
POLITECNICO DI TORINO

**Master Degree in Mathematical Engineering**

Master Degree Thesis

# Longitudinal radiomic features analysis: evaluating immunotherapy efficacy in glioblastoma murine models

**Supervisors**
Prof. Luigi Preziosi
Dr. Chiara Razzetta
Dr. Sara Garbarino

**Candidate**
Luca Mana

Academic Year 2024-2025

**Abstract**

Glioblastoma multiforme (GBM) is one of the most aggressive and lethal brain tumors, characterized by rapid progression, high heterogeneity, and resistance to standard treatments. Understanding its evolution under different therapeutic strategies is crucial to improving patient outcomes. This thesis focuses on the development of an analysis pipeline for longitudinal radiomics applied to MRI data obtained from a study on immunotherapy treatments for glioblastoma in murine models. Radiomics enables the extraction of quantitative information from medical images, allowing for a more detailed assessment of tumor progression. In this work, MRI scans were acquired weekly for four groups of mice subjected to different therapeutic approaches. The main objective of this study is to exploit radiomics to evaluate disease evolution in relation to treatment response and to assess the consistency of tumor progression trends within each group. To achieve this, we developed models to estimate the ODE governing the evolution of some key radiomic features. By analyzing the trends of the most relevant radiomic features, we aim to determine whether different therapeutic strategies lead to distinct tumor evolution patterns and to quantify the variability of responses within each treatment group solely based on imaging exams.

*To my brother,*

# Acknowledgements

First and foremost, I would like to express my sincere gratitude to my thesis supervisor, Professor Preziosi, for giving me the opportunity to carry out this thesis work at the University of Genoa and, most importantly, for his helpfulness and support throughout my academic journey.
I would also like to sincerely thank Dr. Chiara Razzetta, for her immense patience and continuous support during the development of this project, and Dr. Sara Garbarino, who was always available for any clarification.
Special thanks go to my family and friends, especially my parents and brother, who have always been by my side during this long and challenging journey. There are not enough words to express my gratitude.

# Contents

# List of Figures

# Introduction

Glioblastoma, also known as glioblastoma multiforme or grade IV astrocytoma, is an extremely aggressive form of cancer that affects the central nervous system. It accounts for approximately 45% of brain-originating tumors. According to the ISS (Istituto Superiore di Sanità), it is most commonly diagnosed in individuals between the ages of 45 and 75; however, cases have also been reported in pediatric patients. A higher incidence has been recorded in men compared to women. [1]

This thesis builds upon a research project conducted by the University of Genoa in collaboration with the IRCCS Ospedale Policlinico San Martino in Genoa and the Italian Ministry of Health. The primary objective of this study is to comprehensively investigate glioblastoma progression under different treatment conditions by integrating multimodal data sources. These include MRI-based radiomic features, tumor biopsies, and antibody titration analyses. To achieve this goal, four groups of murine models were injected with the tumor and subjected to different treatment protocols: the first group received no treatment (control group), the second group was treated with an epigenetic drug, the third group received a combination of the same epigenetic drug and an immunotherapy drug, and the fourth group was treated exclusively with immunotherapy. Throughout the study, biological assays were performed to assess the efficacy of these treatments and identify potential side effects, while MRI scans were acquired at regular intervals to monitor tumor progression. Additionally, tumor biopsies will be conducted post-mortem to obtain molecular data for further analysis.

A key focus of this thesis is the establishment of robust analytical pipelines for longitudinal radiomic analysis derived from MRI scans. Radiomics enables the extraction of quantitative imaging biomarkers, allowing for in-depth characterization of tumor evolution. These features include fundamental parameters such as tumor volume, diameter, and morphology, as well as more advanced textural attributes that capture gray-level intensity distributions and structural heterogeneity. By systematically comparing these radiomic descriptors with antibody titration data and biopsy-derived molecular markers, future research aims to determine potential concordances between imaging-based assessments and histopathological findings. This multimodal approach could provide critical insights into tumor dynamics, potentially identifying novel imaging biomarkers that predict response to treatment.

[1] https://www.issalute.it

As a first step in this thesis, we perform some basic *Exploratory Data Analysis (EDA)* techniques. We conducted simple survival studies to establish baseline differences between treatment groups. Following this, *Principal Component Analysis (PCA)* was applied to assess whether the extracted radiomic features could effectively differentiate among the four subject groups.

Since the ultimate goal is to identify specific correlations between imaging and other collected data, we aimed to extract radiomic features that are easily interpretable and analyze their longitudinal trends in both individual subjects and groups to determine whether significant intra- and inter-group differences exist. Longitudinal tracking of these features over time provides valuable insights into the progression of glioblastoma under different treatments. To achieve this, we attempted to model the longitudinal trends of these values using the *Sparse Identification of Nonlinear Dynamical Systems (SINDy)* algorithm. However, as will be discussed later, this approach was not effective for this particular dataset due to the complexity and variability of tumor progression: as a matter of facts, for each subject we collected at most three MRIs. Consequently, we adopted an ad hoc pipeline implementation exploiting *Least Absolute Shrinkage and Selection Operator (LASSO)* regression, which allowed us to approximate the data more effectively. This method facilitated the identification of key radiomic features that exhibit meaningful changes over time, ultimately offering a clearer understanding of the temporal evolution of the studied features and providing deeper insights into the efficacy of the different treatments across experimental groups.

By integrating survival analysis, radiomic feature extraction, and statistical modeling, this study aims to develop a framework for assessing glioblastoma progression in a more holistic manner. The insights gained from this research could contribute to the refinement of imaging-based monitoring techniques, potentially informing future preclinical and clinical investigations on glioblastoma treatment efficacy.

This thesis is structured as follows. In Chapter 1 we introduce the essential mathematical instruments used to build the analysis pipeline. An overview of the key aspects of radiomics and a detailed description of the pipeline implemented are provided in Chapter 2. Finally, in Chapter 3 the biological framework of the experiment will be discussed, along with the obtained results and their corresponding conclusions. The preliminary results of these pipelines will be presented, confirming the concordance with the quantitative imaging evaluation and preliminary survival analysis. These results indicate that groups treated with the epigenetic drug exhibited a stronger immune response compared to the others.

# Chapter 1

# Estimating the evolution of the features along time

The purpose of this chapter is to describe the mathematical background used for the elaboration of the pipeline.

## 1.1 Dataset exploration

In the following Section will be described tools used for dataset exploration.
Kaplan-Meier curve are employed to analyze the probability of survival over time in clinical trials.
Secondly, Principal Component Analysis (PCA) will be presented; in our context, it is used to analyze the variance structure of features extracted from medical images, helping to identify which components best describe the variability in the data and may be relevant for understanding the processes undertaken during the trial.

### 1.1.1 Kaplan-Meier survival curve

Kaplan-Meier survival analysis is a non-parametric statistical method widely used in medical research to estimate the survival probabilities of patients or experimental subjects over time.
It provides crucial insights into disease progression and treatment effectiveness by analyzing time-to-event data, where the event of interest is typically death or disease recurrence. The Kaplan-Meier estimator calculates the probability of surviving in a given time interval while accounting for censored data, which occurs when a subject's outcome is unknown due to loss of follow-up or the study ending before the event occurs.
This method enables researchers to compare survival distributions across different treatment groups and assess median survival times, providing valuable information for evaluating therapeutic interventions

For each time instant, it is possible to evaluate the survival probability [4]; let $n_t$ be the number of subjects living at the beginning of the selected moment of time and $d_t$ the amount of those who died. Then, the probability of surviving at time $t$ is defined as

$$\mathbb{P}_t = \frac{n_t - d_t}{n_t} \tag{1.1}$$

Thus, the survival probability function that indicates the probability of survival in a given time interval, is defined as:

$$S(t) = \prod_{t_i \leq t} \mathbb{P}_{t_i} \tag{1.2}$$

The usual picturing of the computed probabilities consists in constant trait function whose traits are connected by vertical lines indicating the probability drop [12]. In detail, time is represented on the X-axis and can be expressed in various units such as years, months, or days, depending on the duration of the trial. The cumulative survival is represented on the Y-axis, thus it belongs to the interval [0,1] or as in percentage [0%,100%]. Since the probability is estimated, it is possible to evaluate the *confidence interval* (CI) for each survival function. This means that for each time point, the estimated probability is joined by a range, known as the confidence interval, in which the true survival probability is expected to lie.

### 1.1.2 Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a statistical technique used to reduce the dimensionality of high-dimensional datasets while preserving essential information. It achieves this by transforming the original set of variables into a new set of uncorrelated ones, called principal components, which are linear combinations of the original variables.

One of its main applications in data analysis is to obtain a graphical representation of the joint distribution of numerical variables when the number of variables exceeds two. This technique allows researchers to visualize correlations between variables and identify patterns that facilitate data interpretation. The goal is to retain the greatest dispersion of points in a lower-dimensional space, ensuring a more effective classification and understanding of the dataset.

Dimensionality reduction is performed by replacing the original set of variables with a new one called principal components, which capture the maximum variance of the data. This process involves translating the coordinate axes by aligning the new origin with the centroid of the dataset. Subsequently, new axes are determined such that the variance of the data projections onto the first principal component is maximized, followed by the second, and so forth. The principal components are ordered according to the eigenvalues of the covariance (or correlation) matrix, with each component capturing a progressively smaller portion of the total variance.

PCA allows to reduce data complexity while maintaining essential features, making

it a valuable tool in exploratory data analysis.
In survival analysis, principal component analysis can be particularly useful in identifying key features that distinguish different patient groups and contribute to a more comprehensive understanding of disease progression and treatment response. In mathematical terms, defined $\mathbf{X} \in \mathbb{R}^{n \times m}$, matrix with the numerical data, where $m$ are the variables and $n$ are the features. Given a generic row of $\mathbf{X}$, denoted by $\mathbf{x}$, with his variance and covariance matrix $\Sigma$, the purpose is to compute the linear combination

$$\mathbf{z} = \mathbf{a}^T \mathbf{x}$$

such that $Var(\mathbf{z})$ is maximized.
However, the solution is not unique, and therefore it is necessary to restrict the problem with the constraint that $\mathbf{a}$ has unitary norm, $\mathbf{a}^T\mathbf{a} = 1$.
Using the variance property, it can be shown that

$$Var(\mathbf{z}) = Var(\mathbf{a}^T\mathbf{x}) = \mathbf{a}^T Var(\mathbf{x})\mathbf{a} = \mathbf{a}^T\Sigma\mathbf{a}$$

where each component of $\Sigma$ is given by

$$\sigma_{ij} = \frac{1}{n}\sum_{k=1}^{n}(x_{ki} - \mu_i)(x_{kj} - \mu_j)$$

The problem described above, i.e maximizing principal component variance, can be expressed as a maximum problem with constraint

$$\max_{\substack{\mathbf{a} \\ s.t \quad \mathbf{a}^T\mathbf{a}=1}} \mathbf{a}^T\Sigma\mathbf{a}$$

The solution of the problem is given by

$$(\Sigma - \lambda I)\mathbf{a} = \mathbf{0}$$

i.e. the solution is expressed by the orthonormal eigenvector of $\Sigma$.
Set of vectors $\mathbf{a}$ can be considered as the eigenvectors associated to the nonzero eigenvalues $\lambda$ of the matrix $\Sigma$.
The solution of the previous diagonalization is

$$\Sigma = \Gamma\Lambda\Gamma^T$$

where $\Gamma$ is a matrix of dimension $m \times p$, with $p = rank(\Gamma) \leq m$, corresponding to the number of non zero eigenvalues, build from the eigenvector $\mathbf{a}$. $\Lambda$ is a diagonal matrix of dimension $p \times p$; the diagonal of $\Gamma$ contains the $p$ eigenvalues $\lambda$ in ascending order. Therefore, if there are $m$ eigenvalues which are non zero, there are $m$ distinct eigenvectors $\mathbf{z}$, one for each solution; for this reason, $\mathbf{z}$ can be written with the following

formula

$$\mathbf{z} = \Gamma^T \mathbf{x}$$

That implies that the variance of $\mathbf{z}$ can be rewritten as follows

$$Var(\mathbf{z}) = Var(\Gamma^T \mathbf{x}) = \Gamma^T Var(\mathbf{x})\Gamma = \Gamma^T \Gamma \Lambda \Gamma^T \Gamma = \Lambda$$

This brings to the definition of the components; for the first one, recalling the optimization problem and using the fact that the eigenvalues are in ascending order, the best result is the eigenvector $\mathbf{v}$ = (1,0,0,...) since it has to have unitary norm.
As a consequence $\mathbf{a} = \mathbf{u}_1$, where $\mathbf{u}_1$ is the eigenvector associated with the largest eigenvalue.
The other components are defined with the optimization problem, adding an additional condition[10][15]

$$\max_{\substack{\mathbf{a} \\ s.t \quad \mathbf{a}^T\mathbf{a}=1, \quad \mathbf{a}^T\mathbf{v}=0}} \mathbf{a}^T \Sigma \mathbf{a}.$$

In *Python* there are multiple libraries that help with the implementation of PCA. One of the most used is *scikit-learn*; from it, it is possible to import two main functions, *StandardScaler* and *PCA* [9].
The former is used in order to standardized the dataset column by column. It normalizes the matrix, subtracting his mean $\bar{\mathbf{x}}$ and dividing it by his standard deviation $s$

$$\mathbf{z} = \frac{\mathbf{x} - \mathbf{u}}{s}.$$

The latter is applied for the PCA implementation; this method gives the possibility to define a priori the number of components or to evaluate them lately. Indeed, it is possible to set a threshold value, in order to quantify the number of components necessary to reach the established threshold [9].
The threshold value is determined by specifying a predefined level of variance that should be preserved in the dimensionality reduction process

## 1.2 Machine learning based techniques for ODE fitting

In the following Section there will be explained two techniques for ODE fitting: *Lasso regression* and *SINDy*.
These techniques are used in order to solve systems of ODEs like

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}(t)) \tag{1.3}$$

where $\mathbf{x}(t) \in \mathbb{R}^n$ defines the system at time $t$, and $\mathbf{f}(\mathbf{x}(t))$ defines the dynamic restriction that determine the equations of motion of the system.
The function $\mathbf{f}$ usually has few terms due to the sparsity of the physical models.

