TPT-DATAAI970 - Project A (2021-2022)

Research Project Report

Clustering for Alzheimer's Disease

Submitted by: Maya Awada

Supervised by: Mounim El Yacoubi

Institut Polytechnique de Paris

Contents

1 Introduction

Alzheimer's Disease is a progressive neurologic disorder that causes degeneration of brain cells leading to loss of memory and other cognitive abilities that might interfere with daily life. Studies have shown that Alzheimer's Disease usually affects motor skills even before medical diagnosis. Therefore, a quantification of the deterioration in motor abilities can be performed based on the gait, voice or handwriting of the patient for example. In particular, alteration in handwriting is one of the first signs of the disease. The handwriting tends to become shaky due to loss of muscle control and forgetfulness [8].

A research project conducted in collaboration between the Institut Polytechnique de Paris and the Broca Hospital of Paris, has set itself the goal of using handwriting as an indicator of early-stage Alzheimer's Disease. Several models, including Bayesian models [14] [15] and a convolutional neural network [4], were successfully built in an aim to classify whether a subject is an Alzheimer patient or not.

This project's goal is to perform a clustering for Alzheimer's Disease patients based on the previously done work. This will allow us to create distinct clusters or groups of Alzheimer patients, each with a potential level of severity. Doctors might then be able to administer different kind of medicine for each cluster or to make the patients follow a certain treatment according to the cluster they belong to.

2 Dataset

The dataset was collected at the Broca Hospital of Paris. The dataset is composed of two cognitive profiles: 27 Healthy Controls (HC) and 27 Early-Stage Alzheimer Disease (ES-AD) individuals. The subjects were asked to perform several handwritten tasks on a tactile tablet. Thanks to the pen associated with the tablet, the following measures were recorded every 8 ms:

- the pen coordinates (x(t), y(t))
- the altitude Alt(t) and the azimuth Az(t)
- the pressure p(t)

The variables collected are illustrated in the following figure. Note that the v_x and v_y velocities were also computed for each recorded observation.

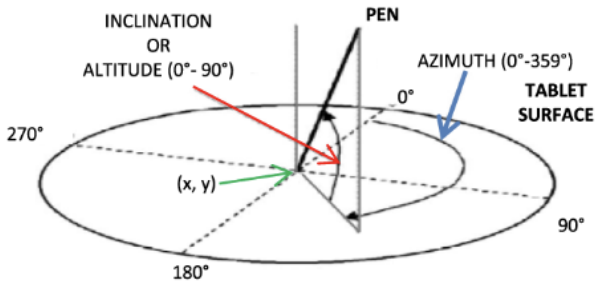


Figure 1: Variables Collected with the Tactile Tablet

In this study, we will focus on the task of drawing 4 series of 4 repetitive l letters. The following figure represents an example of the task performed by each subject.

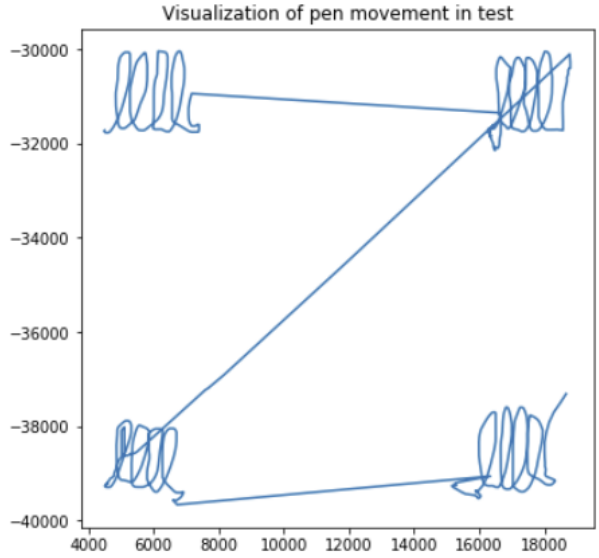


Figure 2: Task Example

A previous study done by S. Lahlali in which he used the same dataset to classify the subjects into HC and ES-AD revealed that the v_x and v_y metrics are the most significant and the most indicative of the presence or not of the disease [4]. Therefore, we mainly used these two metrics in our clustering project.

3 Features Generation

A convolutional neural network was developed in the previous classification project to predict the presence or absence of the disease. We tuned some of the parameters in an aim to reduce over-fitting. The data's input shape of (176, 2) corresponds to the 176 two-dimensional (v_x and v_y) observations for each L letter drawn. The final architecture of the CNN is displayed in the following figure.

Layer (type)	Output Shape	Param #
input (InputLayer)	[(None, 176, 2)]	0
conv1d (Conv1D)	(None, 165, 6)	150
activation (Activation)	(None, 165, 6)	0
conv1d_1 (Conv1D)	(None, 154, 12)	876
activation_1 (Activation)	(None, 154, 12)	0
max_pooling1d (MaxPooling1D)	(None, 38, 12)	0
dropout (Dropout)	(None, 38, 12)	0
conv1d_2 (Conv1D)	(None, 27, 24)	3480
activation_2 (Activation)	(None, 27, 24)	0
conv1d_3 (Conv1D)	(None, 16, 48)	13872
activation_3 (Activation)	(None, 16, 48)	0
max_pooling1d_1 (MaxPooling 1D)	(None, 8, 48)	0
dropout_1 (Dropout)	(None, 8, 48)	0
flatten (Flatten)	(None, 384)	0
dense (Dense)	(None, 144)	55440
dense_1 (Dense)	(None, 1)	145
activation_4 (Activation)	(None, 1)	0

Total params: 73,963
Trainable params: 73,963
Non-trainable params: 0

Figure 3: Model Architecture

The model performs relatively well as it reaches a training accuracy of 80% after 60 epochs as shown in the following plot.

