Data-driven approaches for Amyotrophic Lateral Sclerosis patient stratification using Clinical Profiles.

Eleonora Auletta

Thesis to obtain the Master of Science Degree in

Information Systems and Computer Engineering

Supervisors: Prof. Sara Alexandra Cordeiro Madeira
Prof. Cláudia Martins Antunes

Examination Committee

Chairperson: Prof. David Manuel Martins de Matos
Supervisor: Prof. Sara Alexandra Cordeiro Madeira
Member of the Committee: Prof. Barbara di Camillo

October 2021

Abstract

Amyotrophic Lateral Sclerosis is a neurodegenerative disease, characterized by progressive degeneration of upper and lower motor neurons in a few years from onset. In this context, any significant improvement of the patient's life expectancy and quality is of major relevance. Several studies have been made to address problems such as ALS diagnosis, and more recently, prognosis. In this context, the thesis targets prognostic prediction in ALS using machine learning models based on Patient Profiles, i.e. groups of patients with similar characteristics at diagnosis or similar disease progression patterns. Given the limited knowledge about the disease, this thesis aims to analyse and compare different stratification techniques. In this work, an analysis of the data by constructing patient similarity networks is initiated, by using different feature sets, distance measures and thresholds. Moreover, various clustering algorithms and ensemble techniques are analysed, to identify subgroups of patients to allow the design of more specific treatments to deal with the disease.

Keywords: Amyotrophic Lateral Sclerosis, Patient Profiles, Patient Stratification, Patient Network, Clustering, Ensemble Learning

Resumo

A Esclerose Lateral Amiotrófica é uma doença neurodegenerativa caracterizada por uma progressão geralmente rápida da degeneração muscular, geralmente levando à morte em poucos anos após o seu início. Neste contexto, qualquer melhoramento significativo da esperança e qualidade de vida do paciente é de grande relevância. Vários estudos têm sido feitos para abordar problemas como o diagnóstico da ELA, e mais recentemente, o prognóstico. Neste contexto, esta tese visa a previsão prognóstica na ELA, utilizando modelos de aprendizagem automática baseados em perfis de pacientes, ou seja, grupos de pacientes com características semelhantes no diagnóstico ou padrões semelhantes de progressão da doença. Neste trabalho, é iniciada uma análise dos dados através da construção de redes de semelhança de doentes, utilizando diferentes conjuntos de características, medidas de distância e limiares. Além disso, são utilizadas várias técnicas de *ensemble learning* e *clustering*, com o objectivo de identificar subgrupos de pacientes para permitir a concepção de tratamentos mais específicos para lidar com a doença.

Palavras-chave: Esclerose Lateral Amiotrófica, Perfis de Paciente, Estratificação de Paciente, Patient Network, Clustering, Ensemble Learning

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

Amyotrophic Lateral Sclerosis (ALS) is an aggressive neurodegenerative disease, characterized by progressive degeneration of upper and lower motor neurons, resulting in paralysis and death from respiratory failure. Research is motivated by the absence of a cure or effective treatment for ALS. Individuals living with the disease experience progressive paralysis, including the muscles involved in breathing and swallowing. In Italy and Europe, about 2 out of 100000 people are diagnosed with ALS in their lifetime, typically between the age of 50 and 70. Most ALS cases are sporadic, while only about 8–10% are inherited. The average survival time after ALS diagnosis is only three years. Still, about 20% of people with ALS live five years, 10% will survive ten years and 5% will live 20 years or more. Thanks to the 2014 ice bucket challenge, ALS recently gained new public awareness. Fueled by social media, the campaign prompted millions of people to post videos of themselves dumping cold water on their heads and drove hundreds of millions of dollars to the ALS Foundation. Even if drug developments typically take decades rather than years, it appears that these donations spurred new developments in ALS research. While it is widely accepted that neuron cell death is the reason for ALS symptoms, the underlying cause of ALS is still unclear. In fact, ALS may be caused by a network of cellular pathways and that their respective relevance changes with the course of the disease. Accordingly, several different therapeutic approaches are currently pursued, each addressing a different possible reason for neuron cell death. Currently, available therapy approaches only slow the progression of the disease. But with so many different approaches and clinical studies underway, there is hope that eventually a cure for ALS can be found. In this context, rigorous patient stratification would have an important role in addressing these shortcomings, contributing strongly to ALS research. The short survival time, allied with the lack of an available cure, means the prognosis for ALS patients is not usually the best for these patients. Prognostic studies in ALS have mostly been focused on finding discriminatory prognostic features for survival prediction [1]. Since respiratory failure is the most common cause of death [2], prognostic models able to anticipate the need for Non-Invasive ventilation (NIV) prescription, can have a positive impact on the quality of life and extend survival. ALS is considered a syndrome due to its high variability in presentation, progression and genetics [3]. In this work, a patient stratification approach is used, based on different sets of prognostic markers, called Clinical Profiles. Stratification approaches in ALS are usually patient-based, i.e. patients belong to a single group and all their observations are associated with that group. Subdividing patients into subgroups homogeneous with respect to biology, disease progression, and/or response to treatment enables precision medicine in ALS [4]. Precision medicine is an emerging approach for disease treatment and prevention, that takes into account individual variability in genes, environment, and lifestyle for each person, and aims to predict more accurately which treatment and prevention strategies are more suitable. In this context, time-sensitive analysis of heterogeneous genotype-phenotype data, clinical temporal and remote patient monitoring data, enable the development of automatic prognostic methods. Despite the enormous progress recently made in understanding ALS, the information about clinical and biomarker differences between patients, grouped according to the rate of disease progression, is incomplete. The aim is therefore to optimally manage patients with different progression rates, in order to provide the best-suited treatment for individual patients, according to a precision medicine approach. In an attempt to tackle precision medicine expectations, sophisticated machine learning (ML) algorithms are thus needed to effectively and efficiently extract and integrate knowledge from heterogeneous sources of genotype-phenotype data, clinical temporal data, and data collected using telemedicine. Machine learning became thus instrumental in inferring descriptive and predictive models from biomedical data [5–8]. Integrative learning approaches are also emerging, taking advantage of heterogeneous data and producing promising results [9]. Recent research further explores the potential of data heterogeneity, by learning models from repositories of health records, where a high multiplicity of clinical attributes is monitored over time [5, 10–12]. Nonetheless, most efforts in ALS research still focus on the analysis of a few biomarkers, such as gene expression [13] or clinical measures [14–16]. In this context, advanced ML approaches, able to learn from heterogeneous sources of data, unravel non-trivial models that can capture disease progression patterns over time and highlight subgroups of patients with similar characteristics [4, 17–19]. The new National project, entitled *Advanced learning models using Patient profiles and disease progression patterns for prognostic prediction in ALS* (AIpALS), aims to advance precision medicine and improve supportive care in ALS. This thesis is motivated by the AIpALS project and its results will be considered and possibly included. AIpALS benefits from knowledge gathered during FCT project NEUROCLINOMICS2 (2016-03/2020, PI Sara C. Madeira, ALS case study leader Mamede de Carvalho), whose goal was to unravel prognostic markers in neurodegenerative diseases, through clinical and omics data integration, using ALS and Alzheimer's disease as case studies. In this context, AIpALS defines three major data-driven goals, such as discover patient profiles, discover disease progression patterns, and propose prognostic models based on patient profiles and disease progression patterns.

1.2 Work Objectives

Evaluating multiple methods to find the best one, is an important part of model development, due to the absence of a model that fits all datasets. The ideal clinical risk model is accurate, is generalizable, provides a prediction in a reasonable time frame for clinical decision making and is interpretable by a clinician. The lack of knowledge about the disease demands the use of all available data, thus new analytic methodologies are required to deal with the scale and complexity of data. In this context, the thesis targets prognostic prediction in ALS by exploring, analysing and comparing different patient stratification techniques, based on Clinical Profiles. These analyses use a large dataset of genotype-phenotype data and clinical temporal data, already collected by national FCT project NEUROCLINOMICS2 (2016-2019, PI Sara C. Madeira) and European JPND project OnWebDuals (2016-2019, PI Mamede de Carvalho). The AIpALS project is motivated by several claims from the ALS community, actively investigating biomarkers, with the scope to contribute to earlier diagnosis and more precise monitoring of disease progression, concerning the need for advanced ML models to understand the heterogeneity of patients. Although complex genetic framework changes may partly explain the heterogeneity of the disease, it would be important to understand what are the reasons for the differences in the functional decline rate, using demographic, clinical phenotype, environmental profiles, clinical follow-up and remote patient monitoring. This would provide relevant information to design optimised disease strategies, including end-of-life decisions. Moreover, investigating differences in the expression of neurofilaments (NFs) and inflammatory biomarkers, in groups of patients stratified by disease progression rate, would allow the identification of optimal disease progression markers. In this respect, an objective method of evaluating disease progression is needed, to overcome the constant and so far frustrating effort to find an effective treatment. In this work, an approach using methodologies typical of Network Science is used. Several patient networks are constructed, which allow the subdivision of patients into groups, also in a visual sense, which are then analysed using different metrics. In detail, parameters such as features to be considered, similarity distances and thresholds, are set to identify homogeneous clusters. In addition, clustering algorithms and ensemble learning strategies are analysed, to identify homogeneous patient groups, thus targeting more effective prognostic prediction models.

1.3 Related Work

1.3.1 Data Mining in ALS

In the context of ALS, the related work is mostly associated with a population-based approach, focusing on common features significantly associated with reduced survival. In fact, we can divide the ALS studies into two problems. The first is related to the patients' diagnosis, investigating the heterogeneity in ALS subtypes [20], or the relevance of certain clinical features in the diagnosis, such as the paraspinal muscle EMG and motor-unit potentials (MUP) [21]. The second problem concerns the prognostic prediction, which can be divided into two different analyses. The most explored is the study of ALS survival, and the main associated features, including respiratory measures [22], but also the site of onset and the ALS Functional Rating Scale (ALSFRS) score [23]. The other type of studies, least explored, are related to the prediction of auxiliary respiration requirement, either with a tracheostomy, or Non-invasive ventilation (NIV) [1, 24, 25]. A comprehensive, systematic, and critical review of ML initiatives in ALS is discussed in this paper [1], together with their potential in research, clinical, and pharmacological applications. The focus of this review is to provide a clinical-mathematical perspective on recent advances and future directions of the field. It also discusses the pitfalls and drawbacks of specific models, highlighting the shortcomings of existing studies and providing methodological recommendations for future study designs. This paper primarily focuses on Machine Learning (ML) methods utilized in ALS research, such as Random Forests (RF), Support Vector Machines (SVM), Neural Networks (NN), Gaussian Mixture Models (GMM), Boosting methods and k-Nearest Neighbors (k-NN). Despite the considerable sample size limitations, ML techniques are successfully applied to ALS datasets and a number of promising diagnosis models are proposed. Prognostic models are tested using core clinical variables, biological, and neuroimaging data. These models also offer patient stratification opportunities for future clinical trials. Despite the enormous potential of ML in ALS research, statistical assumptions are often violated, the choice of specific statistical models is seldom justified, and the constraints of ML models are rarely enunciated. From this article, several studies have derived crucial information, which has allowed further investigation of the disease, expanding ALS research [26–28]. In the network context, a convolutional neural network (CNN) is constructed in this work [29], coupled with a fully connected top layer for survival estimation. An objective function is designed to directly estimate the probability of survival at discrete time intervals, conditional to the patient not having incurred any adverse event at previous time points. The CNN and objective function are tested on a large dataset of longitudinal data of patients with Amyotrophic Lateral Sclerosis (ALS) and compared against other neural networks designed for survival analysis, and against the optimization of Cox-partial-likelihood or a simple logistic classifier. The use of the objective function outperforms both Cox-partial-likelihood and logistic classifier, independently of the network architecture, and the deep CNN provides the best results in terms of AU-ROC, accuracy and mean absolute error. In this work [30, 31], a Dynamic Bayesian Network (DBN) model of ALS progression is presented to detect probabilistic relationships among variables included in the *Pooled Resource Open-Access ALS Clinical Trials Database*. A Bayesian Network is a mathematical model that represents the joint probability distribution of a set of random variables as a directed acyclic graph (DAG). A DBN extends a Bayesian Network to model dynamic processes, describing the dependencies among the variables along time [31]. The model unravels new dependencies among clinical variables in relation to ALS progressions, such as the influence of basophil count or bicarbonate on movement, communication and respiratory functional state. Furthermore, it provides an indication of ALS prognosis, in terms of the most probable disease trajectories across time at the level of both patient population and the individual patient. The risk factors identified by this DBN model could allow patients' stratification based on the velocity of disease progression and a sensitivity analysis on this latter in response to changes in input variables, i.e. variables measured at diagnosis. In this study [32], the prognostic models for ALS are mapped to assess their potential contribution and suggest future improvements on modeling strategy. A total of 28 studies describing the development of 34 models and the external validation of 19 models were included. Among the models predicting ALS progression or survival, the most frequently used predictors were age, ALS Functional Rating Scale/ALS Functional Rating Scale-Revised, site of onset, and disease duration. The modeling method adopted most are machine learning, and only one model is assessed with an overall low risk of bias, suggesting a relatively reliable model for practice. The usefulness of the prognostic models reviewed is questionable, due to several methodological pitfalls and the lack of external validation done by fully independent researchers. Different machine learning methods such as Dynamic Bayesian Network (DBN), Random Forest (RF), etc., are adopted for modelling. The outcomes concerned are ALS progression (n = 12; 35%), change in weight (n = 1; 3%), respiratory insufficiency (n = 2; 6%), and survival (n = 19; 56%). The most important finding of this systematic review was that most of the models have dual problems of high risk of bias and low-quality reporting, based on the prediction model risk of bias assessment tool (PROBAST). The high risk of bias was mainly due to the poor reporting of the number of participants with the outcome, selection of predictors merely by univariable analyses, and inappropriate evaluation of model performance. Only one model was assessed with an overall low risk of bias and it performed well in both discrimination and calibration, suggesting a relatively reliable model for practice. Maybe it was due to the fact that the modelling methodology has still been under improvement these years [33] and PROBAST was just proposed in 2019. Besides the problem of modelling methodology, the low-quality reporting was also a problem that cannot be ignored, which was mainly due to the poor model presentation, thus the full model equation should be provided to enable independent external validation, update, and recalibration [34]. In future research, more models based on low- or middle-income countries should be established and more attention should be paid to the addition of novel promising predictors, external validation, and head-to-head comparisons of existing models. A prognostic model is presented [35] for functional decline in ALS where outcome uncertainty is taken into account. Patient data are reduced and projected onto a 2D space using Uniform Manifold Approximation and Projection (UMAP), a novel non-linear dimension reduction technique. Information from 3756 patients is included. Supervised learning models usually require large amounts of data to avoid overfitting and lack of generalisation, which are not available in ALS research. Unsupervised learning methods have the advantage of capturing distribution patterns without data implications. Standard linear methods such as principal component analysis (PCA) [36] have been used in ALS for gene expression analysis [37]. Unfortunately, these conventional linear-based methods are not capable of describing non-linear relationships and have underperformed in this study context. Non-linear methods provide new modelling possibilities given their comprehensive ability to describe data correlations and have successfully been tried out for ALS phenotype identification on clinical trial data [38] with t Student Stochastic Neighbour Embedding (t-SNE), the current state-of-the-art manifold learning model [39]. UMAP is a neighbourhood based approach, that preserves data neighbourhood, distances and density and works in two steps. UMAP projection of patients shows an informative 2D data distribution. As limited data availability precluded complex model designs, the projection is divided into three zones defined by a functional impairment range probability. Zone membership allowed individual patient prediction. Patients belonging to the first zone has a probability of 83% (± 3%) to have an ALSFRS score over 20 at a 1-year follow-up. Patients within the second zone had a probability of 89% (± 4%) to have an ALSFRS score between 10 and 30 at a 1-year follow-up. Finally, patients within the third zone had a probability of 88% (± 7%) to have an ALSFRS score lower than 20 at a 1-year follow-up. This approach requires a limited set of features, is easily updated, improves with additional patient data, and accounts for results uncertainty. This method could therefore be used in a clinical setting for patient stratification and outcome projection. Various ALS staging methods are proposed in [40], used as a tool for rehabilitation, rapid functional assessment, comparison of different treatment models, biomarker analysis and health economics. The most widely studied approaches are the Milano-Torino (MiToS) functional staging and King's clinical staging systems [41, 42]. The MiToS system uses six stages, from 0 to 5 and is based on functional ability as assessed by the ALS Functional Rating Scale-Revised (ALSFRS-R) [43], with stage 0 being normal function and stage 5 being death. The King's system uses five stages, from 1 to 5 and is based on disease burden as measured by clinical involvement and significant feeding or respiratory failure, with stage 1 being symptom onset and stage 5 being death. King's staging is mostly focused on anatomical disease spread and significant involvement of respiratory muscles, whereas MiToS staging is aimed more towards the distinction of functional capabilities during the spread of the disease. Therefore, while the King's clinical staging system is able to differentiate early to mid-disease well, the MiToS staging is able to differentiate late stages in detail. To compare each staging system, King's and MiToS scores are plotted against frequency for all pairwise comparisons. These differences in disease description by the two systems are also proved by a Spearman's rank correlation of 0.54, showing some correspondences between the two systems. Moreover, association testing shows that King's stage 4 and MiToS stage 2 are the most strongly associated between all staging pairs. Linearly weighted kappa coefficient tests the strength of agreement between two ordinal scales, with an increase of penalty based on the level of disagreement. A commonly used scale to interpret kappa values, ranges from 0, which indicates a chance agreement, to 1, i.e. a perfect agreement, with intervals of poor, slight, fair, moderate, substantial and almost perfect. The analysis between King's and MiToS staging systems showed a fair agreement with a linearly weighted kappa coefficient of 0.21. These results support the use of both systems when staging, as they summarise two different aspects of patient information. Clinical stages in amyotrophic lateral sclerosis can be measured using a simple system based on the number of CNS regions involved and the requirement for gastrostomy or noninvasive ventilation. In this study [44], a standard operating procedure (SOP) is designed to define the standardized use and application of the King's staging system. Case vignettes representative of ALS patients at different disease stages are defined. Health care professionals are first trained on how to use the SOP and then asked to stage the vignettes using the SOP. The extent to which SOP staging corresponded with the correct clinical stage is measured. The reliability of staging using the SOP is excellent, with a Spearman's rank coefficient of 0.95, and is high for different groups of health care professionals, and for those with different levels of experience in ALS. The limits of agreement between SOP staging and actual clinical-stage lie within a single stage, confirming that there is a clinically acceptable level of agreement between staging using the SOP and actual King's clinical stage. There are also no systematic biases of the SOP over the range of stages, either for over-staging or under-staging. Thus, the staging SOP provides a reliable method of calculating clinical stages in ALS patients and can be used prospectively by a range of health care professionals with different levels of experience. A patient-driven model for ALS prognosis prediction of respiratory failure is proposed in [25]. However, the strategies adopted for the most part of these studies rely on statistical tests, Kaplan-Meier survival tables, and multivariable Cox proportional hazard regression models, which are typical of population-based studies. Other studies [24, 25] use strategies related to cluster temporally-related tests, yielding patient snapshots, for prognostic prediction using patient snapshots and time windows. Those models are applied to predict disease progression, i.e. if a patient that can breathe without help will be in need of NIV after

90, 180 or 365 days. In the construction of the prognostic models, the impact of preprocessing techniques is assessed, such as missing value imputation, knowledge-based discretization and feature selection, using stratified 5 10-fold CV in the training set (70% of all instances, or snapshots). Feature selection processes are also performed to help clinicians understand what are the best tests and medical exams to predict the need for NIV. The main conclusion was that, even though the results did not improve significantly, the prognostic models obtained were simpler, and thus presented an important advantage, since clinicians can thus prescribe clinical tests according to their weight in the models, as well as their costs. The models achieved AUC values of 78.87%, 79.11% and 78.86% for 90, 180 and 365 days, respectively, for Naive Bayes, followed by Linear Regression and Random Forest. However, it is known that traditional machine-learning approaches excel at performance, but often have limited interpretability, thus involving research into new methods and approaches for dealing with ALS patient data.

1.3.2 Patient Similarity and Patient Stratification

Patient similarity assessment is an important task in the context of patient cohort identification for comparative effectiveness studies and clinical decision support applications. The goal is to derive clinically meaningful distance metrics to measure the similarity between patients represented by their key clinical indicators [45]. Nevertheless, patient similarity, or distance, poses several different challenges, where the subjective notion of similarity rises as one of the most critical. In fact, each physician may have a different perspective about how similar two patients are, as they assign different weights to different features. Some studies have proposed a way of learning these weights automatically [46], and others suggested that the expert knowledge can be integrated, and used to learn a new unified similarity measure [45]. In this paper [47], inspired by the analogy studies in psychology, a novel framework to predict diagnoses is proposed, that computes discharge diagnosis similarity of patient pair, and take this value as the outcome of a supervised prediction model. The input of the supervised prediction model is the feature vectors of each patient pair in the cohort, which is a process that deriving the attribute similarity. The supervised model is to assign weight to attributes automatically and the prediction of the target is a process that deriving relational similarity. Extensive experimental results on real-world networks demonstrate that the patient similarity-based model achieves better performances in the diagnostic prediction task. Several noteworthy contributions are made, such as the idea of a general patient-similarity-based framework for diagnostic prediction, which is inspired by the structure-mapping theory about analogy reasoning in psychology. The measurement of patient similarity by using diagnoses helps machine learning models to learn in ways that are not possible with existing binary targets alone. Rather than attempting to solve the problem of predicting diagnoses just by k-nearest neighbours, a two-step method is instead explored, that retrieves positive analogies to generate hypotheses and negative analogies to reject hypotheses. The results demonstrate that the proposed model advances the performance of diagnostic prediction tasks. Moreover, this method allows to identify analogous patients and predict diagnoses with better performance than the baselines. In fact, the f-1 scores of positive-analogy-based prediction and positive-negative-analogy-based prediction are 0.698, 0.703 respectively, while the f-1 scores of the baselines range from 0.368 to 0.661. In conclusion, the authors show that a patient-similarity-based model provides diagnostic decision support that is more accurate, generalizable, and interpretable than those of previous methods, based on heterogeneous and incomplete data. The model also serves as a new application for the use of clinical big data through artificial intelligence technology. The results obtained in this study [48], can help accelerate disease understanding in several ways. In fact, the stratification scheme suggested in this analysis offers novel insights that can be integrated into the development of novel ALS therapeutics, aiding patient selection and interpretation of results. The method analyzes patient clusters, showing a clear pattern of consistent and clinically relevant subgroups of patients that also enabled the reliable classification of new patients. Other studies [49], propose a patient stratification approach using Clinical and Patient Profiles to tackle the heterogeneity problem associated with ALS. Clustering techniques are used to group patients' observations according to predefined Clinical Profiles, i.e. Prognostic, Respiratory and Functional [25], resulting in homogeneous groups, which are then used to learn specialized prognostic models. The proposed approach resulted in three sets of ALS Patient Profiles, that were then used to create specialized prognostic models capable of predicting early administration of NIV. Some of these models outperformed the baseline model, highlighting the importance of patient stratification to improve prognostic prediction in this heterogeneous disease. In the network context, studies rely on supervised learning approaches, and it is not clear how temporal data can be addressed. Other works propose a novel unsupervised learning strategy using a distance measure capable of dealing with multivariate time series in order to build a network of patients, which can then be analyzed from a modular point of view [50]. The found modules, or patient communities, can be studied according to their particular characteristics, possibly reflecting ALS subtypes, which might help to better understand the disease. Moreover, such modules can then be used in a supervised learning fashion to train expert models for discriminating subgroups of patients. In this study [48], a novel bottom-up method is designed for the identification of consensus patient clusters and the determination of discriminating features, a challenging task since no known ground truth exists for ALS patient stratification. No a priori assumptions are made regarding patient sub-populations, but instead, patient clusters are defined by a consensus vote based on participants' submitted algorithms. The results of this study aim to accelerate disease understanding in several ways. The stratification scheme suggested offers novel insights that can be integrated into the development of novel ALS therapeutics, aiding patient selection and interpretation of results. Novel differentiating features, such as creatinine or SVC, can also help shed light on mechanisms related to disease progression, as well as mechanisms related specifically to end of life in ALS, a topic of critical clinical importance. To characterize clusters and the involved patients for clinical relevance, all pairs of clusters are compared, using ANOVA and t-test, resulting in multiple-testing corrected false discovery rates or FDRs, to assess which features have values specifically different between the clusters. The correlation between feature values and clinical outcomes are examined in each cluster, to identify the features that are important for prediction in some clusters but not in others. Overall this cluster helps integrate information, some already accepted, such as the association of bulbar onset and respiratory signs with a poorer prognosis, and some suggested, the potential predictive roles of creatinine, urine creatinine, neutrophil and others, in a statistically supported unified framework, enabling discerning fast and slow progressing patients earlier in their disease course, as well as markers helping to identify patient reaching the final stages of their disease. Consensus clusters can be broadly regarded as classifying patients as slow progressing, fast progressing, early-stage or late stage. In order to demonstrate how the identified clusters can be utilized in a clinical setup, whether new patients can be assigned into their respective clusters reliably is also examined. Increasing prognostic models for Amyotrophic Lateral Sclerosis have been developed. This study [51] is aimed to uncover new connections within the ALS network through a bioinformatic analysis, by which C13orf18 is identified, recently named Pacer, as a new component of the autophagic machinery and potentially involved in ALS pathogenesis. Expression of Pacer was then investigated in vivo using spinal cord tissue from two ALS mouse models (SOD1G93A and TDP43A315T) and sporadic ALS patients. Copy number variation (CNV) data are collected from 4 published studies [52–55]. A total of 338 genes associated with ALS are included in the analysis. Additionally, genes linked to ALS are collected from The Huge Navigator, an integrated knowledge base of human genome epidemiology. The selected CNV and HuGE genes are uploaded into the Ingenuity Pathway Analysis (IPA) system (Qiagen), which contains protein/protein interaction (PPI) and expression datasets. A “core analysis” approach is used and 241 genes are obtained in 12 ALS-associated subnetworks. This approach allows to build interaction networks with all ALS associated genes, thus uncovering new connections to previously unrecognized genes/proteins as being part of the ALS disease network. Neurodegenerative diseases are multifactorial, involving a combination of genetic and environmental factors. Genetic studies in ALS have made significant advances in the understanding of disease pathogenesis by using whole genome or whole exome sequencing strategies. However, the primary cause of approximately half of familial ALS cases and the majority of sporadic ALS cases remains unexplained. The use of systems biology approaches, to study neurodevelopmental and neurodegenerative diseases, has recently proven to aid our understanding of underlying disease mechanisms by unravelling new genes, pathways or subnetworks responsible for an illness that would not have been recognized using traditional approaches. Pacer expression is up-regulated on the transcriptional and translational level upon autophagy induction, resembling the behaviour of other autophagy genes.

Chapter 2

Data and Exploratory Data Analysis

Python Code: https://colab.research.google.com/drive/14VN-BUow_GF9Pe7zhPspLa04NK4MNVFt?usp=sharing

2.1 Data

Data from a cohort of 1590 Portuguese ALS patients are used, followed between 1992 and 2021. For each patient, demographic and genetic features are gathered, as well as results from multiple clinical exams and tests. An accurate analysis of the features is reported in the study conducted by André V. Carreiro [24], and Table 2.2 contains a description of the main features available in the dataset. Three sets of clinically relevant features are used, called Clinical Profiles, composed of a set of features that are focused on different aspects of the disease [49]. Features can be classified as static or temporal. Static features do not change over time, such as demographic and genetic features. Temporal features are clinical tests that are usually measured at each appointment, approximately every three months. The functional scores (ALSFRS) in Table 2.2 are an aggregation of integer values on a scale $0-4$, where 0 is the worst and 4 is the best, as observations of patient characteristics at a given time point. Table 2.1 compiles how each of the functional scores can be calculated.

Functional Score	**Result**
ALSFRS	sum of Q1 to Q10
ALSFRS-R	sum of Q1 to Q9 + QR1 + QR2 + QR3
ALSFRSb	Q1 + Q2 + Q3
ALSFRSsUL	Q4 + Q5 + Q6
ALSFRSsLL	Q7 + Q8 + Q9
ALSFRSr	Q10
R	QR1 + QR2 + QR3

Table 2.1: Functional Scores and Sub-scores according ALS Functional Rating Scale (ALSFRS).

The observations are the following:

- Q1 - Speech
- Q2 - Salivation
- Q3 - Swallowing
- Q4 - Handwriting
- Q5 - Cutting food and Handling Utensils
- Q6 - Dressing and Hygiene
- Q7 - Turning bed and adjusting bed clothes
- Q8 - Walking
- Q9 - Climbing Stairs
- Q10 - Respiration
- QR1 - Dyspnea
- QR2 - Orthopnea
- QR3 - Respiratory Insufficiency

Two datasets are used for the analysis. The first dataset is a flat table with size 7291x46, where the items represent the ALS patients and the columns the attributes. In particular, for each patient, there can be several items, due to the multiple tests and hospital check-ups over the years. In order to have a coherent dataset, only the first row available for each patient is considered, also referred to as start time, i.e. the first time the patient undergoes a check-up. The second dataset is a flat table with size 1181x8, where patient IDs with the respective cluster are collected. The datasets are independent, so they contain different patients. For the purpose of this analysis, only patients present in both datasets, i.e. patients from the first dataset who have a corresponding group in the second dataset, are considered. The two datasets are merged, obtaining a unique dataset of size 1104x47. Only numerical features are considered, in order to use consistent similarity distances, as described in Section 3.1.3. When data are well prepared, by using methods such as correcting, recording, scaling and missing value imputation (Section 2.4), the next step is to perform statistical description and inference. Table 2.2 displays descriptive statistics of the dataset, that give an estimation of the differences in baseline characteristics, providing evidence for further multivariable analysis. Varieties of methods are available for univariate description and bivariate inference. Mean and standard deviation are used to describe normally distributed data, while median and interquartile ranges are employed for skewed data. For nominal data, the mode is used to describe the central tendency.