### 1.2.1 Lasso regression

In statistics and machine learning, a regression analysis refers to a set of statistical methods used to estimate the relationship between a dependent variable and other independent variables.
Given $N$ cases each with an outcome $\mathbf{y} \in \mathbb{R}^N$ and $n$ covariates, $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\} \in \mathbb{R}^{N \times n}$, the aim of Lasso regression is to solve the following minimization problem:

$$\min_{\beta_0, \beta} \left\{ \sum_{i=1}^{N} (y_i - \beta_0 - \mathbf{x_i}^T \beta)^2 \right\} \tag{1.4}$$

constrained by

$$\sum_{j=1}^{n} |\beta_j| \leq t,$$

where $\beta_0$ is a constant coefficient, $\beta \in \mathbb{R}^n$ is the coefficient vector and $t \in \mathbb{R}$ is a constant used to define the degree of penalization.
Using the $\ell^q$ norm definition

$$\|\mathbf{x}\|_q = \left( \sum_{i=1}^{N} |x_i|^q \right)^{1/q}$$

the minimization problem (1.4), can be written as

$$\min_{\beta_0, \beta} \left\{ \|\mathbf{y} - \beta_0 - \mathbf{X}\beta\|_2^2 \right\} \quad \text{subject to} \quad \|\beta\|_1 \leq 1 \tag{1.5}$$

It is feasible to define with $\bar{\mathbf{x}}$ the mean of $\mathbf{x_i}$ and with $\bar{\mathbf{y}}$ the mean of $y_i$; thus, the estimator for $\beta_0$, $\hat{\beta}$, can be defined as $\hat{\beta}_0 = \bar{\mathbf{y}}\text{-}\bar{\mathbf{x}}^T\beta$, and so:

$$y_i - \hat{\beta}_0 - \mathbf{x_i}^T\beta = y_i - (\bar{\mathbf{y}} - \bar{\mathbf{x}}^T\beta) - \mathbf{x_i}^T\beta = (y_i - \bar{\mathbf{y}}) - (\mathbf{x_i} - \bar{\mathbf{x}})^T\beta$$

The previous approximations allows to rewrite (1.5) as:

$$\min_{\beta \in \mathbb{R}^n} \left\{ \frac{1}{N} \|\mathbf{y} - \mathbf{X}\beta\|_2^2 \right\} \quad \text{subject to} \quad \|\beta\|_1 \leq 1 \tag{1.6}$$

that is commonly used in the Lagrangian form:

$$\min_{\beta \in \mathbb{R}^n} \left\{ \frac{1}{N} \|\mathbf{y} - \mathbf{X}\beta\|_2^2 + \lambda \|\beta\|_1 \right\} \tag{1.7}$$

where the parameter $\lambda$ has to be optimized in order to obtain the best value for the coefficient $\beta$ [14].
The penalty term involving $\ell_1$ is the key point of Lasso regression as it favors sparse solutions. The same structured is recalled in another algorithm, SINDy, presented in

the next subsection.

### 1.2.2 SINDy

*Sparse Identification of Nonlinear Dynamics (SINDy)* leverages that most physical systems have few relevant terms used to define the dynamical equations leading to sparse equations in a high-dimensional nonlinear function space.
Recalling Equation 1.3, the aim of the method is to estimate $\mathbf{f}$ close form with a data driven approach. To do so, it is necessary to collect some observations $\mathbf{x}(t)$, that can be features values, covering the role of a sampling of the domain of $\mathbf{f}$. In addition, it is required to have the corresponding values of $\dot{\mathbf{x}(\mathbf{t})}$. This can be either collected as $\mathbf{x}(\mathbf{t})$ or numerically approximated.
In order to solve (1.3), SINDy requires to set a library, with functions that belong to polynomial, trigonometric or constant class $\Theta(\mathbf{x})$. This library must be defined in order to build a base, necessary to approximate data and it has to be multiplied to a coefficient $\beta$, that guaranties the following approximation:

$$\dot{\mathbf{X}} = \Theta(\mathbf{X})\beta \tag{1.8}$$

which has a similar structure as the regression term in Equation (1.7) in Lasso regression. In order to solve the dynamical system, both $\mathbf{x}(\mathbf{t})$ and $\dot{\mathbf{x}}(t)$, have to be collected at several times $t_1, t_2, ..., t_n$ and structured into two matrices as follows:

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}^T(t_1) \\ \mathbf{x}^T(t_2) \\ \vdots \\ \mathbf{x}^T(t_n) \end{bmatrix} = \begin{bmatrix} x_1(t_1) & x_2(t_1) & ... & x_n(t_1) \\ x_1(t_2) & x_2(t_2) & ... & x_n(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ x_1(t_n) & x_2(t_n) & ... & x_n(t_n) \end{bmatrix} \tag{1.9}$$

$$\dot{\mathbf{X}} = \begin{bmatrix} \dot{\mathbf{x}}^T(t_1) \\ \dot{\mathbf{x}}^T(t_2) \\ \vdots \\ \dot{\mathbf{x}}^T(t_n) \end{bmatrix} = \begin{bmatrix} \dot{\mathbf{x}}_1(t_1) & \dot{\mathbf{x}}_2(t_1) & ... & \dot{\mathbf{x}}_n(t_1) \\ \dot{\mathbf{x}}_1(t_2) & \dot{\mathbf{x}}_2(t_2) & ... & \dot{\mathbf{x}}_n(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ \dot{\mathbf{x}}_1(t_n) & \dot{\mathbf{x}}_2(t_n) & ... & \dot{\mathbf{x}}_n(t_n) \end{bmatrix} \tag{1.10}$$

Consequently, $\Theta(\mathbf{x})$, is a matrix too; each column consists of linear or non liner functions.
In order to implement this algorithm, there exist Python package such as *PySindy* [7], that contain some prebuilt library; alternatively, a library of arbitrary functions can be constructed as follows:

$$\Theta(\mathbf{X}) = \begin{bmatrix} \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \\ ... & \alpha & \mathbf{X} & \mathbf{X}^2 & \mathbf{X}^3 & ... & sin(\mathbf{X}) & cos(\mathbf{X}) & ... \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \end{bmatrix} \tag{1.11}$$

where $\alpha \in \mathbb{R}$ and $\mathbf{X}, \mathbf{X}^2, \mathbf{X}^3$ are polynomials and each column is composed by the transformation of the vector X with the corresponding function.
The polynomials with degree higher than 1 can be constructed by the product of multiple linear functions, such as the degree is equal to the chosen one; for instance, given the polynomial function $\mathbf{X}^n$, it can be written as:

$$\mathbf{X}^n = \begin{bmatrix} x_1^n(t_1) & x_1^{n-1}(t_1)x_2(t_1) & ... & x_1(t_1)x_2(t_1)...x_n(t_1) & x_2^n(t_1) & ... & x_k^n(t_1) \\ x_1^n(t_2) & x_1^{n-1}(t_2)x_2(t_2) & ... & x_1(t_2)x_2(t_2)...x_n(t_2) & x_2^n(t_2) & ... & x_k^n(t_2) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ x_1^n(t_m) & x_1^{n-1}(t_m)x_2(t_m) & ... & x_1(t_m)x_2(t_m)...x_n(t_m) & x_2^n(t_m) & ... & x_k^n(t_m) \end{bmatrix}$$

where all the $x^i$ with $i \in [1, \ldots, k]$ are linear functions.
Consequently $\beta$ is a vector with $n$ components

$$\beta = \{\beta_1, \beta_2, \ldots, \beta_n\}$$

Each component is referred to a function of the base $\Theta(\mathbf{x})$.
Since not all base functions are enrolled in the approximation, this vector will regularize their status in the dynamic system, causing the system sparsity.
Feasibly, this approach could be improved with the introduction of the matrix $\mathbf{Z}$, built with i.i.d Gaussian entries with zero mean and noise magnitude $\lambda$:

$$\dot{\mathbf{X}} = \Theta(\mathbf{X})\beta + \lambda\mathbf{Z} \tag{1.12}$$

## 1.3 How to choose the regularization parameter

For each minimization problem, it is crucial to optimize the regularization parameter. Indeed, it is an open question for almost all minimization problems, since it strongly depends on the data analyzed.
In this study, to find the best parameter $\lambda$, in order to define the best coefficient vector $\beta$, *Pareto curve Method* or *Elbow method* was involved.
A point $\mathbf{x}^* \in \mathbf{X}$ is defined as Pareto optimal if and only if there is no other point $\mathbf{x} \in \mathbf{X}$, such that $F_i(\mathbf{x}) \leq F_i(\mathbf{x}^*)$ and $F_{\hat{i}}(\mathbf{x}) < F_{\hat{i}}(\mathbf{x}^*)$, for at least one $\hat{i}$, where $\mathbf{F}$ is vector of functions $\mathbf{F(x)} = \{F_1(x), F_2(x), \ldots, F_k(x)\}$ [8].
The Pareto Front is defined as the set of all points in the parameter space that are Pareto optimal; each point that builds a Pareto Front corresponds to a solution where no others in the parameter space can improve one objective without deteriorating another

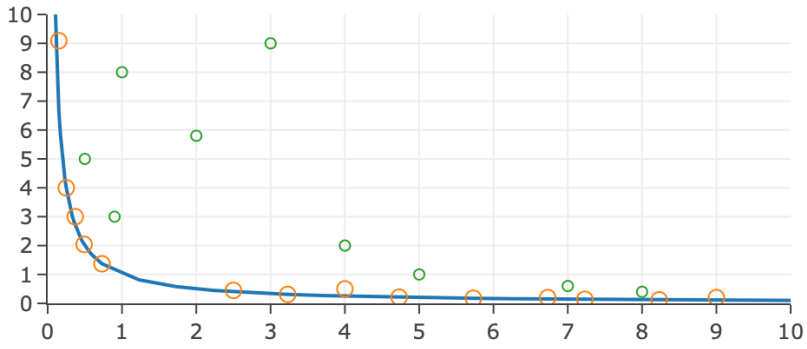


**Figure 1.1:** Picturing of the Pareto front with reference to a generic curve and a random set of points

Figure 1.1 illustrates the functioning of the Pareto frontier in relation to a general curve and a set of points. The set of points that belong to the Pareto Front, and therefore are Pareto optimal, are colored in orange.
In the case study that will be presented, it was not used the entire set of the Pareto Front, since it is supposed continuous and therefore infinite.
Indeed, this method was employed by estimating some values of the front for each fitted problem. Generally, since the problem that has to be solved is in the form 1.7, the two involved norms are represented on the axis. On the X-axis $\|\beta\|_1$ and on the Y-axis $\|\mathbf{y} - \mathbf{X}\beta\|_2^2$.
Hence, the optimal $\lambda$, and consequently the best $\beta$ vector, is obtained when the distance from the origin (0,0) is minimum.

# Chapter 2

# Design and implementation of analysis pipelines

The present chapter explores the fundamental principles of radiomics, which are essential for any study involving medical imaging.
Subsequently, the chapter presents the analytical pipeline, outlining the key steps undertaken in the study. To provide a clearer overview, a schematic representation is given in Figure 2.1, summarizing the sequential steps of the entire methodology.

