Project 2 Report - Optimization of Biological Age Prediction from Brain Anatomical Volume Measurements using Subgrouping Models

Manon Dorster, Alexandre Maillard, Antonin Faure

Date : December 22, 2022

Abstract

The aim of our project with the Medical Image Processing Lab was to find out if splitting patients into subgroups based on brain anatomical measurements obtained from magnetic resonance images and developing a prediction model for each subgroup was more accurate for predicting biological age than a global prediction model. We were provided with two datasets comprised of volumetric measurements of brain areas, the age and the gender for each patient, and found out that subgrouping improved prediction accuracy.

I Introduction

Our project consisted in developing a model from two data sets in order to predict the age of patients based on brain anatomical volumetric measurements. As of now, the most accurate age predictions based on anatomical features are around 3 or 4 years [1] [3]; we attempted to attain higher accuracy throughout this project.

The aim of the project was two-fold. In the first place, we reduced the dimensionality of our features to keep only the most relevant features and then tested several models to determine the age. Then, we searched for subgroups of patients by clustering the data points, trained and tested models on each subgroup separately. Our initial hypothesis was that the second method (clustering the patients into subgroups then training and testing different models on each cluster individually) would be more precise in the age prediction process.

II Data Analysis

a. Data Set Description

We were provided with two datasets for this project, which were sent to us by the MIPLab:

- Dataset 1: a set of 335 features for 133 patients aged between 68 and 85 years. The volume measurements were given in voxels per AAL.
- Dataset 2: a set of 175 features for 231 patients aged between 49 and 73. The volume measurements were given in both voxels per AAL and millimeters cubed.

The number of patients was quite small in both datasets, and we came to the conclusion that our models would be highly affected by the small size of the samples and would most likely **overfit**. For better results, it would be a good choice to use much larger datasets.

We thought of merging the datasets according to matching brain areas to increase the number of samples. However, Dr. Haller, the radiologist who co-supervised the project, advised us not to, since the data acquisition pipelines where not identical, which could lead to false results.

Moreover, as shown in Fig 1 the ages of the patients ranged from 68 to 85 years in the first data set and 49 to 73 in the second. The second dataset covers a wider range of ages, we then determined that our results could be more accurate using the latter since we would like to learn from anatomical changes due to aging. It is important to have baseline values (those for the younger patients in the sample) to determine the relevant changes in brain anatomy that occur during the aging process.

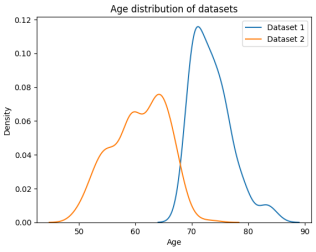


Figure 1: Age distribution of datasets 1 and 2

b. Data Cleaning

Some feature columns comprised of meaningless values, such as columns containing only zeros. These entries were removed. In other feature columns, only a few entries were equal to zero. We chose to replace those entries by the mean value of the corresponding feature column.

c. Splitting into Training and Test Sets

In order to train and test our prediction models, we needed to divide the datasets into training sets and test sets. We generated training and test sets by randomly assigning 70% of the data points to the training set and 30% to the test set.

d. Data Standardization

Standardization is an important step in machine learning and reduces the risks of feature matrix ill-conditioning. We implemented a function to standardize the numerical features of our data sets by subtracting the training mean and dividing by the training standard deviation for each feature column indivudually. The data must be standardized before dimensionality reduction, clustering classification and model fitting.

III Feature engineering

a. Dimensionality Reduction

After analyzing the correlation between each feature and the labels (ages), we decided to reduce the dimensions of our dataset using different methods since we did not see any obvious correlations between the features and the ages of the patients.

Principal Component Analysis

By analyzing the principal components of our data sets, we were able to reduce the dimensions to a few features (approximately 20 features) by selecting the principal components which would allow us to keep the biggest part of the information contained in the dataset. By observing Fig. 2, we noticed a difference between datasets 1 and 2: the first 20 principal components explain 88% of the training variance among patients in dataset 2 and only 64% in dataset 1. Moreover, the first principal component for dataset 2 explains 61% of the variance on its own.

