to function and, as such, requires decent amount of preprocessing to find the optimal value for both of them. For this, two loops are run, testing each value of epsilon with each value of Min Points, adding each resulting silhouette score and its respective variables to a list. Since not all values of epsilon and min points can be tested and the time and computation can get very long for this test, epsilon values will be tested in the range of 0.05 to 0.13, testing in 0.01 intervals while min points were tested from 10-21 in intervals of 1. Past literature on the subject of clustering didn't use DBSCAN so there were no examples to reference for these intervals so an educated guess was made to assume that the optimal number of points would sit somewhere in there. If the outcome shows evidence otherwise (such as the optimal value outputted by the algorithm showing to be a value right on the edge of this range), these values will be changed and tried again. From there, all the values in the newly created list will be sorted from best to worst based on silhouette score and the best will be outputted.

DBSCAN can supposedly deal with outliers itself so the prediction is that it will either remain about the same whether the outliers are there or not or it will perform better with the outliers left in.

3.1.2 Data Visualisation and Cleaning

With any form of data science, there are several steps that need to be followed. Defining what the end goal is, data collection, data cleaning and pre-processing, analysis, and results. (Hillier, Will (2023)) The end goal in this project was to use multiple clustering algorithms on one set of data and data collection was done by finding a pre-made dataset online from an interactive online personality test, making it second-party data.

Dimensionality and Demographic Analysis

To start with, the data - figure 3.1 - was visualised in different ways to examine what it contained. It had 57 columns and 19719 rows in total, consisting of columns for race, age, whether or not the participant was an English native speaker, their gender, their dominant hand, their country of origin, where they found the test, and each and every one of their

Figure 3.1: Initial Dataset

	race	age	engnat	gender	hand	source	country	E1	E2	E3	...	O1	O2	O3	O4	O5	O6	O7	O8	O9	O10
0	3	53	1	1	1	1	US	4	2	5	...	4	1	3	1	5	1	4	2	5	5
1	13	46	1	2	1	1	US	2	2	3	...	3	3	3	3	2	3	3	1	3	2
2	1	14	2	2	1	1	PK	5	1	1	...	4	5	5	1	5	1	5	5	5	5
3	3	19	2	2	1	1	RO	2	5	2	...	4	3	5	2	4	2	5	2	5	5
4	11	25	2	2	1	2	US	3	1	3	...	3	1	1	1	3	1	3	1	5	3
5	13	31	1	2	1	2	US	1	5	2	...	4	2	1	3	3	5	5	4	5	3
6	5	20	1	2	1	5	US	5	1	5	...	3	1	5	1	4	1	4	3	3	4
7	4	23	2	1	1	2	IN	4	3	5	...	3	1	5	1	4	1	5	3	2	5
8	5	39	1	2	3	4	US	3	1	5	...	3	3	5	3	5	1	5	3	4	5
9	3	18	1	2	1	5	US	1	4	2	...	4	2	5	2	4	1	4	3	4	4

10 rows × 57 columns

answers. Except for country, this was all done through numbers, each number corresponding to a value in a dropdown menu.

57 columns is very high dimensionality and neither is all of it necessary.

Further visualising the data before the removal of any columns unnecessary for the clustering itself, the distribution of people who took the initial test was found. From the results, shown in figure 3.2, the demographic consisted mostly of men of European Caucasian descent. Due to the age being found to be incredibly high, outliers were removed to find the mean at 54.51.

From this, it's clear that the data isn't balanced in terms of demographics. However, since the main goal is the clustering itself rather than obtaining any highly accurate results about the data, no changes were made to this. Balancing the data would mean losing some of the data after all which would impact the results negatively.

Because there were so many columns, too many to cluster, dimensionality was reduced by grouping the test answers together. All ten answers for openness were added together and divided by 10 to get the overall mean for Openness, and the same was done to every other personality feature in OCEAN until five new columns existed containing the results. After that, all duplicate rows were dropped to ensure that any errors in the dataset that could have created copies of the same row were dealt with. Then, at last, every single column was dropped except the five new columns for personality types. After all, for the task at hand, only the personality types themselves were needed. If the goal was to find patterns in personality traits and who they came from, some of the other columns could have been left in but this was not the goal this time.

Figure 3.2: Participant Diversity Chart

Demographic Gender	N= 19719 Percent rounded to 2 decimal places
Male	60.78
Female	38.58
Other/Unanswered	0.64
Age (Mean)	54.51
Race	Percent rounded to 2 decimal places
European Caucasian	53.44
South East Asian	9.44
Indian Caucasian	7.7
Mixed Race	7.27
Middle East Caucasian	2.61
North African or Other Caucasian	2.01
Other	17.53

StackOverflow (2022)

Correlation Analysis

The next step from there was to visualise the data to see what initial trends and patterns existed within it.

Using a heatmap, figure 3.3, correlations within the data were visualised.

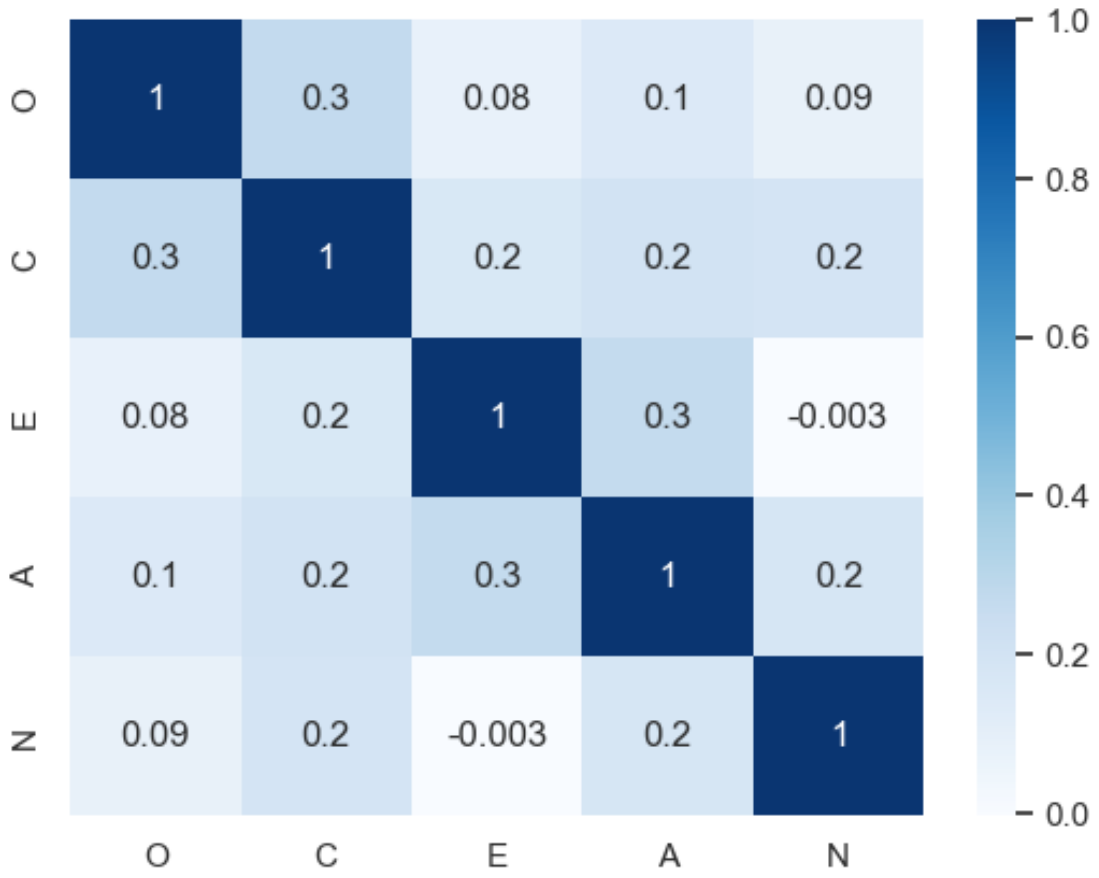
The strongest correlations in the data existed between Openness and Conscientiousness as well as Extroversion and Agreeableness at a correlation score of 0.3. This is still relatively weak, implying that most of the features are relatively independent of each other which is good for personality types. It means nothing is overly similar to each other so every personality feature matters and, as such, there are no columns that need to be dropped or merged to reduce dimensionality or unnecessary data that could affect the results.

Distribution Analysis

Next, a boxplot was used to visualise. This shows the distribution of values in the dataset.

From the graph in figure 3.4, it can be seen that O, C, E, and A are all rather densely dis-

Figure 3.3: Heatmap of Correlations



PythonBasics (2021)

tributed around 3-3.5. Neuroticism, however, is very widely distributed with very few outliers.

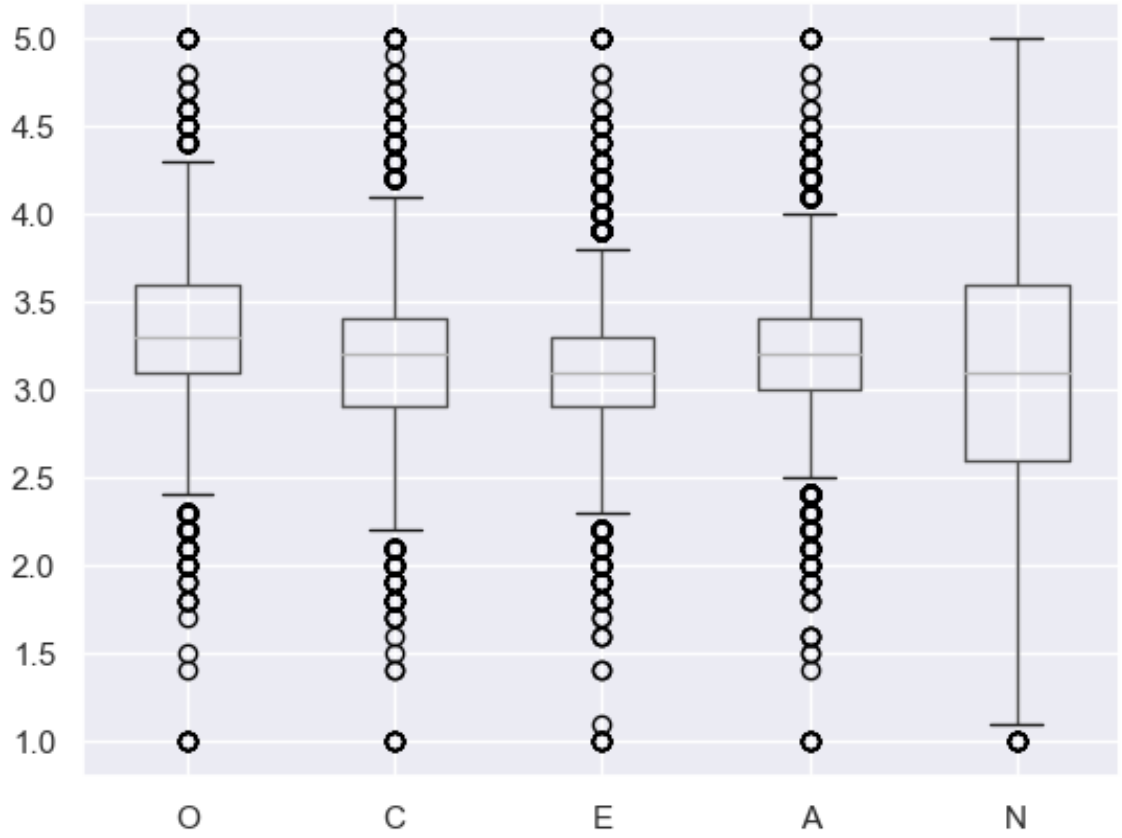
These outliers can affect some clustering algorithms while others fare with them better. As such, the plan for this report is to try each cluster both with and without outliers.

Finally, the results were visualised with a KDE graph, figure 3.5, to see the distribution in a different way. From this, one can see that the features are distributed normally with N once again having a wider shape than any of the others. The normal distribution means that algorithms that assume normalised data such as GMM should work fine without any extra pre-processing.