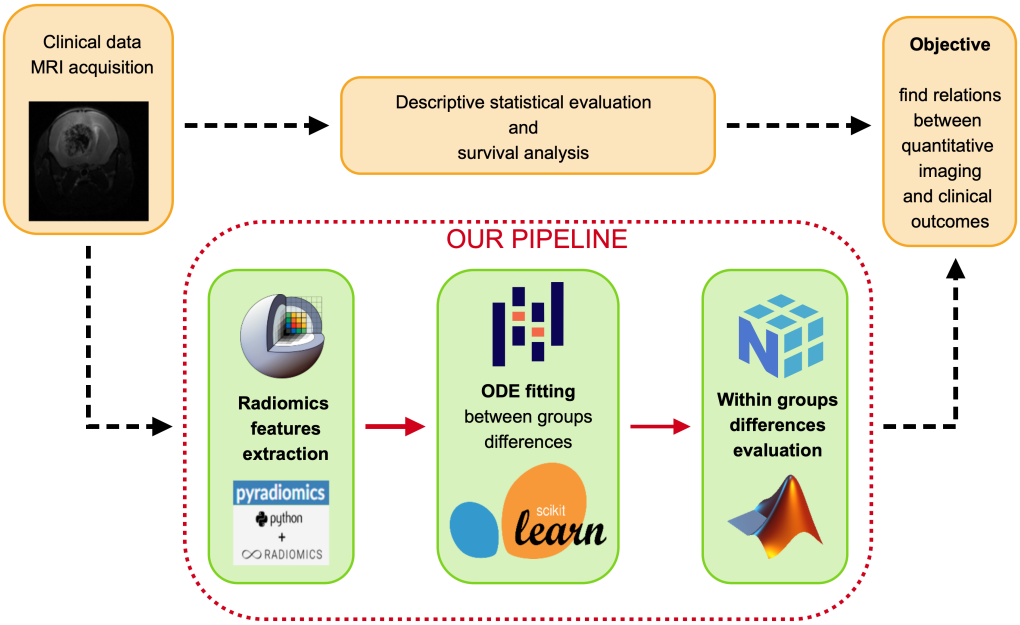


**Figure 2.1:** Schematic overview of the analysis pipeline

## 2.1 Radiomics features extraction

Radiomics is an emerging discipline of medicine and oncology that uses machine learning and artificial intelligence to analyze and extract quantitative data from medical images, including CT scans, MRI and PET.
In order to develop the radiomic pipeline extraction, three steps are required:

1. **Image segmentation**
   *Image segmentation* is widely used across various fields of research. It refers to the process of identifying and extracting some specific regions from an image. Some of the extracted regions are usually called Region of Interest (ROI), because they indicate the areas subject to the subsequent analysis. When dealing with three-dimensional images, the equivalent is the Volume of Interest (VOI). The most common shapes for ROIs and VOIs are circular and spherical respectively but in many case a manual definition is mandatory.
   Image segmentation can be performed manually, semi-automatically, or fully automatically. The first two methods are the most commonly used; however, they come with several limitations, such as being time-consuming and prone to human error.
   There is no single approach to performing image segmentation. Various methods have been developed, such as clustering techniques, graph-based approaches, random walks, and deep learning models, for instance Convolutional Neural Networks (CNNs). These represent just a few of the many strategies that researchers can adopt, depending on the specific requirements of their applications [19].

2. **Image processing**
   In the second step, attention is given to homogenize images used for the evaluation of radiomic features. In this step, interpolation to isotropic voxel spacing is commonly used for almost all features sets, in order to generalize them for multiple datasets. Secondly, range segmentation and intensity outlier filtering are performed. The aim of that process is to remove pixels or voxels from the ROI or VOI that fall outside of a particular range of gray level [20]. The final stage of image processing is the discretization of the image intensity inside the ROI/VOI. It consists in clustering the original values following a particular range intervals.

3. **Feature extraction**
   This step involves evaluating the features within the Region of Interest (ROI) or Volume of Interest (VOI). There are various types of radiomic features; the most commonly used include intensity-based (histogram) features, shape features, texture features, transform-based features, and radial features.
   These features are typically classified as *original*; however, they can also be analyzed after applying specific filters, such as wavelet and logarithmic transformations, to enhance particular characteristics and extract additional information.

These filters are not always helpful. In fact, their usefulness also depends on the type of image studied and the type of analysis performed.

Once these three steps are completed, the output is a matrix where the number of rows corresponds to the number of the image series and the number of columns corresponds to the extracted features.
In Figure 2.2 a MRI obtained during the trial is presented . The tumor zone, which is outlined in green, will be the center of the region of interest (ROI) studied in the radiomics pipeline.

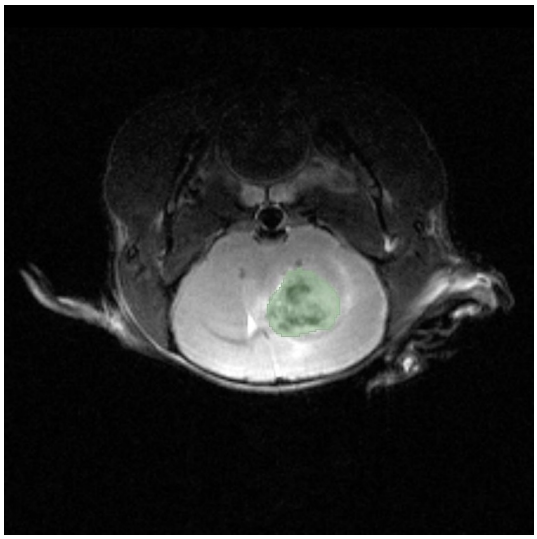

**Figure 2.2:** MRI of mouse with glioblastoma multiforme. The tumor zone, which will later be the center of the ROI studied during radiomics, is highlighted in green.

Furthermore, the number of extractable features might be extremely high. For this reason, dimensionality reduction techniques are often applied before implementing machine learning or deep learning techniques.
In the following case study, radiomic feature extraction was performed using *PyRadiomics* [16] Python package.

## 2.2 Pipeline for fitting ODE models to longitudinal radiomic data

Initially, the Exploratory Data Analysis (EDA) was crucial to understand the dataset obtained from radiomics. From now on we will generally refer to a study in which subjects are divided in different groups

The purpose of the pipeline is to establish a method for assessing the temporal evolution of radiomic feature analysis. In our application contest, it is important to note that each subject has its own set of values over time, but the number of observations for each subject varies depending on the group to which it belongs, as this is linked to the survival rate.

Given that the number of observations differs for each mouse and, in some cases, is fewer than three, the decision was made to evaluate the behavior of the groups, rather than focusing on individual mice.

To achieve this, a new dataset was constructed, containing the average value for each time point. Since the experiment involved three distinct cohorts of subjects and the time intervals between MRIs were not uniform, the inoculation day was set as day 0 for all three cohorts.

Once the new dataset was established, the aim was to investigate the temporal evolution of feature values by fitting an Ordinary Differential Equation (ODE) for each group and each feature using Lasso regression. Initially, an attempt was made to implement SINDy using Python library *PySINDy* [3][7]. However, this algorithm requires a larger number of time points, which made it unsuitable for our dataset.

For that reason it was decided to manually implement Lasso regression. Recalling Section 1.2.1; the following steps were performed

1. **Base definition**: it was necessary to define a library, as in SINDy, used to approximate data. Due to the large number or features involved, a base with ten column was used, constructed as:

$$\Theta = \begin{bmatrix} | & | & | & | & | & | & | & | & | & | \\ \mathbf{x} & \mathbf{x}^2 & \mathbf{x}^3 & \mathbf{x}^4 & \sin(\mathbf{x}) & \cos(\mathbf{x}) & \sin(2\mathbf{x}) & \cos(2\mathbf{x}) & \sin(3\mathbf{x}) & \cos(3\mathbf{x}) \\ | & | & | & | & | & | & | & | & | & | \end{bmatrix} \tag{2.1}$$

2. **Derivative evaluation**: as the derivative of the observations are not in the dataset, we used used two function of the Python library *SciPy* [18] (*splrep* and *splev*) to retrieve $\dot{\mathbf{x}}(t)$ values.

   The former, requires as input two sets of nodes $\mathbf{x}$,$\mathbf{y}$ and as output gives the vector of knots, the B-spline coefficients and the degree of the spline. That output, with the vector $\mathbf{x}$, are used as input for the latter function, that evaluate the derivative of the given spline.

The combination of these two functions gives the left hand side of the ode: $\dot{\mathbf{X}}$. Python code is provided below:

```
from scipy import interpolate

tck = interpolate.splrep(t, X, s=5)
X_dot = interpolate.splev(t, tck, der=1)
```

3. **Lasso Regeression**: Lasso function of the library *sklearn.linear_model* [9] has been used. Firstly, it is necessary to define the model, choosing the parameter $\lambda$, that is given as input; thereafter, the model has to be fitted, and it requires y ($\dot{\mathbf{X}}$) and $\mathbf{X}$ ($\Theta$), obtaining the following minimization problem:

$$\min_{\beta \in \mathbb{R}^n} \left\{ \frac{1}{N} \|\dot{\mathbf{X}} - \Theta\beta\|_2^2 + \lambda \|\beta\|_1 \right\}$$

Python code is provided as below:

```
from import sklearn.linear_model as lm

#This function use alpha insted of gamma, c belong to R
model = lm.Lasso(alpha=c)
model.fit(Theta, X_dot)
```

4. **Approximation definition**: Lasso model returns the coefficients $\beta$, that ensure the solution to the minimization problem. To obtain them it was used model.coef_, which recalls the model defined and fitted in the previous step. It gives as output a vector containing the $n$ values of $\beta$.
Additionally, model.intercept_ gives the independent coefficients, previously denoted as $\beta_0$.
With this two set of parameter it is possible to compute an approximation of $\dot{\mathbf{X}}$:

$$\tilde{\mathbf{X}} = \Theta\beta + \beta_0 = \Theta * \text{model.coef\_} + \text{model.intercept\_} \tag{2.2}$$

The main structure of the pipeline has been described; however, some enhancements were necessary.
Lasso regression was selected over other methods because the aim of regularization was to emphasize the sparsity of the matrix that contains all the $\beta$s. In fact, the penalty function

$$\lambda \sum_{i=1}^{n} |\beta_i| = \lambda \|\beta\|_1$$

which is based on the $l_1$ norm, enhances sparsity more effectively compared to other $l_q$ norms with $q \geq 2$ .

The studied dataset has few observations with respect the number of features, especially considering the group subdivision. This has led to a high degree of data variability; in fact, the metrics exhibit different orders of magnitude, which made it difficult to identify a unique parameter $\lambda$ for each considered feature. To address the parameter choice issue, *Pareto curve method*, described in the Section 1.3, has been applied.
In detail, the points of the front are defined as the couples $(\|\beta\|_1, \|\tilde{\mathbf{X}} - \dot{\mathbf{X}}\|_2^2)$ for each feature and each value of parameter. Thus, a set of $\lambda$ has been defined for each feature to be modeled. Subsequently for each feature and for each value of $\lambda$, the model is fitted and a point of the front is determined. Then, the optimal value of $\lambda$ is chosen by computing the following distance for each point

$$\mathrm{d} = \sqrt{(\|\tilde{\mathbf{X}} - \dot{\mathbf{X}}\|_2^2 - 0)^2 + (\|\beta\|_1 - 0)^2)} = \sqrt{(\|\tilde{\mathbf{X}} - \dot{\mathbf{X}}\|_2^2)^2 + (\|\beta\|_1)^2} \tag{2.3}$$

and selecting the minima one. Taking into account the features have different orders of magnitude, three intervals for sampling possible $\lambda$ values are defined:

- $I_1 = [10^{-5}, 10^{3}]$;
- $I_2 = [10^{-8}, 10^{-3}]$
- $I_3 = [10^{-16}, 10^{-8}]$

This division was necessary because using interval $I_1$ for features with lower order of magnitude, lead to unacceptable fit of the model, giving trivial or null solutions. In order to avoid having to define too wide intervals increasing significantly the computational time, the partition into 3 intervals was used.
Each interval was then sampled into 5000 equispaced point; for each feature, $I_1$ was tested first, with a *for loop*, defining the model and fitting it with $\lambda_i$ with $i = 1 : 5000$; for each iteration the distance defined in (2.3) was evaluated and for $i \geq 2$. If the distance is lower than the previous one, the best $\lambda$ value was updated.
If, at the end of the *for loop*, the $\beta$ vector corresponding to the best value of $\lambda$ is null, the same algorithm is implemented with $I_2$ and later on, if it was given the same output, the same process was done with $I_3$.
The following is the applied Python:

```
import numpy as np
from numpy import linalg as LA
import sklearn.linear_model as lm

def def_parameter(X,t,Theta,a,b):
    #a and b are the extremes of the interval
    tck = interpolate.splrep(t, X, s=5)
    X_dot = interpolate.splev(t, tck, der=1)
    Lambda = np.linspace(a,b,5000)
    for i in range(Lambda.shape[0]):
```

```
            model = lm.Lasso(alpha=Lambda[i])
            model.fit(Theta, X_dot)
            X_tilde = Theta@model.coef_ + model.intercept_
            asseY = LA.norm(X_tilde-X_dot,2)**2
            asseX = LA.norm(model.coef_,1)
            dist = math.sqrt((asseX[i]**2)+(asseY[i]**2))
            if i == 0:
                beta = model.coef_
                intercept =  model.intercept_
                opt_dist = dist
                opt_alpha = Alpha
            elif i > 0 and opt_dist>dist:
                beta = model.coef_
                intercept = model.intercept_
                opt_dist = dist
                opt_alpha = Alpha
        return (c,opt_alpha)
```

```
[beta,best_lambda] = def_parameter(X,t,1e-5,1e3)
if sum(beta>1e-10) == 0:
    [beta,best_lambda] = def_parameter(X,t,1e-8,1e-3)
if sum(beta>1e-10) == 0:
    [beta,best_lambda] = def_parameter(X,t,1e-16,1e-8)
```

As shown in the Python code, not only the cases when the vector of coefficients is null were discarded, but also those in which all $\beta_i$, with $i = 1 : n$, are lower than $10^{-10}$, leading to the next interval. This is because it was observed that a coefficient vector where all values have an order of magnitude smaller than $10^{-10}$, yet are nonzero, does not properly approximate the studied curve and can be approximately considered trivial .