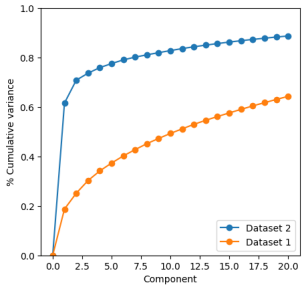


Figure 2: Cumulative variance according to the number of principal components for datasets 1 and 2

Partial Least Squares Regression

After noticing that there was a lot of overlap between points in the subgroups we formed using principal components of our data (see section IV), we decided to implement partial least squares regression in order to reduce the dimensions of the datasets. PLS is a regression method based on covariance. In this case, we exploited the covariance of the features with the age to reduce the number of features.

b. Filtering

MI Score

We compute the MI scores to measure the connection between a feature and the age. High MI score values represent a closer connection between the feature and the label, indicating the level of importance of the feature for model training. However, low MI scores, such as zero, indicate a weak connection between the feature and the label. By computing this score, we isolated the most important features based on a threshold value.

Anatomical relevance

We inspired ourselves from the findings described in the paper [2] to extract features corresponding to the brain areas prone to atrophy during aging. We then trained models on this restrained dataset. Results were, however, not very conclusive.

Lasso

The Lasso method is a regression technique that selects the most relevant variables in a prediction model using L1 regularisation. It sets some of the regression coefficients to zero in order to obtain a simpler model that is easier to interpret, while maintaining good prediction accuracy. The Lasso method is particularly useful in situations where there are a large number of variables and some of them are not very relevant for the prediction of the target variable.

IV Sample Clustering

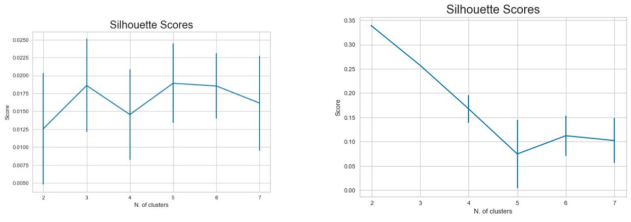
The aim of the project was to determine whether division of the patients into subgroups according to their brain anatomical features would increase the accuracy of our predictions. In the subgroup formation, we only took into consideration the brain area volumetric features and genders of the patients. The biological ages of the patients were left aside in the clustering process and kept as labels for later training.

We thought clustering would allow us to make more accurate predictions of biological age based on different brain aging patterns. In fact, we wanted to see if we would find subgroups of patients whose brain features revealed abnormally advanced brain aging when compared with their biological age, and separate these patients from the rest to predict their biological age more accurately with a targeted model training process. Subgroup division could be useful if such machine learning models were used to predict and diagnose age-related brain diseases such as Alzheimer's disease.

Our idea was to first find an efficient way of clustering the data points, then to test several models on each cluster separately. We implemented k-means and Gaussian Mixture Models to cluster our data and compared the characteristics of the subgroups we found with both clustering methods.

k-Means

K-means is a clustering algorithm that partitions a set of data into K clusters by minimizing the within-cluster variance. It works by iteratively assigning each data point to the cluster whose center is closest and recalculating the centers of each cluster as the mean of the data points in that cluster. K-means is often used for data segmentation and image compression, but it requires specifying the number of clusters in advance and can be sensitive to outliers or complex data structures.



(a) K-Means (b) GMM

Figure 3: Silhouette score plots of K-Means and GMM models

Analysis of the silhouette score, which indicates the proximity of data points to the center of the cluster they belong to as well as the inter-cluster distances, was useful in determining the number of clusters to retain. The silhouette score was high for K=3 clusters (Fig.3a) and we observed an inflection point at K=3 using the elbow method, a method based on the evaluation of the distortion score according to the number of clusters. Thus, we concluded that choosing three clusters would be best; in fact, more than three clusters may be too sensitive to outliers.

