UCLouvain

École polytechnique de Louvain

# Machine learning infrastructure for Alzheimer's disease research

Author: **Colin Van Eycken**
Supervisor: **Benoît Macq**
Readers: **Laurence Dricot**, **Lise Colmant**, **Quentin Dessain**, **Sébastien Jodogne**
Academic year 2021–2022
Master [120] in Computer Science and Engineering

# Abstract

Researchers have used increasingly advanced data analysis tools as they've studied the evolution of Alzheimer's disease. This interest in data sciences led to new research merging machine learning and the medical field. Modern techniques and data storage increase the quantity of gathered data. It is interesting to see how machine learning infrastructure may benefit medical research by handling, storing, and analyzing data.

This paper's major goal is to see how data management and analysis (statistical tools and machine learning) might be used in medical research, namely in the follow-up of Alzheimer's patients. In order to build a powerful machine learning model, researchers must gather and concentrate data in a database. Machine learning will be used to predict a patient's cognitive deterioration.

The end-to-end procedure from data collection to machine learning output will represent the pipeline supplied to the research team as a final result to support them in their job. The pipeline is provided with a purpose of continuous learning, which can be used in order to address the issue regarding the possible lack of data. Despite modest predictive power at the time, machine learning in medical research combined with continuous learning seem very promising.

# Acknowledgements

*I would like to express my gratitude to my supervisor, Pr. Benoît Macq, for providing me with the chance to work on such an intriguing topic as the application of machine learning in medical research. I would also like to express my gratitude to Quentin Dessain for the follow-up and the availability given for the development of this project.*

*I would like to thank Pr. Bernard Hanseeuw for introducing me to the field of medical research, as well as the other members of the Institute of NeuroScience who helped me. During the process of developing this thesis, I was guided by a number of people from the Institute of NeuroScience, including Pr. Laurence Dricot, Lise Colmant, Vincent Malotaux, and Lisa Quenon. Finally, I would want to express my gratitude to my family and friends for all of the support they have given me, as well as to my fellow students for the wonderful years that the university has represented for me.*

# Contents

# List of Figures

# List of Tables

# Glossary

**AD** Alzheimer's disease. 17, 22

**API** Application programming interface. 38, 39

**APOE** Gene responsible for the production of apolipoprotein E. 22

**CERAD** Consortium to Establish a Registry for Alzheimer's Disease. 16, 72

**CV** Cross validation. 73

**DBMS** Database management system. 25

**FCSRT** Free and Cued Selective Reminding Test. 15, 16, 70

**GB** Gradient boosting. 73

**HTML** Hypertext markup language. 37, 74

**HTTP** Hypertext transfer protocol. 34

**ICV** Intracranial volume. 42

**LOOCV** Leave-One-Out cross validated. 7, 60

**MCI** Mild cognitive impairments. 19, 22, 44

**MLP** Multilayer perceptron. 55, 57, 60, 61

**MMSE** Mini-Mental state examination. 15, 22, 29

**MRI** Magnetic Resonance Imaging. 14, 19, 20, 22, 25, 29, 41, 42, 53

**NFT** Neurofibrillary tangles. 19

**PET** Positron emission tomography. 17, 22

**PHP** Hypertext Preprocessor. 37

**RAM** Random access memory. 27

**RDBMS** Relational database management system. 26

**REST** Representational state transfer. 38

**RF** Random forest. 73

**RFE** Recursive feature elimination. 62

**RMSE** Root mean squared error. 59–61, 63, 73

**SQL** Structured Query Language. 3, 6, 24–28, 30, 31, 38

**SVM** Support vector machine. 60

**SVR** Support vector regressor. 6, 52, 53

**TMT** Trail making test. 16, 66

# Chapter 1

# Introduction

Over the years of research regarding Alzheimer's Disease, researchers have been using more and more advanced data analysis techniques. This gain of interest towards data sciences lead to more and more projects combining machine learning techniques with medical expertise. Moreover, the tools and data storage available nowadays makes up for an always increasing amount of collected data. It is interesting to evaluate how a machine learning infrastructure, that is managing, storing and analysing data, can be beneficial to the medical field research.

The main objective of the project is to observe to what extent and in what way data management technologies and analysis (statistical tools and machine learning) could be useful for research in the medical field, in particular in the follow-up of patients suffering from Alzheimer's disease.

This report will be divided into two distinct parts: first, the constitution of a database and an interface facilitating the management of research data, by defining a harmonising formatted framework in order to make data easily exploitable. The second component focuses on the usefulness of data analysis at the level of research in the field of Alzheimer's disease as such. The collection and concentration of data in order to facilitate the researchers' work requires the creation of a database in order for the machine learning aspect of the project to be created. The purpose of this machine learning part of the project will be to identify the most relevant features in order to predict the cognitive decline of a patient. The overall end-to-end process from data collection to machine learning output will be the core of the pipeline that will be provided to the research team as a final result to assist them in their work, with the expectation that this tool will provide helpful information.

The aim of this study is to advance the implementation of these promising technologies. While predictive potential remains modest at present, it nevertheless demonstrates the great potential that the use of machine learning in medical research conceals.

# Part I

# State of the art

# Chapter 2

# Alzheimer's disease

Alzheimer's disease is the most prevalent type of dementia and a degenerative brain illness. However, dementia is not defined as a specific illness as such as it is a word for a collection of symptoms. [4]

This illness is classified as "neurodegenerative," which indicates that it will target certain sections of the brain that will eventually die as a result of the illness.[58]

Given that Alzheimer's disease may manifest itself in a variety of ways, making a definitive diagnosis can be difficult. However, Alzheimer's disease can only be detected via a set of evaluations that look at several aspects of your daily life at the same time.

*Early signs and symptoms of Alzheimer's dementia include:*

- *Memory impairment, such as difficulty remembering events.*
- *Difficulty concentrating, planning or problem-solving.*
- *Problems finishing daily tasks at home or at work.*
- *Confusion with location or passage of time*
- *Having visual or space difficulties, such as not understanding distance in driving, getting lost or misplacing items.*
- *Language problems, such as word-finding problems or reduced vocabulary in speech or writing.*
- *Using poor judgment in decisions.*
- *Withdrawal from work events or social engagements.*
- *Changes in mood, such as depression or other behavior and personality changes*

[23]

Some of the aspects outlined above, on the other hand, can also occur as a result of regular aging. In order to identify dementia in Alzheimer's disease, it is necessary to consider the severity of the impairment as well as the mix of multiple impairments observed. These assessments will be described in 2.1 and will be paired with magnetic resonance imaging (MRI) in order to diagnose the patient even more correctly.

## 2.1 Neuropsychological analysis

To evaluate the neuropsychological state of patient, and later diagnose an eventual dementia, several cognitive assessments are performed in order to compute composite scores which will allow to evaluate the state of a patient. You will find here under a description of the assessments that will be used and analysed in this project:

1. Mini-mental state examination (MMSE):
   The MMSE is used as a short-screening tool to measure the global cognitive decline of a patient. An example of a MMSE can be found in Appendices (See Figure 6.1). It can be used as a indicator of mild dementia. [10] This is the assessment that was the most performed and for which there is the most data. The results of various tests are often paired with the MMSE to have a great overview of the cognitive state of a patient.

2. The clock test:
   According to the review of the clock-drawing test by Berit Agrell and Ove Dehlin [3], the "clock-drawing test" is used to look for dementia and cognitive deficits, the test also assesses spatial dysfunction. In addition to constructive skills, passing the test necessitates verbal comprehension, memory, and spatially coded information. This test has been recommended for assessing visuo-spatial deficits, which are a common and early indicator of dementia.

3. Free and cued selective reminding test (FCSRT):
   *"The Free and Cued Selective Reminding Test[17] is a memory test that controls attention and cognitive processing."* [44]
   The FCSRT already proved to be very useful for the detection of memory dysfunction related to Alzheimer's disease. For this reason, the information provided by this evaluation will be incorporated into this project. The objective of this test is to recall the most possible words from a previously given list. There are three sub-categories for this kind of data:

   (a) Free recall (FR): The patient has to recall words without any intervention nor help. This stimulates the patient's ability to build strategies in order to remember the most possible words.

   (b) Total recall (TR): Patients get indications to remember the words from the previously observed list. These indications are supposed to reduce the strategic aspect of the task in order to highlight the memory performance of the patients independently from the strategic capabilities.

   (c) Delayed free recall (DFR): In this part of the assessment, the patients gets asked to recite the words from the list but after a given period of time. This is done to evaluate the consolidation of the memory regarding those words.

4. Fluency:
   The fluency test consists in citing words in a specific semantic category. This test evaluates the verbal fluency of a patient, in a qualitative aspect as well as a quantitative aspect. It has been shown that patients with dementia of Alzheimer type perform poorly to this test compare to healthy other individuals. [33]

Two sub-categories of this test are defined as following:

   (a) Animals: Naming the most possible animals in a given period of time.

   (b) P: Naming the most possible words starting with P in a given period of time.

5. Naming Lexis:
   This test consists in naming objects that are presented to you. The score gives an evaluation of your naming skills and can reflect your language capabilities.[21]

6. Trail Making Test:
   Trails Making Test is a cognitive test of visual attention and task switching. This assessment can give an insight on visual capabilities, scanning, speed of processing, mental flexibility, as well as executive functioning. [72]

7. Luria:
   This assessment provides evaluates the ability of a patient to reproduce a given pattern. This test is described in the Appendices as *"The test of Graphic Series"* of more information.

8. The Consortium to Establish a Registry For Alzheimer's Disease neuropsychological battery (CERAD):
   This is a tool used for the detection of mild cognitive impairments and signs of dementia related to Alzheimer's disease. [63] As it will be stated further, this assessment provides information on the visuo-spatial functions of a patient.

From all the previous cited assessments, "composite scores" are computed in order to get a better overview of the state of a patient. The "Composite scores" decline in the following ways:

1. Memory: Computed from the three FCSRT scores.

2. Executive: Computed from a combination of The TMT as well as the Luria score.

3. Language: Computed on basis of the Fluency tests as well as the Naming test.

4. Visuo-spatial: Computed on basis of the Clock test as well as the CERAD tests.

While these composite scores will aid in the diagnosis of dementia, it is critical to remember that no precise criterion or threshold for the scores allows for a rigorous diagnosis of a patient with dementia. To begin, no one's performance will be identical to another's merely because no one has the same cognitive abilities. Thus, establishing a threshold is not a good idea, since two individuals with the same score may not be deemed to be in the same condition. This is why the score's **evolution** is more significant. A fall in some composite score implies that an individual is losing cognitive abilities and is on the verge of being demented.

Additionally, a patient's fall in a score suggests that this patient needs medical intervention and follow-up. This is why these evaluations are critical in diagnosing dementia in patients with Alzheimer's disease.

## 2.2 Pet-Scan

PET is a minimally invasive imaging method used to differentiate between normal and sick tissue in illnesses such as cancer, heart disease, and neurological disorders like Alzheimer's disease.[5]

The PET scan involves a radioactive substance (tracer) to detect both normal and abnormal metabolic activity.[60] Generally, the tracer is injected into your forearm. The tracer will then accumulate in parts of your body with greater levels of metabolic or biochemical activity, typically pinpointing the region of the illness.[60]

The analysis of the result provided by a PET-scan exam can get interpreted through the use of a measure called SUVr. The standard uptake value ratio (SUVr), also known as the dose uptake ratio (DUR), is a simple way of determining activity in PET imaging, most commonly used in fluorodeoxyglucose (FDG) imaging.[14]

Like mentionned by Lhommel et al.[45], the SUVr proved to be very helpful in the interpretation of results when used to reinforce the visual analysis. Morevover, this measure allows for a longitudinal analysis of the amyloid presence for a given set of patients.[45]
To further support the importance of the SUVr in longitudinal studies, Hanseeuw et al. emphasized the significance of frequent tau-PET observations to monitor disease development and repeated amyloid-PET observations to identify the early AD pathologic changes.[34].

This is why SUVr will be the measure that will be used in this project for the Amyloid-beta protein (noted $a\beta$). Although being important, the TAU-protein information provided by the SUVr will not be used, which will be explained in dataset description section (See Section 4.2).

The development of amyloid plaques ($a\beta$ plaques) in the brain can be triggered by aging and can be a sign of early Alzheimer's disease. Beta-amyloid protein is the most abundant protein in amyloid plaques, which is a protein aggregation seen in the brains of patients suffering from certain neurodegenerative disorders such as Alzheimer's disease. This peptide would have a considerable impact on neuronal transmission levels. PET scans will then be used to determine whether or not beta-amyloid plaques are present in the patient's brain.

While being closely related to the illness, beta-amyloid still is essential as it plays a key function in the formation and repair of the nervous system. Later in life, however, a mutated version of the protein may harm nerve cells, resulting in the loss of cognition and memory associated with Alzheimer's disease.[6]

PET scans may be carried out in a variety of methods to determine the concentration of certain proteins in the brain. Amyloid-PET scans are used to assess the brain's buildup of abnormal amyloid protein, whereas FDG-PET scans are used to determine the brain's glucose concentrations. This concentration explains how the brain uses energy.[35]

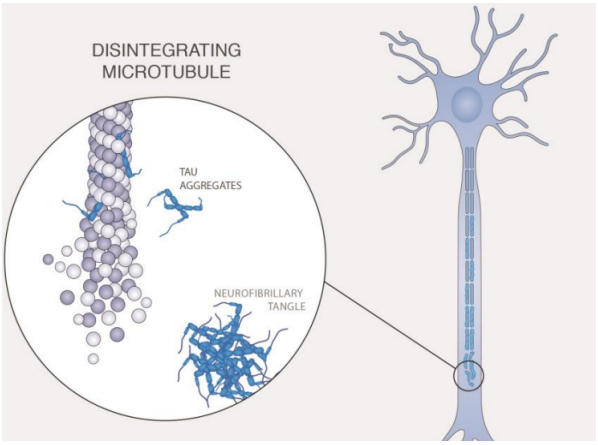



Figure 2.1: The TAU-protein plays a crucial role regarding microtubules in the neuronal axons, a dysfunction in the TAU-protein action results in an overall dysfunctional neuron.[56]

Amongst other proteins, an examination of the presence of the TAU-protein can be performed in order to detect an abnormal concentration of the protein. The presence of TAU-protein, as well as the Amyloid-beta protein are both inspected through the PET-scan process. This is because *"although Aβ accumulation generally precedes tau changes, the exact mechanistic crosstalk between them remains unclear."*[56] The tau protein is located with the highest concentration in neurons.[27] Tau is involved in a variety of functions in healthy brain cells, one of which is maintaining the integrity of intracellular microtubules.[27]

Microtubules are found in abundance in neuronal axons and dendrites, and they account for a significant component of the neuronal cytoskeleton.(See Figure 2.1) This network of microtubules provides structural support for axons and dendrites, allowing them to develop and maintain their unique morphologies.[12]
Microtubules are important in the transport of nutrients and proteins from the cell body to the axon and dendrites of the cell.[36] The TAU-protein having a crucial role regarding microtubules, a dysfunction in the TAU-protein action results in an overall dysfunctional neuron.

In Alzheimer's disease, abnormal molecular changes can cause the tau protein to separate from microtubules and adhere to other tau molecules, generating threads that ultimately unite to create neurofibrillary tangles (NFT) inside neurons which impede the transport mechanism of the cell, hence impairing synaptic contact between neurons.[36]

Increasing evidence suggests that Alzheimer's-related brain changes may be the result of an interaction between abnormal tau and beta-amyloid proteins.[36] These studies suggest that abnormal tau concentrates in certain memory-related brain areas. Plaques of beta-amyloid form between neurons. As the concentration of beta-amyloid hits a critical threshold, tau spreads rapidly all across the brain.[36]

### 2.2.1 Braak stageing[15]

The Braak stageing system gives an overall view of the evolution of Alzheimer's disease. Describing the evolution of neurofibrillary tangles throughout the brain, the differentiation of stages may aid researchers in classifying individuals according to their overall condition of the disease and in determining the level of medical care a patient requires. The different stages are illustrated in Figure 2.2.

Six stages can be described through in Braak's stageing[15]:

1. I-II: Alteration of the transentorhinal.
2. III-IV: Alteration of the entorhinal and transentorhinal.
3. V-VI: Isocortical destruction.

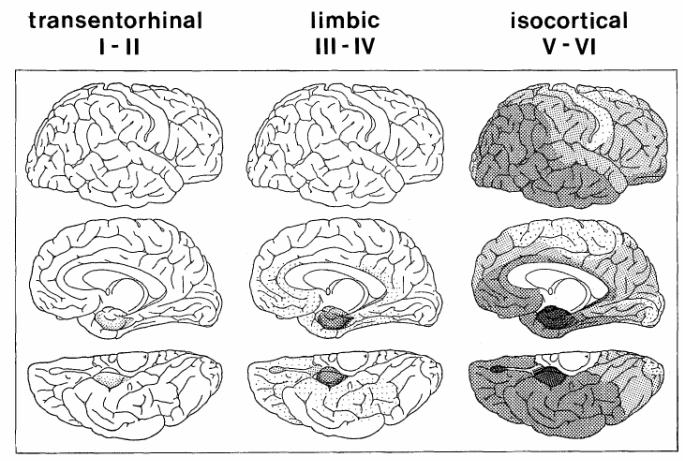



Figure 2.2: Braak stageing for Alzheimer's disease based on the presence of NFT [15]. Variations in the neurofibrillary (NF) distribution pattern. Alterations in the transentorhinal area are restricted to a single layer in the first and second phases. Involvement of the entorhinal and transentorhinal layer to a significant degree is the defining feature of the third and fourth phases. The fifth and sixth phases are distinguished by the destruction of the isocortex. Increasing shade density implies intensifying alterations.[15]

## 2.3 Magnetic Resonance Imaging

Magnetic Resonance Imaging (MRI) is a non-invasive imaging technique that provides anatomical visuals in three dimensions. [76] MRI notably allows for a production of precise images of the brain, including information such as volumes, surfaces, and cortical thicknesses.
MRI may identify brain abnormalities related to mild cognitive impairment (MCI) and is used to predict which MCI patients will suffer from Alzheimer's disease in the future. In the first stages of the illness, a brain MRI may not show any signs of brain alterations. In advanced stages, MRI may reveal a reduction in the size of certain brain regions.[37]

Magnetic resonance imaging (MRI) provides a visual picture of the inside of a body. For Alzheimer's disease research objectives, the MRI will be very beneficial in retrieving information about the patient's brain.

In the article *"Automated cortical thickness measurements from MRI can accurately separate Alzheimer's patients from normal elderly controls"*[11], MRI data have proved to be useful to differentiate demented Alzheimer's patients from normal elderly controls according to [11]. The purpose of this work was to use image-based metrics supplied by the MRI to identify diagnostic areas of interest. The project developed by  is closely related to the objective of this thesis. Which is to attempt to predict composite scores using a variety of data sources, including MRI data.

### 2.3.1 Image acquisition

#### Biomarker cohort

3D T1-weighted images were recorded at 3T (Achieva, Philips Healthcare) with a 32-channel phased-array head coil. One hundred and fifty sagittal slices were acquired using the following parameters: TR/TE/FA 9.1 ms/4.6 ms/8°; slice thickness 1 mm; resolution $0.81 * 0.95\ mm^2$ (acquisition) reconstructed in $0.75 * 0.75\ mm^2$; FOV $220 * 197\ mm^2$; acquisition matrix $296 * 247$; SENSE factor 1.5 (parallel imaging).

#### EPAD protocol

3D T1-weighted images were recorded at 3T (Signa Premier, GE Healthcare) with a 48-channel phased-array head coil. One hundred and ninety-six sagittal slices were acquired using the following parameters: TR/TE/FA 7.2 ms/2.9 ms/11°; slice thickness 1.2 mm; acquisition matrix $256 * 256$; FOV $270 * 270\ mm^2$; resolution $1.055 * 1.055\ mm^2$; ASSET factor 1.75 (parallel imaging).

### 2.3.2 Image processing

The MRI data were processed using *"Freesurfer"*[31].
FreeSurfer is a computer software used for the processing and display of cross-sectional or longitudinal structural and functional neuroimaging data.[30] This allowed for the subdivision of volumes, surfaces, curvatures, and thicknesses into regions. With the objective of predicting the score on a cognitive composite evaluation, the feature selection process that enables the identification of regions of interest will be a significant aspect of this project.

#### Atlases for regions subdivision

In order for researchers to inspect the condition of a patient's brain, MRI data can be provided through the use of atlases. Atlases are maps built with a purpose of labelling and delimit regions. This allows to inspect the brain by region. It is important to note that several atlases have been designed and that no standard has been defined.

Two atlases were used in order to divide the images into regions: Desikan-Killiany and Destrieux. (See Figure 2.3)

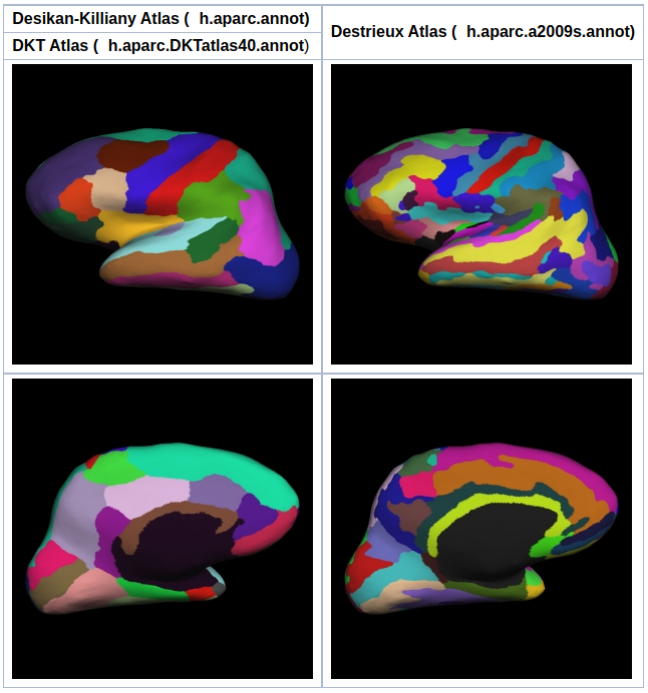



Figure 2.3: Freesurfer[19]: Desikan-Killiany[22] and Destrieux [29]

**Freesurfer**

The data provided by Freesurfer that will be used to filter the regions of interest are from the "aparc" file which is segmented according to the Desikan-Killiany atlas. You can find the details of this segmentation in Appendix 5.1.

A choice had to be made on which atlas to use for this project. We decided to use the Desikan-Killiany. This is why only this segmentation is provided in the Appendices.

## 2.4 Machine learning and Alzheimer

As previously stated, Alzheimer's disease is diagnosed using a range of complementary procedures and examinations. The fact that the number of characteristics to be studied is so large creates challenges with the amount of data to analyze. Using automated technologies to assist medical research with data analysis may improve the productivity and significantly increase the throughput of this area of study. This is why several attempts have been made to analyze data related to Alzheimer's disease using machine learning methods. These approaches may have a variety of objectives, such as differentiating healthy patients from demented patients based on a variety of characteristics or construct a regression to predict a human-readable score. Their training would be based on a list of instances, which would qualify the algorithm as supervised learning. This is not always the case for other projects, such as clustering patients based on shared features discovered by a machine learning system.

### 2.4.1 Systematic literature review

A systematic literature review over machine learning usage in Alzheimer's disease research has been published in 2021 by Kumar et al.[40]. This review suggests that the number of research articles using machine learning-based analysis for AD dementia modeling has increased significantly during the last five years. The authors analyzed 64 pertinent papers in their systematic literature review and findings indicate that most of the previous research has been on predicting the evolution of Alzheimer disease dementia using publically accessible datasets that include neuroimaging and clinical data *"(neurobehavioral status exam scores, patient demographics, neuroimaging data, and laboratory test values)"*[40]. This will be the case in this project where, neuroimaging data will be coupled with PET data as well as demographics in order to predict cognitive assessment scores.

### 2.4.2 Determine region shape differences

In 2007, Li et al. published a an article[46] describing a study using machine learning methods for Alzheimer's disease research. The purpose of this research is to use machine learning to assess important AD-associated regional alterations in the hippocampus. Since Alzheimer disease (AD) is a neurodegenerative illness, the hippocampus is especially sensitive to damage during the early stages of Alzheimer's disease.
The method used was the following: *"High-resolution MR images were acquired from 19 patients with AD and 20 age- and sex-matched healthy control subjects. Regional changes of bilateral hippocampi were characterized using computational anatomic mapping methods. A feature selection method for support vector machine and leave-1-out cross-validation was introduced to determine regional shape differences that minimized the error rate in the datasets."*[46]
Recursive feature selection, support vector machine, as well as leave one out cross-validation will be used in this project and will be explained further in this paper. The conclusion of this paper is that machine learning approaches can identify subtle and spatially complicated deformation patterns in the hippocampus between patients with Alzheimer's disease and healthy control participants.

### 2.4.3 Classify patients

In 2020, Neelaveni and Devasana submitted a relating the use of machine learning forprediction in the field of Alzheimer's disease research. The study described an attempt to classify patients as MCI (mild cognitive impairments) or Demented (from Alzheimer's disease). Their dataset was made of the MMSE (Mini Mental State Examination) score, the age of the patients, their ApoE (*"The APOE gene provides instructions for making a protein called apolipoprotein E"*[9], stringly connected to the quanitiy of TAU protein produced.), gender, and MRI data.
Although the objective is not the same as the one set for this project, the data structure used for the prediciton is very similar. Their outcome was encouraging and confirms that the kind of structure we are going to use for this project should be correct.

# Part II

# Data and Machine learning

# Chapter 3

# Data and database design

The department's study on Alzheimer's disease is a collaborative effort. As explained in the State of the art, research datasets are composed of a variety of distinct types of information, each with a distinct function. This joint effort necessitates that diverse individuals be able to interact and exchange their data in a systematic and practical manner.

An organized, supervised, and simple-to-use framework allowing to share information would boost each researcher's efficiency and production. Various systems have been implemented to date to handle this dataflow until now. After all, a straightforward method for sharing data is to directly reach to the person responsible for the collection of information you would be looking for. This necessitated that both parties be accessible simultaneously. Second, the provider is required to make the data accessible on a timely basis. This implies substantial cooperation on the part of each participant, as well as substantial availability. However, team members' availability is not unlimited. This is why the first proposed project will focus on simplifying data exchange in order to save the most time possible.

The initial phase of this project will focus on developing a centralized database and an easy-to-use interface for researchers to interact with it. More precisely, the system will be based on a relational database, this will be detailed in further depth later. The researchers need a way of interacting with the database without having to learn SQL, which is why the interface has been designed on top of the it.

## 3.1 Overview of the needs

It is intended that this section of the project allows researchers to retrieve all of the evaluations, scans, and examinations that the patient has completed. In order to do this, each patient must be assigned a unique identifier, and a table for each kind of information must be established. In addition, it is required that for each assessment, scan, and test that a date be supplied in order to distinguish between two identical assessments, scans, and examinations that were done at two separate periods of time.

This database will group data studies that has been conducted at different times. This brings an issue for the identification of the patients. Patient number 1 from study

number 1 does not match patient number 1 from study number 2. A global id has to be set for every patient in every study.

## 3.2 Structure

In order to have a well built schema and architecture for the databse, one must specify the structure in which it will be built. First we will talk about the primary keys which is a combination of fields from a record allowing to identify every reord in each table.

After this, an explanation of the foreign keys will be given as it is the link between the different tables when referring to data from one table to another.

### 3.2.1 Primary keys

The primary key for the majority of the data will be a combination of id and date. However, for some types of data, an exception may be made. The first kind is examinations that produce multiple lines of results. Volumes from both the left and right hemispheres may be obtained using MRI data, for example. The main key in this instance would be the id, date, and hemisphere from whence the data are taken.

The storing of patient profiles is, of course, an exception. An id must only match on one single individual. This is why the first name, last name, date of birth are the primary keys for that kind of table. It is considered that the likelihood of two people having the same name and being born on the same day is nil. It should be noted that there are two additional constraints on uniqueness in this table. The first requirement is that the id column must only include unique identifiers (ids). As well, this is true for the administrative number field, which contains the administrative number assigned by the hospital in order to identify the patient. This number is unique, but it is not completely anonymous, which is why a new random id was chosen for this application rather than using the existing one.

### 3.2.2 Foreign keys

An examination result must always be linked to a patient profile. This is why a foreign key in every result tables is the id, which must match the id of a registered patient. Again, having a global profile and id for a patient instead of an id for the study currently conducted allows to store data for a patient that would already be registered from a previous study.

## 3.3 Technology

As explained above, the purpose of this database is to centralize data and group results by patients. This is why the "JOIN" functionality in SQL is the most important functionality needed when considering the kind of database management system (DBMS) to be used. Therefore, NoSQL DBMS such as document stores do not fit the needs for this project. An overview of the main differences between SQL and NoSQL can be

found in Figure 3.1. A relational database is therefor the best choice. However, there exist many different SQL DBMS.

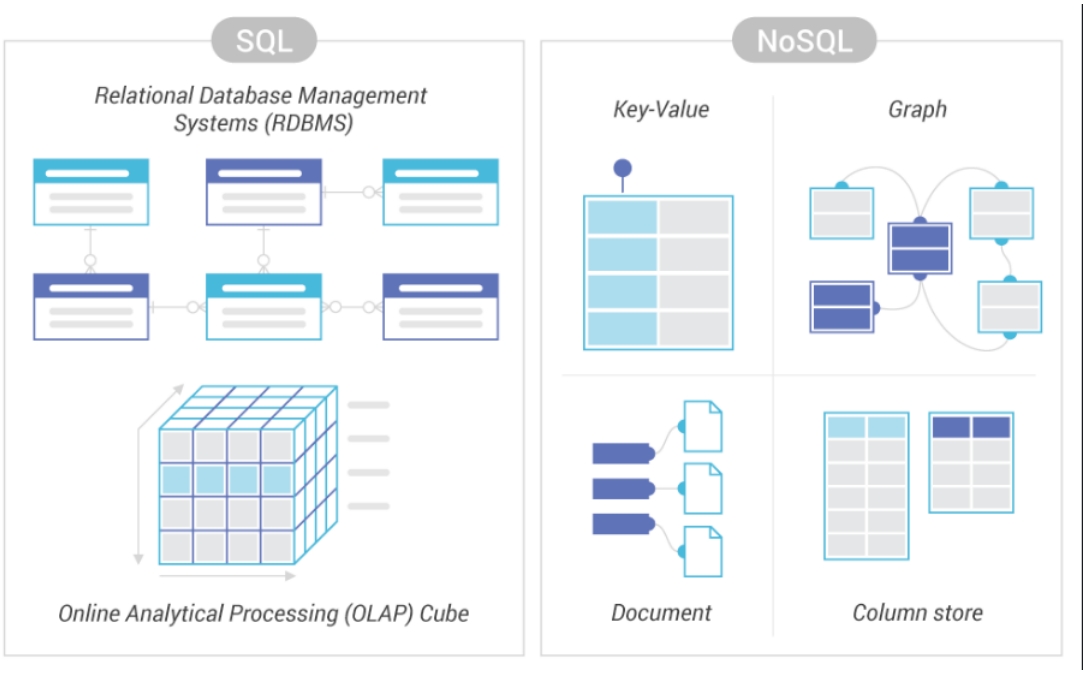



Figure 3.1: Diagram depicting the key differences between SQL database and NoSQL databases. SQL (relational) databases are built around tables linked to each other by keys. NoSQL on the other hand are present in various ways: Key-Value, Graph, Document, Column store are alternative ways to store data. [25]

### 3.3.1 Relational database management system analysis

Considering multiple relational database management systems (RDBMS) goes beyond the basic comparison performed previously between SQL and NoSQL. In fact SQL being the language in which the queries will be performed, the RDBMS will define the technology through which these SQL queries will be performed.

In this subsection, a three RDBMS will be compared in order to consider the most appropriate one regarding the needs of the projects. That is SQLite, PostgreSQL, and MySQL. The article *"SQLite vs MySQL vs PostgreSQL: A Comparison Of Relational Database Management Systems"*[71] was really informative and greatly helped distinguish the advantages and drawbacks of each of these RDBMS.

#### SQLite

SQLite is a simple, easy-to-use, and portable database. The fact that it is practically ready to use "right out of the box" makes it highly appealing. Furthermore, there is no setup for the users or the server. Its portability and usefulness are due to the fact that it is only kept in one file and requires an access to it for read/write operations. This brings up the first disadvantage of SQLite: concurrency. SQLite does not support concurrent

access to the same file. This is the fundamental problem with our application, and it is the primary reason why it will not be employed in it.

The second concern is the lack of user management. Managing users by granting them various privileges may not seem to be necessary at first, but it may become quite complicated when we wish to add new features (and therefore a new user) for which the permissions should be different.

#### PostgreSQL

PostgreSQL is a commonly used RDBMS, it is open and conforms to SQL standards. It is pushed forward by a strong community, despite having lower popularity than MySQL. Additionally, PostgreSQL allocates 10MB of RAM for each new process, which might be an issue if the application is subjected to a high volume of requests. However, the size of our application implies that it will not be managing a high volume of requests, and hence this should not be a problem.

#### MySQL

MySQL is a relational database management system (RDBMS) that is well-known for its stability, simplicity of use, and security above SQLite. Indeed, MySQL enables you to secure the database by managing its users. Its popularity makes it very simple to contribute to the community that has grown up around it over the years, as well as the abundance of resources available online. MySQL is considered to be faster than PostgreSQL and SQLite because its creators chose not to add some SQL capabilities.

MySQL, like any other technology, has certain limitations. To begin, there is no support for the "FULL JOIN," which would have been really handy; this will be detailed in further detail in the Graphical interface Section 3.4. MySQL is also a *"dual-licensed"* RDBMS, which implies that some functionality are not open-source or freely available. However, since the premium features are not required for this project, this will not be a problem.

#### Final RDMBS choice

MySQL was chosen as the preferred platform for this project. While MySQL is quite similar to PostgreSQL, it has a larger community and documentation. The choice was made to prioritize information availability of information above complete open-source and a broader variety of functions.

## 3.4 Graphical user interface

Given that the majority of users of this application will not be computer scientists. It is unlikely that they will know how to address a SQL query. This is why, on top of the database, a graphical user interface (GUI) has been constructed. This section contains as a description of each functionality developed. A global overview of the web page can be found in the Appendices (see Section 7.1).