Dimensionality and Outlier Handling

From there, the next step was to apply PCA - Principal Component Analysis - to the dataset to reduce the dimensionality even further. PCA is a method by which dimensionality is reduced to a specified number, trying to retain as much of the original information as possible. (Whitfield and Pierre (2023)) This runs the risk of losing some accuracy but clustering is

Figure 3.4: Boxplot



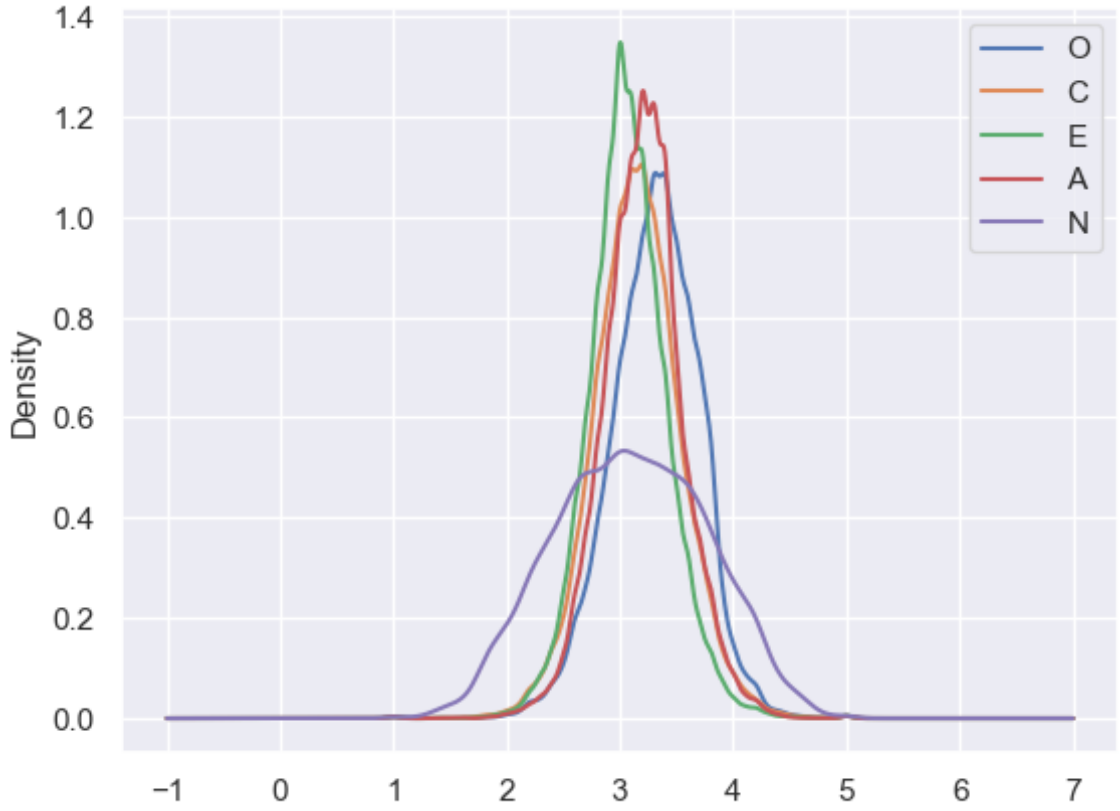
easiest with only two features and some algorithms do worse with higher dimensionality. As such, the five columns were reduced into two.

This was done twice to create two datasets. The first was just left as it was, the second was altered beforehand to remove outliers. To do this, the z scores of each row was calculated relative to the column mean and standard deviation and, from there, any that fell outside of the range -3 to 3 were removed as shown in the algorithm 1. A z-score shows where a point lies in data distribution, hence why certain values of a z-score indicate that a point might be an outlier. (Nevil, Scott (2023)) From this, one can compare clustering algorithms to both datasets – one with outliers and one without to see where changes would lie between them.

3.1.3 Clustering

With this out of the way, the next step was the clustering itself, starting with k-means, the simplest method. To calculate the number of clusters to use, the elbow method was used. This gives a general indication of how many clusters to use by looking for where the 'elbow' of the curve is, either by calculating the distortion – the average of the squared distances from the cluster centers of the respective clusters – or the inertia – the sum of squared distances of samples to their closest cluster center. (GeeksforGeeks (2023)) For this project, inertia was used as shown in algorithm 2.

Figure 3.5: KDE Graph



Algorithm 1 Removing outliers

Input: $df = dataframe$

1: $s \leftarrow df$ where $df.zscore$ ¡ 3 ▷ Create dataframe s containing only values within a z score of 3
2: df.dropNaN ▷ Drop NaN values
3: $pca \leftarrow PCA$ where $PCA(2)$ ▷ Create PCA with a dimensionality of 2
4: $principalComponents \leftarrow PCA$.fit(s) ▷ Fit PCA to s to get principal components
5: $principalDf \leftarrow$ dataframe.$principalComponents$ ▷ Convert components into a dataframe

Sharma, Aditya (2020) Kumar, Akash (2023) Zach (2020)

Algorithm 2 Calculating inertias for elbow method

Input: $df = dataframe$

1: $data1 \leftarrow list(zip(df))$
2: $inertias \leftarrow \emptyset$
3: **for** $i \leftarrow 1$ to 11 **do**
4: $kmeans \leftarrow KMeans(i)$ ▷ Apply k-means method to each number of clusters
5: $kmeans \leftarrow kmeans.fit(data1)$ ▷ Fit data to k-means method
6: $inertias$.insert($kmeans.inertia$) ▷ Calculate inertia and add it to list of inertias
7: **end for**
8: plot($inertias$)

W3Schools (2023)

As can be seen by the graph, figure 3.6, the elbow is about 2.

Since the elbow method is relatively objective, two clusters and four clusters were both tested since those are the main two places where the 'elbow' of the curve could be argued to be at, comparing silhouette scores for each. For good measure, three clusters were also tested since that number lies in the middle of the two possible values. The elbow method helps narrow down all the options that the optimal number of clusters could be so one doesn't have to test too many of them.

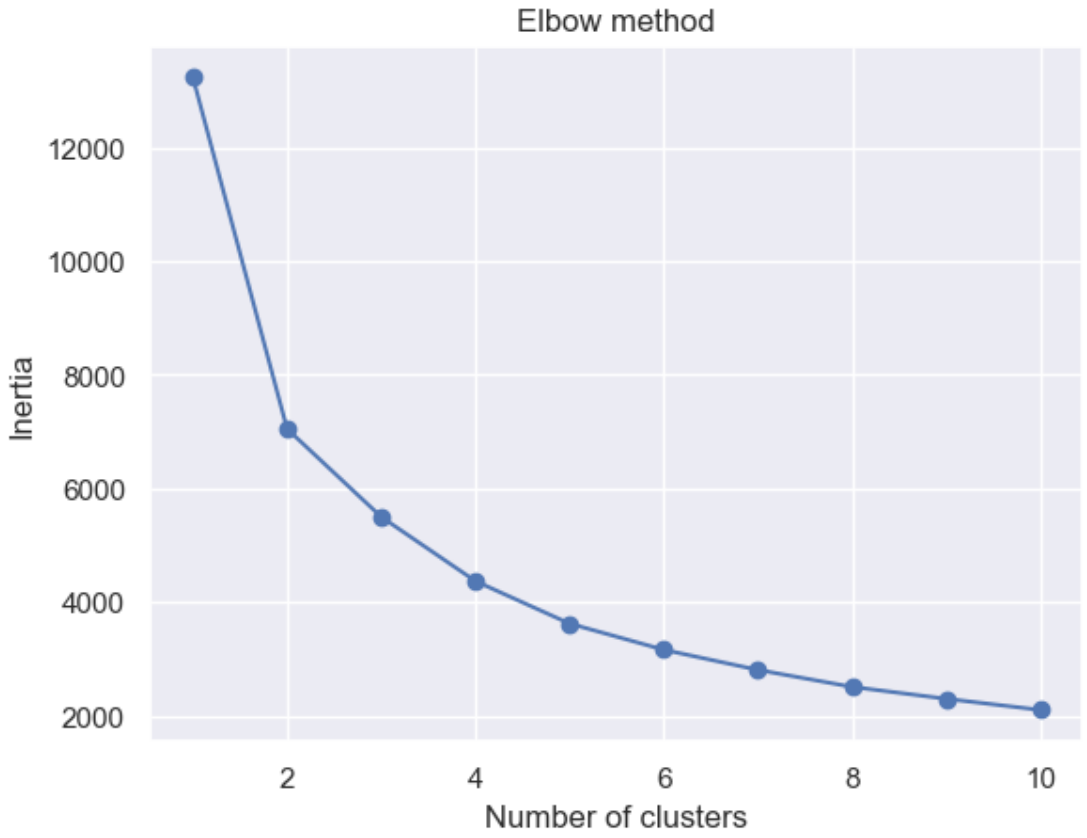
To calculate K-Means, the python library sklearn.cluster was used which contained methods for K-Means already to fit the data to. Then, using pyplot, the outcome was visualised.

This was repeated twice - both on the dataset with outliers and the one without to see the differences. As such, the elbow method needed to be repeated again on the dataset without outliers.

As seen in figure 3.7, the elbow method again came to be the same shape as before with 2 or 4 being the 'elbow' of the graph and, therefore, the number of clusters that were to be tested.

The second clustering algorithm of choice was DBSCAN. Once again, the sklearn.cluster library was used to find the methods needed to apply the clustering.

Figure 3.6: Elbow Method for Outliers



DBSCAN doesn't take an exact number of clusters but instead takes two variables to calculate – Epsilon and Min Points. Epsilon is the threshold that two points can be considered neighbours and, hence, part of the same cluster, while Min Points is the minimum number of points required per cluster. (Maklin, Cory (2019*a*)) Using two 'For' loops as shown in algorithm 3, different values of Epsilon were tested against different values of Min Points until the optimal silhouette value was produced. The variables that achieved that silhouette value were then outputted.

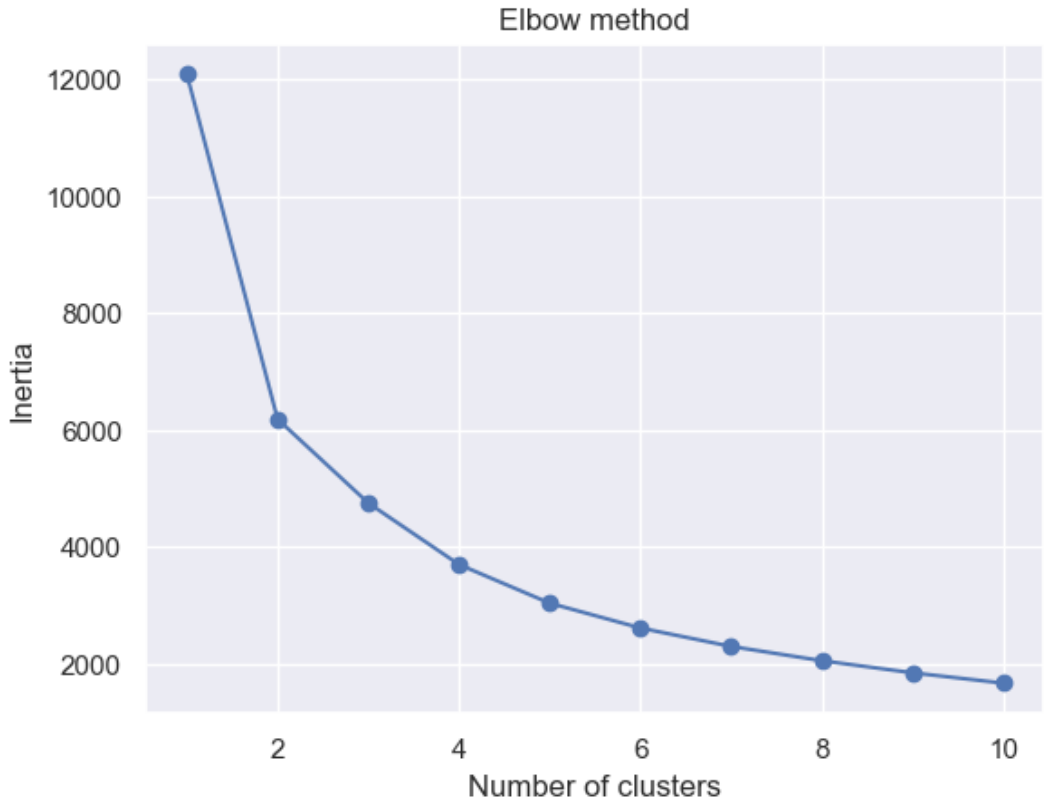
The best silhouette score produced had an Epsilon of 0.12 and a Min Points of 12. This would produce two clusters. (Mane, Tanmay (2021) Toushik, Azmine Wasi (2022))

Next was to see how it fared when there were no outliers. This produced an Epsilon of 0.12 again but a Min Points of 11, just one less than before.

The third model was Gaussian mixture modelling. The sklearn library also had functions for this.

To determine the best number of clusters this time, each number of clusters was tested and compared them to their silhouette score to see what the best achieved was. Based on previous results from literature and my own previous clusters, 2, 3, and 4 clusters were tried.

Figure 3.7: Elbow Method for No Outliers



The fourth and final model was Hierarchical Clustering. Specifically, an Agglomerative approach from the sklearn library.

To calculate the number of clusters required, the method is to create a dendrogram of results and to draw a line where it cuts the tallest vertical line. The number of lines intersected by the new line is the number of clusters that should be used. (Sharma, Pulkit (2022))

```
1 pyplot.figure(figsize=(7, 4))
2 pyplot.title("Dendrograms")
3 dend = shc.dendrogram(shc.linkage(principalDf, method='ward')) #Plot
    dengrgrams from dataset using ward method
4 pyplot.axhline(y=80, color='r', linestyle='--')
```

Listing 3.1: Code snippet of plotting Dendrogram

Sharma, Pulkit (2022)

As can be seen in figure 3.8, the optimal number of clusters was again two.

Next was the dataset without outliers, 3.9. Like usual, it was two clusters.