Gaussian Mixture Model

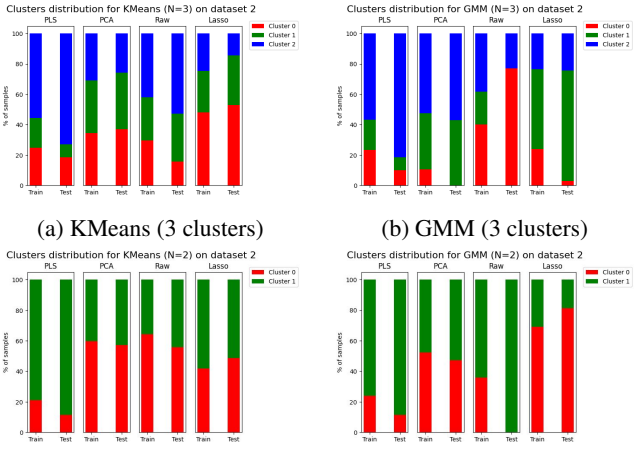
The Gaussian Mixture Model (GMM) is a probabilistic classification model that describes a set of data in terms of multiple Gaussians. Each Gaussian model is defined by its mean and variance and is associated with a weight representing the probability of selecting that Gaussian model when generating data.

We implemented GMM and obtained new clusters for our data. The silhouette score for GMM is highest for K=2 clusters (Fig.3b).

Subgroup Formation

In order to determine which clustering method to use, we analyzed the distribution of data points in the clusters with three or two clusters using GMM or K-means. Ideally, we aimed to have approximately one third of all data points in each cluster when dividing the sample data into three subgroups and one half of all data points in each cluster when dividing into two subgroups.

As can be observed on Fig.4, the data points are highly unequally distributed among clusters with GMM clustering, for both 2-cluster and 3-cluster models. In general, we observed high disproportions between clusters and in some cases absence of one cluster with the GMM clustering method. For example, in dataset 2 with a 3-cluster GMM model, cluster 1 is highly under-represented in the training set and absent in the test set ("Raw" column in Subfig.4 b).



(a) KMeans (3 clusters) (b) GMM (3 clusters)

(c) KMeans (2 clusters) (d) GMM (2 clusters)

Figure 4: Clusters distribution of Lasso, PLS, PCA decompositions and raw data on dataset 2

Moreover, Fig.5 shows cluster densities and the range of patient ages spanned by each cluster. GMM clusters have noticeably unequal densities, thus these would produce biased subsets and lead to inaccurate model training due to the abberantly small sizes of some clusters. Cluster distributions using K-means were not ideal, but we judged that they were much better than the ones obtained with GMM. Therefore, we decided to cluster the samples into three subgroups with K-means clustering. Similar phenomenons concerning the cluster distributions were observed in both datasets.

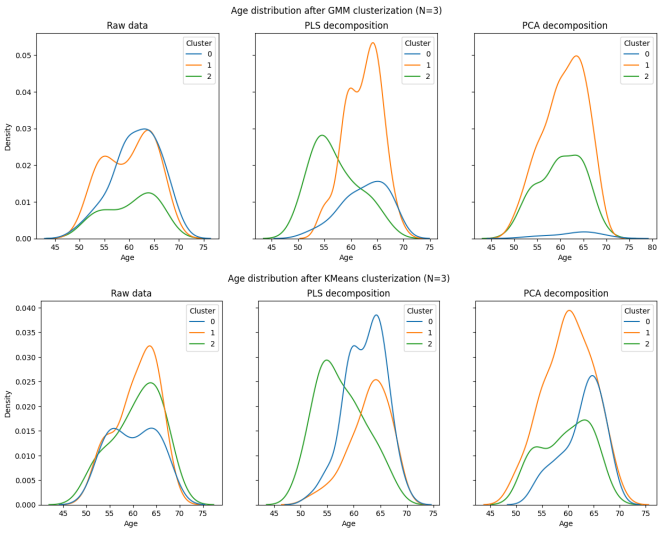


Figure 5: Age distribution of GMM and KMeans clusterization for 3 clusters on dataset 2