As previously mentioned, the features had different order of magnitude, leading to difficulties in visually representing the results, thus complicating the comparison among groups and features.

To easy the clinical interpretation of the results, a normalization of the data with respect to the norm of the vector containing the analyzed feature value was used. In this way, all the values belong to the range $[-1,1]$ and can be compared more easily.

A further analysis was done by comparing the *Week_Dataset* with *Onset_Dataset* and *Last_Dataset*; the former contains the values extracted from the first tumor image of each subject, while the latter includes the values from the last image before death.

In cases where a mouse has only one image, the two datasets will display identical values in the corresponding row.

These dataset will be represented as dots overlapped on the curves generated by $\dot{\mathbf{X}}$ and $\tilde{\mathbf{X}}$. These scatter plot also need to be normalized, and to do so the norm of the vector containing the value of the corresponding feature in the *Week_Dataset* was

used.
Python code for the normalization is provided as below:

```
import numpy as np
from numpy import linalg as LA

def norm_feature (feature_vect):
    vect_norm = LA.norm(feature_vect,2)
    return feature_vect/vect_norm

def norm_scatter (feature_vect, scatter_value):
    vect_norm = LA.norm(feature_vect,2)
    return scatter_value/vect_norm
```

At the end, we evaluate the intra-group variance by computing the distance between the interpolated curve $\tilde{\mathbf{X}}$ and the values present in datasets *Onset_Dataset* and *Last_Dataset* separately.
In order to do so Matlab symbolic computation is adopted.
Using the **syms** tool, a generic $x$ was defined and consequently the base $\Theta$ was set as explained in (2.1).
Since $\tilde{\mathbf{X}}$ has been defined through the choice of $\Theta$ (2.2) and the fit of $\beta$, it is possible to define the distance as:

$$\text{dist} = \sqrt{(x - P_x)^2 + (g(x) - P_y)^2} \tag{2.4}$$

where $g(x)$ is the symbolic $\tilde{\mathbf{X}}$ and $(P_x, P_y)$ are the coordinates of the generic points in the dataset *Onset_Dataset* or *Last_Dataset*.
However, in order to find the minimum distance, it was necessary to solve the minimization problem:

$$\min_x \sqrt{(x - P_x)^2 + (g(x) - P_y)^2} \tag{2.5}$$

where the set of points that minimize the distance are given by solving:

$$\frac{d\left[\sqrt{(x - P_x)^2 + (g(x) - P_y)^2}\right]}{dx} = 0 \tag{2.6}$$

By differentiation, the following equation is obtained:

$$\frac{d\left[\sqrt{(x - P_x)^2 + (g(x) - P_y)^2}\right]}{dx} = \frac{[2(x - P_x) + 2(g(x) - P_y)g'(x)]}{2\sqrt{(x - P_x)^2 + (g(x) - P_y)^2}} = 0$$

$$\Rightarrow (x - P_x) + (g(x) - P_y)g'(x) = 0 \tag{2.7}$$

Once find the set of point $(x, g(x))$ that nullify Equation (2.7); these were replaced

in the Equation (2.5), allowing the minimum distance to be found. Thus the corresponding orthogonal projection of the point $(P_x, P_y)$ onto the curve generated by the interpolation of the vector $\tilde{\mathbf{X}}$ is given.
Once the distances were computed using the MATLAB pipeline, they were re-imported into Python to analyze their trends across different groups.
For each group, the mean was calculated for every relevant feature. In fact, since that kind of symbolic computation is highly demanding in terms of computational resources, a selection of the most significant features was made. These key features, chosen for their greater relevance, will be presented in the following chapter with the corresponding results. Figure 2.3 illustrates the fundamental concept used for distance computation.

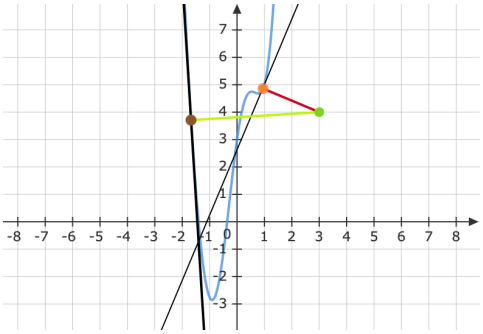


**Figure 2.3:** Picturing of how the distance is computed. The light blue curve is the symbolic representation of $\tilde{\mathbf{X}}$; the red and green segments are the distances. The red one is the lower and chosen one.

The light blue curve is the symbolic representation of $\tilde{\mathbf{X}}$. The light green point corresponds to a data point from *Onset_Dataset* or *Last_Dataset*. Orange and brown dots are its orthogonal projections and the red and light green segments are the relative distances between these points and the light green one.
The red distance is provided as output by the MATLAB pipeline, since it is the smaller.

The following code contains the procedure for distance computing (the data import and export sections have been omitted). At lines 20 and 35 it can be observed that a check was done on the solutions found; this is caused by the fact that, given the complexity of the function $g(x)$, due to the chosen base $\Theta$, some solutions might belong to the complex space $\mathbb{C}$. These solutions were automatically discarded.

```
syms x
base=[x;x^2;x^3;x^4;sin(2*x);cos(2*x);...
                          ...sin(x);cos(x);sin(3*x);cos(3*x)];
g = coef*base+intercept;
gd = diff(g,x);
%b_values = vector containing data extracted from Onset_Dataset
%l_values = vector containing data extracted from Last_Dataset
Theta_o = double(subs(base,x,table2array(b_values(1))));
Theta_l = double(subs(base,x,table2array(l_values(1))));
P_o = [onset_time(k),(coef(j,:)*(Theta_o))+intercept(j)];
P_l = [last_time(k),(coef(j,:)*(Theta_l))+intercept(j)];
dist_onset = ((x-P_o(1))^2+(g-P_o(2))^2)^(0.5);
dist_onset_min = (x-P_o(1))+(g-P_o(2))*gd;
dist_last = ((x-P_l(1))^2+(g-P_l(2))^2)^(0.5);
dist_last_min = (x-P_l(1))+(g-P_l(2))*gd;
sol_onset = double(solve(dist_onset_min,x));
sol_last = double(solve(dist_last_min,x));
for p = 1:length(sol_onset)
    if imag(sol_onset(p)) == 0
        eval_o = double(subs(dist_onset,x,sol_onset(p)));
        if p == 1
            opt_dist_onset = eval_o;
            y_onset = double(subs(g,x,sol_onset(p)));
            x_onset = double(sol_onset(p));
        elseif eval_o < opt_dist_onset
            opt_dist_onset = eval_o;
            y_onset = double(subs(g,x,sol_onset(p)));
            x_onset = double(sol_onset(p));
        end
    end
end
PO_onset = [x_o,y_o];
for p = 1:length(sol_last)
    if imag(sol_last(p)) == 0
        eval_l = double(subs(dist_last,x,sol_last(p)));
        if p == 1
            opt_dist_last = eval_l;
            y_last = double(subs(g,x,sol_last(p)));
            x_last = double(sol_last(p));
        elseif eval_l < opt_dist_last
            opt_dist_last = eval_l;
            y_last = double(subs(g,x,sol_last(p)));
            x_last = double(sol_last(p));
        end
    end
end
PO_last = [x_last,y_last];
```

# Chapter 3

# Biological settings and results

Glioblastoma is one of the most aggressive brain tumors, characterized histologically by necrosis and/or microvascular proliferation. Nowadays, with molecular studies, it is possible to make diagnosis of Isocitrate dehydrogenase wildtype (IDH-wildtype) GBM on the basis of mutations alone. In addition, a very important marker in the prognostic (but not diagnostic) definition of glioblastoma is methylation of the O6-methylguanine-DNA methyltransferase (MGMT) gene promoter, which encodes an enzyme responsible for DNA repair. After methylation, the MGMT gene is silenced, thus increasing drug efficacy.
The experiment was carried three times, with three groups of 32 mice. These subjects were eight-week-old **C57black/6J** mice, housed in pathogen-free colony [1].
Subjects were injected with tumor cells in their brain (from now on that day will be called *baseline*) and they underwent MRI after approximately 20 days after the baseline.
After the injection mice were divided in four groups, one per therapeutic process. However, some mice died before the first MRI, and were discarded from the trial.
Comprehensively 60 mice were analyzed, divided as follows:

- **A**: it is the control group, no treatment has been used. It consist of 16 mice;
- **B**: it is composed of 15 mice, treated with an epigenetic drug;
- **C**: it is composed of 15 mice. They receive both epigenetic drug and immunotherapy;
- **D**: it is the smallest group, with 14 mice, treated with immunotherapy only.

After the first MRI, another exam is performed once a week to monitor the development of the tumor. At the end of the experiment, almost all mice died due to the tumor; the few who survived were sacrificed.

In this chapter we describe some preliminary results obtained by processing the tumoral MRI images acquired during the trial.
The trial timeline is summarized in Figure 3.1.

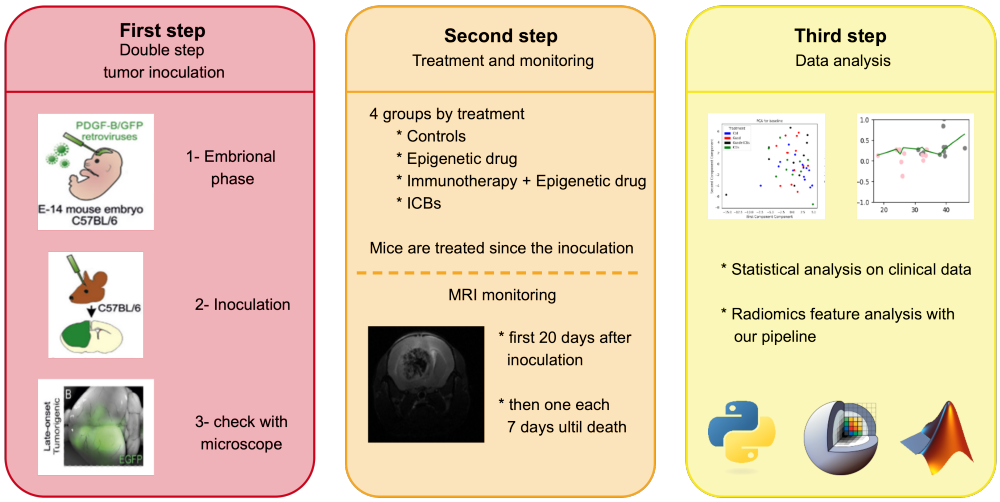


**Figure 3.1:** Principal steps of the trial: Tumor cell inoculation and tumor growth; four groups treatment with two different drugs; Data analysis with Python and Matlab

## 3.1 Dataset exploration

In this Section are reported the first descriptive results on radiomics data extracted from MRI scans.
Since performing MRI on mice is a complex and time-consuming process—requiring anesthesia followed by the imaging procedure some mice underwent MRI one or two days before or after the preset seven-day interval. Therefore, although MRIs were performed weekly, they were not always exactly seven days apart.
For this reason, we added a column to the dataset containing the difference between the date of the MRI and the baseline date for each performed MRI. We take in account three radiomics datasets:

- Onset_Dataset: it contains data from the first MRI for each mouse;
- Last_Dataset: it contains data from the last MRI for each mouse
- Main_Dataset: it contains data from all the MRI for each mouse

### 3.1.1 Survival analysis

Firstly, survival rate among the groups has been studied using Kaplan-Meier curve, set with a confidence interval of 0.1. Results are reported in Figures 3.2 and 3.3.