With this done, the next step was do some extra examination of the clusters to see what different kinds of personalities had been defined. Since two clusters would simply produce two

Algorithm 3 Calculating optimal epsilon and min samples for DBSCAN

Input: $minsamples = range(10, 21)$
Input: $eps = range(0.05, 0.13)$
Input: $X = dataframe$

1: $output \leftarrow \emptyset$
2: **for** $ms \leftarrow minsamples$ **do**
3: **for** $ep \leftarrow eps$ **do**
4: $labels \leftarrow DBSCAN.fit(minsamples, eps, X).clusters$ ▷ get the clusters by fitting the data to DBSCAN along with the current min sample and epsilon
5: $score \leftarrow silhouette_score(X, labels)$ ▷ Calculate silhouette score
6: $output$.insert$(ms, ep, score)$ ▷ Add the epsilon, min sample, and the calculated silhouette score to the output list
7: **end for**
8: **end for**
9: $output$.sort ▷ Sort output to get the highest silhouette score and its respective values first

mirrors of each other, three clusters were used for this. DBSCAN didn't allow more than two clusters to be used so the K-Means results were chosen as the next best scoring algorithm in terms of silhouette score.

The K-Means, outliers, 3 cluster solution was used to find the points from each of the three clusters. To do this, the initial dataset was concatinated with the PCA dataset results. This allowed me to see both the final results and what those results had initially been before the dimensions had been reduced. The outlier dataset had to be used since any removed outliers would mean the two datasets wouldn't line up as equivalent to one another.(Data To Fish (2021), Pandas Pydata (2023*a*))

From this, each feature was plotted in terms of its cluster to see what features each cluster had. (Pandas Pydata (2023*b*), Seaborn Pydata (2023))

Since K-Means had a preference for four clusters, I also reattempted the experiment with four instead of three.

3.2 Summary

In summary, this chapter, the code and graphs were created to clean the dataset, cluster it according to the four algorithms, and produce the silhouette score from each for scoring, repeating results for both with and without outliers in the dataset to see how well they were handled. Clustering was repeated with 2-5 clusters for each algrorithm except DBSCAN which doesn't easily support choosing the number of clusters. As well as this, the clusters for K-Means without outliers with three clusters had some extra exploration into the values in each cluster to see how they were grouped.

Figure 3.8: Dengrogram for Outliers

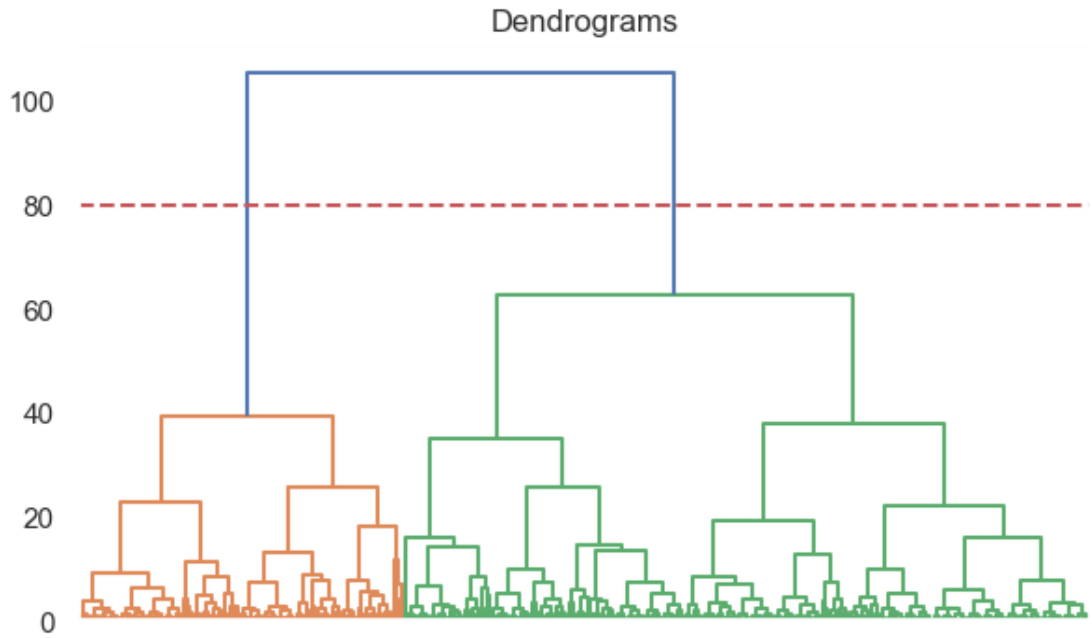
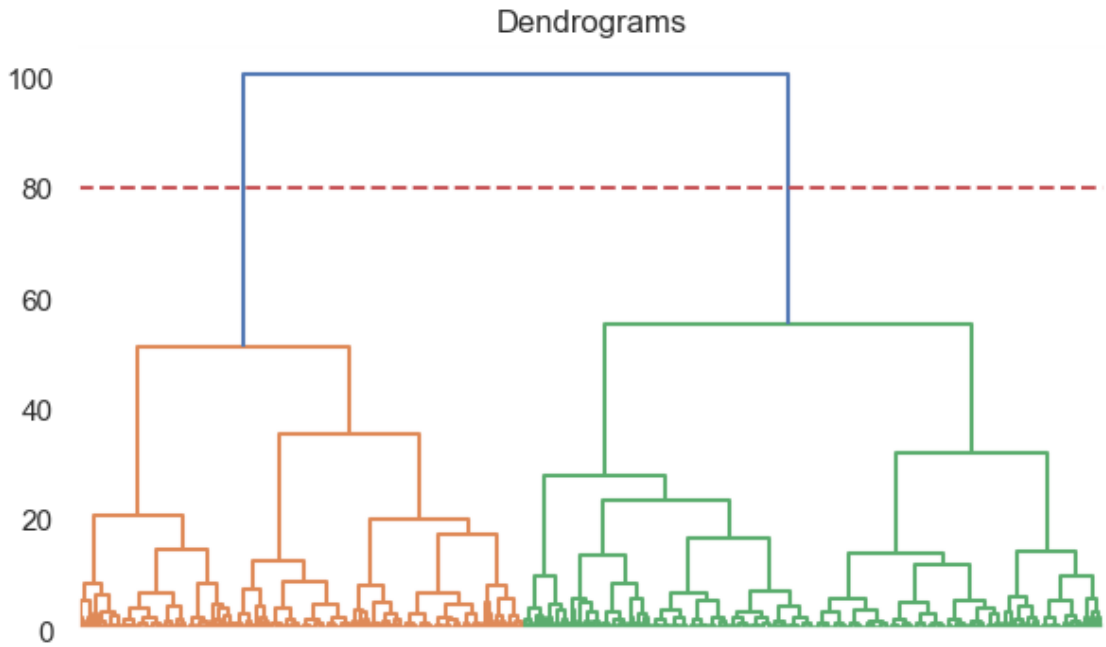


Figure 3.9: Dengrogram for No Outliers



Chapter 4

Results

4.1 Results

The results looked like tables 4.1, 4.2, and 4.4 for each dataset, ordered from best to worst.

Table 4.1: Results With Outliers Dataset

Clustering Algorithms With Outliers		
Algorithm	Cluster No.	Silhouette Score
DBSCAN	2	0.521
K-Means	2	0.407
GMM	2	0.396
Hierarchical	2	0.382
K-Means	5	0.325
K-Means	4	0.321
K-Means	3	0.320
GMM	3	0.299
Hierarchical	3	0.290
GMM	5	0.265
GMM	4	0.258
Hierarchical	4	0.255
Hierarchical	5	0.240

It can also be seen that 2 clusters was, for all of them, the optimal number. For a lot of reports on the subject of clustering personality types, however, the goal is often to see if people fall into a certain number of categories. This means most other papers use 3 or 4 clusters. This would make DBSCAN an inefficient model since you can't determine the number of clusters beforehand. As such, the results when 2 clusters are removed are as seen in table 4.4.

In that case, K-Means can be seen to be the best algorithm for the job, the optimal number of clusters becoming 4. With different numbers of clusters, K-Means tends to be the algorithm that does best as well as allowing a programmer to determine how many clusters they want themselves. Both other clustering algorithms did best with three clusters instead.

Table 4.2: Results Without Outliers Dataset

Clustering Algorithms Without Outliers		
Algorithm	Cluster No.	Silhouette Score
DBSCAN	2	0.459
K-Means	2	0.413
GMM	2	0.411
Hierarchical	2	0.349
K-Means	4	0.332
K-Means	5	0.327
K-Means	3	0.321
GMM	3	0.316
GMM	4	0.309
Hierarchical	4	0.286
GMM	5	0.284
Hierarchical	3	0.266
Hierarchical	5	0.257

Table 4.3: Overall Best Algorithms

Clustering Algorithms Overall			
Algorithm	Cluster No.	Outliers	Silhouette Score
DBSCAN	2	With	0.521
K-Means	2	Without	0.413
GMM	2	Without	0.411
Hierarchical	2	With	0.382

Table 4.4: Best Algorithms with Over 2 Clusters

Clustering Algorithms Above 2 Clusters			
Algorithm	Optimal Cluster No.	With or Without Outliers	Silhouette Score
K-Means	4	Without	0.332
GMM	3	Without	0.316
Hierarchical	3	With	0.266

Looking into the clusters themselves to find out what these different clusters mean. Since most of the algorithms preferred 3 clusters, the 3 cluster solution was used. In terms of algorithm, K-Means was chosen since it did best and the dataset with outliers. From this, it can be seen that the three main personality types found was a cluster where all features were higher than the other clusters, one where all the features were lower than the other clusters, and one that simply sat in the middle. Attempting the same on 4 clusters since that was K-Means preferred number found similar results with two clusters sitting on opposite extremes. However, out of the other two clusters formed, one got generally higher values with O, C, E, and A compared to the other but scored lower in terms of N. The vice-versa applied to the other.

Considering someone with high openness, conscientiousness, extroversion, and agreeableness as a highly sociable person who probably gets along well with people, this could imply that the four main personality types are: highly sociable with high neuroticism, not very social but low neuroticism, highly sociable with low neuroticism, and not very sociable but with high neuroticism.

4.2 Summary

In conclusion, it was found from this report that DBSCAN is the best clustering algorithm for two clusters but, if any variation in the number of clusters is needed, the best algorithm is K-Means with outliers removed and four clusters. Looking deeper into the different clusters formed found that O, C, E, and A values seem to often be linked together (High O values tend to come with high C, E, and A values while a low value makes every other feature also tend to be low) while N behaves individually leading to the four clusters being: High O, C, E, and A values with low N, low O, C, E, and A values, high N, all low values, and all high values.

Chapter 5

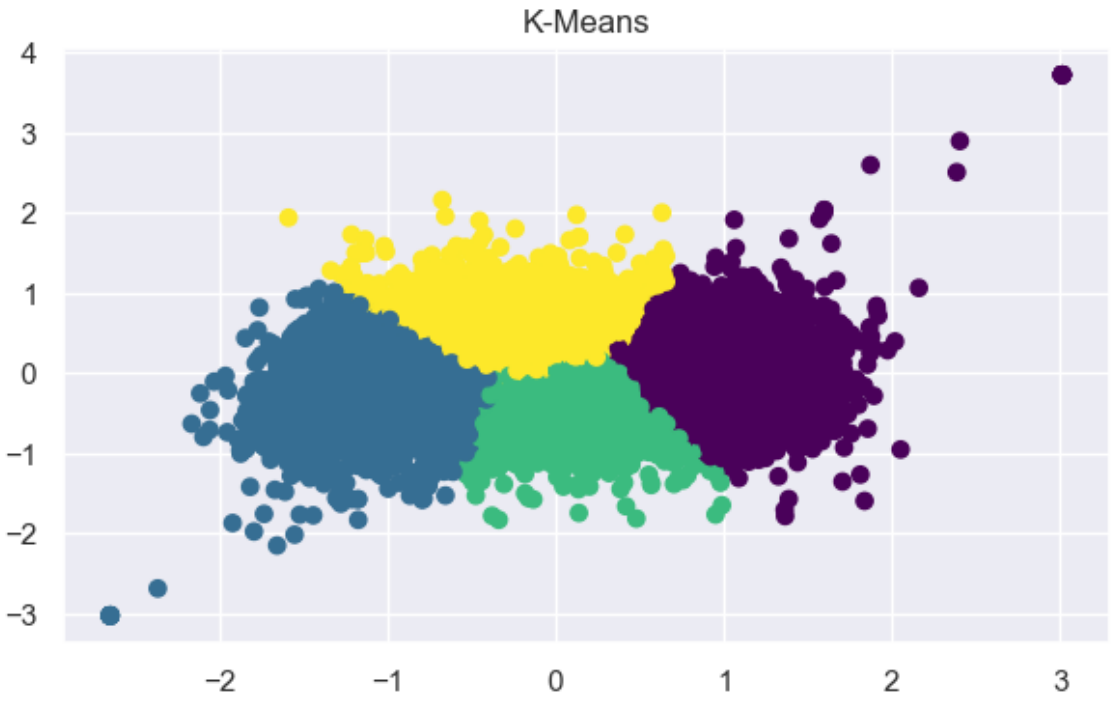
Discussion and Analysis

5.1 Clustering Results

For K-Means, the elbow method showed that the optimal number of clusters lay at about 2 but, due to how objective the elbow method can be, clusters was also tried since it could be argued to be the elbow and 3 was also tested since it was in between these values.