V Global Model Prediction

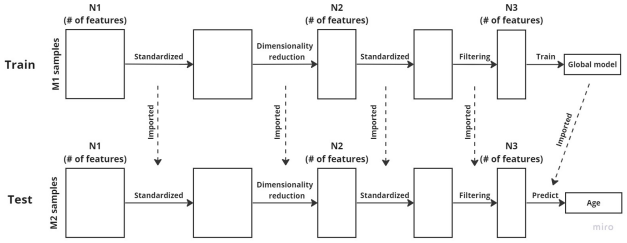


Figure 6: Diagram showing the construction and validation of the global prediction model.

Each of the following models was trained and tested on the entire datasets according to the workflow shown on Fig 6, using GridSearchCV for parameter tuning: *Elastic Net Regression*, *Random Forest Regression*, *Support Vector Regression*, *AdaBoost Regression* and *K-Nearest Neighbors Regression*

VI Model Prediction on Subgroups

After clustering the samples into subgroups as explained in Section IV, we tried to train local models, one per cluster as shown on Fig. 7, and compared them to the global model. If locally the predictions of the global model were better than the predictions of the local one, we used the global model for this cluster sample. We tried to optimize the R2 and MAE scores. For pseudocode of MAE and R2 optimization see B.1 and B.2.

In short, the optimization is all about selecting, for each cluster, the best prediction model between local and global models. The optimized results correspond to the results of the concatenation of the predictions made by the best model (local or global) for each cluster.

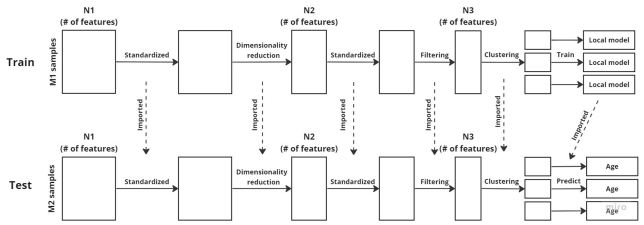


Figure 7: Diagram showing the construction and validation of the local prediction models.

VII Results

According to our results, the optimization process described in Section VI was most effective on dataset 1 (see Table 1) with an R2 score gain of 45% (+0.1) and a MAE score decrease of 10% (-0.26 years) obtained with optimized models compared to the best global results, while on dataset 2 (see Table 2) local results, and thus optimized models, were not better than the best global results.

	Decomposition	Filtering	Clustering	Model	Criterion	Train R2	Test R2	Train MAE	Test MAE
Global	PCA	MI_Score	KMeans (n=3)	ElasticNet	R2	0.46	0.210	1.94	2.670
	PCA	MI_Score	KMeans (n=3)	ElasticNet	MAE	0.43	0.200	1.84	2.621
Optimized	PCA	None	KMeans (n=3)	-	R2	0.571	0.304	1.692	2.401
	PCA	None	KMeans (n=3)	-	MAE	0.737	0.289	1.179	2.361

Table 1: Best global and optimized models on dataset 1 with their corresponding scores

	Decomposition	Filtering	Clustering	Model	Criterion	Train R2	Test R2	Train MAE	Test MAE
Global and optimized	Raw	None	KMeans (n=3)	SVR	R2	0.455	0.255	2.479	3.673
	Raw	None	KMeans (n=3)	ElasticNet	MAE	0.460	0.252	2.797	3.649

Table 2: Best global and optimized models on dataset 2 with their corresponding scores

VIII Conclusion and Future Improvements for Age Prediction Accuracy

To conclude, subgrouping the patients yielded interesting results concerning the accuracy of age prediction. Generally, although the difference in accuracy was not striking, we were able to obtain slightly more accurate age predictions by clustering and fitting models to each cluster separately than by fitting models trained on the entire dataset.