Name	Temporal/Static	Type	SubGroup	Mean/Mode
Gender	Static	Categorical	Demographics	Male
BMI	Static	Numeric	Demographics	24.80
MND	Static	Categorical	Medical/Family History	No
UMN vs LMN	Static	Categorical	Onset Evaluation	LMN
Age at Onset	Static	Numeric	Onset Evaluation	62.41
Onset Form	Static	Categorical	Onset Evaluation	Spinal
Disease Duration	Static	Numeric	Onset Evaluation	18.33
EERC	Static	Categorical	Onset Evaluation	Probable
Expression C9orf72	Static	Categorical	Genetic	Unknown
ALS-FRS	Temporal	Numeric	Functional Scores	31.15
ALS-FRS-R	Temporal	Numeric	Functional Scores	38.81
ALS-FRSb	Temporal	Numeric	Functional Scores	10.13
ALS-FRSsUL	Temporal	Numeric	Functional Scores	8.90
ALS-FRSsLL	Temporal	Numeric	Functional Scores	8.51
ALS-FRSr	Temporal	Numeric	Functional Scores	3.61
R	Temporal	Numeric	Functional Scores	11.24
Vital Capacity (VC)	Temporal	Numeric	Respiratory Tests	83.91
Forced VC (FVC)	Temporal	Numeric	Respiratory Tests	84.58
P0.1	Temporal	Numeric	Respiratory Tests	98.47
SNIP	Temporal	Numeric	Respiratory Tests	57.04
MIP	Temporal	Numeric	Respiratory Tests	53.14
MEP	Temporal	Numeric	Respiratory Tests	64.07
Date of NIV	Temporal	Date	Respiratory Status	-
PhrenMeanAmpl	Temporal	Numeric	Neurophysiological Tests	0.51
PhrenMeanLat	Temporal	Numeric	Neurophysiological Tests	8.47
Cervical Extension	Temporal	Numeric	Other Physical Values	4.77
Cervical Flexion	Temporal	Numeric	Other Physical Values	4.36
Prog_group	Temporal	Categorical	Progression Group	Normal

Table 2.2: Main features in the Portuguese ALS dataset.

2.2 Data Preprocessing

Any machine learning algorithm that computes the distance between the data points needs Feature Scaling, i.e. standardization and/or normalization. Therefore, data preprocessing techniques are applied to the dataset in order to make it more clean, consistent and noise-free, thus improving the efficiency of clustering algorithms. In fact, variables that are measured at different scales do not contribute equally to the analysis and might end up creating a bias. There are various data normalization methods like Min-Max, Z-Score and Decimal Scaling, the best normalization method depends on the data to be normalized. Here, the normalize function is used to normalize the data, which takes an array as an input and normalizes its values between 0 and 1, returning an array with the same dimension as output. The new point is computed as:

$$\mathbf{x}_{new} = \frac{\mathbf{x} - \mathbf{x}_{min}}{\mathbf{x}_{max} - \mathbf{x}_{min}}$$

Standardization, or Z-Score normalization, is another scaling technique where the values are centred around the mean with a unit standard deviation. Therefore, the mean of the attribute becomes zero and the resultant distribution has a unit standard deviation. Standardizing the features is important when measurements that have different units are compared. The Z-score is computed as:

$$\mathbf{x}_{new} = \frac{\mathbf{x} - mean(\mathbf{x})}{Std(\mathbf{x})}$$

2.3 Descriptive and Inferential statistics

Because the choice of statistical methods depends on the distribution of data, the skewness of data needs to be examined. The distribution can be visualized using histogram, as shown in Figure 2.1. A histogram divides the values of a numerical variable into bins and counts the number of observations that fall into each bin. By visualizing these binned counts in a columnar fashion, a very immediate and intuitive sense of the distribution of values within a variable can be obtained. The distribution of the features BMI, FVC and Age on onset appears to be symmetrical, while the other variables are skewed. However, graphic visualization only gives a hint on the distribution of data. Two classes of techniques are used

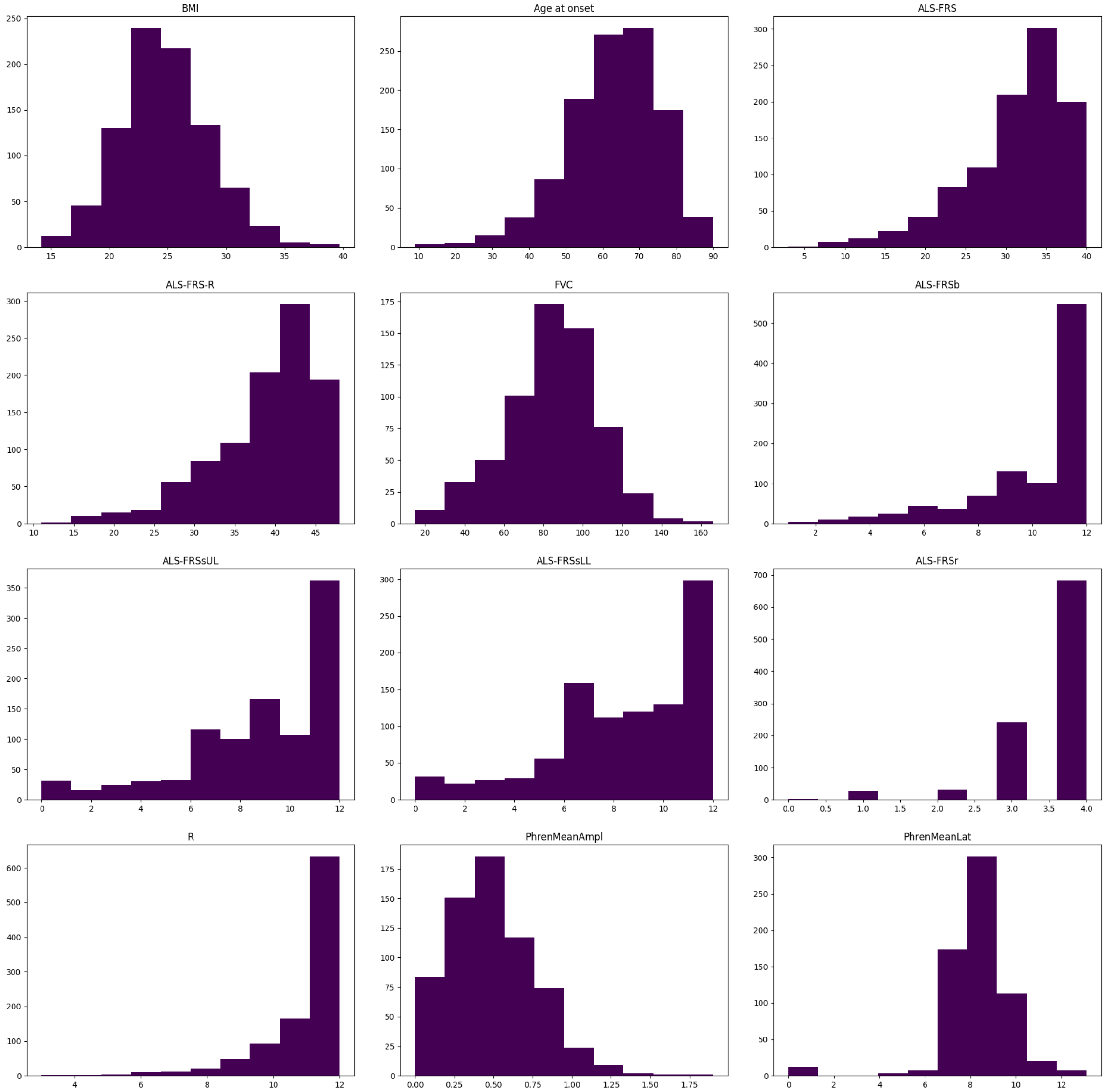

Figure 2.1: Main numerical fearures histogram.

for checking whether a sample of data is Gaussian, which are Graphical Methods, for plotting data and qualitatively evaluating whether they look Gaussian, and Statistical Tests, that calculate statistics on the data and quantify the probability that they are drawn from a Gaussian distribution. Figure 2.2 shows the quantile-quantile plots, or QQ plots for short, of features with symmetrical histogram, that could therefore have a Gaussian distribution. Despite these checks are qualitative, so less accurate than the statistical methods, if the distribution visually markedly differs from that expected, it is possible to draw reliable conclusions. The resulting points are plotted as a scatter plot, with the idealized value on the x-axis and the data sample on the y-axis. A perfect match for the distribution is shown by a line, dots on a 45-degree angle from the bottom left of the plot to the top right, thus deviations from the line correspond to deviations from the expected distribution.

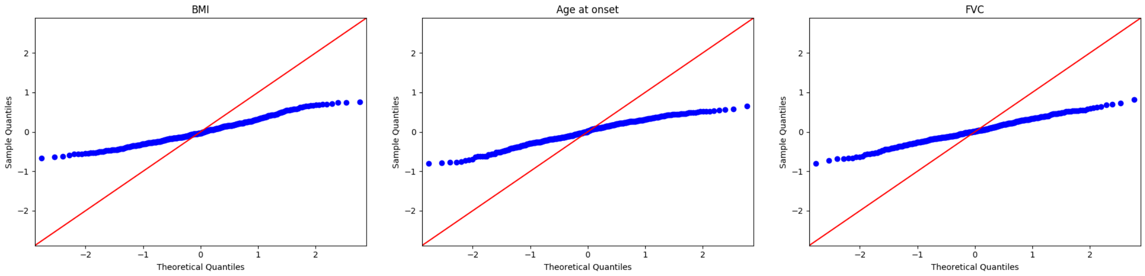

Figure 2.2: QQ Plot Normality Checks.

There are many statistical tests used to quantify whether a sample of data is drawn from a Gaussian distribution. Each test calculates a test-specific statistic, which can aid in the interpretation of the result. The p-value can be used to quickly and accurately interpret the statistic in practical applications. A significance level of 5% is used in the following analysis to evaluate the test hypotheses. In a statistical test, critical values are a range of predefined significance boundaries, at which the $H0$ can be failed to be rejected. The results are interpreted by failing to reject the null hypothesis if the calculated test statistic is less than the critical value, at the chosen significance level. Table 2.3 shows the results of the tests described below. The Shapiro-Wilk test, named for Samuel Shapiro and Martin Wilk, evaluates a data sample and quantifies how likely it is that is drawn from a Gaussian distribution. The p-value obtained for the FVC attribute states that the feature is likely drawn from a Gaussian distribution. The D'Agostino's K^2 test, named for Ralph D'Agostino, calculates summary statistics, namely kurtosis and skewness, to determine if the data distribution departs from the normal distribution. Skew is a quantification of the asymmetry in the distribution, while kurtosis quantifies how much of the distribution is in the tail. The p-value is interpreted against a significance level equal to 5%, and finds that the test dataset does not significantly deviate from normal, again with the exception of the FVC feature. Anderson-Darling Test is a statistical test, named for Theodore Anderson and Donald Darling, used to evaluate whether a data sample comes from one of among many known data samples. The test is a modified version of a more sophisticated nonparametric goodness-of-fit statistical test called the Kolmogorov-Smirnov test. A feature of the Anderson-Darling test is that it returns a list of critical values rather than a single p-value. At each significance level, the test confirms that FVC follows a normal distribution.

Test	Feature	Statistic	p-value	H0
Shapiro-Wilk	BMI	0.9860	0.0020	reject H0
	Age at onset	0.9820	0.0000	reject H0
	FVC	0.9960	0.5700	accept H0
D'Agostino's K^2	BMI	9.6040	0.0080	reject H0
	Age at onset	10.7030	0.0050	reject H0
	FVC	1.3970	0.4970	accept H0
Anderson-Darling	BMI	1.0280	0.0092	reject H0
	Age at onset	1.4040	0.0010	reject H0
	FVC	0.2920	0.5927	accept H0

Table 2.3: Statistical normality tests results.

Categorical data represent characteristics, that can be observed and sort into groups. If this data happens to be numerical, then the numbers would not have any mathematical meaning or proper order. To graph categorical data bar charts can be used, that use rectangular bars to plot qualitative data against its quantity, as shown in Figure 2.3. One useful way to explore the relationship between a continuous and a categorical variable is using a set of side by side box plots, one for each of the categories. Similarities and differences between the category levels can be detected, in the length and position of the boxes and whiskers. A box plot is a graph of the distribution of a continuous variable, based on the quartiles of the variables. The quartiles divide a set of ordered values into four groups with the same number of observations, the smallest values are in the first quartile and the largest values in the fourth quartiles. The plot uses a box to show the values that are larger than the first quartile and smaller than the fourth quartile, which is closest to the centre, i.e. the median, of the values. The values within the first and fourth quartiles are shown as a line, referred to as whiskers, and are further from the centre of the values. Boxplots are particularly useful for comparing the spread of categorical data, in Figure 2.4 the numerical features are represented for each progression group in the data set, also distinguishing patients according to the Gender feature. It clearly shows that for some features, such as ALS-FRS, ALS-FRS-R, ALS-FRSb, ALS-FRSr ALS-FRSsUL and R, the Slow group is in a higher value range than the others, while for features Age at onset and ALS-FRSsLL, the values appear lower. The box plot shows that the distributions of the feature values are different within the three levels of progression groups, except for the BMI attribute, which shows a similar shape for the three classes of patients. It is also noted that the differentiation is attributable to the group, but not to the sex of the patients. From this analysis it is therefore concluded that sex, for the purposes of clustering patients, is not a relevant characteristic, confirming the previous studies carried out on this subject.

Correlation Analysis is a fundamental method of exploratory data analysis, used to find a relationship between different attributes in a dataset. Statistically, correlation can be quantified by means of a correlation coefficient, which is always in the range $[-1, 1]$. A value of -1 and 1 indicates a totally negative and positive relationship, respectively. Any number close to zero represents a very low or non-existent relationship. Pearson's correlation coefficient is one of the statistical tests that measure the relationship, or association, between two continuous variables. It gives information about the magnitude of the association, or correlation, as well as the direction of the relationship. For the Pearson correlation, both variables should be normally distributed, and other assumptions include linearity and homoscedasticity. Linearity

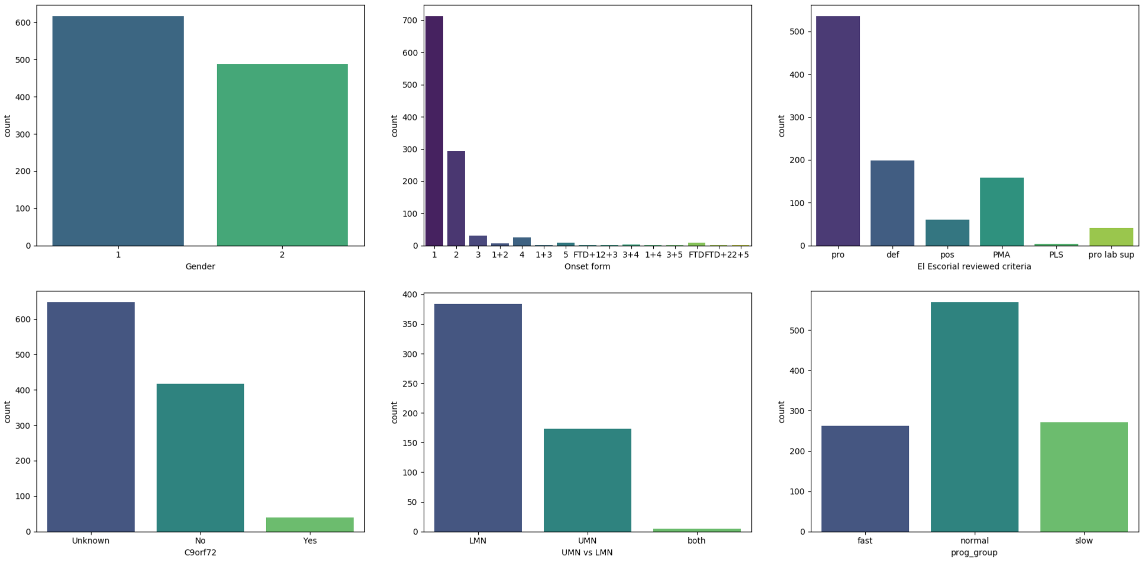

Figure 2.3: Categorical features Bar plot.

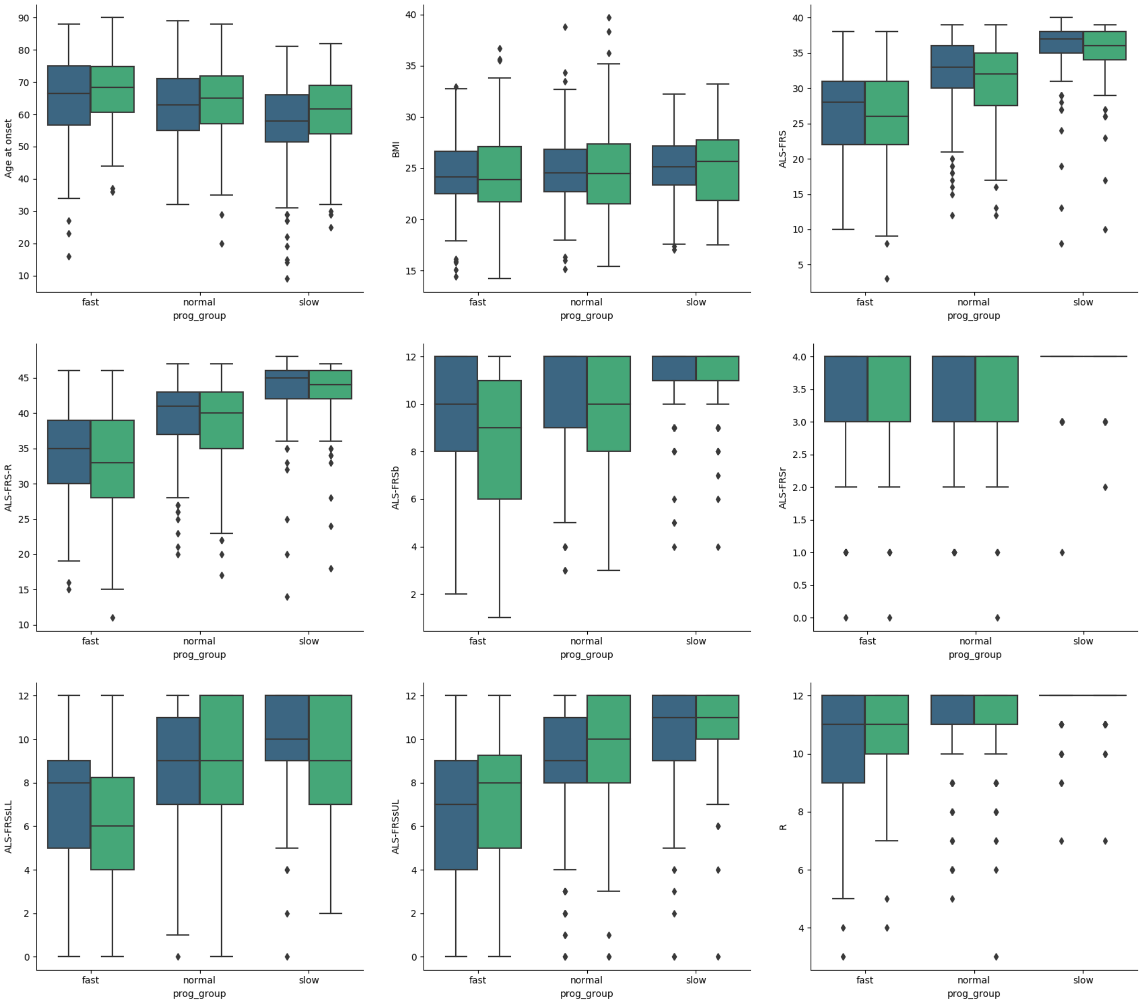

Figure 2.4: Box plot.

assumes a straight-line relationship between each of the two variables and homoscedasticity assumes that data is equally distributed about the regression line. For this reason, and because the dataset also contains categorical variables, the Spearman rank correlation is preferred. The Spearman correlation method is a non-parametric test, that computes the correlation between the rank values of the variables. The Spearman rank correlation test does not carry any assumptions about the distribution of the data and is appropriate when the variables are measured on a scale that is at least ordinal. In the formula below, n is the number of observations and d_i the difference between the ranks of corresponding variables.

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)} \tag{2.1}$$

Correlation plots can be used to quickly calculate the correlation coefficients without dealing with a lot of statistics, effectively helping to identify correlations in a dataset. A correlation matrix is a table, where each cell shows the correlation between two variables, used to summarize data. Figure 2.5 shows the correlation values between the features of the dataset. The line going from the top left to the bottom right is the main diagonal, which shows that each variable always perfectly correlates with itself. The matrix is symmetrical, thus the same correlation values are shown above and below the main diagonal, as a mirror image, and the strength of the correlation is provided by the depth of the colour.

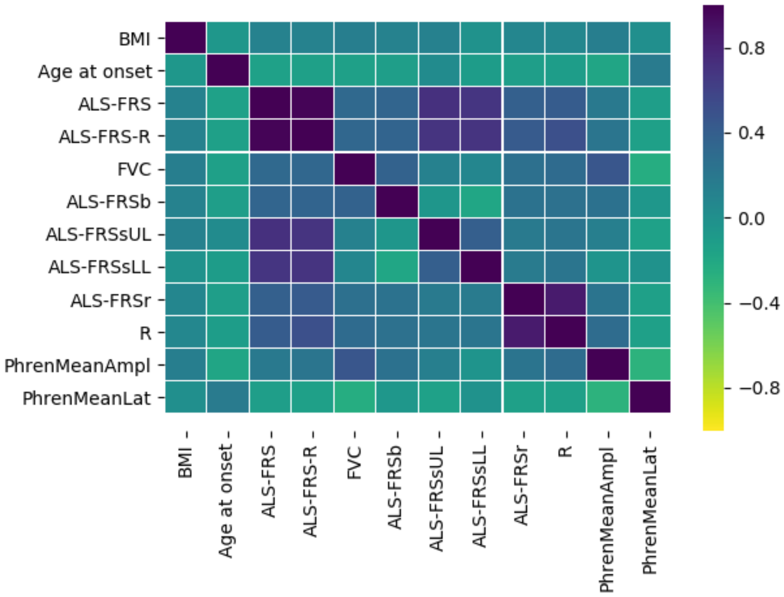

Figure 2.5: Correlation matrix.

The significance level of the correlation can be determined by using the correlation coefficient table, for degrees of freedom $df = n - 2$, where n is the number of observation, or by calculating the t value as follow:

$$t = \frac{r}{\sqrt{1 - r^2}} \sqrt{n - 2} \tag{2.2}$$

If the p-value is less than 5%, the correlation between the attributes is significant. The matrix shows that features belonging to the same subgroups have a higher correlation, confirming the consistent construction of the Clinical Profiles. In particular, features from the respiratory and neurophysiological tests show a significant correlation. The p-values obtained are consistent with the assumptions made previously and can be retrieved using the algorithm accessible at the following link, reported in Chapter 1.

2.4 Missing data

While most ML models require complete datasets for adequate learning, medical data are seldom complete and missing features are also common. Missing data may originate from data censoring in longitudinal studies or differences in data acquisition, which would be helpful to estimate the likelihood of diagnoses and predict treatments effectiveness. Missing data may seriously compromise inferences, especially if missing data are not handled appropriately. The potential bias due to missing data depends on the mechanism causing the data to be missing, and the analytical methods applied to amend the missingness. Therefore, the analysis of data with missing values requires careful planning and attention. Imputation can be conceptually split into methods that are applied prospectively, where a possibly complete or incomplete training database is used to estimate missing values for an incomplete data vector, and methods that are applied retrospectively, where information from an incomplete database is extracted to estimate its own missing values. In a clinical context, prospective imputation is of greater utility, allowing new patient records to be processed, although retrospective is more commonly used in research contexts in which an entire database is often analysed at the same time [56]. The most common way of performing retrospective imputation is case deletion, in which every sample with at least one missing value is removed from the database. When there are a huge number of records with missing values, data omission may cause a major loss of information. Missing data imputation is a very critical issue, especially when dealing with the medical field, as diagnosis and treatment are affected by the models' output. When the missed items are limited, compared to the scale of the dataset, the missed data omission might be accepted. On the other hand, if the number of missed items rises with the dimension of the dataset, data imputation is essential to preserve and even increase the statistical power of the data. Table 2.4 shows the database features with the respective number of missing values in the column *Na*. In this work, performance values of the algorithms, reported in Chapter 4, obtained either by removing *Na* data or by imputation, are compared. The results are not significantly different, thus leading to the momentary choice of not implementing any data imputation method. This may be due to several factors, such as the limited size of the dataset, and the small number of *Na*, the removal of which does not significantly affect the performance of the algorithm. In fact, the feature with the highest number of *Na* values is PhrenMeanAmpl, present exclusively in the Respiratory Profile dataset, which nevertheless returns good performance values, sometimes better than the Prognostic and Functional Profile. Moreover, for the purpose of this work, that is to clustering patients, exploring different techniques and setting algorithm parameters to increase the quality of the clusters obtained, the imputation of values is superfluous and deviant. However, if all the data collected in the dataset are considered, thus allowing a more complete and temporal analysis, imputation methods may be useful to increase the performance of the algorithm. The description of the imputation techniques given below is justified by their possible future use in more complex and comprehensive analyses.

The first step to missing data management is to explore the mechanisms behind missing data features. Features can be missing completely at random, without modifying the overall data distribution, missing at

Name	**Na**
BMI	237
MND	86
Age at onset	1
ALS-FRS	113
ALS-FRS-R	112
ALS-FRSb	114
ALS-FRSsUL	117
ALS-FRSsLL	117
ALS-FRSr	117
R	117
PhrenMeanAmpl	473

Table 2.4: Portuguese ALS dataset *Na* values count.

random, when missing feature patterns are based on other features available in the dataset or non-missing at random for the remaining cases. Depending on the type of missing data, an appropriate imputation method should be selected. A possible solution to process missing data is given by the replacement of the missed value with the mean value of its cluster, after eliminating the null values. The imputation is done by dividing the dataset according to its classes, i.e. Slow, Normal and Fast, calculating the average value for each class, searching for missed values and finally replacing the missed values with the average of its class. Until recently, the most frequently used method for imputation was the mean or median replacement in which the missing variables are replaced with a constant, specifically the mean or median of the database as a whole [57, 58]. However, this reduces the variability in a database and thus can bias down-stream statistical methods [58]. A common but more nuanced method for data imputation is fitting a linear model to the data, in which individual variables are sequentially imputed in an entire database using simple linear regression, starting with the variable with the least number of missing values and using complete data points to initiate the process [59]. The downside of this approach is that it is designed specifically for retrospective use, so it is unclear how accurate it would be on unseen data that does not contribute to the construction of the model. Alternatively, Principal Component Analysis (PCA) can be naturally extended to perform prospective imputation, by removing the PCA eigenvector components corresponding to the missing values when calculating the PCA scores, but using the full eigenvector when transforming the scores back into the data space [60]. Another method of dealing with missing data is to use Autoencoders, which are a family of artificial neural networks, trained to reproduce its input with a lower-dimensional immediate stage or bottleneck [61]. These networks often consist of a series of encoding layers, leading up to a central bottleneck, which is then followed by symmetric series of decoding layers. Autoencoder-based approaches to analysing medical data have been shown to provide useful patient representations for screening broad disease classes [62].

2.5 Progression Groups

Although the average survival of an ALS patient is about 3-5 years, survival can vary between less than a year to over 10 years, confirming the heterogeneity of the disease. Disease progression can be analyzed by considering the ALS Functional Rating Scale (ALSFRS) decay in a period of time. The ALSFRS is a standard test, used by clinicians to estimate the outcome of a treatment or the progression of the disease. Since this scale has only a small respiratory component, the ALS functional rating scale-revised (ALSFRS-R) was later proposed, which adds additional respiratory assessments, becoming the preferred test to quantify disease progression. The test is composed of 13 questions, to be answered by the patient using a 5-point scale, ranging from 0 to 4, where 0 corresponds to the worse condition and 4 to the best. The questions addressed by this scale are the following:

- Speech
- Salivation
- Swallowing
- Handwriting
- Cutting and handling utensils
- Dressing and hygiene
- Turning in bed and adjusting bedclothes
- Walking
- Climbing stairs
- Breathing
- Dyspnea
- Orthopnea
- The need for respiratory support

By measuring the change in ALSFRSR over time, the disease progressing can be estimated, computed using the information about the time of the first occurrence of symptoms and the time of the first appointment:

$$ProgressionRate = \frac{48 - ALSFRSR_{Visit_1}}{t_{Visit_1} - t_{Symptoms_1}} \tag{2.3}$$

where 48 is the maximum score of the ALSFRSR scale and the assumed score of a patient at the time of its first symptoms, $ALSFRSR_{Visit_1}$ is the ALSFRSR score of a given patient at the beginning of the first appointment, and $\Delta t_1 = t_{Visit_1} - t_{Symptoms_1}$ is the time in months between the time of first symptoms and the first visit. Given the heterogeneous nature of ALS, the progression rate is highly variable across all patients. Moreover, patients with different progression rates usually have different prognoses. Three Progression Groups are created, i.e. Slow, Neutral and Fast, from a cohort of 1590 patients using the information at the time of disease onset and the ALSFRSR scale at the first appointment. Only 989 of 1590 patients (62.2%) can be used for analysis, as the other 601 patients lacked at least one of the information needed to compute the progression rate. Using the progression rate of the selected patients, the distribution presented in Figure 2.6 is obtained. The higher the progression rate, the faster the patient's disease development, while lower progression rates are usually associated with a slower disease progression. Following consensual clinical insight, patients are stratified into three disease progression groups. The 25% of the patients with higher progression rates are grouped together and labeled as Fast Progressors. The 25% of the patients with lower progression rates are also grouped together to create the Slow Progressors group. The remaining 50%, with an average progression, are grouped together and called Neutral or Normal Progressors. In the analysis carried out in Chapters 3 and 4, the possibility of further subdivision of patients is examined. In fact, the performance of some algorithms is improved by identifying 5 groups of patients, instead of the conventional Slow, Normal and Fast groups. The two new groups are positioned at the borderline between one group and the other and can be referred to as Normal-Slow and Normal-Fast, as shown in Figure 2.7. Network metrics, such as modularity, also show the difficulty of effectively distinguishing groups, particularly the Slow from the Normal, and the Fast from the Normal. Indeed, although the Slow and Fast progressors exhibit distinct characteristics, even visually, the Normal group lies somewhere in between the two and represents roughly an average of the two. However, it is important to notice that these new groups need validation, including by clinicians, before they can actually be used and their performance verified. In Section 4.2, the 5 clusters are evaluated only from a graphical point of view, given the impossibility of calculating the accuracy obtained, as the two new proposed groups are not present in the original dataset.