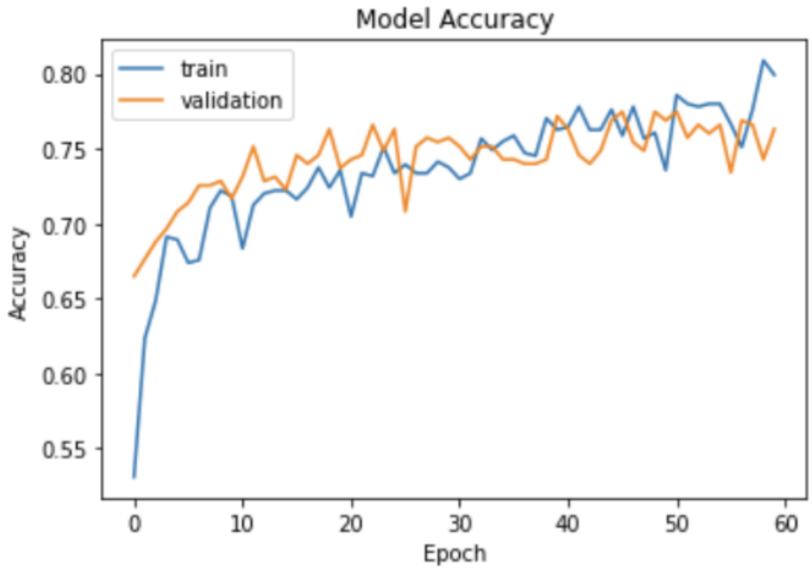


Figure 4: Accuracy versus Epoch Plot

For our clustering project, we need to generate features for each individual. To do so, we built the exact same model as the previous one but we removed the last dense layer. That way, the output shape of our new model is (None, 144) instead of (None, 1). These 144 values are considered to be features generated for each input element, which is an L letter in our case. Given that each individual drew 16 L letters, we computed the average of the 16 extracted features to get a one-dimensional array of features for each individual which will be used as input in the clustering model. This CNN-based feature extraction method has been used in several work including Pneumonia Detection [13] .

Finally, in an aim to study the effect of the number of generated features on the clustering results, we built two new models with an additional final dense layer of 50 and 10 neurons respectively. That way, we get 144, 50 and 10 separately generated features for each subject.

4 Clustering

As mentionned in the previous section, the extracted features were used as input to the clustering model. The k-means algorithm was mainly chosen to perform the clustering. Finding the optimal number of clusters is a key challenge in clustering. We relied on several methods - detailed below - to do so.

4.1 Separability Study

In order to determine the optimal number of clusters, we performed a separability study. Its aim is to find out which clustering separates the two groups of individuals, control and Alzheimer, to the maximum. The study was done on clusterings of size 2 up to 5 for each number of generated features. The separability tables shown below display the proportion of control and Alzheimer patients along with the total number of individuals inside each cluster.

In order to visualize our data on a 2 dimensional graph, we used T-distributed Stochastic Neighbor Embedding. T-SNE is a nonlinear dimensionality reduction technique. After converting similarities between data points to joint probabilities, it tries to minimize the Kullback-Leibler divergence between the joint probabilities of the low-dimensional embedding and the high-dimensional data. T-SNE's non-linearity behaviour makes the technique different from linear dimensionality reduction algorithms such as PCA. In fact, one of the advantages of T-SNE over PCA is that T-SNE retains the local variance of the data while PCA only retains the global variance. In other words, T-SNE works well at preserving small pairwise distance, unlike PCA that only preserves large pairwise distances which can sometimes lead to poor visualization of high dimensional data [2].

T-SNE's main tuneable parameter is perplexity. It refers to the number of effective nearest neighbors. After performing hyperparameter tuning, the optimal perpexity value was found to be equal to 10. Thus, for each point, the algorithm only preserves the distances of its first 10 neighboring points.

The separability graphs shown below plot the embedding of the data in a 2 dimensional space. Each marker color corresponds to a certain cluster. Moreover, the markers' shapes are a visual indication of separability. In fact, the cross-shaped markers represent the Alzheimer patients while the circle-shaped markers represent the control individuals.

	Control	Alzheimer	Size
Cluster 1	0%	100%	22
Cluster 2	84%	16%	32

(a) Number Clusters = 2

	Control	Alzheimer	Size
Cluster 1	0%	100%	18
Cluster 2	57%	43%	21
Cluster 3	100%	0%	15

(b) Number Clusters = 3

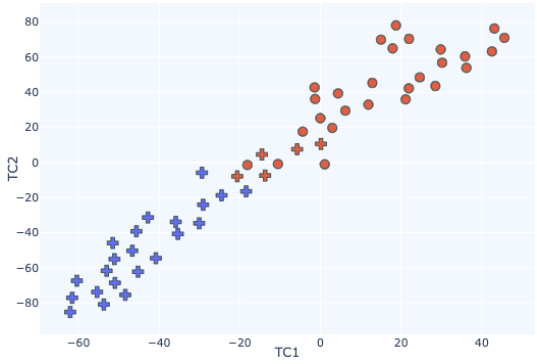
	Control	Alzheimer	Size
Cluster 1	96%	4%	22
Cluster 2	0%	100%	13
Cluster 3	19%	81%	16
Cluster 4	100%	0%	3

(c) Number Clusters = 4

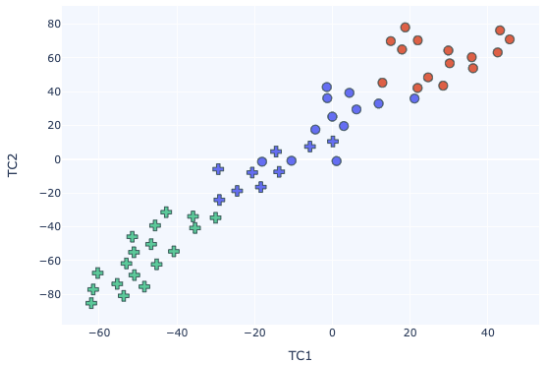
	Control	Alzheimer	Size
Cluster 1	100%	0%	2
Cluster 2	100%	0%	19
Cluster 3	0%	100%	9
Cluster 4	0%	100%	9
Cluster 5	40%	60%	15