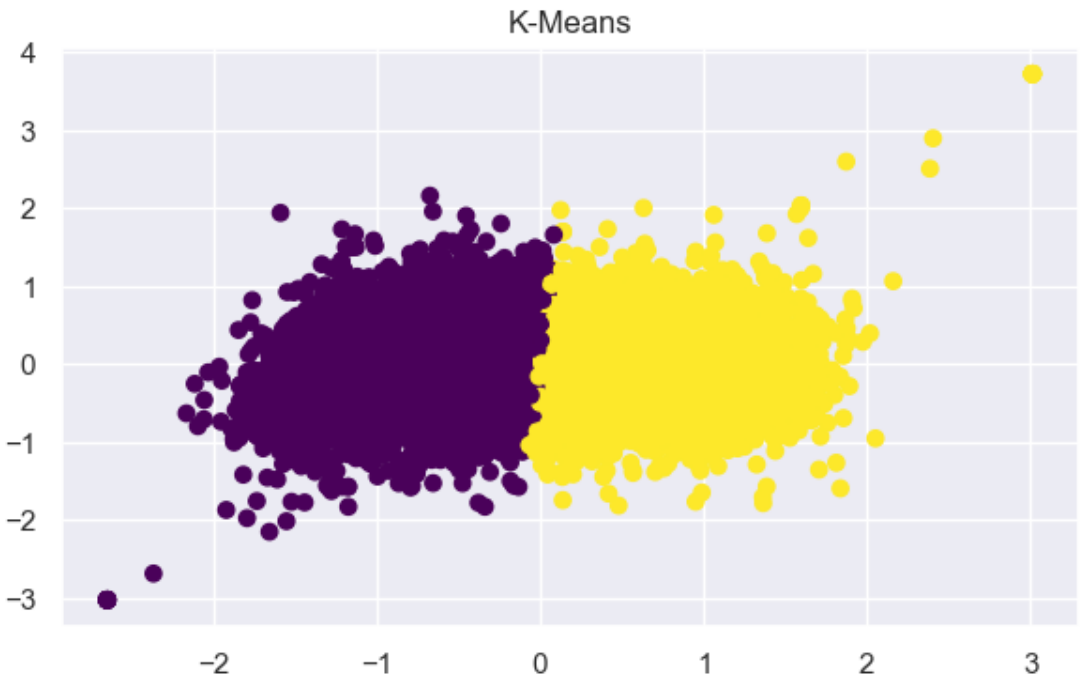
First, four clusters were tried on the dataset with outliers. (Figure 5.1) For the four cluster with outliers, the silhouette score was 0.321. (Kumar, Ajitesh (2020))

Figure 5.1: K-Means Outliers 4 Clusters



Then it was the outlier dataset with two clusters (Figure 5.2) that was tested which came out as a silhouette score of 0.407. As the silhouette score for two clusters is higher, this means

Figure 5.2: K-Means Outliers 2 Clusters



that two clusters was the optimal value.

For good measure, three clusters were also tested. However, the silhouette score came out as 0.320, proving that two clusters were still the optimal number. (figure 5.3)

The next step was to test the same dataset without outliers. The elbow method again came to be the same shape as before so the same number of clusters was tried. Without outliers and with four clusters, the silhouette score came out as 0.332 (Figure 5.4) which was an improvement on the dataset with outliers.

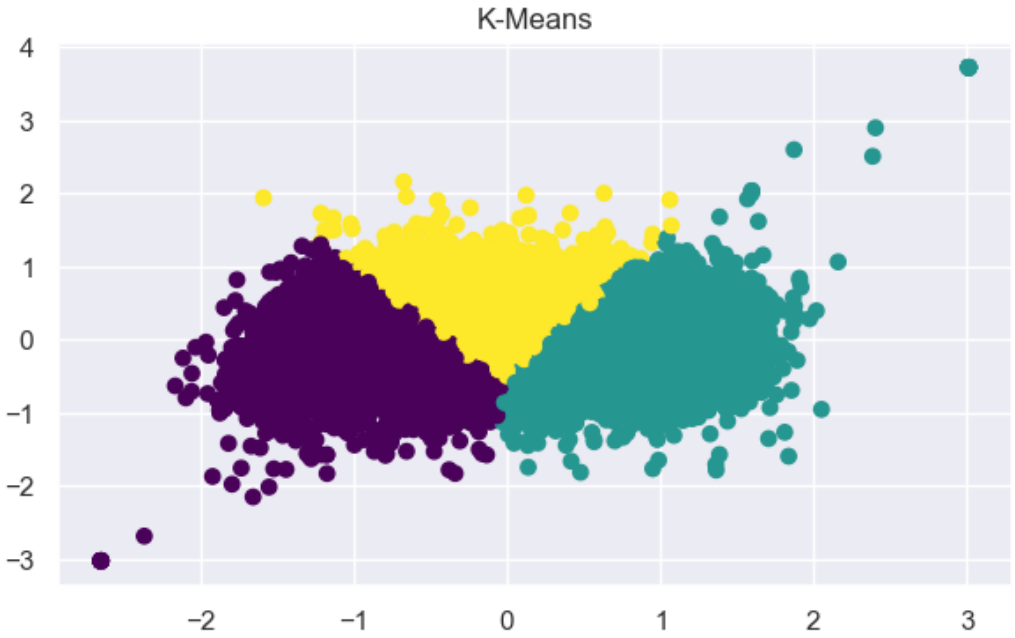
Three clusters produced 0.321, (figure 5.5) once again following the same pattern from the outliers dataset, not being as good as the other possible cluster values. Two clusters similarly showed improvement, coming out as 0.413. (figure 5.6) However, while definitely an improvement, the improvement wasn't by much.

The second clustering algorithm of choice was DBSCAN. The silhouette score produced was 0.5214. (figure 5.7) This produced two clusters and, as can be seen, this is already an improvement of the best value K-Means could produce.

Next was to see how it fared when there were no outliers. The silhouette score produced this time was 0.4591. (figure 5.8)

Unlike K-Means which improved with outliers removed, DBSCAN performs worse. DBSCAN is a model that can already handle outliers after all which could help explain why it did worse.

Figure 5.3: K-Means Outliers 3 Clusters



The third model was Gaussian mixture modelling. Like usual, the first test was on the dataset with outliers. (VanderPlas, Jake (2016) Toushik, Azmine Wasi (2022)) The results were 0.257589 for four clusters (figure 5.11), 0.298867 for three (figure 5.10, and 0.395608 for two (figure 5.9) making two clusters best.

Next, the same was performed on the dataset without outliers. The results were 0.309124 for four (Figure 5.14), 0.316148 for three (Figure 5.13), and 0.410804 for two (Figure 5.12) making two clusters the best again. The dataset without outliers showed a decent improvement in silhouette score.

The fourth and final model was Hierarchical Clustering. Like all the past algorithms, two clusters came out as the optimal number. There was a silhouette score of 0.382 for two clusters. (figure 5.15)

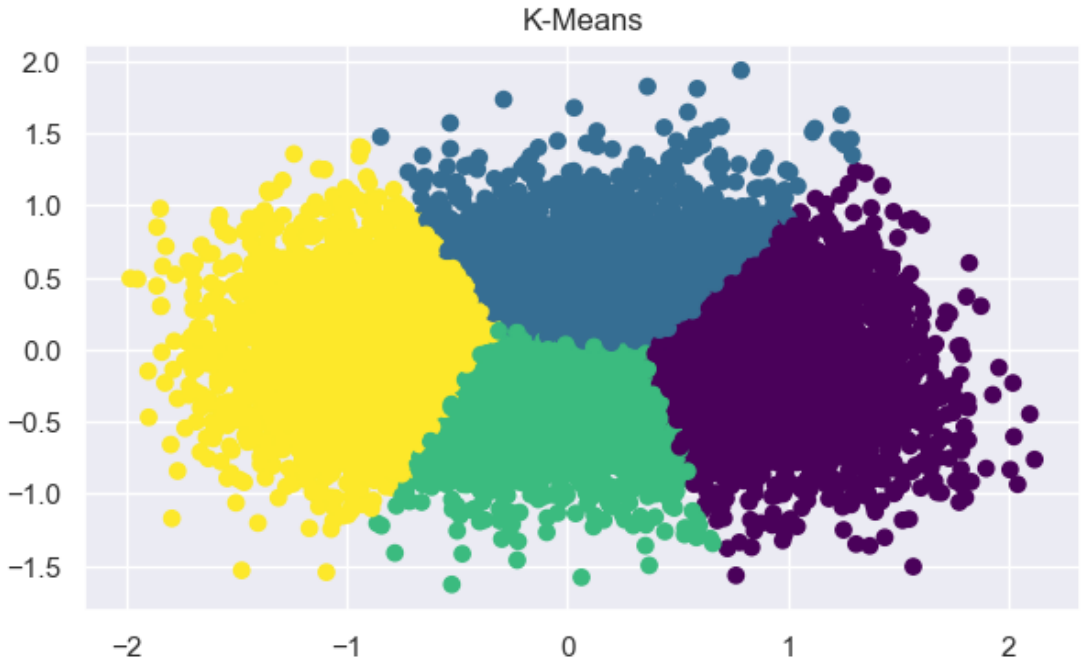
Next was the dataset without outliers. (Figure 5.16) A silhouette score of 0.349 was produced, showing that there is no improvement from removing outliers. Instead, there was only a small decrease.

With this done, the next step was do some extra examination of the clusters to see what different kinds of personalities had been defined. K-Means with no outliers and three clusters was used for this. (figure 5.3)

From this, each feature was plotted in terms of its cluster to see what features each cluster had. The results can be seen in figure 5.17 to 5.21

In the context of results, the features are scored from 1-5 as a measure of how well each

Figure 5.4: K-Means No Outliers 4 Clusters



feature fits each person in the dataset. 1 means the feature doesn't fit the person very well while 5 means it fits them a lot. From what can be seen, cluster 0 has the largest spread and tends to contain the lowest values of each feature while cluster 2 had the highest values for every feature. Cluster 2 was mostly medium.

The experiment was also reattempted with four instead of three clusters. Again, O, C, E, and A were directly correlated between algorithms, however the N values varied for each cluster.

Only the O and N graph is displayed in figures 5.22 and 5.23 but C,E, and A had very similar results to O, hence why they weren't included.

Since time was available afterwards for more work on the project, several of the algorithms were redone again with more clusters to replicate the experiments in past literature. In past papers, 2 clusters were considered invalid since it would make both clusters simply a pair of opposite personality types rather than determining a series of different personality types like what would be needed for any sort of real analysis.

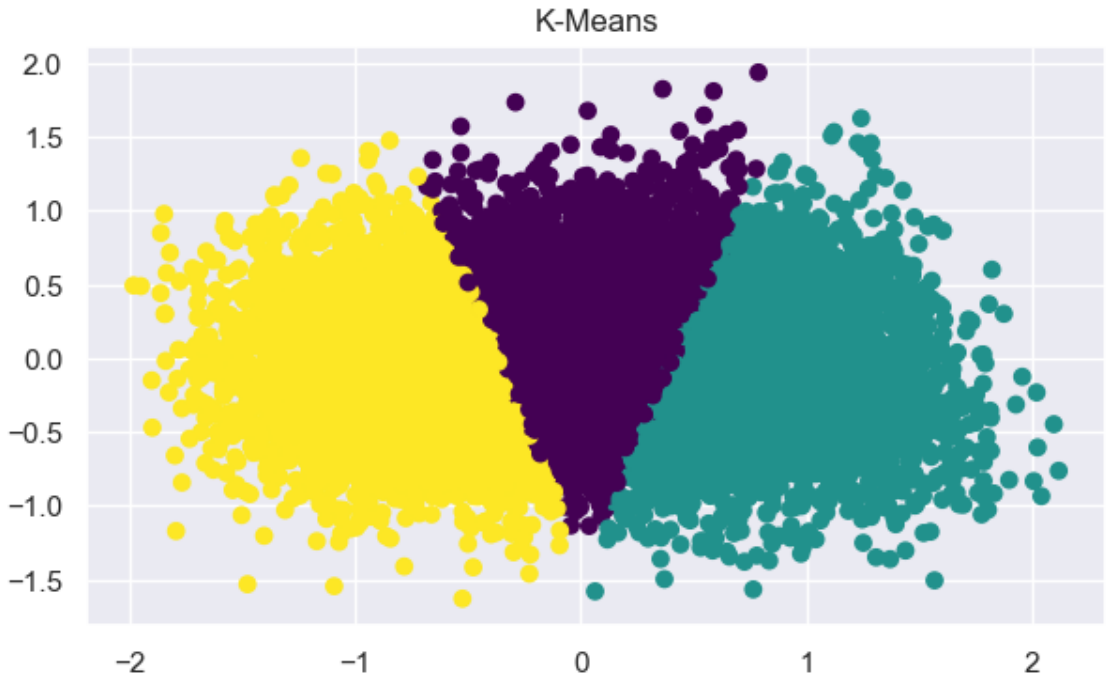
For this, every algorithm was done with clusters from 3-5.