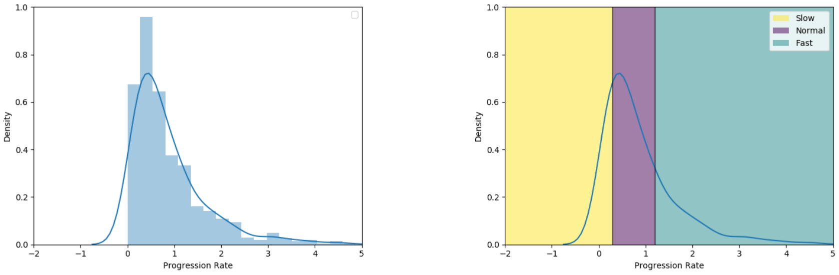

Figure 2.6: Progression Rate Distribution among all patients.

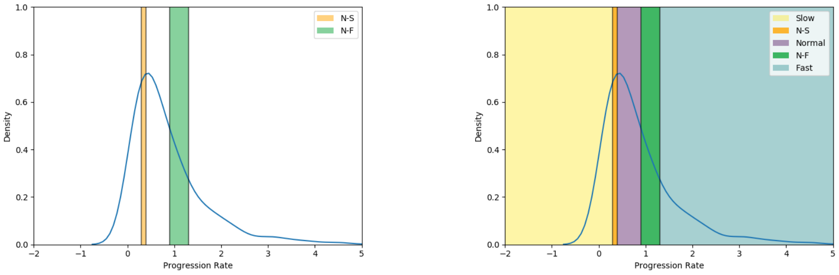

Figure 2.7: Progression Rate Distribution among all patients, with 5 Progression Groups.

2.6 Creating datasets with Clinical Profiles

The groups obtained by clustering data via each Clinical Profile are then used to determine Patient Profiles, in order to create patient groups that best reflect those identified by clinicians. Clinical Profiles are sets of features that describe specific patient conditions, thus defined in close collaboration with clinical experts. For each Clinical Profile, a new version of the dataset is created, by selecting the distinctive features. Three sets of clinically relevant features are used, such as Prognostic, Respiratory and Prognostic sets, called Clinical Profiles [49], composed by a set of features that are focused on different aspects of the disease:

- Prognostic Profile: the set of features described in the literature as good prognostic features for ALS.
- Functional Profile: ALS-FRS and ALS-FRS-R scales and sub-scales assessing the functional status of the patient.
- Respiratory Profile: the set of features describing the respiratory status of the patient.

Table 2.5 shows the features included in each Clinical Profile. Figure 2.8 shows the bar plots of the progression group variable. It can be seen that all profiles have more patients classified as normal than the others, as expected given the original composition of the dataset. In addition, the Functional Profile has a higher number of patients than the other profiles, which is due to the selection of data and deletion of missing data, as described in Section 2.4.

Clinical profile	**Features used**
Prognostic	BMI, MND, Age at Onset, FVC ALS-FRS, ALS-FRS-R
Functional	ALS-FRS, ALS-FRS-R, ALS-FRSb, ALS-FRSsUL ALS-FRSsLL, ALS-FRSr, R
Respiratory	FVC, PhrenMeanAmpl, ALS-FRSr, R

Table 2.5: Set of Features used to perform Clustering for each Clinical Profile.

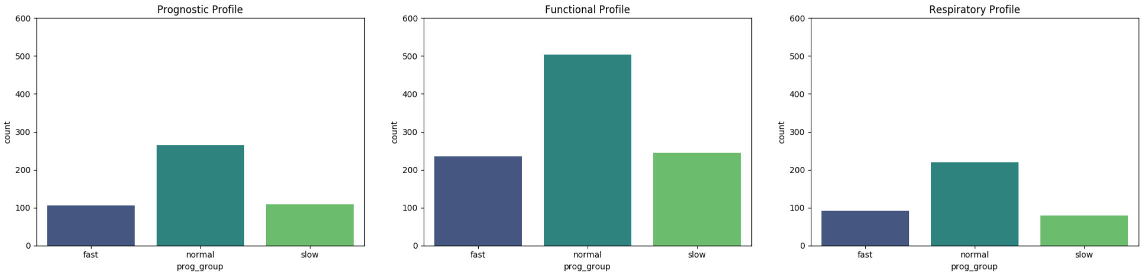

Figure 2.8: Clinical Profiles Progression Group.

2.6.1 Prognostic Profile

A prognostic factor is any variable associated with a subsequent outcome, such as death or disability among people with a disease or health condition, that can be used to estimate the chance of recovery from a disease or the chance of the disease recurring. Prognostic factors range from simple measures, such as age, gender, temperature, or pulse rate, to test results such as X-rays or psychological scores, whilst novel biomarkers and genetic information are increasingly studied. Different values of a prognostic factor are associated with a different prognosis and can be used to stratify overall prognosis estimates. In Figure 2.9, the histograms of the features belonging to the Prognostic Profile are shown on the left, the correlation matrix is shown on the right.

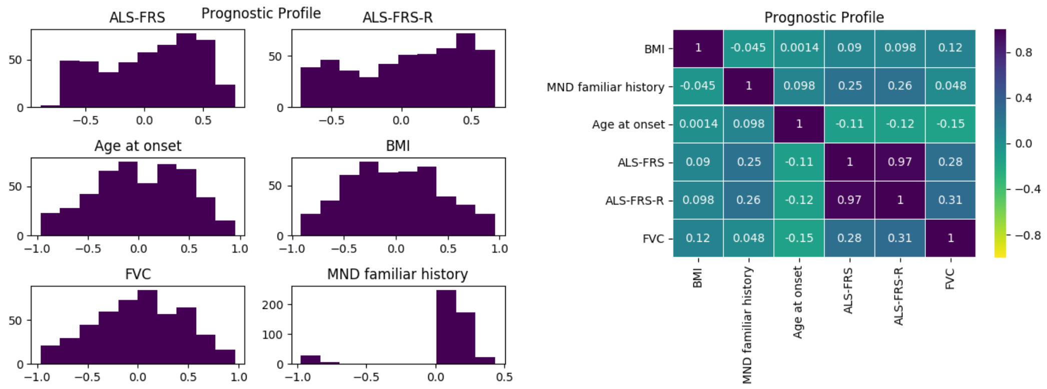

Figure 2.9: Prognostic Profile Visualization.

2.6.2 Functional Profile

A functional diagnosis is an analytical description of the functional impairment of a patient's psychophysical state. Therefore, the Functional Profile concerns the abilities and functions of the subject under examination, and synthesises this information within a psychological-functional framework, that enables the scope of the pathology found at the time of assessment to be understood. The subgroup of Functional Features is a cognitive tool that, starting from the impairment and its effects on the subject, aims to identify the set of disabilities and difficulties induced by the disease. It also identifies the framework of capacities and a developmental perspective, that highlights the developmental potential of each individual, which is an extremely significant prediction for subsequent intervention.

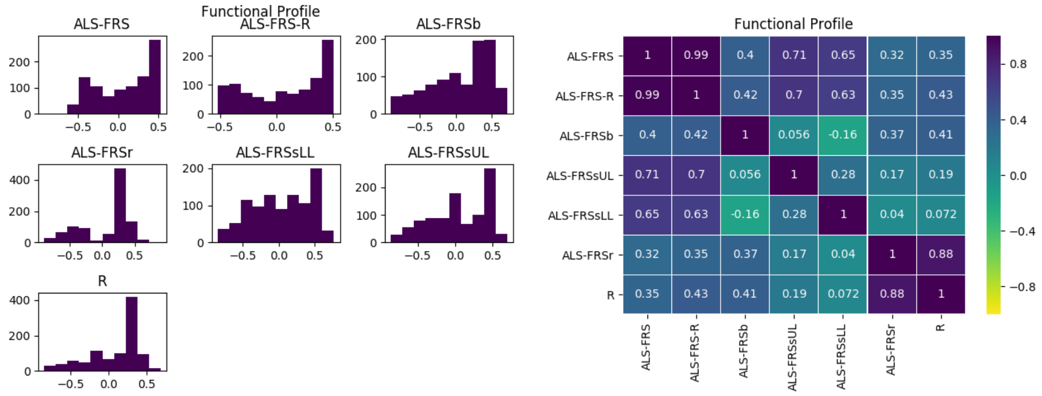

Figure 2.10: Functional Profile Visualization.

2.6.3 Respiratory Profile

The Respiratory profile groups all the features of the dataset that concern the respiratory system, that can help clinicians to diagnose and decide the treatment of certain lung disorders. Pulmonary function tests (PFTs) are noninvasive tests, that show how well the lungs are working. The tests measure lung volume, capacity, rates of flow, and gas exchange. Forced vital capacity (FVC) is the amount of air exhaled forcefully and quickly after inhaling as much as a patient can. The amount of air inhaled and exhaled in test results are compared both with the average values of other patients of the same age, height, sex, and race and with other tests the same patient has previously undergone. Phrenic nerve stimulation is a non-volitional test, that can be performed quickly in most patients and it is a powerful predictor of survival in ALS.

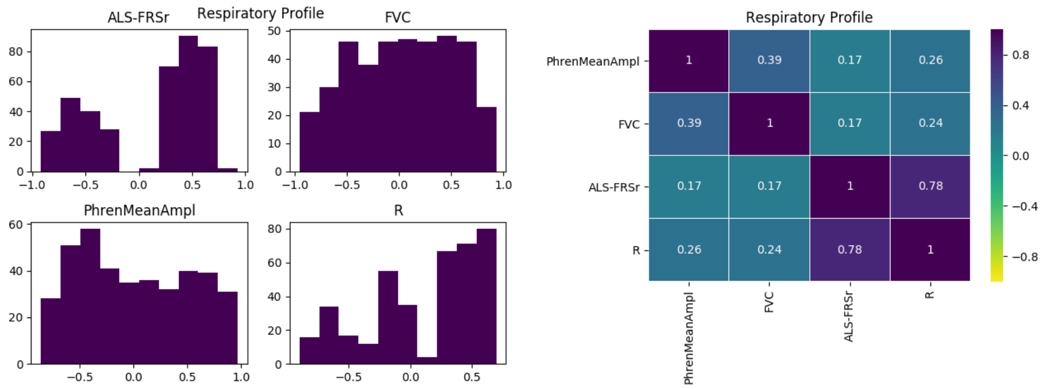

Figure 2.11: Respirtory Profile Visualization.

2.7 PCA

The Principal Component Analysis (PCA) is a widely used method of reducing the dimensionality of high-dimensional data, followed by visualizing two of the components on a scatterplot. Standardization prior is performed on the continuous initial variables so that each of them contributes equally to the analysis since PCA is quite sensitive regarding the variances of the variables. In order to identify correlations between the features, the covariance matrix is computed, to understand how the variables of the input data set are varying from the mean with respect to each other, i.e. if there is any relationship between them, and to identify any redundant information. The covariance matrix is a *pxp* symmetric matrix, where *p* is the number of features, that shows the covariances associated with all possible pairs of the initial variables. Since the covariance of a variable with itself corresponds to its variance, the main diagonal contains the variances of each initial variable. Moreover, since the covariance is commutative, the entries of the covariance matrix are symmetric with respect to the main diagonal, which means that the upper and the lower triangular portions are equal. If the sign of the covariance is positive, the two variables increase or decrease together, i.e. they are correlated, otherwise, one increases when the other decreases, i.e. they are inversely correlated. Principal components are new variables that are constructed as linear combinations or mixtures of the initial variables. These combinations are done in such a way that the new variables, i.e. principal components, are uncorrelated and most of the information within the initial variables is squeezed or compressed into the first components. Organizing information in principal components allows to reduce dimensionality without losing much information, discarding the components with low information and considering the remaining components as the new variables. However, it is important to notice that the principal components are less interpretable and don't have any real meaning, since they are constructed as linear combinations of the initial variables. Geometrically speaking, principal components represent the directions of the data that explain a maximal amount of variance, i.e. the lines that capture most information of the data. The larger the variance carried by a line, the larger the dispersion of the data points along with it, and the larger the dispersion along a line, the more information it has. As there are as many principal components as there are variables in the data, principal components are constructed in such a manner that the first principal component accounts for the largest possible variance in the data set. By definition, the direction of every single principal component is not uniquely determined, all points are mirrored along with one of the axes, without changing the meaning of the plot. Groups are marked with different colours, according to their progression group, in order to allow a comparison. Eigenvectors and eigenvalues are linear algebra concepts, computed to determine the principal components of the data. Computing the eigenvectors and ordering them by their eigenvalues in descending order, allows finding the principal components in order of significance. The feature vector is a matrix, that has as columns the eigenvectors of the components that are taken and not discarded. This enables a dimensionality reduction, as choosing to keep only q eigenvectors out of p, the final data set will have only q dimensions. Finally, the feature vector is used to reorient the data from the original axes to the ones represented by the principal components. This can be done by multiplying the transpose of the original dataset by the transpose of the feature vector:

$$FinalDataset = FeatureVector^T * StandardizedOriginalDataset^T \tag{2.4}$$

PCA aims to estimate how many components are needed to describe the data, by looking at the cumulative explained variance ratio as a function of the number of components. The Clinical Profiles curves are shown in Figure 2.12, the first two components contain approximately 80% of the variance, while around three components are needed to describe close to 90% of the variance. Figure 2.13 shows the two-dimensional principal subspace for the Clinical Profiles.

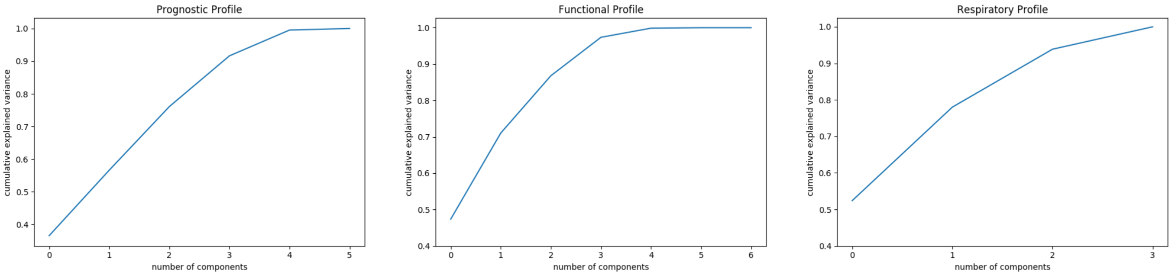

Figure 2.12: Clinical Profiles Explained Variance Ratio.

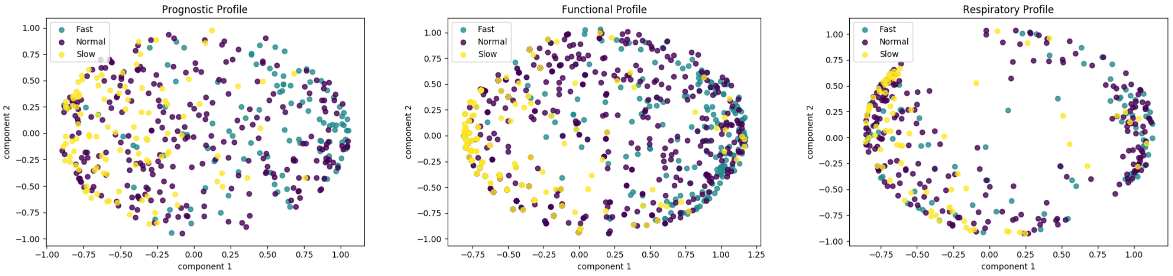

Figure 2.13: Clinical Profiles PCA visualization.

2.8 Summary

In this chapter the Portuguese ALS Dataset is analysed and explored, using descriptive and inferential statistical techniques. The data are normalised and standardised, the correlation between attributes is assessed both numerically and visually. An analysis of missing values is started, where different techniques to deal with them are proposed. Finally, the construction of the Clinical Profiles, with the description of each subgroup of features, is reported. The key contribution is the possibility of further subdividing the patients, identifying other groups besides the Slow, Normal and Fast of the clinicians.

Chapter 3

Patient stratification using Network Science Approaches

A detailed analysis of the construction of a network that allows the stratification of patients is reported. The groups obtained, by clustering data using each Clinical Profile, are then used to determine the parameters of the network that allow the creation of patient groups that best reflect those identified by clinicians. This chapter proposes an unsupervised learning strategy, where different distance measures are used and tested, that builds networks of patients then analyzed using different metrics and from a modular point of view. The found modules, or patient communities, can then be used in a supervised learning fashion to train expert models for discriminating subgroups of patients. For the analysis done in this work, Python 3 with the NetworkX library is used.

3.1 Methods

This section presents and discusses the methodology used in this work. The workflow is in Figure 3.1.

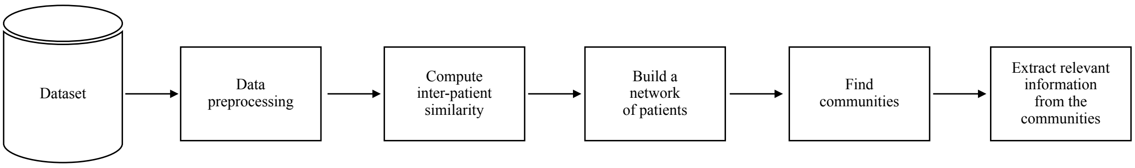
Dataset
Data preprocessing
Compute inter-patient similarity
Build a network of patients
Find communities
Extract relevant information from the communities

Figure 3.1: Workflow used in this work for community finding and interpretation using a network of ALS patients.

3.1.1 Patient similarity networks

The patient similarity network (PSN) paradigm is a recently developed analytical framework, that addresses a number of challenges in data analytics and is naturally interpretable [4]. In a PSN, each node is an individual patient and an edge between two patients corresponds to the pairwise similarity for a given feature. In this paradigm, each input patient data feature is represented as a network of pairwise patient similarities, used to identify patient subgroups or predict the outcome. As a simple example of the concept, the progression of ALS disease can be represented as a PSN, as shown in Figure 3.2.

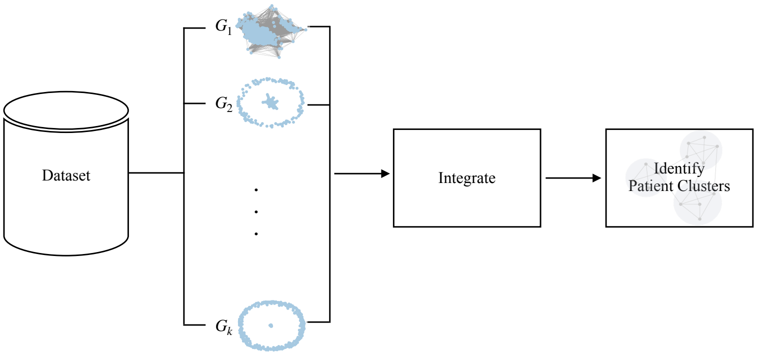

Figure 3.2: PSNs for a hypothetical example of predicting ALS risk.

Nodes are patients and edge weights reflect the data type similarity. Patients affected by the disease, whose progression is considered slow, would be tightly connected to each other and those with a fast progression would separately be tightly connected. If a new patient presents feature values indicating a slow development of the disease, he would be more similar to the slow group and the algorithm would predict him as such. However, in Figure 3.2 there is a process that is not carried out in this preliminary work, that will be explored in more extensive future research, namely integration. In this step, a single network is created, using all the networks built using the Clinical Profiles, which generates a consensus clustering and general patient profiles. When compared to other clustering and classification approaches, these methods can demonstrate superior performance. PSNs naturally handle heterogeneous data, as any data type can be converted into a similarity network by defining a similarity measure. Once converted, all data are represented in the same manner as a network that can be directly input into analysis methods [50]. Missing data are also naturally handled, as a patient missing in one network may be in another and could still be used. Furthermore, patient similarity measures are robust even if part of the input data vectors is missing. Representing patients by similarity is conceptually intuitive because it can convert the data into network views, where the decision boundary can be visually evident. Moreover, algorithms that take PSNs as input use data transformed from the raw values, thus sensitive raw data are not directly used. However, since the research community increasingly pools its patient cohorts, to expand the sample sizes for clinical discovery, protocols and technologies for maintaining patient privacy are developed in parallel [63].

3.1.2 Building Network of Patients

In order to build a network of patients, a measure that reflects the relationship between two patients is needed. An overall distance measure can be given by averaging the distance calculated for all attributes in the Clinical Profiles, although a weighted version can be used to assign greater relevance to a sub-type of data. Various measures of similarity can be analysed for this purpose and compared, such as Euclidean distance, Hamming distance, Manhattan/City Block distance, Minkowski distance, Cosine distance, Correlation distance, Mahalanobis distance, Yule distance and Matching distance [47]. After computing the distance matrix for all the patients, the network or graph is built. However, in network analysis, it is usual to use similarities instead of distances, which can be derived in two distinct ways. The first is computing the similarities as $S_{ij} = 1 - Dij$, where S is the similarities matrix and D is the distances matrix. The other is based on the binary adjacencies matrix A, defined as:

$$A_{ij} = 1 \iff D_{ij} \leq \tau \tag{3.1}$$

where τ is a given threshold. As stated above, in a network context, the graph nodes represent the patients, whereas the edges represent their connection. When the similarities matrix is used, each edge has an associated weight representing the similarity between both patients. Eventually, some edges with lower similarities can be filtered out. In the case, where the adjacencies matrix is used, each edge states that the two patients it connects are similar, i.e. their distance is below the threshold τ. Several networks are built, visualized and analyzed, by using metrics such as the network density, modularity, average path length and many others, as reported in Section 3.2.

3.1.3 Finding communities in the network of patients

The identification of communities in a network is of crucial importance, as it may help to uncover a priori unknown functional modules. The problem of community detection requires the partition of a network into communities of densely connected nodes, with the nodes belonging to different communities being only sparsely connected. Each patient in the dataset is associated with a node. The following procedure is applied to the subgroups of features identified, to the different distance measurements and to the various threshold values. Suppose that communities are identified considering the subgroup of Functional Features, using the Euclidean distance and with a threshold equal to 0.7. The algorithm runs all the nodes, i.e. the patients, and compares them in pairs, calculating the similarity values for each feature. Subsequently, the average of similarities is considered, and if it exceeds the threshold fixed, an edge is created between the nodes. The result is a network, created by grouping nodes that are similar in general for that subgroup of attributes. Therefore, the algorithm builds the networks considering the following three variables:

- Subgroup of features
- Distance measurement
- Threshold

Subgroup of features

Each set of Patient Profiles is obtained by clustering the patients using subsets of features, listed in Table 2.5. When a patient undergoes a hospital medical inspection, clinicians can decide which Clinical Profile, or set of Clinical Profiles, are more adequate to predict the desired clinical outcome. For each Clinical Profile selected, the patient's data are compared with the average of those of the patients belonging to the various clusters. The patient is then assigned to the Patient Profile more suitable. Then, the data can be used as input to the specialized model for that Profile, in order to predict the need for NIV, or any other outcome.

Distance measurements

Distance measures play an important role in machine learning and are an objective score that summarizes the relative difference between two objects in a problem domain. In this context, the two objects are rows of data that describe a subject, i.e. a patient. When calculating the distance between two examples or rows of data, it is possible that different data types are used for different columns, e.g. real, boolean, categorical, and ordinal. Different distance measures may be required, that are then summed together into a single distance score. Since numerical values may have different scales, the calculation of distance measures can be highly impacted, thus it is a good practice to normalize or standardize numerical values before computing the distance measures. In fact, if columns have values with differing scales, those with larger values will dominate the distance measure. Below, a detailed description of the distance measurements used is reported.

The Euclidean distance coincides with the most basic physical idea of distance, and it is generalized to multidimensional points. If a and b are points of $\mathbb{R}^n$, where n is the number of dimensions considered, i.e. variables, the Euclidean distance from a to b is given by:

$$d_{E(a,b)} = \sqrt{\left(\sum_{i=1}^{n}(a_i - b_i)^2\right)} \tag{3.2}$$

The Manhattan distance, also called the Taxicab distance, calculates the distance between two real-valued vectors. Given a and b, it is computed as in (3.3).

$$d_{Ma(a,b)} = \sum_{i=1}^{n} \mid a_i - b_i \mid \tag{3.3}$$

The Minkowski distance is a generalization of the Euclidean and the Manhattan distances, and adds a parameter, called the *order* or p, that allows different distance measures to be calculated. When p is set to 1, the calculation is the same as the Manhattan distance, while when set to 2, it is the same as the Euclidean distance. Intermediate values provide a controlled balance between the two measures. It is common to use Minkowski distance when implementing a machine learning algorithm that uses distance measures, as it gives control over the type of distance measure used for real-valued vectors via the hyperparameter p. The distance between a and b is given by:

$$d_{Mi(a,b)} = \left(\sum_{i=1}^{n} \mid a_i - b_i \mid^p \right)^{1/p} \tag{3.4}$$

The Cosine distance measures the distance between two vectors of an inner product space. It is measured by the cosine of the angle between two vectors and determines whether two vectors are pointing in roughly the same direction. The smaller the angle, the lower the cosine distance, thus two vectors in exactly opposite directions, i.e., 180° between them, would result in a value of -1, whereas two identical vectors, i.e., 0° between them, would yield a value of 1. Unlike measuring Euclidean distance, cosine similarity captures the orientation and not the magnitude. If negative values are encountered in the input, the cosine distances will not be computed. The distance between a and b is defined as:

$$d_{Cos(a,b)} = 1 - \frac{\sum_{i=1}^{n} a_i b_i}{\sqrt{\sum_{i=1}^{n} a_i^2}\sqrt{\sum_{i=1}^{n} b_i^2}} \tag{3.5}$$

Distance correlation is a measure of association strength between non-linear random variables. It goes beyond Pearson's correlation because it can spot more than linear associations and it can work multidimensionally. Given a and b points of $\mathbb{R}^n$, distance correlation can range from 0 to 1, where 0 implies independence between a and b and 1 implies that the linear subspaces of a and b are equal. Distance correlation is not the correlation between the distances themselves, but it is a correlation between the scalar products. The distance correlation of two variables is obtained by dividing their distance covariance by the product of their distance standard deviations. The formula for distance correlation is as follows, where $\bar{a}$ and $\bar{b}$ indicate the average of the vector a and b, respectively:

$$d_{Cor(a,b)} = 1 - \frac{\sum_{i=1}^{n}(a_i - \bar{a})(b_i - \bar{b})}{\sqrt{\sum_{i=1}^{n}(a_i - \bar{a})^2}\sqrt{\sum_{i=1}^{n}(b_i - \bar{b})^2}} \tag{3.6}$$

Thresholds

Numerical attributes are geared to model edges, that connect nodes if they have the same values or significantly close values. Two values are considered significantly close if the normalized difference d between them is less than or equal to a specified threshold. The normalized difference d between two data points a_i and b_i for the numerical attribute $f_i \in f_1, ..., f_p$, with p equal to the total number of attributes in the dataset considered, is defined according to the following equation:

$$d_{(a,b,i)} = \frac{d_{a_i,b_i}}{\mid \max(f_i) - \min(f_i) \mid} \tag{3.7}$$

where $\max(f_i)$ and $\min(f_i)$ represent the maximum and minimum values respectively of the attribute f_i. When a link is demonstrated between two nodes with numerical attributes, it is checked if the two nodes have the same, or a significantly close value, for any numerical attribute. If the similarity between the values of the patient attributes exceeds a certain threshold, a link is created between them. Four different threshold values are analysed, i.e. 0.6, 0.7, 0.8 and 0.9.

3.2 Results evaluation and discussion

3.2.1 Experimental setup

The code, implemented using Python 3, all the needed packages are imported. In order to correctly run the code, the directory path has to be changed, indicating the one where the additional data file dataWithoutDunnoNIV_PP.csv and make_prog_groups.xlsx are located. The first part of the code concerns the data preprocessing, i.e. the creation of three datasets using three different subgroups of features, reported in Table 2.5, plus an attribute containing cluster labels, i.e. Slow, Normal and Fast [25], as additional information used for a further subgraphs analysis. Then, the function *generateLinkingConditions()* is created, that given the indexes of two nodes and a dataset, returns three similar measures between them. In each of the three datasets, all the attributes are numerical, except for the cluster label, but in different scales, thus the function first normalizes them between 0 and 1. Then all the normalized values are collected into two arrays, and five distance measures are computed, i.e. Euclidean, Manhattan, Minkowski, with order p equal to the number of attributes analyzed, Cosine and Correlation distances, handling *Nan* cases. Finally, the similarities measures are returned, obtained as described in Section 3.1.2, as a tuple. Furthermore, twenty networks are constructed for each Clinical Profile, and *generateLinkingConditions()* is used for each possible couple of nodes. Sixty networks are obtained, by exploring the five similarities measures using different threshold values, i.e. 0.6, 0.7, 0.8 and 0.9. For each network, different metrics are collected, reported in Section 3.2, such as the number of edges, density, average degree and centrality measures, that consider all the graph and the subgraphs corresponding to the clusters known a priori, i.e. Slow, Normal and Fast. The plots of each network and their metrics are given as output of the code, also shown in Figure 3.3.