### 3.4.1 Query generator

An SQL query can be created using this first functionality, which is intended to aid the user in the process. The user has the ability to join data from multiple tables, which will be joined naturally ("INNER JOIN") based on their unique id. This feature is really implemented as a "WITH... AS... SELECT" statement, in which each "sub-query" is joined with the others in sequence. As a consequence, each "sub-query" must be constructed one at a time. The procedure in order to create a query is the following:

1. Select the class:
   This is a particularity of the implementation for the application. Tables are divided into Classes to make them easier to use. To access a list of accessible tables, one must first pick the class in which the table has been classified before being presented with a list of available tables. This is accomplished via the use of a table that holds a list of the tables and assigns one class to each of the tables. The tables that are presented after the user has selected a class are, in this case, the tables for which the class matches the selection. The fact that a table exists in the database but isn't mentioned in the "class" table provides another benefit in that it enables you to conceal tables from view by users. This is done in the event that we do not intend for them to make use of the tables in question.

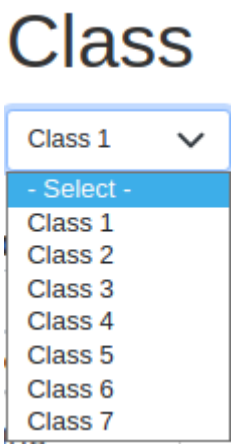



Figure 3.2: Class component

2. Select the table:
   The "Fields" banner will be automatically updated as a result of the class selection. Remember that you may switch between tables at any time without having to re-select the associated class as long as the table in question is in the same class as the one that was previously chosen. However, if you wished to use a table from a different class for your second sub-query, altering the class (see Fig 3.2) would immediately reset the Tables banner as well as the fields banner.

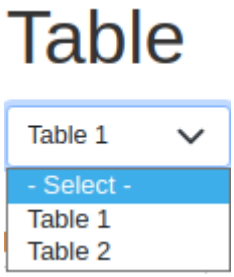



Figure 3.3: Table component

3. Select the desired fields:
   This banner provides the user with the ability to choose or deselect any desired or undesirable field. It is important to note that it is not feasible to unselect either the "id" column or the "Date" field since both are required for the query we are attempting to construct.

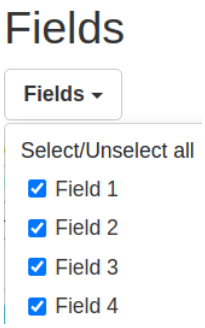



Figure 3.4: Fields component

4. Name your sub-query with a tag:

   As part of the whole procedure, the user is given the option to name his or her sub-query, which will be added to the list of sub-queries on the left side of the screen during the entire process.


Figure 3.5: Tag component

5. Add conditions to your query:

   The user has the option of including up to six conditions in his or her search query. Each of these requirements will be separated by the conjunction "AND". The construction gets done by:

   (a) Choosing the field on which one would like to apply the condition.

   (b) Choose the kind of comparator: "Greater than", "Equal to", "Less than".

   (c) Choose the value to which the field must be compared.

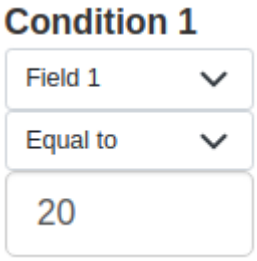



Figure 3.6: Condition component

6. Add conditions on dates:

   Two fields "N days before" and "N days after" are available to the user in order to get the closest possible exams in time for each patient.
   Let us consider the following case: You select all the results for the "MMSE" assessment, then you proceed to select the results for some MRI data. The reference table will be MMSE because you selected it at first. The tables before being joined are shown in Figure 3.8.

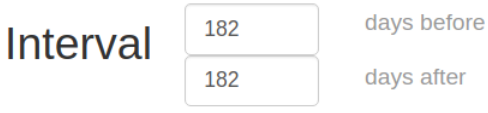



Figure 3.7: Interval component

| id | Cognitive (score) | Date_Cognitive |
|---|---|---|
| 1 | 11 | 03/03/2022 |
| 1 | 12 | 01/05/2020 |
| 2 | 22 | 04/05/2022 |
| 2 | 23 | 03/01/2020 |

| id | MRI (score) | Date_MRI |
|---|---|---|
| 1 | 4 | 01/01/2022 |
| 2 | 5 | 01/02/2022 |

Figure 3.8: Tables selected before being joined.

(a) "N days before"=1000 and "N days after"=1000 implies that every assessment joined to the "MMSE" table should have been performed within 1000 days before and 1000 days after the MMSE. The result is show in Figure 3.9. The "Diff" column indicates the number of days between the assessments.

| id | Cognitive (score) | Date_Cognitive | MRI (score) | Date_MRI | Diff_MRI_Cognitive |
|---|---|---|---|---|---|
| 1 | 11 | 03/03/2022 | 4 | 01/01/2022 | 61 |
| 1 | 12 | 01/05/2020 | 4 | 01/01/2022 | -610 |
| 2 | 22 | 04/05/2022 | 5 | 01/02/2022 | 92 |
| 2 | 23 | 03/01/2020 | 5 | 01/02/2022 | -760 |

Figure 3.9: Tables joined with "N days before"=1000 and "N days after"=1000

(b) "N days before"=0 and "N days after"=100 implies that every assessment joined to the "MMSE" table should have been performed within 0 days before and 100 days after the MMSE. The result is show in Figure 3.10. The "Diff" column indicates the number of days between the assessments.

| id | Cognitive (score) | Date_Cognitive | MRI (score) | Date_MRI | Diff_MRI_Cognitive |
|---|---|---|---|---|---|
| 1 | 11 | 03/03/2022 | 4 | 01/01/2022 | 61 |
| 2 | 22 | 04/05/2022 | 5 | 01/02/2022 | 92 |

Figure 3.10: Tables joined with "N days before"=0 and "N days after"=100

7. Confirm your query and execute by pressing the "Add" button.

8. Download or save query:
By clicking on download, you will have the results of the global query stored as a ".csv" file, which will allow you to open it in Excel or handle it in any other way that you want. On the right-hand side of this button is the "Save query" button, which will create a file with the global SQL query, allowing the user to re-execute the same query later if any new data has been submitted.


Figure 3.11: Download and save query components

### 3.4.2 SQL querying

This page enables the user to re-execute queries that have been previously stored, hence avoiding the need to repeat the same actions over and over again if you simply wish to get new information.

### 3.4.3 Create patient

Prior to entering assessment results into the database, it is necessary to ensure that the patient to whom the data refers already exists in the database. In order to do this, a page has been created in order to gather all of the necessary information and continue with the creation of a profile for each patient. Because each patient is considered unique, he or she is assigned a unique identification number, which must be used when uploading new data.

One difficulty here is ensuring that the new patient does not have the same name as a previously generated patient. The administrative number must be given in order to take use of the unique constraint of the patients table and determine whether the newly provided administrative number already corresponds to an existing administrative number in the system. If this is an administrative number in the table, the patient is either updated if the provided first name, last name, and date of birth match an existing primary key in the table or created if the given first name, last name, and date of birth do not match an existing primary key in the table. An id is generated automatically to match to this profile.

### 3.4.4 Upload data

In order to submit data, the user must first create a ".csv" file containing all of the information that will be uploaded beforehand. Each assessment result must be provided with the ability to identify the individual to whom the result belongs as well as the date on which the assessment was carried out. In order to do this, the date as well as the administrative number must be included in this file.
To identify the patient in this file, it was decided to use the administrative number rather than the random id since the people in charge of uploading the new data were not familiar with working with the random id. Furthermore, the fact that this id must remain anonymous implies that it should not be coupled with the patient's name or any other information that may be used to identify the patient anywhere else in local files. Thus, while uploading, a mapping is automatically carried out and administrative numbers are replaced with their corresponding randomly generated Ids. If the administrative number does not match any existing ones, a notification is presented, and the user is required to build a patient profile for the administrative number in question.

### 3.4.5 Carry forward

This functionality serves a purpose that is quite similar to that of the "Query generator"3.4.1. The distinction here is that instead of picking every evaluation result that is available for each patient, (See Figure 3.12), the first selection only outputs one line

per patient. This line is chosen under the constraint that it represents the earliest time this evaluation was performed for each subject in the sample. (See Figure 3.13)

| id | Cognitive | Date_Cognitive |
|---|---|---|
| 1 | 11 | 03/03/2022 |
| 1 | 12 | 01/05/2020 |
| 2 | 22 | 04/05/2022 |
| 2 | 23 | 03/01/2020 |

Figure 3.12: Base selection without filter

| id | Cognitive | Date_Cognitive |
|---|---|---|
| 1 | 12 | 01/05/2020 |
| 2 | 23 | 03/01/2020 |

Figure 3.13: Base selection after carry forward filter

Following that, a second table may be selected. In this case, no such filtering will be applied to the selection (See Figure 3.14). The first selection is simply combined with the second choice in its basic form. So, one variable will stay constant over time, while another will evolve with time (See Figure 3.15).

| id | MRI | Date_MRI |
|---|---|---|
| 1 | 44 | 01/01/2020 |
| 1 | 45 | 01/01/2022 |
| 2 | 43 | 01/02/2020 |
| 2 | 42 | 01/02/2022 |

Figure 3.14: Second selection without join

| id | Cognitive | Date_Cognitive | MRI | Date_MRI |
|---|---|---|---|---|
| 1 | 12 | 01/05/2020 | 44 | 01/01/2020 |
| 1 | 12 | 01/05/2020 | 45 | 01/01/2022 |
| 2 | 23 | 03/01/2020 | 43 | 01/02/2020 |
| 2 | 23 | 03/01/2020 | 42 | 01/02/2022 |

Figure 3.15: Final result of carry forward query

### 3.4.6 Cross sectional

This functionality allows to join multiple tables, again like the "Query generator" functionality (See subsection 3.4.1). The particularity of this functionality is that it allows to join tables and filter them based on the minimum difference of days between the two dates. This means that neither of both tables will have the earliest assessment selected on purpose. If a closer match is found for a late date, the late date will be selected, unlike "Carry forward" (See subsection 3.4.5). Let us illustrate this with an example. Consider Figure 3.12 Which would be the first selection and Figure 3.14 as our second selection. The intermediate result of the cross sectional operation would be the natural inner join of both tables with a column representing the number of days between the two dates. This intermediate result is shown in Figure 3.16. The selection is then simply made by taking the line with the minimum difference of days between the two assessments. The final result of this functionality can seen in Figure 3.17.

| id | Cognitive | Date_Cognitive | MRI | Date_MRI | Diff_Cognitive_MRI |
|---|---|---|---|---|---|
| 1 | 11 | 03/03/2022 | 44 | 01/01/2020 | 792 |
| 1 | 12 | 01/05/2020 | 44 | 01/01/2020 | 122 |
| 1 | 11 | 03/03/2022 | 45 | 01/01/2022 | 61 |
| 1 | 12 | 01/05/2020 | 45 | 01/01/2022 | 611 |
| 2 | 22 | 04/05/2022 | 43 | 01/02/2020 | 824 |
| 2 | 23 | 03/01/2020 | 43 | 01/02/2020 | 30 |
| 2 | 22 | 04/05/2022 | 42 | 01/02/2022 | 93 |
| 2 | 23 | 03/01/2020 | 42 | 01/02/2022 | 761 |

Figure 3.16: Cross sectional: intermediate result

| id | Cognitive | Date_Cognitive | MRI | Date_MRI |
|---|---|---|---|---|
| 1 | 11 | 03/03/2022 | 45 | 01/01/2022 |
| 2 | 23 | 03/01/2020 | 43 | 01/02/2020 |

Figure 3.17: Cross sectional: final result

## 3.5 Web application: architecture and technical aspects

### 3.5.1 Architecture

The architecture of an application is a representation of the general method through which it will be built. Every aspect of the project, from the way each functionality will be separated from another to the way the front-end (graphical aspect) will interact with the back-end, will be carefully considered. The choice of architecture raises the issue of what technology should be used in conjunction with it.

Some architectural designs are more suited for certain uses than others, yet they may need a significant amount of additional effort to put in place. Another architecture may be less appropriate, but it may be simpler to deploy, maintain, and enhance. And other architectural designs are plain inappropriate for the situation.

The developer of this application must thus assess his or her own capabilities, as well as the amount of time available and the environment in which the application should be developed. However, although time and skills may be improved or optimized, the environment limitation is such that the application's design will have to be modified in order to meet the environment. It is not possible to think about it the other way around.

There are two well known ways to design an architecture for an application: Monolithic and Micro-services (See Figure 3.18). While monolithic has been around for a long time, micro-service becomes more and more popular due to the fact that it uses containers, thus has the advantage of avoiding scalability issues and is very well fitted for a deployment on cloud based services such as Amazon Web Service.

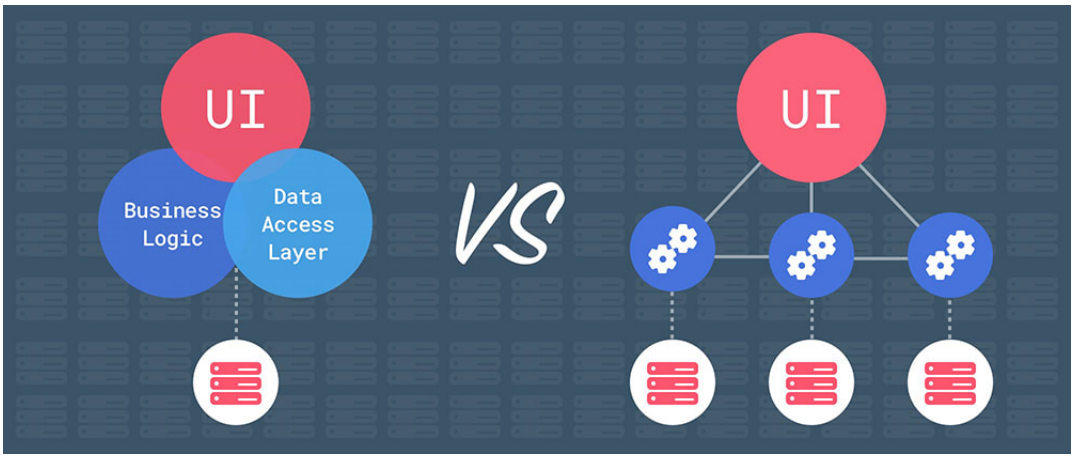



Figure 3.18: Monolithic vs Micro-services architecture [50]

### Monolithic

The monolithic architecture is based on the principle that everything get developed together in one **monolith**.[49] To have a working application, we need:

1. A database composed of several tables, often inside a RDBMS.
2. A graphical user interface.
3. A server that manages HTTP requests, executes domain-specific logic, retrieves and updates data from the database.[49]

The fact that everything gets built in one block brings some advantages:

1. Easy to understand structure:
   If one must continue the development, the new developer must understand the architecture but it is not very complicated to understand. However, this might not be the case for a large project where multiple developers work in different fashions on the same project which can look a bit inconsistent.

2. Easy to deploy:
   There is one folder, and only one part to deploy. There is no need to manage a lot of different environments like in the micro-services architecture which can save time and make it easier for system engineer to manage their servers.

3. Quick to launch a prototype:
   The architecture does not need to be split for each service (like explained for the micro-services). It is fairly quick to define the connection between the back-end and front-end as well as the database.

However, the monolithic architecture has some downsides:

1. Can get "messy":
   Not having to define a very strict structure can lead to messy code. Which is not

always easy to interpret afterwards if ever a new developer needed to work on the project.

2. Interpretability:
   As the application grows, the services multiply and it can become quite difficult to manage multiple services because of the lack of structure.

3. Not easily scalable:
   Consider a developer wanted to update or add a feature to the existing website. This developer would have to deploy the whole application entirely because services are not separated.

**Micro-services**

The micro-services architecture was first introduced in 2011[51], which is a long time after the monolithic architecture was initially introduced. Micro-services, on the other hand, provide several benefits and features that help to overcome the problems associated with monolithic design. The advantages of micro-services are the following:

1. Clear structure and architecture:
   Regardless of how complex your application's design is, the structure should be clear. Because of the way this, if it is designed in micro-services, it should be highly obvious and easy to understand so a new architect would have no difficulties understanding the way this application is supposed to work.

2. Easy to work on:
   In contrast to traditional architecture and services, micro-services do not need a developer to be familiar with the whole architecture and services in order to work on the application. Micro-services, on the other hand, are designed in such a manner that you, as a developer, are not required to understand the whole program. You have the option of creating your own service and integrating it into the current framework. All that is required is that you modify your service in order for it to be able to communicate with others.

3. Easily scalable:
   Adding a feature does not require for the developer to rebuild the whole application a re-deploy it again. Like previously stated, services are built and working independently and can be deployed on their own. Furthermore, the term "scalable" refers to the fact that certain of your services may be built to be horizontally scalable. This implies that, if a service is configured to do so, it may replicate its instance when under a heavy load of traffic. This may be very beneficial for applications that experience significant peak loads but do not need many instances of the service to be active at all times.

While micro-services have a number of interesting benefits, they also have some significant disadvantages[1]:

1. Developers vs DevOps:
   Developers have usually worked on their code using technologies that are very well known to them (java, C, javascript, ...). As long as another person is in charge of the infrastructure management of the system in which the application is meant to be running, this is perfectly fine. If, on the other hand, an application is divided into micro-services, each developer must be aware of, and may even be required to configure, the environment in which the service on which he or she is now working will be deployed. Because each micro-service is developed independently of the others, each one has its own environment that must be configured. The developer will no longer be referred to as a developer, but rather as a DevOps engineer (Development and operations). This implies that developers will need to broaden their skills to include system administration, which is not always possible or cost-effective.

2. Communication issues:
   Because one micro-service may interact with the others, this architecture was designed with the hypothesis that connection is stable and available. But keep in mind that separate instances, other computers, and even different networks might be used to host distinct micro-services. This might cause a variety of problems when it comes to connecting to certain services. Bad connections or a server that is unavailable might make it very difficult to operate with some programs.

3. Testing:
   Testing and debugging systems that are constructed using micro-services might be more difficult than testing and debugging apps that are designed in a monolithic way. As a result of the possibility that each micro-service may use a distinct technology, anytime the output of testing for one micro-service does not meet expectations, the developer would be required to investigate the functioning of the service itself. As a result, debugging a micro-service application is somewhat more difficult than debugging a monolithic program.

**Final choice**

Given that the application is relatively small for the time being, it is likely that a monolithic design will be sufficient. In the event that the program was to be continued later and some new features were to be added, a micro-services architecture might be used to establish a precise and strict structure from the beginning.

Because of the number of users and the frequency with which the application is used, the issues of load balancing and horizontal scalability do not need to be addressed for the time being for this project.

Finally, remember that the only condition which is non-negotiable is the environment in which the application would be deployed. The environment considered for this project does not support the softwares required for micro-service architecture. This is why the application has been built in a monolithic architecture. However, a description of how

the application should have been built in micro service is provided in the event that the project would be redesigned later (See subsection 3.5.3).

### 3.5.2 Technical aspects

This section will provide an overview of the various technologies employed, including the languages in which the application was created.

#### HTML - Css

First and foremost, HTML - Css is being used for the front-end of a website. Actually, HTML and Css are two separate technologies, with HTML being used to produce (static) information on a web page and Css being used to enhance the presentation of that material. In today's world, these technologies are seldom employed on their own. As a result, HTML-Css will be referred to as a single technology for the remainder of this section.

#### JavaScript

When it comes to making the front-end component more interactive, HTML-CSS is not enough. Javascript enables for the creation of animated web pages, among other things, in order to make the user's experience more pleasant. Furthermore, Javascript enables for the transmission of requests to PHP files in order to render the information to be shown.

#### PHP

In addition to HTML-Css and Javascript, PHP is the next programming language required for this project. This language will be utilized to administer the back end of the application. PHP is a programming language that was first developed in 1994, but it is still in widespread usage today and is considered a must-know for everyone working in the web development industry. As previously said, HTML is a programming language that allows you to display static material on a web page. PHP provides the ability to produce HTML code for display. This one enables the application to be more dynamic. As an illustration:
Logging in to a page that is exclusively built of HTML-Css would neither allow for the validation of the user's password and login, nor would it allow for the modification of the content of the homepage.

### 3.5.3 Improvements

After analyzing the application's development and scope, one might conclude that it needs certain functionality or that its architecture could be improved. It is essential to remember that no program or application is final, since it is constantly subject to improvement; the same holds true for this application. Therefore, a summary of the architectural improvements will be provided below.

### Micro-services prototype

The architecture for a micro-services architecture prototype has been built in the event that the project has to be updated with a micro-services design. This will make it easier to start a new project in the future.
As seen in Figure 3.19, this design is based on the use of APIs, namely an API gateway, to accomplish its goals. An API is a specified set of rulesand instructions that enable users and applications to communicate with – and get data from – a particular application or micro-service.[64] Whenever you construct a microservices-based application, APIs establish the rules that restrict and authorize any actions between isolated services.[64]

The rules and restrictions set by the developer are managed using REST APIs. There are several commands a developer can use in order to authorize or forbid an action:

- PUT
- POST
- DELETE
- GET
- PATCH

These rules will be used accordingly to the actions that should be allowed or not. Like show in Figure 3.19, the design will make use of an API gateway in order to route each and every query to the appropriate resource. The question arriving from the user interface (front end), as well as the query going back from the service, should both follow the format that has been specified in the specification. If this is not the case, the gateway will not continue with the transmission of the result. Because of this quality, it makes no difference what sort of technology is utilized in the service as long as the output is delivered in the format that was anticipated. This is what allows for the architecture to be both flexible and changeable. For each service, a Docker[73] container providing the environment required for it to function effectively would be used. Ports from this Docker[73] container would be left open in order for the service to interact with one another. It is necessary for each service to have its own database in order for each service to remain independent of the others. Furthermore, this opens the door to the possibility of using various types of databases for different types of services.

The last service represented in Figure 3.19 could be a service to retrieve raw data such as document, images ... from a databse that would support this kind of files unlike the SQL database actively in use.

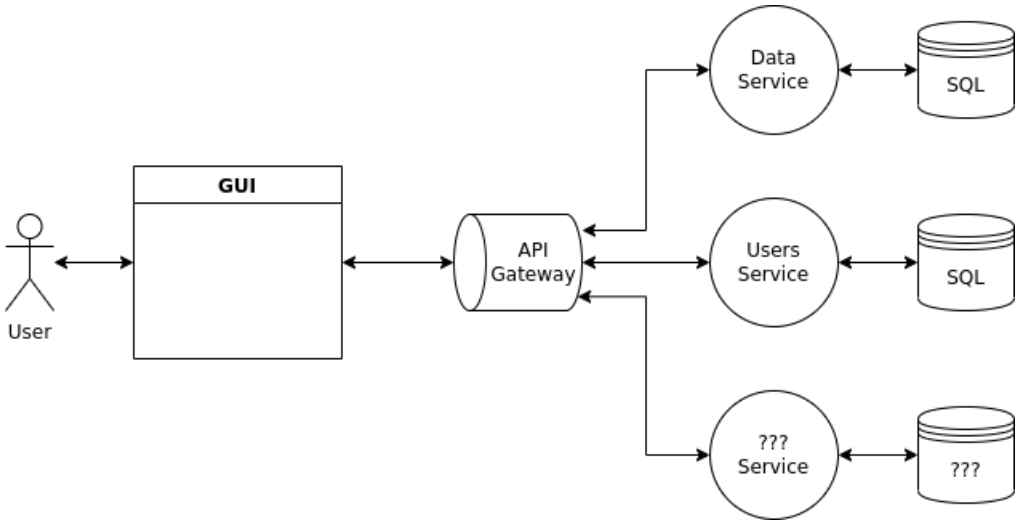



Figure 3.19: Alternative architecture based on micro-service principle: A graphical user interface connected to an API gateway able to redirect to different services, each responsible for a different aspect of the project. This architecture is ready to welcome new services because of its flexibility.

# Chapter 4

# Data analysis and machine learning

The design and development of a database as described in Section 3 has allowed to centralize all the information needed in order to perform machine learning. This gives the opportunity to analyse all the centralized data and use it to train machine learning models.
In this chapter, the data analysis and the machine learning approaches (represented in Figure 4.1) will be discussed, however, the process in which this machine learning project has been developed slightly varies from the classical approach. First, an introduction will be presented in order to recall the objective end challenges of this project. A dataset description is provided afterwards in order to discuss the role of each group of features in this project. The pre-process step will also be discussed to have a better understanding of the way the data has been treated and observed before entering the modeling part which will explain the different kinds of models assessed to try and reach the objective. This will be followed by the recursive feature selection section. The fifth part of this chapter will be about the results, discussion and possible improvements. And finally, the deployment of the final pipeline will be explained as the final step of this project.

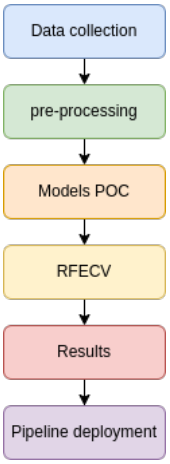



Figure 4.1: Machine learning process

## 4.1 Introduction

As stated in the introduction to this work, the goal of this study is to determine whether or not machine learning techniques may be of any use in the investigation of Alzheimer's disease and related disorders. In the case of Alzheimer's illness, the use of machine learning has already shown some intriguing outcomes (See subsection 2.4.2 and 2.4.3). In this section of the project, we will attempt to determine the effectiveness of several regression models for a given regression task. As previously stated in subsection 2.1, composite scores have been developed in order to assess the overall health of a patient in regard to a faculty. The composite scores are defined in four different types: executive function, memory, language, and visuo-spatial. It might be possible to use

machine learning to anticipate the decline of a patient in one or more faculties in this situation. It would be great if this prediction could be made when a patient first enrolls in the study and once he/she has passed all of the examinations and evaluations. This initial piece of information about the patient is referred to as the "Baseline" information. Based on the baseline data, we could be able to forecast cognitive decline and so provide a better follow-up treatment.

This problem will be well defined after a first dataset has been created, which will allow various simple machine learning models to be generated. These will be referred to as "Benchmark" predictors. Following the construction of this dataset, some basic processing will be performed. Following pre-processing of the dataset, the basic models will be created, followed by recursive feature selection. Finally, the best model will be selected, and its hyper-parameters will be tuned in order to increase its accuracy.

## 4.2 Dataset description

A dataset made out of two parts will be built for this project. The first part is made out of few features for which the literature has shown for sure that these have an impact on dementia in Alzheimer's disease. A second part will be built with additional features. Feature selection will be performed on this whole dataset.

At this point, the project achieves its first limitation. The quantity of accessible data is the greatest challenge to the project. For each feature that may be added to the general dataset, there is a chance that not all patients included in the new feature are also present in the general dataset. This will result in fewer rows (i.e., instances) in the dataset. As is often the case with machine learning, the most significant aspect of the project is the kind and diversity of the data. However, feature selection cannot be accomplished effectively if the dataset is insufficient. There is a compromise to be made as a consequence but this issue will be resolved in Section 4.7 where an explanation of the future of this project will be explained.

### 4.2.1 Base Dataset

#### Hippocampal volume

There is evidence that hippocampus volume may help identify patients with Alzheimer's disease from healthy individuals, this measure has thus been included in the benchmark dataset. The identification of patients with Alzheimer's disease thanks the hippocampus volume has been described in a paper published in 2003 by Wang et al.[74]

MRI's ability to follow morphological brain changes over time is demonstrated by the hippocampal decline and accelerated decline in mild cognitive impairment and Alzheimer's disease patients over 6 and 12 months, respectively according to Schuff et al., in "MRI of hippocampal volume loss in early Alzheimer's disease in relation to ApoE genotype and biomarkers" [67].

This volume will be corrected according to the intracranial volume (ICV) of the patient. In an article published by BMC Bioinformatics, *"A practical guideline for*

*intracranial volume estimation in patients with Alzheimer's disease"*[66] it was stated that in research that depend on morphological aspects of the brain, ICV constancy over time makes it a great tool for *"correction of head size variation"*[66] between patients due to its consistency over time. When researching progressive neurodegenerative brain illnesses such as Alzheimer's disease, aging, and cognitive decline, ICV, along with age and gender, are reported as variables to account for regression analysis.

**Age**

Because Alzheimer's disease is associated with advancing age, it is critical that we include this piece of information in the dataset in the hopes of improving the accuracy of our model. It is not necessary to demonstrate the significance of age in Alzheimer's disease since it is well acknowledged that they are both connected.

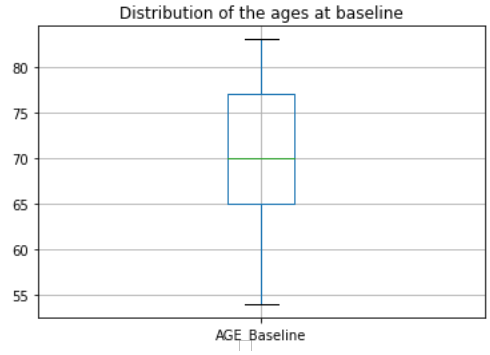



Figure 4.2: Age distribution at baseline

**Standardized Uptake Values ratio: Amyloid**

Noted SUVr, the Standardized Uptake Values ratio is the most often utilized quantitative metric in positron emission tomography for evaluating Amyloid-beta tracers (PET). The importance of the amyloid-beta SUVr regarding Alzheimer's disease has been emphasized in the state of the art, the amyloid-beta SUVr was therefore added to the dataset. We could have included the tau-PET data in our dataset, but because the majority of the patients already included in this dataset do not yet have pet-TAU information, this would have resulted in a significant reduction in the size of the dataset, which would have caused problems during the training of our machine learning algorithms.

### 4.2.2 Additional data

The MRI has supplied the information that will be used to supplement the basic dataset. That is, the cortical thickness. The cortical thickness per region has been used in order to perform feature selection later on. It is important to note that for the purpose of this project, the average thickness has been computed for both hemispheres. This was a choice during this project in order to make the number of features smaller.

However, it is important to note that some cognitive functions might be lateralised and this method might make it difficult to distinguish important features from less important ones. These areas will be analyzed via the feature selection process, and it is possible that the average thicknesses of particular brain regions may disclose the influence of certain brain regions on the cognitive deficits of Alzheimer's disease patients.

The regions that will be tested can be found in the Appendices in subsection 5.1.1.

Dickerson et al. through the study "Detection of cortical thickness correlates of cognitive performance: reliability across MRI scan sessions, scanners, and field strengths", stated that when using an automated data analysis system, it is possible to reliably identify brain-behavior relationships between regional cortical thickness and cognitive performance. This explains why the choice was made to use thickness over volume or surface area for the analysis of the biomarkers.

### 4.2.3 Target

This machine learning project aims to predict a decline in neuropsychological composite score. These composite scores are based on the results of various cognitive exams. Recall that the following composite scores were used to evaluate the cognitive condition of the patient:

1. Memory
2. Executive
3. Language
4. Visuospatial

Each of these scores represents a function prone to impairment as Alzheimer's disease progresses. The decline of the patients is given per year. This gets computed like following:

$$Delta\_per\_year = \frac{Score_{T=MAX} - Score_{T=0}}{N_{years}(Date_0, Date_{MAX})}$$

Where the function $N_{years}(Date_1, Date_2)$ computes the number of years (not rounded) between two given dates.

This provides an overview of the annual drop for a specific composite score. This methods shows the linear decline of a patient for a certain composite score. It is possible that some cognitive skills are not deteriorating in a linear way, and this may be addressed by creating a new method of computation. Despite the importance of identifying this potential non-linearity, the majority of patients have been evaluated for their cognitive capabilities no more than four times, which is not necessarily sufficient to answer this question.

Consider Figure 4.3; this patient initially has shown no evidence of cognitive deterioration but subsequently had a significant drop. This demonstrates why the subject of linearity is raised, but it does not give evidence or a conclusion on the matter. Moreover, this situation is not typical, since neither every patient nor the majority exhibited this pattern. The computation method used for this project is therefor valid regarding the amount of data for which we only have 2 measures through time.

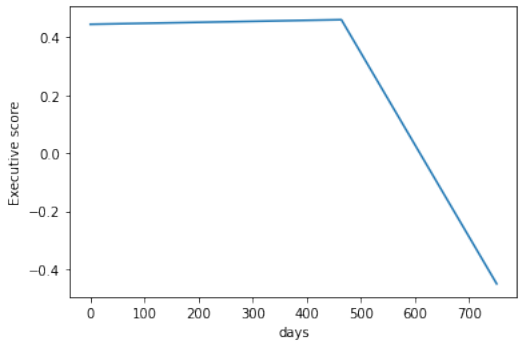



Figure 4.3: A patient's score in execution stays stable up to a sudden drop. This bring the question of linearity in cognitive decline.

If each patient had more than 2 measurements for each composite score, a non-linear regression may be used in order to define a metric to asses the decline of the patient.

Nonetheless, the method of computation used does not require each patient to have more than two measurements, which is crucial given the size of the dataset. More than likely, the size of the dataset would have been considerably decreased if the approach employed for this project required more than two measurements.

Importantly, the ideal patient to be included to this dataset would be in great health at the start of the study and would only develop dementia later on. This nevertheless may not always be the case. One may identify three cognitive states for a specific patient:

1. Healthy: Signs of cognitive decline only related to normal aging.
2. Mild cognitive impairment (MCI): State between normal aging and dementia.
3. Demented: When a patient has a deterioration in cognitive functioning that interferes with daily tasks.[2]

Some patients in our data set may already have MCI or dementia, indicating that their deterioration may differ from that of a healthy patient.

## 4.3 Pre-processing

Pre-processing is the initial phase of the data analysis part of the project. This section enables a properly structured, cleaned, and organised dataset. It is necessary for the remainder of the project since noisy, unstructured data will prevent the establishment of a machine learning architecture.

### 4.3.1 Correction of the hippocampal volume

An analysis of linear regression was used to adjust the hippocampus volume in relation to the total intracranial volume. This action does not seem to have had a significant impact on the data. Although not time-consuming, this ensured that the data would be simpler to analyze by the model as a result of the procedure.