GMM with 5 clusters (Figure 5.25) made a silhouette score of 0.265 with outliers and 0.284 without. (Figure 5.24)

Hierarchical with outliers produced a silhouette score of 0.266 with three clusters (Figure 5.26), 0.255 with four (Figure 5.27), and 0.240 with five. (Figure 5.28)

Without outliers, the silhouette score was 0.266 for three (Figure 5.29), 0.286 for four

Figure 5.5: K-Means No Outliers 3 Clusters



(Figure 5.30), and 0.257 for five. (Figure 5.31)

K-Means with five clusters with outliers produced 0.325 (Figure 5.32) while the dataset without outliers produced 0.327. (Figure 5.33)

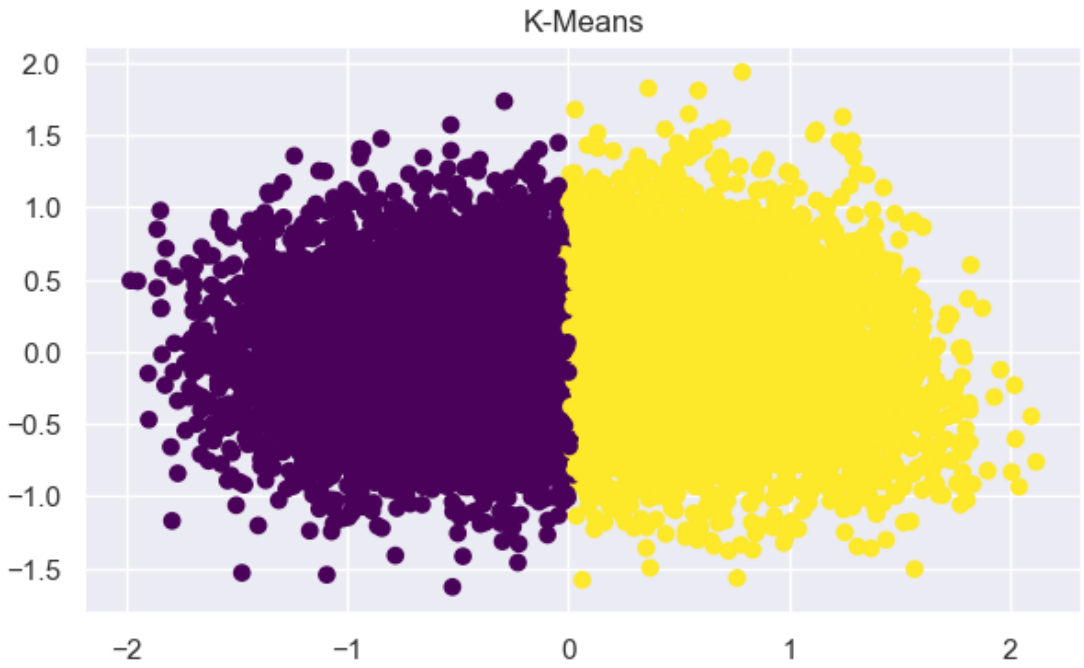
DBSCAN, unfortunately, doesn't pick and choose cluster numbers as easily as the other algorithms as shown in figures 5.34 and 5.35. While a new algorithm was produced to only select cluster numbers above 2, it initially produced overly high numbers of clusters, above 100. A limitation was added to allow only cluster numbers beneath 10 and, although it produced three clusters like wanted, it's very visually clear that the new cluster is too small to really be considered a proper cluster.

If the goal is to find different personality types, the third clusters in both datasets are obviously not large enough to really be considered common and would certainly make a very specific personality type. DBSCAN is obviously not an algorithm made for picking and choosing the number of clusters needed. As such, any further numbers of clusters for DBSCAN were discarded as data.

5.2 Significance of the findings

The results imply that K-Means is the best potential algorithm to use to finding and analyzing different categorizations of personality types. Meanwhile, hierarchical performed the worst, implying that the paper which chose hierarchical clustering for OCEAN clustering chose poorly.

Figure 5.6: K-Means No Outliers 2 Clusters



Out of all the algorithms, K-Means and GMM did best with the outliers removed while DBSCAN and Hierarchical did best when the outliers were left in.

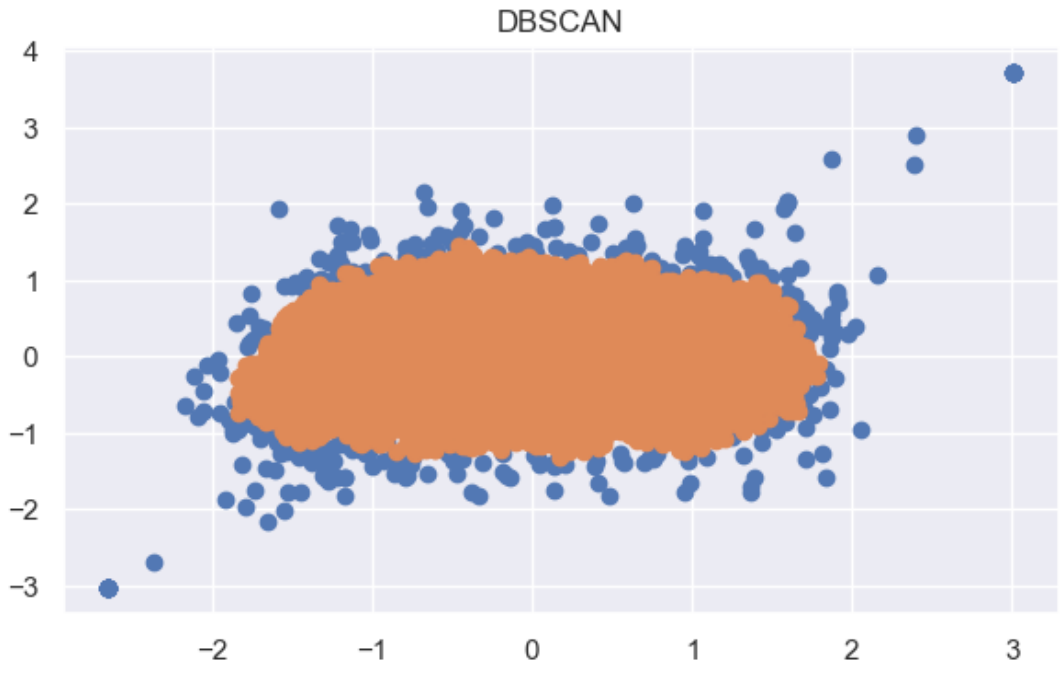
However, due to some of the algorithms results changing with subsequent tests due to the way they work, these scores can change. For example, since K-Means initiates by randomly picking points to form clusters around, the result is affected by that – and therefore makes it less reliable. After all, the program needs to be run multiple times to find the best result and, even then, it's up to chance. On account of this, K-Means may have come out as the best scoring result but it may be less reliable since even if the scores were better since you can't guarantee it will always be better. In that case, it's arguably not the worst decision to choose hierarchical clustering despite its lower score if one needs to be assured of the repeatability of the results. However, due to how much lower the hierarchical score is from all the other algorithms, there would possibly be better results outside of this particular pool of algorithms.

5.3 Limitations

While the findings use a small pool of algorithms and techniques, there are plenty more methods that could have been tested such as using average linkage method on the hierarchical clustering or other variations of K-Means such as K-Medoids.

It could also be worth attempting the algorithms with randomised initialisations multiple times with different initial values to see how consistently they achieve their current ranking.

Figure 5.7: DBSCAN Outliers 2 Clusters

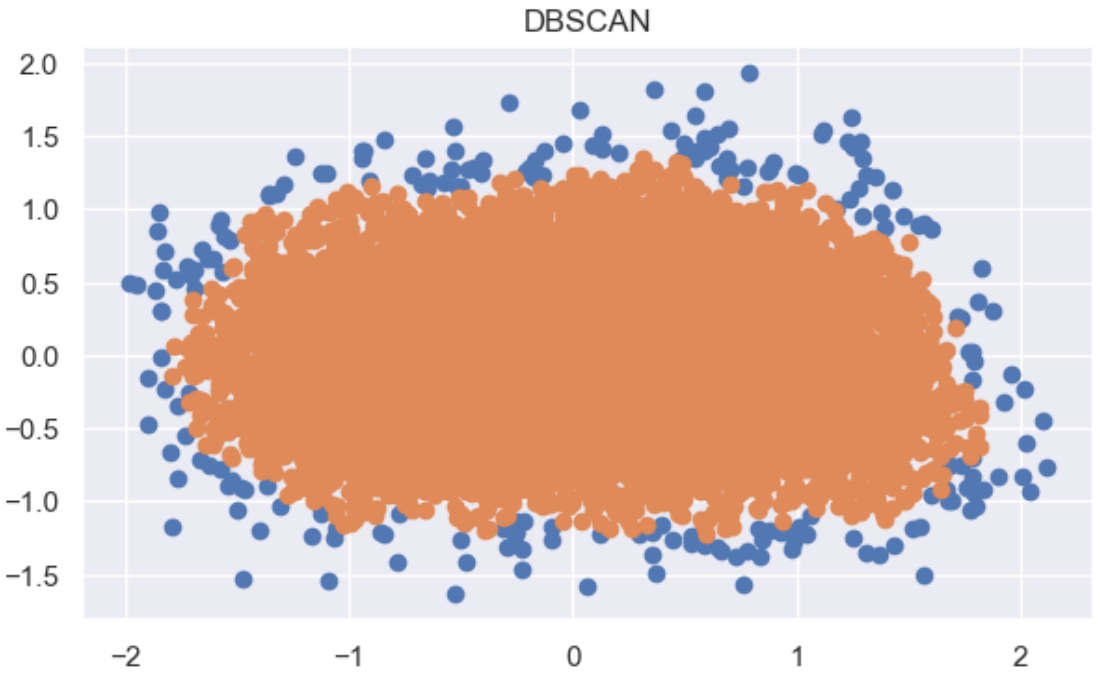


Juma, Stanley (2021) StackOverflow (2021) Mane, Tanmay (2021)

5.4 Summary

In summary, while the evidence points towards K-Means being the best, there are valid reasons to use other algorithms as well for the consistency they bring or simply because they weren't attempted in this paper.