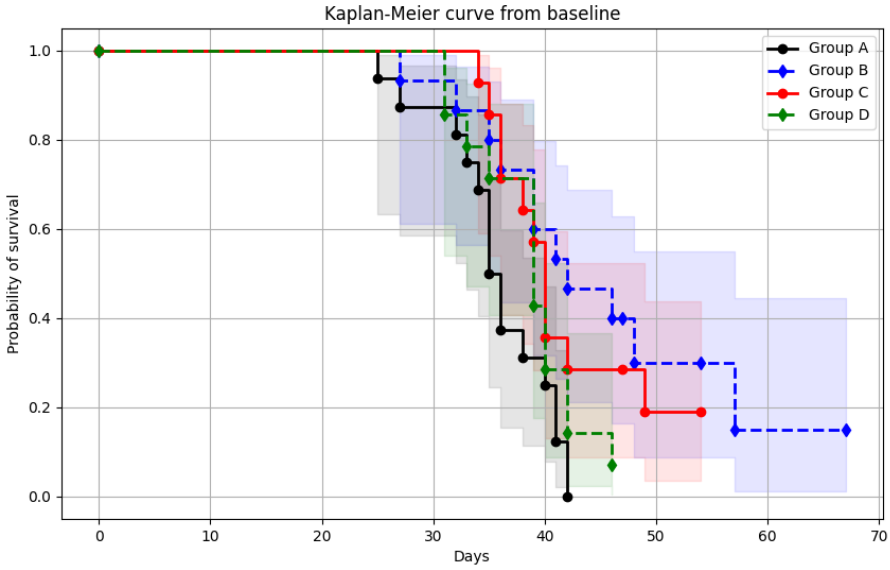


**Figure 3.2:** Kaplan-Meier curve from baseline day to the end of the trial. Each group has been reported according to the legend. A: control group; B: epigenetic drug; C: epigenetic drug and immunotherapy; D: immunotherapy

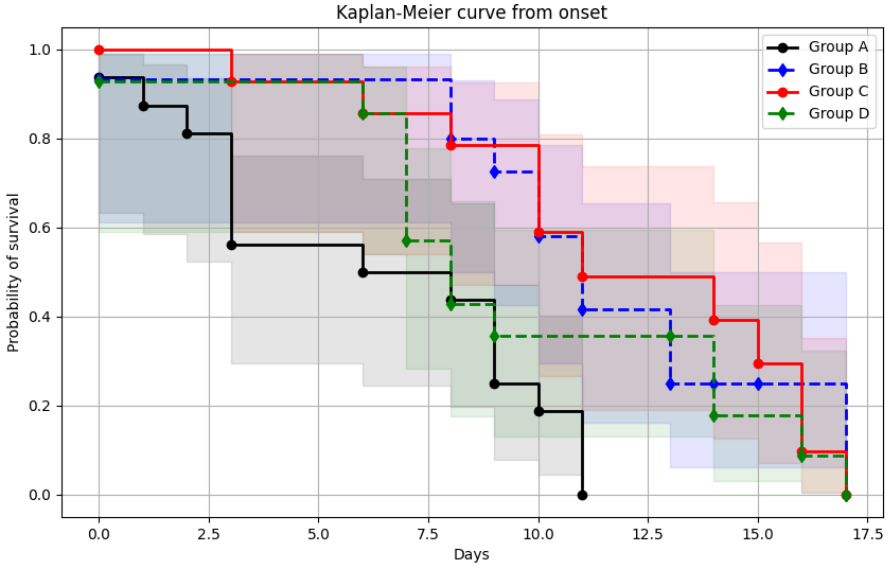


**Figure 3.3:** Kaplan-Meier curve from onset day to the end of the trial. Each group has been reported according to the legend. A: control group; B: epigenetic drug; C: epigenetic drug and immunotherapy; D: immunotherapy

It is remarkable that the two simulations present different behavior for each curve as the probability of survival at instant $t$ is influenced by the previous values for $t_i \leq t$ (Section 1.1.1).

Figure 3.2 represents the effective survival from the baseline. It can been observed

that, for groups B and C, the probability at the last time point has approximately the same value, which is not zero. Instead, the control group A ends with no possibility of survival, and group C has a similar behavior.
In Figure 3.3, which represents the survival probability taking the tumor onset as the first time point for each mouse, we can observe a different behavior. Group C is the only one that starts with the maximum probability, instead, in according to the plot, the other three at the onset has less possibility to survive. However, group C and B has a similar trend for all the simulation, while D after one week deviates from the other lines.
The combination of these results led to the conclusion that groups B and C start the tumor evolution lately compared to A and D; accordingly, further analysis is needed to understand whether a correlation can be found between this result and the composition of the GBM.
This observation is one of the main reasons for undertaking a radiomic analysis to determine whether MRI data can be used to describe tumor growth and its response to different treatments.

### 3.1.2 Radiomics extraction and dataset evaluation

Using the Python package *PyRadiomics* we extracted all the 1130 features. A list of all the original ones can be found in Appendix A; the total amount of 1130 features is obtained by extracting the same features with logarithmic and wavelet filters applied to the image. Due to the wide number of extrapolated features, PCA is employed to understand if we can reduce the dimensionality of the database by exploiting correlation between features.
In Figure 3.4, it is shown the results of the PCA respectively for *Onset_Dataset*, *Last_Dataset* and *Main_Dataset*, with their matrix of correlation related to the extrapolated features.
Since the amount of features is extremely high, it is easier to find group of features correlated with each other. In this case study, despite the implementation of the PCA, pictured with the principal and the second component, no set of features can be found that can be said to be uncorrelated.
This could lead to the conclusion that radiomics is not useful in order to differentiate groups and so that it might not fit for dealing with these kinds of problems.
However, since the main goal of the experiment is to observe and study the evolution over time an attempt was made by studying only the interpretable features and see if there are longitudinal differences in the values.

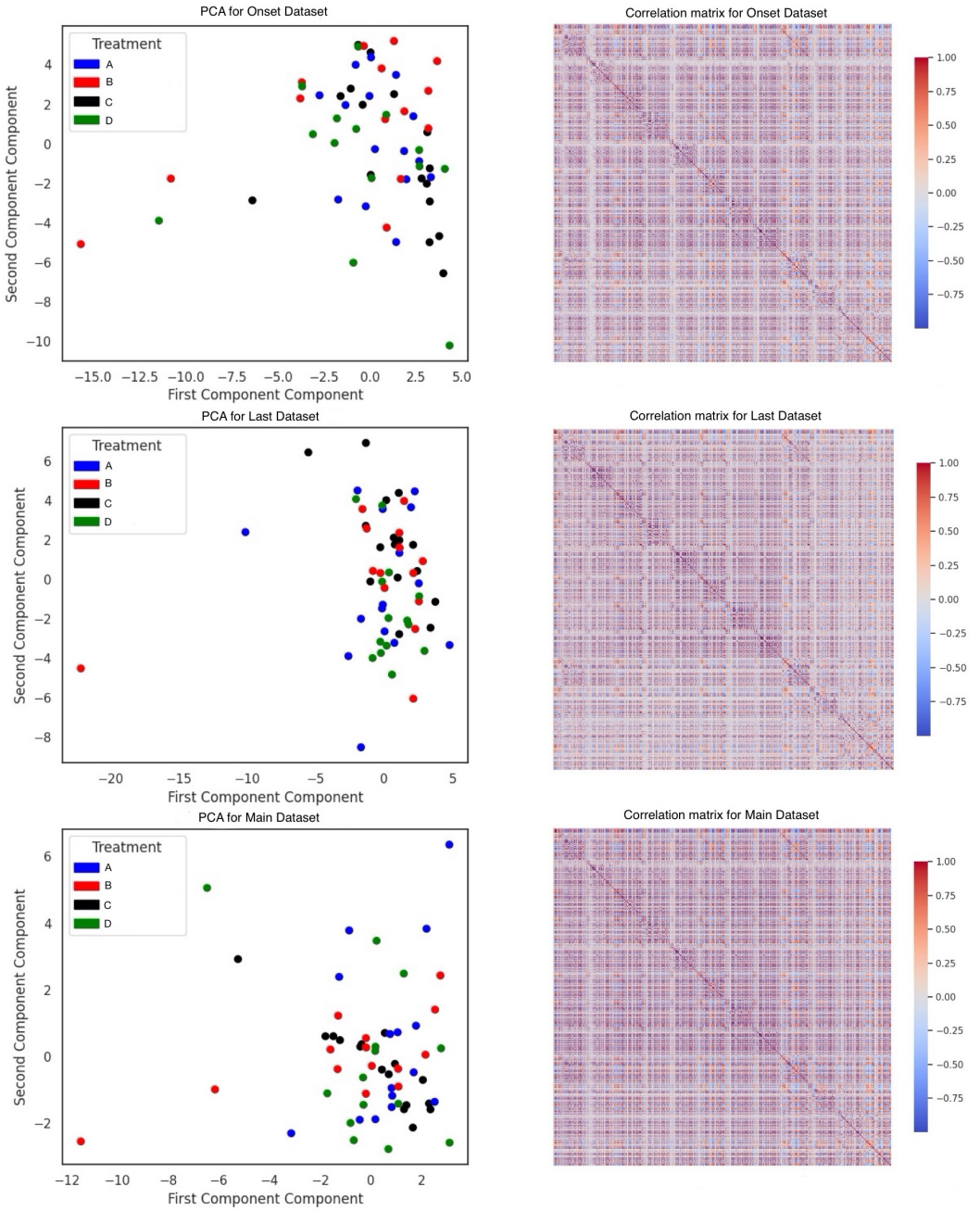


**Figure 3.4:** PCA analysis and Correlation matrix for the three datasets: Onset Dataset, Last Dataset and Main Dataset

## 3.2 Longitudinal radiomic features analysis

In this Section the main selected features, in their original form, are presented along with the performed preprocessing to fit properly the corresponding ODEs.

### 3.2.1 Features selection

Among all the interpretable features, we selected the ones we consider the most relevant in describing the texture of the tumor in MRI:

- First Order
  - *$10^{th}$ and $90^{th}$ Percentile*: these features return, respectively, the $10^{th}$ and the $90^{th}$ percentile of the studied image. In medical imaging, they represent, respectively, the amount of black and white pixel. The former color goes to indicate the presence of blood or dead tissue, which has begun, or is already in a state of necrosis. The latter, represents that in the analyzed zone cysts or other kind of fluid collections there were detected.
  - *Mean*: this features return the mean of the matrix I that contains values of the pixel in the ROI. So, the results is given by:

    $$\bar{I} = Mean = \frac{1}{N}\sum_{i=1}^{N} I(i) \quad \text{where} \quad I \quad \text{is a set of } N \text{ voxels included in the ROI}$$

    In our case a change of this value should indicate a change in the composition of the tumor.
  - *Kurtosis*: this features is computed as follows:

    $$Kurtosis = \frac{\frac{1}{N}\sum_{i=1}^{N}\left(I(i)-\bar{I}\right)^4}{\left(\frac{1}{N}\sum_{i=1}^{N}\left(I(i)-\bar{I}\right)^2\right)^2}$$

    where $N$ is the number of voxels in the ROI and $\bar{I}$ is their mean. It represents how far the values deviates from the mean. A higher value of Kurtosis means that values are concentrated around the mean value, otherwise they are sparser.
  - *Skewness*: this feature represents values asymmetry with reference to the mean value. It is computed as:

    $$Skewness = \frac{\frac{1}{N}\sum_{i=1}^{N}\left(I(i)-\bar{I}\right)^3}{\left(\sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(I(i)-\bar{I}\right)^2}\right)^3}$$

  - *Uniformity*: it represent the homogeneity of the image, and it is a useful marker for the tissue uniformity. It is calculated as

    $$Uniformity = \sum_{i=1}^{N} J(i)^2$$

where $J(i)$s are the normalized probability of intensity level $i$. It indicates how frequently a specific intensity level appears in the ROI.