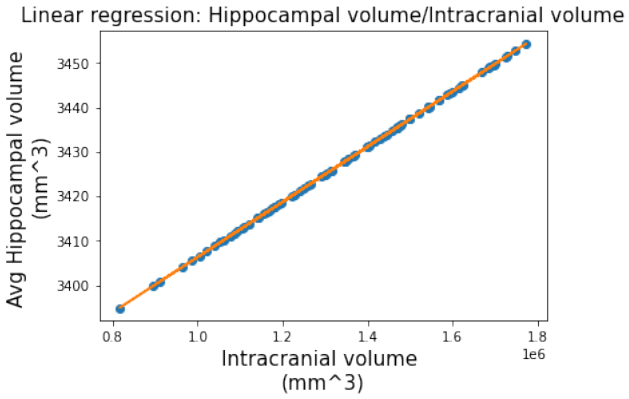



Figure 4.4: Hippocampal volume corrected based on linear regression with total intracranial volume.

### 4.3.2 Variance analysis

There has been a study of the variation in the data that has been collected. There are a number of goals for this investigation:

1. Look for 0 variance features:
   Features with a variance of zero provide no information to the model and should be avoided. In reality, having a value of 0 is an extreme example in which all of the values provided for a feature are equal. This would not assist the model in distinguishing between the multiple samples provided as training set and would thus provide no information. It is possible that training a model using characteristics that do not provide any information will have no effect on its accuracy. However, by avoiding computing over irrelevant variables during the model's training, it is possible to decrease the amount of computation time required.

   Fortunately, the data did not present features with 0 variance with means that potentially, every feature can bring some kind of information.

2. Find out the high variance variables:
   Features with a high level of variance have the potential to provide more information to the model. As a result of the same reasons why zero variance characteristics do not provide any information. If we can see which features cause the most variation, we may be able to better understand or validate the selection

of a certain feature via recursive feature selection, for example.
The top-5 features (regions) with the most variance in the cortical thicknesses were:

(a) temporal pole: 0.152285

(b) entorhinal: 0.134788

(c) parahippocampal: 0.061713

(d) Caudal anterior cingulate: 0.045746

(e) Transverse temporal: 0.036395

It is interesting to note that the cortical thickness of the regions revealing a strong variance match with the first braak stage[15] which states that the enthorinal and hippocampal regions are amongst the first to be affected by Alzheimer's disease.

### 4.3.3 Outliers

A common thing in every machine learning problem is that data is contaminated with outliers, which makes it difficult for models to develop a good predictor. Outliers must be identified and carefully evaluated in this project, as they are in many others. In truth, certain data may seem to be outliers from a statistical standpoint, but the fact that we are dealing with data related to an illness implies that some data is expected to diverge from the expected distribution. The goal would be to be able to forecast when a patient was on the edge of a great cognitive decline. In contrast, if we "fake" the data by substituting alleged "outliers" with the median value of the column, the mean, we may end up with a dataset that only contains healthy patients, which is not what we were aiming for.

As a result, it was decided to keep the data in its current state. Furthermore, the thickness data that was obtained has already been rectified and thoroughly reviewed by the person who is in responsibility of giving this information. We will assume that the information supplied is accurate and that it will not interfere with our models' ability to learn effectively.

The study of the outliers does not come to an end when the input is finished. The output we are attempting to forecast is just as important. In reality, providing any model with an unpredictable target to forecast would merely result in the model discovering a meaning behind the data that would not otherwise exist.

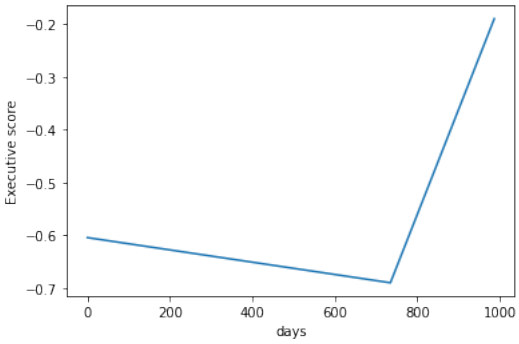



Figure 4.5: Improvements over time in Executive composite score

While working on this phase of the project, a problem with the target was discovered. Certain people with Alzheimer's disease had improvements in their cognitive abilities (see Figure 4.5), which happened for each composite test.

The target represents the evolution of a composite score in time. This is computed by taking the difference between two assessment taken by the same patient, divided by the number of years that have elapsed between these two assessments. The observation was made that some patient presented a positive evolution in some scores which was an issue for the model. The model would deduce a rule where some "weak" cortical thickness could result in an improvement in some cognitive capacities; however, it is considered very unlikely.

Multiple hypothesis were evaluated:

1. An error when encoding the data happened: After double checking, this did not seem to be the case.
2. Some of these patients had a second health condition that were cured: These patients would have supposedly improved in every composite score. This was not the case.
3. Some patients actually really performed better the second time.

The third proposition was considered the most likely. This was discussed with the person responsible for collecting this data as well as analysing it.

Some patient are likely to perform better for some assessments when taking those for the second time for multiple reasons:

1. The two assessments were not taken in the same conditions:
   Some patients might have been slightly sick for the first assessment. Might not be in a great mood for personal reasons. The time at which the assessment was taken was not the same for the two assessments (some people might perform poorly when tired for example).
2. A learning effect may occur when cognitive tests are taken multiple times by a patient [55]. This suggests that a patient who has undergone the same test twice would do better since he or she will be more used to the exam and will have a partial memory of how to pass the second time. As illustrated in Figure 4.6, it is possible that the gain in executive skills for this patient is due to the two assessments being close in time and would have produced a learning effect. This was not proven however and is given as illustration for an understanding of the learning effect.

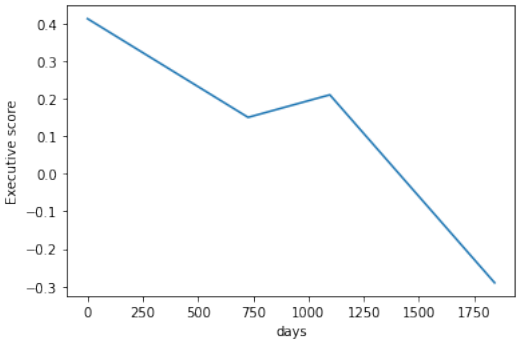



Figure 4.6: Illustration of possible learning effect: The graph shows a tendency to decline in execution. However, the performance of the patient illustrated here have improved when considering two assessments close in date. This can illustrate the fact that a patient can be subject to learning effects when the assessments gets taken multiple times in a short period of time.

As a consequence, it was determined to eliminate problematic rows where the patient's cognitive abilities would unnaturally increase since fixing the data would fake it and maybe affect the outcome.

Problematic rows are defined as such when value are above the following value:

$$\mu + 1.5 * \sigma$$

Where $\mu$ is the mean of the distribution and $\sigma$ is the standard deviation of the distribution. This method was not applied for values below

$$\mu - 1.5 * \sigma$$

since removing declining patients would not help the model predict this particular phenomenon. The Final distribution of the target for the execution score can be found alongside the original distribution in Figure 4.7. Other distributions for the composite scores can be found in Appendices 5.2.

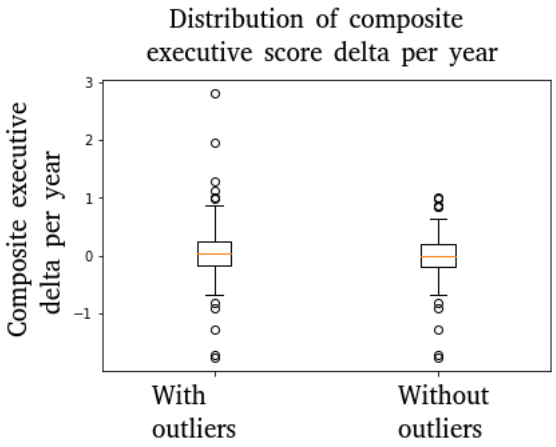



Figure 4.7: Distribution of delta composite executive score per year before and after removing outliers from the distribution

### 4.3.4 Linear and non-linear relationships

To get a better knowledge of the dataset and its effect on the goal, it is necessary to examine both linear and nonlinear relationships. A strong connection between two variables in the input data may also be extremely beneficial in evaluating the features that have been selected.

When observing two strongly connected variables, it is common practice to delete one of them in order to avoid training the model on the "same" data twice. This is frequently done in order to save time. This is not a concern in this project since the size of the dataset does not necessitate a considerable computing time. Furthermore, since the goal of the project is to have an overview of the key characteristics regardless of their relationship, omitting certain aspects from a statistical criteria may result in a loss of interpretability and information.

### Correlation

The correlation method used in this section is more formally called the *"Pearson Correlation Coefficient"*[78]:
Pearson correlation evaluates the linear connection between two given variables and has a value between -1 and 1.[78]
However, the correlation here has been taken in absolute value in order to select the top-10 most correlated thicknesses for several regions.

The first result of the correlation analysis to be shown here is the relations between the variables present in the input dataset. No relations with the target variable will be shown here. The top-10 regions for which the cortical thicknesses are correlated can be find here under:

Table 4.1: Top-10 features with highest Pearson correlation coefficient (absolute value)

| Region 1 | Region 2 | Correlation |
|---|---|---|
| Inferior parietal | Supramarginal | 0.858203 |
| Inferior parietal | Precuneus | 0.848680 |
| Caudal middle frontal | Superior frontal | 0.837264 |
| Postcentral | Supramarginal | 0.835050 |
| Paracentral | Precentral | 0.833144 |
| Inferior temporal | Middle temporal | 0.825756 |
| Postcentral | Precentral | 0.817766 |
| Precentral | Supramarginal | 0.810749 |
| Paracentral | Postcentral | 0.805490 |
| Caudal middle frontal | Pars opercularis | 0.801660 |

The amount of available data is a key source of concern for this project; a purely statistical study, due to the amount of irregularity in the data, may result in inaccurate conclusions being drawn. It is worth noting, however, that almost all of the highly related regions are situated in the same lobes. Moreover, going over the atlas supplied in Section 5.1.1 of the Appendices, you will see that the cases that are not both located in the same lobe are quite close to each other. This confirms the consistency of the input data which, in order for machine learning models to exploit the data as efficiently as possible, must be ensured.

### Mutual information[43]

Mutual information is one of the metrics that assesses how much a given random variables informs us about another.[43] It may be described as a decrease in the amount of uncertainty around a random variable as a result of learning about another random variable. When the mutual information between two random variables is 0, the variables are independent, whereas high mutual information suggests a significant decrease in uncertainty.

Described by Shannon[68] and Cover[20], the mututal information is given by the

following formula:

$$I(X;Y) = \sum_{x,y} P_{XY}(x,y) log \frac{P_{XY}(x,y)}{P_X(x)P_Y(y)} = E_{P_{XY}} log \frac{P_{XY}}{P_X P_Y}$$

With $P_{XY}(x,y)$ being the joint probability distribution for two discrete variables X and Y and where:

$$P_X(x) = \sum_y P_{XY}(x,y) \text{ AND } P_Y(y) = \sum_x P_{XY}(x,y)$$

Although the emphasis here is on discrete variables, the majority of findings obtained for discrete variables easily apply to continuous variables by simply substituting integrals for sums.[43]

Table 4.2: Top-10 features with highest mutual information (absolute value)

| Region 1 | Region 2 | Correlation |
|---|---|---|
| Caudal middle frontal | Superior frontal | 0.665129 |
| Inferior parietal | Supramarginal | 0.662540 |
| Paracentral | Precentral | 0.658358 |
| Caudal middle frontal | Supramarginal | 0.605004 |
| Inferior parietal | Precuneus | 0.601114 |
| Inferior temporal | Middle temporal | 0.595335 |
| Postcentral | Precentral | 0.566268 |
| Postcentral | Supramarginal | 0.555929 |
| Pars opercularis | Superior frontal | 0.545589 |
| Precentral | Supramarginal | 0.545093 |

When the relative cortical thicknesses of areas from the same lobe or neighboring regions are investigated, the outcomes of this mutual information analysis reveal, as expected, that areas from the same lobe or neighboring regions have a tendency to communicate information about one another. Given the borders of the areas, this was to be anticipated, since a decrease in cortical thickness in one region is very unlikely to occur without a corresponding decrease in thickness in the nearby regions.
However, there are two items from this list that need to be mentioned:

1. Caudal middle frontal and supramarginal

2. precentral and supramarginal

When inspecting the atlas given in Appendices section 5.1.1, we can observe for both of these pair of regions that they are not neighbours. The fact that these regions shared information may represent a statistical anomaly, however it is not to be excluded that these regions share functional links. This is crucial because the recursive feature selection method may consider both of these regions to be relevant, even if only one of them would make sense in the context of the problem. This may subsequently be used to help in the understanding and interpretation of the results.

## 4.4 First prototypes

The development of a first prototype for each of the several models selected will be required in order to utilize existing machine learning models for recursive feature selection.

### 4.4.1 Models

A description of each of the models that were evaluated will be provided in this subsection. As well as an example of how these models are put to use in some use cases. The scoring used to assess if the model performed well or not is the Root Mean Squared Error (RMSE). This scorer was picked since it is more easily interpretable. Indeed, when this scoring is combined with the distribution of the target, it may provide a comprehensive overview of the models' performance.

#### Linear regressor

Linear models have already been used in order to try and predict cognitive data from medical images.[39]. However, this was done with a sparse linear model which is not the case here. This linear regression is the simplest and well know linear model which will serve as a comparison for the next models.

#### Lasso regressor

Lasso regression stands for Least Absolute Shrinkage and Selection Operator regression. Lasso regression is a specific example of linear regression that makes use of shrinkage in addition to the standard linear regression method.[42] The word "shrinkage" refers to the process by which data values are compressed towards a central point, such as the mean, in order to improve efficiency. The lasso method encourages the use of sparse, basic models in research (i.e. models with fewer parameters). This particular kind of regression is well-suited for models with a high degree of multicollinearity, as well as when certain aspects of model selection, such as selecting features and parameter removal, must be carried out by computer programs.

Multicollinearity occurs when multiple variables in a model measure the same phenomena.[53] This kind of regression may be quite beneficial in light of the fact that one of the primary goals of this study is to be able to choose the greatest available features for predicting cognitive deterioration. On the other hand, if numerous variables are co-linear, some of them may be dropped, resulting in a loss of easy - to - interpret data.

This model has the aim to optimize the following objective:[41]

$$\sum_{i=1}^{n}(y_i - \sum_{j=1}^{p} x_{ij} * w_j)^2 + \alpha * \sum_{j=1}^{p} |\beta_j|$$

Where $\alpha$ is the regularization strength of the regression. Setting $\alpha$ to 0 is equivalent to an ordinary least square, solved by the Linear regression.[69]

### Ridge regressor

The ridge regression is a kind of linear regression that, like the Lasso regression, performs well when the ratio ("number of features"/"dataset size") is large or when the data has substantial multicolinearity.[65] The ridge regression creates a *"Parsimonious model"*.

A Parsimonious model[57], is some kind of model that has a *"great explanatory predictive power"*[57]. They use a small number of features, or predictor variables, to describe data.

The problem of such a model is that lowering the dimensions of the dataset by removing the least "important" variables contradicts the project's goal of highlighting the elements that have an influence on the target.

This model has the aim to optimize the following objective:[41]

$$\sum_{i=1}^{n}(y_i - \sum_{j=1}^{p} x_{ij} * w_j)^2 + \alpha * \sum_{j=1}^{p} \beta_j^2$$

Where $\alpha$ is the regularization strength of the regression.

Linear least squares loss function and l2 regularization criteria are used in this approach to solve a regression model.[70]

### Huber regressor

This linear model is a model that is designed to be more robust to outliers than a regular linear regression. In fact, the Huber Regression opts for two different functions to optimize. Depending on a factor $\epsilon$, this regression will be performed with either the squared loss function or the absolute squared function.

If $|\frac{(y-Xw)}{\sigma}| < \epsilon$: The squared loss function gets used.

If $|\frac{(y-Xw)}{\sigma}| > \epsilon$: The absolute loss function gets used.

As a consequence, $\sigma$ and $\epsilon$ need to be tuned in order for this model to perform correctly.

### Support vector regressor[7]

SVMs, or support vector machines, are popular for their use in classifying data. Despite this, the use of SVMs for regression problem is possible and Support Vector Regression is the name given to these models (SVR).

A significant advantage of SVR over linear regression is that it allows us to choose the amount of error we are willing to tolerate in our model, and it will find a suitable line (or hyperplane in higher dimensions) to match the data.

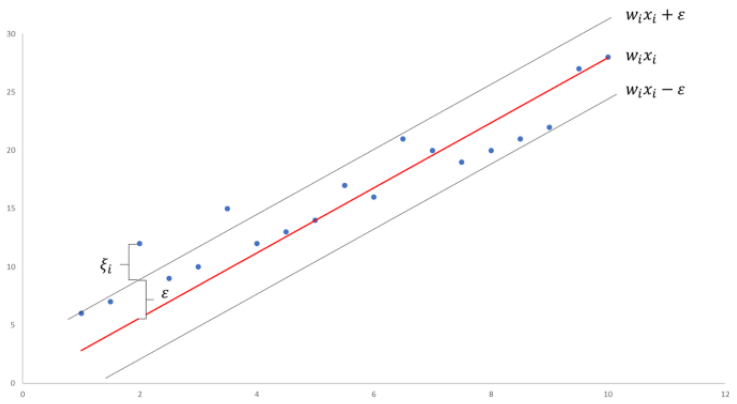



Figure 4.8: Illustrative Example of SVR with Slack Variables[7]

Unlike a classical least squared method, the SVR will make use of the l2 regularization (like the Ridge regressor) in order to to minimize the coefficient of the regression. The objective is thus to minimize:[7]

$$\frac{1}{2} * ||w||^2 + C \sum_{i=1}^{n} |\xi_i|$$

Where $\xi$ is the deviation from the maximum allowed error (See Figure 4.8). C on the other hand is a hyper-parameter that needs to be tuned in order to adjust the tolerance of the error. This is done while respecting the following constraint:[7]

$$\forall i |y_i - w_i x_i| < \epsilon + |\xi_i|$$

Therefor, $\epsilon$ needs to be tuned accordingly to allow for the best regression to be performed by the model.

In the article *"Support vector machine-based classification of Alzheimer's disease from whole-brain anatomical MRI"*[48], a support vector classifier has been used in a classification problem. The objective was to distinguish elderly control patients from patients with Alzheimer's disease. Based on MRI data, the model was able to achieve a mean accurate classification rate of 94.5 percent. This is a promising sign for our project. However, since this project is a regression problem, it does not necessarily follow that the result of the regression will be as accurate as the classification result.

### Decision tree regression

When it comes to the model that was used to execute the regression, decision trees are not employed in the traditional sense in this particular project. Decision trees, on the other hand, are employed in both the Random forest and the Gradient boosting regressor, as shown in the following example. Hence, it has been necessary to write this part in order to clarify how decision trees operate. The purpose of the following section is to show how decision trees, which are primarily trained for classification issues, may be used to regression problems.

| *Outlook* | *Temp* | *Humidity* | *Windy* | *Golf?* |
|---|---|---|---|---|
| rainy | hot | high | false | no |
| rainy | hot | high | true | no |
| overcast | hot | high | false | yes |
| sunny | mild | high | false | yes |
| sunny | cool | normal | false | yes |
| sunny | cool | normal | true | no |
| overcast | cool | normal | true | yes |
| rainy | mild | high | false | no |
| rainy | cool | normal | false | yes |
| sunny | mild | normal | false | yes |
| rainy | mild | normal | true | yes |
| overcast | mild | high | true | yes |
| overcast | hot | normal | false | yes |
| sunny | mild | high | true | no |

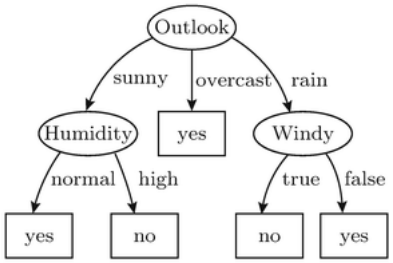



Figure 4.9: A data set describing weather conditions and a target variable (Play Golf?) and a decision tree learned for this dataset (Quinlan 1986) [32]

Decision trees build rules based on the observations provided in the dataset. As its name indicated, a tree gets built based on the decisions the algorithm will make regarding the provided data. An example of the building of a decision tree for classification will be provided in Figure 4.9.

These trees are thus built on the basis of criterion of decision. In order for it to select features on which it would choose to branch, criterion must be chosen to evaluate each feature. The criterion on which the tree bases its choices is called *"impurity"*[32]. Available criteria for classification problems are "Gini" and "Entropy". However, the criterion used for this project will be squared_error because of the nature of the problem.

As a reminder, the objective of this project is to predict the decline of a composite score representing a cognitive faculty. However, the example provided in Figure 4.9 does not involve any regression. Still, decision trees have also been adapted in order to provide a regression output. The disadvantage of that kind of model is that it is often prone to overfitting due to its rule based training. For instance, if the tree managed to build the same number of set of rules as the number of examples, it may be able to perfectly predict every case of the train set but will not be able to adapt to the test set if the case provided in the test set have never been observed. This is handled by using random forests or gradient boosting regressor instead as explained in the following sections.

### Random forest

The random forest is the first presented ensemble model of this project. An ensemble machine learning model is a model that in itself combines several models to base its prediction. The objective of combining multiple instances of a model is that the prediction should be more accurate and stable than with one model.

In this instance, the random forest is composed of a collection of decision trees.This method is referred to as *"Bagging"*[13] where decision trees are built in parallel. For the sake of this study, the decision trees are constructed with the goal of regression rather than categorization in mind. As a result, the random forest discussed in this section is more exactly referred to as a random forest regression.

The forest allows you to fine-tune the amount of decision trees that need to be con-

structed. The greater the number of decision trees that are constructed, the more accurate and reliable the conclusion will be. However, increasing the number of trees in the forest would, of course, increase the amount of time it takes to compute while training the trees. As a result, there is a desire to have several trees to train, but one must make trade-offs between the time available for training and the stability required for the project.

The word random from this model implies that its computation is subject to randomness. This is a feature present in multitude of models as it can ensure that the models predict a target based on data without bias. However, in the field of research, randomness is not allowed since it make the repeatability of some models impossible. It is critical that the findings of the algorithm be able to be confirmed by running it numerous times. The randomness may be "controlled" by setting a parameter called *"random_state"*, which is defined as follows: Setting the random state to a number (42 for instance) may assure that the shuffling, random selections made in the model's calculation will be executed in the same manner if the model is re-run again and again.

> **Note**
> Any other random_state in this project will be set to 42 to be able to reproduce the results.

### Gradient boosting regressor

When doing regression or classification, the gradient boosted predictor and random forests are two ways that combine the outputs of individual trees to provide a more accurate result.

Gradient boosted trees and random forests, on the other hand, differ in terms of how individual trees are produced and how the results are combined (See Figure 4.10). Random forests build distinct decision trees that are subsequently joined in parallel, resulting in a decision tree forest. In contrast, gradient boosting, use a technique referred as boosting[13].

By combining weak learners in a series of steps, boosting ensures that each new weak learner improves the ensemble model based on the mistakes of its predecessor.At the first iteration, a single decision tree get built and the model evaluates its performance by comparing it to a given loss function. In order for the second tree to be effective when combined with the first, it must be created in such a manner that the loss is reduced compared to loss obtained at the previous iteration.

### Multi layer perceptron

As a multilayer perceptron (MLP) neural network model, this kind of neural network is often regarded as the initial step toward the development of deep learning applications.[75]

The MLP is used in a range of applications, including market analysis, which is more of a regression problem, and image analysis, which is more of a classification problem.[75]

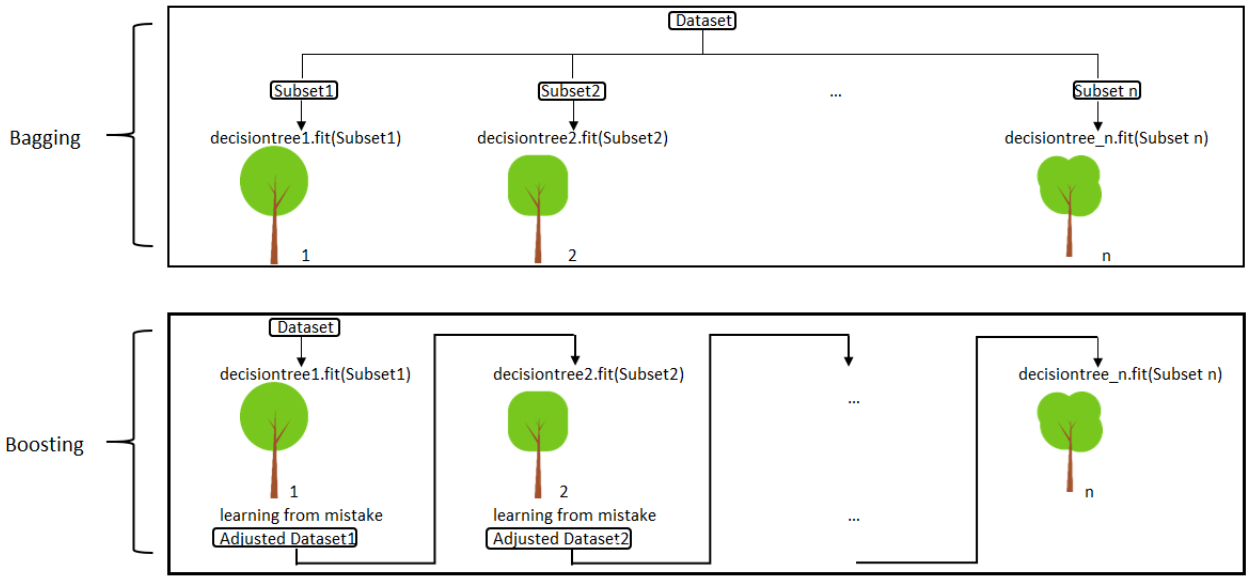



Figure 4.10: Bagging involves training multiple decision trees in parallel whereas boosting involves training the next tree based on the flaws of the actual one.[13]

In a multilayer perceptron, there are three different sorts of layers that are all coupled with each other (See figure 4.11). The input layer is of course, in a hierarchical structure, the first layer to be reached. The output layer is the final layer in the hierarchy. It is responsible for providing the regression or classification result. Hidden layers are all levels that are located between the input and output layers, and are so named because the data that flows in and out of these layers will not be inspected, thus the word "hidden."

While the input layer deals with the values that have been given, a differentiable nonlinear activation function on the data, which includes the weights and values from the layer before it, is used to generate the output values of all the other neurons in the network. As a consequence, the model may be able to learn more intricate functions than a network that has been trained using a linear activation function alone.[38]

The interpretation and feature significance of the multilayer perceptron may be challenging to extract due to the large number of links between the neurons in the MLP. For instance, in figure 4.11, consider the layer h1-h2-h3-h4 with $tanh$ as activation function and layer h5-h6-h7 with activation function $reLu$.
The output of the h5 neuron will be define like following:

$$h5 = reLu(h1 * w8 + h2 * w9)$$

$$h1 = tanh(w1 * i1)$$

$$h2 = tanh(i1 * w2 + i2 * w4 + i3 * w6)$$

Replacing h1 and h2 in h5 give the following value to h5:

$$h5 = reLu(tanh(w1 * i1) * w8 + tanh(i1 * w2 + i2 * w4 + i3 * w6) * w9)$$

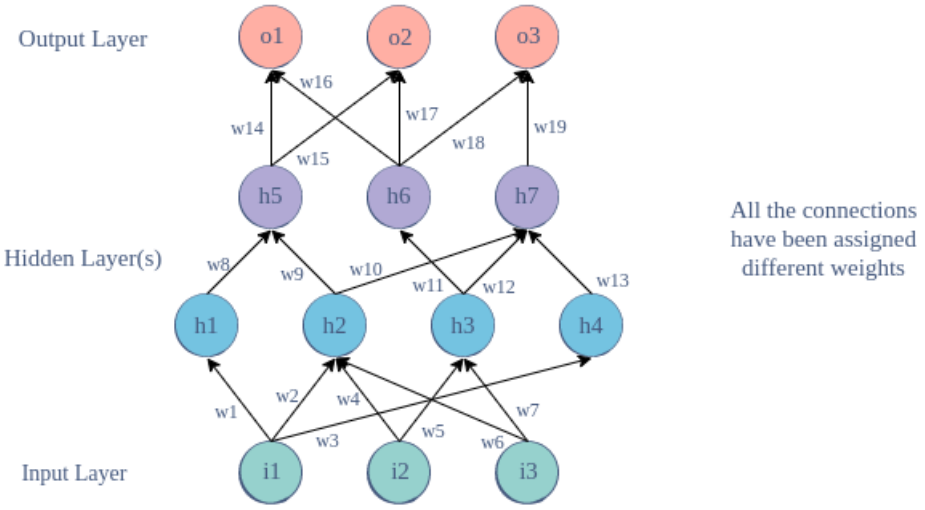



Figure 4.11: Example of a Multilayer perceptron architecture [75]

As demonstrated by this example, the implication of the original input data might be difficult to interpret since it is engaged in numerous calculation processes and cannot always be differentiated from other input variables when analyzing a neuron's output. This is why the MLP is unlikely to be used in the final version of the process designed to extract the feature relevance. However, like explained in Section 4.5, techniques have been designed in order to assess the importance of a feature in a model.

### 4.4.2 Regular train-test split

The data has been divided into two sets: a train set and a test set. Ten percent of the whole dataset is used to create the test set. The initial prototypes will enable us to get a sense of how well various models can perform before improving the train with the Leave One Out method 4.4.3.
The results of the first prototypes for the execution composite scores are given in the figures below. The graphs regarding the first prototypes of the other composite scores can be found in Section 5.3.

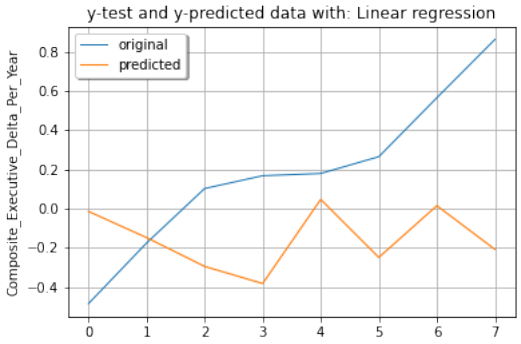



Figure 4.12: Executive score prediction: linear regression

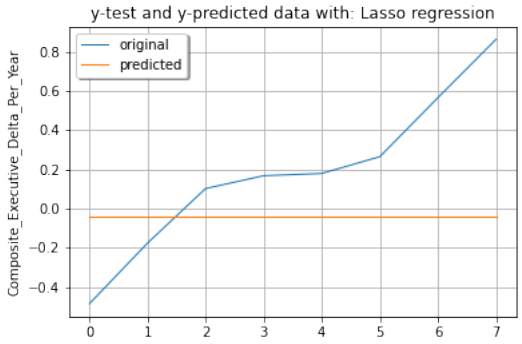



Figure 4.13: Executive score prediction: lasso regression

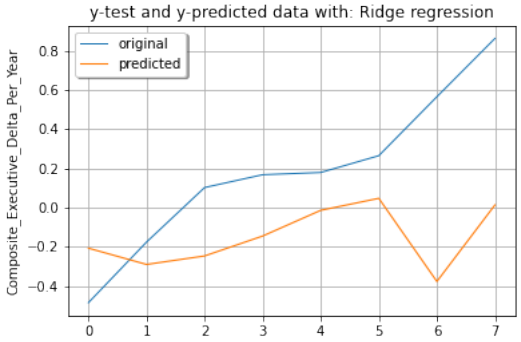



Figure 4.14: Executive score prediction: ridge regression

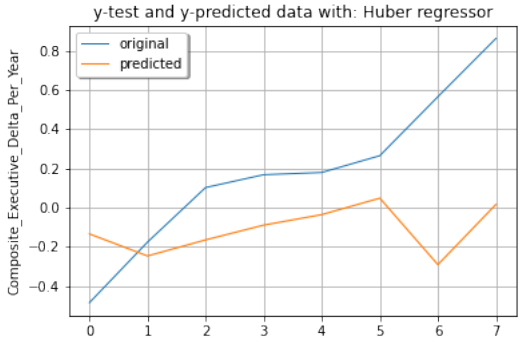



Figure 4.15: Executive score prediction: huber regression

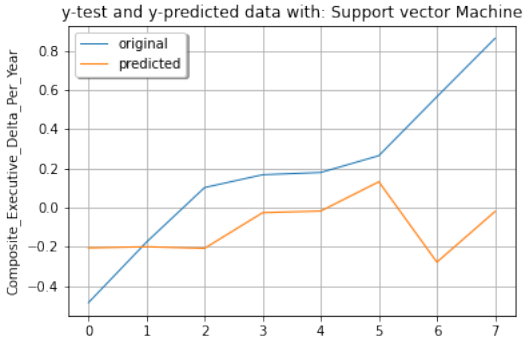



Figure 4.16: Executive score prediction: support vector regression

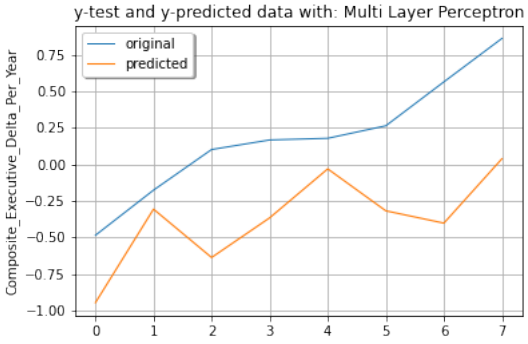



Figure 4.17: Executive score prediction: Multi layer perceptron regression

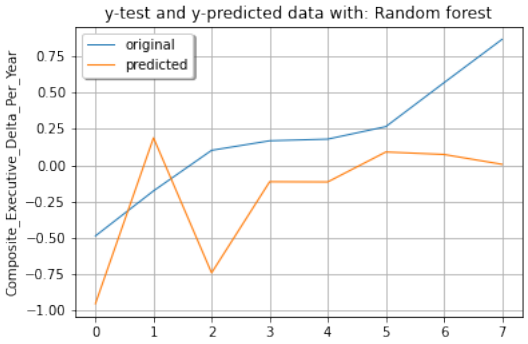



Figure 4.18: Executive score prediction: random forest regression

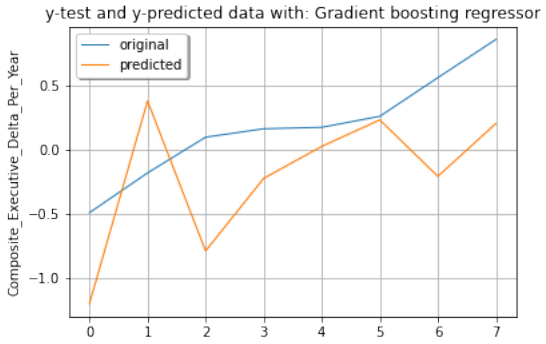



Figure 4.19: Executive score prediction: Gradient boosting regression

| | Model | RMSE | Std_in_pred |
|---|---|---|---|
| **0** | Linear regression | 0.550166 | 0.146983 |
| **1** | Lasso regression | 0.448489 | 0.000000 |
| **2** | Ridge regression | 0.501139 | 0.145044 |
| **3** | Huber regressor | 0.476159 | 0.113025 |
| **4** | Support vector Machine | 0.469674 | 0.130936 |
| **5** | Random forest | 0.528143 | 0.390892 |
| **6** | Gradient boosting regressor | 0.590994 | 0.510365 |
| **7** | Multi Layer Perceptron | 0.619089 | 0.294967 |

Figure 4.20: Results summary for simple train-test split for executive composite score

The RMSE in Figure 4.20 is provided for information purposes only and should not be regarded as conclusive, since the test-train split may have chosen a test set that is either very simple to predict or extremely difficult to predict. The RMSE provides an initial overview of the performance and will serve as a benchmark when the *Leave-one-out* method will be applied.