Algorithm analysis

For each of the three datasets, the function *generateLinkingConditions(node1,node2,dataset)* is computed for all the possible node pairs of patients, leading to $\binom{1104}{2} * 3 = 608856 * 3 = 1826568$ comparisons. Moreover, varying three parameters, i.e. attribute subsets, similarity measures and thresholds, 60 networks are constructed and for each of them, several metrics are computed. The high number of comparisons is justified by the need to confirm the hypotheses formulated and the conclusions drawn on the networks. In fact, the possibility of observing the constructed graphs and interpreting the results makes it possible to obtain more detailed and reliable information. In conclusion, it has to be highlighted that the code needs around $6 - 7$ hours to run. Despite the time and computational complexity, this kind of analysis is useful to compare different network models, allowing the possible inclusion of further network parameters to be explored in the future.

3.2.2 Network metrics

In the following subsections, several network metrics are computed and analysed. For some measures, the values obtained both on the whole graph, indicated as G, and on the three subgraphs obtained using the progression groups provided by the clinicians, *G Slow*,*G Normal* and *G Fast*, are explored. The results obtained are reported in the Appendix A.1. In particular, the values shown below refer either to the entire subgraph or to the individual nodes, depending on how the measurement is defined. For measurements related to individual nodes, the results shown are obtained by averaging the node values. Considering the variable parameters described above, 60 different networks are obtained for each subgroup of features. The different networks are indicated as G_{ijk}, where i indicates the subgroup of features considered with $i = 1, \ldots, 3$, i.e. the subgroup relative to the Prognostic, Functional and Respiratory Profile, respectively. The threshold considered is indicated with $j = 1, \ldots, 4$, i.e. a threshold equal to 0.6, 0.7, 0.8 and 0.9, respectively. Finally k indicates the distance measurement used, with $k = 1, \ldots, 5$, which are the Euclidean, the Manhattan, the Minkowski, the Cosine and the Correlation, respectively.

Density

Density captures how many edges there are in a network, divided by the total possible number of edges. In an undirected network of size n, there will be $(n * (n-1))/2$ possible edges. Considering the matrix underlying each network, $(n * (n-1))$ refers to the number of rows times the number of columns minus 1 so that the diagonal is excluded, i.e. ties to oneself. The number is then divided by 2 in the case of an undirected network since it is symmetrical. The density for undirected graphs is defined as in the equation 3.8, where n is the number of nodes and m is the number of edges. The values obtained for the number of edges is reported in the Appendix A.1. The density values are shown in Tables A.1 and 3.1, respectively.

$$d = \frac{2m}{n(n-1)} \tag{3.8}$$

Threshold	**Distance**	**Prognostic profile**	**Functional profile**	**Respiratory profile**
0.6	Euclidean	0.49801	0.1819	0.7094
	Manhattan	0.2625	0.0524	0.5717
	Minkowski	0.6240	0.3514	0.7472
	Cosine	0.9838	0.8023	0.9107
	Correlation	0.6357	0.2847	0.4331
0.7	Euclidean	0.3322	0.0897	0.5704
	Manhattan	0.1713	0.0337	0.4607
	Minkowski	0.4461	0.2163	0.6144
	Cosine	0.9578	0.7953	0.9097
	Correlation	0.5679	0.2246	0.4279
0.8	Euclidean	0.1690	0.0423	0.3941
	Manhattan	0.0909	0.0215	0.3792
	Minkowski	0.2316	0.0739	0.3990
	Cosine	0.8953	0.7762	0.9041
	Correlation	0.4739	0.1639	0.4183
0.9	Euclidean	0.0494	0.0146	0.2676
	Manhattan	0.0319	0.0056	0.2650
	Minkowski	0.0614	0.0254	0.2685
	Cosine	0.8081	0.7079	0.8967
	Correlation	0.3359	0.0996	0.3861

Table 3.1: Density.

Average degree

The average degree of an undirected graph is used to measure the number of edges compared to the number of nodes, given by the average number of edges per node in the graph. Since this type of measurement works better for the undirected case compared to the directed, the datasets used in this work are suitable. Given n the number of nodes and m the number of edges, reported in Table A.1, the average degree is computed as in (3.9) and the values obtained are collected in Table 3.2.

$$AverageDegree = \frac{TotalEdges}{TotalNodes} = \frac{m}{n} \tag{3.9}$$

Threshold	Distance	Prognostic profile	Functional profile	Respiratory profile
0.6	Euclidean	234.610	165.172	276.578
	Manhattan	174.550	137.924	244.742
	Minkowski	273.858	211.820	298.924
	Cosine	813.867	714.322	773.795
	Correlation	622.918	430.429	511.785
0.7	Euclidean	191.874	145.490	235.242
	Manhattan	154.902	133.334	227.274
	Minkowski	222.292	162.150	237.538
	Cosine	799.612	710.493	773.197
	Correlation	585.721	397.416	508.980
0.8	Euclidean	154.718	135.156	209.232
	Manhattan	139.224	130.574	203.910
	Minkowski	168.994	142.638	211.036
	Cosine	765.320	699.980	770.134
	Correlation	534.171	364.130	503.682
0.9	Euclidean	301.343	282.237	421.030
	Manhattan	291.722	277.343	419.622
	Minkowski	307.940	288.204	421.525
	Cosine	717.519	662.511	766.076
	Correlation	458.469	328.907	486.029

Table 3.2: Average degree.

Shortest path length

Defining the length as the number of intermediate edges contained in the path between two nodes, it is possible to define the shortest path length. It is calculated by finding the shortest path between all pairs of nodes and taking the average over all these paths. Since the graphs are disconnected, this metric is not defined.

Diameter

Diameter is a similar metric to the average shortest path length, i.e. the longest of all the calculated shortest paths in a network. Therefore, this metric is not defined, or infinite, for the entire graph.

Eigenvector centrality

A natural extension of the simple degree centrality is eigenvector centrality, i.e. the assignment of a centrality point for every network neighbor a vertex has. The importance of a vertex in a network is increased by having connections to other vertices that are themselves important, thus eigenvector centrality gives each vertex a score proportional to the sum of the scores of its neighbors. A small value is therefore expected for this graphs, and this hypothesis is confirmed by the empirical results, as shown in Table 3.3.

Threshold	Distance	Prognostic profile	Functional profile	Respiratory profile
0.6	Euclidean	0.0275	0.0215	0.0288
	Manhattan	0.0259	0.0167	0.0277
	Minkowski	0.0282	0.0247	0.0289
	Cosine	0.0302	0.0286	0.0298
	Correlation	0.0284	0.0235	0.0254
0.7	Euclidean	0.0263	0.0186	0.0277
	Manhattan	0.0248	0.0147	0.0258
	Minkowski	0.0271	0.0225	0.0281
	Cosine	0.0011	0.0285	0.0298
	Correlation	0.0301	0.0225	0.0253
0.8	Euclidean	0.0244	0.0156	0.0249
	Manhattan	0.0229	0.0135	0.0248
	Minkowski	0.0252	0.0180	0.0249
	Cosine	0.0297	0.0284	0.0298
	Correlation	0.0278	0.0211	0.0253
0.9	Euclidean	0.0208	0.0116	0.0230
	Manhattan	0.0195	0.0088	0.0231
	Minkowski	0.0215	0.0128	0.0230
	Cosine	0.0290	0.0278	0.0298
	Correlation	0.0272	0.0182	0.0250

Table 3.3: Average Eigenvector centrality.

Closeness centrality

The closeness centrality measures the mean distance from a vertex to other vertices. Specifically, it is the inverse of the average shortest distance between the vertex and all other vertices in the network. The formula to compute it is given by *1/average distance to all other vertices.* The inverse is used so that a higher closeness centrality indicates a more desirable centrality score. The values obtained are shown in Table 3.4.

Betweenness centrality

The betweenness centrality measures the extent to which a vertex lies on paths between other vertices. The betweenness centrality of any vertex in a complete graph is zero, since no vertex lies in between any geodesic as every geodesic is of length 1. The betweenness centrality increases with the number of vertices in the network, so a normalized version is often considered with the centrality values scaled to between 0 and 1. For the networks constructed here, a value close to 0 is expected. Table **??** provides the values for the metric.

The Pagerank

The Pagerank is a variant of the eigenvector centrality score, that uses backlinks/in-degrees. A number equal to $1/n = 0.0009$ is obtained, where n is the number of nodes, but this result does not bring any relevant information, since the network constructed is undirected.

Threshold	Distance	Prognostic profile	Functional profile	Respiratory profile
0.6	Euclidean	0.6787	0.3634	0.7711
	Manhattan	0.5814	0.2010	0.6916
	Minkowski	0.7434	0.4921	0.7982
	Cosine	0.9852	0.8030	0.9233
	Correlation	0.7422	0.4724	0.4754
0.7	Euclidean	0.6023	0.2718	0.6858
	Manhattan	0.5430	0.1625	0.6265
	Minkowski	0.6501	0.3947	0.7128
	Cosine	0.9641	0.7971	0.9224
	Correlation	0.7042	0.4445	0.4720
0.8	Euclidean	0.5205	0.1730	0.5789
	Manhattan	0.4907	0.1005	0.5705
	Minkowski	0.5437	0.2416	0.5807
	Cosine	0.9184	0.7818	0.9186
	Correlation	0.6566	0.4155	0.4653
0.9	Euclidean	0.3858	0.0696	0.4802
	Manhattan	0.3665	0.0332	0.4794
	Minkowski	0.3953	0.0909	0.4805
	Cosine	0.8593	0.7279	0.9120
	Correlation	0.5963	0.3722	0.4445

Table 3.4: Average Closeness centrality.

Threshold	Distance	Prognostic profile	Functional profile	Respiratory profile
0.6	Euclidean	0.0005	0.0010	0.0003
	Manhattan	0.0007	0.0016	0.0004
	Minkowski	0.0003	0.0005	0.0002
	Cosine	0.0000	0.0000	0.0000
	Correlation	0.0003	0.0005	0.0001
0.7	Euclidean	0.0006	0.0013	0.0004
	Manhattan	0.0008	0.0014	0.0005
	Minkowski	0.0005	0.0008	0.0004
	Cosine	0.0000	0.0000	0.0000
	Correlation	0.0004	0.0006	0.0001
0.8	Euclidean	0.0009	0.0017	0.0007
	Manhattan	0.0010	0.0014	0.0007
	Minkowski	0.0008	0.0014	0.0006
	Cosine	0.0000	0.0000	0.0000
	Correlation	0.0005	0.0007	0.0002
0.9	Euclidean	0.0015	0.0004	0.0010
	Manhattan	0.0015	7.7715	0.0010
	Minkowski	0.0014	0.0007	0.0010
	Cosine	0.0002	0.0000	0.0000
	Correlation	0.0006	0.0009	0.0001

Table 3.5: Average Betweenness centrality.

Clustering coefficient

The clustering coefficient measures the average probability that two neighbours of a vertex are neighbours themselves. The clustering coefficient of a node is the ratio of the number of connections in the neighbourhood of a node and the number of connections if the neighbourhood is fully connected, Table 3.6. To prove that there is a significant community structure, the results are compared to the clusters provided by the clinicians. From the results reported in the Appendix A.1, it can be seen that the Slow group always presents greater values for this coefficient. This is in accordance with the expectations, as Slow progressors have a slower progression rate, meaning they survive longer. The number of patients for each time window is also higher in Slow progressors than in Fast progressors, despite the fact that is composed of approximately the same number of patients. This is because Fast progressors evolve faster and some of them arrive at the hospital already using NIV, thus they are not considered as learning instances. In addition, Respiratory features show a higher coefficient, demonstrating a greater ability to group similar patients into clusters. However, all the Profiles return good values for this coefficient, which tells how well connected the neighbourhood of a node is. If the neighbourhood is fully connected, the clustering coefficient is 1, while values close to 0 mean that there are few connections in the neighbourhood.

Threshold	Distance	Prognostic profile	Functional profile	Respiratory profile
0.6	Euclidean	0.7899	0.5970	0.8389
	Manhattan	0.6976	0.5263	0.7491
	Minkowski	0.8414	0.6959	0.8691
	Cosine	0.9872	0.8963	0.9514
	Correlation	0.8358	0.7014	0.6944
0.7	Euclidean	0.7111	0.5410	0.7606
	Manhattan	0.6554	0.4984	0.7382
	Minkowski	0.7437	0.6225	0.7968
	Cosine	0.9738	0.8922	0.9505
	Correlation	0.8008	0.6888	0.6905
0.8	Euclidean	0.6019	0.8845	0.7291
	Manhattan	0.6079	0.6815	0.6999
	Minkowski	0.5871	0.8564	0.7375
	Cosine	0.9580	0.8845	0.9508
	Correlation	0.7539	0.6815	0.6816
0.9	Euclidean	0.5467	0.3809	0.6524
	Manhattan	0.5322	0.4109	0.6453
	Minkowski	0.5230	0.4639	0.6543
	Cosine	0.9490	0.8564	0.9445
	Correlation	0.6818	0.6732	0.6508

Table 3.6: Average Clustering coefficient.

Modularity

The goal of this work is to investigate communities within networks of patients, each presenting particular characteristics that might bring new insights into the disease. Thus, one of the most relevant metrics is modularity, since it is related to how well the network can be divided into modules. This metric can be seen as the difference between the number of edges within identified communities and the random expectation. A higher value means that the network presents a modular structure, hence vertices in each community are more similar [50]. Modularity is computed for all the Profiles, by setting the values of the distance and threshold parameters to Cosine and 0.7, i.e. those that lead to the best performance. The *community.best_partition*() function is used, which compute the partition of the graph nodes that maximises the modularity, through the Louvain heuristics. Slow and Fast groups are distinguished from each other, while it is more difficult to separate the Slow and Normal groups and the Fast and Normal groups. New communities can be then identified, indicated as N-S and N-F, respectively. These results are expected since the data are not balanced, e.g the number of patients for the Slow group is 276, 562 for the Normal and 260 for the Fast. Furthermore, although Slow and Fast groups have different characteristics, it is more difficult to distinguish them from the Normal group, which is almost an average between the two. The number of communities found for the Prognostic, Functional and Respiratory profile is respectively 4, 118 and 13. The values obtained for communities containing at least 2 nodes are given in Table 3.7.

Profile	**Community**	**Slow subgraph**	**Normal subgraph**	**Fast subgraph**
Prognostic	1	0.4211	0.4689	0.1100
	2	0.0685	0.5178	0.4137
	3	0.2222	0.5778	0.200
	4	0.3113	0.5472	0.1415
Functional	1	0.4280	0.4703	0.1017
	2	0.2075	0.6321	0.1604
	3	0.0667	0.4638	0.4696
	4	0.4421	0.4895	0.0684
Respiratory	1	0.3488	0.4794	0.1717
	2	0.2513	0.5327	0.2161
	3	0.0848	0.5515	0.3636

Table 3.7: Profiles modularity.

3.2.3 Networks analysis

The features selected to construct the datasets are recognized in the literature as prognostic indicators in ALS patients. Since a respiratory target is predicted, respiratory features are expected to be more important than features concerning other aspects of the disease. In fact, from an analysis of the metrics carried out in Section 3.2.2 and visually exploring the networks obtained (Figure 3.3), it is possible to conclude that the Respiratory subgroup of features best identifies the clusters. In this study, Euclidean, Manhattan, Minkowski, Cosine and Correlation distance metrics are used to estimate distances from patients. From the values obtained and reported above, it is possible to conclude that Cosine similarity always leads to better results when compared to the other distance measures, showing, in particular, better performance for the first subgroup of features, i.e. the Prognostic profile. However, since ALS is a disease that affects the respiratory system, it is expected that the most promising values should be obtained for the subgroup of features belonging to the Respiratory Profile. A qualitative analysis of the values obtained is necessary, e.g. by calculating some disease prediction measures, such as accuracy, precision, recall, F1 score and patient clustering measure, i.e. Rand index (RI), purity, modularity and normalized mutual information (NMI), in order to evaluate the performance of all the patient similarity learning approaches on disease prediction. Figure 3.3 shows the networks obtained from the three Profiles, considering the Minkowski distance and a threshold equal to 0.7. It can be noticed that the graph related to the Respiratory profile is very distinct in the clusters formed, compared to the others.

Network validation

The clusters $C = C_1, C_2, \dots, C_m$ are compared to a potentially different partition data $P = P_1, P_2, \dots, P_s$, which represent the expert knowledge of the analyst, prior knowledge of the data in the form of class

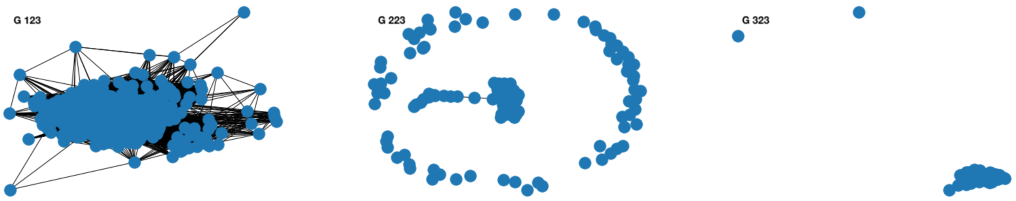

Figure 3.3: From left to right networks $G_{123}, G_{223}, G_{323}$.

labels. The clustering results are evaluated using Rand index (RI), which measures the percentage of correct decisions, can be calculated via:

$$RI = \frac{TP + TN}{TP + TN + FP + FN} = \frac{a+b}{\binom{n}{2}} \tag{3.10}$$

where a is incremented each time two nodes belonging to the same group are joined, while b is a factor incremented each time an edge is not created between nodes belonging to different groups. The higher the RI, the better the clustering result is, including values between 0 and 1, with 0 indicating that the two data clusters do not agree on any pair of points and 1 indicating that the data clustering is exactly the same. The values obtained for the different network parameters, shown in Figure 3.4, enable a qualitative evaluation of the similarity measures and thresholds, while also providing an analysis of the composition of the clusters and the Profiles that best identify them. It is therefore concluded that Cosine similarity is the most accurate measure for all profiles, with better results for the Respiratory profile, as expected, and for all threshold values selected, confirming the inferences made examining the values of the network metrics.

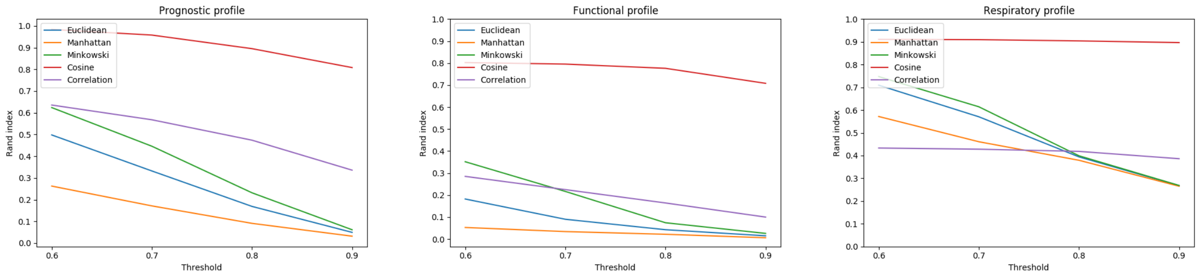

Figure 3.4: Rand index.

3.3 Summary

The most original approach introduced in this work concerns the exploration of data using the graph theory, to construct a patient network. In a Patient Similarity Network (PSN), each node represents a patient and the edge that connects them the existence of similarity between them. The implemented algorithm considers three variables to build the network, such as Feature Subgroup, Similarity Measures and Thresholds. Thus, for each Profile, two patients are considered similar if the chosen similarity measure has a value greater than a certain threshold. A total of 60 networks are constructed and analysed using different metrics.

Chapter 4

Patient stratification using Clustering

Classification systems are either supervised or unsupervised, depending on whether they assign new inputs to one of a finite number of discrete supervised classes or unsupervised categories, respectively [64–66]. In supervised classification, the mapping from a set of input data vectors to a finite set of discrete class labels is modeled in terms of some mathematical function. The aim is to minimize an empirical risk functional on a finite data set of input-output examples [64, 65, 67]. When the inducer reaches convergence or terminates, an induced classifier is generated [67]. In unsupervised classification, called clustering or exploratory data analysis, no labeled data are available [68, 69]. In cluster analysis, a group of objects is split up into several more or less homogeneous subgroups, based on a chosen measure of similarity, such that the similarity between objects within a subgroup is larger than the similarity between objects belonging to different subgroups [70]. Both the similarity and the dissimilarity should be examinable in a clear and meaningful way. Given a set of input patterns $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_j, \ldots, \mathbf{x}_n)$, where $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{id})^T \in \mathbb{R}^d$ and each measure x_{id} is said to be a feature. Partitional clustering attempts to seek a k-partition of $\mathbf{X}$, $C = (C_1, \ldots, C_k)$, with $k \leq n$, such that:

– $C_i \neq \emptyset, \quad i = 1, \ldots, k$

– $\cup_{i=1}^{k} C_i = \mathbf{X}$

– $C_i \cap C_j = \emptyset, \quad i, j = 1, \ldots, k, i \neq j$

Figure 4.1 depicts the procedure of cluster analysis with four basic steps. Ideal features should be of use in distinguishing patterns belonging to different clusters, immune to noise, easy to extract and interpret. The clustering algorithm design or selection is usually combined with the selection of a corresponding proximity measure and the construction of a criterion function. Patients are grouped according to whether they resemble each other. The proximity measure directly affects the formation of the resulting clusters. Almost all clustering algorithms are explicitly or implicitly connected to some definition of proximity measure. Once a proximity measure is chosen, the construction of a clustering criterion function makes the partition of clusters an optimization problem, well defined mathematically.

Clustering is ubiquitous, and a wealth of clustering algorithms has been developed to solve different

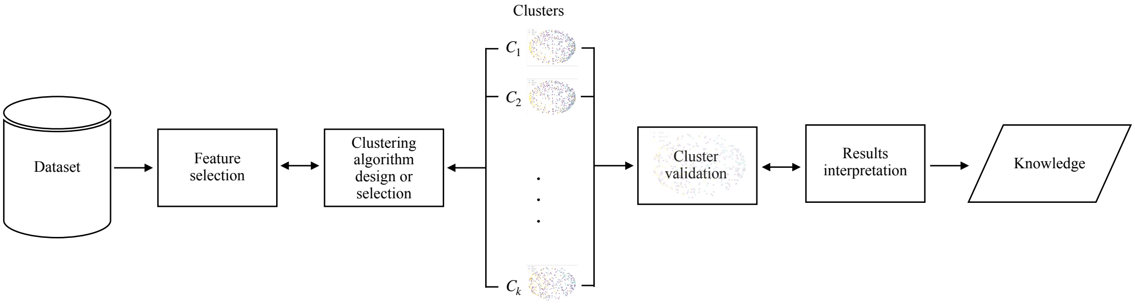

Figure 4.1: Clustering procedure.

problems in specific fields. However, no clustering algorithm can be universally used to solve all problems. Therefore, it is important to carefully investigate the characteristics of the problem treated, to select or design an appropriate clustering strategy. The quality of clustering results depends both on how the algorithms are implemented, and on their ability to find the underlying and hidden knowledge that governs the data. The traditional clustering algorithms can be divided into 9 categories which mainly contain 26 commonly used ones, summarized in Table 4.1. In Section 4.2, some of these algorithms are implemented and applied to the ALS dataset. Training clustering algorithms is a random process, thus each time different results might be obtained for the same algorithm. Therefore, the experiments below are repeated three times and also the final choices made mediate the results obtained in these three tests. However, for simplicity, below are reported only the values obtained for the first one. The source of randomness comes from the random number generator, which is generated by a deterministic process and is seeded with an initial random number. The seed is a state, storing the previous random number, which generates the sequence of random numbers.

Category	**Typical algorithm**
Clustering algorithm based on partition	K-means, K-medoids, PAM, CLARA, CLARANS
Clustering algorithm based on hierarchy	BIRCH, CURE, ROCK, Chameleon
Clustering algorithm based on fuzzy theory	FCM, FCS, MM
Clustering algorithm based on distribution	DBCLASD, GMM, BGM
Clustering algorithm based on density	DBSCAN, OPTICS, Mean-shift
Clustering algorithm based on graph theory	CLICK, MST
Clustering algorithm based on grid	STING, CLIQUE
Clustering algorithm based on fractal theory	FC
Clustering algorithm based on model	COBWEB, GMM, BGM, SOM, ART

Table 4.1: Traditional algorithms.

Given a dataset, each clustering algorithm can always generate a division, no matter whether the structure exists or not. Moreover, different approaches usually lead to different clusters and even for the same algorithm, parameter identification or the presentation order of input patterns may affect the final results. Note that the flow chart (Figure 4.1) also includes a feedback pathway cluster analysis is not a one-shot process. In many circumstances, it needs a series of trials and repetitions. Moreover, there are no universal and effective criteria to guide the selection of features and clustering schemes. Validation criteria provide some insights on the quality of clustering solutions, but also the choice of criteria to be used requires effort. Therefore, in the next sections, the analysis of the clustering algorithms is presented, the parameters are set to achieve the best performance and resemble as closely as possible the groups identified by the clinicians. Generally, there are three categories of testing criteria, i.e. external indices, internal indices, and relative indices, which are defined on three types of clustering structures, known as partitional clustering, hierarchical clustering, and individual clusters [69]. A more detailed description of these metrics is given in Section 4.1. In this chapter, both data clustering and visualization are investigated, discovering important patterns in the dataset. Data visualization has a long history behind it and can be applied to any step of the data analysis, to find ways of presenting the results of clustering, so that it is easy to understand the clustering results and draw valuable conclusions about the dataset. For each Profile, features are selected that enable good visualisation of the data in 2-dimensions. Different combinations of features are explored for each Clinical Profile, also following the correlation matrices in Section 2.6, shown in Figures 4.2. In the following analysis, clusters that visually present almost the same division of the patients, in the xy-plane, are expected. For the Prognostic Profile ALS-FRS and FVC are chosen, while for the Functional Profile ALS-FRS-R and R. For the Respiratory Profile, which groups together features different from the previous ones, FVC and PhrenMeanAmpl are selected. It can be seen that the three Profiles appear similar thanks to the presence of patients classified as Slow, generally in the top right-hand part of the graph, and those Fast in the part closest to the origin. Finally, the patients identified as Normal, stay in a range between the two, as expected.

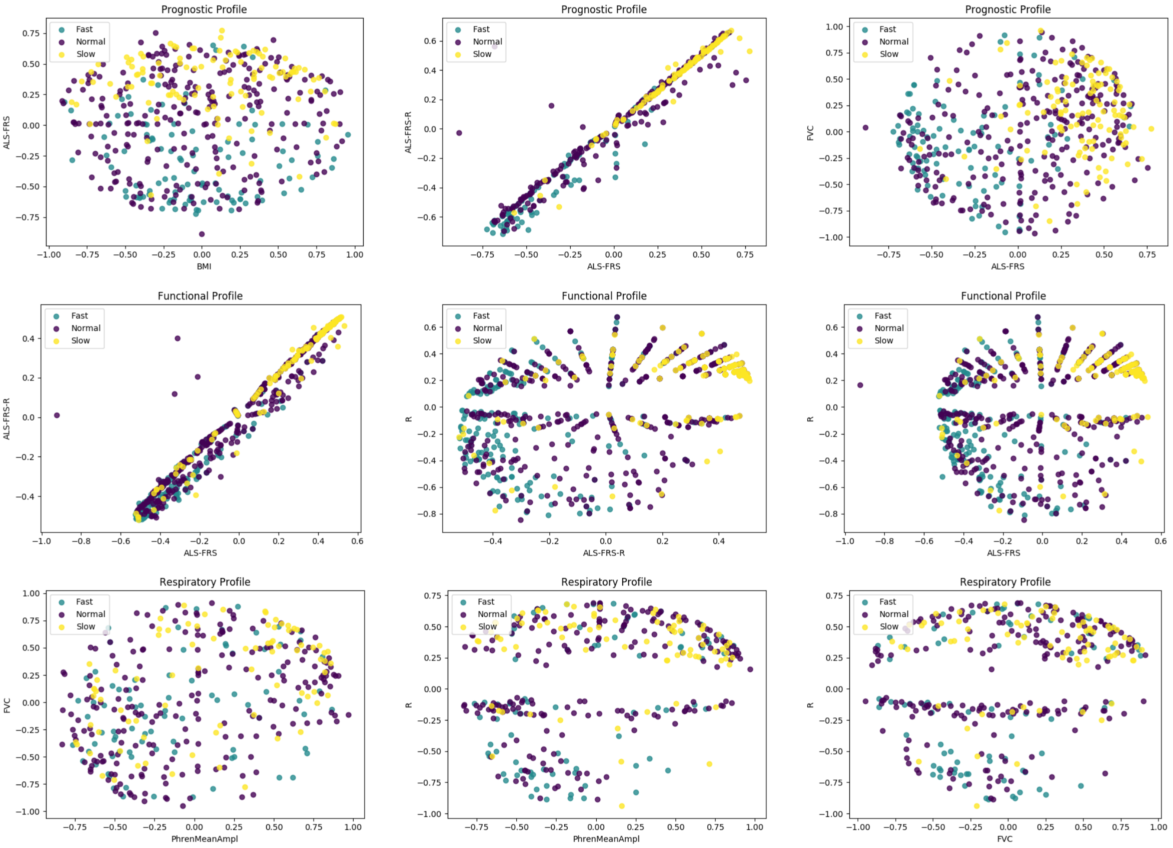

Figure 4.2: Clinical Profiles features visualization.

4.1 Clustering evaluation

Evaluating the results of a clustering algorithm is a very important part of the process of clustering data. In supervised learning, the evaluation of the resulting classification model is an integral part of the process of developing a classification model, and there are well-accepted evaluation measures and procedures [71]. In unsupervised learning, because of its very nature, cluster evaluation, also known as cluster validation, is not as well-developed. Thus, it is not easy to determine the quality of a clustering algorithm, giving rise to multiple evaluation techniques.