- Shape 2D
  - *Elongation*: in medical imaging values around 1 represents that shape of the studied ROI, in this case the tumor, is closer to a circle.
    This features returns the square root of the ratio between the length of the smallest and largest principal component, which respectively refers to the smallest and larger eigenvalue of the covariance matrix
    $$Elongation = \sqrt{\frac{\lambda_{min}}{\lambda_{max}}}$$
  - *SurfaceArea*: this features returns the value of the area, calculated as the sum of the area of all the triangle that compose the mesh used to extract the ROI. It helps to understand the shape, and consequently the complexity of the tumor.
- GLCM
  Define $P \in \mathbb{R}^{M\times M}$ the co-occurrence matrix, which is a square matrix that describes the second-order joint probability function of a ROI constrained by a mask and $p \in \mathbb{R}^{M\times M}$ its normalization.
  Called $\mu_x$, $\mu_y$, $\sigma_x$ and $\sigma_y$ respectively the mean and the standard deviation of the marginal row and column probability. Then, we consider:
  - *Autocorrelation*: it is a indicator of how pixels are correlated. Indeed, it helps to understand image homogeneity and it is evaluated as
    $$Autocorrelation = \sum_{i=1}^{M}\sum_{j=1}^{M} p(i,j)ij$$
  - *Cluster Shade*: represents pixel discrepancy from the mean value; a higher value implies greater asymmetry about the mean. It is evaluated as follows
    $$Cluster\_Shade = \sum_{i=1}^{M}\sum_{j=1}^{M} (i+j-\mu_x-\mu_y)^3\, p(i,j)$$
  - *Cluster Tendency*: it is computed as
    $$Cluster\_Tendency = \sum_{i=1}^{M}\sum_{j=1}^{M} (i+j-\mu_x-\mu_y)^2\, p(i,j)$$
    This feature shows the extent to which voxels are grouped with similar gray scales.

– *Contrast*: It quantifies the variation in intensity between adjacent voxel.

$$Contrast = \sum_{i=1}^{M} \sum_{j=1}^{M} (i-j)^2 \, p(i,j)$$

Lower values mean an higher homogeneity.

– *Correlation*:

$$Correlation = \frac{\sum_{i=1}^{M} \sum_{j=1}^{M} p(i,j)ij - \mu_x \mu_y}{\sigma_x(i)\sigma_y(j)}$$

His value shows the linear dependency of gray level values to their respective voxel. It belongs to the interval 0, which means that they are uncorrelated and 1, which means that they are highly correlated.

– *Difference Entropy*: it estimates the texture inhomogeneity of a medical image and it is evaluated as follows

$$DE = \sum_{k=1}^{M} p_{x-y}(k) \log_2 \left( p_{x-y}(k) + \varepsilon \right)$$

$$\text{where} \quad p_{x-y}(k) = \sum_{i=1}^{M} \sum_{j=1}^{M} p(i,j) \quad \text{with} \quad |i-j| = k$$

$$\varepsilon \text{ a fixed small positive number}$$

- NGTDM:
  This acronym stands for Neighboring Gray Tone Difference Matrix. This matrix quantifies the difference between a gray value and the average gray value of its neighbors within a fixed distance $\delta$. NGDTM contains the sum of absolute differences for gray level $i$.

  – *Complexity*: it assesses how complex the distribution of intensities in the region of interest (ROI) is, and it is calculated as

$$Complexity = \frac{1}{M_{v,p}} \sum_{i=1}^{M} \sum_{j=1}^{M} |i-j| \frac{p_i s_i + p_j s_j}{p_i + p_j} \quad \text{where} \quad p_i \neq 0, p_j \neq 0$$

  $p_i$ and $p_j$ are the gray level probabilities; $s_i$ and $s_j$ are the sum of absolute difference for gray level $i$ and $j$. $N_g$ is the number of considered voxel, while $N_{v,p}$ is the total number of voxel.

### 3.2.2 Dataset Manipulation

Before pipeline implementation, some changes had to be made to the dataset. In fact, during the EDA, it was noticed that many mice, either died too early or presented no more than two MRIs with tumor.

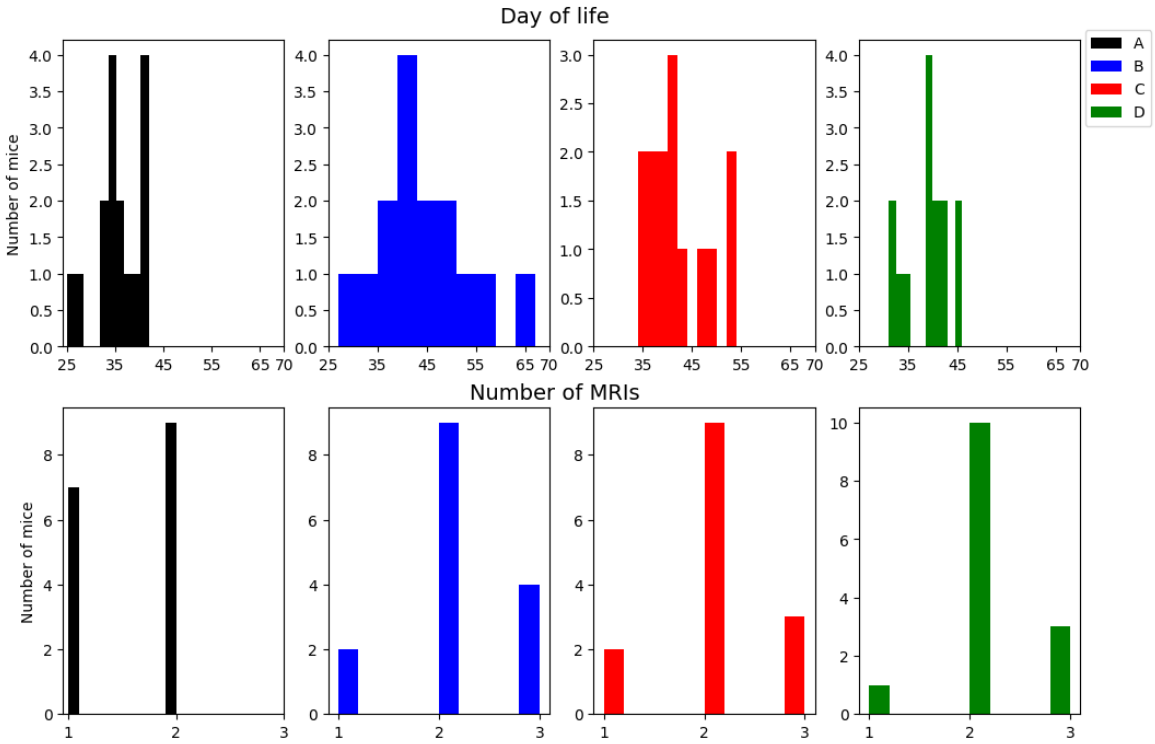


**Figure 3.5:** Mice distribution related to days of life and MRIs obtained, presented for each group

Figure 3.5 shows the distribution of mice versus the day of life (top row) and the number of MRIs obtained for each group (bottom row). It is evident that in the control group A mice live less days compared to the other; in addition, that group is the only one in which no mouse has three MRIs.
Both *pySINDy* and *LASSO* need at least three time points, and so it was necessary to manipulate the dataset before setting up the analysis pipeline.
A new dataset, called *Dataset_Week*, was built by mediating values over time for each group. For each features, at each time-point values were summed and divided by the number of MRIs performed; this resulted a dataset based on the four groups and not mice with more values to be studied

- Group A: 7 time-points;
- Group B: 14 time-points;
- Group C: 9 time-points;
- Group D: 10 time-points

The obtained dataset, containing the mean values computed from *Main_Dataset*, has the same features as this one. The aim, by studying this dataset, is to estimate the average behavior of a group with respect to the treatment to which it is subjected.

### 3.2.3 ODE fitting

A preliminary attempt was made using SINDy, implemented with the Python package PySindy. However, although the $Dataset\_Week$ contains more than three points for each group, they are not evenly distributed along the timeline. This prevents the algorithm from precisely approximating the curve, leading to poor approximations when the time interval between the two curves is larger.
For that reason, Lasso regression has been implemented manually, in order to solve Equation (1.8).
One of the most crucial steps in this type of analysis is to build a suitable base for the available data. First, we have made some attempts exploiting SINDy's preconstructed base.
These libraries consist of Fourier series or, more simply, polynomial functions. However, these type of base do not allow adequate data approximation. Therefore, a custom library was built with multiple functions so that it could be possible to study which ones were the most useful.
Initially, a base was defined, such as the one presented below.

$$\Theta(\mathbf{X}) = \begin{bmatrix} | & | & | & | & | & | & | & | & | & | \\ \mathbf{x} & \mathbf{x}^2 & \log \mathbf{x} & \log(\mathbf{2x}) & \exp(\mathbf{x}) & \exp(\mathbf{2x}) & \sin(\mathbf{x}) & \cos(\mathbf{x}) & \sin(\mathbf{2x}) & \cos(\mathbf{2x}) \\ | & | & | & | & | & | & | & | & | & | \end{bmatrix} \tag{3.1}$$

It was tested, and its coefficients were saved in order to analyze which functions were more efficient compared to the others.
As a result, it was obtained that the logarithmic and exponential functions were quite ineffective since their coefficient in almost all features were null. In addition, it was tested that despite the absence of these functions the approximations were too close to each other. Another reason for ignoring logarithmic and exponential functions is that the former, due to the fact that many features have values around zero, cannot be evaluated with this type of transformation. Exponential functions don't encounter the same problem; however, their coefficients play no significant role in relation to almost all features. The only features for which these functions are relevant are those that have previously been subjected to a logarithmic transformation; nevertheless, the selected features express data without any transformation being taken necessary.
All these considerations led to the definition of the library defined in Equation (2.1).

## 3.3 Results

Lasso regression was implemented as explained in Section 2.2.
Once the approximation of $\dot{\mathbf{X}}$ has been evaluated with *splrep* and *splev* from the library *SciPy* [18], and $\tilde{\mathbf{X}}$ has been calculated Lasso from *sklearn.linear_model* library [9], results have been plotted.

### 3.3.1 Group behavior evaluation

First, we want to investigate whether there are significative differences between groups. Figures 3.6 and 3.7 show some plots representing the fit of the ODE with our pipeline. In each box which represents each group, the darker color (Black for A, Blue for B, Red for C and Dark Green for D) indicates the curve $\tilde{\mathbf{X}}$, while the lighter color (Grey for A, light blue for B, orange for C and light green for D) indicates the curve $\dot{\mathbf{X}}$.
Plots are shown for features *Complexity*, *Surface Area*, *Autocorrelation*, *Contrast*, *Cluster Tendency* and *Cluster Shade*.
Appendix B presents the order of magnitude related to the estimated $\beta$ coefficients, which are used to define $\tilde{\mathbf{X}}$.

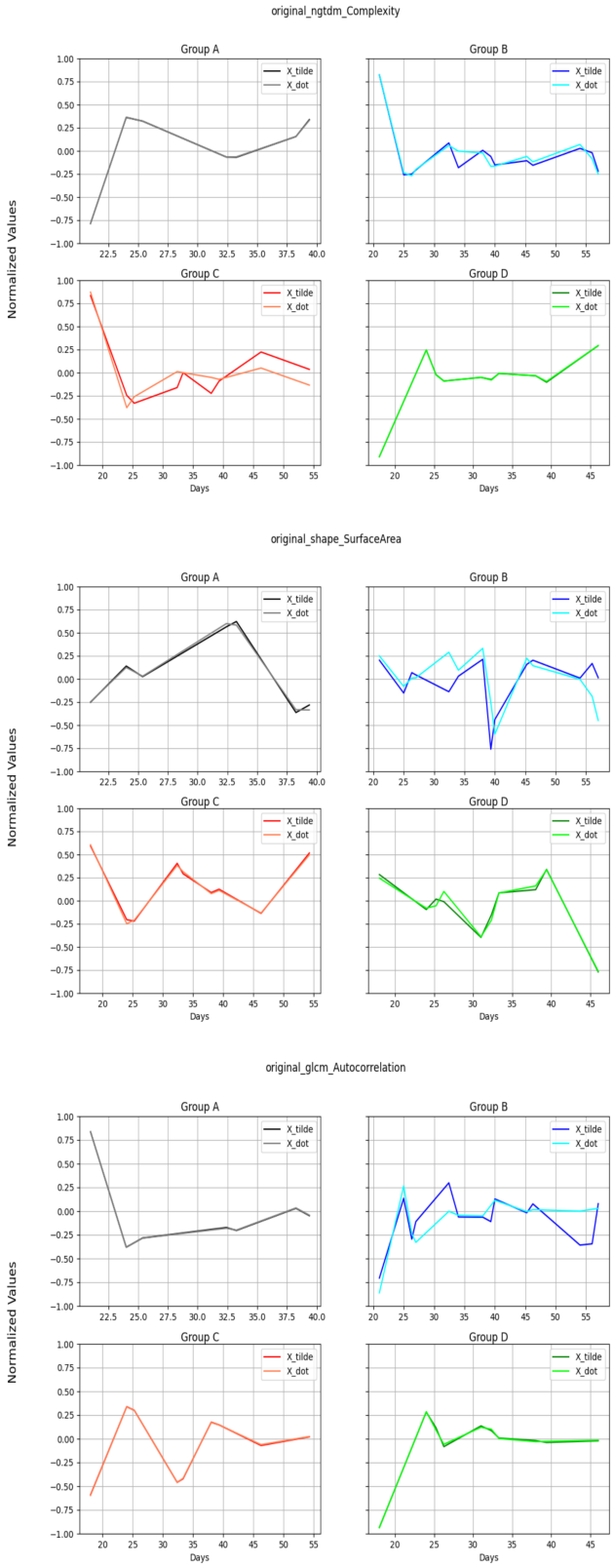


**Figure 3.6:** Graphical representation of the curve $\dot{\mathbf{X}}$ and its approximation $\tilde{\mathbf{X}}$ for the features Complexity, Surface Area and Autocorrelation. The four groups have been studied and their curves have been plotted as reported in each legend.