(d) Number Clusters = 5

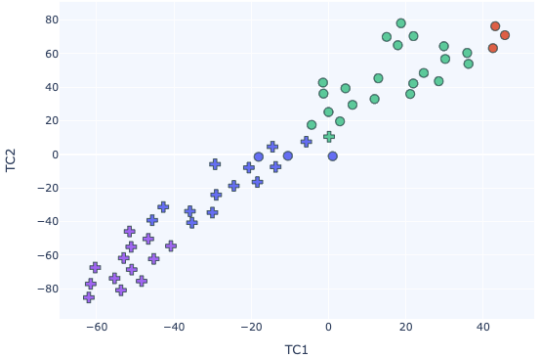
Figure 5: Separability Tables - 144 Features



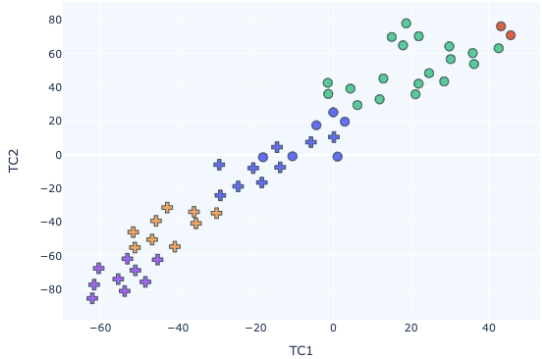
(a) Number Clusters = 2



(b) Number Clusters = 3



(c) Number Clusters = 4



(d) Number Clusters = 5

Figure 6: Separability Graphs - 144 Features

	Control	Alzheimer	Size
Cluster 1	0%	100%	18
Cluster 2	75%	25%	36

(a) Number Clusters = 2

	Control	Alzheimer	Size
Cluster 1	25%	75%	24
Cluster 2	100%	0%	21
Cluster 3	0%	100%	9

(b) Number Clusters = 3

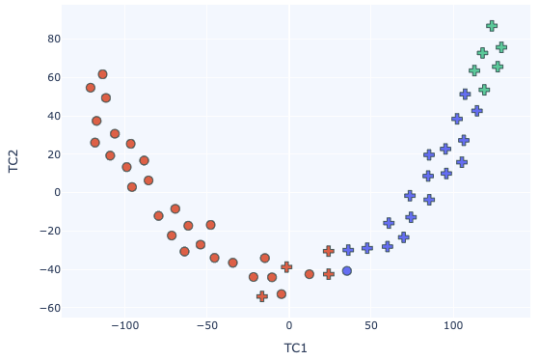
	Control	Alzheimer	Size
Cluster 1	6%	94%	18
Cluster 2	92%	8%	24
Cluster 3	0%	100%	8
Cluster 4	100%	0%	4

(c) Number Clusters = 4

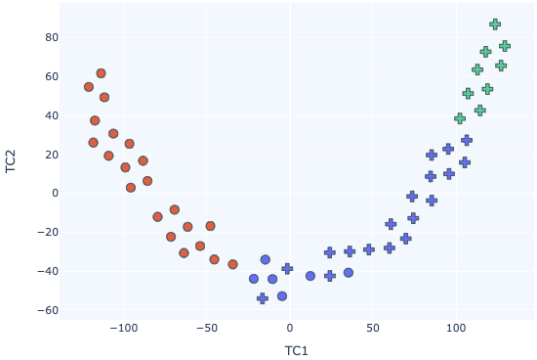
	Control	Alzheimer	Size
Cluster 1	0%	100%	1
Cluster 2	75%	25%	20
Cluster 3	0%	100%	14
Cluster 4	0%	100%	7
Cluster 5	100%	0%	12

(d) Number Clusters = 5

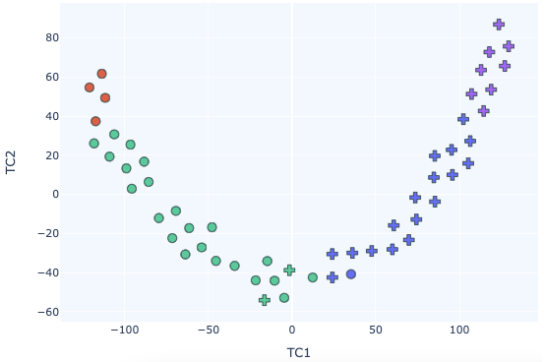
Figure 7: Separability Tables - 50 Features



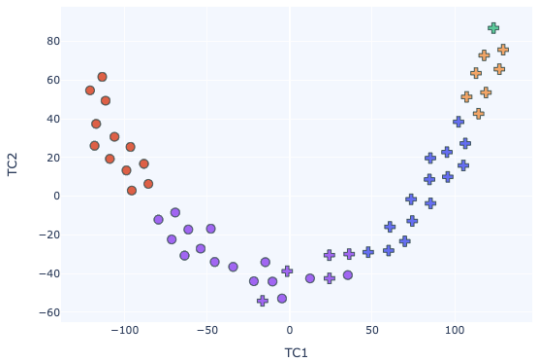
(a) Number Clusters = 2



(b) Number Clusters = 3



(c) Number Clusters = 4



(d) Number Clusters = 5

Figure 8: Separability Graphs - 50 Features

	Control	Alzheimer	Size
Cluster 1	93%	7%	28
Cluster 2	4%	96%	26

(a) Number Clusters = 2

	Control	Alzheimer	Size
Cluster 1	0%	100%	20
Cluster 2	100%	0%	11
Cluster 3	70%	30%	23

(b) Number Clusters = 3

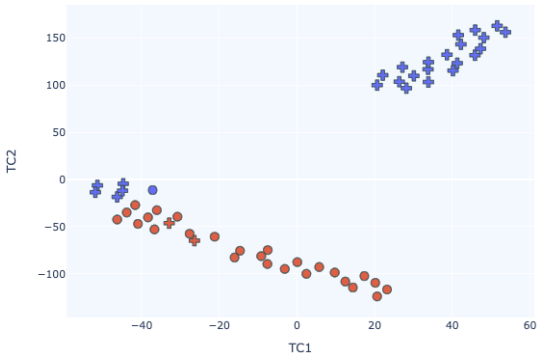
	Control	Alzheimer	Size
Cluster 1	91%	9%	22
Cluster 2	5%	95%	19
Cluster 3	100%	0%	6
Cluster 4	0%	100%	7