4.1.1 Internal validation

Internal validation methods make it possible to establish the quality of the clustering structure without having access to external information, i.e. they are based on the information provided by the data used as input to the clustering algorithm. In general, two types of internal validation metrics can be combined, which are cohesion and separation measures. Cohesion evaluates how closely the elements of the same cluster are to each other, while separation quantifies the level of separation between clusters. These measures are also known as internal indices because they are computed from the input data without any external information [71]. Internal indices are usually employed in conjunction with two clustering algorithm families, i.e. hierarchical clustering algorithms and partitional algorithms [72]. Internal validation is used when there is no additional information available. In most cases, the particular metrics used by the evaluation methods are the same metrics that the clustering algorithm tries to optimize, which can be counterproductive in determining the quality of a clustering algorithm and deliver unfair validation results. On the other hand, in the absence of other sources of information, these metrics allow different algorithms to be compared under the same evaluation criterion [73], yet care must be taken not to report biased results. Internal evaluation methods are commonly classified according to the type of clustering algorithm they are used with. For partitional algorithms, metrics based on the proximity matrix, as well as metrics of cohesion and separation, such as the Silhouette coefficient, are often used. For hierarchical algorithms, the cophenetic coefficient is the most common. In general, the internal validation value of a set of k clusters can be decomposed as the sum of the validation values for each cluster.

$$general\ validity = \sum_{i=1}^{k} w_i\ validity(C_i) \tag{4.1}$$

This measure of validity can be cohesion, separation, or some combination of both. Quite often, the weights that appear in the previous expression correspond to cluster size. The individual measures of cohesion and separation are defined as follows:

$$cohesion(C_i) = \sum_{\mathbf{x}_i \in C_i, \mathbf{x}_j \in C_i} proximity(\mathbf{x}_i, \mathbf{x}_j) \tag{4.2}$$

$$separation(C_i, C_j) = \sum_{\mathbf{x}_i \in C_i, \mathbf{x}_j \in C_j} proximity(\mathbf{x}_i, \mathbf{x}_j) \tag{4.3}$$

Both cohesion and separation are both based on a proximity function that determines how similar a pair of examples is. These metrics can also be defined for prototype-based clustering techniques, where proximity is measured from data examples to cluster centroids of medoids. It should be noted that the cohesion metric defined above is equivalent to the cluster SSE (Sum of Squared Errors), also known as SSW (Sum of Squared Errors Within Cluster), when the proximity function is the squared Euclidean distance:

$$SSE(C_i) = \sum_{\mathbf{x}_i \in C_i} d(\mathbf{c}_i, \mathbf{x})^2 = \frac{1}{2m_i} \sum_{\mathbf{x}_i \in C_i, \mathbf{x}_j \in C_j} d(\mathbf{x}_i, \mathbf{x}_j)^2 \tag{4.4}$$

where $\mathbf{x}_i$ is an example in the cluster, c_i is a cluster representative, e.g. its centroid, and m_i is the number of examples in the cluster C_i. When using the SSE metric, small values indicate a good cluster quality. This metric is minimized in those clusters that were built from SSE-optimization-based algorithms such as k-means but are suboptimal for clusters detected using other techniques, such as density-based algorithms, e.g. DBSCAN. Likewise, we can maximize the distance between clusters using a separation metric. This approach leads to the between-group sum of squares or SSB:

$$SSB = \sum_{i=1}^{k} m_i d(\mathbf{c}_i, c)^2 = \frac{1}{2k} \sum_{i,j=1}^{k} \frac{m}{K} d(\mathbf{c}_i, \mathbf{c}_j)^2 \tag{4.5}$$

where c_i is the mean of the ith cluster and c is the overall mean. Unlike the SSE metric, a good cluster quality is given by the high SSB values. As before, SSB is biased in favour of algorithms based on maximizing the separation distances among cluster centroids. As mentioned above, clustering is considered to be good when it has a high separation between clusters and high cohesion within clusters [74]. Instead of dealing with separate metrics for cohesion and separation, several metrics try to quantify the level of separation and cohesion in a single measure [75]. The Dunn index is the ratio of the smallest distance between data from different clusters and the largest distance between clusters. Again, this ratio should be maximized [76]:

$$D = \min_{1<i<k} \left\{ \min_{1<j<k, i \neq j} \left\{ \frac{\delta(C_i, C_j)}{\max_{1<l<k} \Delta C_l} \right\} \right\} \tag{4.6}$$

$$\Delta C_i = \max_{\mathbf{x}_i, \mathbf{x}_j \in c_i} \{ d(\mathbf{x}_i, \mathbf{x}_j) \} \tag{4.7}$$

$$\delta(C_i, C_j) = \min_{\mathbf{x}_i \in C_i, \mathbf{x}_j \in C_j} \{ d(\mathbf{x}_i, \mathbf{x}_j) \} \tag{4.8}$$

The Silhouette coefficient is the most common way to combine the metrics of cohesion and separation in a single measure. Computing the Silhouette coefficient at a particular point consists of the following steps. For each example, the average distance $a(i)$ to all the examples in the same cluster, is computed:

$$a(i) = \frac{1}{C_a} \sum_{j \in C_a, i \neq j} d(i, j) \tag{4.9}$$

For each example, the minimum average distance $b(i)$, between the example and the examples contained in each cluster that do not contain the analyzed example, is given by the following equation:

$$b(i) = \min_{C_b \neq C_a} \frac{1}{C_b} \sum_{j \in C_b} d(i, j) \tag{4.10}$$

For each example, the Silhouette coefficient is determined by the following expression:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \tag{4.11}$$

The Silhouette coefficient is defined in the interval $[-1, 1]$ for each example in the dataset. Unfortunately, one of the main drawbacks of the Silhouette coefficient is its high computational complexity, $O(dn^2)$, which makes it impractical when dealing with huge datasets. The global Silhouette coefficient is just the average of the particular Silhouette coefficients for each example:

$$S = \frac{1}{n} \sum_{i=1}^{n} s(i) \tag{4.12}$$

Another validation method uses the similarity matrix of a dataset and the clustering obtained by a clustering algorithm. The actual proximity matrix can be compared to an ideal version of the proximity matrix, that is based on the provided clustering. Reordering rows and columns, so that all the examples of the same cluster appear together, the ideal proximity matrix has a block diagonal structure. A high correlation between the actual and ideal proximity matrices indicates that the subjects in the same cluster are close to each other, although it might not be a good measure for density-based clusters. Unfortunately, the mere construction of the whole proximity matrix is computationally expensive, $O(n^2)$, and this validation method cannot be used without sampling for large datasets.

4.1.2 External validation

External validation methods can be associated with a supervised learning problem. External validation proceeds by incorporating additional information in the clustering validation process, i.e. external class labels for the training examples. Since unsupervised learning techniques are primarily used when such information is not available, external validation methods are not used on most clustering problems. However, since the ALS dataset contains the progression groups provided by the clinicians, the following is a description of the metrics, which are used to evaluate the performance of the clustering algorithms in Section 4.2. Like internal validation methods, it is also possible to classify external metrics depending on the algorithmic approach of the clustering technique used, to solve a particular clustering problem. Different external validation metrics can be used to compare two sets of clusters, the first one obtained by the clustering algorithm under evaluation and the second one provided by an external source. The result of a clustering algorithm $C = C_1, C_2, \ldots, C_m$ are compared to a potentially different partition data $P = P_1, P_2, \ldots, P_s$, which represent the expert knowledge of the analyst, prior knowledge of the data in the form of class labels. To carry out this analysis, a contingency matrix must be built to evaluate the clusters detected by the algorithm. This contingency matrix contains the number of data pairs found in the same cluster, both in C and in P, the number of data pairs found in the same cluster in C but in different clusters in P, the number of data pairs found in different clusters in C but in the same cluster in P, the number of data pairs found in different clusters, both in C and in P. These numbers are referred to as indicators TP, FP, FN and TN, respectively. From these four indicators, m_1 and m_2 can be obtained, which are the number of pairs found in the same cluster in C, i.e. $m_1 = TP + FP$, and the number of pairs found in the same cluster in P, i.e. $m_2 = TP + FN$. Obviously, the total number of pairs must be $M = TP + FP + FN + TN = n(n-1)/2$. Since multi-class classification is performed, by dividing patients into Slow, Normal and Fast groups, according to the speed of disease progression, the measurements for binary classification cannot be directly used. Therefore, macro-averaging [77] is used to evaluate how the algorithms work overall across the sets of data.

Matching Sets

The first family of external validation methods, that can be used to compare two partitions of data, consists of those methods that identify the relationship between each cluster detected in C and its natural correspondence to the classes in the reference result defined by P. Several measures can be defined to measure the similarity between the clusters in C, obtained by the clustering algorithm, and the clusters in P, corresponding to the prior knowledge. Precision counts the true positives, i.e. how many examples are properly classified within the same cluster:

$$Precision = \frac{TP}{TP+FP} = \frac{TP}{P} = \frac{p_{ij}}{p_i} \tag{4.13}$$

Recall evaluates the percentage of elements that are properly included in the same cluster:

$$Recall = \frac{TP}{TP+FN} = \frac{p_{ij}}{p_j} \tag{4.14}$$

Purity evaluates whether each cluster contains only examples from the same class:

$$Purity = \sum_i p_i \max \frac{p_{ij}}{p_i} \tag{4.15}$$

In the expressions above, $p_i = n_i/n$, $p_j = n_j/n$, and $p_{ij} = n_{ij}/n$, where n_{ij} is the number of examples belonging to the class i found in the cluster j and n_i (n_j) is the number of examples in the cluster i (j). The upper bound of Purity is 1, which indicates perfect match between the partitions.

Peer-to-peer Correlation

Some metrics based on measuring the correlation between pairs are the Jaccard and the Rand coefficients. The Jaccard coefficient assesses the similarity of the detected clusters C to the provided partition P:

$$J = \frac{TP}{TP+FP+FN} = \frac{\sum_{ij}\binom{n_{ij}}{2}}{\sum_i \binom{n_i}{2} + \sum_j \binom{n_j}{2} - \sum_{ij}\binom{n_{ij}}{2}} \tag{4.16}$$

The Rand coefficient is similar to the Jaccard coefficient, yet it is measured against the total dataset, thus it is equivalent to accuracy in a supervised machine learning setting:

$$RI = \frac{TP + TN}{M} = \frac{\binom{n}{2} - \sum_i \binom{n_i}{2} + \sum_j \binom{n_j}{2} - \sum_{ij} \binom{n_{ij}}{2}}{\binom{n}{2}} \tag{4.17}$$

The adjusted Rand index is the corrected-for-chance version of the Rand index. Such a correction for chance establishes a baseline by using the expected similarity of all pair-wise comparisons between clusterings, specified by a random model. Though the Rand Index may only yield a value between 0 and 1, the adjusted Rand index can yield negative values, if the index is less than the expected index.

$$\begin{aligned} ARI &= \frac{RI - Expected(RI)}{\max(RI) - Expected(RI)} \\ &= \frac{\sum_{ij} \binom{n_{ij}}{2} - [\sum_i \binom{n_i}{2} + \sum_j \binom{n_j}{2}]/\binom{n}{2}}{\frac{1}{2}[\sum_i \binom{n_i}{2} + \sum_j \binom{n_j}{2}] - [\sum_i \binom{n_i}{2} + \sum_j \binom{n_j}{2}]/\binom{n}{2}} \end{aligned} \tag{4.18}$$

Measures Based on Information Theory

A third family of external cluster validation metrics is based on Information Theory concepts, such as the existing uncertainty in the prediction of the natural classes provided by the partition P. This family includes basic measures such as entropy and mutual information, as well as their respective normalized variants. Entropy is a reciprocal measure of purity that allows us to measure the degree of disorder in the clustering results:

$$H = -\sum_i p_i \left(\sum_j \frac{p_{ij}}{p_i} \log \frac{p_{ij}}{p_i} \right) \tag{4.19}$$

Mutual information measures the reduction in uncertainty about the clustering results, given the prior partition:

$$MI = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{p_i p_j} \tag{4.20}$$

Again, $p_i = n_i/n$, $p_j = n_j/n$, and $p_{ij} = n_{ij}/n$. The mutual information has many possible upper bounds, that might be used to obtain the Normalized Mutual Information (NMI).

$$\begin{aligned} MI(C,P) &\leq \min\{H(C), H(P)\} \leq \sqrt{H(C)H(P)} \leq \frac{1}{2}(H(C) + H(P)) \\ &\leq \max\{H(C), H(P)\} \leq H(C,P) \end{aligned} \tag{4.21}$$

The value of NMI also can vary between 0 and 1 and achieves its maximum value when grouping clusterings are the same as the real cohorts. Unlike the purity, but like the Rand Index, it is symmetric. The approach used in [78], consists in dividing the mutual information by the arithmetic mean of the entropies:

$$NMI(C,P) = \frac{2I(C,P)}{H(C) + H(P)} = \frac{-2\sum_{ij} p_{ij} \log(p_i/p_i p_j)}{\sum_i \log p_i + \sum_j \log p_j} \tag{4.22}$$

4.2 Clustering Algorithms

4.2.1 K-means

K-means is a well-known partitioning method, which classifies objects into k groups, with k chosen a priori. Cluster membership is determined by computing the centroid for each group, i.e. the multidimensional version of the mean, and assigning each object to the group with the closest centroid. This approach minimizes the overall within-cluster dispersion by iterative reallocation of cluster members. In a general sense, a k-partitioning algorithm takes as input a set C of objects and an integer k, and outputs a partition of C into subsets $C_1, C_2, \ldots, C_k$. It uses the sum of squares as the optimization criterion. Let $\mathbf{x}_r^i$ be the r^{th} element of C_i, $|C_i|$ be the number of elements in C_i, and $d(\mathbf{x}_r^i, \mathbf{x}_s^i)$ be the distance between $\mathbf{x}_r^i$ and $\mathbf{x}_s^i$. The sum of squares criterion is defined by the cost function:

$$c(C_i) = \sum_{r=1}^{|C_i|} \sum_{s=1}^{|C_i|} (d(\mathbf{x}_r^i, \mathbf{x}_s^i))^2 \tag{4.23}$$

In particular, as mentioned above, k-means works by calculating the centroid of each cluster C_i, denoted x^{-i}, and optimizing the cost function 4.24. The goal of the algorithm is to minimize the total cost $c(C_i) + ... + c(C_k)$.

$$c(C_i) = \sum_{r=1}^{|C_i|} (d(\mathbf{x}^{-i}, \mathbf{x}_r^i))^2 \tag{4.24}$$

The time complexity is $O(nkl)$, where n is the number of patterns, k is the number of clusters, and l is the number of iterations taken by the algorithm to converge. Its space complexity is $O(k + n)$ and it requires additional space to store the data matrix. Moreover, it is order-independent, thus for a given initial seed set of cluster centres, it generates the same partition of the data irrespective of the order in which the patterns are presented to the algorithm. K-means requires k as an input, which in this case is set to 3, following the number of groups identified by the clinicians. However, two metrics such as the Elbow method and the Silhouette analysis are used, to confirm the number k of patient groups and to identify which subgroup of features best identifies this number. The results obtained are shown in Figures 4.3 and 4.4. The Elbow method selects the number of clusters k based on the sum of squared distance (SSE) between data points and their assigned clusters' centroids. The spot where SSE starts to flatten out and forming an elbow is chosen as the k value. Silhouette analysis can be used to study the separation distance between the resulting clusters. The Silhouette plot displays a measure of how close each point in

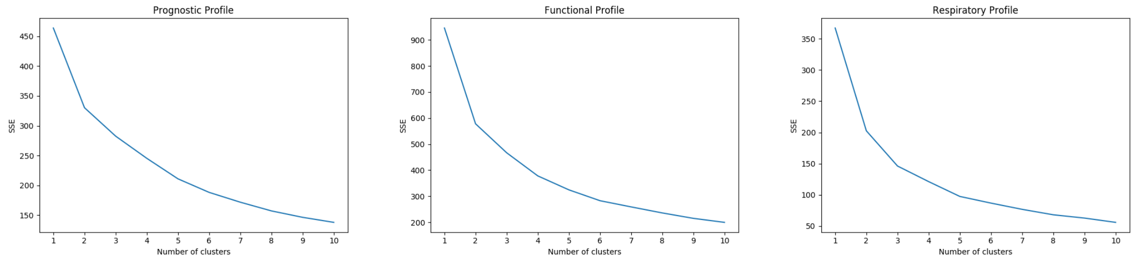

Figure 4.3: Sum of squared distance (K-means Clustering Model).

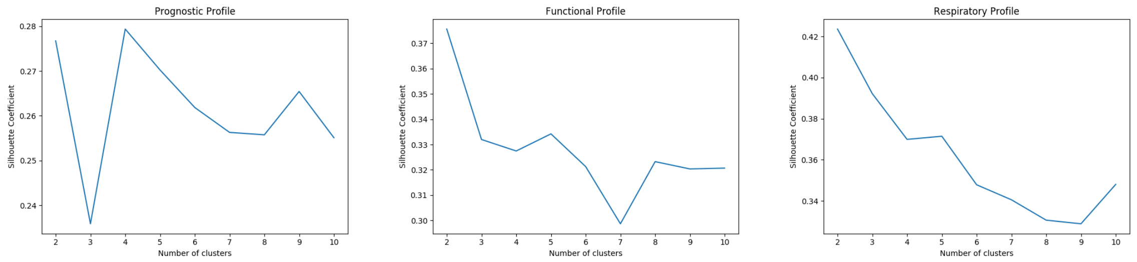

Figure 4.4: Silhouette (K-means Clustering Model).

one cluster is to points in the neighbouring clusters, thus providing a way to assess parameters, like the number of clusters, visually. This measure has a range of $[-1, 1]$. Silhouette coefficients near 1 indicate that the cluster is dense and well-separated from other clusters. A value of 0 indicates that the sample is on or very close to the decision boundary between two neighbouring clusters and negative values indicate that those samples might have been assigned to the wrong cluster. Table 4.2 also shows the obtained values of the metrics used to evaluate the clusters for the three Profiles. Finally, Figure 4.5 shows the clusters identified by the algorithm. A comparison with the clusters identified by the clinicians in Figure 3.2, reveals some differences in terms of group composition, which is also underlined by the low NMI values obtained. Figure 4.6 shows the graphs representing the dataset divided into five clusters.

Profile	**ARI**	**NMI**	**Silhouette**	**SSE**
Prognostic Profile	0.0601	0.0994	0.2359	282.8184
Functional Profile	0.1019	0.1401	0.3323	466.8319
Respiratory Profile	0.0214	0.0409	0.3922	146.2140

Table 4.2: K-means performance for the Clinical Profiles.

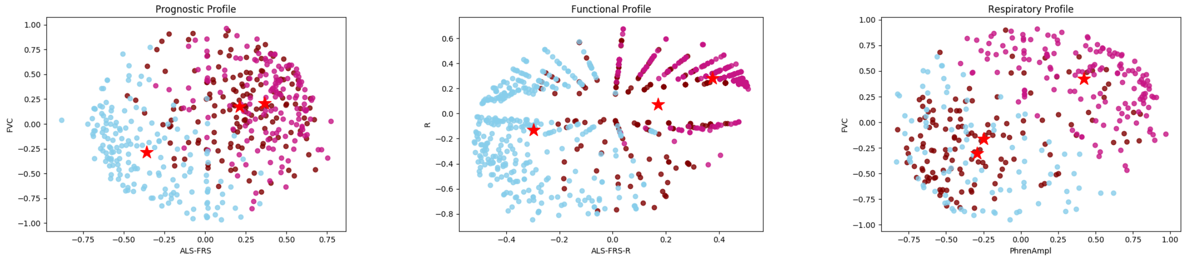

Figure 4.5: 3 Clusters of Patients (K-means Clustering Model)

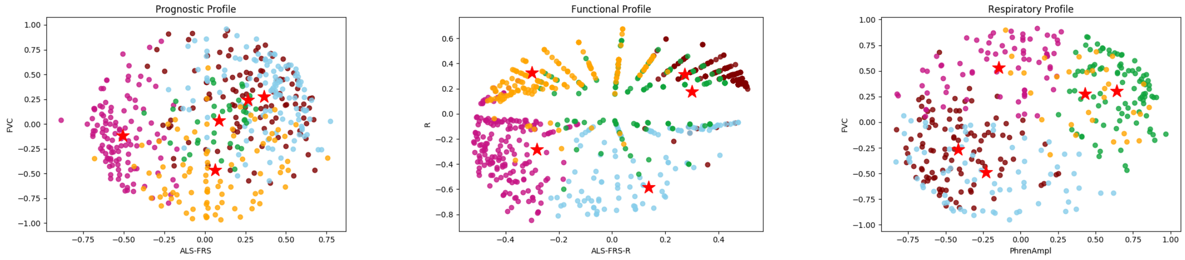

Figure 4.6: 5 Clusters of Patients (K-means Clustering Model)

4.2.2 K-medoids

K-means clustering iteratively finds the k centroids and assigns every object to the nearest centroid. Unfortunately, k-means clustering is known to be sensitive to the outliers although it is quite efficient in terms of the computational time. In k-medoids clustering representative objects, called medoids, are considered instead of centroids. K-medoids is a clustering algorithm related to the k-means and the medoidshift algorithms. Both the k-means and k-medoids algorithms are partitioning techniques of clustering that clusters a dataset of n objects into k clusters, with k known a priori. However, while k-means attempts to minimize the total squared error, k-medoids minimizes the sum of dissimilarities between points labeled to be in a cluster and a point designated as the centre of that cluster. A medoid can be defined as the point in the cluster, whose dissimilarities with all the other points in the cluster is minimum. Because it is based on the most centrally located object in a cluster, it is less sensitive to outliers in comparison with the k-means clustering. The samples are grouped into k ($k < n$) clusters, where n is the number of patterns and k is the number of clusters assumed to be given and set to 3, following the number of groups identified by the clinicians. The time complexity is $O(k(n-k)^2)$, which makes it much slower than the k-means algorithm. Again, the Elbow method and the Silhouette analysis are used. The results obtained are shown below in Figures 4.7 and 4.8. The SSE values obtained are not very informative, whereas the Silhouette score shows for the Prognostic Profile a higher peak at some clusters other than three, thus different from what was expected. Tables 4.3 and 4.4 show the ARI and NMI values, obtained by testing different parameters, such as the distance metrics reported in Section 3.1.3, and the algorithm used to compute the nearest neighbours. Possible values of the `method` parameter are *alternate*, which is faster and *pam*, which is more accurate and leads to better performance for all the Profiles, as expected. Finally, Figure 4.9 shows the clusters identified by the *pam* algorithm, using Euclidean distance. The results are similar to the ones identified by the clinicians (Figure 3.2), even

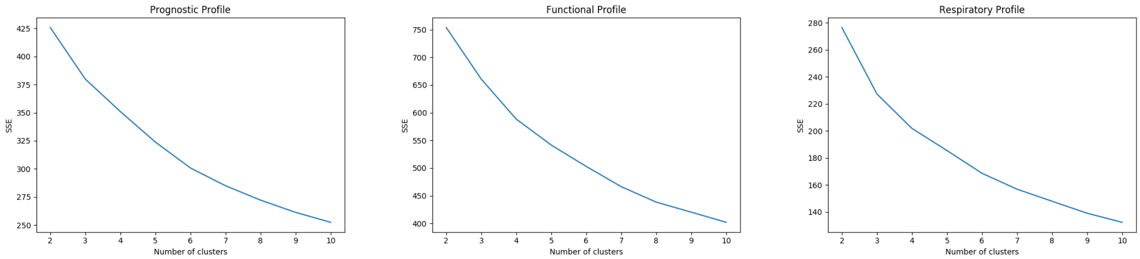

Figure 4.7: Sum of squared distance (K-medoids Clustering Model).

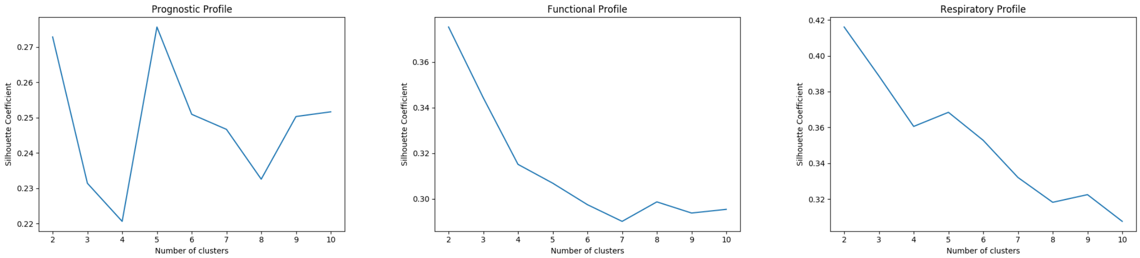

Figure 4.8: Silhouette (K-medoids Clustering Model).

though the ARI and NMI values obtained are not very high. Figure 4.10 shows the graphs representing the dataset divided into five clusters. No profile shows a substantial visual improvement in the composition of the clusters, although the Prognostic Profile shows a peak value of the Silhouette in 5, suggesting a possible further subdivision of patients.

Method	Distance	Prognostic profile	Functional profile	Respiratory profile
Alternate	Euclidean	0.0514	0.0759	0.0157
	Manhattan	0.0471	0.0865	0.0174
	Minkowski	0.0453	0.0916	0.0141
	Cosine	0.0444	0.0821	0.0123
	Correlation	0.0393	0.0282	0.0032
Pam	Euclidean	0.0509	0.0606	0.0113
	Manhattan	0.0479	0.0864	0.0109
	Minkowski	0.0509	0.0775	0.0113
	Cosine	0.0396	0.0783	0.0109
	Correlation	0.0315	0.0345	0.0007

Table 4.3: K-medoids Adjusted Rand Index (ARI).

Method	Distance	Prognostic profile	Functional profile	Respiratory profile
Alternate	Euclidean	0.0989	0.1179	0.0299
	Manhattan	0.0893	0.1315	0.0440
	Minkowski	0.0795	0.1295	0.0367
	Cosine	0.0857	0.1242	0.0405
	Correlation	0.0814	0.0282	0.0071
Pam	Euclidean	0.0925	0.1052	0.0345
	Manhattan	0.0959	0.1296	0.0353
	Minkowski	0.0925	0.1192	0.0345
	Cosine	0.0763	0.1191	0.0341
	Correlation	0.0610	0.0556	0.0078

Table 4.4: K-medoids Normalized Mutual Information (NMI).

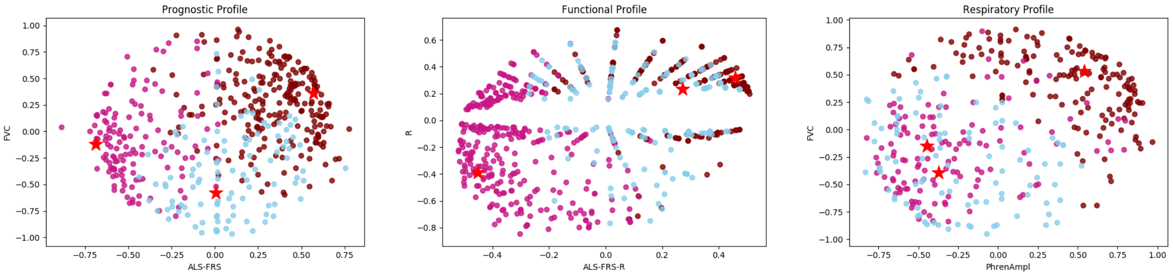

Figure 4.9: 3 Clusters of Patients (K-medoids Clustering Model)

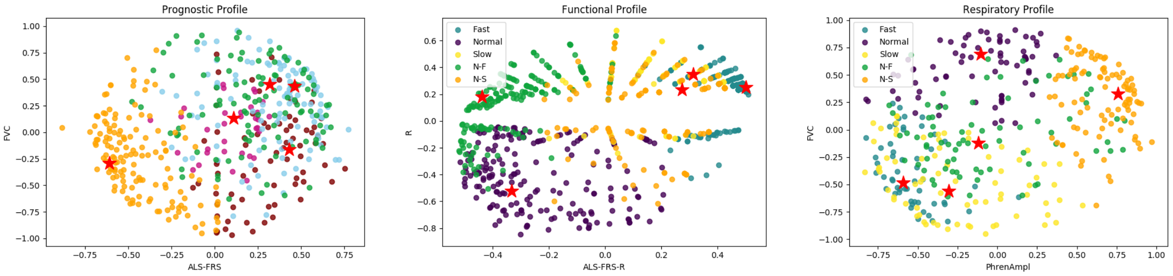

Figure 4.10: 5 Clusters of Patients (K-medoids Clustering Model)

4.2.3 DBSCAN

Density-Based Spatial Clustering of Applications with Noise is a non-parametric, density-based clustering technique. Compared to non-density based clustering methods, the DBSCAN algorithm has unique and advanced features that are useful when detecting a class of different shapes and sizes. In addition to the possibility of detecting clusters of arbitrary shapes, DBSCAN is relatively fast when clustering small and medium datasets and is robust to outliers. The space complexity is $O(n)$, while the complexity of DBSCAN is at least $O(nlog(n))$. The most time-consuming step of clustering is the calculation of the similarity measure between data objects, while the clustering itself requires only a single scan through the data objects. K-Means clustering may cluster loosely related observations together. Every observation becomes a part of some cluster eventually, even if the observations are scattered far away in the vector space. Since clusters depend on the mean value of cluster elements, each data point plays a role in forming the clusters. A slight change in data points might affect the clustering outcome. This problem is greatly reduced in DBSCAN, due to the way clusters are formed. Unlike k-Means, where the number of centroids needs to be specified, DBSCAN requires only two parameters, i.e. epsilon, that is the radius of the circle to be created around each data point to check the density, and the minimum number of data points required inside that circle for that data point to be classified as a core point. The key idea of DBSCAN is that for each object of a cluster, the neighbourhood of a given radius ϵ has to contain at least a minimum number of objects, which means that the cardinality of the neighbourhood has to exceed some threshold. The ϵ-neighborhood of an arbitrary point $\mathbf{x}_i$ is defined as:

$$N_\epsilon = \{\mathbf{x}_i \in \mathbf{X}/dist(\mathbf{x}_i, \mathbf{x}_j) < \epsilon\} \tag{4.25}$$

where $\mathbf{X}$ is the database of objects. If the ϵ–neighbourhoods of a point $\mathbf{x}_i$ at least contain a minimal number of points, it is called core point. A point is a core point if $N_\epsilon(\mathbf{x}_i) > min\_samples$, where $min\_samples$ is the minimum number of points in the ϵ-neighborhood of a core point. Epsilon values equal to 0.5, 0.6 and 0.7 are analysed, while the min_samples parameter is varied in a range from 5 to 7. Tables 4.5 and 4.6 show the ARI and NMI values for the different distances obtained for an epsilon equal to 0.7, which yields better performance. Figure 4.11 shows the graphs representing the Profiles clusters obtained by k-means and DBSCAN, which does not properly identify the clusters, thus suggesting a standard form of the clusters. For all the Clinical Profiles, DBSCAN clusters patients into almost a single type, which corresponds to the Normal group. This may be due either to data imbalance or to the inability of the algorithm to recognise the particular characteristics that make a patient Slow or Fast, confirmed by the accuracy values obtained.