*Std_in_pred* on the other hand is a variable that indicates the standard deviation of the prediction for the specified test set. It is important to have some standard deviation in your prediction. However, a great standard deviation can indicate a difficulty for the model to adapt to the problem.

As you can observe from figures 4.20, 5.26, 5.17 and 5.35, the Lasso regression has no deviation in its prediction which indicates that the model has considered that return a constant value is efficient. Although the Lasso regression is not performing poorly, we may not consider it for the final selection model regarding the fact that its prediction (see Figure 4.13) does fit in any way to the real data.

The Ridge regression (see Figure 4.14) and Huber regression (see Figure 4.15) performed similarly. The fact that the curves are very similar is probably due to the fact that the huber regression method is some kind of "hybrid" method regarding the fact that depending on the case if will use a different loss function. The ridge and the

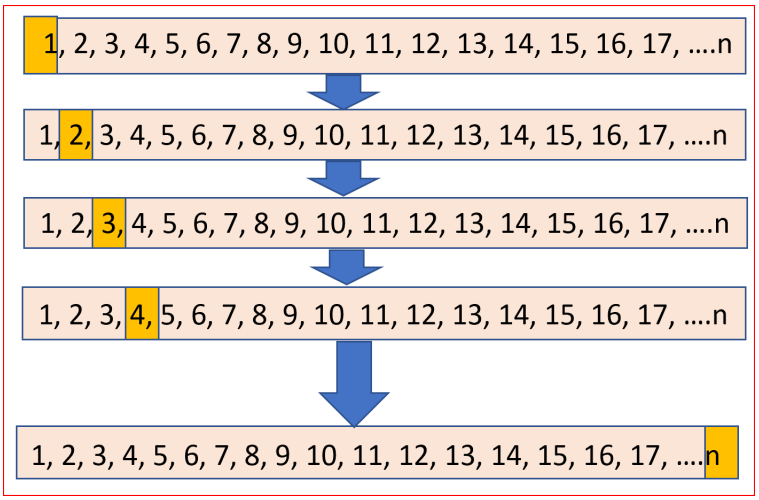



Figure 4.21: *Leave-One-Out- Cross-Validation - LOOCV* [28]

huber regression are thus prone to predict the same output considering that they both sometimes use the square loss function.

The support vector machine (see Figure 4.16) and Multi layer perceptron (see Figure 4.17) both show (for a portion of the test), some similar trends as the curve. However, regarding the standard deviation of each of these two models, the SVM tends to have the better trend while the MLP shows more chaotic behaviour. This needs to be reviewed using the *Leave-one-out* section of the procedure because the chaotic component might be created by an unusually hard to forecast test set.

Finally, the Random forest (see Figure 4.18) and the Gradient boosting regressor (see Figure 4.19) seem to have a very similar prediction. Figure 4.20 shows that the Random forest provides better results with fewer deviation in its prediction. However, when looking at the graphs, we can observe that in some cases, the Gradient boosting regressor is able to predict closer than the random forest and show more ease in getting the trend of the curve.

### 4.4.3 Leave-one out

For this step, the same selected models are trained and tested using the *Leave-one-out* method. This method allows to observe the adaptability of the model to the whole dataset, with all features.

Like illustrated in Figure 4.21, this is accomplished by training the new model on all of the data except for one line (represented in orange in Figure 4.21) and then testing it on the line that was left out. This procedure is performed over and over again on the whole dataset, with a new model being created for each iteration .

The RMSE is calculated for each of these stages and is saved in an array. Finally, the mean of the array, as well as the standard deviation of the RMSEs, are calculated. Model adaptability may be determined from the mean and standard deviation when they are used in conjunction with one another.

It is extremely valuable to have a good mean, yet having a good mean by itself does not provide much information. For example, if half of the predictions overestimate the target while the other half underestimate it, this will result in a small mean RMSE which will not represent the stability of the model in relation to the data, the model will be unstable.

A low standard deviation, on the other hand, provides information about the model's stability. A low standard deviation shows that the mean is reliable.
Results of the *Leave-one-out* method can been see for the executive score below in Figure 4.22. The results for the other composite scores are provided in Section 5.4 of the Appendices.

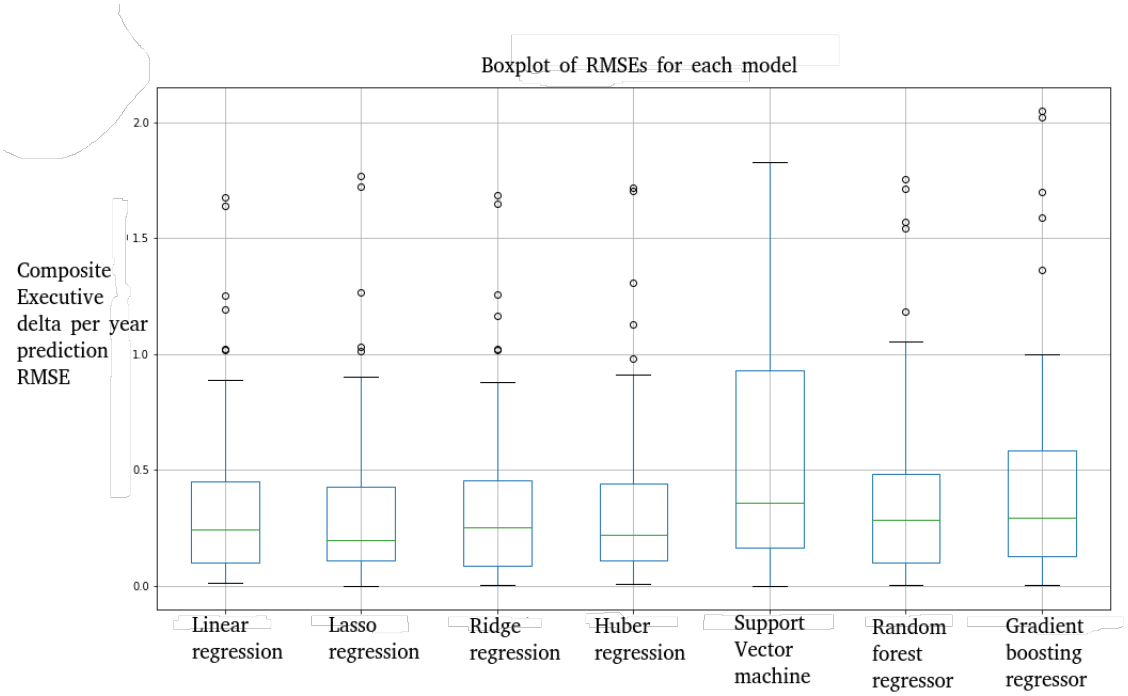



Figure 4.22: Results summary of the leave-one-out method for executive composite score before recursive feature selection

Note that the multilayer perceptron was removed from the process since it had a mean RMSE of arround 109. This could not be illustrated for graphical reasons. The MLP was then of course removed from the candidates.

Like observed in the above figure, the Lasso regression model is the most successful. According to the Lasso regression, which has been previously mentioned, a constant value is predicted in all instances. As a consequence, since it makes no effort to forecast the target, this model will not be considered for further consideration.

Among the available models, the support vector machine has the worst performance. This is due to the fact that the support vector machine was unable to converge for

the vast majority of the time. Moreover, the support vector machine seemed to have performed poorly in terms of forecasting the target when it finally reached convergence.

Both the Huber and Ridge regressions worked rather well, yielding essentially identical findings. However, it seemed as if the Huber regression performed marginally better than the Ridge regression. It is worth noting that the Huber regression performed similarly to the standard linear regression.

It was anticipated that the performance of both the random forest and the gradient boosting regressor would be similar since they are both tree-based ensemble models. Despite the fact that they have relatively similar performance, it is important to note that they do not deviate much from linear models.

## 4.5 Feature selection

For the purpose of extracting the most relevant possible features from the datasets and being able to compare them, it is necessary to consider the potential connection between the four scores we are attempting to forecast. It is true that the extraction of the same features should be more or less the same when two strongly correlated scores are used together. It is reasonable to conclude that if two highly linked composite scores did not enable us to extract the same features, there is a problem with the process by which we are attempting to identify the relevant features. Correlations between the four composite scores can be observed in Figure 4.23

As a reminder, note that the composite score are given as difference in score per year. Two composite score variations that are substantially connected would indicate that these two cognitive skills are deteriorating at the same time.
Figure 4.23 shows that there is no clear linear relationship between two variations in composite scores. The greatest correlation was found between the language and memory scores, with a correlation coefficient of 0.49. We may expect to see variances in the features chosen for each of the four scores as a consequence of this outcome.

### 4.5.1 Recursive feature selection

Recursive feature selection, also know as recursive feature elimination (RFE) is a technique allowing to select the best features for a given model. This is done in order to keep the features that have the most importance in the model's prediction. As a consequence, features that have been eliminated by this algorithm may have not been deleted with a classical feature selection approach.

The first classical way to deal with the features to select is to set the desired number of features to some "K" number. The RFE algorithm will then be able to select the top-K features in order to optimize the predictor.
The challenge for that kind of technique is thus to choose the best number of features to keep.

Through the process of recursive feature extraction, each model has been evaluated

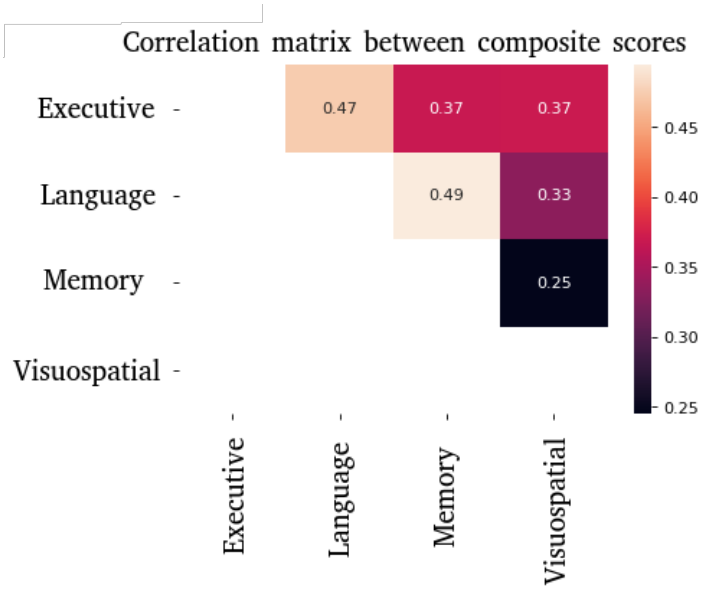



Figure 4.23: Correlation between composite scores

in the same way as in Section 4.4.3, that is to say with the Leave-one-out method. The results of the RMSE after the recursive feature selection is given below.

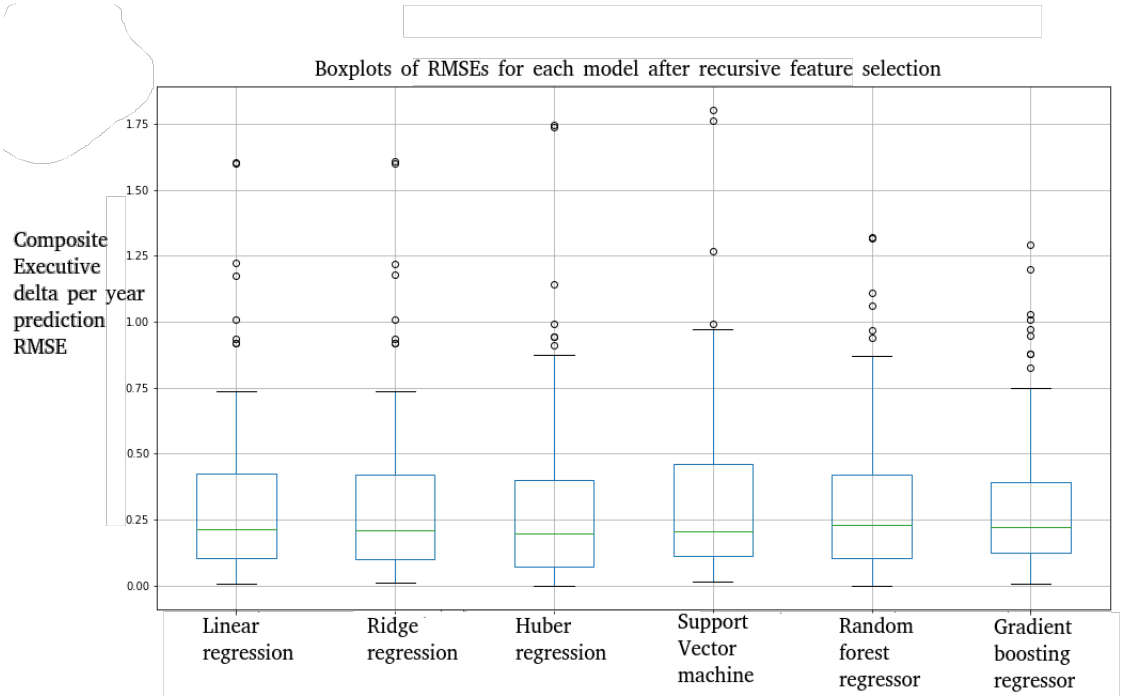



Figure 4.24: Results summary of the leave-one-out method for executive composite score after recursive feature selection

Features for the each composite score have been selected. It is important to remember that composite scores are made up of numerous tests, each of which may need the use of various lobes of the brain to be completed. As a result, it is crucial to remember that for each composite score, several regions from various lobes may be picked because they

are judged relevant by the models. The top-10 features selected through permutation feature importance are given below for the ensemble models.

The areas chosen will be classified according to their respective lobe. Each of the existing lobes has a distinct role, however they are not mutually exclusive. Still, as shown in this article *"What Does the Brain's Cerebral Cortex Do?"*[77], each lobe is well-known for performing a specific cognitive function.

### 4.5.2 Permutation feature importance

Permutation feature importance is a technique used for assessing, for a given trained machine learning model, the importance of each feature of the predictive power of the model. This technique can become very handy for models that do not provide an easy to interpret feature importance.

Some linear regression models can provide the coefficients associated to each feature which can indicate the importance of the feature for this model. On the other hand, non-linear models that do not necessarily provide any possible coefficient for the given features since they do no work the same way as linear regression models. Ensemble models for instance are a kind a model that provide a feature importance property. In order to compare linear models to ensemble models we must use the same way to assess the importance given to each feature for each model.

Figure 4.25 illustrates a feature permutation importance illustration. To determine the significance of a feature in a pre-trained model, a benchmark score or error is calculated using the original dataset. This benchmark will act as a comparison for the remainder of the procedure.
Afterwards, each characteristic is assessed individually. One feature is shuffled (i.e., patients' results for a specific feature are swapped), but all other features remain unchanged. The pre-trained model is then used to predict the same target as the benchmark with the shuffled feature. If the error has grown, it suggests that the feature has a significant impact on the predictive ability of the model. In contrast, an error equal to or less than the initial error from the benchmark implies that the feature does not aid the model in its prediction and is, thus, of lesser importance. The feature importance is computed as following:

$$i_j = s - \frac{1}{K} \sum_{k=1}^{K} s_{k,j}$$

With:

- $s$ the reference score without shuffling any feature
- $j$ the index of the feature
- $K$ the number of permutations and score to evaluate
- $s_{k,j}$ the $kth$ iteration for shuffling feature $j$

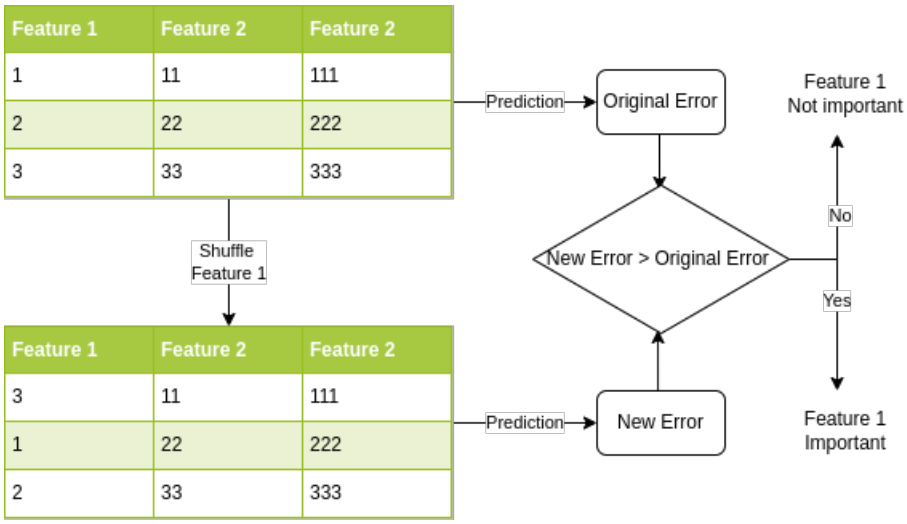



Figure 4.25: Permutation feature importance technique example

## 4.6 Results and interpretation

In this part, the permutation importance results for both the random forest and gradient descent models will be shown. These will be listed for each of the predicted composite scores. While both the linear and ensemble models predicted with the same degree of accuracy, the ensemble models were chosen to identify the most significant features to look for. This is due to the fact that linear models performed poorly during recursive feature selection. In fact, the linear models selected relatively few features (i.e. between 1 and 3). In contrast, the ensemble models exhibited a variety of features and were thus selected as the final models to utilize.

For each of the composite scores, the top-10 features for both of the two models. Common selected features will be written in bold for ease of reading. A discussion and interpretation for the brain regions and lobe will be provided for each of the results shown below. Note that if a brain region name is given as important, we are in fact talking about the cortical thickness of this region.

### 4.6.1 Executive

As a reminder, the assessments taken into account for the **Executive** composite score are the Trail Making Test and the Luria test described in the state of the art.

The link between the TMT and the cortical thickness of several regions was already assessed by MacPherson et al. in the paper "Processing speed and the relationship between Trail Making Test-B performance, cortical thinning and white matter microstructure in older adults"[47].

MacPherson et al. concluded that in five large and statistically significant studies, significant (RFT-corrected) relationships were discovered between TMT-B completion time and cortical thickness while controlled for age, gender, education, and intracranial volume. These included the lateral frontal and temporal regions, the Sylvian fissure/insula, supramarginal and Inferior parietal regions, Inferior motor and sensory areas, and the cingulate gyrus isthmus (all bilateral). The thinner the cortex in these locations, the slower the TMT-B performance of elderly individuals.[47]

The different regions selected by the random forest and the gradient descent regressor are listed below:

Table 4.3: Regions selected by recursive feature selection for the Executive composite score: List of regions selected for both the random forest regressor and the gradient boosting regressor. Regions in bold are common to both models. Scores attributed to regions are their respective permutation feature importance.

| | Random Forest | *Score* | Gradient Descent | *Score* |
|---|---|---|---|---|
| 1 | **Lingual** | 46.6 | **Lingual** | 48.4 |
| 2 | **Isthmus cingulate** | 14.3 | **Isthmus cingulate** | 15.8 |
| 3 | **Pars opercularis** | 10 | Hippocampal volume | 11.9 |
| 4 | **Inferior parietal** | 7.4 | **Pars operculiaris** | 6.4 |
| 5 | Insula | 7.3 | **Superior temporal** | 4.2 |
| 6 | **Superior temporal** | 7 | **Inferior parietal** | 2.9 |
| 7 | **Age at Baseline** | 5.4 | **Age at Baseline** | 2.4 |
| 8 | Caudal anterior cingulate | 4.5 | Paracentral | 2.3 |
| 9 | Enthorinal | 4 | Precentral | 1.9 |
| 10 | Pars triangularis | 3.8 | Temporal pole | 1.8 |

Regions of interest will be grouped by lobe for interpretation:

1. Occipital: Lingual
2. Cingulate: Isthmus cingulate
3. Frontal: Pars opercularis
4. Parietal: Inferior parietal
5. Temporal: Superior temporal

In this case, we can see that the model considered numerous lobes to be engaged in the execution assessment of a patient's. This might be a consequence of the fact that the evaluations that were necessary for the execution assessment required the involvement of different parts of the brain at different times.

Like the results found in the article described before[47], we can indeed highlight a relation between the Isthmus cingulate and Inferior parietal region with the execution composite score which is encouraging for the validation of the process.

The Luria test mentioned in the Appendices might be involved in the fact that the selected regions are widespread over multiple lobes. However, no study to link the cortical thickness to this particular version of the test was conducted so this will remain a hypothesis.
In reality, the assessments taken by the patients necessitated the use of vision, motility, and spatial relation treatment, which necessitated the use of various regions across different lobes in order to complete the tasks successfully.

### 4.6.2 Language

The language function is well known for being lateralized. *"Language lateralisation refers to the phenomenon in which one hemisphere (typically the left) shows greater involvement in language functions than the other."*[16]. The verbal fluency tasks especially tend to be lateralized which might be a problem regarding the fact that we used an average cortical thickness for both hemispheres.

As a reminder, the assessments taken into account for the language composite score are the Fluency 4 test as well as the Naming 5 test. In 2013, Eastman et al. published the following paper: "Cortical thickness and semantic fluency in Alzheimer's disease and mild cognitive impairment". This study being closely related to the objective of this project, it will be use to try and interpret the quality of the results.

The result published by Eastman et al. suggested that cortical atrophy of the Inferior parietal lobe (BA 39 and 40) and the premotor and dorsolateral prefrontal cortices (BA 4, 6, 8, 9 and 46) bilaterally correlated with poor semantic fluency on the vegetables test.[26]
Left lateral frontal (BA 10, 44, 45), medial frontal (BA 4,6, 8, 9 and 32), lateral temporal (BA 22), medial parietal (BA 31 and 7) and peristriate (BA 18,19) cortices had more diffuse relationships.[26]

The Lexis Naming test included in the language composite score construction will also have an influence on the selected region. Depending on the nature of the assessment, the recursive feature selection approach may pick locations different than those already indicated for the fluency test.

Table 4.4: Regions selected by recursive feature selection for the Language composite score: List of regions selected for both the random forest regressor and the gradient boosting regressor. Regions in bold are common to both models. Scores attributed to regions are their respective permutation feature importance.

| | Random Forest | *Score* | Gradient Descent | *Score* |
|---|---|---|---|---|
| 1 | **Amyloid SUVr** | 17.4 | **Amyloid SUVr** | 20.3 |
| 2 | **Lingual** | 7.8 | **Superior frontal** | 11.7 |
| 3 | Age at Baseline | 5.2 | **Caudal middle frontal** | 9.7 |
| 4 | **Superior frontal** | 4.6 | Fusiform | 5.9 |
| 5 | Lateral occipital | 4.6 | Superior temporal | 5.4 |
| 6 | **Caudal anterior cingulate** | 4.4 | **Caudal anterior cingulate** | 4.8 |
| 7 | Precuneus | 4.2 | **Lingual** | 4.0 |
| 8 | **Pericalcarine** | 4 | **Pericalcarine** | 2.6 |
| 9 | Hippocampal volume | 3.9 | Inferior temporal | 2.2 |
| 10 | **Caudal middle frontal** | 3.3 | Pars triangularis | 2 |

Regions of interest grouped by lobe:

1. Frontal: Superior frontal, Caudal middle frontal

2. Occipital: Lingual, pericalcarine

3. Cingulate: Caudal anterior cingulate

4. Amyloid SUVr

The amyloid SUVr was selected in contrast to the results given for the execution. This is interesting to observe since it has already been attested that amyloid plays a crucial role in Alzheimer's disease regarding the memory functions of patients.
It is reasonable to assess that the frontal lobe is involved in this the language decline, since the fluency test necessitates to develop strategies in order to cite the most name possible. Moreover, like mentioned in the previous article[26], the cortical thickness of the **medial frontal** seemed to have an influence on the fluency assessment. As we can see from the results above, the **superior frontal** and **Caudal middle frontal** were selected by recursive feature selection. These two regions are located near the medial frontal which is not strictly defined in the atlas used with Freesurfer. Corresponding regions for Brodmann's areas (BA 4, 6, 8, 9 and 46) include the caudal middle frontal and the superior frontal regions which confirms the expected results.

### 4.6.3 Memory

As a reminder, the assessment taken into account for the **Memory** composite score is the FCSRT3 test.

In the article "Prediction of free and cued selective reminding test performance using volumetric and amyloid-based biomarkers of Alzheimer's disease", relationships between the cortical thicknesses of a number of regions and the FCSRT evaluation are discussed by Quenon et al.[62]. This paper suggests that our study will emphasize temporal lobe areas, namely the entorhinal cortex. However, the outcome may vary due to the way cortical thicknesses were accounted for in this study (i.e. average of both hemispheres).

Table 4.5: Regions selected by recursive feature selection for the Memory composite score: List of regions selected for both the random forest regressor and the gradient boosting regressor. Regions in bold are common to both models. Scores attributed to regions are their respective permutation feature importance.

| | Random Forest | *Score* | Gradient Descent | *Score* |
|---|---|---|---|---|
| 1 | **Enthorinal** | 31.3 | **Transverse temporal** | 18 |
| 2 | **Amyloid SUVr** | 28.3 | **Amyloid SUVr** | 17.7 |
| 3 | **Transverse temporal** | 18.7 | **Inferior temporal** | 15.8 |
| 4 | Caudal anterior cingulate | 11 | **Enthorinal** | 9.3 |
| 5 | **Medial orbitofrontal** | 11 | Hippocampal volume | 8.5 |
| 6 | **Inferior temporal** | 10 | Pars triangularis | 6.9 |
| 7 | **Rostral anterior cingulate** | 10 | **Rostral anterior cingulate** | 6.6 |
| 8 | Superior temporal | 7 | Fusiform | 4.3 |
| 9 | | | Cuneus | 2.6 |
| 10 | | | **Medial orbitofrontal** | 2.2 |

Regions of interest are grouped by lobe:

1. Temporal: Enthorinal, Transverse temporal, Inferior temporal
2. Frontal: Medial orbitofrontal
3. Cingulate: Rostral anterior cingulate
4. Amyloid SUVr

As expected, the temporal lobe and in particular the entorhinal cortex have been highlighted. Just like the results provided for the language composite score, the amyloid SUVr data has been selected as an important feature for the model to asses the decline of the memory composite score.

While less expected, the Medial orbitofrontal, and the frontal lobe in general seem to have play role in the FCSRT assessment according to the article *"Impairment of episodic memory in genetic frontotemporal dementia: A GENFI study"*[59]. The frontal lobe may be solicited for strategic purposes regarding the fact that the patient may build strategies in order to recall the words learning earlier during the assessment.

It is interesting to note that the link between $\beta$ - amyloid deposits and with the atrophy of the frontal and cingulate selected regions have also been assessed by the following article: "Relationship between atrophy and $\beta$-amyloid deposition in Alzheimer disease"[18].
This article suggested that global neocortical $\beta$ - amyloid deposition was associated with atrophy in a large brain network which included hippocampus, medial frontal and parietal areas, and lateral temporoparietal cortex, whilst local $\beta$ - amyloid load was only associated with local atrophy in the regions with the highest $\beta$ - amyloid load, namely the medial orbitofrontal and anterior and posterior cingulate/precuneus regions.[18]

### 4.6.4 Visuo-Spatial

As a reminder, the assessments taken into account for the **Visuo-spatial** composite score are the Clock test2 and the CERAD8 set of tests.

The cortical thickness of several regions have already been analysed in parallel with scores on *"CERAD protocol subtests"* by Youn et al. in the study "Decreased Cortical Thickness and Local Gyrification in Individuals with Subjective Cognitive Impairment"[79].

This study highlighted that there was a reduction in cortical thickness for the SCI (Subjective cognitive impairment) group, in the left entorhinal, superior temporal, insular, rostral middle frontal, precentral, superior frontal, and supramarginal regions, and in the right supramarginal, precentral, insular, postcentral, and posterior cingulate regions. Under the CERAD protocol, cortical thickness in these regions was correlated with scores on *"the constructional praxis, word list memory, word list recall, constructional recall, trail making test A, and verbal fluency tests."*[79]

Table 4.6: Regions selected by recursive feature selection for the Visuospatial composite score: List of regions selected for both the random forest regressor and the gradient boosting regressor. Regions in bold are common to both models. Scores attributed to regions are their respective permutation feature importance.

| | Random Forest | *Score* | Gradient Descent | *Score* |
|---|---|---|---|---|
| 1 | **Rostral middle frontal** | 38 | **Rostral middle frontal** | 68 |
| 2 | **Age at Baseline** | 20 | **Age at Baseline** | 40,1 |
| 3 | **Superior temporal** | 13.2 | **Superior temporal** | 33.3 |
| 4 | **Caudal anterior cingulate** | 6 | **Caudal anterior cingulate** | 21.2 |
| 5 | **Lateral orbitofrontal** | 5.4 | **Lateral orbitofrontal** | 12.5 |
| 6 | Rostral anterior cingulate | 5 | | |
| 7 | Pars opercularis | 4.2 | | |
| 8 | Pars triangularis | 4 | | |
| 9 | Caudal middle frontal | 3.8 | | |
| 10 | Insula | 3.6 | | |

Regions of interest are grouped by lobe:

1. Frontal: Rostral middle frontal, Lateral orbitofrontal
2. Temporal: Superior temporal
3. Cingulate: Caudal anterior cingulate
4. Age at baseline

Like mentioned in the previously cited article[79], the rostral middle frontal cortical thickness as well as the superior frontal cortical thickness both seem to be correlated, or in any case related, to the CERAD battery test.

According to both ensemble models, the age factor looked significant. This is noteworthy since this is the only composite score for which the age has been chosen. However, while it is reasonable to think that age may influence the deterioration of visuo-spatial abilities, in the context of Alzheimer's disease, it is not sufficient to explain a sudden decline in visuo-spatial functions.

The selection of the cingulate lobe cannot be explained fully at this time, however the Caudal anterior cingulate area was likewise chosen for the language composite. This area may serve a broad purpose with respect to a cognitive assessment-required function.

### 4.6.5 RMSE after recursive feature selection

The RMSEs for each composite score described above have been computed for both the random forest regressor and the gradient boosting regressor. Each of these models have seen their hyper-parameters optimized using a Grid Search CV which is a method trying out every combination of a given set of parameters in order to find the best performing one. This was done on 85 percent of the dataset, the 15 percent remaining have been used for the test to assess the root squared mean error of the model. The training performed with the gridsearch were conducted with Leave-one-out.

The RMSE is given for each model along side the standard deviation and the mean of the distribution for each composite score:

Table 4.7: Final performance of Random forest and Gradient boosting regressor for Executive, Language, Memory and Visuospatial composite scores decline prediction.

| | RF | GB | Mean | Std |
|---|---|---|---|---|
| Executive | 0,528 | 0,58 | -0,018 | 0,487 |
| Language | 0,387 | 0,399 | -0,076 | 0,351 |
| Memory | 1,055 | 1,146 | -0,143 | 0,7 |
| Visuospatial | 0,645 | 0,45 | -0,211 | 0,771 |

The RMSE for each composite score is not accurate enough to qualify a model as suitable for correctly forecasting the deterioration of a patient. However, data do not demonstrate chaotic predictions and the feature selection process provided coherent results which is promising.

Most likely, the lack of accuracy is due to a lack of data. As the absence of data is a problem that may be resolved, this will be explored under the discussion and improvements section.

### 4.6.6 Discussion and improvements

The coherence of the method that was established during the course of this project is validated by the overall findings that were supplied by the recursive feature selection.

These outcomes, however, are not flawless and there is room for improvement.

After all, the quantity of accessible data is most likely the project's biggest obstacle. To enhance outcomes using machine learning, adequate data must be supplied. Without a substantial quantity of data, machine learning and data analysis algorithms cannot provide accurate findings. As a consequence of this, repeating this process whenever there is fresh data available will allow us to acquire conclusions that are more reliable and will also let us to monitor the development of the model-based selection whenever there is new data presented.

## 4.7 Continuous learning

The remedy to the data shortage is pretty simple: acquire more data. However, the medical research sector is not prone to swiftly gathering large volumes of data.

To address this problem, the notion of continuous learning must be implemented. Continuous learning is a method for supplying machine learning with fresh data as it becomes accessible. This guarantees that the model grows more precise and robust over time. This is why the infrastructure is the key objective of the project.

The infrastructure began with the development of a database, followed by data analysis and machine learning prototypes. When these two components are combined, machine learning may be executed anytime fresh data becomes available.

Every month, the training of the model will be restarted from scratch by accessing the database for the necessary training data. The progress of the model, including the quantity of fresh information, the prior and current accuracy, and the specified characteristics, will be shown on an HTML page.