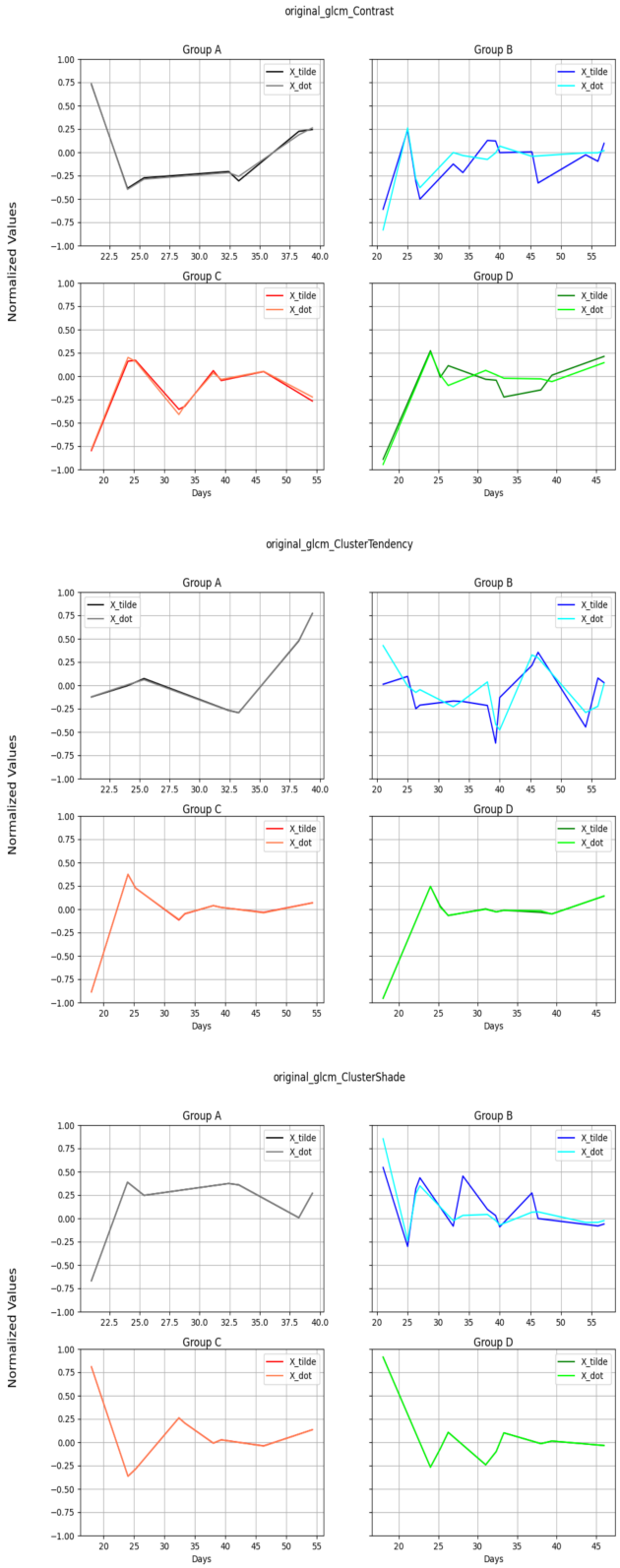


**Figure 3.7:** Graphical representation of the curve $\dot{\mathbf{X}}$ and its approximation $\tilde{\mathbf{X}}$ for the features Contrast, Cluster Tendency and Cluster Shade. The four groups have been studied and their curves have been plotted as reported in each legend.

Curves shown in Figures 3.6 and 3.7 represent groups behavior along time-line; they do not provide any information regarding the individual mouse's response.
Table 3.3.1 presents *MSEs* (Mean Square Error) evaluated as reported in Equation (3.2)

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(\dot{\mathbf{X}}_i - \tilde{\mathbf{X}}_i\right)^2 \tag{3.2}$$

| | Group A | Group B | Group C | Group D |
|---|---|---|---|---|
| $10^{th}$ Percentile | $2.603 \cdot 10^{-1}$ | $5.889 \cdot 10^{-2}$ | $4.239 \cdot 10^{-1}$ | $1.235 \cdot 10^{-1}$ |
| $90^{th}$ Percentile | $1.285 \cdot 10^{-2}$ | $4.503 \cdot 10^{-2}$ | $7.894 \cdot 10^{-3}$ | $7.052 \cdot 10^{-2}$ |
| Autocorrelation | $1.544 \cdot 10^{-5}$ | $3.423 \cdot 10^{-2}$ | $2.161 \cdot 10^{-5}$ | $2.003 \cdot 10^{-4}$ |
| Cluster Shade | $3.859 \cdot 10^{-7}$ | $2.479 \cdot 10^{-2}$ | $1.271 \cdot 10^{-6}$ | $2.896 \cdot 10^{-7}$ |
| Cluster Tendency | $4.763 \cdot 10^{-5}$ | $4.3 \cdot 10^{-2}$ | $1.130 \cdot 10^{-5}$ | $4.225 \cdot 10^{-5}$ |
| Complexity | $1.188 \cdot 10^{-5}$ | $4.051 \cdot 10^{-3}$ | $1.59 \cdot 10^{-2}$ | $1.259 \cdot 10^{-5}$ |
| Contrast | $6.477 \cdot 10^{-4}$ | $1.986 \cdot 10^{-2}$ | $8.741 \cdot 10^{-4}$ | $1.252 \cdot 10^{-2}$ |
| Correlation | $6.991 \cdot 10^{-3}$ | $8.075 \cdot 10^{-2}$ | $2.451 \cdot 10^{-2}$ | $1.365 \cdot 10^{-1}$ |
| Difference Entropy | $3.973 \cdot 10^{-2}$ | $9.451 \cdot 10^{-2}$ | $1.574 \cdot 10^{-2}$ | $1.31 \cdot 10^{-1}$ |
| Elongation | $7.342 \cdot 10^{-2}$ | $1.41 \cdot 10^{-1}$ | $1.469 \cdot 10^{-1}$ | $1.75 \cdot 10^{-1}$ |
| Kurtosis | $1.563 \cdot 10^{-1}$ | $4.654 \cdot 10^{-2}$ | $6.971 \cdot 10^{-2}$ | $3.759 \cdot 10^{-2}$ |
| Mean | $4.62 \cdot 10^{-3}$ | $9.336 \cdot 10^{-2}$ | $3.072 \cdot 10^{-2}$ | $1.74 \cdot 10^{-1}$ |
| Skewness | $9.881 \cdot 10^{-2}$ | $4.455 \cdot 10^{-2}$ | $1.310 \cdot 10^{-1}$ | $1.951 \cdot 10^{-2}$ |
| Surface Area | $9.6 \cdot 10^{-4}$ | $6.004 \cdot 10^{-2}$ | $4.442 \cdot 10^{-4}$ | $2.452 \cdot 10^{-3}$ |
| Uniformity | $3.519 \cdot 10^{-2}$ | $3.21 \cdot 10^{-2}$ | $9.618 \cdot 10^{-2}$ | $3.124 \cdot 10^{-2}$ |

**Table 3.1:** Mean Square Error of the selected features presented in Section 3.2. It has been evaluated to evaluate the discrepancy between the curve $\tilde{\mathbf{X}}$ and $\dot{\mathbf{X}}$

Clear agreement can be seen between the errors reported in Table 3.2 and the graphical representations shown in Figures 3.6 and 3.7.
In most cases, the approximation error between the two curves results negligible, which serves to indicate that the method is appropriate for the available data.
However, it is important to note that Group B has higher errors on average than the other groups. This discrepancy can be attributed to the fact that Group B, despite having the largest number of time points, also has a higher degree of heterogeneity in the distribution of the data; in fact, by analyzing *Dataset_Week*, a greater variability in timeline is observed.
*Dataset_Week* was built in order to study groups average behavior; nonetheless, it might happen that some measures are more mouse dependent than others.
In group B there are more time-points than in the other groups for two reasons:
First, as shown in Figure 3.5, group B has the highest number of MRIs, and consequently, this group has the greatest disparities in days of image acquisition.
As a result, its values reported in *Dataset_Week*, are much more mouse dependent, causing this heterogeneity in the data and therefore higher error in data approximation.
A second analysis was conducted to further investigate the distribution of individual

mice compared with the mean calculated for the each group. Unfortunately, given that almost all mice have only two MRIs, and in group A none of them have three measurements, it was not possible to follow the previous pipeline. Thus, we decide to find at least a way to measure the discrepancy between our group models and each single subject.

### 3.3.2 Intra group variability estimation

As a result, the initial and final distributions, respectively taken from *Onset_Dataset* and *Last_Datset*, were analyzed for each mouse with respect to the curve $\tilde{\mathbf{X}}$.
In Figures 3.8 and 3.9, scatter plots for *Onset_Dataset* are shown in pink, while those for the *Last_Dataset* are colored in gray, for the same features displayed in Figures 3.6 and 3.7.
The plots of the same features presented in Figures 3.6 and 3.7 are shown.
In each plot, the first row represents the curve $\tilde{\mathbf{X}}$ with the distribution of onset values colored in pink.
Instead, the second row, represents the curve $\tilde{\mathbf{X}}$ with the distribution of last values colored in gray.
It is important to note that these plots have been limited to the interval $[-1,1]$, within which the curve exists, as it has been normalized with respect to its Euclidean norm. However, the scatter plots, also normalized with respect to the same norm, do not always fall within this range. These point so are considered as outliers.