Figure 5.8: DBSCAN No Outliers 2 Clusters



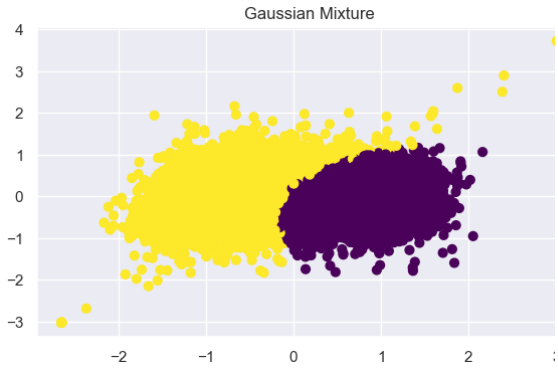


Figure 5.9: GMM Outliers 2 Clusters

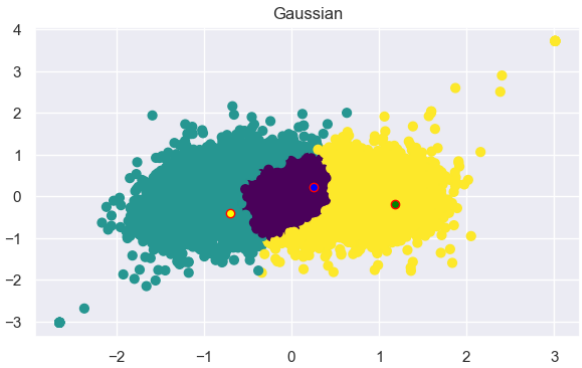


Figure 5.10: GMM Outliers 3 Clusters

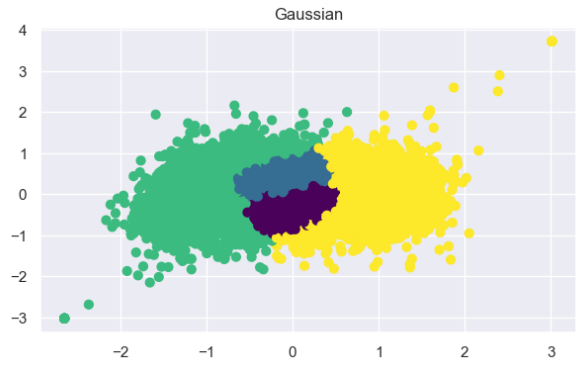


Figure 5.11: GMM Outliers 4 Clusters

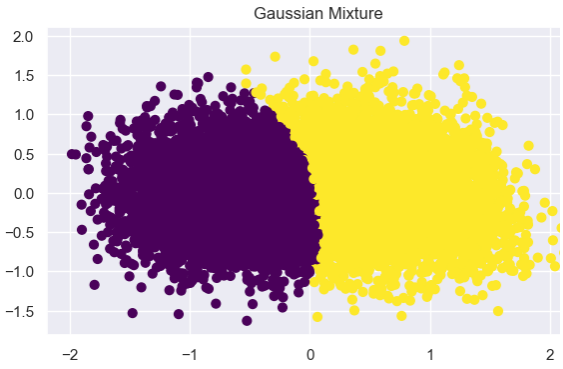


Figure 5.12: GMM No Outliers
2 Clusters

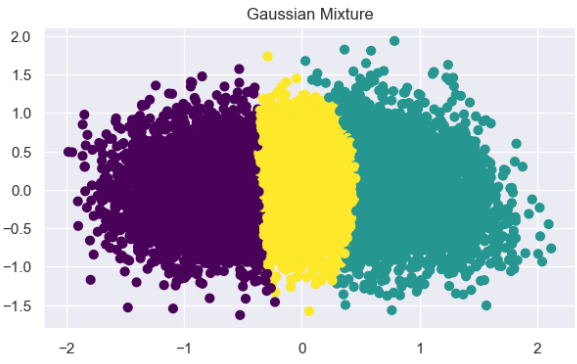


Figure 5.13: GMM No Outliers
3 Clusters

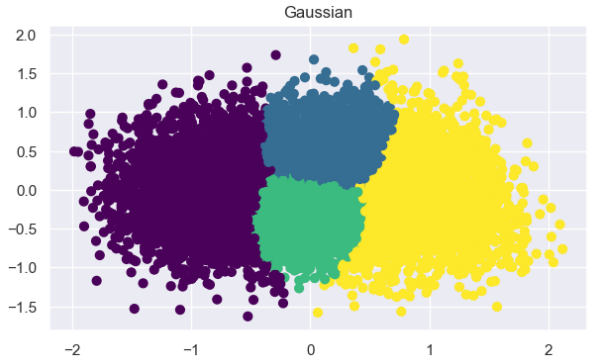


Figure 5.14: GMM No Outliers
4 Clusters

Figure 5.15: Hierarchical Outliers 2 Clusters

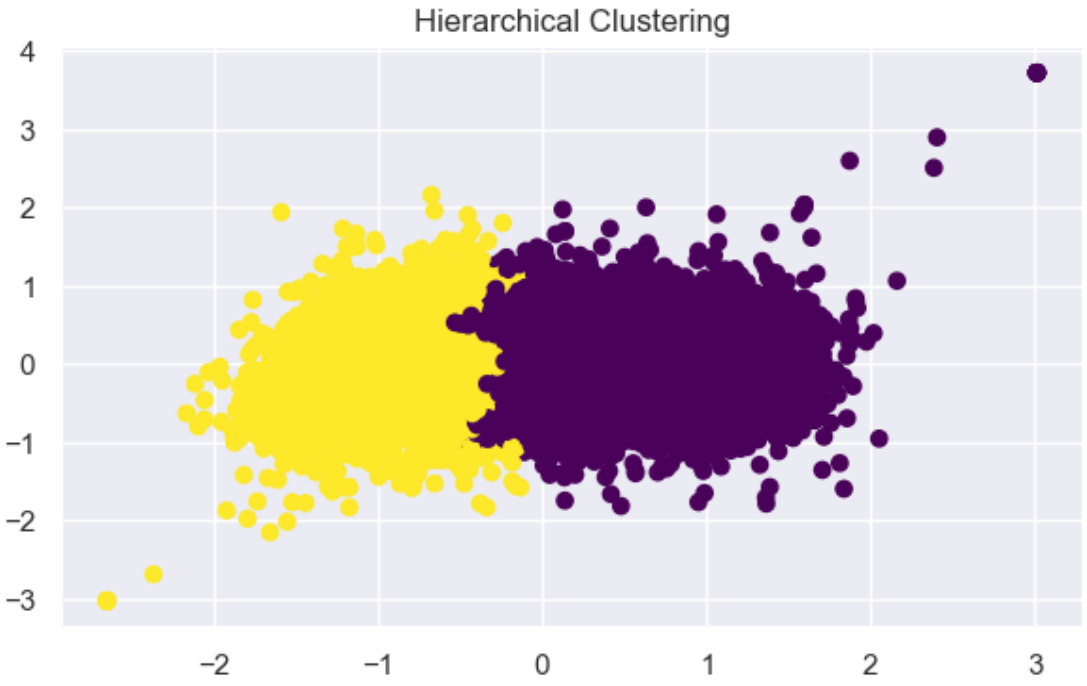
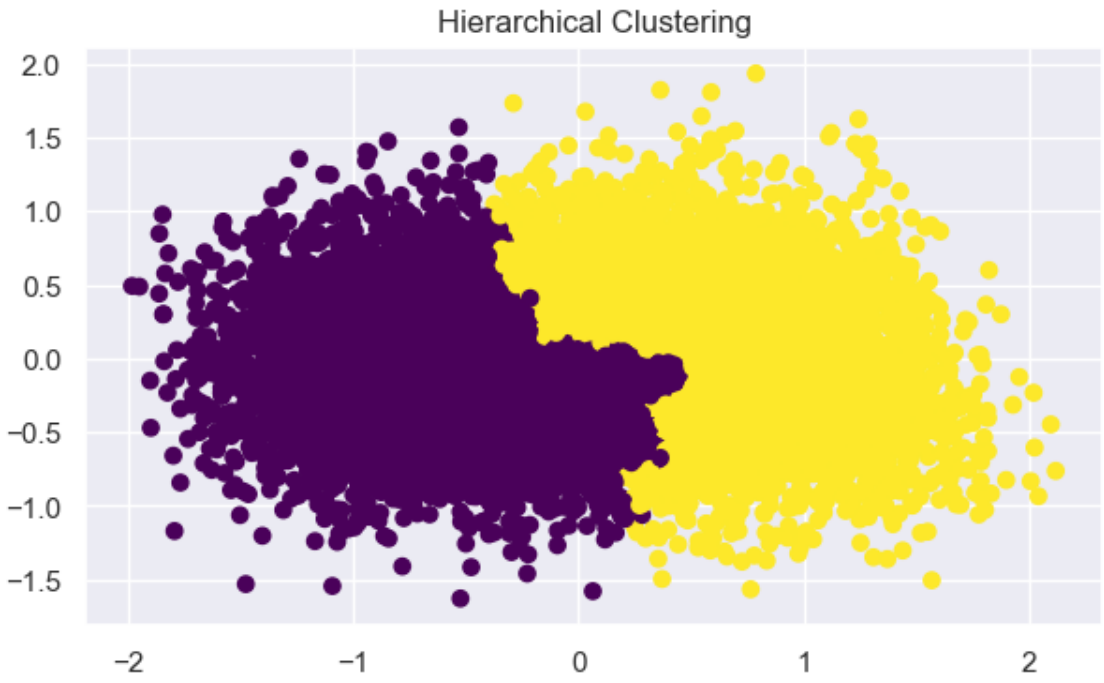