(c) Number Clusters = 4

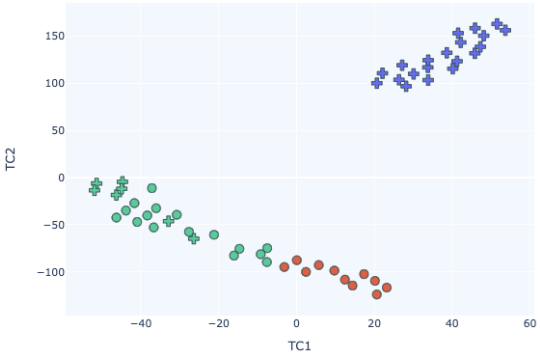
	Control	Alzheimer	Size
Cluster 1	0%	100%	14
Cluster 2	68%	32%	22
Cluster 3	100%	0%	2
Cluster 4	100%	0%	10
Cluster 5	0%	100%	6

(d) Number Clusters = 5

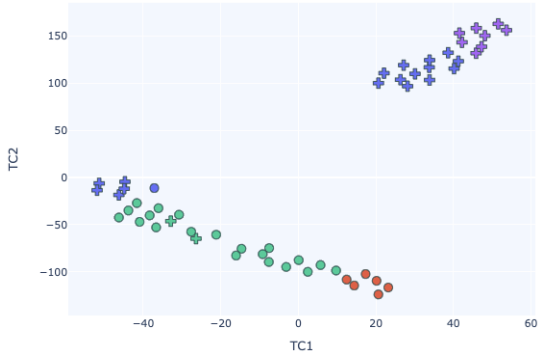
Figure 9: Separability Tables - 10 Features



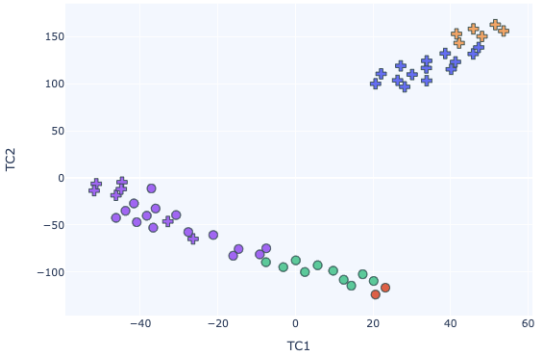
(a) Number Clusters = 2



(b) Number Clusters = 3



(c) Number Clusters = 4



(d) Number Clusters = 5

Figure 10: Separability Graphs - 10 Features

After inspecting each of the tables and graphs and comparing them, we deduce that the clusterings of size 4 produce the best separability. In fact, when the total number of clusters is set to 4, all the clusters are almost fully homogeneous, unlike clusterings of size 3 and 5 in which a heterogeneous cluster with a 60:40 or 70:30 ratio is always found. When k = 4, all the clusters are composed of either a ∼90% control majority, or a ∼90% Alzheimer majority, or they only carry one type of individuals.

We also notice when k = 4, the lower the number of generated features, the more separate the two groups of individuals are. In fact, when we compare the 'c' sub-figures above, we can see that the clusters become more homogeneous when the number of features is lower. Moreover, the 2 dimensional embedding with number of features set to 10 graphically splits the group of individuals, which is not the case with 144 or 50 features.

Clusterings of size 2 also return a relatively good separability. However, one of the goals of this project is to be able to create distinct clusters or groups of Alzheimer patients, each with a potential level of severity. This type of clustering allow doctors to possibly administer different kind of medicine for each cluster or to make the patients follow a certain treatment according to the cluster they belong to. By setting the total number of clusters to 2, we are limited to one cluster with a majority of control individuals and one cluster with a majority of Alzheimer individual, which is not in line with the aim of the project.

4.2 Affinity Propagation

Affinity Propagation, which is considered a similarity-based clustering method, does not require the estimation of the number of clusters before running the algorithm. Therefore, we decided to perform affinity propagation clustering on our data to see how many clusters it will generate. This approach was utilized in a community detection task to determine the optimal number of clusters [11].

In this algorithm, the data points can be seen as nodes in a network where all the points exchange real-valued messages between each other. The content of these messages are the willingness of the points being exemplars. Exemplars are members of the input set that represent a cluster. They are the most significant point of their cluster as they 'best' explain the other data points in the cluster. Every point wants to determine which point corresponds to its exemplar [3]. The messages being exchanged are stored in two matrices.

- The 'responsibility' matrix R. In this matrix, r(i,k) indicates how well-suited point k is to be an exemplar for point i.
- The 'availability' matrix A. In this matrix, a(i,k) reflects how appropriate it would be for point i to choose point k as its exemplar.

The messages sent between pairs of data points are updated in response to the values from other pairs. This updating happens iteratively until a high-quality set of exemplars and corresponding clusters emerges. The final clustering is thus obtained at convergence.

The damping factor is tuneable parameter in affinity propagation that controls the

updating mechanism. It represents the extent to which the current value, or message, is maintained relative to incoming values. The damping factor allows numerical stabilization and can be regarded as a slowly converging learning rate. Out of the [0.5, 1.0) range of possible values, tuning the parameter led an optimal damping factor value of 0.7.

The algorithm returned 6 different clusters, indicating that this is the optimal number of clusters. We built the separability table and graph to see if this clustering separates well the two groups of individuals.

	Control	Alzheimer	Size
Cluster 1	0%	100%	9
Cluster 2	100%	0%	8
Cluster 3	100%	0%	3
Cluster 4	0%	100%	9
Cluster 5	27%	73%	11
Cluster 6	93%	7%	14

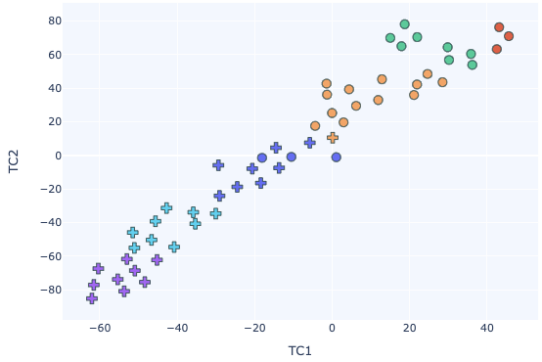


Figure 11: Affinity Propagation Separability Table and Graph

We can see that 3 clusters containing a majority of control individuals and 3 clusters carrying a majority of Alzheimer patients emerged. The results are interesting because all the Alzheimer clusters are big enough be considered a distinct group of patients following a specific treatment. In fact, if the cluster is very small, it might not be worth it for scientists to customize a treatment only for this cluster of patients.