4.2.4 GMM

Unlike similarity-based clustering, which generates hard partition of data, model-based clustering can generate soft partition which is sometimes more flexible. Model-based methods use mixture distributions to fit the data and the conditional probabilities of data points are naturally used to assign probabilistic labels. Gaussian Mixture Model is one of the most widely used mixture models for clustering [79]. Each Gaussian

Samples	Distance	Prognostic profile	Functional profile	Respiratory profile
5	Euclidean	0.0079	-0.0008	0.0000
	Manhattan	0.0626	-0.0129	0.0164
	Minkowski	0.0626	-0.0129	0.0164
	Cosine	0.0000	0.0000	0.0000
	Correlation	0.0000	0.0000	0.0000
6	Euclidean	0.0079	-0.0017	0.0036
	Manhattan	0.0645	-0.0183	0.0197
	Minkowski	0.0645	-0.0183	0.0197
	Cosine	0.0000	0.0000	0.0000
	Correlation	0.0000	0.0000	0.0000
7	Euclidean	0.0079	-0.0017	0.0036
	Manhattan	0.0671	-0.0210	0.0141
	Minkowski	0.0671	-0.0210	0.0141
	Cosine	0.0000	0.0000	0.0000
	Correlation	0.0000	0.0000	0.0000

Table 4.5: DBSCAN Adjusted Rand Index (ARI).

Samples	Distance	Prognostic profile	Functional profile	Respiratory profile
5	Euclidean	0.0133	0.0013	0.0000
	Manhattan	0.1129	0.0139	0.0174
	Minkowski	0.1129	0.0139	0.0174
	Cosine	0.0000	0.0000	0.0000
	Correlation	0.0000	0.0000	0.0000
6	Euclidean	0.0133	0.0026	0.0082
	Manhattan	0.1167	0.0140	0.0050
	Minkowski	0.1167	0.0140	0.0050
	Cosine	0.0000	0.0000	0.0000
	Correlation	0.0000	0.0000	0.0000
7	Euclidean	0.0133	0.0026	0.0082
	Manhattan	0.1200	0.0171	0.0026
	Minkowski	0.1200	0.0171	0.0026
	Cosine	0.0000	0.0000	0.0000
	Correlation	0.0000	0.0000	0.0000

Table 4.6: DBSCAN Normalized Mutual Information (NMI).

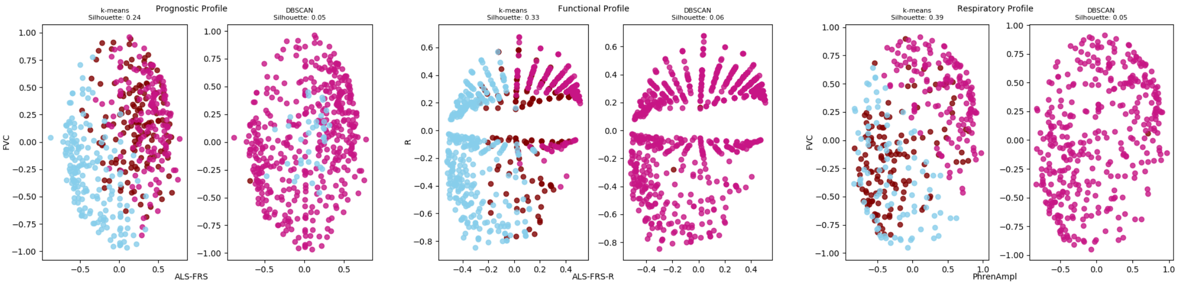

Figure 4.11: Comparison between DBSCAN and K-means.

density is called a component of the mixture and has its own mean and covariance. In many applications, their parameters are determined by maximising likelihood, typically using the Expectation-Maximization algorithm [80]. From a model-based perspective, each cluster can be mathematically represented by a parametric distribution. Therefore, the entire dataset is modeled by a mixture of these distributions. The most widely used model in practice is the mixture of Gaussians:

$$P(\mathbf{x}|\Theta) = \sum_{i=1}^{k} \alpha_i p_i(\mathbf{x}|\theta_i) \tag{4.26}$$

where the parameters are $\Theta = (\alpha_1, \ldots, \alpha_k, \theta_1, \ldots, \theta_k)$ such that $\sum_{i=1}^{k} \alpha_i = 1$ and each p_i is a Gaussian density function parameterized by θ_i. In other words, k component densities are mixed together with k mixing coefficients α_i. Let $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ be a set of data points. The goal is to find Θ such that $p(\mathbf{X}|\Theta)$ is a maximum. This is known as the Maximum Likelihood (ML) estimate for Θ. In order to estimate Θ, it is typical to introduce the logarithmic likelihood function, defined as follows:

$$\begin{aligned} \mathcal{L}(\Theta) &= \log P(\mathbf{X}|\Theta) = \log \prod_{i=1}^{n} P(\mathbf{x}_i|\Theta) \\ &= \sum_{i=1}^{n} \log \left(\sum_{j=1}^{k} \alpha_j p_j(\mathbf{x}_i|\theta_j) \right) \end{aligned} \tag{4.27}$$

which is difficult to optimize because it contains the logarithm of the sum. If we know the value of $\mathbf{y} = (y_1, \ldots, y_n)$, the likelihood becomes:

$$\begin{aligned} \mathcal{L}(\Theta) &= \log P(\mathbf{X}, \mathbf{y}|\Theta) = \log \prod_{i=1}^{n} P(\mathbf{x}_i, y_i|\Theta) \\ &= \sum_{i=1}^{n} \log P(\mathbf{x}_i|y_i) P(y_i) = \sum_{i=1}^{n} \log(\alpha_{y_i} p_{y_i}(\mathbf{x}_i|\theta_{y_i})) \end{aligned} \tag{4.28}$$

which can be optimized using a variety of techniques, such as the Expectation-Maximization algorithm. The time complexity is $O(nk^3)$, where n is the number of iterations and k is the number of parameters. Comparison of the k-means algorithm with the EM algorithm for Gaussian mixtures shows that there is a close similarity [79]. Whereas the k-means algorithm performs a hard assignment of data points to clusters, in which each data point is associated uniquely with one cluster, the EM algorithm makes a soft assignment based on the posterior probabilities. The fact that GMM is a generative model gives us a natural means of determining the optimal number of components for a given dataset. A generative model is inherently a probability distribution for the dataset, and so the likelihood of the data can be simply evaluated under the model, using cross-validation to avoid over-fitting. Another means of correcting for over-fitting is to adjust the model likelihoods using some analytic criteria, such as the Akaike information criterion (AIC) or the Bayesian information criterion (BIC). The best known of the information criteria used for determining the number of components is Akaike's Information Criterion (AIC). The AIC is calculated for mixtures as:

$$AIC = -2 \log \mathcal{L}(\Theta) + 2K \tag{4.29}$$

where K is the number of free parameters in the mixture. The theoretical justification for AIC is that choosing the minimum value of the AIC asymptotically minimizes the mean Kullback-Leibler information for discrimination between the proposed distribution and the true distribution, i.e. the model with the minimum value of the AIC should be asymptotically closest in Kullback-Leibler distance to the true model. However, several studies have found that the AIC overestimates the number of components for mixtures, most likely due to violations of the regularity conditions required for the approximation to hold. Compared to BIC, the AIC penalizes models with larger numbers of parameters less, leading to the choice of more mixture components. AIC also has a Bayesian interpretation, leading to the MAP estimate in regular models, when the amount of the information in the prior is of the same order as the amount of information in the data. This is a highly informative prior, and will not be plausible in most cases, so the Bayesian interpretation of AIC is questionable in many situations. The BIC provides a widely used approximation to the integrated likelihood for regular models. It is defined as:

$$BIC = 2\mathcal{L}(\Theta) + K \log n \tag{4.30}$$

where K is the number of free parameters in the mixture model. For regular models, BIC is derived as an approximation to twice the logarithm integrated likelihood using the Laplace method, but the necessary regularity conditions do not hold for mixture models in general. However, it is known from the literature that BIC leads to a consistent estimator of the mixture density and is consistent for choosing the number of components in a mixture model. Figure 4.12 shows the AIC and BIC as a function of the number of GMM components for the dataset. The optimal number of clusters is the value that minimizes the AIC or BIC, so clearly three for the Prognostic Profile, while for the other profiles the choice seems uncertain, particularly for the Respiratory Profile, where the minimum point is four. AIC and BIC also show a similar trend for all the Profiles, even if usually the BIC recommends a simpler model.

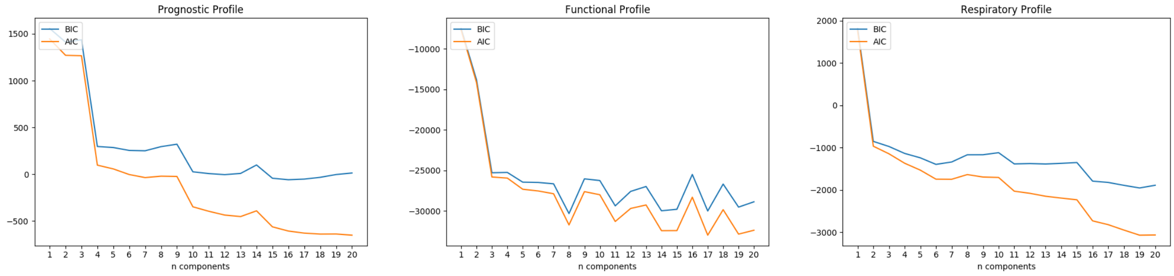

Figure 4.12: AIC and BIC values (GMM Clustering Model).

Tables 4.7 and 4.8 show the ARI and NMI values, for the three Profiles, obtained by exploring the parameters of the function. The covariance_type option controls the degrees of freedom in the shape of each cluster. The default is *diag*, which means that the size of the cluster along each dimension can be set independently, with the resulting ellipse constrained to align with the axes. A slightly simpler and faster model is *spherical*, which constrains the shape of the cluster such that all dimensions are equal. The resulting clustering will have similar characteristics to that of k-means. A more complex and computationally expensive model uses *full*, which allows each cluster to be modeled as an ellipse with arbitrary orientation. The method used to initialize the weights, the means and the precisions, is controlled by the init_params hyperparameter. It must be one of *kmeans* or *random*, i.e. responsibilities are initialized using k-means or randomly respectively. A Gaussian mixture model attempts to find a mixture of multi-dimensional Gaussian probability distributions that best model any input dataset. This is done using the predict_proba method, which returns a matrix of size [n_samples, n_clusters], which measures the probability that any point belongs to the given cluster.

Init_params	**Covariance**	**Prognostic profile**	**Functional profile**	**Respiratory profile**
kmeans	full	0.0234	0.0310	0.0172
	tied	0.0738	0.0681	0.0206
	diag	0.0781	0.1317	0.0078
	spherical	0.0468	0.0843	0.0215
random	full	0.0234	0.0037	0.0098
	tied	0.0339	0.0060	0.0184
	diag	0.0922	0.1317	0.0045
	spherical	0.0455	0.1197	0.0215

Table 4.7: GMM Adjusted Rand Index (ARI).

Init_params	**Covariance**	**Prognostic profile**	**Functional profile**	**Respiratory profile**
kmeans	full	0.0350	0.0567	0.0436
	tied	0.1325	0.0946	0.0422
	diag	0.1314	0.1571	0.0530
	spherical	0.0936	0.1197	0.0370
random	full	0.0350	0.0052	0.0535
	tied	0.0556	0.0291	0.0331
	diag	0.1294	0.1571	0.0421
	spherical	0.0966	0.1431	0.0371

Table 4.8: GMM Normalized Mutual Information (NMI).

This uncertainty can be visualized by making the size of each point proportional to the certainty of its prediction, as in Figures 4.13 and 4.14.

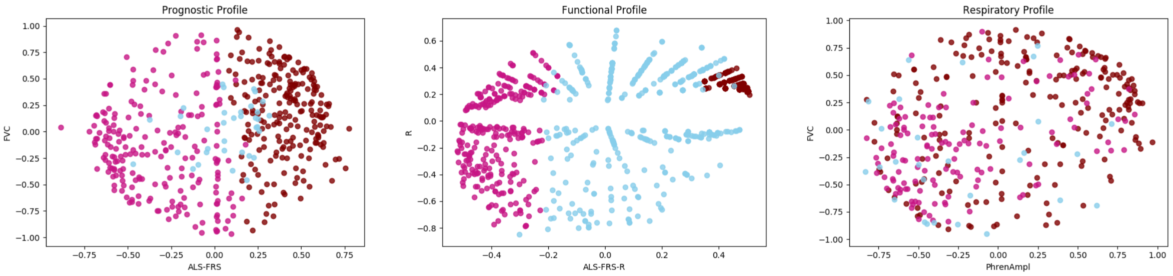

Figure 4.13: 3 Clusters of Patients (GMM Clustering Model).

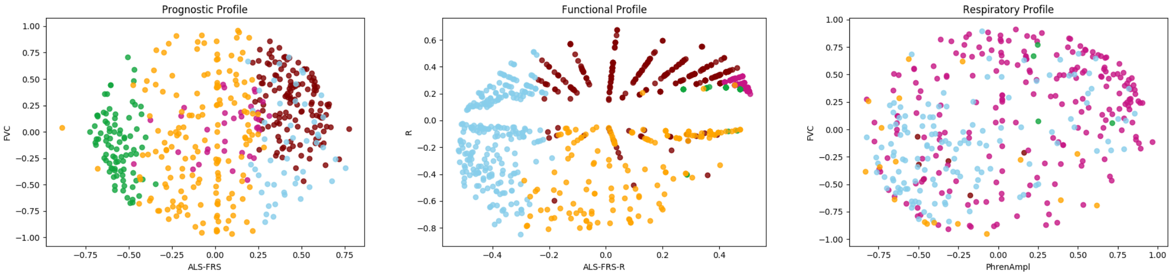

Figure 4.14: 5 Clusters of Patients (GMM Clustering Model).

4.2.5 BGM

A Gaussian Mixture Model assumes the data to be segregated into clusters in such a way that each data point in a given cluster follows a particular multi-variate Gaussian distribution, independent of the others. To cluster data in such a model, the posterior probability of a data point needs to be calculated. An approximate method for this purpose is the Variational Bayesian Inference method, which incorporates the prior structure of the Gaussian mixture model with almost no penalty for the reasoning time. Variational reasoning is an extension of Expectation-Maximization that maximizes the lower bound of model evidence, including a priori, rather than the data likelihood function. The principle is the same, but the variational approach integrates prior distribution information to increase the regularization limit. This avoids the singularity, that is often expected to maximize the solution, but it also introduces slight deviations from the model. The variational method calculation process is usually significantly slower, but it is usually not slow enough to be used. The primary two parameters of the Bayesian Gaussian Mixture Class are n_components and covariance_type, which determine the maximum number of clusters in the given data and the type of covariance parameters to be used, respectively. In the below-given steps, the parameter n_components is fixed at 3, while the parameter covariance_type varies, to explore the impact of this parameter on the clustering. Due to its Bayesian nature, the variational algorithm requires more hyperparameters than the EM, the most important of which is the concentration parameter weight_concentration_prior. The concentration prior gives most of the components in the hybrid model a certain weight. The parameter implementation of the Bayesian Gaussian Mixture class proposes two weight distribution priors, i.e. the Dirichlet distribution finite mixture model and the Dirichlet Process infinite mixture model. In practical applications, the Dirichlet Process inference algorithm uses a truncated approximated distribution and a fixed maximum component number. It can be found that the value of the weight_concentration_prior parameter has not a great influence on the number of valid activation components obtained, however when the prior is of the type *dirichlet_distribution*, higher values in terms of performance are achieved. Moreover, there is no substantial difference in terms of the weights of each mixture component, which is almost identical for both the prior types, with generally the 60%, 30% and 10% of the patients being randomly allocated to the three classes of patients. Tables 4.9 and 4.10 show the ARI and NMI values, obtained by comparing different types of weight concentration prior, by keeping parameters covariance_type and init_params fixed to *full* and *kmeans* respectively, which led to the best performance for the GMM algorithm. Figures 4.15 and 4.16 shows the clusters obtained considering the default prior *dirichlet_process*, and wights equal to 0.33, i.e. $1/n_{components}$. Again, there are options to limit the different types of estimated covariance, as in the GMM algorithm.

4.2.6 Hierarchical Clustering

Partitioning algorithms are based on specifying an initial number of groups and iteratively reallocating objects among groups to convergence. In contrast, hierarchical algorithms combine or divide existing groups, creating a hierarchical structure that reflects the order in which groups are merged or divided. In an agglomerative method, which builds the hierarchy by merging, the objects initially belong to a list of

Prior type	**Weights**	**Prognostic profile**	**Functional profile**	**Respiratory profile**
dirichlet process	0.01	0.0234	0.0569	0.0172
	0.10	0.0234	0.0294	0.0172
	0.33	0.0234	0.0294	0.0172
	0.50	0.0234	0.0682	0.0172
	1.00	0.0234	0.0294	0.0172
dirichlet distribution	0.01	0.0234	0.0310	0.0172
	0.10	0.0234	0.0310	0.0172
	0.33	0.0234	0.0294	0.0172
	0.50	0.0234	0.0294	0.0063
	1.00	0.0234	0.0294	0.0172

Table 4.9: BGM Adjusted Rand Index (ARI).

Prior type	**Weights**	**Prognostic profile**	**Functional profile**	**Respiratory profile**
dirichlet process	0.01	0.0350	0.0675	0.0436
	0.10	0.0350	0.0567	0.0436
	0.33	0.0350	0.0567	0.0436
	0.50	0.0350	0.0758	0.0436
	1.00	0.0350	0.0567	0.0436
dirichlet distribution	0.01	0.0350	0.0567	0.0436
	0.10	0.0350	0.0567	0.0436
	0.33	0.0350	0.0567	0.0436
	0.50	0.0350	0.0567	0.0410
	1.00	0.0350	0.0567	0.0436

Table 4.10: BGM Normalized Mutual Information (NMI).

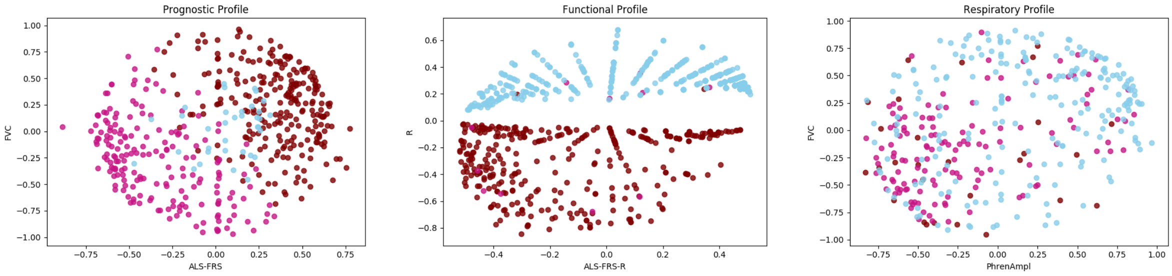

Figure 4.15: 3 Clusters of Patients (BGM Clustering Model).

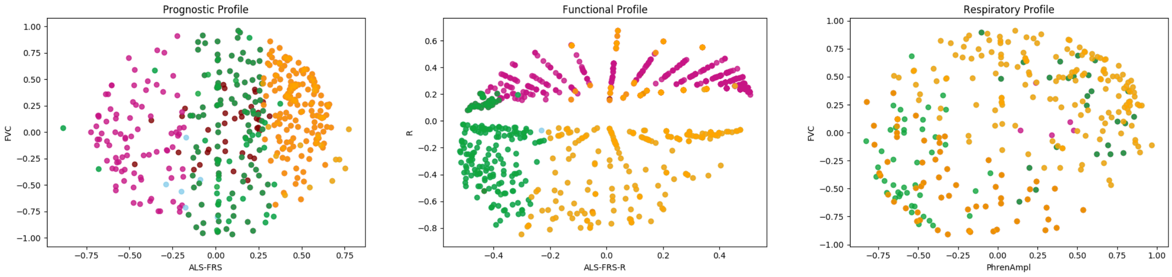

Figure 4.16: 5 Clusters of Patients (BGM Clustering Model).

singleton sets $C_1, C_2, \ldots, C_n$. Then a cost function is used to find the pair of sets C_i, C_j from the list to merge. Once merged, C_i and C_j are removed from the list of sets and replaced with $C_i \cup C_j$. The value of their similarity is retained and used to draw the typical result of these techniques, the dendrogram. This process iterates until all objects are in a single group. Different variants of agglomerative hierarchical clustering algorithms may use different cost functions. Complete linkage, average linkage, and single linkage methods use maximum, average, and minimum distances between the members of two clusters, respectively. With n objects, the final cluster is obtained after $(n-1)$ steps. The hierarchy is a consequence of the fact that larger clusters are always obtained by the merger of smaller ones. The space required for the Hierarchical clustering Technique is very high when the number of data points is high, as the similarity matrix is stored in the RAM. The space complexity is the order of the square of n, i.e. $O(n^2)$ where n is the number of data points. Since n iterations are performed in each iteration, the similarity matrix is updated and restored, the time complexity is the order of cube of n, i.e. $O(n^3)$ where n is the number of data points. Hierarchical clustering algorithms have embedded flexibility regarding a level of granularity, and are more versatile, e.g. can handle any form of similarity and any attribute type. Agglomerative hierarchical clustering algorithms can be characterized as greedy, in the algorithmic sense. A sequence of irreversible algorithm steps is used to construct the desired data structure. The following is an analysis of the algorithm, obtained by modifying the parameters of the *AgglomerativeClustering()* function in Python, concerning the number of clusters, the metric used to compute the linkage and the linkage criterion. Tables 4.11 and 4.12 show the ARI and NMI values as the linkage parameter changes, considering a number of clusters equal to 3, for all the similarity measures reported in section 3.1.3. The linkage criterion determines which distance to use between sets of observation. The algorithm will merge the pairs of the cluster that minimize this criterion. Possible values are *complete*, which uses the maximum distances between all observations of the sets, *average*, which uses the average of the distances of each observation of the sets, and *single*, which uses the minimum of the distances between all observations of the sets. The *ward* criterion, which minimizes the variance of the clusters being merged, is not considered as it is only compatible with the Euclidean distance, thus not allowing a more complete and consistent analysis with those made previously. It is clear that using the average distances between all observations, the algorithm leads to better performance for all the Profiles.

Linkage	**Distance**	**Prognostic profile**	**Functional profile**	**Respiratory profile**
Complete	Euclidean	0.0351	0.0510	0.0150
	Manhattan	0.0500	0.0823	0.0199
	Minkowski	0.0351	0.0510	0.0150
	Cosine	0.0351	0.0510	0.0150
	Correlation	0.0288	0.0166	-0.0045
Average	Euclidean	0.0718	0.0913	0.0012
	Manhattan	0.0626	0.0770	0.0202
	Minkowski	0.0718	0.0913	0.0012
	Cosine	0.0778	0.0901	0.0002
	Correlation	0.0328	0.0412	-0.0013
Single	Euclidean	0.0119	-0.0017	0.0078
	Manhattan	0.0119	-0.0008	0.0010
	Minkowski	0.0119	-0.0017	0.0078
	Cosine	0.0119	-0.0017	0.0078
	Correlation	-0.0041	0.0000	-0.0016

Table 4.11: HC Adjusted Rand Index (ARI).

The dendrogram is the graphical representation of the clustering. Usually, it is drawn backwards, starting from the final cluster with all the objects and a similarity equal to 0. At the similarity where two clusters are merged to generate the final cluster, the final cluster splits into the two-parent clusters and so on.

Linkage	Distance	Prognostic profile	Functional profile	Respiratory profile
Complete	Euclidean	0.0643	0.0679	0.0363
	Manhattan	0.0824	0.1170	0.0447
	Minkowski	0.0643	0.0679	0.0363
	Cosine	0.0643	0.0679	0.0363
	Correlation	0.0533	0.0180	0.0038
Average	Euclidean	0.0942	0.1267	0.0440
	Manhattan	0.0823	0.1065	0.0468
	Minkowski	0.0942	0.1267	0.0440
	Cosine	0.1082	0.1261	0.0421
	Correlation	0.0522	0.0539	0.0006
Single	Euclidean	0.0085	0.0026	0.0162
	Manhattan	0.0085	0.0027	0.0109
	Minkowski	0.0085	0.0026	0.0162
	Cosine	0.0085	0.0026	0.0162
	Correlation	0.0048	0.0040	0.0445

Table 4.12: HC Normalized Mutual Information (NMI).

Figure 4.17 represents the dendrograms for the Profiles. The length of the vertical lines measures the separation between the merged clusters, so that it is common practice to cut the dendrogram at the similarity corresponding to the longest branches, to obtain significant clusters [81]. The dendrograms represented show on the y-axis the distance that leads to the best accuracy value, i.e. the Euclidean for the Prognostic and Functional Profile, and the Manhattan for the Respiratory. The linkage criterion used is as discussed above. The optimal choice of the number of clusters appears to coincide with that identified by the clinicians. The graphs representing the Profiles divided into three and five clusters are shown in Figure 4.18 and 4.19. In this case, for none of the three feature subgroups is there any visual improvement by further subdividing patients.

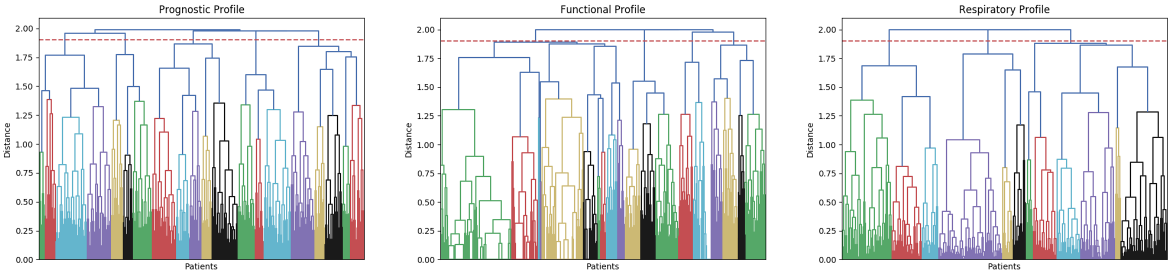

Figure 4.17: Dendrograms of the Clinical Profiles (Hierarchical Clustering Model).

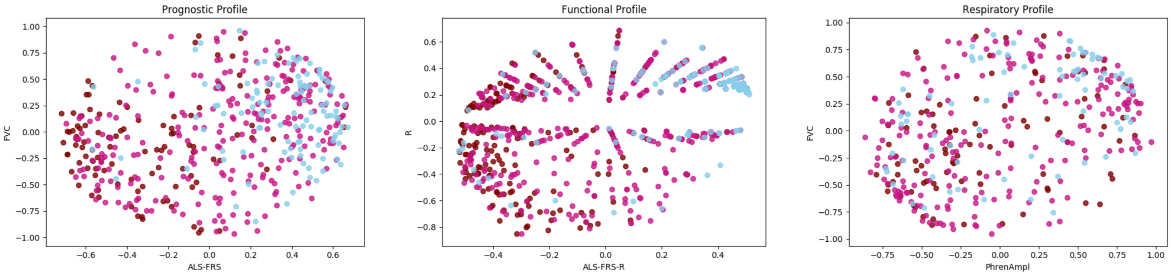

Figure 4.18: 3 Clusters of Patients (Hierarchical Clustering Model).

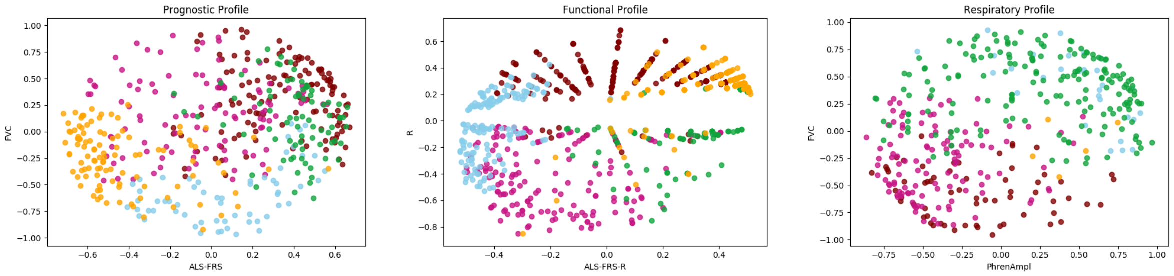

Figure 4.19: 5 Clusters of Patients (Hierarchical Clustering Model).

4.3 Summary

Clustering algorithms identify subgroups of patients in such a way that the similarity between objects within a subgroup is greater than the similarity between objects belonging to different subgroups, according to a certain similarity measure. The clustering algorithms analysed include k-means, k-medoids, DBSSCAN, Gaussian and Bayesian Mixture model and Hierarchical clustering. For each algorithm, all possible parameter values of the implemented function were explored and the final clusters are obtained by maximising the performance of each algorithm. The adjusted rand index and the Normalized Mutual Information are used to evaluate the obtained clusters. Two main aspects emerge from this analysis, such as the importance of the choice of the clusters and the setting of the hyperparameters, which have to be chosen also based on the features considered, and the possibility of identifying 5 groups of patients through a visual and coefficient analysis, such as that of Silhouette.