Figure 5.16: Hierarchical No Outliers 2 Clusters



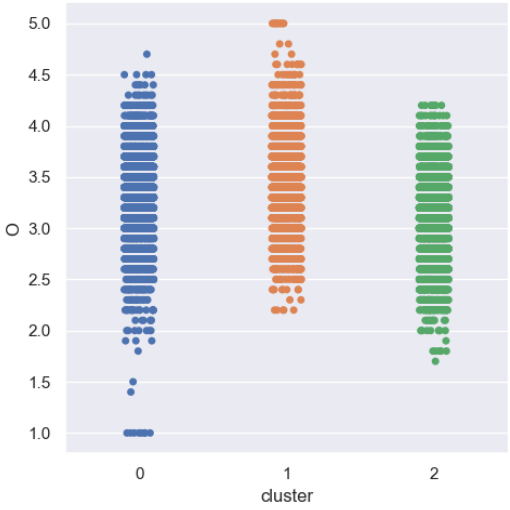


Figure 5.17: Openness Clusters

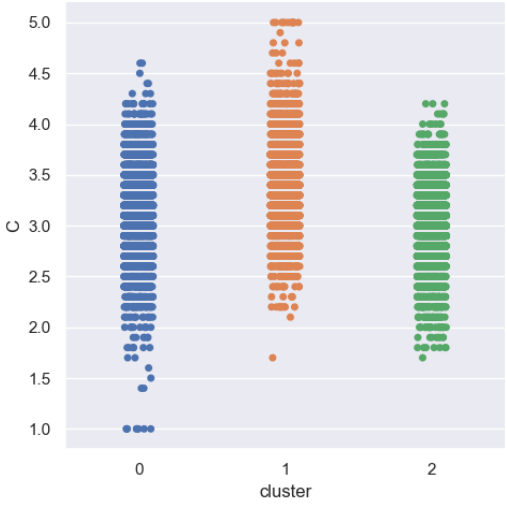


Figure 5.18: Conscientiousness Clusters

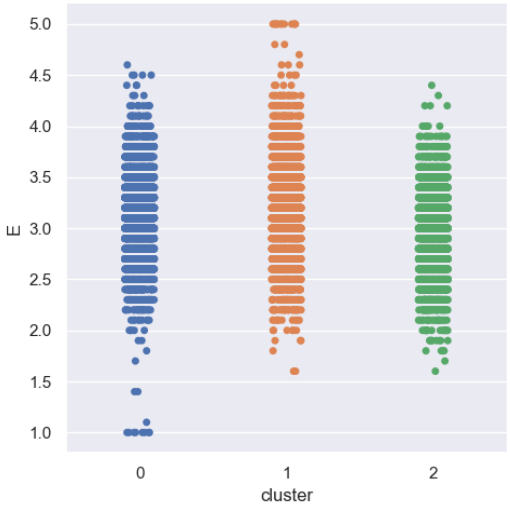


Figure 5.19: Extroversion Clusters

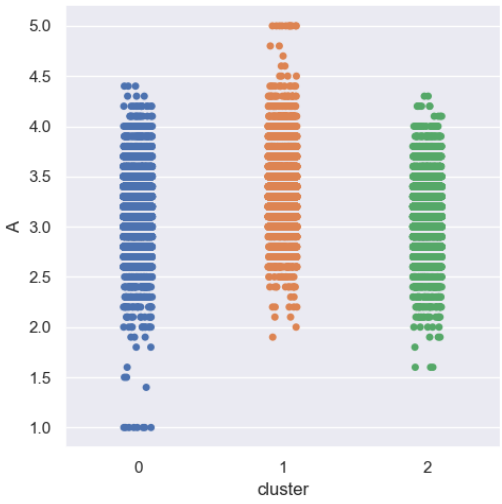


Figure 5.20: Agreeableness Clusters

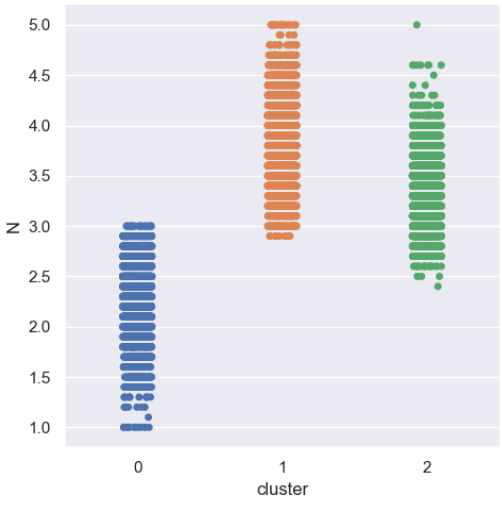


Figure 5.21: Neuroticism Clusters

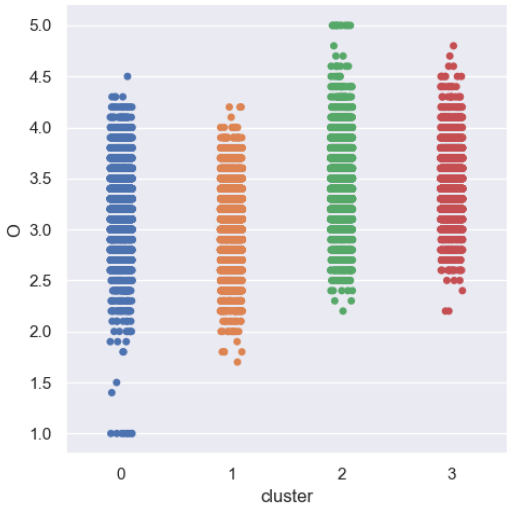


Figure 5.22: Openness Clusters

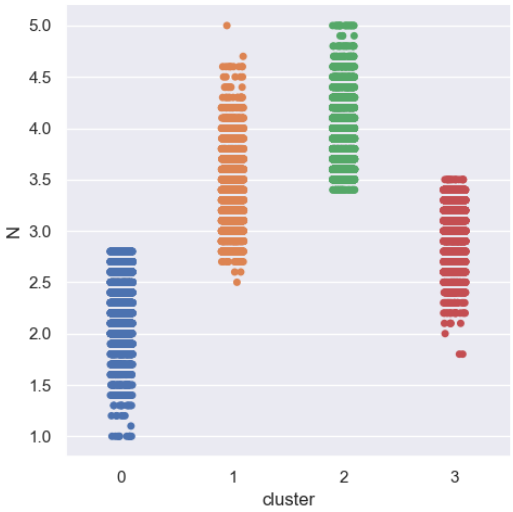


Figure 5.23: Neuroticism Clusters

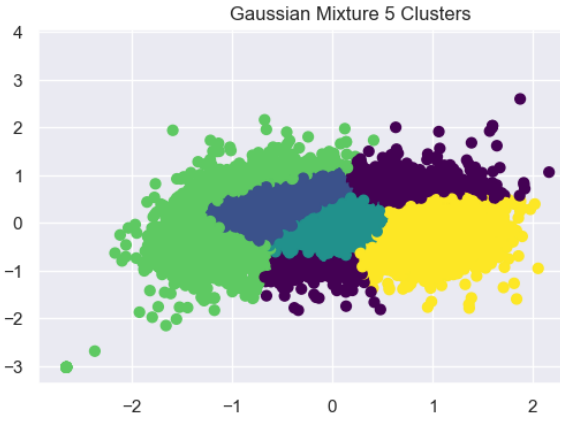


Figure 5.24: GMM Outliers 5 Clusters

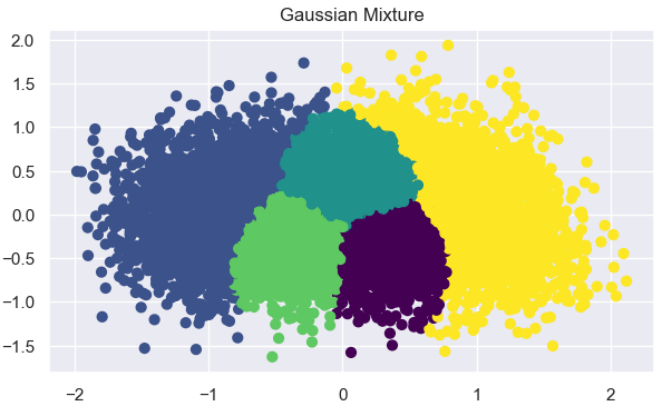


Figure 5.25: GMM No Outliers 5 Clusters

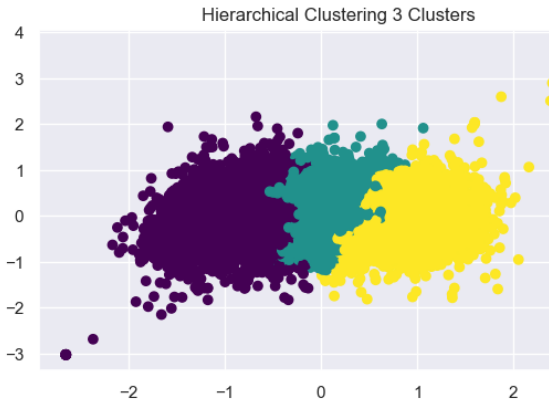


Figure 5.26: Hierarchical Outliers 3 Clusters

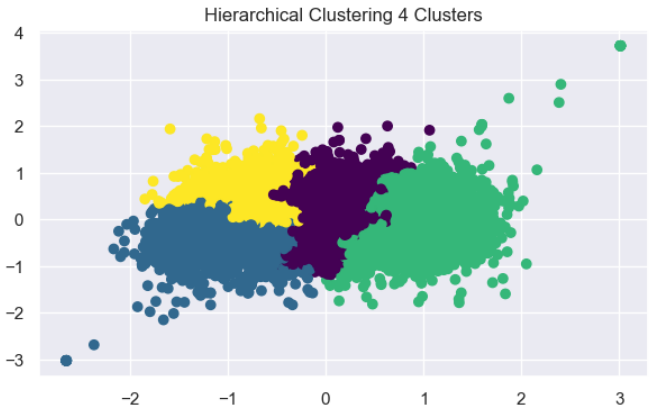


Figure 5.27: Hierarchical Outliers 4 Clusters

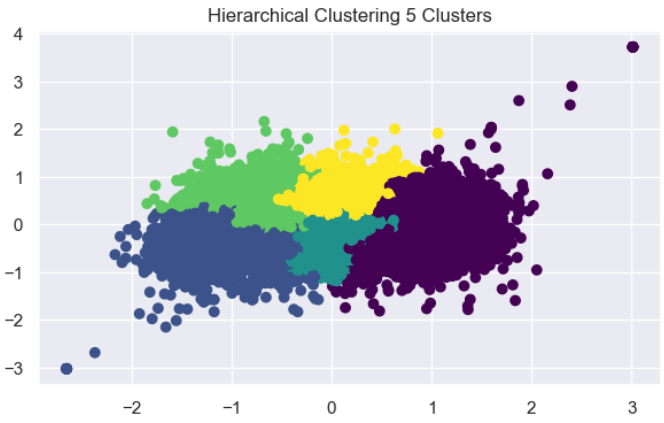


Figure 5.28: Hierarchical Outliers 5 Clusters

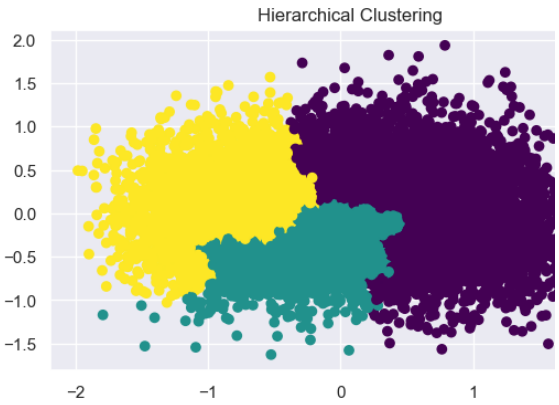


Figure 5.29: Hierarchical No Outliers 3 Clusters

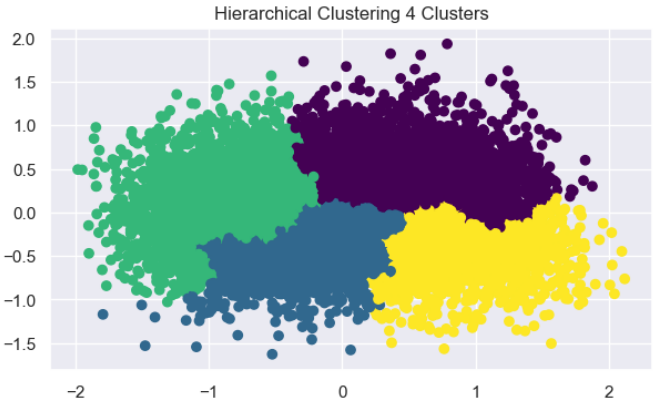


Figure 5.30: Hierarchical No Outliers 4 Clusters

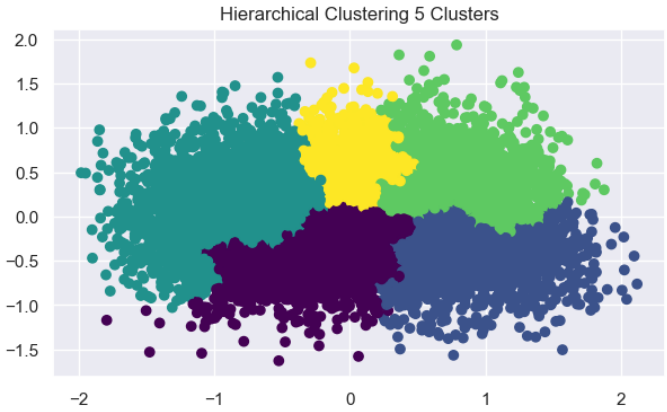


Figure 5.31: Hierarchical No Outliers 5 Clusters

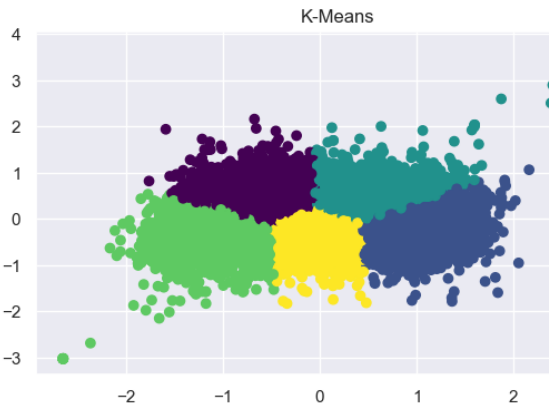


Figure 5.32: K-Means Outliers 5 Clusters

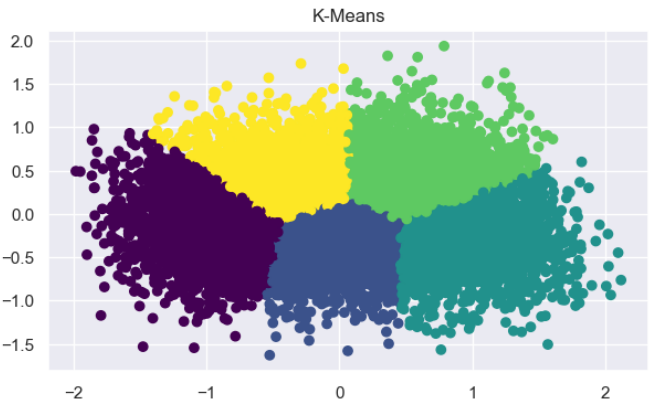


Figure 5.33: K-Means No Outliers 5 Clusters

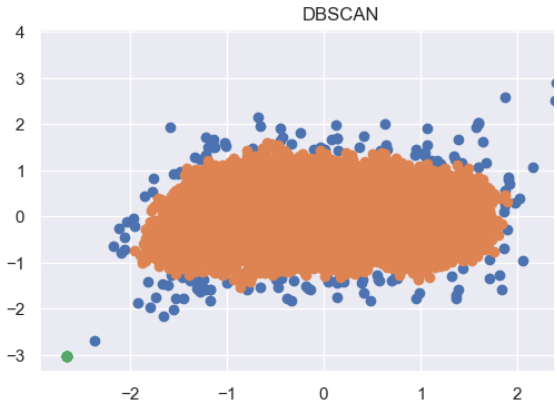


Figure 5.34: DBSCAN Outliers 3 Clusters

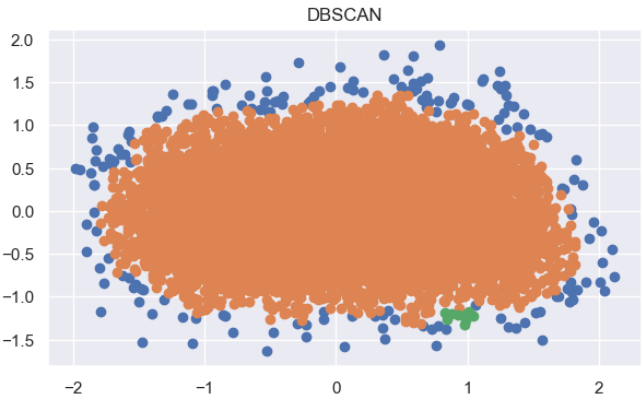


Figure 5.35: DBSCAN No Outliers 3 Clusters

Chapter 6

Conclusions and Future Work

6.1 Conclusions

In conclusion, this report has looked into the usage of different clustering techniques on OCEAN personality types, testing K-Means, DBSCAN, Agglomerative Hierarchical clustering, and GMM with the aims of studying which one worked the best for this scenario. These tests were repeated with and without outliers and with clusters from 2-5 to see what combination of factors worked best for each one. Moreover, clusters of two were treated differently from any other number of clusters since personalities identified in this method would just be reflections of one another and, moreover, no prior papers into the subject needed two clusters. On this note, two clusters did still get the best scores from every algorithm. Scoring how well each algorithm did was performed using silhouette scores. For this, the results of a survey were found online, consisting of 19719 participants. The data was unbalanced but it was left as it was since the goal of this experiment was mainly focused on the clustering algorithms rather than the personality results.

After some initial visualizations to check for any features of note in the data that might need to be corrected or changed during clustering, unnecessary data was dropped. The data was split into two datasets, one with the outliers removed and one with the outliers still in, and PCA was used to reduce the data to two clusters on each data set.

After that, the clustering itself was performed starting with K-Means. The elbow method was used to identify where the optimal number of clusters should lie which sat between 2 and 4. Two was technically the best but four was used due to previously mentioned reasons. It was also found that K-Means worked best with no outliers.

Next was DBSCAN which worked best with the outliers left in. Due to the nature of DBSCAN, only two proper clusters could be created with this method. Taking two variables, epsilon and min-samples, to calculate results, different values of these two values were tested with each other until the perfect combination was found which, for the dataset with outliers, was 12 min samples and 0.12 epsilon.

Then was GMM which did best with no outliers and 3 clusters. The optimal number of clusters was calculating simply by trying all numbers of clusters and checking how each one scored.

Finally, hierarchical clustering did best with 3 clusters and the outliers left in. The optimal

number of clusters was calculated through the dendrogram method.

Overall, DBSCAN did the best but, since it couldn't do more than two clusters, the next best was K-Means followed by GMM with Hierarchical in last.

Extra examination was done into the different personality types obtained from these clusters using four clusters and it was found that the main factors were whether O, C, E, and A together were high or low and whether N was high or low.

6.2 Future work

While some surface level study was done into each algorithm in this paper, there are more factors with each algorithm that can be tested including distance measures, and different types and linkage methods of hierarchical clustering. Without these being tested, it can't truly be certain that the order of "best" algorithms found in this paper is accurate or not, it can only be assumed that the optimal choices have been made and none of these factor changes would push anything further up the list. Moreover, it's possible there's a method of properly changing the number of clusters for DBSCAN which could be found with more research which would certainly influence rankings.

For future work, the first thing that should be tried is testing different linkage methods for hierarchical, specifically the average linkage method since this was the method used in past papers on the subject. Most of the initial structuring work is already done but, due to problems getting other methods to work, more research will need to be done into how to implement different methods to find out where the problem is and how to fix it. From there, the silhouette score for the new experiments should be tested on the outlier and non-outlier data and ranked against the other algorithms.