4.3 Silhouette Analysis

A classical technique used to determine the optimal number of cluster and to evaluate the clustering performance is the Silhouette method [10]. The silhouette coefficient is a measure of how similar a data point is to its own cluster in comparison to other clusters. In other words, it is a measure of cohesion within the cluster compared to separation from other clusters. This index has been used in multiple clustering projects including the assessment of Single Nucleotide Polymorphism (SNP) genotype clusters [7]. The silhouette score for a particular data point is computed using the following formula [9].

$$S(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

Figure 12: Silhouette Coefficient Formula

- S(i) is the silhouette coefficient of the data point i.
- a(i) is the mean intra-cluster distance for data point i.
- b(i) is the mean nearest-cluster distance for data point i.

The silhouette score variables are illustrated in the following figure for better visualization.

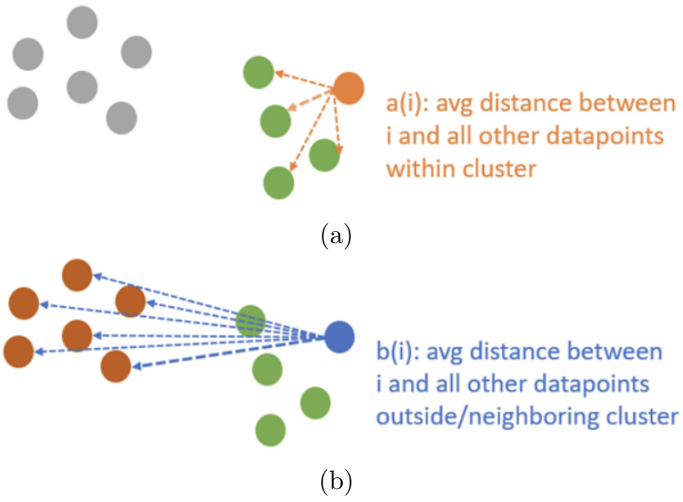
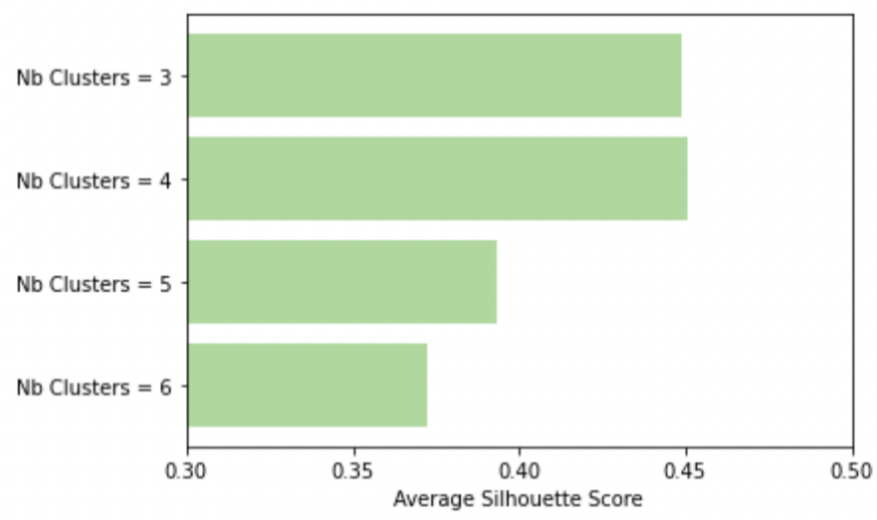


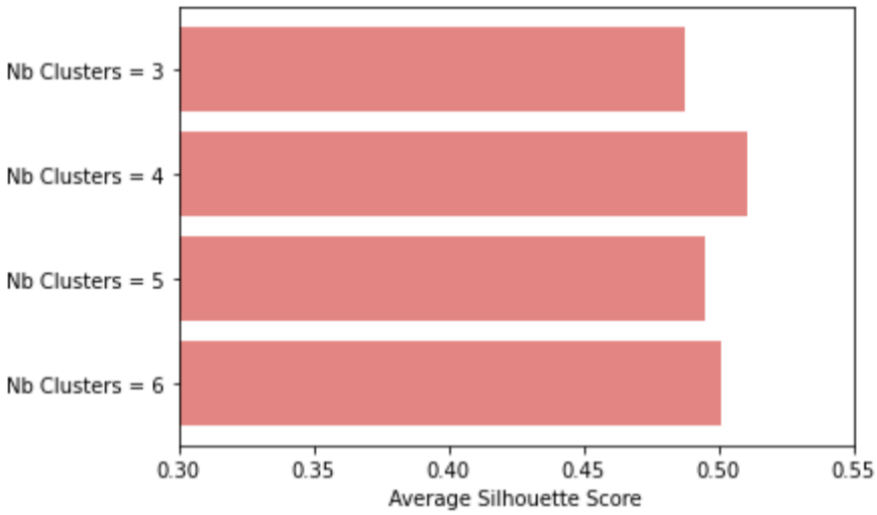
Figure 13: Illustration of Silhouette Score Variables

The resulting silhouette coefficient lies between -1 and 1. A score close to 1 means that the sample is well-clustered and assigned to a very appropriate cluster. While a negative score means that the sample is misclassified.