It is important to mention that the datasets we were provided with contained weaknesses, which resulted in high risks of model **overfitting**. The sizes of the datasets were quite small, which proved to be an issue especially when we divided the data points into even smaller subsets. A convenient method to increase the size of the datasets would have been to combine both. However, the problem in image processing data analysis is that it is not good practice to merge datasets that were acquired with different scanners in the case of MR imaging, due to the different acquisition parameters (field strength, flip angle, repetition time, image weighting, etc) that influence the feature measurements. If larger datasets were to be used for future improvements of the age prediction model development, the same scanning conditions would have to be applied for acquisition of data from more patients.

In addition, it would be interesting to acquire data from patients spanning a wider age range. In our datasets, patient ages ranged between 49 and 85 years old. Hence, our results were biased due to the lack of a baseline to quantify age-related anatomical features. Acquiring data from younger patients could potentially yield more accurate results.

To develop a more accurate subgroup it could also be useful to have multiple samples from a same patient but at different ages in order to develop aging pattern from it and thus a subgroup.

A Results

	Decomposition	Filtering	Clustering	Set	Model	Criterion	Model	Parameters	Train R2	Test R2	Train MAE	Test MAE
Dataset 1	PLS	MIscore	KMeans (n=3)	Global	Global	R2	AdaBoost	n estimators=181, random state=123	0.93	0.07	0.73	2.78
					Global	MAE	KNN	n neighbors=4	0.73	-0.0	1.34	2.73
				Cluster 0	Local	R2	AdaBoost	n estimators=181, random state=123	0.82	-0.28	0.78	1.99
					Local	MAE	KNN	n neighbors=4	0.57	-1.73	1.18	2.40
				Cluster 1	Local	R2	KNN	n neighbors=8	0.57	0.30	2.00	2.55
					Local	MAE	KNN	n neighbors=8	0.57	0.30	2.00	2.55
				Cluster 2	Local	R2	AdaBoost	n estimators=181, random state=123	0.90	0.09	0.84	2.77
					Local	MAE	RandomForest	max depth=16, n estimators=50	0.94	0.054	0.61	2.65
				Optimized	-	R2	-	-	0.72	0.18	1.34	2.58
					-	MAE	-	-	0.71	0.11	1.34	2.58
	PCA	None	KMeans (n=3)	Global	Global	R2	Elastic Net	alpha=0.825, l1 ratio=0.0, max iter=5000	0.44	0.20	1.97	2.64
					Global	MAE	ElasticNet	alpha=0.825, l1 ratio=0.0, max iter=5000	0.44	0.20	1.97	2.64
				Cluster 0	Local	R2	Elastic Net	alpha=0.825, l1 ratio=0.30, max iter=5000	0.52	0.38	1.92	2.22
					Local	MAE	Elastic Net	alpha=0.825, l1 ratio=0.30, max iter=5000	0.52	0.38	1.92	2.22
				Cluster 1	Local	R2	RandomForest	max depth=14, n estimators=30	0.87	0.02	0.91	2.25
					Local	MAE	RandomForest	max depth=14, n estimators=30	0.87	0.02	0.91	2.25
				Cluster 2	Local	R2	KNN	n neighbors=4	0.49	0.13	1.57	2.79
					Local	MAE	AdaBoost	learning rate=0.1, n estimators=181, random state=123	0.94	0.11	0.63	2.60