Besides the silhouette score, there are also other factors that can be analyzed in regards to the clustering algorithms. For example, by starting a timer when the algorithms start running and stopping it when they're finished, the time that each algorithm takes can also be analyzed and compared. Moreover, a silhouette score is made up of two parts – the distances between points within the cluster and the distances between each cluster. These two scores could be calculated individually instead of together, allowing any differences between them to be seen. After all, a lot of the data is pretty tightly grouped together in one location rather than dotted around so the distances between clusters would have less meaning in this case but would still be influencing the score equally as much as the distances between individual points. Moreover, there might be interesting differences between these scores worth comparing and considering.

Another thing that was never finished in this project were the initial plans to display work in a web application for easier understanding for other people. While the work can still be complete, without proper visualization the results become more difficult to share and discuss. This can be done in multiple ways such as finding an application online, using HTML, or finding a python package such as Flask. A good way of utilizing this would be to create tabs or buttons to show the results of each clustering algorithm individually rather than having them all up on screen at once. There could also be a button to switch between the outlier datasets and the non-outlier datasets as well as the numbers of clusters. Finally, a good feature would be a table of the results to display the rankings.

Chapter 7

Reflection

From this project, I've learned how to use different clustering techniques and how each of them works including different methods within those clustering techniques such as Ward and different distance measurements such as euclidean. I've also learned from past papers and my own research what makes a cluster in unsupervised learning considered 'good' as well as how to examine the different points inside different clusters to see what was clustered in what way.

Although there were some ideas to put the results into a web application by the end to make everything more understandable to other people for demonstration, the applications I was recommended to use for this ended up costing money which was outside of budget for the project. Even though there were likely other methods of doing this, I thought they would probably take a lot more coding and programming which I most likely wouldn't have time for. Not if I wanted to do a decent report anyway. As such, this idea was dropped but, if I did this again with more time available, I would do this or, at least, find some other way of visualizing the data in a more pleasing way.

Another thing that could have been done if more time was given would be to try out different methods of hierarchical linkage since those were involved in some of the past literature. However, upon attempt, the outputted results appeared very off leading me to believe there was an error in the code somewhere. Unsure where the error lied and not wanting to waste too much time on it, this idea was dropped. As such, there could be better potential results to be gained from hierarchical under a different method such as average linkage or single linkage.

A few changes were made from the plan to the final result, not for convenience or struggles but because the changes would improve the report. For one, SOM was never tested like planned, instead being replaced by DBSCAN since DBSCAN was recommended more. For another, further study into each of the clusters and their features was performed, despite not being part of the initial plan. This was done to make up for the lack of web page - if I couldn't get the visualisation right, it seemed best to focus on getting better, more perfected research instead with the extra time.

References

Abbas, O. A. (2008), 'Comparisons between data clustering algorithms', *The International Arab Journal of Information Technology* **5**(3), 320–325.

AI, H. (2022), 'Data quality'. (accessed April 17, 2023.
URL: *https://www.heavy.ai/technical-glossary/data-quality*

Bhardwaj, Ashutosh (2020), 'Silhouette coefficient'. (accessed March 28, 2023).
URL: *https://towardsdatascience.com/silhouette-coefficient-validating-clustering-techn*

Bock, T. (2022), 'What is hierarchical clustering?'. (accessed April 6, 2023).
URL: *https://www.displayr.com/what-is-hierarchical-clustering/*

Calzon, B. (2023), 'Your modern business guide to data analysis methods and techniques'. (accessed April 14, 2023.
URL: *https://www.datapine.com/blog/data-analysis-methods-and-techniques/*

Cherry, K. (2022*a*), 'How the myers-briggs type indicator works'. (accessed April 13, 2023.
URL: *https://www.verywellmind.com/the-myers-briggs-type-indicator-2795583*

Cherry, K. (2022*b*), 'What is the enneagram of personality?'. (accessed April 13, 2023.
URL: *https://www.verywellmind.com/the-enneagram-of-personality-4691757*

Cherry, K. (2023), 'Cattell's 16 personality factors'. (accessed April 13, 2023.
URL: *https://www.verywellmind.com/cattells-16-personality-factors-2795977*

Data To Fish (2021), 'Convert a list to pandas dataframe (with examples)'. (accessed April 6, 2023.
URL: *https://datatofish.com/list-to-dataframe/*

Delua, Julianna (2021), 'Supervised vs. unsupervised learning: What's the difference?'. (accessed March 28, 2023).
URL: *https://www.ibm.com/cloud/blog/supervised-vs-unsupervised-learning*

Elias, L. (2020), 'What is personality analysis?'. (accessed April 13, 2023.
URL: *https://santelo.com/what-is-personality-analysis/*

Ellis, Christina (2021), 'When to use gaussian mixture models'. (Accessed 04/04/23).
URL: *https://crunchingthedata.com/when-to-use-gaussian-mixture-models/*

Engati (2021), 'Dbscan'. (accessed March 31, 2023).
URL: *https://www.engati.com/glossary/dbscan*

Fasulo, D. (1999), 'An analysis of recent work on clustering algorithms'.

Garg, Sanjay and Jain, Ramesh Chandra (2006), 'Variations of k-mean algorithm: A study for high-dimensional large data sets', *Information Technology Journal* **5**(6), 1132–1135.

GeeksForGeeks (2022), 'Ml — hierarchical clustering (agglomerative and divisive clustering)'. (accessed March 31, 2023).
URL: *https://www.geeksforgeeks.org/ml-hierarchical-clustering-agglomerative-and-divis*

GeeksforGeeks (2023), 'Elbow method for optimal value of k in kmeans'. (accessed March 28, 2023).
URL: *https://www.geeksforgeeks.org/elbow-method-for-optimal-value-of-k-in-kmeans/*

Gerlach, M., Farb, B., Revelle, W. and Amaral, L. (2018), 'A robust data-driven approach identifies four personality types across four large data sets', *Nature Human Behaviour* **2**, 1.

Gohrani, Kunal (2019), 'Different types of distance metrics used in machine learning'. (accessed April 4, 2023).
URL: *https://medium.com/@kunal_gohrani/different-types-of-distance-metrics-used-in-m*

Google Developers (2022), 'k-means advantages and disadvantages'. (accessed November 28, 2022).
URL: *https://developers.google.com/machine-learning/clustering/algorithm/advantages-disadvantages*

Gordon, S. (2023), 'What are the five love languages?'. (accessed April 13, 2023.
URL: *https://www.verywellmind.com/can-the-five-love-languages-help-your-relationship-*

Graves, Christopher and Matz, Sandra (2018), 'What marketers should know about personality-based marketing'. (accessed March 31, 2023).
URL: *https://hbr.org/2018/05/what-marketers-should-know-about-personality-based-mark*

Hillier, Will (2023), 'A step-by-step guide to the data analysis process'. (accessed March 26, 2023).
URL: *https://careerfoundry.com/en/blog/data-analytics/the-data-analysis-process-step-by-step/*

Hunsley, J., Lee, C. M. and Wood, J. M. (2003), 'Controversial and questionable assessment techniques', *Science and pseudoscience in clinical psychology* p. 39–76.

Irfana Sultana, Shaik (2020), 'How the hierarchical clustering algorithm works'. (accessed March 31, 2023).
URL: *https://dataaspirant.com/hierarchical-clustering-algorithm/*

Jagota, Arun (2020), 'K-means clustering and variants'. (accessed November 28, 2022).
URL: *https://towardsdatascience.com/k-means-clustering-and-variants-703f0a09ac36*

Juma, Stanley (2021), 'Dbscan algorithm clustering in python'. (accessed January 4, 2023).
URL: *https://www.section.io/engineering-education/dbscan-clustering-in-python/*

Kumar, Ajitesh (2020), 'Kmeans silhouette score explained with python example'. (accessed February 20, 23).
URL: *https://dzone.com/articles/kmeans-silhouette-score-explained-with-python-exam*

Kumar, Akash (2023), 'Principal component analysis with python'. (accessed January 23, 2023).
URL: *https://www.geeksforgeeks.org/principal-component-analysis-with-python/*

Lavorini, Vincenzo (2018), 'Gaussian mixture model clustering: how to select the number of components (clusters)'. (accessed March 31, 2023).
URL: *https://towardsdatascience.com/gaussian-mixture-model-clusterization-how-to-sele*

Ligato, J. (2021), 'Personality style clusters using unsupervised machine learning'.

Maklin, Cory (2019*a*), 'Dbscan python example: The optimal value for epsilon (eps)'. (accessed March 28, 2023).
URL: *https://towardsdatascience.com/machine-learning-clustering-dbscan-determine-the-*

Maklin, Cory (2019*b*), 'Gaussian mixture models clustering algorithm explained'. (accessed April 4, 2023).
URL: *https://towardsdatascience.com/gaussian-mixture-models-d13a5e915c8e*

Mane, Tanmay (2021), 'Nearestneighbors to find optimal 'eps' in dbscan'. (accessed January 13, 2023).
URL: *https://www.kaggle.com/code/tanmaymane18/nearestneighbors-to-find-optimal-eps-in-dbscan*

Nevil, Scott (2023), 'How to calculate z-score and its meaning'. (accessed March 28, 2023).
URL: *https://www.investopedia.com/terms/z/zscore.asp*

Pandas Pydata (2023*a*), 'Merge, join, concatenate and compare'. (accessed April 6, 2023.
URL: *https://pandas.pydata.org/docs/user_guide/merging.html*

Pandas Pydata (2023*b*), 'pandas.dataframe.plot.bar'. (accessed April 6, 2023.
URL: *https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.plot.bar.html*

PythonBasics (2021), 'Seaborn heatmap'. (accessed April 4, 2023).
URL: *https://pythonbasics.org/seaborn-heatmap/*

Reddy Patlolla, Chaitanya (2018), 'Understanding the concept of hierarchical clustering technique'. (accessed March 31, 2023).
URL: *https://towardsdatascience.com/understanding-the-concept-of-hierarchical-cluster*

Reece, T. J. (2009), 'Personality as a gestalt: A cluster analytic approach to the big five', *Masters Theses and Specialist Projects.* .

Sava, F. and Popa, R. (2011), 'Personality types based on the big five model: A cluster analysis over the romanian population', *Cognition, Brain, Behavior. An Interdisciplinary Journal* **15**, 359–384.

Sckit Learn (2023), 'Gaussian mixture models'. (accessed April 4, 2023).
URL: *https://scikit-learn.org/stable/modules/mixture.html*

Seaborn Pydata (2023), 'seaborn.catplot'. (accessed April 6, 2023.
URL: *https://seaborn.pydata.org/generated/seaborn.catplot.html*

Sharma, Aditya (2020), 'Principal component analysis (pca) in python tutorial'. (accessed January 28, 2023).
URL: *https://www.datacamp.com/tutorial/principal-component-analysis-in-python*

Sharma, Pulkit (2022), 'A beginner's guide to hierarchical clustering and how to perform it in python'. (accessed January 12, 2023).
URL: *https://www.analyticsvidhya.com/blog/2019/05/beginners-guide-hierarchical-clustering/*

Shilpa, Dang (2015), 'Performance evaluation of clustering algorithm using different datasets', *IJARCSMS* **3**(1), 167–173.

Singh, Aishwarya (2022), 'Build better and accurate clusters with gaussian mixture models'. (accessed April 4, 2023).
URL: *https://www.analyticsvidhya.com/blog/2019/10/gaussian-mixture-models-clustering/*

Singh Chauhan, Nagesh (2022), 'Dbscan clustering algorithm in machine learning'. (accessed November 28, 2022).
URL: *https://www.kdnuggets.com/2020/04/dbscan-clustering-algorithm-machine-learning.html*

StackOverflow (2021), ''dbscan' object has no attribute 'predict' using gridsearchcv and pipeline'. (accessed January 3, 2023).
URL: *https://stackoverflow.com/questions/68920638/dbscan-object-has-no-attribute-predict-using-gridsearchcv-pipeline*

StackOverflow (2022), 'Pandas get frequency of item occurrences in a column as percentage [duplicate]'. (Accessed February 14, 2023).
URL: *https://stackoverflow.com/questions/50558458/pandas-get-frequency-of-item-occurrences-in-a-column-as-percentage*

Thomas (2022), 'What are the big 5 personality traits?'. (accessed April 13, 2023.
URL: *https://www.thomas.co/resources/type/hr-guides/what-are-big-5-personality-traits*

Toushik, Azmine Wasi (2022), 'Different clustering techniques and algorithms'. (accessed December 21, 2022).
URL: *https://www.kaggle.com/code/azminetoushikwasi/different-clustering-techniques-and-algorithms#>>-4.2.-DBSCAN-clustering-algorithm*

VanderPlas, Jake (2016), *Python Data Science Handbook*, O'Reilly Media Inc.

W3Schools (2023), 'Machine learning - k-means'. (accessed January 28, 2023).
URL: *https://www.w3schools.com/python/python_ml_k-means.asp*

Whitfield, B. and Pierre, S. (2023), 'A step-by-step explanation of principal component analysis (pca)'. (accessed May 1, 2023.
URL: *https://builtin.com/data-science/step-step-explanation-principal-component-analy*

Zach (2020), 'How to remove outliers in python'. (accessed April 25, 2023).
URL: *https://www.statology.org/remove-outliers-python/*

Appendix A

An Appendix Chapter

Link to source code and other materials used: https://csgitlab.reading.ac.uk/sw002711/final-year-project
Link to E-Logbook: https://docs.google.com/document/d/1H1HqRYVg6pFszx-ikrRn2wVCfIdPCPiasN9E edit