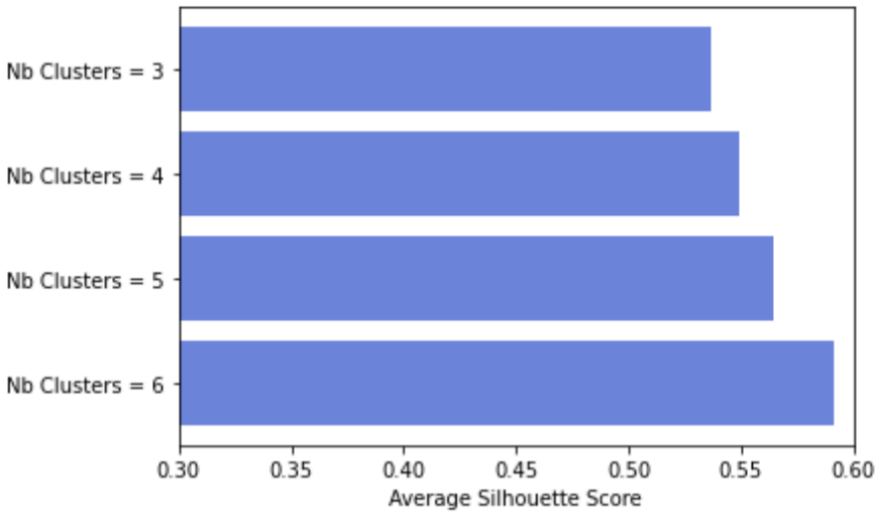
The average silhouette score for all data points is then computed for different values of k from 3 to 6. k=2 was disregarded in this study as per the explanation provided at the end of section 4.1. Moreover, we performed these computations for each number of generated features. The following bar charts plot the average silhouette score for different number of clusters and different number of features.



(a) Number Features = 144



(b) Number Features = 50



(c) Number Features = 10

Figure 14: Average Silhouette Score Graphs

The optimal number of clusters k is usually the one that maximizes the average silhouette score. With 144 and 50 features, the highest score was obtained with k=4 clusters. However, we can see that all the scores are close to each other, so we cannot draw a firm conclusion solely from this method. Moreover, with 10 features, the winning number of clusters is k=6.

When we compare the 3 bar charts, we realise that all the scores increase as the number of features decreases. In fact, we observe the highest scores when the number of features is equal to 10. This result is in line with the 2 dimensional plots of section 4.1 because the separation distance between the resulting clusters is more significant with 10 features than 50 and 144 features.

Finally, we generated a silhouette plot for k=4 as it corresponds to the winning number of clusters in the first two cases. This plot summarizes all the silhouette coefficient values for each cluster.

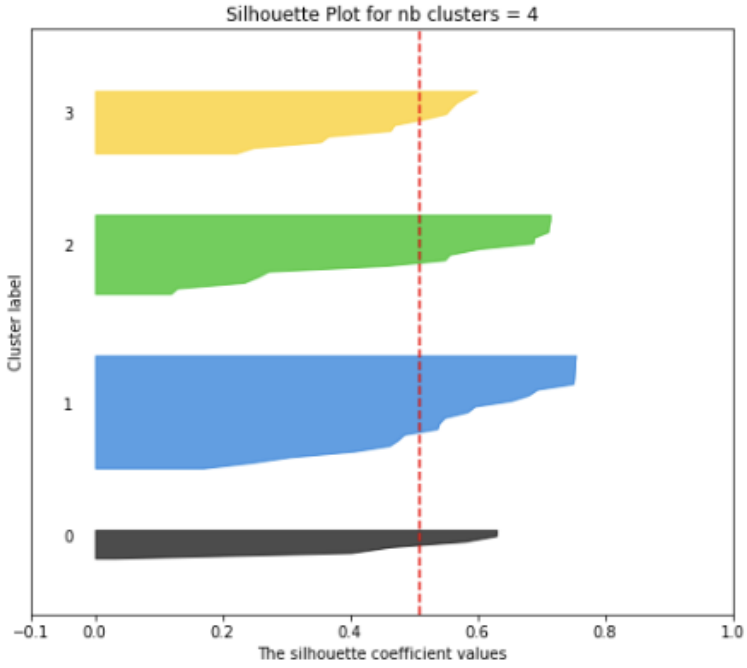


Figure 15: Silhouette Plot for Nb Clusters= 4

From the silhouette plot, we can see that all the clusters have above average silhouette scores with the average being represented by the red dotted line. The cluster sizes can be also be visualized from the thickness of the plot. The 4 clusters have different sizes as previously observed in the separability tables in section 4.1.

4.4 Prediction Strength

Prediction strength is a measure proposed by Tibshirani and Walther in 2005 to evaluate the number clusters in a dataset [12]. The main idea is to view clustering as a supervised classification problem, in which the "true" class labels should also be estimated. The resulting prediction strength value assesses how well and how many groups can be predicted from the data.

We can break down the algorithm into the following steps:

1. Splitting the dataset into training (X_{tr}) and test (X_{te}) sets.
2. Running the clustering algorithm on both sets using a particular value of k.
3. Creating a co-membership matrix D[C(X_{tr}, k), X_{te}] of size $n_{test} \times X_{te}$, where X_{te} is the number of observations in the test set and C(X_{tr}, k) is the clustering algorithm fitted to the training set.
4. Setting the ii'-th element of the co-membership matrix to 1 if elements i and i' of the test set fall into the same cluster, and to 0 otherwise.
5. For each pair of test observations that are assigned to the same test cluster, determining whether they are also assigned to the same cluster based on the training set centroids. The goal of this step is to measure how well the training set centroids predict co-memberships in the test set. The following image extracted from the authors' paper illustrates the idea.