Note that for the purpose of this project, the random aspects have been fixed (random_state=42) which might a poor choice. This was done for repeatability purposes. In "production", this will be removed as the models and functions used are subject to randomness and because repeatability is not a necessary aspect here. The model will be saved locally after training in case it was needed for prediction purposes.

Moreover, this pipeline is flexible in terms of features to study as well as target to predict. This part of the project can thus be adapted in the future and can be used to predict other target or to assess the importance of a new feature in the prediction of the actual or a different target.

An overview of the graphical interface displaying the reports regarding continuous learning can be found in appendix, Section 7.2.

## 4.8 Conclusion

The purpose of the project was to set up a data analysis pipeline to study and demonstrate the added value of the implementation of machine learning in Alzheimer's disease research.

Although built on a relatively restricted amount of data, the results of the study are encouraging. However, it is noteworthy that, due to the limited data available, one should be cautious about the findings at this stage and regard them as insufficiently robust to make firm conclusions about the evolutionary nature of the various medical parameters related to the development of the different stages of Alzheimer's disease.

In response to this lack of data and in order to achieve a sustainable and continuous influx from the patient population, a continuous learning system has been implemented. This approach allows the continuous inclusion and analysis of new data.

As the subject matter remains vast and the in regard to the broad field of application of machine learning techniques included in a continuous dynamic, it goes without saying that this study is merely one step in a far-reaching process. This study contributes to the reflection on the integration of artificial intelligence technology in the service of medical science. It does not constitute an end in itself, but rather a start towards the development of effective means of exploring many other possibilities on the one hand, and the deepening and permanent adjustment of this process on the other.

On a personal level, the exploration of the theme of the application of machine learning in the medical context, more particularly in the field of Alzheimer's disease, has been a very stimulating experience in terms of personal development. The study has intensified my curiosity and eagerness to explore the possibilities of marrying data analysis techniques, algorithms and machine learning processes with the medical world. I deign to have brought, in the course of time, a stone to this edifice by means of this project which is intended to facilitate, support and assist the work of researchers in their work.
I hope that this milestone will inspire others to pursue the work I had undertaken in order to improve it and make it evolve in line with the medical discoveries it will contribute to identify.

# Bibliography


[1] *Advantages of Microservices*. URL: https://www.javatpoint.com/advantges-and-disadvantages-of-microservices. (accessed: 19.04.2022).

[2] NIH National Institute on Aging. *What Is Dementia? Symptoms, Types, and Diagnosis*. URL: https://www.nia.nih.gov/health/what-is-dementia. (accessed: 20.03.2022).

[3] Berit Agrell and Ove Dehlin. "The clock-drawing test". In: *Age and Ageing* 27 (1998), pp. 309–403.

[4] *Alzheimer's and Dementia*. URL: https://www.alz.org/alzheimer_s_dementia. (accessed: 27.03.2022).

[5] *Amyloid PET*. URL: https://www.cms.gov/Medicare/Coverage/Coverage-with-Evidence-Development/Amyloid-PET. (accessed: 12.04.2022).

[6] *Amyloid-beta Precursor Protein*. URL: https://pdb101.rcsb.org/motm/79. (accessed: 12.04.2022).

[7] *An Introduction to Support Vector Regression (SVR)*. URL: https://towardsdatascience.com/an-introduction-to-support-vector-regression-svr-a3ebc1672c2. (accessed: 26.04.2022).

[8] *APARC Parcellations by Lobes*. URL: https://bookdown.org/u0243256/tbicc/freesurfer.html. (accessed: 12.04.2022).

[9] *APOE gene*. URL: https://medlineplus.gov/genetics/gene/apoe/. (accessed: 12.04.2022).

[10] Ingrid Arevalo[U+2010]Rodriguez et al. "Mini[U+2010]Mental State Examination (MMSE) for the detection of Alzheimer's disease and other dementias in people with mild cognitive impairment (MCI)". In: *Cochrane Database of Systematic Reviews* 3 (2015). URL: https://doi.org//10.1002/14651858.CD010783.pub2.

[11] "Automated cortical thickness measurements from MRI can accurately separate Alzheimer's patients from normal elderly controls". In: *Neurobiology of Aging* 29.1 (2008), pp. 23–30. DOI: https://doi.org/10.1016/j.neurobiolaging.2006.09.013.

[12] Peter W. Baas et al. "Stability properties of neuronal microtubules". In: *Cytoskeleton* 73.9 (Sept. 2016), pp. 442–460. DOI: 10.1002/cm.21286. URL: https://doi.org/10.1002/cm.21286.

[13] *Basic Ensemble Learning (Random Forest, AdaBoost, Gradient Boosting)- Step by Step Explained*. URL: https://towardsdatascience.com/basic-ensemble-learning-random-forest-adaboost-gradient-boosting-step-by-step-explained-95d49d1e2725. (accessed: 26.04.2022).

[14] Daniel Bell and Ki Yap. *Standard uptake value*. July 2011. DOI: 10.53347/rid-14456. URL: https://doi.org/10.53347/rid-14456.

[15] Heiko Braak and Eva Braak. "Neuropathological stageing of Alzheimer-related changes". In: *Acta neuropathologica* 82.4 (1991), pp. 239–259.

[16] Abigail R. Bradshaw et al. "Measuring language lateralisation with different language tasks: a systematic review". In: *PeerJ* 5 (Oct. 2017), e3929. DOI: 10.7717/peerj.3929. URL: https://doi.org/10.7717/peerj.3929.

[17] Herman Buschke. "Cued recall in amnesia". In: *Journal of Clinical and Experimental Neuropsychology* 6 (1984), pp. 433–440.

[18] Gaël Chételat et al. "Relationship between atrophy and $\beta$-amyloid deposition in Alzheimer disease". In: *Annals of neurology* 67.3 (2010), pp. 317–324.

[19] *Cortical Parcellation*. URL: https://surfer.nmr.mgh.harvard.edu/fswiki/CorticalParcellation. (accessed: 19.04.2022).

[20] Thomas M Cover. *Elements of information theory*. John Wiley & Sons, 1999.

[21] MP De Partz et al. "LEXIS: Tests pour l'évaluation des troubles lexicaux chez la personne aphasique". In: *Marseille: Solal* (2001).

[22] Rahul S Desikan et al. "An automated labeling system for subdividing the human cerebral cortex on MRI scans into gyral based regions of interest". In: *Neuroimage* 31.3 (2006), pp. 968–980.

[23] *Diagnosing Alzheimer's: How Alzheimer's is diagnosed*. URL: https://www.mayoclinic.org/diseases-conditions/alzheimers-disease/in-depth/alzheimers/art-20048075. (accessed: 12.04.2022).

[24] Bradford C Dickerson et al. "Detection of cortical thickness correlates of cognitive performance: reliability across MRI scan sessions, scanners, and field strengths". In: *Neuroimage* 39.1 (2008), pp. 10–18.

[25] *Differences Between SQL vs NoSQL*. URL: https://www.scylladb.com/learn/nosql/nosql-vs-sql/. (accessed: 14.04.2022).

[26] Jennifer A Eastman et al. "Cortical thickness and semantic fluency in Alzheimer's disease and mild cognitive impairment". In: *American journal of Alzheimer's disease (Columbia, Mo.)* 1.2 (2013), p. 81.

[27] James M. Ellison. *Tau Protein and Alzheimer's Disease: What's the Connection?* URL: https://www.brightfocus.org/alzheimers/article/tau-protein-and-alzheimers-disease-whats-connection. (accessed: 20.03.2022).

[28] *Eviter le overfitting*. URL: https://act6100.netlify.app/perform_ml/2_eviter_overfitting. (accessed: 27.04.2022).

[29] Bruce Fischl et al. "Automatically parcellating the human cerebral cortex". In: *Cerebral cortex* 14.1 (2004), pp. 11–22.

[30] *FreeSurfer*. URL: https://surfer.nmr.mgh.harvard.edu/fswiki. (accessed: 26.05.2022).

[31] *FreeSurfer software suite*. URL: https://surfer.nmr.mgh.harvard.edu/. (accessed: 12.04.2022).

[32] Johannes Furnkranz. "Decision Tree". In: *Encyclopedia of Machine Learning and Data Mining*. Ed. by Claude Sammut and Geoffrey I. Webb. Boston, MA: Springer US, 2017, pp. 330–335. ISBN: 978-1-4899-7687-1. DOI: 10.1007/978-1-4899-7687-1_66. URL: https://doi.org/10.1007/978-1-4899-7687-1_66.

[33] Rowena G.Gomez and Desirée A.White. "Using verbal fluency to detect very mild dementia of the Alzheimer type". In: *Archives of Clinical Neuropsychology* 21 (2006), pp. 771–775.

[34] Bernard J Hanseeuw et al. "Association of amyloid and tau with cognition in preclinical Alzheimer disease: a longitudinal study". In: *JAMA neurology* 76.8 (2019), pp. 915–924.

[35] *Having a PET scan*. URL: https://www.alzheimers.org.uk/research/take-part-research/pet-scan. (accessed: 12.04.2022).

[36] National Institutes of Health et al. *What happens to the brain in Alzheimer's disease*. 2017.

[37] *How is Alzheimer's disease diagnosed and evaluated?* URL: https://www.radiologyinfo.org/en/info/alzheimers. (accessed: 12.04.2022).

[38] *How to Choose an Activation Function for Deep Learning*. URL: https://machinelearningmastery.com/choose-an-activation-function-for-deep-learning. (accessed: 26.04.2022).

[39] Benjamin M Kandel et al. "Predicting cognitive data from medical images using sparse linear regression". In: *International Conference on Information Processing in Medical Imaging*. Springer. 2013, pp. 86–97.

[40] Sayantan Kumar et al. "Machine learning for modeling the progression of Alzheimer disease dementia using clinical data: a systematic literature review". In: *JAMIA Open* 4.3 (Aug. 2021). ISSN: 2574-2531. DOI: 10.1093/jamiaopen/ooab052.

[41] *L1 and L2 Regularization Methods*. URL: https://towardsdatascience.com/l1-and-l2-regularization-methods-ce25e7fc831c. (accessed: 26.04.2022).

[42] *Lasso Regression: Simple Definition*. URL: https://www.statisticshowto.com/lasso-regression/. (accessed: 26.04.2022).

[43] P. E. Latham and Y. Roudi. "Mutual information". In: *Scholarpedia* 4.1 (2009). revision #186917, p. 1658. DOI: 10.4249/scholarpedia.1658.

[44] Raquel Lemos et al. "The Free and Cued Selective Reminding Test Distinguishes Frontotemporal Dementia From Alzheimer's Disease". In: *Archives of Clinical Neuropsychology* 29 (2014), pp. 670–679.

[45] Renaud Lhommel et al. *In vivo Amyloid Plaques Quantification using F18-Flutemetamol in 30 Healthy Elderly Controls and 62 MCI patients: SUVr comparison between PMOD 3.2 and PNEURO 3.5 analysis*. 2016.

[46] S Li et al. “Hippocampal shape analysis of Alzheimer disease based on machine learning methods”. In: *American Journal of Neuroradiology* 28.7 (2007), pp. 1339–1345.

[47] Sarah E MacPherson et al. “Processing speed and the relationship between Trail Making Test-B performance, cortical thinning and white matter microstructure in older adults”. In: *Cortex* 95 (2017), pp. 92–103.

[48] Benoıˆt Magnin et al. “Support vector machine-based classification of Alzheimer’s disease from whole-brain anatomical MRI”. In: *Neuroradiology* 51.2 (2009), pp. 73–83.

[49] *Microservices vs Monolithic Architecture*. URL: https://www.mulesoft.com/resources/api/microservices-vs-monolithic. (accessed: 18.04.2022).

[50] *Microservices vs. Monolith Architecture*. URL: https://dev.to/alex_barashkov/microservices-vs-monolith-architecture-4l1m. (accessed: 19.04.2022).

[51] *Microservices: What They Are and Why Use Them*. URL: https://www.leanix.net/en/blog/a-brief-history-of-microservices. (accessed: 18.04.2022).

[52] *Mini-Mental State Examination (MMSE)*. URL: https://oxfordmedicaleducation.com/geriatrics/mini-mental-state-examination-mmse/. (accessed: 26.05.2022).

[53] *Multicolinearite: definition*. URL: https://larmarange.github.io/analyse-R/multicolinearite.html. (accessed: 26.04.2022).

[54] J Neelaveni and MS Geetha Devasana. “Alzheimer disease prediction using machine learning algorithms”. In: *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*. IEEE. 2020, pp. 101–104.

[55] Rafaela Sanches de Oliveira et al. “Learning effect of computerized cognitive tests in older adults”. In: *Einstein (Sao Paulo)* 12 (2014), pp. 149–153.

[56] Francesco Panza et al. “Amyloid- immunotherapy for alzheimer disease: Is it now a long shot?” In: *Annals of Neurology* 85.3 (Jan. 2019), pp. 303–315. DOI: 10.1002/ana.25410. URL: https://doi.org/10.1002/ana.25410.

[57] *Parsimonious Model: Definition, Ways to Compare Models*. URL: https://www.statisticshowto.com/parsimonious-model/. (accessed: 26.04.2022).

[58] *Plaque amyloide : qu’est-ce que c’est ?* URL: https://www.futura-sciences.com/sante/definitions/biologie-plaque-amyloide-11751/. (accessed: 12.04.2022).

[59] Jackie M Poos et al. “Impairment of episodic memory in genetic frontotemporal dementia: A GENFI study”. In: *Alzheimer’s & Dementia: Diagnosis, Assessment & Disease Monitoring* 13.1 (2021), e12185.

[60] *Positron Emission Tomography Scan*. Aug. 2021. URL: https://www.mayoclinic.org/tests-procedures/pet-scan/about/pac-20385078.

[61] Lisa Quenon. *Rapport de stage. Centre de Revalidation Neuropsychologique.* Université Catholique de Louvain, 2011.

[62] Lisa Quenon et al. “Prediction of free and cued selective reminding test performance using volumetric and amyloid-based biomarkers of Alzheimer’s disease”. In: *Journal of the International Neuropsychological Society* 22.10 (2016), pp. 991–1004.

[63] Haanpa R.M.a et al. “The CERAD Neuropsychological Battery in Patients with Frontotemporal Lobar Degeneration”. In: (2015). URL: https://doi.org/10.1159/000380815.

[64] *REST APIs vs Microservices: The Differences and How They Work Together*. URL: https://blog.dreamfactory.com/restful-api-and-microservices-the-differences-and-how-they-work-together/. (accessed: 19.04.2022).

[65] *Ridge Regression: Simple Definition*. URL: https://www.statisticshowto.com/ridge-regression/. (accessed: 26.04.2022).

[66] Sargolzaei S. et al. “A practical guideline for intracranial volume estimation in patients with Alzheimer’s disease.” In: *BMC Bioinformatics* 16 (2015). DOI: https://doi.org/10.1186/1471-2105-16-S7-S8.

[67] N. Schuff et al. “MRI of hippocampal volume loss in early Alzheimer’s disease in relation to ApoE genotype and biomarkers”. In: *Brain* 132.4 (Feb. 2009), pp. 1067–1077. ISSN: 0006-8950. DOI: 10.1093/brain/awp007. eprint: https://academic.oup.com/brain/article-pdf/132/4/1067/16694927/awp007.pdf. URL: https://doi.org/10.1093/brain/awp007.

[68] Claude Elwood Shannon. *The mathematical theory of communication, by CE Shannon (and recent contributions to the mathematical theory of communication), W. Weaver*. University of illinois Press Champaign, IL, USA, 1949.

[69] *sklearn linear model: Lasso*. URL: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Lasso.html#sklearn.linear_model.Lasso. (accessed: 26.04.2022).

[70] *sklearn linear model: Ridge*. URL: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Ridge.html. (accessed: 26.04.2022).

[71] *SQLite vs MySQL vs PostgreSQL: A Comparison Of Relational Database Management Systems*. URL: https://www.digitalocean.com/community/tutorials/sqlite-vs-mysql-vs-postgresql-a-comparison-of-relational-database-management-systems. (accessed: 14.04.2022).

[72] *Trail Making Test*. URL: https://www.neura.edu.au/apps/trail-making-test/. (accessed: 28.03.2022).

[73] *Use containers to Build, Share and Run your applications*. URL: https://www.docker.com/resources/what-container/. (accessed: 19.04.2022).

[74] Lei Wang et al. “Changes in hippocampal volume and shape across time distinguish dementia of the Alzheimer type from healthy aging[U+2606]”. In: *NeuroImage* 20.2 (2003), pp. 667–682. ISSN: 1053-8119. DOI: https://doi.org/10.1016/S1053-8119(03)00361-6. URL: https://www.sciencedirect.com/science/article/pii/S1053811903003616.

[75] *What is a multi-layered perceptron?* URL: https://www.educative.io/edpresso/what-is-a-multi-layered-perceptron. (accessed: 26.04.2022).

[76] *What is MRI?* URL: https://www.nibib.nih.gov/science-education/science-topics/magnetic-resonance-imaging-mri. (accessed: 27.03.2022).

[77] *What is the Cerebral Cortex?* URL: https://www.simplypsychology.org/what-is-the-cerebral-cortex.html. (accessed: 19.04.2022).

[78] *What it takes to be correlated Cand how it could be interpreted for our analysis*. URL: https://towardsdatascience.com/what-it-takes-to-be-correlated-ce41ad0d8d7f. (accessed: 27.04.2022).

[79] HyunChul Youn et al. "Decreased Cortical Thickness and Local Gyrification in Individuals with Subjective Cognitive Impairment". In: *Clinical Psychopharmacology and Neuroscience* 19.4 (2021), p. 640.

# Part III

# Appendices

# Chapter 5

# Figures and graphs

## 5.1 Atlases

### 5.1.1 FreeSurfer[8]

1. Frontal
    - Superior Frontal
    - Rostral and Caudal Middle Frontal
    - Pars Opercularis, Pars Triangularis, and Pars Orbitalis
    - Lateral and Medial Orbitofrontal
    - Precentral
    - Paracentral
    - Frontal Pole
2. Parietal
    - Superior Parietal
    - Inferior Parietal
    - Supramarginal
    - Postcentral
    - Precuneus
3. Temporal
    - Superior, Middle, and Inferior Temporal
    - Banks of the Superior Temporal Sulcus
    - Fusiform
    - Transverse Temporal
    - Entorhinal
    - Temporal Pole
    - Parahippocampal

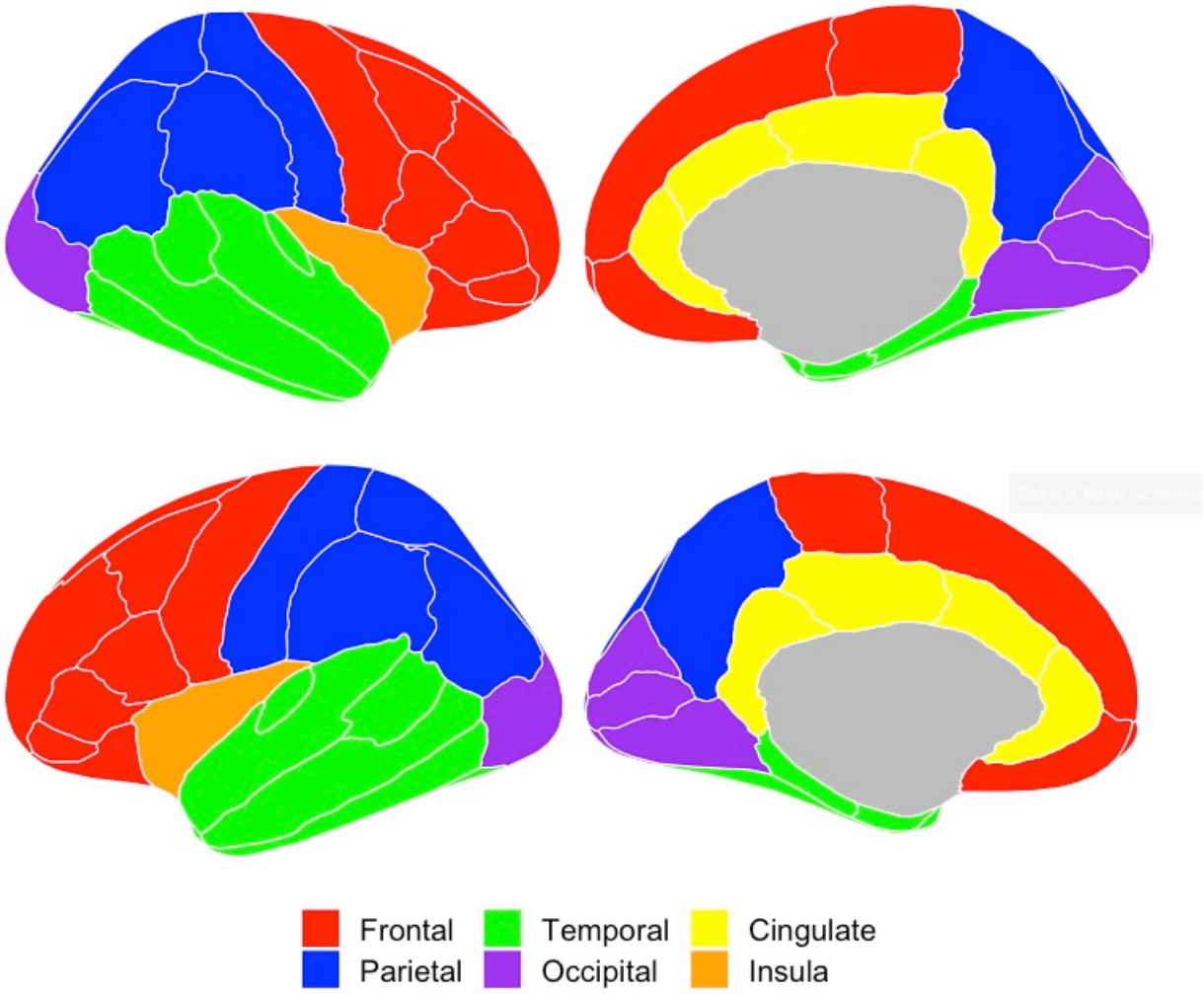



Figure 5.1: Cortical Parcellation (aparc)

4. Occipital
    - Lateral Occipital
    - Lingual
    - Cuneus
    - Pericalcarine
5. Cingulate (if you want to include in a lobe)
    - Rostral Anterior (Frontal)
    - Caudal Anterior (Frontal)
    - Posterior (Parietal)
    - Isthmus (Parietal)
6. Other

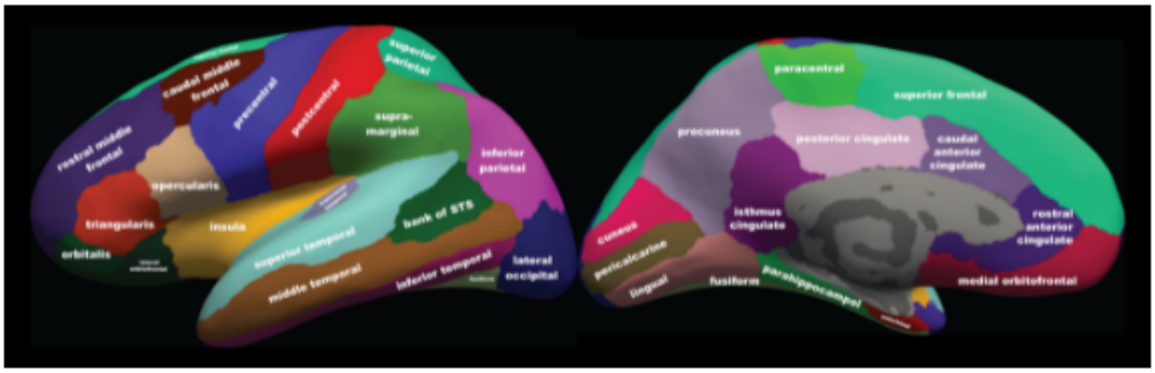



Figure 5.2: APARC Parcellations by Lobes

- Insula

## 5.2 Distributions of the composite scores evolution

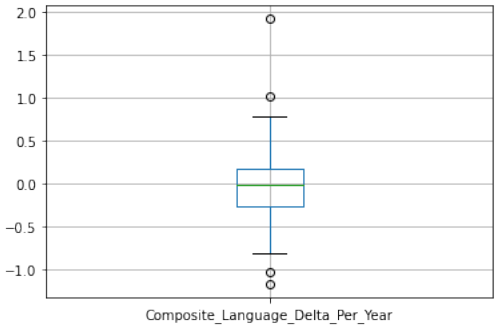



Figure 5.3: Language composite score with outliers

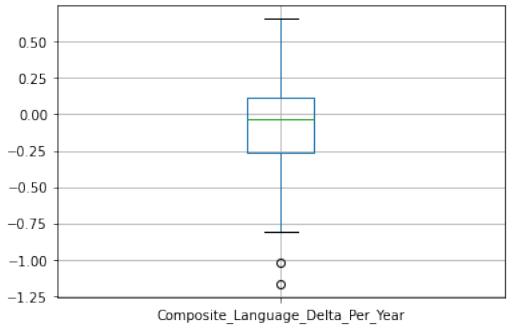



Figure 5.4: Language composite score without outliers

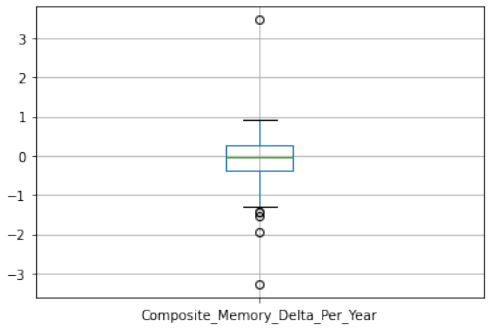



Figure 5.5: Memory composite score with outliers

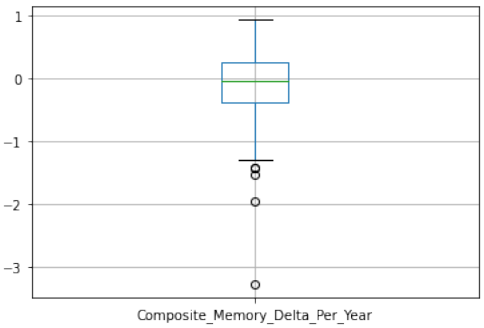



Figure 5.6: Memory composite score without outliers

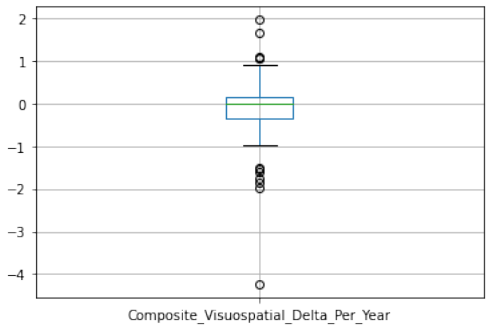



Figure 5.7: Visuospatial composite score with outliers

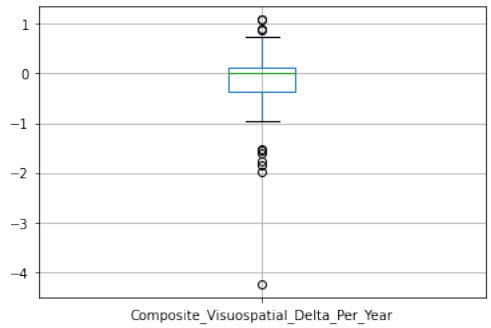



Figure 5.8: Visuospatial composite score without outliers

## 5.3 Simple train-test split results

### 5.3.1 Language

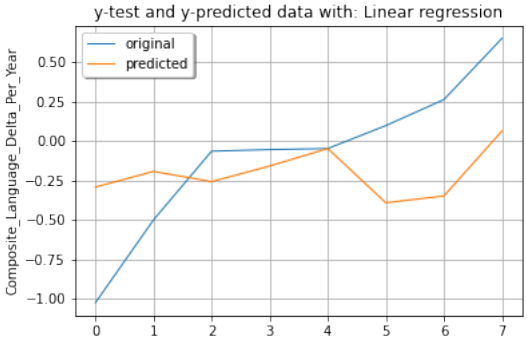



Figure 5.9: Language score prediction: linear regression

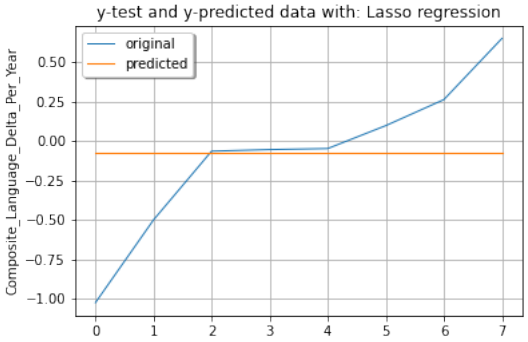



Figure 5.10: Language score prediction: lasso regression

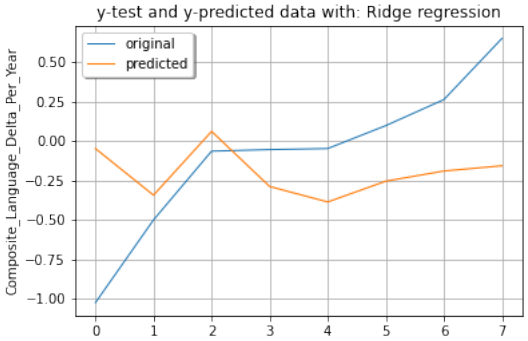



Figure 5.11: Language score prediction: ridge regression

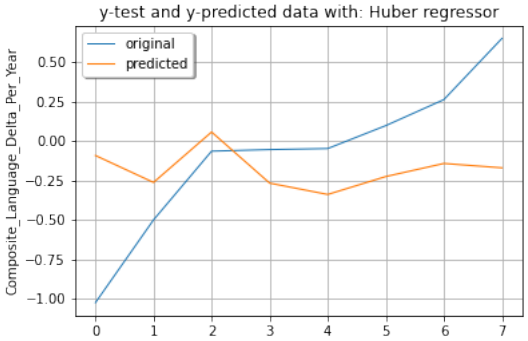



Figure 5.12: Language score prediction: huber regression

| | Model | RMSE | Std_in_pred |
|---|---|---|---|
| 0 | Linear regression | 0.452194 | 0.143489 |
| 1 | Lasso regression | 0.468057 | 0.000000 |
| 2 | Ridge regression | 0.517621 | 0.141069 |
| 3 | Huber regressor | 0.501425 | 0.115676 |
| 4 | Support vector Machine | 0.484329 | 0.092745 |
| 5 | Random forest | 0.491863 | 0.119123 |
| 6 | Gradient boosting regressor | 0.545341 | 0.146236 |
| 7 | Multi Layer Perceptron | 0.551395 | 0.189915 |

Figure 5.17: Results summary for simple train-test split for language composite score