Chapter 5

Patient stratification using a Clustering Ensemble

Ensemble techniques have been successfully applied in supervised learning to improve the accuracy and stability of classification algorithms [82, 83]. Ensemble techniques require three key issues to be addressed, i.e. generate a collection of base clusterings from which the ensemble is composed, determine the number of clusterings required to give a stable accurate solution, combine the ensemble members to produce the final partition. The term clusterer refers to the clustering algorithms used for the ensemble. The lack of training labels makes the design of ensemble methods for unsupervised learning much more difficult than that for supervised learning. Applying different methods, or the same methods with different parameter choices to the same data, varying clustering results can be obtained. A clustering ensemble framework typically produces a large set of clustering results and then combines them using a consensus function to create a final clustering that is considered to encompass all the information contained in the ensemble. In practice, a cluster ensemble can be obtained in many different ways. Multiple clustering algorithms, different representations of the data, and different parameter choices can all be used to produce a diverse set of clustering solutions. One of the arguments for the ensemble approach is the absence of a universal clustering algorithm since each method has a specific area of its implementation. Some algorithms give more accurate results on data described by spherical patterns in multidimensional feature space, while other methods are intended for searching strip-like clusters or groups of other complicated forms. To deal with complex datasets the advised approach is to apply not a single algorithm, but an organized collection of highly specialized algorithms. A serious problem lies in the possible ambiguous interpretation of obtained clustering solutions. Methods based on different approaches can produce incompatible variants of grouping. In this chapter, a process for aligning the clusters discovered by different clusterers is developed, which works by measuring the similarity between the clusteres by counting their overlapped data items. Then, four methods for combining the aligned clusterers are proposed. They are voting, weighted-voting where the mutual information weights are used in voting, selective voting where the mutual information weights are used to select a subset of clusterers to vote, and selective weighted-voting where the mutual information weights are used not only in selecting but also in voting. A common clustering ensemble framework is represented in Figure 5.1, which consists of three components: ensemble member generation, consensus function and evaluation. As can be seen, the input of the clustering ensemble framework is a given dataset to be clustered, and the output is the final clustering result of this dataset.

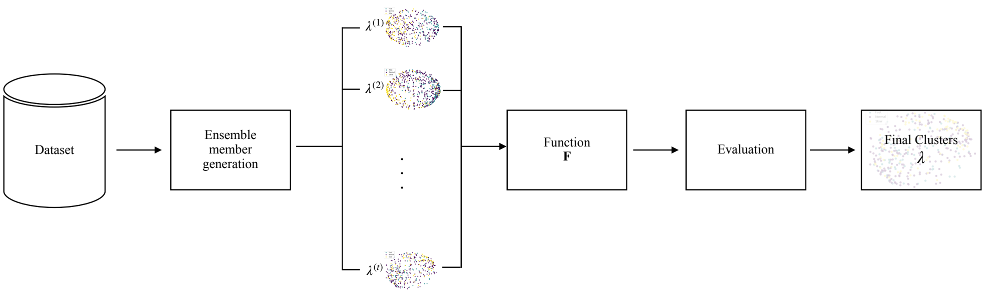

Figure 5.1: A generic clustering ensemble framework.

5.1 Generate component clusterers

Let $\mathbf{X} = \mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n} \subset \mathbb{R}^d$ denotes an unlabeled data set in a feature space of dimension d. The ith data item $\mathbf{x_i}$ is a d-dimensional feature vector $[x_{i1}, x_{i2}, \ldots, x_{id}]^T$, where T denotes vector transpose. Here all the features are numerical, i.e. x_{ij}, with $i = 1, \ldots, n$ and $j = 1, \ldots, d$, is numerical. A clusterer dividing $\mathbf{X}$ into k clusters, could be regarded as a label vector $\boldsymbol{\lambda} \in \mathbf{N}^n$, which assigns the data item $\mathbf{x_i}$ to the λ_ith cluster, i.e. C_{λ_i}, where $\lambda_i \in \{1, 2, \ldots, k\}$. A clusterer ensemble with size t comprises t clusterers, i.e. $\{\lambda^{(1)}, \lambda^{(2)}, \ldots, \lambda^{(t)}\}$, which could also be regarded as a label vector $\boldsymbol{\lambda}$, $\boldsymbol{\lambda} \in \mathbf{N}^n$ and $\boldsymbol{\lambda} = \mathbf{F}(\{\lambda^{(1)}, \lambda^{(2)}, \ldots, \lambda^{(t)}\})$, where $\mathbf{F}(\cdot)$ is a function corresponding to the combining methods presented in Section 5.2.

5.2 Combine component clusterers

The algorithms selected in the ensemble are those that performed best, described in detail in Section 4.2. The simplest combining method is voting, where the ith component of the label vector corresponding to the ensemble, i.e. λ_i, is determined by the plurality voting result of $\lambda^{(1)}, \lambda^{(2)}, \ldots, \lambda^{(t)}$. The plurality is determined in different ways, using mean, median and the mode function. Tables 5.2 and 5.3 show the results obtained using the complete dataset, i.e. the dataset including the features of all the Clinical Profiles, and the Clinical Profiles. The second method, i.e. weighted-voting, employs mutual information between a pair of clusterers [84] to compute the weight for each clusterer. For two label vectors, i.e. $\lambda^{(a)}$ and $\lambda^{(b)}$, suppose there are n objects where n_i are in cluster $C_i(a)$, n_j are in cluster $C_j^{(b)}$, and n_{ij} are in both $C_i(a)$ and $C_j(b)$. The $[0, 1]$-normalized mutual information Φ^{NMI} can be defined as:

$$\Phi^{NMI}(\lambda^{(a)}, \lambda^{(b)}) = \frac{2}{n} \sum_{i=1}^{K} \sum_{j=1}^{K} n^{ij} \log_{k^2} \left(\frac{nn^{ij}}{n_i n_j} \right) \tag{5.1}$$

Then, for every clusterer, the average mutual information can be computed as follows, with $m = 1, \ldots, t$:

$$\beta_m = \frac{1}{t-1} \sum_{l=1, l \neq m}^{t} \Phi^{NMI}(\lambda^{(m)}, \lambda^{(l)}) \tag{5.2}$$

The bigger the value of β_m is, the less statistical information contained by the m-th clusterer has not been contained by other clusterers. Therefore, the weights of the clusterers can be defined as:

$$w_m = \frac{1}{\beta_m Z} \tag{5.3}$$

where Z is used to normalize the weights so that $w_m > 0$ and $\sum_{m=1}^{t} w_m = 1$. In [85], it is shown that selective ensemble methods that select a subset of learners to ensemble may be superior to ensembling all the component learners.

Algorithms	Complete Dataset	Prognostic profile	Functional profile	Respiratory profile
K-means	0.1984	0.2047	0.1988	0.2373
K-medoids	0.1956	0.2175	0.2024	0.2388
GMM	0.3991	0.4340	0.3613	0.2695
HC	0.2069	0.3293	0.2375	0.2545

Table 5.1: Clusters weights.

Voting	Complete Dataset	Prognostic profile	Functional profile	Respiratory profile
Mean	-0.0122	0.0543	0.0827	0.0052
Median	0.0550	0.0542	0.0861	0.0273
Mode	0.0526	0.0351	0.0837	0.0231
Weighted	-0.0131	0.0488	0.0691	-0.0036

Table 5.2: Plurality voting Adjusted Rand Index (ARI).

Voting	Complete Dataset	Prognostic profile	Functional profile	Respiratory profile
Mean	0.1015	0.0667	0.0781	0.0993
Median	0.0801	0.0580	0.0632	0.0878
Mode	0.0899	0.0589	0.0681	0.0763
Weighted	0.0967	0.0621	0.0693	0.0846

Table 5.3: Plurality voting Normalized Mutual Information (NMI).

Figures 5.2 and 5.3 below show the plots of the Clinical Profiles, obtained using the mean and weighted mean of the algorithms, which yield the best performance. From the literature on ensemble learning, it could be found that voting is an effective combining method that is often used in building ensembles of supervised learning algorithms. However, the figures and values of ARI and NMI, show that voting performs quite poor. Combining by the arithmetic mean the algorithms best identify the patient groups, which most closely resemble those identified by the clinicians. A possible explanation can be found in the almost similar performance of the algorithms. Furthermore, as mentioned in the previous chapters, each algorithm is more suitable and leads to better performance for different Profiles, so the combination of all cluster methods enables satisfactory results to be achieved. The mutual information weights, i.e. $\{w_1, w_2, \ldots, w_t\}$, can be used to select the clusterers. This is realized by excluding from the ensemble the clusterers whose mutual information weight is smaller than a threshold. In this work, the threshold is set to $1/t = 1/4 = 0.25$. The selected clusterers can be combined via voting or weighted-voting, based on re-normalized mutual information weights of the selected clusterers. Thus, another two combining methods, i.e. selective voting and selective weighted-voting, are obtained. In the case of the Complete Dataset and the Functional Profile, only one algorithm has a weight that exceeds the threshold value, so the ensemble loses meaning. In the case of the Prognostic and the Respiratory Profile, the GMM and HC algorithms are selected, however, the performance improves insignificantly. The time cost of weighted-voting, selective voting, and selective weighted-voting are comparable, while that of voting is slightly less because it does not require the computation of the mutual information weights. However, the time cost of computing the mutual information weights is negligible if comparing with that of the clustering process. Therefore, the time cost of building an ensemble of clustering algorithms by the proposed methods is roughly m times that of training a single clusterer, where m is the number of clusterers that are trained to be considered for ensembling.

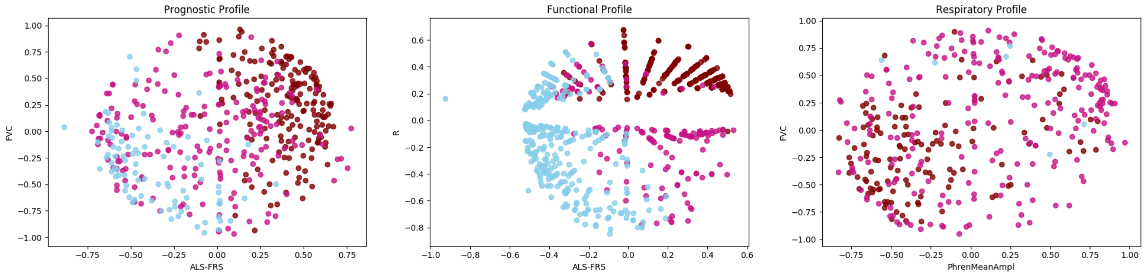

Figure 5.2: Mean Voting Ensemble Clusters.

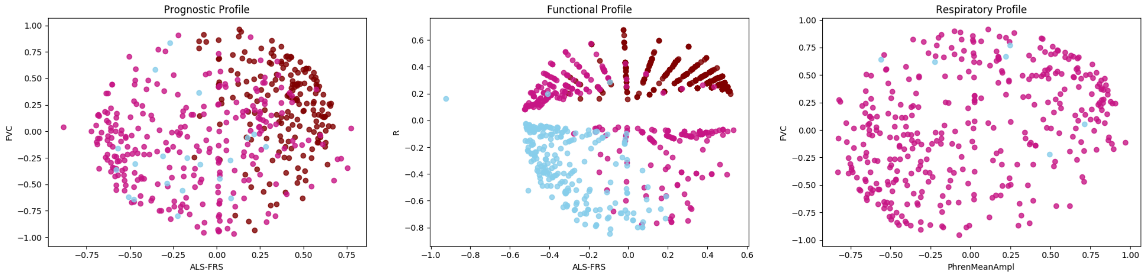

Figure 5.3: Average Voting Ensemble Clusters.

5.3 Summary

Ensemble techniques have been applied to improve the accuracy and performance of clustering algorithms. The simplest method of combination is voting, which can be defined by mean, median or mode function. Weighted voting assigns a weight to the algorithms based on their performance values, so algorithms with better performance are given more weight. From the results obtained, it can be deduced that the best grouping is obtained through the mean, which is expected both because the algorithms present almost the same results in terms of rand index and NMI, and because of the limited number of algorithms considered.

Chapter 6

Conclusions

Machine Learning models have enormous academic and clinical potential in ALS. ML is a rapidly evolving field of applied mathematics, focusing on the development and implementation of computer software that can learn autonomously. In medicine, it has promising diagnostic, prognostic, and risk stratification applications. With the increasing availability of large datasets, multicentre initiatives, high-performance computer platforms, open-source analysis suites, the insights provided by flexible ML models are likely to supersede those gained from conventional statistical approaches. The choice of the ML model needs to be carefully tailored to a proposed application, based on the characteristics of the available data and the flexibility, assumption and limitation profile of the candidate model. While ALS research to date has overwhelmingly relied on conventional ML approaches, emerging models and network architectures have considerable potential to advance the field. The overarching intention of this work is to outline best practice recommendations for ML applications in ALS. Machine learning encompasses two main approaches, i.e. supervised and unsupervised learning. Unsupervised learning can be particularly helpful in addressing patient stratification problems. Clustering methods can be superior to current clinical criteria, which are often based on a limited set of clinical observations, rigid thresholds, and conservative inclusion/exclusion criteria for class membership. Moreover, unsupervised learning methods have been successfully used in other fields of medicine [2, 86–89], thus giving hope that their use will also boost ALS research. The new National project, entitled *Advanced learning models using Patient profiles and disease progression patterns for prognostic prediction in ALS* (AIpALS), aims to advance precision medicine and improve supportive care in ALS. This thesis is motivated by the AIpALS project and its results will be considered and possibly included.

6.1 Achievements

The analysis presented in this work, use a large dataset of genotype-phenotype data and clinical temporal data, collected by the national FCT project NEUROCLINOMICS2 (2016-2019, PI Sara C. Madeira) and European JPND project OnWebDuals (2016-2019, PI Mamede de Carvalho). Since they are so fundamental to data analytics, descriptive and inferential statistics are explored, which focus on describing the visible characteristics of a dataset and on making predictions or generalizations about a dataset, respectively. Moreover, given the importance of the data type in the implementation of ML algorithms, an analysis of missing values and possible methods for dealing with them is initialised. Finding the best configuration for the parameters of an algorithm is known as hyperparameter tuning. Properly selecting hyperparameters can significantly speed up the search for a proper generalized model without sacrificing performance. In this work, much importance is given to the choice of hyperparameters, which are explored and analysed to set values that lead to a good performance of the algorithm. The implemented algorithms are constructed to test all possible values of a parameter, and the choice combines the best values for each parameter of the function. This allows, in a more detailed future research, the reuse of these algorithms, which according to the dataset provided as input, will set the hyperparameters to maximise the performance of the algorithm. The major achievement of the present work concerns the exploration of data using graph theory to construct a patient network. Several patient networks are constructed, which allow the subdivision of patients into groups, also in a visual sense, which are then analysed using different metrics. Patients are compared in a pairwise manner, clinical features are integrated into a vector and the diagnostic similarity is computed. In detail, parameters such as features to be considered, similarity distances and thresholds, are set to identify homogeneous clusters and predict diagnoses, i.e. demonstrate or verify that the clusters created to reflect those identified by clinicians. In addition, clustering algorithms and ensemble learning strategies are analysed, to identify homogeneous patient groups and the best approach to treat ALS disease. The results obtained show the ability of the implemented algorithms to identify homogeneous subgroups of patients, which similarly reflect those provided by clinicians. They also suggest the possibility, in future research, of dividing patients into a larger number of groups, to increase the homogeneity within each cluster. As in the case of the network approach, the values that can be assumed by the parameters are explored, to derive the best results from the algorithms and optimise their performance. From the analysis reported in this dissertation, it is possible to derive important information for future ALS research, using the preliminary results and the implemented algorithms. In particular, the work suggests the possibility of revising the groups of patients identified by the clinicians, introducing new ones that are more able to describe the peculiarities of the patients. The construction of more homogeneous subgroups should promote the effectiveness of prognostic prediction models.

6.2 Future Work

Machine Learning models have considerable advantages over traditional statistical approaches for modeling complex datasets. Most ML models, including the approaches presented in Section 4.2, do not require stringent assumptions on data characteristics. Despite the pragmatic advantages, the application of ML models requires a clear understanding of what determines model performance and the potential pitfalls of specific models. The most common shortcomings concern data sparsity, data bias, and causality assumptions. Good practice recommendations for model design include the management of missing data, model overfitting, model validation, and performance reporting. Data sparsity refers to working and interpreting limited datasets, which is particularly common in medical applications. Medical data is often costly, difficult to acquire, frequently require invasive, uncomfortable, or time-consuming procedures. Other factors contributing to the sparsity of medical data include strict anonymization procedures, requirements for informed consent, institutional, and cross-border data management regulations, ethics approvals, and other governance issues. The processing, storage, and labeling of medical data is also costly and often requires specific funding to upkeep registries, DNA banks, brain banks, biofluid facilities, or magnetic resonance repositories. Multicenter protocols are particularly challenging and require additional logistics, harmonization of data acquisition, standardized operating procedures, and bio-sample processing, such as cooling, freezing, spinning, staining, etc. Most ML models have originally been intended, developed, and optimized for huge quantities of data. Accordingly, the generalizability of most ML models depends heavily on the number of samples upon which they can effectively learn. Additionally, the number of samples required for a specific level of accuracy grows exponentially with the number of features [90]. If the number of samples is restrictively low, then the features lose their discriminating power, as all samples in the dataset seem very distinct from one another [91]. Also discussing data bias is particularly pertinent when dealing with medical data. The entire spectrum of data distribution should be represented in the dataset, just as observed in the overall population, otherwise, the model is not able to generalize properly. Medical data are particularly prone to suffer from a variety of data biases that affect recorded data at different analysis levels. The four most common types of bias include the study participation bias, the study attrition bias, the prognostic factor measurement bias and the outcome measurement bias [92]. In ALS, study participation bias is by far the most significant. It affects prognostic modeling in particular, as patients in clinical trials do not reflect the general ALS population. Unfortunately, very little can be done to correct for participation bias post-hoc, therefore its potential impact needs to be carefully considered when interpreting the results. Study attrition bias also influences ALS studies, as data censoring is not always systematically recorded. Prognostic factor measurements can be influenced by subjective and qualitative medical assessments and by machine bias in imaging data interpretation. The single most important principle to manage these factors, especially if limited data are available, is overtly discussing the type of bias affecting a particular study, and openly reporting them. An extension of the work concerns the possibility of considering all available patient data, i.e. the results of every test carried out over the years, by solving the data quality problem. Furthermore, the approaches implemented are evaluated only with numerical data, whereas distributed clinical data possibly included images and text. The applicability of the proposed methods to different types of data and data gathered from many independent hospitals should be verified in the future. This would provide a larger dataset allowing for more in-depth and detailed analysis and thus more accurate stratification. A possible way of integrating more data into the algorithms is proposed here, where each hospital appointment of a patient is treated as a different patient. Each item in the dataset is then renamed, with the original patient identification number in each row, to verify, once clusters have been identified, how patients are classified over time. For example, a patient who undergoes 10 visits has 10 rows related to him/her in the dataset, which are renamed as ij where i indicates the patient ID and j the appointment number considered. As a result of the clustering process, it can be seen whether the various appointments of the patient, now considered as different patients, fall within the same progression group. From this analysis, it is then possible to demonstrate whether the stratification is effective, i.e. if the appointments of the same patient are grouped in the same cluster. One of the future directions which concern the construction of networks will be to further investigate which weights should be given to each type of data, and each particular feature. This weighting could be performed automatically [46], or based on expert knowledge. By using feature weights, a single network can be constructed, as shown in Figure 3.2, using all the networks built using the Clinical Profiles, which generates a consensus clustering and general patient profiles. Feedback analysis is also intended to be conducted, recalculating the distance/similarity between patients using only the relevant features for the modular structure. Methodologies such as this look promising in terms of knowledge discovery with little or no prior knowledge, where the conclusions are achieved in an unsupervised fashion and may help to gain new insights on different diseases. Finally, such modules could be used to train expert models for classification problems regarding subgroups of patients, possibly discriminating the ones with different disease progression rates. Future works should further explore these approaches by using more data, other clustering approaches and similarity measures.

Bibliography

[1] Vincent Grollemund, Pierre-François Pradat, Giorgia Querin, François Delbot, Gaétan Le Chat, Jean-François Pradat-Peyre, and Peter Bede. Machine learning in amyotrophic lateral sclerosis: achievements, pitfalls, and future directions. *Frontiers in neuroscience*, 13:135, 2019.

[2] Henk-Jan Westeneng, Thomas PA Debray, Anne E Visser, Ruben PA van Eijk, James PK Rooney, Andrea Calvo, Sarah Martin, Christopher J McDermott, Alexander G Thompson, Susana Pinto, et al. Prognosis for patients with amyotrophic lateral sclerosis: development and validation of a personalised prediction model. *The Lancet Neurology*, 17(5):423–433, 2018.

[3] Michael A van Es, Orla Hardiman, Adriano Chio, Ammar Al-Chalabi, R Jeroen Pasterkamp, Jan H Veldink, and Leonard H Van den Berg. Amyotrophic lateral sclerosis. *The Lancet*, 390(10107):2084–2098, 2017.

[4] Shraddha Pai and Gary D Bader. Patient similarity networks for precision medicine. *Journal of molecular biology*, 430(18):2924–2938, 2018.

[5] Rui Henriques and Sara C Madeira. Bicpam: Pattern-based biclustering for biomedical data analysis. *Algorithms for Molecular Biology*, 9(1):27, 2014.

[6] Telma Pereira, Francisco L Ferreira, Sandra Cardoso, Dina Silva, Alexandre de Mendonça, Manuela Guerreiro, Sara C Madeira, Alzheimer's Disease Neuroimaging Initiative, et al. Neuropsychological predictors of conversion from mild cognitive impairment to alzheimer's disease: a feature selection ensemble combining stability and predictability. *BMC medical informatics and decision making*, 18(1):137, 2018.

[7] Alvin Rajkomar, Jeffrey Dean, and Isaac Kohane. Machine learning in medicine. *New England Journal of Medicine*, 380(14):1347–1358, 2019.

[8] Helena Aidos, Ana Fred, Alzheimer's Disease Neuroimaging Initiative, et al. Discrimination of alzheimer's disease using longitudinal information. *Data Mining and Knowledge Discovery*, 31(4):1006–1030, 2017.

[9] Evelina Gabasova, John Reid, and Lorenz Wernisch. Clusternomics: Integrative context-dependent clustering for heterogeneous datasets. *PLoS computational biology*, 13(10):e1005781, 2017.

[10] Zachary C Lipton, David C Kale, Charles Elkan, and Randall Wetzel. Learning to diagnose with lstm recurrent neural networks. *arXiv preprint arXiv:1511.03677*, 2015.

[11] Caio Eduardo Ribeiro and Luis Enrique Zárate. Classifying longevity profiles through longitudinal data mining. *Expert Systems with Applications*, 117:75–89, 2019.

[12] Juan Zhao, QiPing Feng, Patrick Wu, Roxana A Lupu, Russell A Wilke, Quinn S Wells, Joshua C Denny, and Wei-Qi Wei. Learning from longitudinal data in electronic health record and genetic data to improve cardiovascular event prediction. *Scientific reports*, 9(1):1–10, 2019.

[13] Fang Zhang, Mei Liu, Qun Li, and Fei-Xue Song. Exploration of attractor modules for sporadic amyotrophic lateral sclerosis via systemic module inference and attract method. *Experimental and therapeutic medicine*, 17(4):2575–2580, 2019.

[14] Stephen R Pfohl, Renaid B Kim, Grant S Coan, and Cassie S Mitchell. Unraveling the complexity of amyotrophic lateral sclerosis survival prediction. *Frontiers in neuroinformatics*, 12:36, 2018.

[15] Özden O Dalgıç, F Safa Erenay, Kalyan S Pasupathy, Osman Y Özaltın, Brian A Crum, and Mustafa Y Sir. Tollgate-based progression pathways of als patients. *Journal of neurology*, 266(3):755–765, 2019.

[16] Cláudia S Silva, Filipe B Rodrigues, Gonçalo S Duarte, João Costa, and Mamede de Carvalho. Prognostic value of phrenic nerve conduction study in amyotrophic lateral sclerosis: Systematic review and meta-analysis. *Clinical Neurophysiology*, 131(1):106–113, 2020.

[17] Sherry-Ann Brown. Patient similarity: emerging concepts in systems and precision medicine. *Frontiers in physiology*, 7:561, 2016.

[18] Alessandro Zandonà, Rosario Vasta, Adriano Chiò, and Barbara Di Camillo. A dynamic bayesian network model for the simulation of amyotrophic lateral sclerosis progression. *BMC bioinformatics*, 20(4):118, 2019.

[19] David Westergaard, Pope Moseley, Freja Karuna Hemmingsen Sørup, Pierre Baldi, and Søren Brunak. Population-wide analysis of differences in disease progression patterns in men and women. *Nature communications*, 10(1):1–14, 2019.

[20] Martin R Turner, Jakub Scaber, John A Goodfellow, Melanie E Lord, Rachael Marsden, and Kevin Talbot. The diagnostic pathway and prognosis in bulbar-onset amyotrophic lateral sclerosis. *Journal of the neurological sciences*, 294(1-2):81–85, 2010.

[21] Mamede de Carvalho, Susana Pinto, and Michael Swash. Motor unit changes in thoracic paraspinal muscles in amyotrophic lateral sclerosis. *Muscle & nerve*, 39(1):83–86, 2009.

[22] Fusun Baumann, Robert D Henderson, Stephen C Morrison, Michael Brown, N Hutchinson, James A Douglas, Peter J Robinson, and Pamela A McCombe. Use of respiratory function tests to predict survival in amyotrophic lateral sclerosis. *Amyotrophic Lateral Sclerosis*, 11(1-2):194–202, 2010.

[23] Katja Kollewe, Ulrike Mauss, Klaus Krampfl, Susanne Petri, Reinhard Dengler, and Bahram Mohammadi. Alsfrs-r score and its ratio: a useful predictor for als-progression. *Journal of the neurological sciences*, 275(1-2):69–73, 2008.

[24] André V Carreiro, Pedro MT Amaral, Susana Pinto, Pedro Tomás, Mamede de Carvalho, and Sara C Madeira. Prognostic models based on patient snapshots and time windows: Predicting disease progression to assisted ventilation in amyotrophic lateral sclerosis. *Journal of biomedical informatics*, 58:133–144, 2015.

[25] Sofia Pires, Marta Gromicho, Susana Pinto, Mamede Carvalho, and Sara C Madeira. Predicting non-invasive ventilation in als patients using stratified disease progression groups. In *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 748–757. IEEE, 2018.

[26] Jil M Meier, Hannelore K van der Burgh, Abram D Nitert, Peter Bede, Siemon C de Lange, Orla Hardiman, Leonard H van den Berg, and Martijn P van den Heuvel. Connectome-based propagation model in amyotrophic lateral sclerosis. *Annals of Neurology*, 87(5):725–738, 2020.

[27] William R Swindell, Colin PS Kruse, Edward O List, Darlene E Berryman, and John J Kopchick. Als blood expression profiling identifies new biomarkers, patient subgroups, and evidence for neutrophilia and hypoxia. *Journal of translational medicine*, 17(1):1–33, 2019.

[28] José Verdú-Díaz, Jorge Alonso-Pérez, Claudia Nuñez-Peralta, Giorgio Tasca, John Vissing, Volker Straub, Roberto Fernández-Torrón, Jaume Llauger, Isabel Illa, and Jordi Díaz-Manera. Accuracy of a machine learning muscle mri-based tool for the diagnosis of muscular dystrophies. *Neurology*, 94(10):e1094–e1102, 2020.

[29] Enrico Grisan, Alessandro Zandonà, and Barbara Di Camillo. Deep convolutional neural network for survival estimation of amyotrophic lateral sclerosis patients. In *ESANN*, 2019.

[30] Finn V Jensen et al. *An introduction to Bayesian networks*, volume 210. UCL press London, 1996.

[31] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[32] Lu Xu, Bingjie He, Yunjing Zhang, Lu Chen, Dongsheng Fan, Siyan Zhan, and Shengfeng Wang. Prognostic models for amyotrophic lateral sclerosis: a systematic review. *Journal of Neurology*, pages 1–10, 2021.

[33] Ewout W Steyerberg et al. *Clinical prediction models*. Springer, 2019.

[34] Karel GM Moons, Robert F Wolff, Richard D Riley, Penny F Whiting, Marie Westwood, Gary S Collins, Johannes B Reitsma, Jos Kleijnen, and Sue Mallett. Probast: a tool to assess risk of bias and applicability of prediction model studies: explanation and elaboration. *Annals of internal medicine*, 170(1):W1–W33, 2019.

[35] Vincent Grollemund, Gaetan Le Chat, Marie-Sonia Secchi-Buhour, Francois Delbot, Jean-Francois Pradat-Peyre, Peter Bede, and Pierre-Francois Pradat. Manifold learning for amyotrophic lateral sclerosis functional loss assessment. *Journal of Neurology*, 268(3):825–850, 2021.

[36] Karl Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.

[37] YH Taguchi, Mitsuo Iwadate, and Hideaki Umeyama. Heuristic principal component analysis-based unsupervised feature extraction and its application to gene expression analysis of amyotrophic lateral sclerosis data sets. In *2015 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, pages 1–10. IEEE, 2015.

[38] Ming Tang, Chao Gao, Stephen A Goutman, Alexandr Kalinin, Bhramar Mukherjee, Yuanfang Guan, and Ivo D Dinov. Model-based and model-free techniques for amyotrophic lateral sclerosis diagnostic prediction and patient clustering. *Neuroinformatics*, 17(3):407–421, 2019.