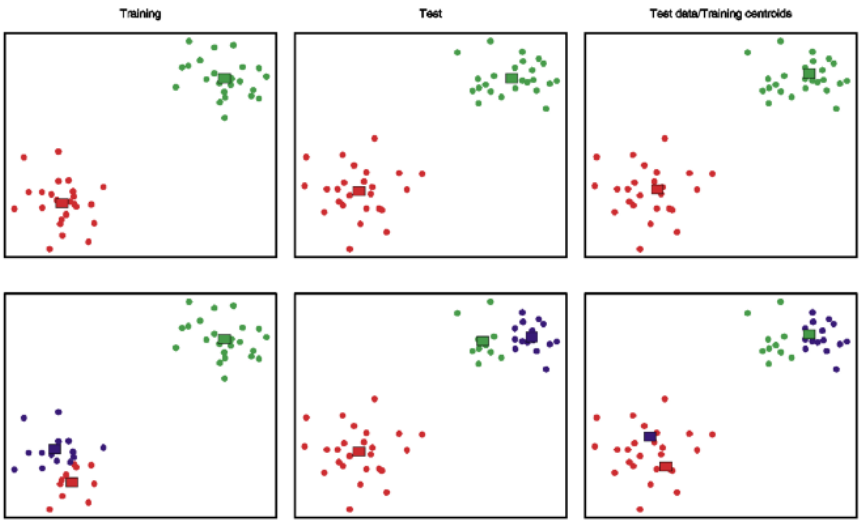


Figure 1. Illustration of prediction strength idea. Data are simulated in two well-separated clusters. In the top row k-means clustering with two centroids is applied to both the training and test data. In the top right panel, the training centroids classify the test points into the same two green and red clusters that appear in the middle panel. In the bottom row, however, when three centroids are used, the classifications by test and training centroids differ considerably.

Figure 16: Illustration of Prediction Strength Idea

6. The prediction strength of the clustering C(., k) is defined as:

$$\mathrm{ps}(k) = \min_{1 \le j \le k} \frac{1}{n_{kj}(n_{kj}-1)} \sum_{i \neq i' \in A_{kj}} D[C(X_{\mathrm{tr}}, k), X_{\mathrm{te}}]_{ii'}.$$

Figure 17: Prediction Strength Formula

where n_{kj} is the number of observations in the j-th cluster and A_{kj} is the index of the test observations in test cluster j for a candidate number of clusters k.

7. For each of the test clusters, calculating the proportion of observation pairs in that cluster that are also assigned to the same cluster using the training set centroids. The prediction strength corresponds to the minimum of this quantity over the k test clusters.

We ran steps 2–7 of the algorithm described above for cluster sizes up to 6. Note that we decided to work with the 10-features samples because they are the one that formed most separated clusters. The results are summarized in the following figure.

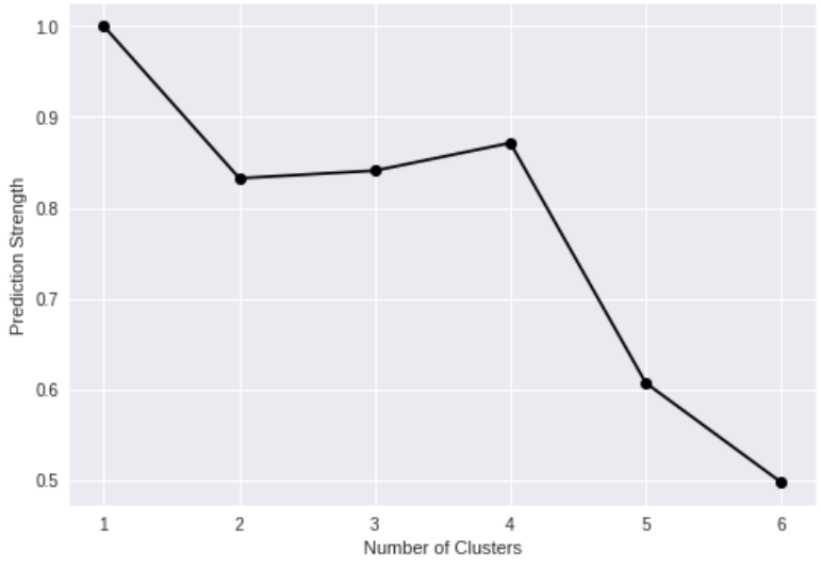


Figure 18: Prediction Strength Graph

The maximum cluster size for which the prediction strength is above a given threshold is then selected. Experiments ran by the authors suggest 0.8 as a good value for the threshold. Based on the results, 2, 3 and 4 clusters are good candidates for the number of clusters but the optimal one appears to be 4.

4.5 Normalized Mutual Information

Normalized Mutual Information (NMI) is a measure for assessing the quality of a clustering. It is comprehensive in the sense that it allows the comparison of two partitions even with a different number of clusters [1]. NMI is a mutual information metric whose value is normalized to ranges between 0, indicating the absence of mutual information, and 1, indicating perfect correlation. It is computed using the following formula [5].

$$NMI(Y, C) = \frac{2 \times I(Y; C)}{[H(Y) + H(C)]}$$

Figure 19: Normalized Mutual Information Formula

where,

- Y = class labels
- C = cluster labels
- H(.) = Entropy
- I(Y;C) = Mutual Information between Y and C

An interesting observation is that the NMI's denominator penalises the increase in the number of clusters. We computed the NMI scores for k ranging between 2 and 6 for the different number of generated features. The results are summarized in the following tables.

	NMI Score
Clusters = 2	0.593
Clusters = 3	0.480
Clusters = 4	0.505
Clusters = 5	0.482
Clusters = 6	0.427

(a) 144 Features

	NMI Score
Clusters = 2	0.479
Clusters = 3	0.524
Clusters = 4	0.462
Clusters = 5	0.465
Clusters = 6	0.449

(b) 50 Features

	NMI Score
Clusters = 2	0.695
Clusters = 3	0.488
Clusters = 4	0.513
Clusters = 5	0.443
Clusters = 6	0.416

(c) 10 Features

Figure 20: Normalized Mutual Information Scores

We noticed that k = 2 is always associated with the highest or one of the highest NMI scores. This behavior is expected as the targets or the class labels are also binary. Moreover, we realized that k = 4 always corresponds to the second highest NMI score proving that it is a good candidate.

4.6 Consensus Clustering

Consensus Clustering is a technique of combining multiple clusters into a more robust single cluster which is usually better than the input clusters. In fact, methods like K-means use a random start procedure where every run may produce slightly different results. By combining several runs of the clustering algorithm, we can generate a more stable clustering. The process is done iteratively by generating a Consensus Matrix at each level [6]. The technique's workflow is illustrated in the following figure.