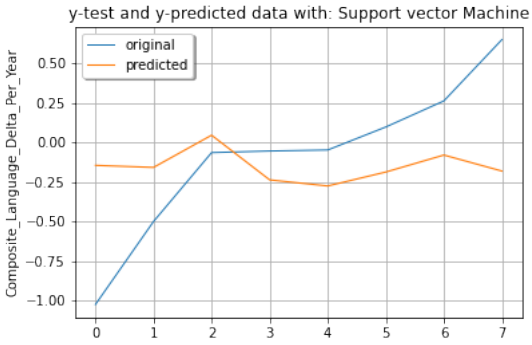



Figure 5.13: Language score prediction: support vector regression

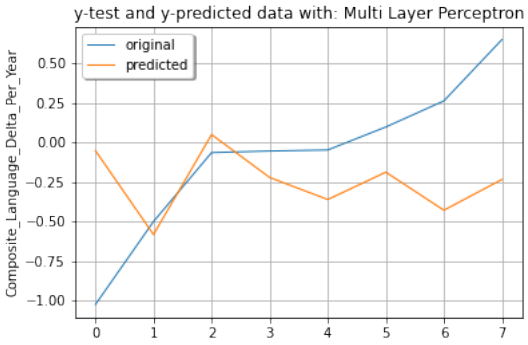



Figure 5.14: Language score prediction: Multi layer perceptron regression

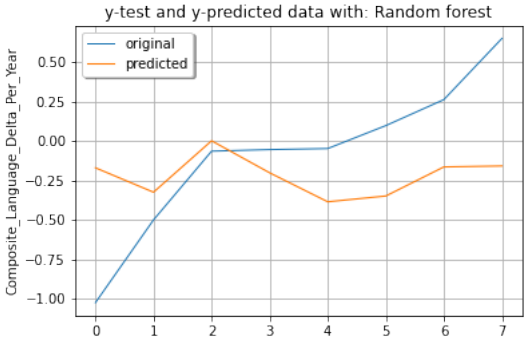



Figure 5.15: Language score prediction: random forest regression

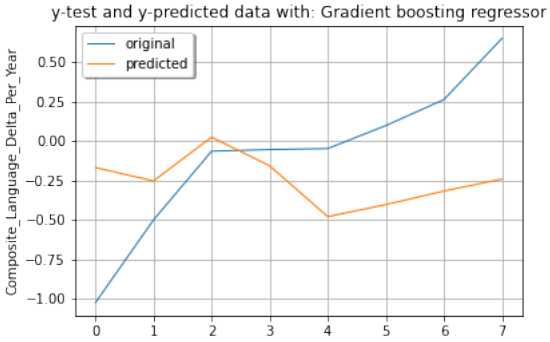



Figure 5.16: Language score prediction: Gradient boosting regression

### 5.3.2 Memory

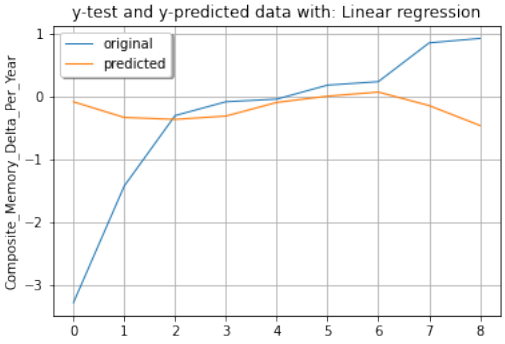



Figure 5.18: Memory score prediction: linear regression

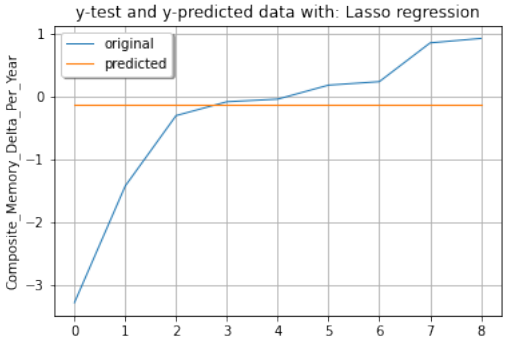



Figure 5.19: Memory score prediction: lasso regression

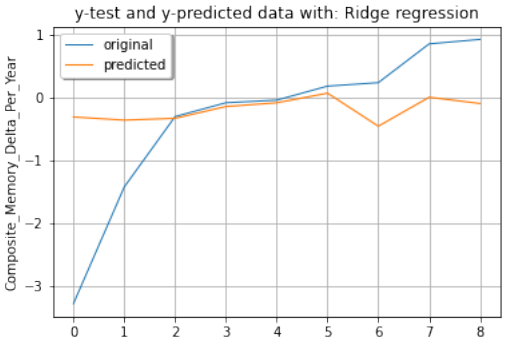



Figure 5.20: Memory score prediction: ridge regression

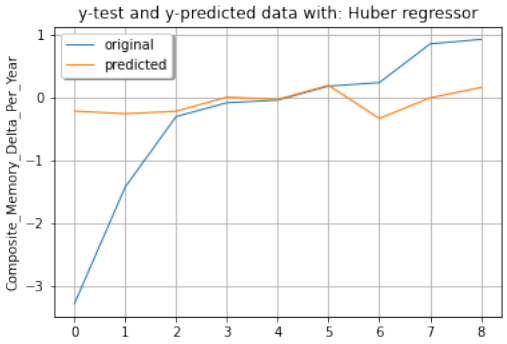



Figure 5.21: Memory score prediction: huber regression

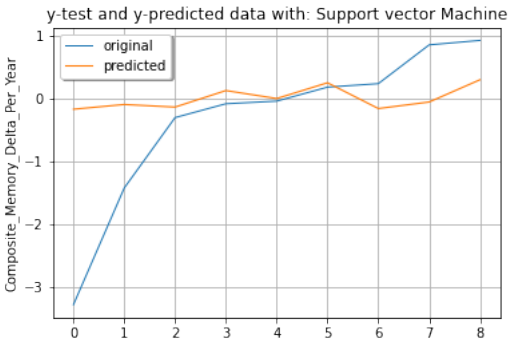



Figure 5.22: Memory score prediction: support vector regression

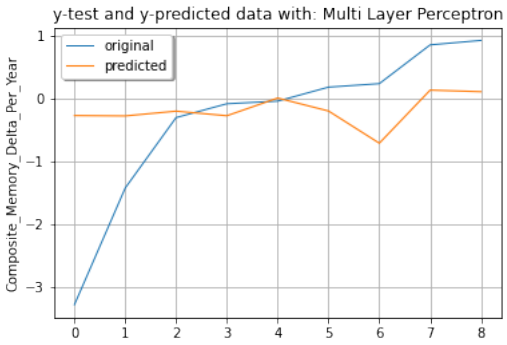



Figure 5.23: Memory score prediction: Multi layer perceptron regression

| | Model | RMSE | Std_in_pred |
|---|---|---|---|
| 0 | Linear regression | 1.264824 | 0.172252 |
| 1 | Lasso regression | 1.243501 | 0.000000 |
| 2 | Ridge regression | 1.163458 | 0.170249 |
| 3 | Huber regressor | 1.172847 | 0.176736 |
| 4 | Support vector Machine | 1.195227 | 0.167342 |
| 5 | Random forest | 1.141615 | 0.275529 |
| 6 | Gradient boosting regressor | 1.193352 | 0.304707 |
| 7 | Multi Layer Perceptron | 1.183758 | 0.241381 |

Figure 5.26: Results summary for simple train-test split for memory composite score

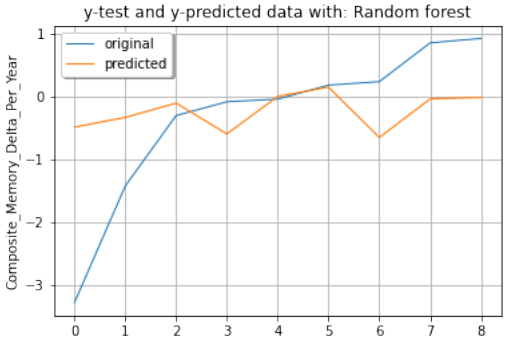



Figure 5.24: Memory score prediction: random forest regression

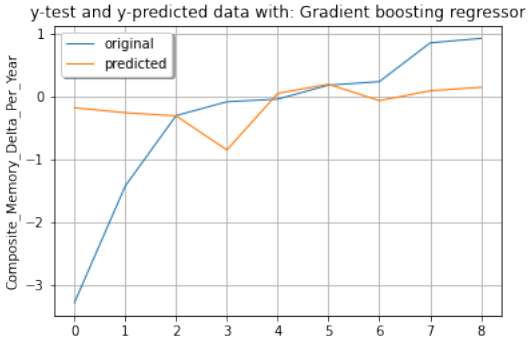



Figure 5.25: Memory score prediction: Gradient boosting regression

### 5.3.3 Visuospatial

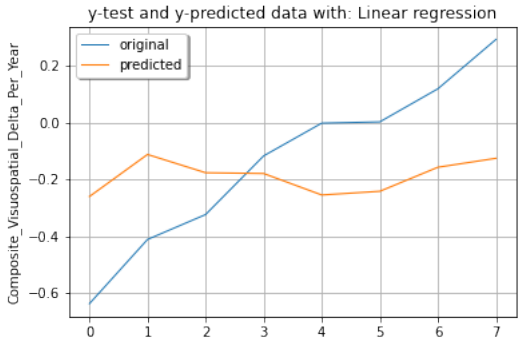



Figure 5.27: Visuospatial score prediction: linear regression

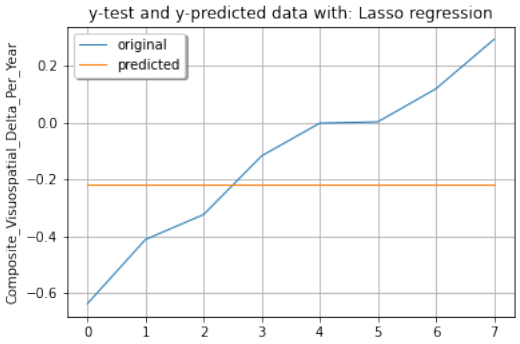



Figure 5.28: MemVisuospatialory score prediction: lasso regression

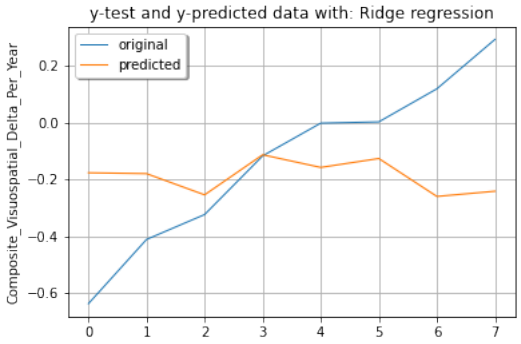



Figure 5.29: Visuospatial score prediction: ridge regression

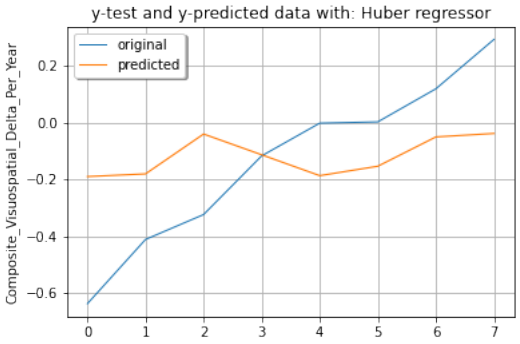



Figure 5.30: Visuospatial score prediction: huber regression

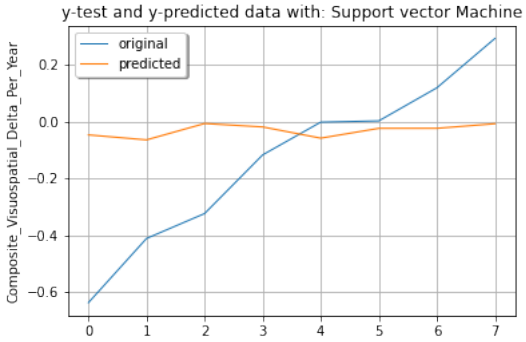



Figure 5.31: Visuospatial score prediction: support vector regression

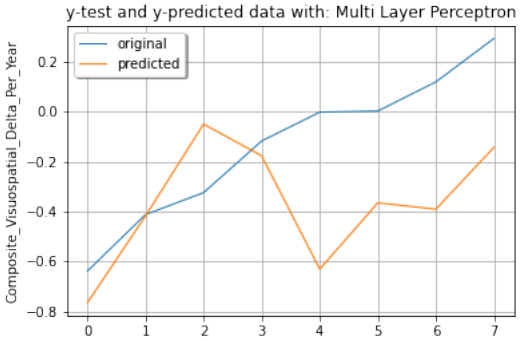



Figure 5.32: Visuospatial score prediction: Multi layer perceptron regression

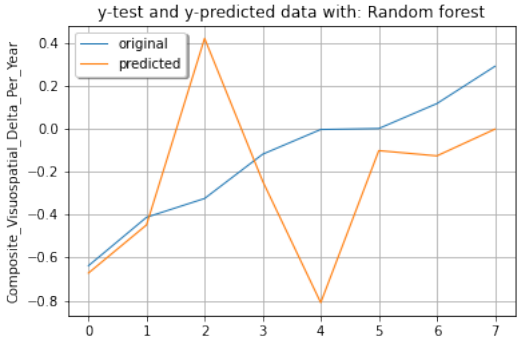



Figure 5.33: Visuospatial score prediction: random forest regression

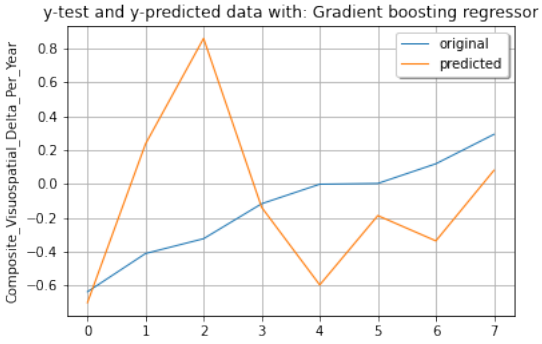



Figure 5.34: Visuospatial score prediction: Gradient boosting regression

| | Model | RMSE | Std_in_pred |
|---|---|---|---|
| 0 | Linear regression | 0.281150 | 0.054067 |
| 1 | Lasso regression | 0.297109 | 0.000000 |
| 2 | Ridge regression | 0.304315 | 0.053540 |
| 3 | Huber regressor | 0.257553 | 0.063322 |
| 4 | Support vector Machine | 0.294472 | 0.020933 |
| 5 | Random forest | 0.415900 | 0.367861 |
| 6 | Gradient boosting regressor | 0.555234 | 0.466339 |
| 7 | Multi Layer Perceptron | 0.366065 | 0.228176 |

Figure 5.35: Results summary for simple train-test split for visuospatial composite score

## 5.4 Leave one out results

### 5.4.1 Language

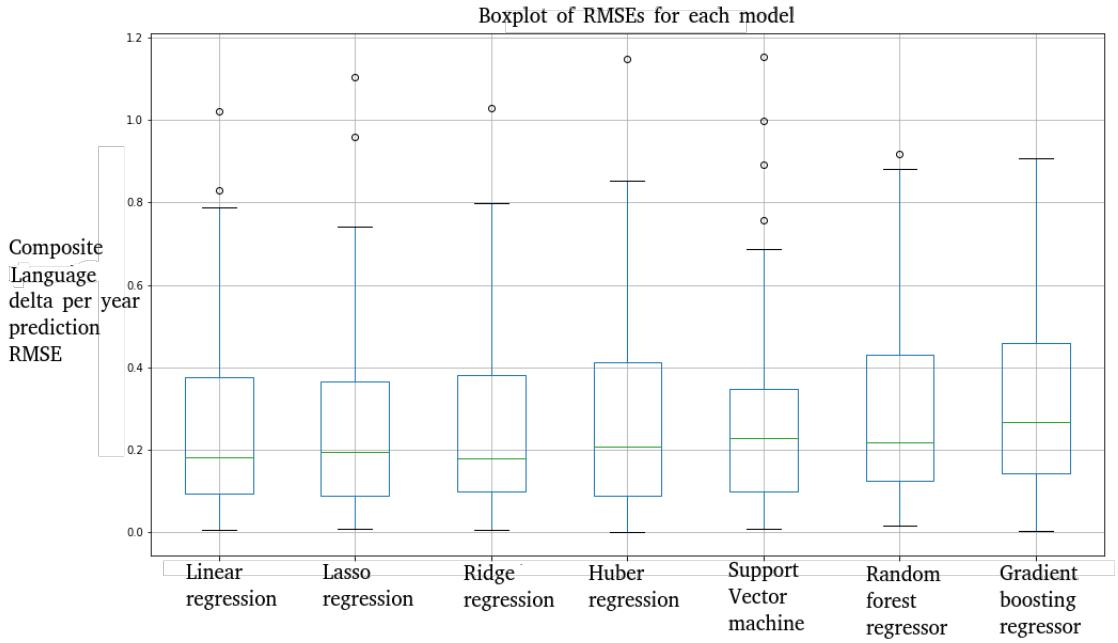



Figure 5.36: Leave-one-out results comparison for language composite score prediction

### 5.4.2 Memory

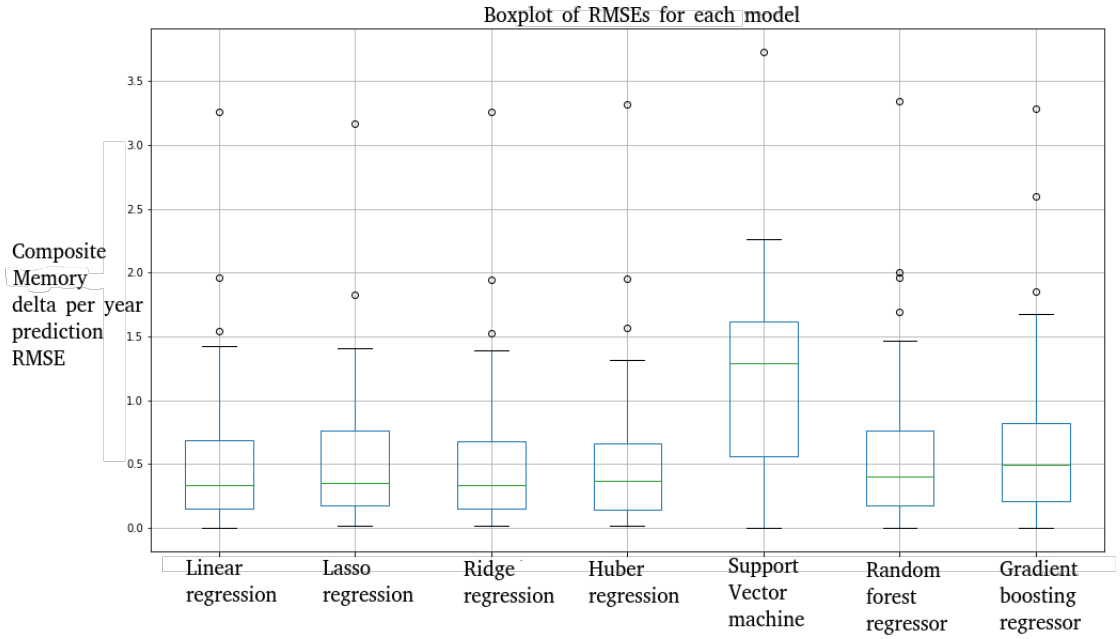



Figure 5.37: Leave-one-out results comparison for memory composite score prediction

### 5.4.3 Visuospatial

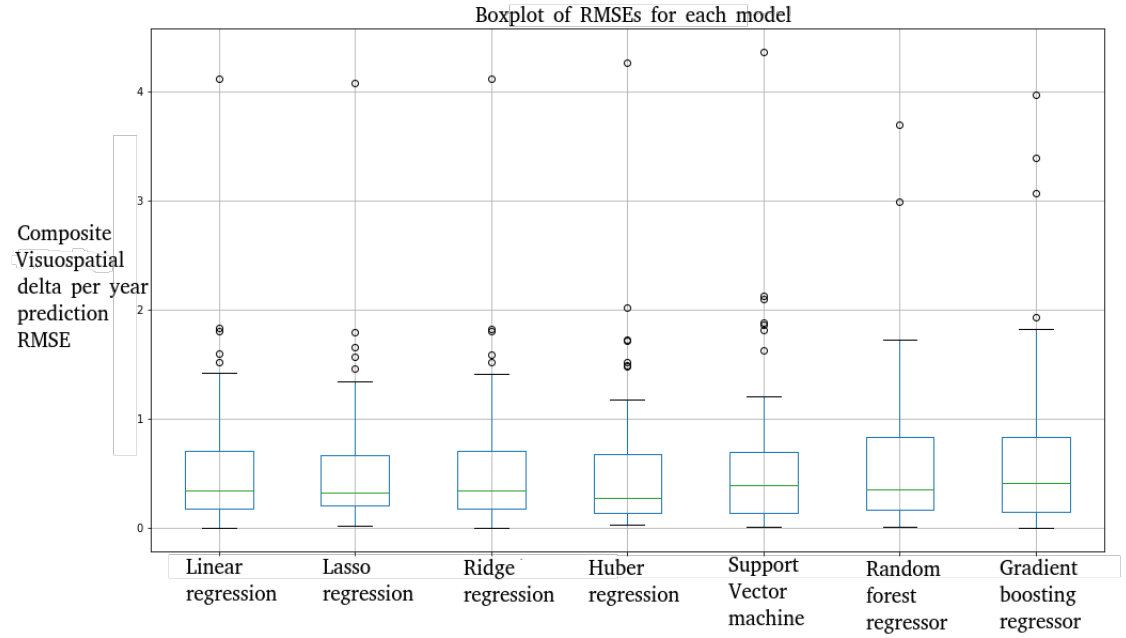



Figure 5.38: Leave-one-out results comparison for visuospatial composite score prediction

# Chapter 6

# Neuropsychological tests descriptions

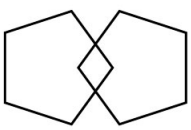
## 6.1 MMSE

**MINI MENTAL STATE EXAMINATION (MMSE)**

Name:

DOB:

Hospital Number:

| One point for each answer | DATE: | | |
|---|---|---|---|
| **ORIENTATION** Year Season Month Date Time | ...../ 5 | ...../ 5 | ...../ 5 |
| Country Town District Hospital Ward/Floor | ...../ 5 | ...../ 5 | ...../ 5 |
| **REGISTRATION** Examiner names three objects (e.g. apple, table, penny) and asks the patient to repeat (1 point for each correct. THEN the patient learns the 3 names repeating until correct). | ...../ 3 | ...../ 3 | ...../ 3 |
| **ATTENTION AND CALCULATION** Subtract 7 from 100, then repeat from result. Continue five times: 100, 93, 86, 79, 72, 65 (Alternative: spell "WORLD" backwards: DLROW). | ...../ 5 | ...../ 5 | ...../ 5 |
| **RECALL** Ask for the names of the three objects learned earlier. | ...../ 3 | ...../ 3 | ...../ 3 |
| **LANGUAGE** Name two objects (e.g. pen, watch). | ...../ 2 | ...../ 2 | ...../ 2 |
| Repeat "No ifs, ands, or buts". | ...../ 1 | ...../ 1 | ...../ 1 |
| Give a three-stage command. Score 1 for each stage. (e.g. "Place index finger of right hand on your nose and then on your left ear"). | ...../ 3 | ...../ 3 | ...../ 3 |
| Ask the patient to read and obey a written command on a piece of paper. The written instruction is: "Close your eyes". | ...../ 1 | ...../ 1 | ...../ 1 |
| Ask the patient to write a sentence. Score 1 if it is sensible and has a subject and a verb. | ...../ 1 | ...../ 1 | ...../ 1 |
| **COPYING:** Ask the patient to copy a pair of intersecting pentagons  | ...../ 1 | ...../ 1 | ...../ 1 |
| | **TOTAL:** | ...../ 30 | ...../ 30 | ...../ 30 |

**MMSE scoring**
24-30: no cognitive impairment
18-23: mild cognitive impairment
0-17: severe cognitive impairment

## 6.2 The test of Graphic Series [61]

### Presentation of the instrument

The Graphic Series Test was designed within the CRN (Centre de Revalidation Neuropsychologique) on the basis of the "Graphic Series" test that Luria had originally developed to test praxis functions (1966, cited by Bianconi and Busigny, 2005; Van Laethem, 2006, cited by Quenon, 2011). Since its creation, this tool has already been adapted twice, the first containing six series, while the second and current version includes two additional series (Van Laethem, 2006, cited by Quenon, 2011). The Graphical Series Test is used to probe executive functioning and, to a lesser extent, to assess the degree of integrity of the praxis (Bianconi et al., 2005; Van Laethem, 2006, cited by Quenon, 2011). In particular, planning, inhibition and flexibility abilities are investigated at the executive level (Bianconi et al., 2005; Van Laethem, 2006, cited by Quenon, 2011). The interest of this test lies in its response mode, which is motor(Van Laethem, 2006, cited by Quenon, 2011)

### Mode of administration

The Graph Series Test is a "paper and pencil" test presented on two horizontally oriented sheets (Van Laethem, 2006, cited by Quenon, 2011). On each sheet, four series are sketched out, made up of elements characterised by a certain logical alternation (Bianconi et al., 2005; Van Laethem, 2006, cited by Quenon, 2011). Five of the eight series are discontinuous, while the remaining three are continuous, and are to be performed without lifting the pencil from the sheet (Bianconi et al., 2005; Van Laethem, 2006, cited by Quenon, 2011). In this test, the patient is invited to complete the different series by reproducing the elements as in the sketches provided (Bianconi et al., 2005; Van Laethem, 2006, cited by Quenon, 2011). No constraint is imposed in terms of speed of execution, as this tool is primarily concerned with the quality and accuracy of the subject's productions; however, time is measured as an indication (Bianconi et al., 2005; Van Laethem, 2006, cited by Quenon, 2011).

### Scoring

The performances demonstrated in the Graph Series Test are analyzed both quantitatively and qualitatively (Van Laethem, 2006, cited by Quenon, 2011). At the qualitative level, different types of errors can be distinguished, and these are recorded in a grid to help classify them (Van Laethem, 2006, cited by Quenon, 2011). A first group of errors includes so-called sequence errors, which are thought to be related to executive disorders (Bianconi et al., 2005; Van Laethem, 2006, cited by Quenon, 2011). A second group includes graphism errors, which are attributed to praxis or motor difficulties (Bianconi et al., 2005; Van Laethem, 2006, cited by Quenon, 2011). A final set of errors concerns global errors, which are thought to result from mixed executive and praxis disorders (Bianconi et al., 2005; Van Laethem, 2006, cited by Quenon, 2011). At the quantitative level, only sequence and global errors are penalized, as the evaluation of graphism errors excessively involves the subjectivity of the examiner (Bianconi et al., 2005; Van Laethem, 2006, cited by Quenon, 2011). Each series is scored out of four points. Sequence errors result in a deduction of one point per error and half a point in

the case of self-correction by the subject (Bianconi et al., 2005; Van Laethem, 2006, cited by Quenon, 2011). Global errors are penalized by a loss of 4 points. The scores obtained for each series are then summed to arrive at a total of 32 points (Bianconi et al., 2005; Van Laethem, 2006, cited by Quenon, 2011).

# Chapter 7

# Graphical User Interface

## 7.1 Database user interface

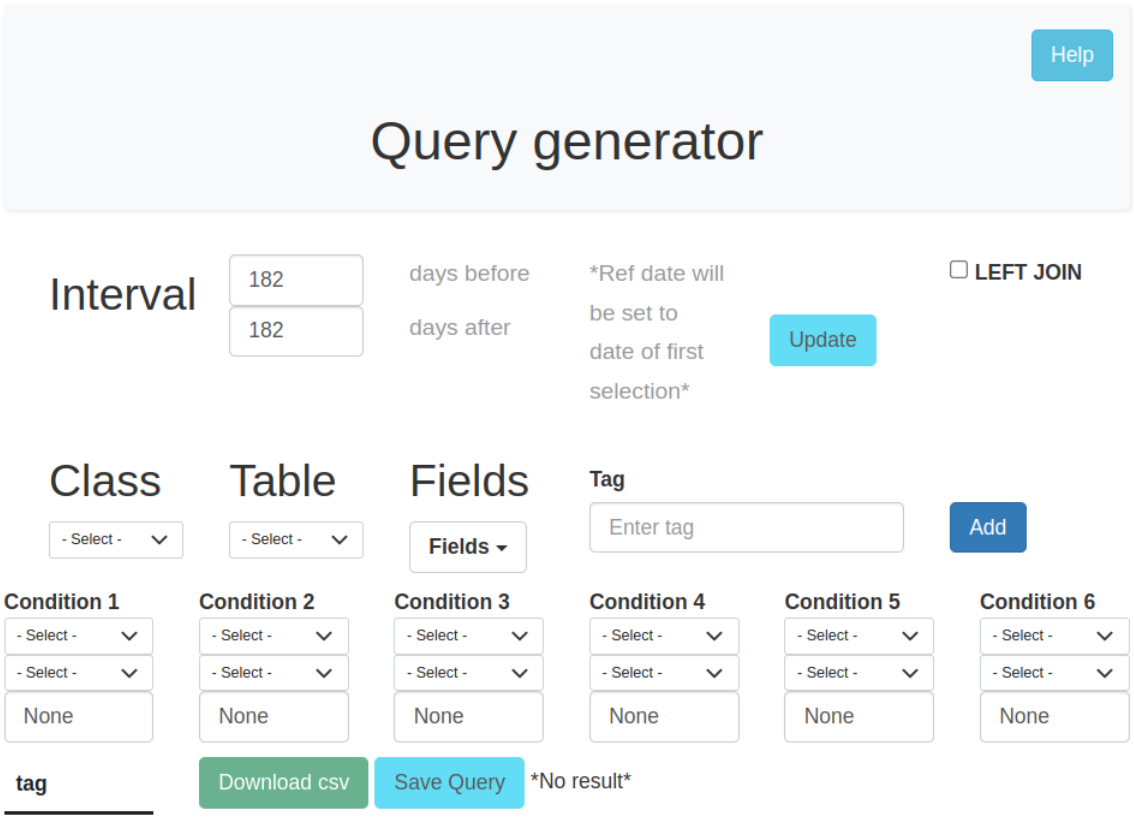



Figure 7.1: Screenshot of the Query Generator functionality

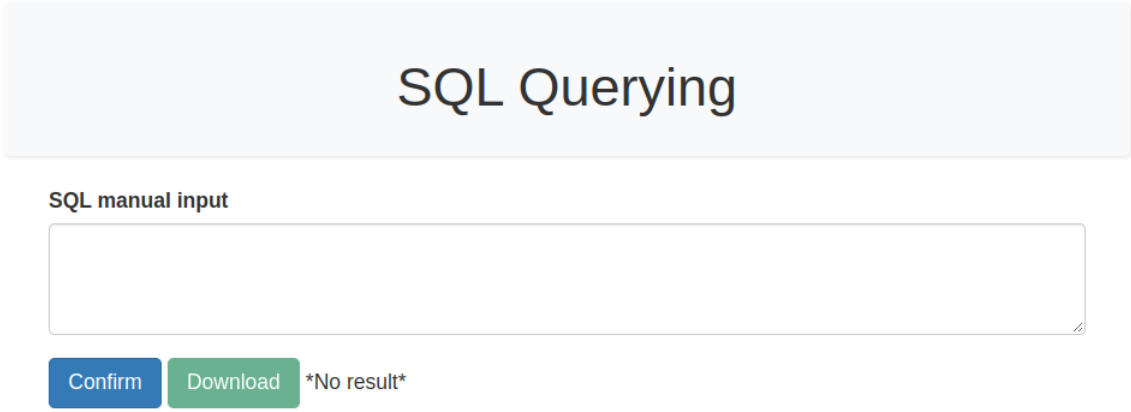



Figure 7.2: Screenshot of the SQL querying functionality

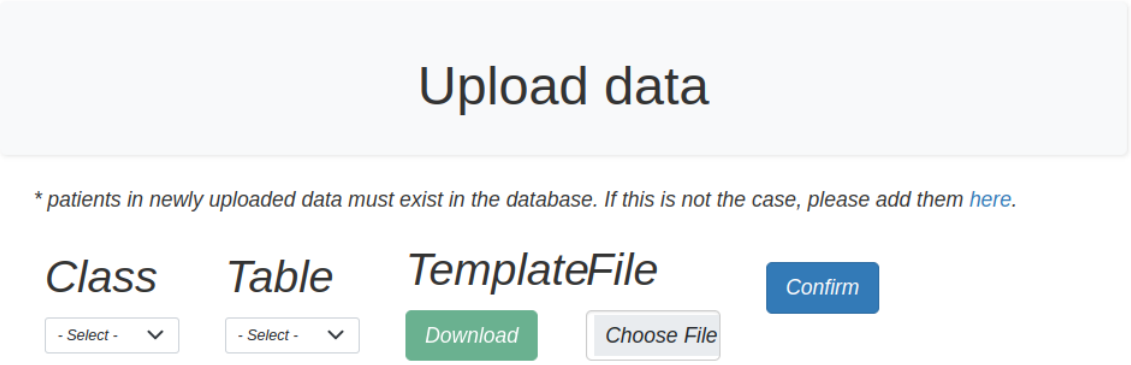



Figure 7.3: Screenshot of the Upload data functionality

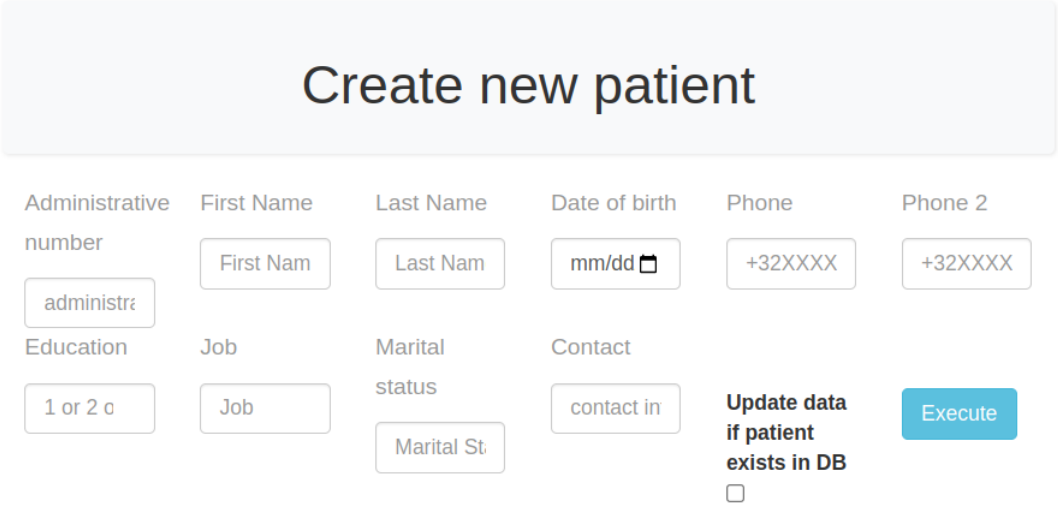



Figure 7.4: Screenshot of the Create patients functionality

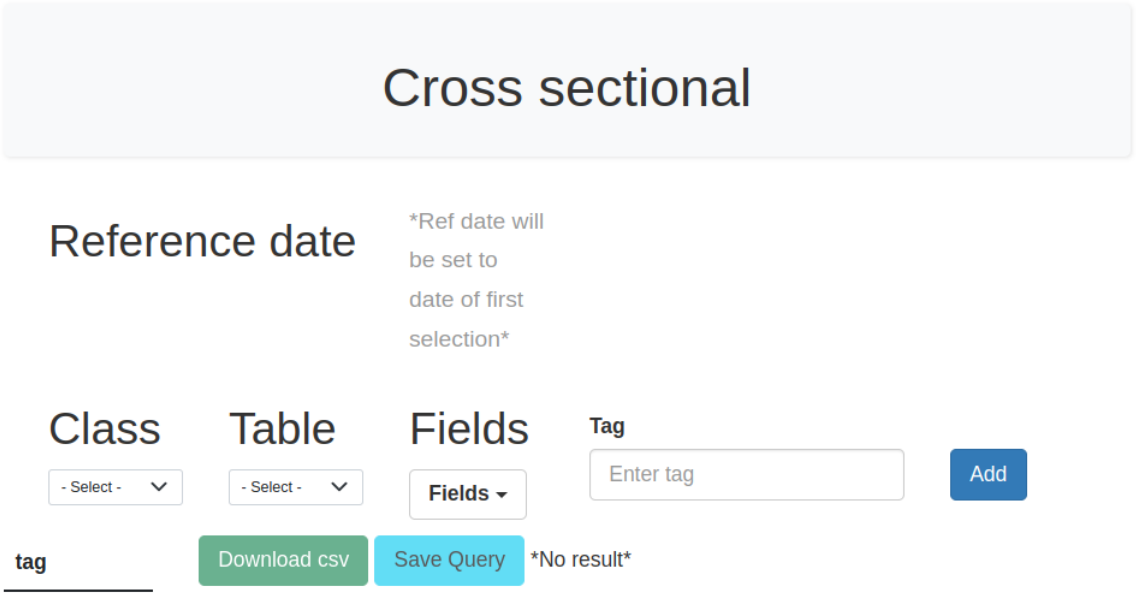



Figure 7.5: Screenshot of the Cross sectional functionality

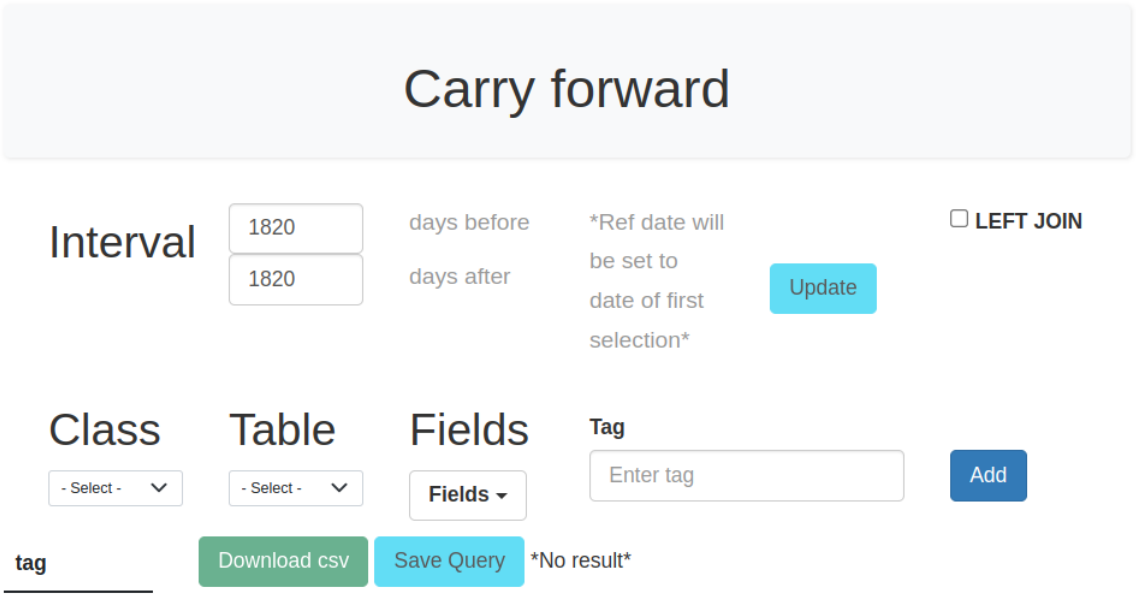



Figure 7.6: Screenshot of the Carry forward functionality

# 7.2 Continuous learning user interface

## 7.2.1 General Information

Machine Learning Report

General information (GeneralInfo.html)

First performance (First_Performance.html)

Recursive feature elimination (RFECV.html)

Final Models (FinalModels.html)

Sections

Executive

Language

Memory

Visuospatial

### General information

This page is dedicated to the analysis of the dataset.