[39] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[40] Ton Fang, Ahmad Al Khleifat, Daniel R Stahl, Claudia Lazo La Torre, Caroline Murphy, Uk-Mnd LicalS, Carolyn Young, Pamela J Shaw, P Nigel Leigh, and Ammar Al-Chalabi. Comparison of the king's and mitos staging systems for als. *Amyotrophic Lateral Sclerosis and Frontotemporal Degeneration*, 18(3-4):227–232, 2017.

[41] Adriano Chiò, Edward R Hammond, Gabriele Mora, Virginio Bonito, and Graziella Filippini. Development and evaluation of a clinical staging system for amyotrophic lateral sclerosis. *Journal of Neurology, Neurosurgery & Psychiatry*, 86(1):38–44, 2015.

[42] Jose C Roche, Ricardo Rojas-Garcia, Kirsten M Scott, William Scotton, Catherine E Ellis, Rachel Burman, Lokesh Wijesekera, Martin R Turner, P Nigel Leigh, Christopher E Shaw, et al. A proposed staging system for amyotrophic lateral sclerosis. *Brain*, 135(3):847–852, 2012.

[43] Jesse M Cedarbaum, Nancy Stambler, Errol Malta, Cynthia Fuller, Dana Hilt, Barbara Thurmond, Arline Nakanishi, Bdnf Als Study Group, 1A complete listing of the BDNF Study Group, et al. The alsfrs-r: a revised als functional rating scale that incorporates assessments of respiratory function. *Journal of the neurological sciences*, 169(1-2):13–21, 1999.

[44] Rubika Balendra, Ahmad Al Khleifat, Ton Fang, and Ammar Al-Chalabi. A standard operating procedure for king's als clinical staging. *Amyotrophic Lateral Sclerosis and Frontotemporal Degeneration*, 20(3-4):159–164, 2019.

[45] Jimeng Sun, Fei Wang, Jianying Hu, and Shahram Edabollahi. Supervised patient similarity measure of heterogeneous patient records. *Acm Sigkdd Explorations Newsletter*, 14(1):16–24, 2012.

[46] Sebastian Klenk, Jürgen Dippon, Peter Fritz, and Gunther Heidemann. Determining patient similarity in medical social networks. In *Proceedings of the First International Workshop on Web Science and Information Exchange in the Medical Web*, pages 6–14, 2010.

[47] Zheng Jia, Xian Zeng, Huilong Duan, Xudong Lu, and Haomin Li. A patient-similarity-based model for diagnostic prediction. *International Journal of Medical Informatics*, 135:104073, 2020.

[48] Robert Kueffner, Neta Zach, Maya Bronfeld, Raquel Norel, Nazem Atassi, Venkat Balagurusamy, Barbara Di Camillo, Adriano Chio, Merit Cudkowicz, Donna Dillenberger, et al. Stratification of amyotrophic lateral sclerosis patients: a crowdsourcing approach. *Scientific reports*, 9(1):1–14, 2019.

[49] Sofia Pires, Marta Gromicho, Susana Pinto, Mamede de Carvalho, and Sara C Madeira. Patient stratification using clinical and patient profiles: Targeting personalized prognostic prediction in als. In *International Work-Conference on Bioinformatics and Biomedical Engineering*, pages 529–541. Springer, 2020.

[50] André Carreiro, Sara C . Madeira, and Alexandre Francisco. Unravelling communities of als patients using network mining. 08 2013.

[51] S Beltran, M Nassif, E Vicencio, J Arcos, L Labrador, BI Cortes, C Cortez, CA Bergmann, S Espinoza, MF Hernandez, et al. Network approach identifies pacer as an autophagy protein involved in als pathogenesis. *Molecular neurodegeneration*, 14(1):1–18, 2019.

[52] Simon Cronin, Hylke M Blauw, Jan H Veldink, Michael A Van Es, Roel A Ophoff, Daniel G Bradley, Leonard H Van Den Berg, and Orla Hardiman. Analysis of genome-wide copy number variation in irish and dutch als populations. *Human molecular genetics*, 17(21):3392–3398, 2008.

[53] Hylke M Blauw, Ammar Al-Chalabi, Peter M Andersen, Paul WJ van Vught, Frank P Diekstra, Michael A van Es, Christiaan GJ Saris, Ewout JN Groen, Wouter van Rheenen, Max Koppers, et al. A large genome scan for rare cnvs in amyotrophic lateral sclerosis. *Human molecular genetics*, 19(20):4091–4099, 2010.

[54] Hylke M Blauw, Jan H Veldink, Michael A van Es, Paul W van Vught, Christiaan GJ Saris, Bert van der Zwaag, Lude Franke, J Peter H Burbach, John H Wokke, Roel A Ophoff, et al. Copy-number variation in sporadic amyotrophic lateral sclerosis: a genome-wide screen. *The Lancet Neurology*, 7(4):319–326, 2008.

[55] Louise V Wain, Inti Pedroso, John E Landers, Gerome Breen, Christopher E Shaw, P Nigel Leigh, Robert H Brown, Martin D Tobin, and Ammar Al-Chalabi. The role of copy number variation in susceptibility to amyotrophic lateral sclerosis: genome-wide association study and comparison with published loci. *PloS one*, 4(12):e8175, 2009.

[56] Maxime Peralta, Pierre Jannin, Claire Haegelen, and John SH Baxter. Data imputation and compression for parkinson's disease clinical questionnaires. *Artificial Intelligence in Medicine*, 114:102051, 2021.

[57] Xiao-Hua Zhou, George J Eckert, and William M Tierney. Multiple imputation in public health research. *Statistics in medicine*, 20(9-10):1541–1549, 2001.

[58] A Rogier T Donders, Geert JMG Van Der Heijden, Theo Stijnen, and Karel GM Moons. A gentle introduction to imputation of missing values. *Journal of clinical epidemiology*, 59(10):1087–1091, 2006.

[59] Ian R White, Patrick Royston, and Angela M Wood. Multiple imputation using chained equations: issues and guidance for practice. *Statistics in medicine*, 30(4):377–399, 2011.

[60] Stéphane Dray and Julie Josse. Principal component analysis with missing values: a comparative survey of methods. *Plant Ecology*, 216(5):657–667, 2015.

[61] Mussa Abdella and Tshilidzi Marwala. The use of genetic algorithms and neural networks to approximate missing data in database. In *IEEE 3rd International Conference on Computational Cybernetics, 2005. ICCC 2005.*, pages 207–212. IEEE, 2005.

[62] Riccardo Miotto, Li Li, Brian A Kidd, and Joel T Dudley. Deep patient: an unsupervised representation to predict the future of patients from the electronic health records. *Scientific reports*, 6(1):1–10, 2016.

[63] Feng Chen, Shuang Wang, Xiaoqian Jiang, Sijie Ding, Yao Lu, Jihoon Kim, S Cenk Sahinalp, Chisato Shimizu, Jane C Burns, Victoria J Wright, et al. Princess: Privacy-protecting rare disease international network collaboration via encryption through software guard extensions. *Bioinformatics*, 33(6):871–878, 2017.

[64] Christopher M Bishop et al. *Neural networks for pattern recognition*. Oxford university press, 1995.

[65] Vladimir Cherkassky and Filip M Mulier. *Learning from data: concepts, theory, and methods*. John Wiley & Sons, 2007.

[66] Richard O Duda, Peter E Hart, and David G Stork. Pattern classification, 2nd edn new york. *NY: Wiley*, 2001.

[67] Ron Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145. Montreal, Canada, 1995.

[68] Brian S Everitt, S Landau, and M Leese. Cluster analysis arnold. *A member of the Hodder Headline Group, London*, pages 429–438, 2001.

[69] Anil K Jain and Richard C Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., 1988.

[70] Eric Backer and Anil K Jain. A clustering performance measure based on fuzzy set decomposition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (1):66–75, 1981.

[71] Pang-Ning Tan, Micahel Steinbach, and Vipin Kumar. Introduction to data mining, pearson education. *Inc., New Delhi*, 2006.

[72] Guojun Gan, Jialun Yin, Yulia Wang, and Jianhong Wu. Complex data clustering: From neural network architecture to theory and applications of nonlinear dynamics of pattern recognition. In *BIOMAT 2013: International Symposium on Mathematical and Computational Biology*, pages 85–106. World Scientific, 2014.

[73] Charu C Aggarwal. *Data mining: the textbook*. Springer, 2015.

[74] Julia Handl, Joshua Knowles, and Douglas B Kell. Computational cluster validation in post-genomic data analysis. *Bioinformatics*, 21(15):3201–3212, 2005.

[75] Qinpei Zhao. *Cluster validity in clustering methods*. PhD thesis, Itä-Suomen yliopisto, 2012.

[76] Joseph C Dunn. Well-separated clusters and optimal fuzzy partitions. *Journal of cybernetics*, 4(1):95–104, 1974.

[77] Marina Sokolova and Guy Lapalme. A systematic analysis of performance measures for classification tasks. *Information processing & management*, 45(4):427–437, 2009.

[78] LNF Ana and Anil K Jain. Robust data clustering. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, volume 2, pages II–II. IEEE, 2003.

[79] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977.

[80] Pak K Chan, Martine DF Schlag, and Jason Y Zien. Spectral k-way ratio-cut partitioning and clustering. *IEEE Transactions on computer-aided design of integrated circuits and systems*, 13(9):1088–1096, 1994.

[81] M Forina, C Armanino, and V Raggio. Clustering with dendrograms on interpretation variables. *Analytica Chimica Acta*, 454(1):13–19, 2002.

[82] Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

[83] Alexey Tsymbal, Mykola Pechenizkiy, and Pádraig Cunningham. Diversity in ensemble feature selection. In *Technical report*. Citeseer, 2003.

[84] Alexander Strehl, Joydeep Ghosh, and Raymond Mooney. Impact of similarity measures on web-page clustering. In *Workshop on artificial intelligence for web search (AAAI 2000)*, volume 58, page 64, 2000.

[85] Zhi-Hua Zhou, Jianxin Wu, and Wei Tang. Ensembling neural networks: many could be better than all. *Artificial intelligence*, 137(1-2):239–263, 2002.

[86] Roberto Gomeni, Maurizio Fava, and Pooled Resource Open-Access ALS Clinical Trials Consortium. Amyotrophic lateral sclerosis disease progression model. *Amyotrophic Lateral Sclerosis and Frontotemporal Degeneration*, 15(1-2):119–129, 2014.

[87] Benoît Marin, Philippe Couratier, Simona Arcuti, Massimiliano Copetti, Andrea Fontana, Marie Nicol, Marie Raymondeau, Giancarlo Logroscino, and Pierre Marie Preux. Stratification of als patients' survival: a population-based study. *Journal of neurology*, 263(1):100–111, 2016.

[88] Brett K Beaulieu-Jones, Casey S Greene, et al. Semi-supervised learning of the electronic health record for phenotype stratification. *Journal of biomedical informatics*, 64:168–178, 2016.

[89] Mei-Lyn Ong, Pei Fang Tan, and Joanna D Holbrook. Predicting functional decline and survival in amyotrophic lateral sclerosis. *PLoS One*, 12(4):e0174925, 2017.

[90] Hanan Samet. *Foundations of multidimensional and metric data structures.* Morgan Kaufmann, 2006.

[91] Vladimir Pestov. An axiomatic approach to intrinsic dimension of a dataset. *Neural Networks*, 21(2-3):204–213, 2008.

[92] Jill A Hayden, Danielle A van der Windt, Jennifer L Cartwright, Pierre Côté, and Claire Bombardier. Assessing bias in studies of prognostic factors. *Annals of internal medicine*, 158(4):280–286, 2013.

Appendix A

A.1 Network metrics

Threshold	Distance	Prognostic profile	Functional profile	Respiratory profile
0.6	Euclidean	299926	109523	427244
	Manhattan	158116	31556	344289
	Minkowski	375808	211616	449994
	Cosine	592499	483200	548500
	Correlation	382838	171484	260813
0.7	Euclidean	200070	54016	343551
	Manhattan	103181	20310	277471
	Minkowski	268654	130279	370025
	Cosine	576847	478995	547844
	Correlation	341995	135236	257734
0.8	Euclidean	101756	25452	237346
	Manhattan	54765	12919	228365
	Minkowski	139510	44517	240302
	Cosine	539195	467451	544481
	Correlation	285393	98688	251916
0.9	Euclidean	29748	8770	161164
	Manhattan	19184	3396	159618
	Minkowski	36992	15322	161708
	Cosine	486709	426310	540025
	Correlation	202272	60013	232533

Table A.1: Number of edges m.

Threshold	Distance	Complete graph	Slow subgraph	Normal subgraph	Fast subgraph
0.6	Euclidean	0.0403	0.0859	0.0560	0.08367
	Manhattan	0.0372	0.0787	0.0515	0.07656
	Minkowski	0.0417	0.0876	0.0578	0.0867
	Cosine	0.0302	0.0602	0.0422	0.0620
	Correlation	0.0284	0.0567	0.0399	0.0594
0.7	Euclidean	0.0381	0.0809	0.0529	0.0795
	Manhattan	0.0350	0.0733	0.0483	0.0714
	Minkowski	0.0396	0.0844	0.0552	0.0825
	Cosine	0.0301	0.0601	0.0421	0.0617
	Correlation	0.0282	0.0561	0.0395	0.0587
0.8	Euclidean	0.0346	0.0725	0.0479	0.0722
	Manhattan	0.0314	0.0648	0.0432	0.0630
	Minkowski	0.0362	0.0762	0.0506	0.0761
	Cosine	0.0297	0.0596	0.0415	0.0610
	Correlation	0.0290	0.0574	0.0407	0.0601
0.9	Euclidean	0.0208	0.0404	0.0286	0.0422
	Manhattan	0.0195	0.0369	0.0271	0.0396
	Minkowski	0.0215	0.0418	0.0296	0.0437
	Cosine	0.0272	0.0537	0.0376	0.0549
	Correlation	0.0278	0.0552	0.0389	0.0574

Table A.2: Prognostic group: average Eigenvector centrality

Threshold	Distance	Complete graph	Slow subgraph	Normal subgraph	Fast subgraph
0.6	Euclidean	0.0011	0.0045	0.0018	0.0055
	Manhattan	0.0017	0.0077	0.0031	0.0082
	Minkowski	0.0008	0.0035	0.0012	0.0039
	Cosine	0.0000	0.0000	0.0000	0.0000
	Correlation	0.0003	0.0011	0.0006	0.0012
0.7	Euclidean	0.0015	0.0069	0.0027	0.0073
	Manhattan	0.0020	0.0091	0.0037	0.0098
	Minkowski	0.0013	0.0057	0.0020	0.0063
	Cosine	0.0000	0.0001	0.0000	0.0002
	Correlation	0.0004	0.0013	0.0007	0.0015
0.8	Euclidean	0.0023	0.0106	0.0042	0.0103
	Manhattan	0.0027	0.0129	0.0050	0.0126
	Minkowski	0.0020	0.0102	0.0036	0.0100
	Cosine	0.0000	0.0003	0.0002	0.0004
	Correlation	0.0005	0.0015	0.0009	0.0019
0.9	Euclidean	0.0015	0.0074	0.0029	0.0064
	Manhattan	0.0015	0.0077	0.0029	0.0056
	Minkowski	0.0014	0.0071	0.0028	0.0060
	Cosine	0.0002	0.0007	0.0003	0.0007
	Correlation	0.0006	0.0020	0.0011	0.0025

Table A.3: Prognostic group: average Betweenness centrality

Threshold	Distance	Complete graph	Slow subgraph	Normal subgraph	Fast subgraph
0.6	Euclidean	0.6501	0.6862	0.6859	0.6253
	Manhattan	0.5476	0.5548	0.5602	0.5113
	Minkowski	0.7275	0.7409	0.7748	0.7058
	Cosine	0.9853	0.9867	0.9898	0.9857
	Correlation	0.7422	0.7912	0.7707	0.7735
0.7	Euclidean	0.5696	0.5818	0.5920	0.5383
	Manhattan	0.4980	0.4689	0.5041	0.4405
	Minkowski	0.6214	0.6300	0.6577	0.5890
	Cosine	0.9641	0.9747	0.9715	0.9630
	Correlation	0.7042	0.7628	0.7296	0.7323
0.8	Euclidean	0.4716	0.4383	0.4827	0.4069
	Manhattan	0.4137	0.3260	0.4076	0.3046
	Minkowski	0.5017	0.4875	0.5230	0.4504
	Cosine	0.9184	0.9297	0.9276	0.9287
	Correlation	0.6566	0.7212	0.6751	0.6828
0.9	Euclidean	0.3858	0.3218	0.3792	0.3344
	Manhattan	0.3665	0.2702	0.3442	0.2802
	Minkowski	0.3953	0.3430	0.3927	0.3499
	Cosine	0.8593	0.8567	0.8843	0.8655
	Correlation	0.5963	0.6562	0.6046	0.6172

Table A.4: Prognostic group: average Closeness centrality

Threshold	Distance	Complete graph	Slow subgraph	Normal subgraph	Fast subgraph
0.6	Euclidean	0.7596	0.7750	0.7514	0.7639
	Manhattan	0.6802	0.6721	0.6863	0.6739
	Minkowski	0.8153	0.8338	0.8094	0.8111
	Cosine	0.9872	0.9879	0.9869	0.9871
	Correlation	0.8358	0.8565	0.8265	0.8339
0.7	Euclidean	0.6822	0.6909	0.6771	0.6859
	Manhattan	0.6427	0.6237	0.6535	0.6363
	Minkowski	0.7054	0.7235	0.6981	0.7049
	Cosine	0.9738	0.9725	0.9733	0.9761
	Correlation	0.8008	0.8213	0.7910	0.8000
0.8	Euclidean	0.6019	0.5921	0.6077	0.5979
	Manhattan	0.6079	0.5661	0.6256	0.6070
	Minkowski	0.5871	0.6038	0.0036	0.5708
	Cosine	0.9580	0.9471	0.9582	0.9690
	Correlation	0.7539	0.7719	0.7443	0.7554
0.9	Euclidean	0.5467	0.5364	0.5563	0.5369
	Manhattan	0.5322	0.5069	0.5465	0.5279
	Minkowski	0.5230	0.5122	0.5291	0.5213
	Cosine	0.9490	0.9555	0.9492	0.9418
	Correlation	0.6818	0.6928	0.6766	0.6813

Table A.5: Prognostic group: average Clustering coefficient

Threshold	Distance	Complete graph	Slow subgraph	Normal subgraph	Fast subgraph
0.6	Euclidean	0.0307	0.0813	0.0435	0.0538
	Manhattan	0.0251	0.0705	0.0376	0.0248
	Minkowski	0.0366	0.0852	0.0513	0.0744
	Cosine	0.0286	0.0575	0.0397	0.0588
	Correlation	0.0235	0.0461	0.0316	0.0499
0.7	Euclidean	0.0272	0.0747	0.0397	0.0403
	Manhattan	0.0234	0.0649	0.0327	0.0225
	Minkowski	0.0292	0.0786	0.0404	0.0479
	Cosine	0.0285	0.0574	0.0397	0.0587
	Correlation	0.0225	0.0438	0.0299	0.0474
0.8	Euclidean	0.0237	0.0670	0.0363	0.0236
	Manhattan	0.0222	0.0615	0.0270	0.0168
	Minkowski	0.0262	0.0718	0.0391	0.0368
	Cosine	0.0284	0.0573	0.0397	0.0582
	Correlation	0.0211	0.0416	0.0277	0.0437
0.9	Euclidean	0.0116	0.0338	0.0190	0.0128
	Manhattan	0.0088	0.0262	0.0127	0.0077
	Minkowski	0.0128	0.0372	0.0213	0.0149
	Cosine	0.0280	0.0571	0.0393	0.0562
	Correlation	0.0182	0.0389	0.0241	0.0375

Table A.6: Functional group: average Eigenvector centrality

Threshold	Distance	Complete graph	Slow subgraph	Normal subgraph	Fast subgraph
0.6	Euclidean	0.0017	0.0026	0.0028	0.0102
	Manhattan	0.0028	0.0043	0.0052	0.0005
	Minkowski	0.0010	0.0013	0.0015	0.0062
	Cosine	0.0000	0.0000	0.0000	0.0000
	Correlation	0.0005	0.0020	0.0011	0.0027
0.7	Euclidean	0.0022	0.0040	0.0036	0.0039
	Manhattan	0.0023	0.0051	0.0023	0.0000
	Minkowski	0.0022	0.0030	0.0030	0.0032
	Cosine	0.0000	0.0000	0.0000	0.0001
	Correlation	0.0006	0.0021	0.0013	0.0034
0.8	Euclidean	0.0025	0.0052	0.0047	0.0001
	Manhattan	0.0013	0.0054	0.0012	0.0000
	Minkowski	0.0025	0.0044	0.0044	0.0039
	Cosine	0.0000	0.0000	0.0000	0.0002
	Correlation	0.0007	0.0020	0.0014	0.0051
0.9	Euclidean	0.0004	0.0021	0.0006	0.0000
	Manhattan	0.0000	0.0014	0.0000	0.0000
	Minkowski	0.0007	0.0022	0.0009	0.0009
	Cosine	0.0000	0.0001	0.0001	0.0009
	Correlation	0.0009	0.0021	0.0018	0.0076

Table A.7: Functional group: average Betweenness centrality

Threshold	Distance	Complete graph	Slow subgraph	Normal subgraph	Fast subgraph
0.6	Euclidean	0.3444	0.5427	0.3635	0.2126
	Manhattan	0.1892	0.4155	0.2012	0.0157
	Minkowski	0.4791	0.6419	0.5039	0.4159
	Cosine	0.8030	0.8331	0.7879	0.8038
	Correlation	0.4724	0.5066	0.4496	0.4636
0.7	Euclidean	0.2581	0.4563	0.2788	0.0658
	Manhattan	0.1510	0.3731	0.1494	0.0072
	Minkowski	0.3165	0.5215	0.3417	0.1323
	Cosine	0.7971	0.8273	0.7858	0.7904
	Correlation	0.4445	0.4837	0.4245	0.4066
0.8	Euclidean	0.1739	0.3833	0.1796	0.0111
	Manhattan	0.1052	0.3306	0.1045	0.0039
	Minkowski	0.2339	0.4269	0.2479	0.0579
	Cosine	0.7818	0.8234	0.7763	0.7492
	Correlation	0.4155	0.4654	0.3971	0.3313
0.9	Euclidean	0.0696	0.2443	0.0580	0.0029
	Manhattan	0.0332	0.1458	0.0212	0.0006
	Minkowski	0.0909	0.2781	0.0788	0.0110
	Cosine	0.7279	0.8056	0.7347	0.6453
	Correlation	0.3723	0.4350	0.3486	0.2170

Table A.8: Functional group: average Closeness centrality

Threshold	Distance	Complete graph	Slow subgraph	Normal subgraph	Fast subgraph
0.6	Euclidean	0.5745	0.6504	0.5765	0.4978
	Manhattan	0.5030	0.6434	0.5124	0.3484
	Minkowski	0.6804	0.7465	0.6800	0.6187
	Cosine	0.8963	0.9116	0.8863	0.9018
	Correlation	0.7014	0.7601	0.6912	0.6610
0.7	Euclidean	0.5496	0.6364	0.5561	0.4524
	Manhattan	0.4624	0.6750	0.4491	0.2913
	Minkowski	0.6141	0.7036	0.6171	0.5223
	Cosine	0.8922	0.9079	0.8824	0.8966
	Correlation	0.6888	0.7548	0.6764	0.6455
0.8	Euclidean	0.4935	0.6743	0.4923	0.3245
	Manhattan	0.4191	0.6611	0.4120	0.2057
	Minkowski	0.5549	0.6582	0.5560	0.4546
	Cosine	0.8845	0.9008	0.8754	0.8867
	Correlation	0.6815	0.7498	0.6704	0.6330
0.9	Euclidean	0.3809	0.6048	0.3730	0.1605
	Manhattan	0.4109	0.7049	0.3925	0.1384
	Minkowski	0.4639	0.6263	0.4622	0.2952
	Cosine	0.8564	0.8738	0.8512	0.8491
	Correlation	0.6732	0.7500	0.6637	0.6124

Table A.9: Functional group: average Clustering coefficient

Threshold	Distance	Complete graph	Slow subgraph	Normal subgraph	Fast subgraph
0.6	Euclidean	0.0422	0.0926	0.0577	0.0878
	Manhattan	0.0395	0.0896	0.0543	0.0853
	Minkowski	0.0430	0.0932	0.0588	0.0883
	Cosine	0.0298	0.0594	0.0417	0.0615
	Correlation	0.0254	0.0474	0.0361	0.0539
0.7	Euclidean	0.0388	0.0885	0.0534	0.0835
	Manhattan	0.0385	0.0883	0.0530	0.0786
	Minkowski	0.0389	0.0885	0.0535	0.0846
	Cosine	0.0298	0.0594	0.0417	0.0615
	Correlation	0.0253	0.0474	0.0360	0.0538
0.8	Euclidean	0.0373	0.0868	0.0515	0.0725
	Manhattan	0.0370	0.0865	0.0510	0.0705
	Minkowski	0.0373	0.0868	0.0516	0.0734
	Cosine	0.0298	0.0594	0.0417	0.0611
	Correlation	0.0253	0.0473	0.0359	0.0536
0.9	Euclidean	0.0231	0.0530	0.0318	0.0411
	Manhattan	0.0230	0.0529	0.0318	0.0410
	Minkowski	0.0231	0.0530	0.0319	0.0411
	Cosine	0.0298	0.0594	0.0416	0.0610
	Correlation	0.0250	0.0472	0.0356	0.0529

Table A.10: Respiratory group: average Eigenvector centrality

Threshold	Distance	Complete graph	Slow subgraph	Normal subgraph	Fast subgraph
0.6	Euclidean	0.0008	0.0018	0.0014	0.0044
	Manhattan	0.0010	0.0028	0.0019	0.0059
	Minkowski	0.0006	0.0014	0.0010	0.0035
	Cosine	0.0000	0.0002	0.0001	0.0003
	Correlation	0.0001	0.0004	0.0003	0.0008
0.7	Euclidean	0.0012	0.0035	0.0021	0.0067
	Manhattan	0.0012	0.0038	0.0022	0.0071
	Minkowski	0.0011	0.0035	0.0020	0.0066
	Cosine	0.0000	0.0002	0.0001	0.0003
	Correlation	0.0001	0.0004	0.0003	0.0008
0.8	Euclidean	0.0015	0.0048	0.0028	0.0084
	Manhattan	0.0016	0.0050	0.0029	0.0087
	Minkowski	0.0015	0.0047	0.0027	0.0084
	Cosine	0.0000	0.0002	0.0001	0.0004
	Correlation	0.0002	0.0004	0.0003	0.0009
0.9	Euclidean	0.0010	0.0027	0.0019	0.0050
	Manhattan	0.0010	0.0027	0.0019	0.0051
	Minkowski	0.0010	0.0026	0.0019	0.0050
	Cosine	0.0000	0.0002	0.0001	0.0005
	Correlation	0.0002	0.0005	0.0004	0.0011

Table A.11: Respiratory group: average Betweenness centrality

Threshold	Distance	Complete graph	Slow subgraph	Normal subgraph	Fast subgraph
0.6	Euclidean	0.7231	0.8446	0.7252	0.6714
	Manhattan	0.6603	0.7810	0.6600	0.6015
	Minkowski	0.7747	0.8749	0.7877	0.7228
	Cosine	0.9233	0.9238	0.9181	0.9372
	Correlation	0.4754	0.3692	0.5078	0.5312
0.7	Euclidean	0.6378	0.7392	0.6384	0.5725
	Manhattan	0.6232	0.7234	0.6233	0.5577
	Minkowski	0.6415	0.7431	0.6424	0.5766
	Cosine	0.9224	0.9238	0.9181	0.9321
	Correlation	0.4720	0.3681	0.5036	0.5260
0.8	Euclidean	0.5745	0.6750	0.5717	0.5131
	Manhattan	0.5643	0.6644	0.5611	0.5050
	Minkowski	0.5771	0.6775	0.5747	0.5155
	Cosine	0.9186	0.9238	0.9176	0.9186
	Correlation	0.4653	0.3651	0.4979	0.5134
0.9	Euclidean	0.4802	0.5863	0.4675	0.4222
	Manhattan	0.4794	0.5853	0.4665	0.4216
	Minkowski	0.4805	0.5868	0.4678	0.4223
	Cosine	0.9120	0.9227	0.9103	0.8988
	Correlation	0.4445	0.3562	0.4744	0.4867

Table A.12: Respiratory group: average Closeness centrality

Threshold	Distance	Complete graph	Slow subgraph	Normal subgraph	Fast subgraph
0.6	Euclidean	0.7766	0.8022	0.7700	0.7675
	Manhattan	0.7574	0.7928	0.7522	0.7357
	Minkowski	0.8226	0.8468	0.8155	0.8156
	Cosine	0.9514	0.9544	0.9470	0.9575
	Correlation	0.6944	0.6192	0.7094	0.7418
0.7	Euclidean	0.7559	0.7821	0.7509	0.7424
	Manhattan	0.7122	0.7474	0.7091	0.6862
	Minkowski	0.7677	0.7925	0.7625	0.7561
	Cosine	0.9505	0.9534	0.9462	0.9567
	Correlation	0.6905	0.6161	0.7052	0.7377
0.8	Euclidean	0.6782	0.7040	0.6758	0.6589
	Manhattan	0.6449	0.6769	0.6437	0.6171
	Minkowski	0.6865	0.7122	0.6842	0.6673
	Cosine	0.9508	0.9542	0.9466	0.9564
	Correlation	0.6816	0.6086	0.6956	0.7289
0.9	Euclidean	0.6524	0.7052	0.6471	0.6077
	Manhattan	0.6453	0.6996	0.6394	0.6004
	Minkowski	0.6543	0.7068	0.6491	0.6097
	Cosine	0.9445	0.9480	0.9401	0.9502
	Correlation	0.6508	0.5819	0.6644	0.6946

Table A.13: Respiratory group: average Clustering coefficient