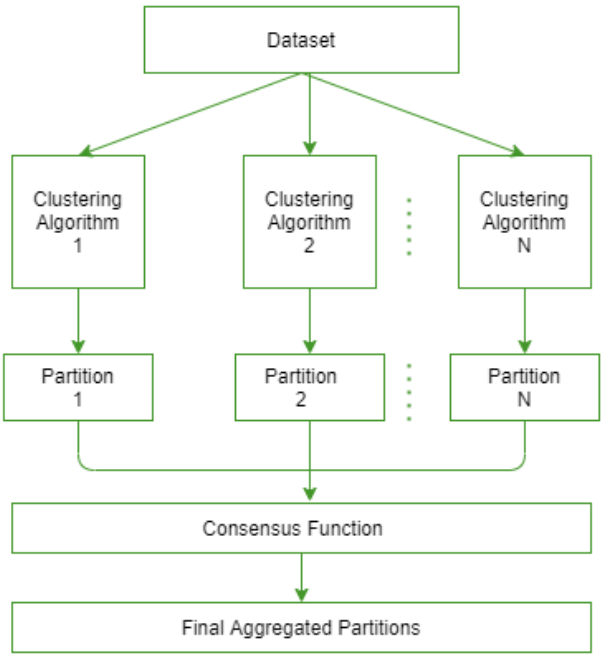


Figure 21: Workflow of Consensus Clustering

We studied the separability of control and Alzheimer individuals before and after applying consensus clustering as seen in the following tables.

	Control	Alzheimer	Size
Cluster 1	100%	0%	15
Cluster 2	57%	43%	21
Cluster 3	0%	100%	18

(a) Before Consensus

	Control	Alzheimer	Size
Cluster 1	0%	100%	18
Cluster 2	88%	12%	17
Cluster 3	63%	37%	19

(b) After Consensus

Figure 22: Consensus Clustering Effect

We noticed that consensus clustering improved the separability of control and Alzheimer individuals to some extent. In fact, it reduced the heterogeneity of the largest cluster in the above case. Consensus clustering can therefore be used in our study in an aim to built a more stable clustering.

5 Conclusion

This research project successfully clustered the individuals at our disposal. It was observed that a lower number of generated features leads to better clustering results. Several techniques were also employed in an aim to determine the optimal number of clusters, being one of the main goals of the project. The separability study revealed a high separation of the two groups of individuals when k = 4, as the clusters were the most homogeneous in this case. The affinity propagation clustering algorithm returned 6 clusters after running it on our data, indicating that it is also a good candidate. In the silhouette analysis, the highest scores were observed at k = 4 and 6, confirming the results of the two previous techniques. The prediction strength method showed that k = 4 is the one associated with the highest prediction strength even though 2 and 3 clusters were also good candidates as they all exceeded the 0.8 threshold. Finally, the normalized mutual information technique linked k = 4 to the second highest NMI score after k = 2. However, by setting the total number of clusters to 2, we are unable to create subgroups of Alzheimer patients, which is why we decided to disregard it.

To conclude, both k = 4 and 6 appeared to be the most suitable numbers of clusters in this study. Choosing between the two depends on the doctors' goal. If they are interested in creating two groups of Alzheimer patients, each with a potential level of severity, when they should opt for k = 4. However, if they want to generate three distinct groups of patients, then they should go for k = 6. This clustering can now be applied on new patients at the Broca Hospital to discover into which cluster they fall and possibly make them follow a certain treatment according to the cluster they belong to.

References

[1] A. Amelio and C. Pizzuti. "Is normalized mutual information a fair measure for comparing community detection methods?" In: *Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, Paris*. 2015, pp. 1584–1585.

[2] B. Devassy and S. George. "Dimensionality reduction and visualisation of hyperspectral ink data using t-SNE". In: *Forensic science international* 311 (2020), p. 110194.

[3] Delbert Dueck. *Affinity propagation: clustering data by passing messages*. University of Toronto Toronto, ON, Canada, 2009.

[4] S. Lahlali. *Detection of Alzheimer Disease on small dataset using ensemble Deep Learning*. Institut Polytechnique de Paris, 2021.

[5] A. Lancichinetti, S. Fortunato, and J. Kertész. "Detecting the overlapping and hierarchical community structure in complex networks". In: *New journal of physics* 11.3 (2009).

[6] Andrea Lancichinetti and Santo Fortunato. "Consensus clustering in complex networks". In: *Scientific reports* 2.1 (2012), pp. 1–7.

[7] L. Lovmar et al. "Silhouette scores for assessment of SNP genotype clusters". In: *BMC genomics* 6.1 (2005), pp. 1–6.

[8] R. Saha et al. "Handwriting Analysis for Early Detection of A lzheimer's Disease". In: *Intelligent Data Analysis: From Data Gathering to Data Comprehension* (2020), pp. 369–385.

[9] K. Shahapure and C. Nicholas. "Cluster quality analysis using silhouette score". In: *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE. 2020, pp. 747–748.

[10] Meshal Shutaywi and Nezamoddin N Kachouie. "Silhouette analysis for performance evaluation in machine learning with applications to clustering". In: *Entropy* 23.6 (2021), p. 759.

[11] S. Taheri and A. Bouyer. "Community detection in social networks using affinity propagation with adaptive similarity matrix". In: *Big data* 8.3 (2020), pp. 189–202.

[12] R. Tibshirani and G. Walther. "Cluster validation by prediction strength". In: *Journal of Computational and Graphical Statistics* 14.3 (2005), pp. 511–528.

[13] D. Varshni et al. "Pneumonia detection using CNN based feature extraction". In: *2019 IEEE international conference on electrical, computer and communication technologies (ICECCT)*. IEEE. 2019, pp. 1–7.

[14] M. El-Yacoubi et al. "Characterizing early-stage Alzheimer through spatiotemporal dynamics of handwriting". In: *IEEE Signal Processing Letters* 25.8 (2018), pp. 1136–1140.

[15] M. El-Yacoubi et al. "From aging to early-stage Alzheimer's: Uncovering handwriting multimodal behaviors by semi-supervised learning and sequential representation learning". In: *Pattern Recognition* 86 (2019), pp. 112–133.