Previous iteration

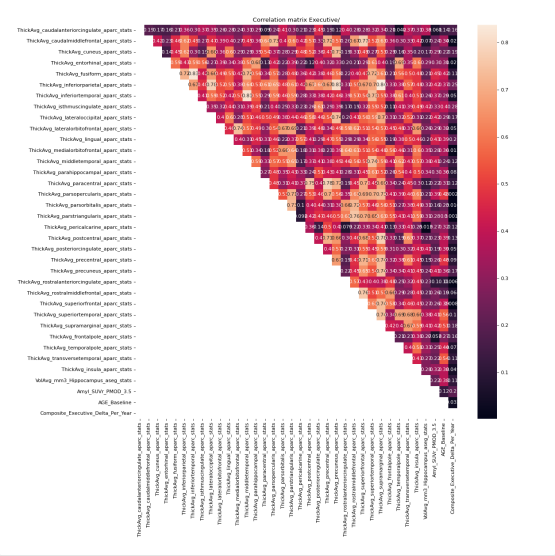
A correlation matrix and mutual information matrix have been computed in order to reflect the possible relations between the different input features.

### 1) Executive

This section is dedicated to the executive composite dataset.

#### 1.a) Correlation matrix

The pearson correlation matrix has been computed. For more inforation see: pandas.DataFrame.corr (https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.corr.html)




1.a.1) Top-10 correlated features in input

| | | 0 |
|---|---|---|
| ThickAvg_inferiorparietal_aparc_stats | ThickAvg_supramarginal_aparc_stats | 0.837981 |
| ThickAvg_fusiform_aparc_stats | ThickAvg_inferiortemporal_aparc_stats | 0.826438 |
| ThickAvg_inferiortemporal_aparc_stats | ThickAvg_middletemporal_aparc_stats | 0.813116 |
| ThickAvg_inferiorparietal_aparc_stats | ThickAvg_precuneus_aparc_stats | 0.806829 |
| ThickAvg_paracentral_aparc_stats | ThickAvg_precentral_aparc_stats | 0.779090 |
| ThickAvg_inferiorparietal_aparc_stats | ThickAvg_lateraloccipital_aparc_stats | 0.776811 |
| ThickAvg_postcentral_aparc_stats | ThickAvg_supramarginal_aparc_stats | 0.775321 |
| ThickAvg_caudalmiddlefrontal_aparc_stats | ThickAvg_superiorfrontal_aparc_stats | 0.772931 |
| ThickAvg_rostralmiddlefrontal_aparc_stats | ThickAvg_superiorfrontal_aparc_stats | 0.760930 |
| ThickAvg_parstriangularis_aparc_stats | ThickAvg_rostralmiddlefrontal_aparc_stats | 0.755676 |

#### 1.b) Mutual information matrix

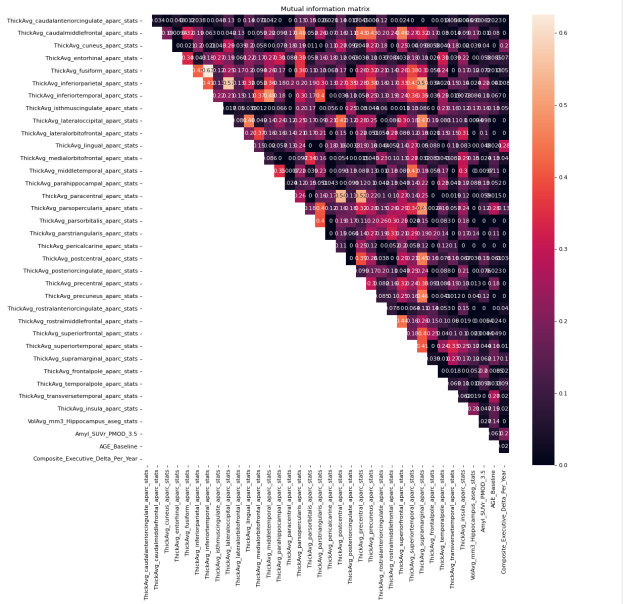
The mutual information matrix has been computed. For more inforation see: sklearn.metrics.mutual_info_score (https://scikit-learn.org/stable/modules/generated/sklearn.metrics.mutual_info_score.html)

# Machine Learning Report



## Sections



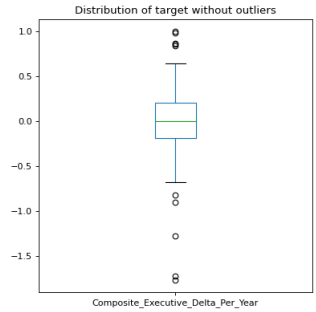



1.b.1) Top-10 correlated features in input

| | | 0 |
|---|---|---|
| **ThickAvg_fusiform_aparc_stats** | **ThickAvg_inferiortemporal_aparc_stats** | 0.625671 |
| **ThickAvg_inferiorparietal_aparc_stats** | **ThickAvg_lateraloccipital_aparc_stats** | 0.570197 |
| **ThickAvg_paracentral_aparc_stats** | **ThickAvg_postcentral_aparc_stats** | 0.538203 |
| **ThickAvg_inferiorparietal_aparc_stats** | **ThickAvg_supramarginal_aparc_stats** | 0.527238 |
| **ThickAvg_paracentral_aparc_stats** | **ThickAvg_precentral_aparc_stats** | 0.524435 |
| **ThickAvg_inferiortemporal_aparc_stats** | **ThickAvg_middletemporal_aparc_stats** | 0.487700 |
| **ThickAvg_parsopercularis_aparc_stats** | **ThickAvg_supramarginal_aparc_stats** | 0.485804 |
| **ThickAvg_lateraloccipital_aparc_stats** | **ThickAvg_supramarginal_aparc_stats** | 0.463497 |
| **ThickAvg_precuneus_aparc_stats** | **ThickAvg_supramarginal_aparc_stats** | 0.457346 |
| **ThickAvg_caudalmiddlefrontal_aparc_stats** | **ThickAvg_superiorfrontal_aparc_stats** | 0.455665 |

### 1.c) Target distribution

The target distribution can been seen below. Outliers have been deleted from the original target distribution.




## 2) Language

This section is dedicated to the language composite dataset.

### 2.a) Correlation matrix

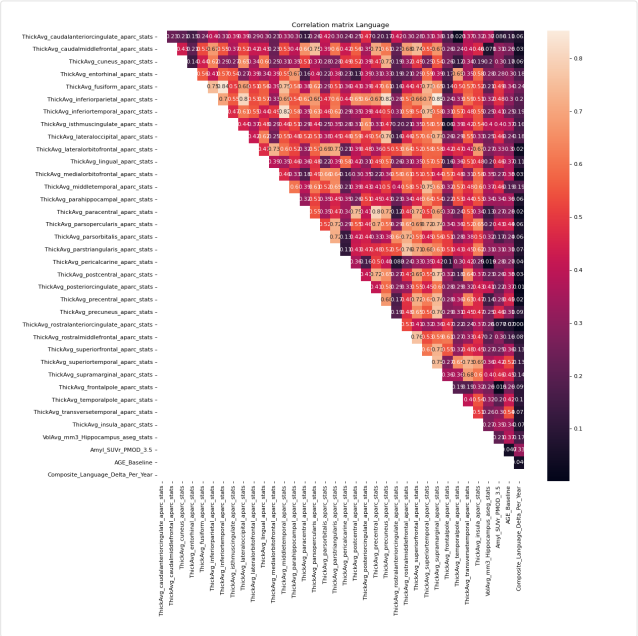
The pearson correlation matrix has been computed. For more inforation see: pandas.DataFrame.corr (https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.corr.html)

# Machine Learning Report



## Sections



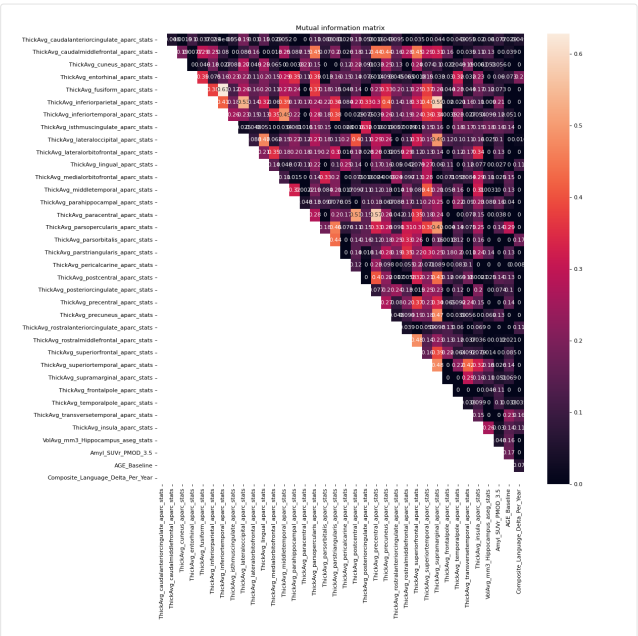



2.a.1) Top-10 correlated features in input

| | | 0 |
|---|---|---|
| ThickAvg_inferiorparietal_aparc_stats | ThickAvg_supramarginal_aparc_stats | 0.852694 |
| ThickAvg_fusiform_aparc_stats | ThickAvg_inferiortemporal_aparc_stats | 0.841978 |
| ThickAvg_inferiortemporal_aparc_stats | ThickAvg_middletemporal_aparc_stats | 0.824203 |
| ThickAvg_inferiorparietal_aparc_stats | ThickAvg_precuneus_aparc_stats | 0.823345 |
| | ThickAvg_lateraloccipital_aparc_stats | 0.803045 |
| ThickAvg_paracentral_aparc_stats | ThickAvg_precentral_aparc_stats | 0.796448 |
| ThickAvg_postcentral_aparc_stats | ThickAvg_supramarginal_aparc_stats | 0.769055 |
| ThickAvg_parstriangularis_aparc_stats | ThickAvg_rostralmiddlefrontal_aparc_stats | 0.763004 |
| ThickAvg_precuneus_aparc_stats | ThickAvg_supramarginal_aparc_stats | 0.762823 |
| ThickAvg_rostralmiddlefrontal_aparc_stats | ThickAvg_superiorfrontal_aparc_stats | 0.762679 |

## 2.b) Mutual information matrix

The mutual information matrix has been computed. For more inforation see: sklearn.metrics.mutual_info_score (https://scikit-learn.org/stable/modules/generated/sklearn.metrics.mutual_info_score.html)

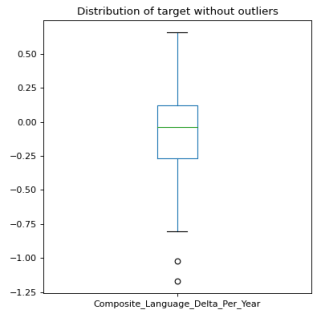



2.b.1) Top-10 correlated features in input

| | | 0 |
|---|---|---|
| ThickAvg_fusiform_aparc_stats | ThickAvg_inferiortemporal_aparc_stats | 0.633617 |
| ThickAvg_inferiorparietal_aparc_stats | ThickAvg_supramarginal_aparc_stats | 0.586514 |
| ThickAvg_paracentral_aparc_stats | ThickAvg_precentral_aparc_stats | 0.574300 |
| ThickAvg_inferiorparietal_aparc_stats | ThickAvg_lateraloccipital_aparc_stats | 0.532272 |
| ThickAvg_paracentral_aparc_stats | ThickAvg_postcentral_aparc_stats | 0.506194 |
| ThickAvg_parsopercularis_aparc_stats | ThickAvg_supramarginal_aparc_stats | 0.493349 |
| ThickAvg_lateraloccipital_aparc_stats | ThickAvg_supramarginal_aparc_stats | 0.490693 |
| ThickAvg_inferiortemporal_aparc_stats | ThickAvg_middletemporal_aparc_stats | 0.488378 |
| ThickAvg_rostralmiddlefrontal_aparc_stats | ThickAvg_superiorfrontal_aparc_stats | 0.477894 |
| ThickAvg_superiortemporal_aparc_stats | ThickAvg_supramarginal_aparc_stats | 0.476941 |

### 2.c) Target distribution

The target distribution can been seen below. Outliers have been deleted from the original target distribution.

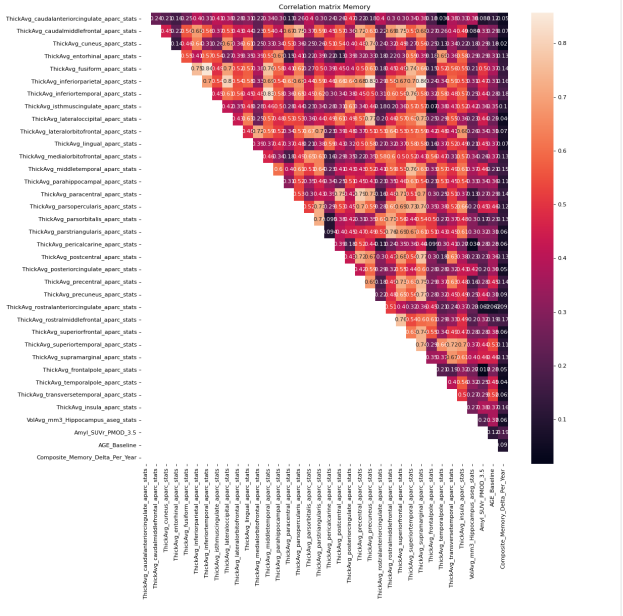



## 3) Memory

This section is dedicated to the memory composite dataset.

### 3.a) Correlation matrix

The pearson correlation matrix has been computed. For more inforation see: pandas.DataFrame.corr (https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.corr.html)

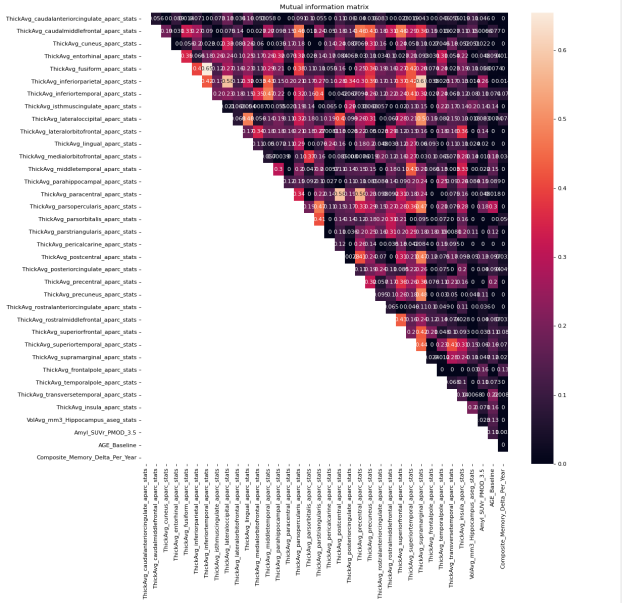



3.a.1) Top-10 correlated features in input

| | | 0 |
|---|---|---|
| ThickAvg_inferiorparietal_aparc_stats | ThickAvg_supramarginal_aparc_stats | 0.857206 |
| ThickAvg_fusiform_aparc_stats | ThickAvg_inferiortemporal_aparc_stats | 0.844152 |
| ThickAvg_inferiorparietal_aparc_stats | ThickAvg_precuneus_aparc_stats | 0.828178 |
| ThickAvg_inferiortemporal_aparc_stats | ThickAvg_middletemporal_aparc_stats | 0.826074 |
| ThickAvg_inferiorparietal_aparc_stats | ThickAvg_lateraloccipital_aparc_stats | 0.798323 |
| ThickAvg_paracentral_aparc_stats | ThickAvg_precentral_aparc_stats | 0.791516 |
| ThickAvg_lateraloccipital_aparc_stats | ThickAvg_precuneus_aparc_stats | 0.772212 |
| ThickAvg_postcentral_aparc_stats | ThickAvg_supramarginal_aparc_stats | 0.768970 |
| ThickAvg_precuneus_aparc_stats | ThickAvg_supramarginal_aparc_stats | 0.768191 |
| ThickAvg_fusiform_aparc_stats | ThickAvg_middletemporal_aparc_stats | 0.762661 |

## 3.b) Mutual information matrix

The mutual information matrix has been computed. For more inforation see: sklearn.metrics.mutual_info_score (https://scikit-learn.org/stable/modules/generated/sklearn.metrics.mutual_info_score.html)

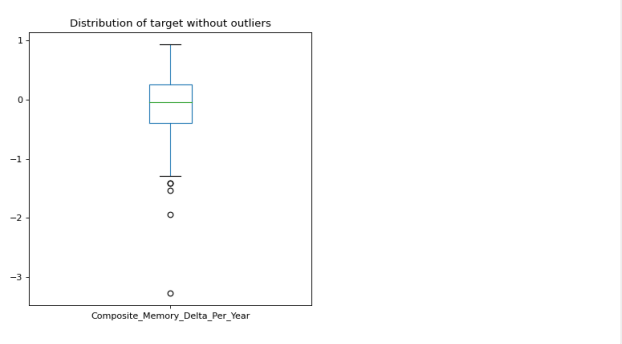



### 3.b.1) Top-10 correlated features in input

| | | 0 |
|---|---|---|
| ThickAvg_fusiform_aparc_stats | ThickAvg_inferiortemporal_aparc_stats | 0.657283 |
| ThickAvg_inferiorparietal_aparc_stats | ThickAvg_supramarginal_aparc_stats | 0.609092 |
| ThickAvg_paracentral_aparc_stats | ThickAvg_postcentral_aparc_stats | 0.584207 |
| ThickAvg_inferiorparietal_aparc_stats | ThickAvg_lateraloccipital_aparc_stats | 0.582713 |
| ThickAvg_paracentral_aparc_stats | ThickAvg_precentral_aparc_stats | 0.556455 |
| ThickAvg_lateraloccipital_aparc_stats | ThickAvg_supramarginal_aparc_stats | 0.499531 |
| ThickAvg_precuneus_aparc_stats | ThickAvg_supramarginal_aparc_stats | 0.482534 |
| ThickAvg_lateraloccipital_aparc_stats | ThickAvg_lingual_aparc_stats | 0.478070 |
| ThickAvg_parsopercularis_aparc_stats | ThickAvg_supramarginal_aparc_stats | 0.474768 |
| ThickAvg_inferiortemporal_aparc_stats | ThickAvg_middletemporal_aparc_stats | 0.472245 |

## 3.c) Target distribution

The target distribution can been seen below. Outliers have been deleted from the original target distribution.

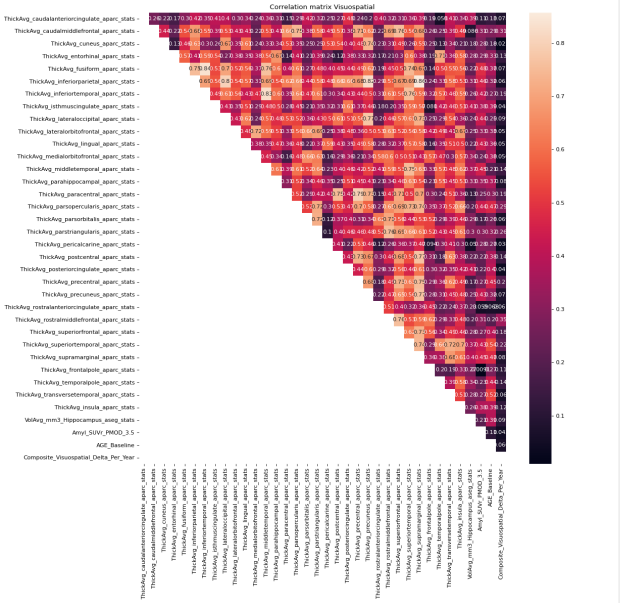


# 4) Visuospatial

This section is dedicated to the visuospatial composite dataset.

## 4.a) Correlation matrix

The pearson correlation matrix has been computed. For more inforation see: pandas.DataFrame.corr (https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.corr.html)




4.a.1) Top-10 correlated features in input

| | | 0 |
|---|---|---|
| ThickAvg_inferiorparietal_aparc_stats | ThickAvg_supramarginal_aparc_stats | 0.857062 |
| ThickAvg_fusiform_aparc_stats | ThickAvg_inferiortemporal_aparc_stats | 0.837162 |
| ThickAvg_inferiortemporal_aparc_stats | ThickAvg_middletemporal_aparc_stats | 0.826310 |
| ThickAvg_inferiorparietal_aparc_stats | ThickAvg_precuneus_aparc_stats | 0.824770 |
| | ThickAvg_lateraloccipital_aparc_stats | 0.799135 |
| ThickAvg_paracentral_aparc_stats | ThickAvg_precentral_aparc_stats | 0.791872 |
| ThickAvg_lateraloccipital_aparc_stats | ThickAvg_precuneus_aparc_stats | 0.774098 |
| ThickAvg_postcentral_aparc_stats | ThickAvg_supramarginal_aparc_stats | 0.766191 |
| ThickAvg_precuneus_aparc_stats | ThickAvg_supramarginal_aparc_stats | 0.765082 |
| ThickAvg_fusiform_aparc_stats | ThickAvg_middletemporal_aparc_stats | 0.763310 |

## 4.b) Mutual information matrix

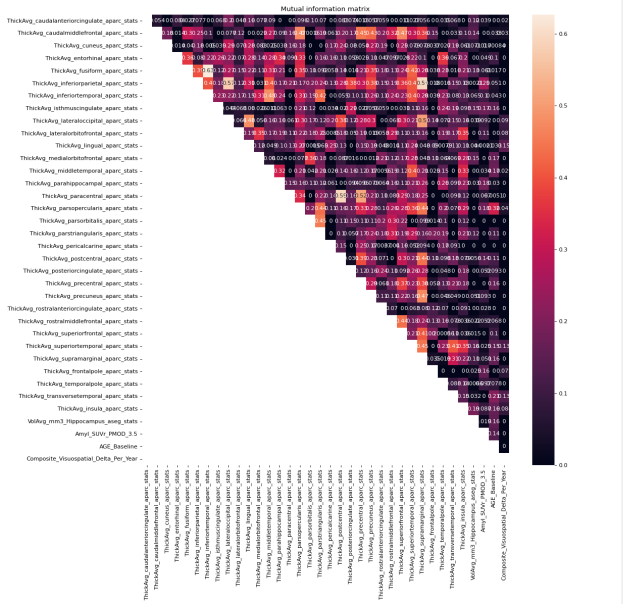
The mutual information matrix has been computed. For more inforation see: sklearn.metrics.mutual_info_score (https://scikit-learn.org/stable/modules/generated/sklearn.metrics.mutual_info_score.html)

# Machine Learning Report



## Sections



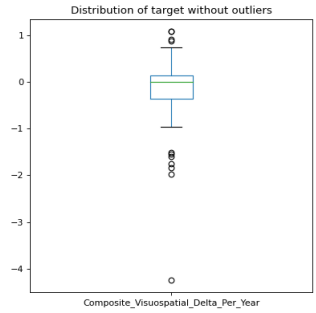



4.b.1) Top-10 correlated features in input

| | | 0 |
|---|---|---|
| ThickAvg_fusiform_aparc_stats | ThickAvg_inferiortemporal_aparc_stats | 0.621793 |
| ThickAvg_inferiorparietal_aparc_stats | ThickAvg_supramarginal_aparc_stats | 0.593855 |
| ThickAvg_paracentral_aparc_stats | ThickAvg_postcentral_aparc_stats | 0.592505 |
| ThickAvg_inferiorparietal_aparc_stats | ThickAvg_lateraloccipital_aparc_stats | 0.572642 |
| ThickAvg_paracentral_aparc_stats | ThickAvg_precentral_aparc_stats | 0.515632 |
| ThickAvg_lateraloccipital_aparc_stats | ThickAvg_supramarginal_aparc_stats | 0.503618 |
| ThickAvg_inferiortemporal_aparc_stats | ThickAvg_middletemporal_aparc_stats | 0.481940 |
| ThickAvg_lateraloccipital_aparc_stats | ThickAvg_lingual_aparc_stats | 0.478833 |
| ThickAvg_caudalmiddlefrontal_aparc_stats | ThickAvg_superiorfrontal_aparc_stats | 0.475706 |
| ThickAvg_precuneus_aparc_stats | ThickAvg_supramarginal_aparc_stats | 0.474254 |

## 4.c) Target distribution

The target distribution can been seen below. Outliers have been deleted from the original target distribution.



# 7.2.2 First performances

Machine Learning Report

General information (GeneralInfo.html)

First performance (First_Performance.html)

Recursive feature elimination (RFECV.html)

Final Models (FinalModels.html)

Sections

Executive

Language

Memory

Visuospatial

## First performance

This page is dedicated to the report of the different models performance without before using recursive feature selection.

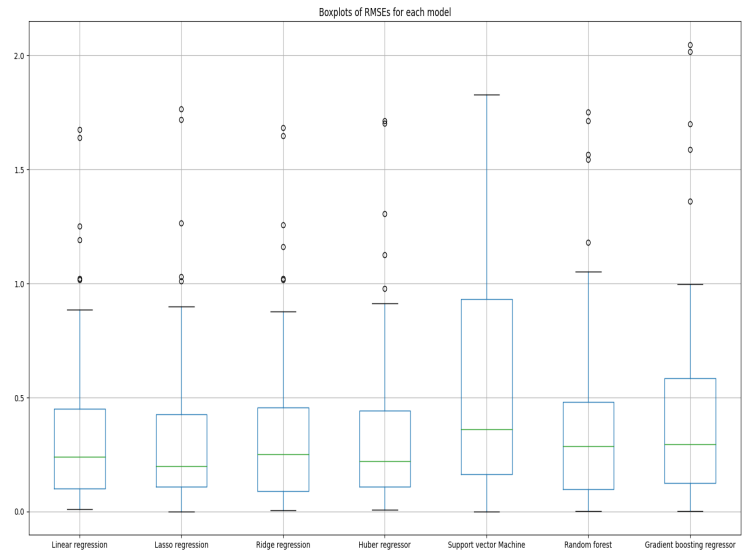
Previous iteration

The models have been trained and tested with the Leave-One-Out method. For more information see: A Quick Intro to Leave-One-Out Cross-Validation (LOOCV) (https://www.statology.org/leave-one-out-cross-validation/)

### 1) Executive

This section is dedicated to the executive composite results.

### 1.a) Performance box plots



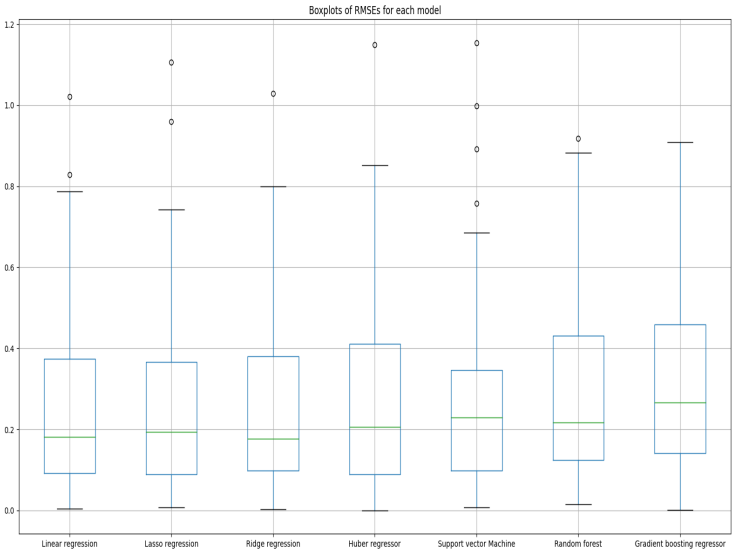

#### 1.a.1) Performance details

| | Model | RMSE_mean | RMSE_std |
|---|---|---|---|
| 0 | Linear regression | 0.351221 | 0.353743 |
| 1 | Lasso regression | 0.335874 | 0.356965 |
| 2 | Ridge regression | 0.349996 | 0.353484 |
| 3 | Huber regressor | 0.345691 | 0.355054 |
| 4 | Support vector Machine | 0.543750 | 0.465751 |
| 5 | Random forest | 0.392673 | 0.395058 |
| 6 | Gradient boosting regressor | 0.428884 | 0.435448 |

### 2) Language

This section is dedicated to the language composite results.

### 2.a) Performance box plots



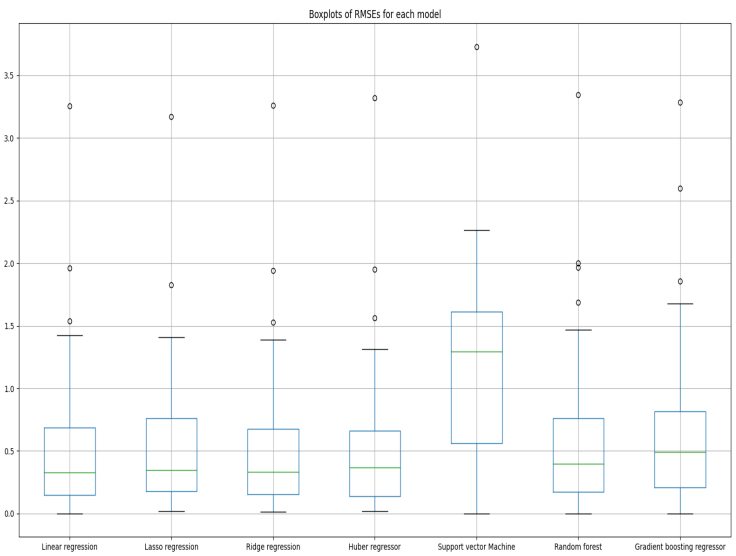

### 2.a.1) Performance details

| | Model | RMSE_mean | RMSE_std |
|---|---|---|---|
| **0** | Linear regression | 0.262017 | 0.225351 |
| **1** | Lasso regression | 0.267113 | 0.230551 |
| **2** | Ridge regression | 0.260902 | 0.225687 |
| **3** | Huber regressor | 0.265491 | 0.228717 |
| **4** | Support vector Machine | 0.278328 | 0.239073 |
| **5** | Random forest | 0.287726 | 0.205900 |
| **6** | Gradient boosting regressor | 0.306062 | 0.212298 |

## 3) Memory

This section is dedicated to the memory composite results.

## 3.a) Performance box plots



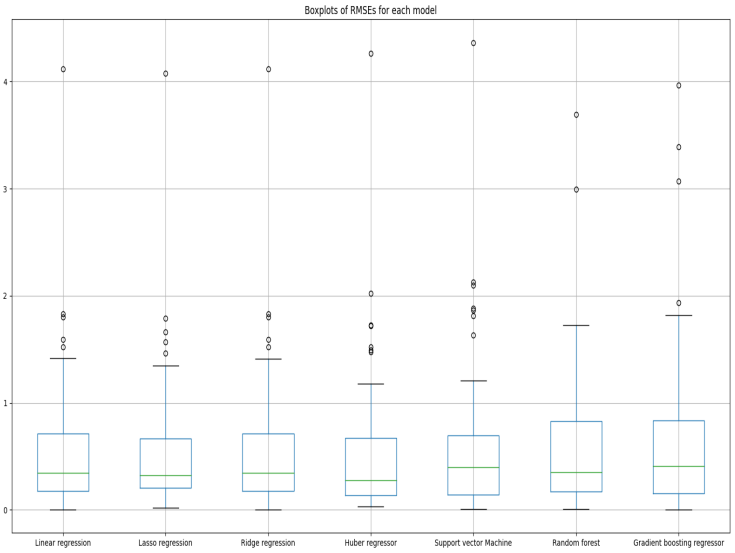
### 3.a.1) Performance details

| | Model | RMSE_mean | RMSE_std |
|---|---|---|---|
| 0 | Linear regression | 0.496424 | 0.512751 |
| 1 | Lasso regression | 0.507968 | 0.487798 |
| 2 | Ridge regression | 0.496178 | 0.511575 |
| 3 | Huber regressor | 0.481318 | 0.516156 |
| 4 | Support vector Machine | 1.198007 | 0.677941 |
| 5 | Random forest | 0.559347 | 0.546909 |
| 6 | Gradient boosting regressor | 0.610621 | 0.572522 |

## 4) Visuospatial

This section is dedicated to the visuospatial composite results.