				Optimized	-	R2	-	-	0.57	0.30	1.69	2.40
					-	MAE	-	-	0.74	0.29	1.18	2.36
	PCA	MI Score	KMeans (n=3)	Global	Global	R2			0.46	0.21	1.94	2.67
					Global	MAE			0.43	0.20	1.84	2.62
				Optimized	-	R2			0.56	0.28	1.69	2.50
					-	MAE			0.51	0.30	1.84	2.46
	Raw	None	KMeans (n=3)	Global	Global	R2	RandomForest	max depth=10, n estimators=50	0.85	0.07	0.99	2.90
					Global	MAE	KNN	n neighbors=6	0.37	0.03	2.01	2.87
				Cluster 0	Local	R2	RandomForest	max depth=13, n estimators=70	0.88	-0.07	0.96	2.84
					Local	MAE	RandomForest	max depth=13, n estimators=70	0.88	-0.07	0.96	2.84
				Cluster 1	Local	R2	RandomForest	max depth=19, n estimators=50	0.85	0.18	1.04	2.49
					Local	MAE	AdaBoost	learning rate=0.1, n estimators=161, random state=123	0.99	0.10	0.25	2.29
				Cluster 2	Local	R2	SVR	C=1000.0, gamma=1.83e-05	0.97	0.22	0.22	2.44
					Local	MAE	SVR	C=1000.0, gamma=1.83e-05	0.97	0.22	0.22	2.44
				Optimized	-	R2	-	-	0.91	0.22	0.60	2.52
					-	MAE	-	-	0.96	0.19	0.33	2.45
	Raw	Lasso (thr=0.2)	KMeans (n=3)	Global	Global	R2	KNN		0.58	0.05	1.72	2.83
					Global	MAE	KNN		0.58	0.05	1.72	2.83
				Cluster 0	Local	R2	AdaBoost	n estimators=21, random state=123	1.00	0.06	0.08	3.71
					Local	MAE	AdaBoost	n estimators=21, random state=123	1.0.0	0.06	0.08	3.71
				Cluster 1	Local	R2	Elastic Net	alpha=0.16, l1 ratio=0.0, max iter=5000	0.94	0.29	0.60	2.38
					Local	MAE	Elastic Net	alpha=0.16, l1 ratio=0.0, max iter=5000	0.94	0.29	0.60	2.38
				Cluster 2	Local	R2	Elastic Net	alpha=0.83, l1 ratio=0.0, max iter=5000	0.79	0.04	0.94	2.56
					Local	MAE	Elastic Net	alpha=0.83, l1 ratio=0.0, max iter=5000	0.79	0.039	0.94	2.56
				Optimized	-	R2	-	-	0.82	0.25	1.00	2.66
					-	MAE	-	-	0.77	0.20	1.16	2.65
	PCA	None	KMeans (n=2)	Global	Global	R2	Elastic Net	alpha=0.83, l1 ratio=0.0, max iter=5000	0.44	0.20	1.97	2.64
					Global	MAE	Elastic Net	alpha=0.83, l1 ratio=0.0, max iter=5000	0.44	0.20	1.97	2.64
				Optimized	-	R2			0.61	0.22	1.39	2.53
					-	MAE			0.61	0.22	1.39	2.53
	PLS	MI Score	KMeans (n=2)	Global	Global	R2	AdaBoost	n estimators=181, random state=123	0.93	0.07	0.73	2.78
					Global	MAE	KNN	n neighbors=4	0.73	-0.0	1.34	2.74
				Optimized	-	R2			0.93	0.08	0.73	2.78
					-	MAE			0.73	-0.03	1.35	2.74
	Raw	Lasso (thr=0.2)	KMeans (n=2)	Global	Global	R2	SVR	C=1000.0, gamma=6.16e-05	0.34	0.23	2.92	3.68
					Global	MAE	SVR	C=1000.0, gamma=6.16e-05	0.34	0.23	2.92	3.69
				Optimized	-	R2	-	-	0.98	0.19	0.27	2.54
					-	MAE	-	-	0.98	0.19	0.27	2.54

Table 3: Training and test scores of the best global and local models, as well as the optimized models, for several combinations of decomposition, filtering and clustering methods on dataset 1. **Legend:** light red : the local model is worse than the global model for the cluster; light green : the local model is better than the global model for the cluster; orange : the best global model results; red : the optimized model is worse than the global model; green : the optimized model is better than the global model.