## 4.a) Performance box plots




### 4.a.1) Performance details

| | Model | RMSE_mean | RMSE_std |
|---|---|---|---|
| 0 | Linear regression | 0.537220 | 0.600697 |
| 1 | Lasso regression | 0.522154 | 0.581787 |
| 2 | Ridge regression | 0.536553 | 0.600157 |
| 3 | Huber regressor | 0.504064 | 0.618497 |
| 4 | Support vector Machine | 0.553401 | 0.660876 |
| 5 | Random forest | 0.583010 | 0.634256 |
| 6 | Gradient boosting regressor | 0.640403 | 0.743543 |

# 7.2.3 Recursive feature elimination



## Recursive feature elimination

This page is dedicated to the descriptions of the features selected for each given model.

Previous iteration

These features will be selected by recursive feature elimination. For more information see: Recursive Feature Elimination (RFE) for Feature Selection in Python (https://machinelearningmastery.com/rfe-feature-selection-in-python)

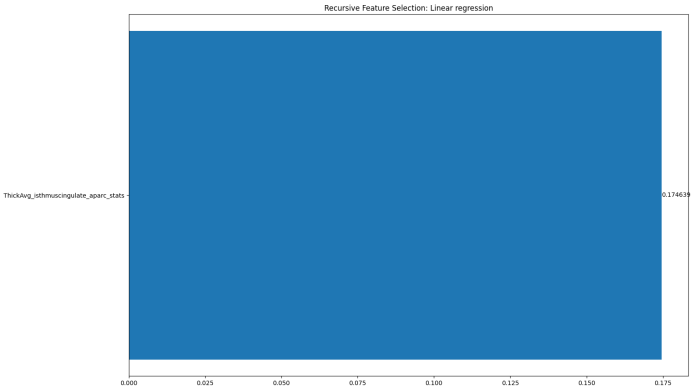
## Feature importance

Here below, you will find the features selected as well as their importance for each model. The feature importances have been assessed using "Permutation feature importance". For more information see: Permutation feature importance (https://scikit-learn.org/stable/modules/permutation_importance.html)

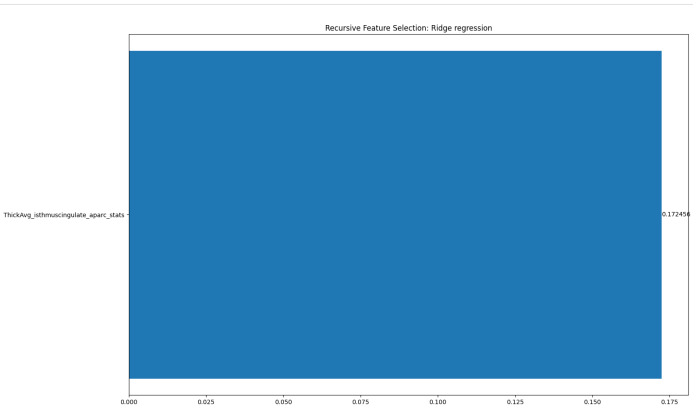
## 1) Executive

This section is dedicated to the executive composite results.

### 1.a) Linear Regression




### 1.b) Ridge Regression



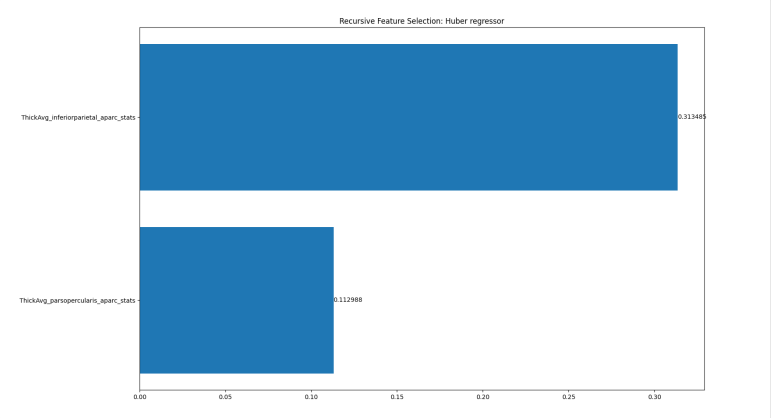

### 1.c) Huber Regression

## Machine Learning Report

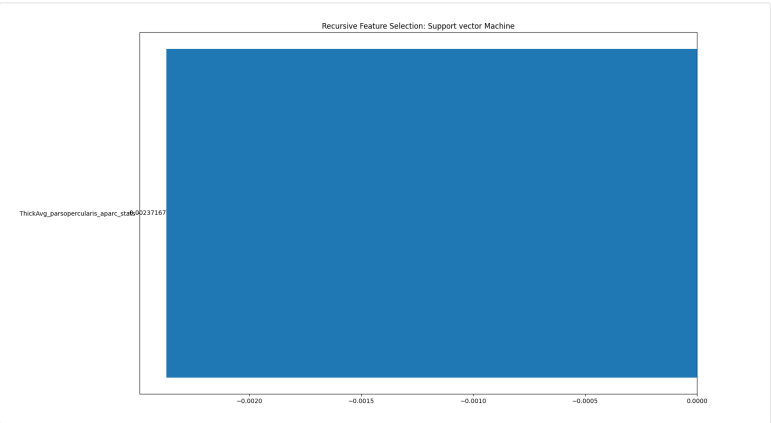


## Sections

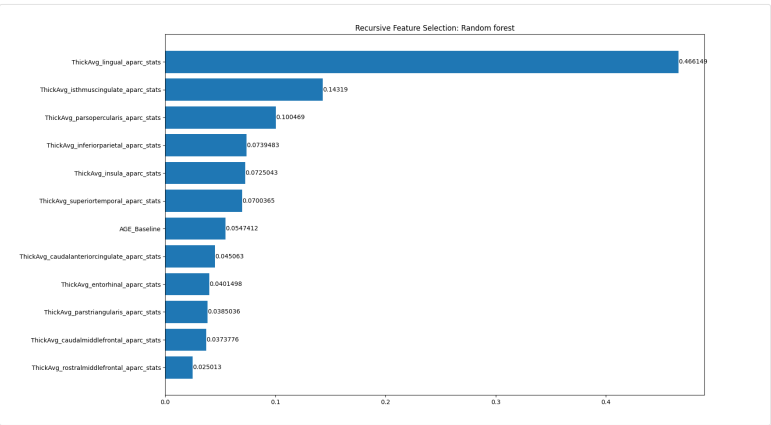





1.d) Support vector machine




1.e) Random forest



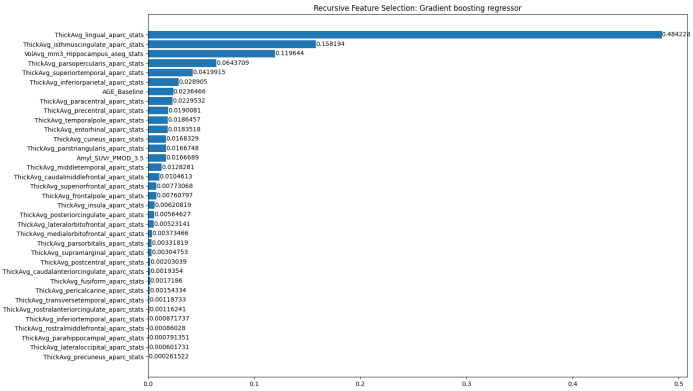

1.f) Gradient boosting regressor

## Machine Learning Report



## Sections






### 1.g) Summary of performance using recursive feature elimination

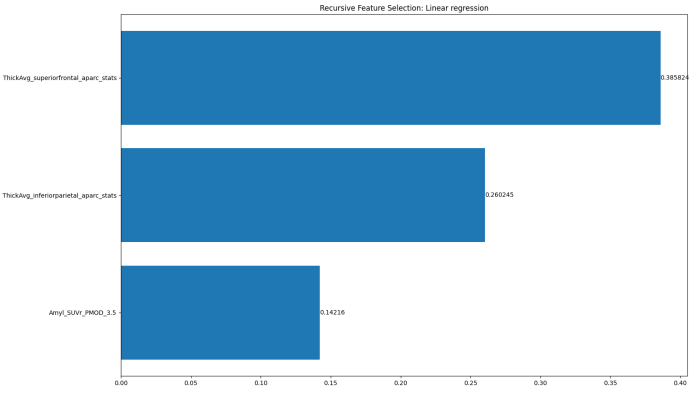
The performance of the models after recursive feature elimination have been assessed. These performances have been assessed using Leave-One-Out.

| | Model | RMSE_mean | RMSE_std | NFeatures |
|---|---|---|---|---|
| 0 | Linear regression | 0.330082 | 0.346266 | 1 |
| 1 | Ridge regression | 0.329736 | 0.346461 | 1 |
| 2 | Huber regressor | 0.317680 | 0.353529 | 2 |
| 3 | Support vector Machine | 0.343176 | 0.355143 | 1 |
| 4 | Random forest | 0.311869 | 0.307406 | 12 |
| 5 | Gradient boosting regressor | 0.324386 | 0.298156 | 35 |

## 2) Language

This section is dedicated to the language composite results.

### 1.a) Linear Regression



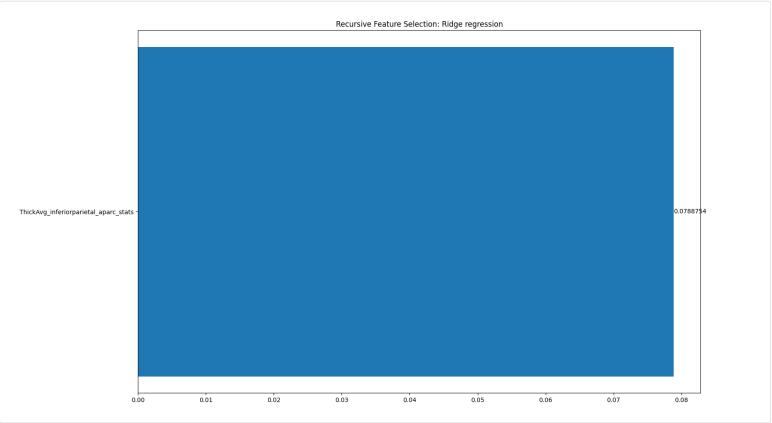

### 1.b) Ridge Regression

## Machine Learning Report

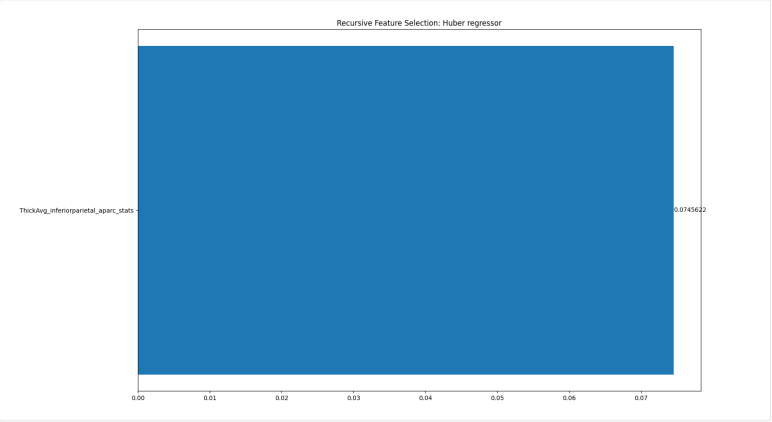


## Sections

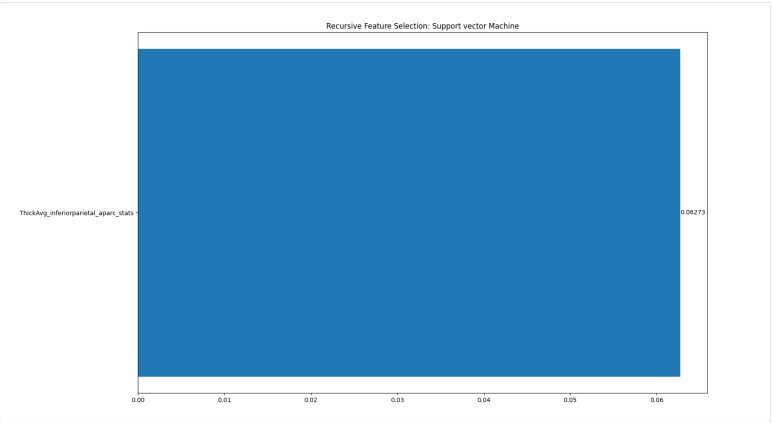





1.c) Huber Regression




1.d) Support vector machine



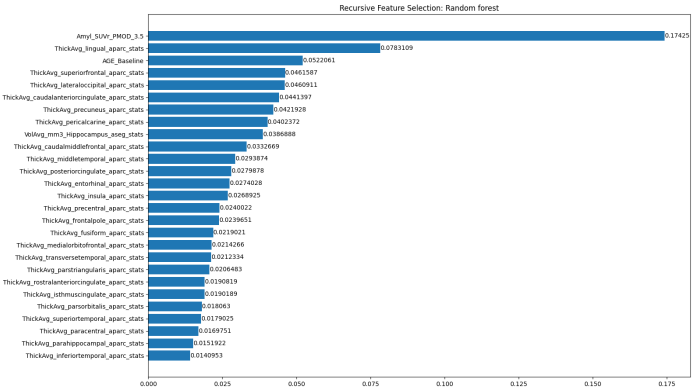

1.e) Random forest

## Machine Learning Report

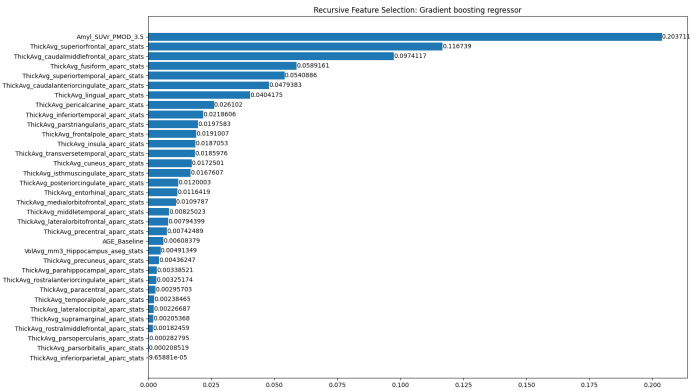


## Sections






### 1.f) Gradient boosting regressor




### 1.g) Summary of performance using recursive feature elimination

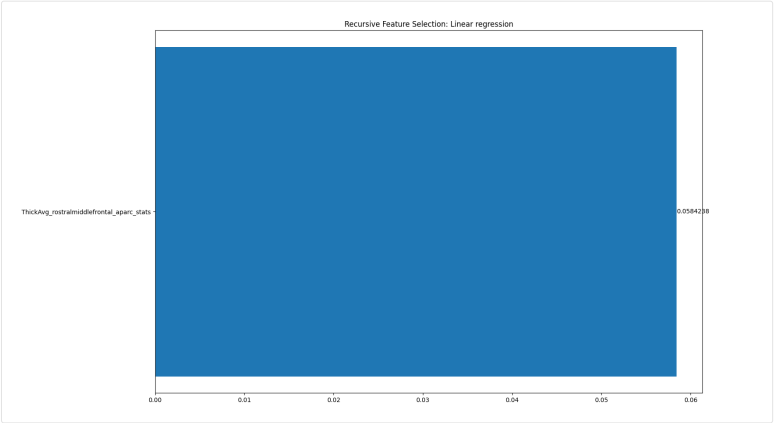
The performance of the models after recursive feature elimination have been assessed. These performances have been assessed using Leave-One-Out.

| | Model | RMSE_mean | RMSE_std | NFeatures |
|---|---|---|---|---|
| 0 | Linear regression | 0.261066 | 0.202711 | 3 |
| 1 | Ridge regression | 0.265828 | 0.228078 | 1 |
| 2 | Huber regressor | 0.265222 | 0.228368 | 1 |
| 3 | Support vector Machine | 0.265069 | 0.230312 | 1 |
| 4 | Random forest | 0.282797 | 0.220858 | 27 |
| 5 | Gradient boosting regressor | 0.297473 | 0.220279 | 34 |

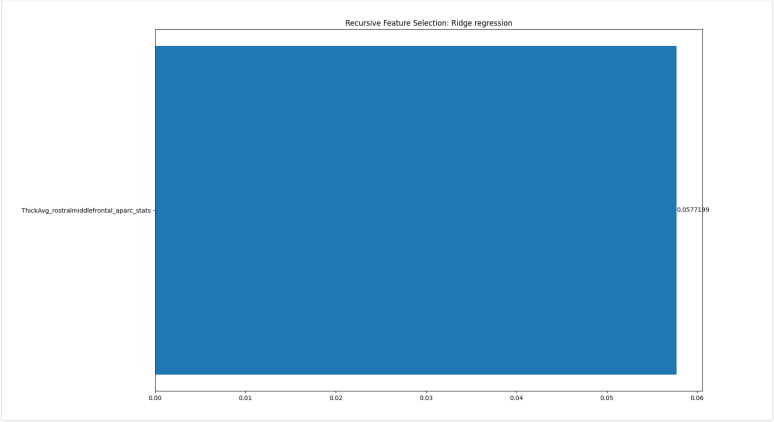
## 3) Memory

This section is dedicated to the moemory composite results.

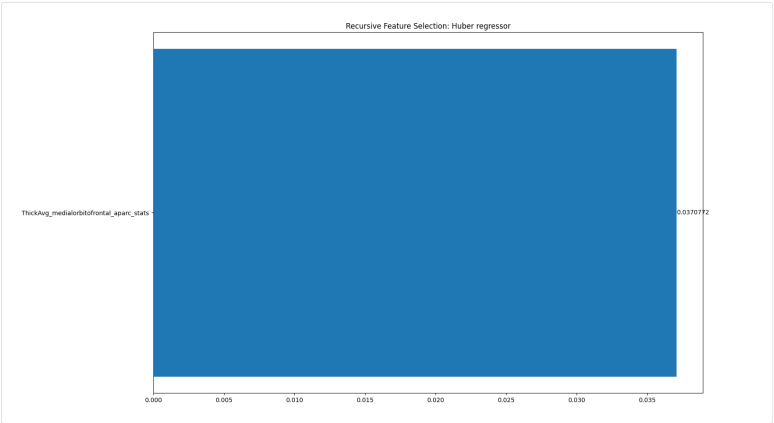
### 3.a) Linear Regression




3.b) Ridge Regression




3.c) Huber Regression



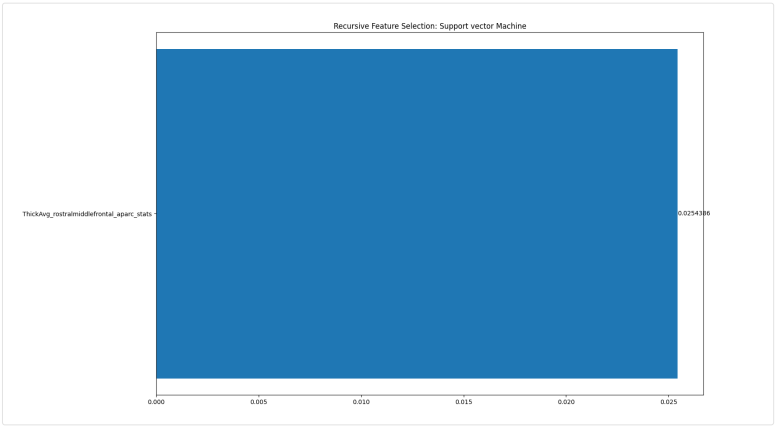

3.d) Support vector machine

## Machine Learning Report

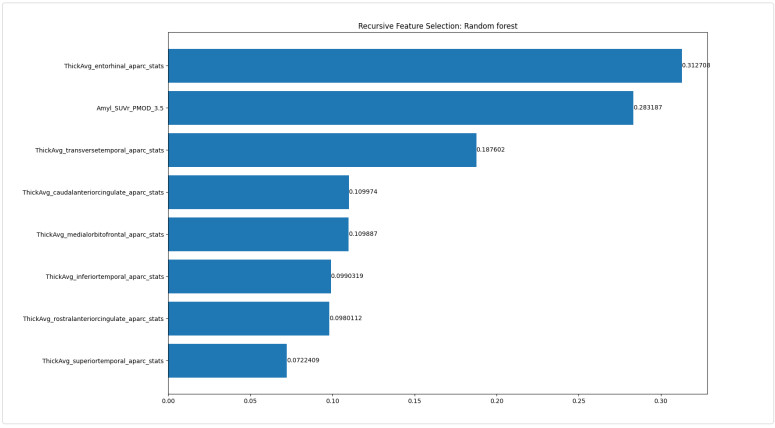


## Sections

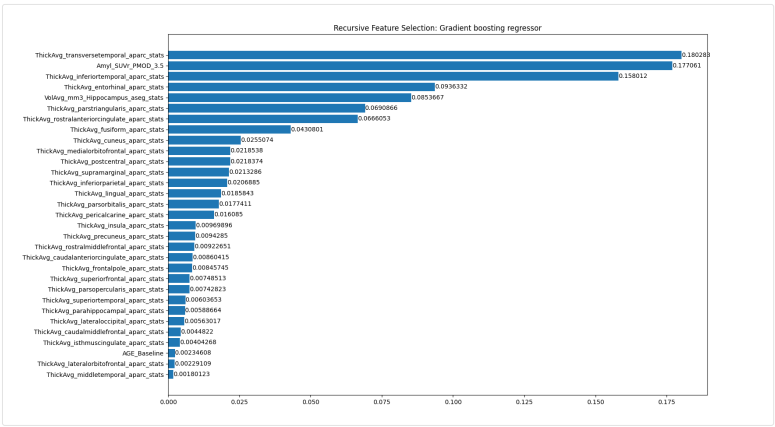





### 3.e) Random forest




### 3.f) Gradient boosting regressor




### 3.g) Summary of performance using recursive feature elimination

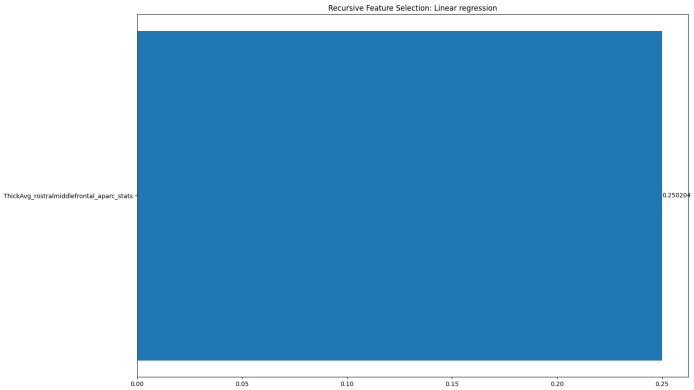
The performance of the models after recursive feature elimination have been assessed. These performances have been assessed using Leave-One-Out.

| | Model | RMSE_mean | RMSE_std | NFeatures |
|---|---|---|---|---|
| 0 | Linear regression | 0.507675 | 0.480303 | 1 |
| 1 | Ridge regression | 0.507505 | 0.480389 | 1 |
| 2 | Huber regressor | 0.486538 | 0.514168 | 1 |
| 3 | Support vector Machine | 0.507257 | 0.509561 | 1 |
| 4 | Random forest | 0.530281 | 0.521504 | 8 |
| 5 | Gradient boosting regressor | 0.625263 | 0.654157 | 31 |

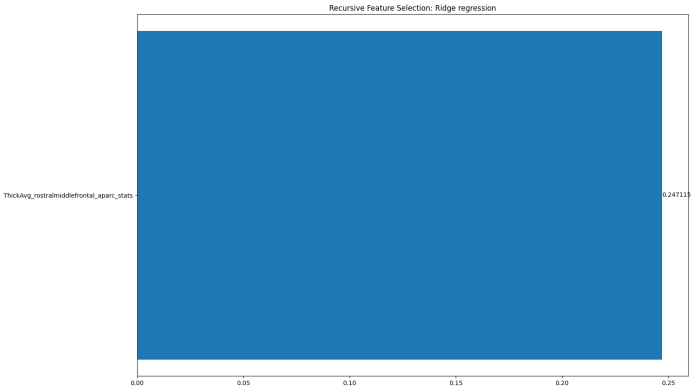
## 4) Visuospatial

This section is dedicated to the visuospatial composite results.

### 4.a) Linear Regression




### 4.b) Ridge Regression



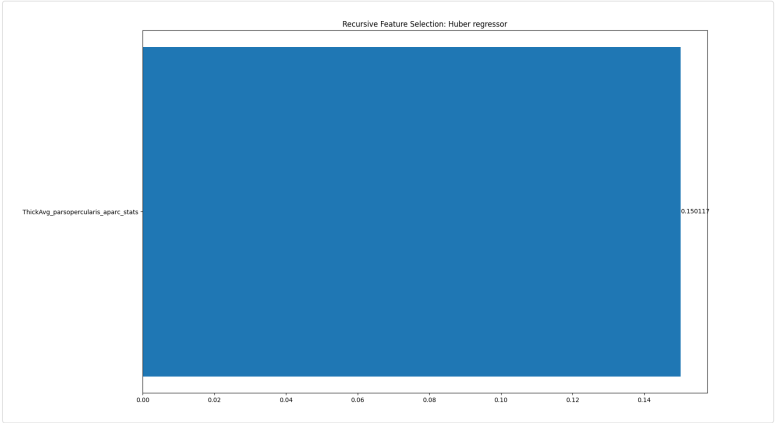

### 4.c) Huber Regression

## Machine Learning Report

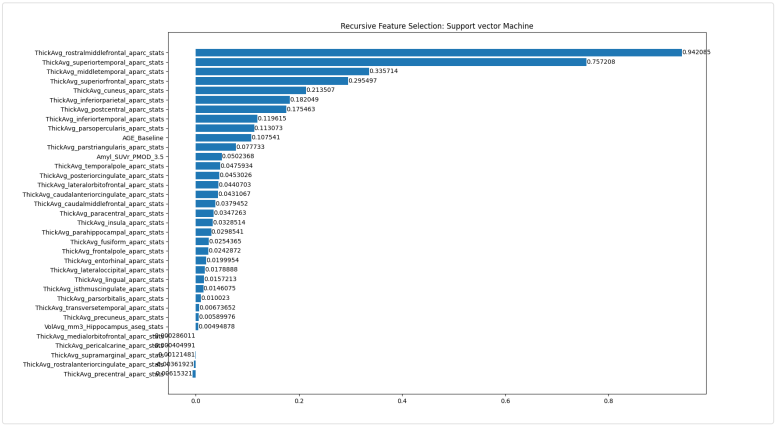


## Sections

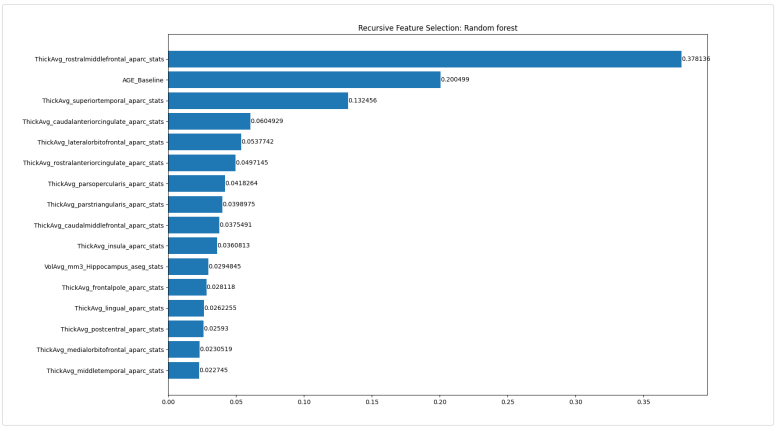





1.d) Support vector machine




4.e) Random forest



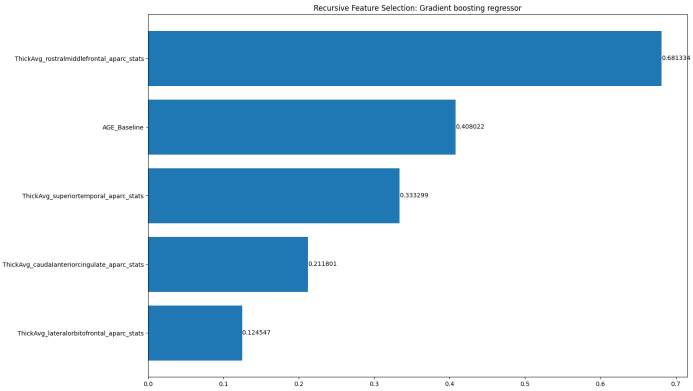

4.f) Gradient boosting regressor

## Machine Learning Report



## Sections






### 4.g) Summary of performance using recursive feature elimination

The performance of the models after recursive feature elimination have been assessed. These performances have been assessed using Leave-One-Out.

| | Model | RMSE_mean | RMSE_std | NFeatures |
|---|---|---|---|---|
| 0 | Linear regression | 0.513884 | 0.541642 | 1 |
| 1 | Ridge regression | 0.513256 | 0.541967 | 1 |
| 2 | Huber regressor | 0.484216 | 0.583091 | 1 |
| 3 | Support vector Machine | 0.529434 | 0.542569 | 35 |
| 4 | Random forest | 0.525876 | 0.547672 | 16 |
| 5 | Gradient boosting regressor | 0.517230 | 0.475005 | 5 |

### 7.2.4 Final performances



## Final models

This page is dedicated to the performances of both the Random forest regressor and the Gradient boosting regressor after hyperparamaters tuning.

Previous iteration

### 1) Executive

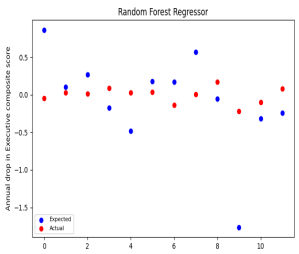
This section is dedicated to the executive composite results.

#### Random Forest Regressor

Root mean squared error:

0.5932134349309415

Test performance:




Download trained model

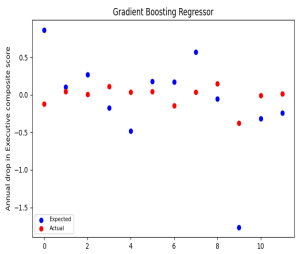
Download (./Pipeline/Executive/Models/Trained_Random_Forest.pkl)

#### Gradient Boosting Regressor

Root mean squared error:

0.5699886480806103

Test performance:




Download trained model

Download (./Pipeline/Executive/Models/Trained_Gradient_Boosting_Regressor.pkl)

### 2) Language

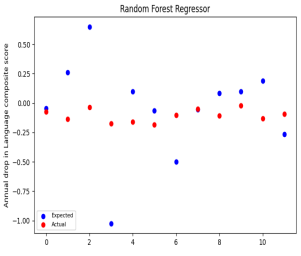
This section is dedicated to the language composite results.

#### Random Forest Regressor

Root mean squared error:

0.38449979263005063

Test performance:




Download trained model

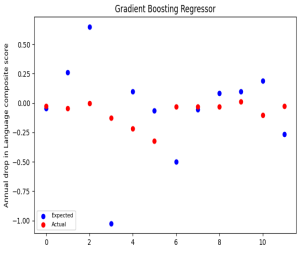
Download (./Pipeline/Language/Models/Trained_Random_Forest.pkl)

#### Gradient Boosting Regressor

Root mean squared error:

0.39517373343138856

Test performance:




Download trained model

Download (./Pipeline/Language/Models/Trained_Gradient_Boosting_Regressor.pkl)

### 3) Memory

This section is dedicated to the memory composite results.

#### Random Forest Regressor

Root mean squared error:

1.1229201081197728

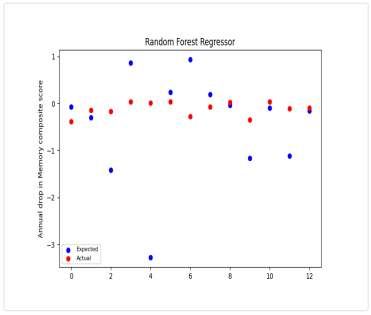
Test performance:

#### Gradient Boosting Regressor

Root mean squared error:

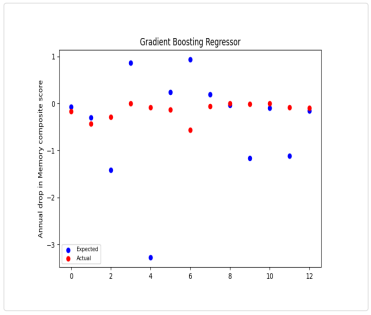
1.1463397110246156

Test performance:




Download trained model

Download (./Pipeline/Memory/Models/Trained_Random_Forest.pkl)




Download trained model

Download (./Pipeline/Memory/Models/Trained_Gradient_Boosting_Regressor.pkl)

# 4) Visuospatial

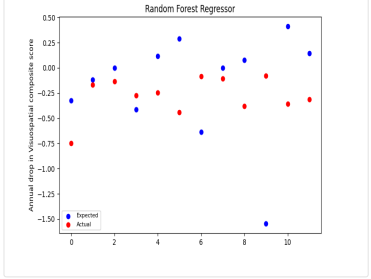
This section is dedicated to the visuospatial composite results.

## Random Forest Regressor

Root mean squared error:

**0.6026597231839657**

Test performance:




Download trained model

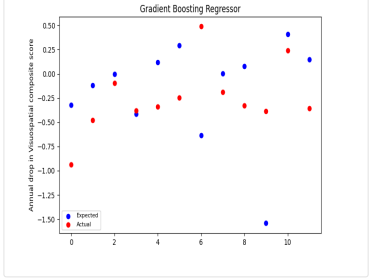
Download (./Pipeline/Visuospatial/Models/Trained_Random_Forest.pkl)

## Gradient Boosting Regressor

Root mean squared error:

**0.584621068029966**

Test performance:




Download trained model

Download (./Pipeline/Visuospatial/Models/Trained_Gradient_Boosting_Regressor.pkl)

**UNIVERSITÉ CATHOLIQUE DE LOUVAIN**
**École polytechnique de Louvain**
Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve, Belgique | **www.uclouvain.be/epl**