Dataset 2	Decomposition	Filtering	Clustering	Set	Model	Criterion	Model	Parameters	Train R2	Test R2	Train MAE	Test MAE
	PLS	MI Score	KMeans (n=3)	Global	Global	R2	AdaBoost	n estimators=181, random state=123	0.85	0.15	1.55	3.84
					Global	MAE	RandomForest	max depth=19, n estimators=50	0.93	0.13	0.98	3.84
				Cluster 0	Local	R2	AdaBoost	learning rate=0.5, n estimators=181	0.96	0.24	0.71	3.63
					Local	MAE	RandomForest	max depth=19, n estimators=50	0.94	0.13	0.89	3.60
				Cluster 1	Local	R2	AdaBoost	n estimators=181, random state=123	0.84	0.16	1.63	3.60
					Local	MAE	RandomForest	max depth=19, n estimators=50	0.94	0.16	0.97	3.56
				Cluster 2	Local	R2	KNN	n neighbors=7	0.62	0.22	2.25	4.15
					Local	MAE	AdaBoost	n estimators=181, random state=123	0.91	0.19	1.15	4.06
				Optimized	-	R2	-	-	0.78	0.22	1.68	3.84
					-	MAE	-	-	0.93	0.16	1.04	3.78
	PLS	None	KMeans (n=3)	Global	Global	R2	AdaBoost	n estimators=181	0.85	0.17	1.54	3.81
					Global	MAE	AdaBoost	n estimators=181	0.85	0.17	1.54	3.82
				Cluster 0	Local	R2	KNN	n neighbors=4	0.65	0.20	2.36	3.87
					Local	MAE	AdaBoost	learning rate=0.5, n estimators=181	0.96	0.20	0.77	3.81
				Cluster 1	Local	R2	RandomForest	max depth=10, n estimators=70	0.90	0.13	0.95	4.40
					Local	MAE	RandomForest	max depth=10, n estimators=70	0.90	0.13	0.95	4.40
				Cluster 2	Local	R2	RandomForest	max depth=8, n estimators=50	0.91	0.14	1.17	3.29
					Local	MAE	AdaBoost	learning rate=0.5, n estimators=121, random state=123	0.94	0.10	1.03	3.27
				Optimized	-	R2	-	-	0.88	0.20	1.38	3.82
					-	MAE	-	-	0.86	0.17	1.55	3.82
	PCA	None	KMeans (n=3)	Global	Global	R2	SVR	C=215.44, gamma=6.15e-05	0.18	0.19	3.38	3.87
					Global	MAE	SVR	C=215.44, gamma=6.15e-05	0.18	0.19	3.38	3.87
				Cluster 0	Local	R2	AdaBoost	learning rate=0.1, n estimators=101	0.92	0.25	1.13	3.91
					Local	MAE	RandomForest	max depth=1, n estimators=90	0.35	0.25	2.84	2.68
				Cluster 1	Local	R2	SVR	C=10.0, gamma=2.33e-3	0.24	0.11	3.35	3.74
					Local	MAE	KNN	n neighbors=19	0.05	0.11	3.76	3.89
				Cluster 2	Local	R2	AdaBoost	n estimators=141	0.89	0.30	1.32	3.76
					Local	MAE	AdaBoost	learning rate=0.5, n estimators=141	0.89	0.24	1.39	4.67
				Optimized	-	R2	-	-	0.68	0.24	1.91	3.74
					-	MAE	-	-	0.68	0.24	1.91	3.74
	Raw	None	KMeans (n=3)	Global	Global	R2	SVR	C=1000.0, gamma=1.83e-05	0.45	0.25	2.47	3.67
					Global	MAE	Elastic Net	alpha=0.82, l1 ratio=0.0, max iter=5000	0.46	0.25	2.80	3.65
				Cluster 0	Local	R2	Elastic Net	alpha=4.21, l1 ratio=0.0, max iter=5000	0.48	-0.56	2.71	4.95
					Local	MAE	Elastic Net	alpha=4.21, l1 ratio=0.0, max iter=5000	0.48	-0.56	2.71	4.95
				Cluster 1	Local	R2	AdaBoost	learning rate=0.1, n estimators=121	0.98	0.33	0.59	3.62
					Local	MAE	Elastic Net	alpha=0.82, l1 ratio=0.0, max iter=5000	0.80	0.29	1.71	3.55
				Cluster 2	Local	R2	RandomForest	max depth=17, n estimators=70	0.83	0.08	1.53	4.08
					Local	MAE	RandomForest	max depth=17, n estimators=70	0.83	0.08	1.53	4.08
				Optimized	-	R2	-	-	0.45	0.25	2.47	3.67
					-	MAE	-	-	0.46	0.25	2.80	3.65
	Raw	Lasso (thr=0)	KMeans (n=3)	Global	Global	R2	SVR	C=1000.0, gamma=6.15e-05	0.34	0.23	2.92	3.68
					Global	MAE	SVR	C=1000.0, gamma=6.15e-05	0.34	0.23	2.92	3.69
				Cluster 0	Local	R2	SVR	C=46.41, gamma=7.84e-3	0.60	0.18	1.58	3.33
					Local	MAE	SVR	C=46.41, gamma=7.84e-3	0.60	0.18	1.58	3.33
				Cluster 1	Local	R2	KNN	n neighbors=2	0.83	0.30	1.48	3.34
					Local	MAE	Elastic Net	alpha=0.82, l1 ratio=0.0, max iter=5000	0.48	0.24	2.74	3.09
				Cluster 2	Local	R2	SVR	C=2.15, gamma=8.85e-2	0.39	0.17	2.74	4.31
					Local	MAE	SVR	C=2.15, gamma=8.85e-2	0.39	0.17	2.74	4.31
				Optimized	-	R2	-	-	0.43	0.24	2.60	3.73
					-	MAE	-	-	0.39	0.23	2.72	3.67

Table 4: Training and test scores of the best global and local models, as well as the optimized models, for several combinations of decomposition, filtering and clustering methods on dataset 2. **Legend:** light red : the local model is worse than the global model for the cluster; light green : the local model is better than the global model for the cluster; orange : the best global model results; red : the optimized model is worse than the global model; green : the optimized model is better than the global model.

B Algorithms

Algorithm 1 MAE Model Cluster Optimization

procedure MAE_OPTIMIZATION(n, x, y, $global$, $locals$)
  for $i \leftarrow 0$ to $n-1$ **do**
    $x \leftarrow x_{cluster=i}$
    $y \leftarrow y_{cluster=i}$
    $local \leftarrow locals(i)$ ▷ Select cluster's local model
    $predict_{local} \leftarrow$ PREDICT($local, x, y$)
    $predict_{global} \leftarrow$ PREDICT($global, x, y$)
    $MAE_{local} \leftarrow$ MAE($y, predict_{local}$)
    $MAE_{global} \leftarrow$ MAE($y, predict_{global}$)
    if $MAE_{local} < MAE_{global}$ **then**
      $BEST\_MODEL(i) \leftarrow local$
    else
      $BEST\_MODEL(i) \leftarrow global$
    end if
  end for
return $BEST\_MODELS$
end procedure

Algorithm 2 R2 Model Cluster Optimization

procedure R2_OPTIMIZATION(n, x, y, $global$, $locals$)
  for $i \leftarrow 0$ to $n-1$ **do**
    $x \leftarrow x_{cluster=i}$
    $y \leftarrow y_{cluster=i}$
    $local \leftarrow locals(i)$ ▷ Select cluster's local model
    $predict_{local} \leftarrow$ PREDICT($local, x, y$)
    $predict_{global} \leftarrow$ PREDICT($global, x, y$)
    $R2_{local} \leftarrow$ R2($y, predict_{local}$)
    $R2_{global} \leftarrow$ R2($y, predict_{global}$)
    if $R2_{local} > R2_{global}$ **then**
      $BEST\_MODEL(i) \leftarrow local$
    else
      $BEST\_MODEL(i) \leftarrow global$
    end if
  end for
return $BEST\_MODELS$
end procedure

References

[1] Liu et al. Predicting chronological age from brain magnetic resonance imaging using artificial neural networks. 2018.

[2] S.A. Valizadeh et al. Age Prediction on the Basis of Brain Anatomical Measures. 2016.

[3] Zhan et al. A machine learning approach to predict chronological age from brain magnetic resonance imaging. 2018.