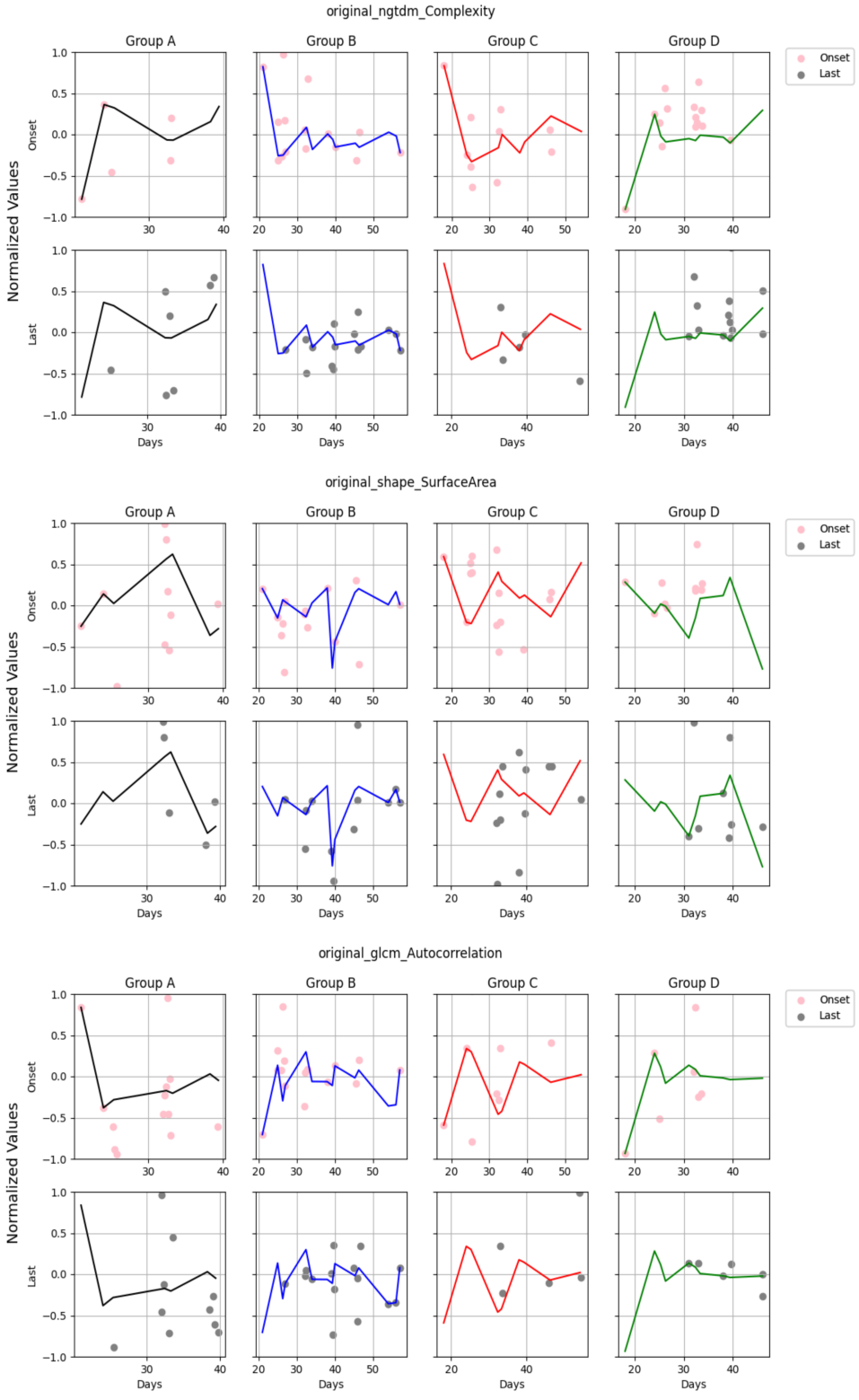


**Figure 3.8:** Graphical representation of the curve $\tilde{\mathbf{X}}$ and onset and last values distribution for the features Complexity, Surface Area, Autocorrelation. Pink scatters represent onset values; gray scatters indicates last values.

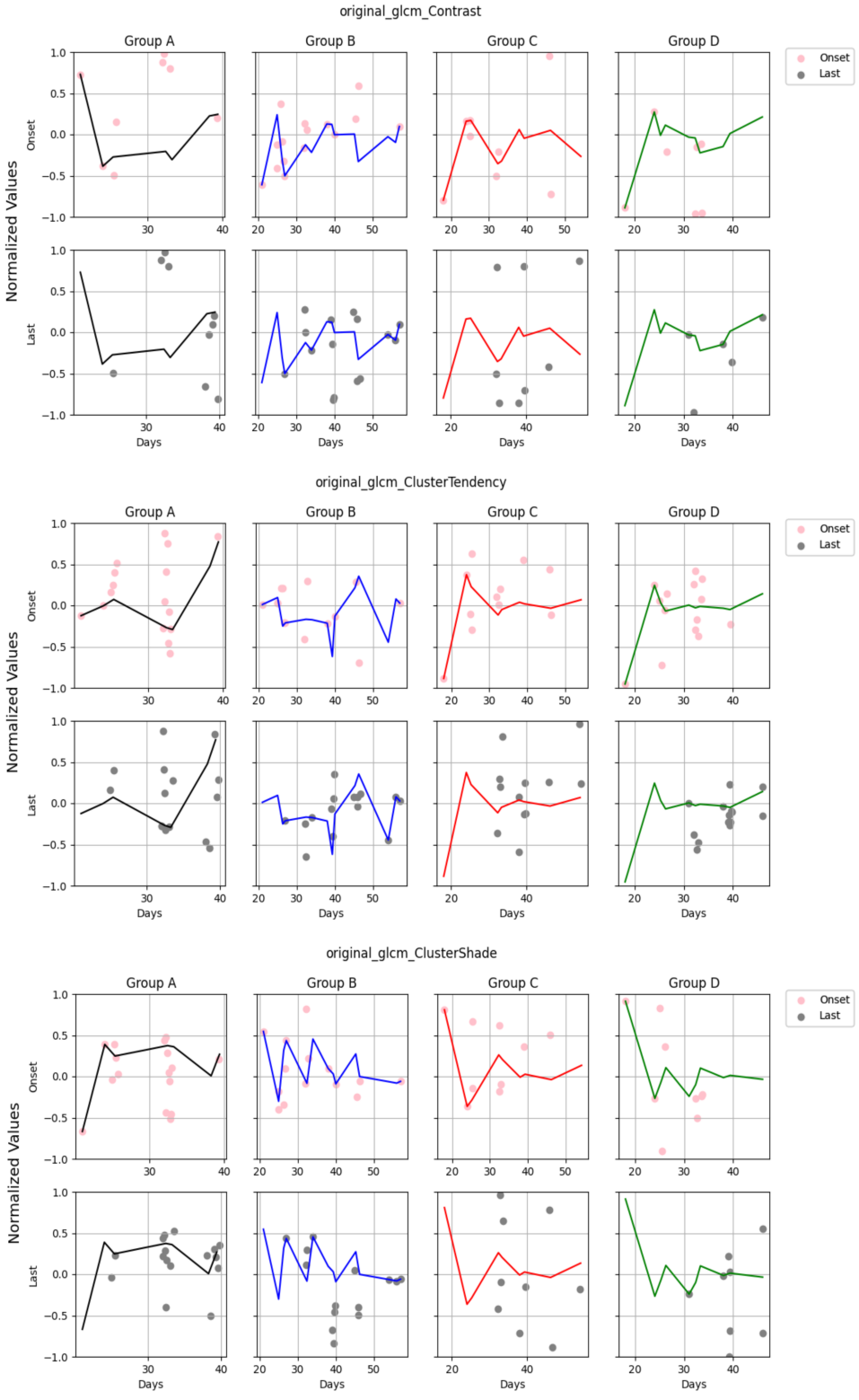


**Figure 3.9:** Graphical representation of the curve $\tilde{\mathbf{X}}$ and onset and last values distribution for the features Contrast, Cluster Tendency, Cluster Shade. Pink scatters represent onset values; gray scatters indicates last values.

We report in Table 3.2 the number of outlier for each feature and each group.

| | **Onset_Dataset** | | | | **Last_Dataset** | | | |
|---|---|---|---|---|---|---|---|---|
| | A | B | C | D | A | B | C | D |
| $10^{th}$ Percentile | 1 | 3 | 1 | 0 | 1 | 0 | 2 | 0 |
| $90^{th}$ Percentile | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 2 |
| Autocorrelation | 3 | 0 | 0 | 7 | 6 | 0 | 9 | 8 |
| Cluster Shade | 0 | 1 | 5 | 5 | 0 | 1 | 5 | 6 |
| Cluster Tendency | 0 | 1 | 3 | 0 | 1 | 0 | 2 | 0 |
| Complexity | 11 | 0 | 4 | 0 | 9 | 0 | 9 | 2 |
| Contrast | 8 | 0 | 6 | 7 | 7 | 0 | 6 | 9 |
| Correlation | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 4 |
| Difference Entropy | 4 | 0 | 0 | 1 | 2 | 0 | 0 | 0 |
| Elongation | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Kurtosis | 0 | 1 | 3 | 2 | 0 | 0 | 1 | 0 |
| Mean | 3 | 0 | 0 | 2 | 3 | 0 | 1 | 0 |
| Skewness | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Surface Area | 6 | 0 | 0 | 4 | 11 | 3 | 2 | 6 |
| Uniformity | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Table 3.2:** Outlier distribution among features and groups. For each group, the number of outlier has been evaluated and reported in the table, to obtain a general overview of the number of points that are not represented in Figures 3.8 and 3.9

Combining information presented in Table 3.2, Figures 3.8 and 3.9, it can be concluded that some features, e.g *Complexity*, despite being interpolated with a small approximation error, present limitations in drawing conclusions about the behavior of individual mice.

Whereas, features as *Cluster Tendency*, has smaller number of outlier, both in *Onset_Dataset* and *Last_Dataset*; that helps in order to understand data distribution for the fist and last MRI. It can be observed that for the first MRI data are focused in a neighborhood of the initial time values; on the other side, for the last MRI values are grouped around final time points.

Graphically it is complex to observe that the pink and gray values match, but it may happen. That means that the corresponding mouse at the end of the trial, exhibited only one MRI with tumor; numerically it is easier to be observed from the datasets.

Figures 3.6, 3.7, 3.8 and 3.9 partially validate arguments made with the Kaplan-Meier curves, and presented in Figures 3.2 and 3.3.

In fact, group A, has a shorter time interval compared to the other three, which means that mice who belong to the control group have a briefer prospectus of life. Groups B, C and D show similar behavior, since both B and C are cured with epigenetic drug, and both C and D with immunotherapy.

Hence, the last analysis performed is the evaluation of the distance between values of individual mice in *Onset_Dataset* and *Last_Dataset* and the $\tilde{\mathbf{X}}$, as presented in Section 2.2 using Matlab.

A change of software was chosen, since Python presented computational difficulties.

However, Matlab, despite its high capacity in symbolic computation, also suffered difficulties; in fact, some values were computed with the symbolic function *syms* [6], but using a numerical approximation throught *vpasolve* [5]. This process is automatically implemented by Matlab, when the equation to be solved is particularly complex as the one in this study, presented in Equation (3.3).

$$min\_dist = (x - P_x) + (\beta\boldsymbol{\Theta} + \beta_0 - P_y)\,\beta\frac{d\boldsymbol{\Theta}}{dx} \tag{3.3}$$

where $P_x$ and $P_y$ are the values taken from the dataset.
Average distances values are reported in Table 3.3.

| | **Onset_Dataset** | | | | **Last_Dataset** | | | |
|---|---|---|---|---|---|---|---|---|
| | A | B | C | D | A | B | C | D |
| $10^{th}$Percentile | 0.239 | 0.242 | 0.256 | 0.249 | 0.243 | 0.251 | 0.261 | 0.243 |
| $90^{th}$Percentile | 0.249 | 0.258 | 0.267 | 0.249 | 0.25 | 0.258 | 0.267 | 0.25 |
| Autocorrelation | 0.179 | 0.238 | 0.244 | 0.179 | 0.187 | 0.218 | 0.263 | 0.187 |
| Cluster Shade | 0.146 | 0.161 | 0.108 | 0.146 | 0.112 | 0.12 | 0.127 | 0.112 |
| Cluster Tendency | 0.214 | 0.243 | 0.142 | 0.214 | 0.222 | 0.258 | 0.169 | 0.222 |
| Complexity | 0.244 | 0.255 | 0.263 | 0.244 | 0.246 | 0.257 | 0.218 | 0.246 |
| Contrast | 0.239 | 0.25 | 0.172 | 0.239 | 0.239 | 0.244 | 0.155 | 0.239 |
| Correlation | 0.207 | 0.209 | 0.216 | 0.207 | 0.214 | 0.219 | 0.228 | 0.214 |
| Difference Entropy | 0.246 | 0.232 | 0.25 | 0.246 | 0.247 | 0.236 | 0.259 | 0.247 |
| Elongation | 0.206 | 0.204 | 0.201 | 0.206 | 0.203 | 0.219 | 0.228 | 0.203 |
| Kurtosis | 0.242 | 0.24 | 0.252 | 0.242 | 0.245 | 0.25 | 0.26 | 0.245 |
| Mean | 0.245 | 0.239 | 0.26 | 0.245 | 0.246 | 0.249 | 0.267 | 0.246 |
| Skewness | 0.245 | 0.245 | 0.225 | 0.246 | 0.247 | 0.251 | 0.241 | 0.247 |
| Surface Area | 0.246 | 0.243 | 0.263 | 0.246 | 0.118 | 0.152 | 0.241 | 0.118 |
| Uniformity | 0.226 | 0.239 | 0.245 | 0.226 | 0.22 | 0.222 | 0.246 | 0.22 |

**Table 3.3:** Distance between $\tilde{\mathbf{X}}$ and data in *Onset_Dataset* and *Last_Dataset*. For each group, the distance has been evaluated and normalized, in order to be compared with Figures 3.8 and 3.9

These distances have been evaluated symbolically and subsequently normalized with respect to their norm, in order to have a data that can be compared with Figures 3.8 and 3.9.
Since they are normalized, their maximum value is equal to *2*, when for instance, the curve value is *-1* and the point values is *1* at the same time point. Instead, their minimum is equal to *0*, when the curve and the point have the same value at the same time point.
They were normalized to have a data that can be comparable with Figures 3.8 and 3.9.
It can be said that all reported values in Table 3.3 agree each other and show, in general, a good approximation between the average trend of the groups and individual mice.
However, it is pointed out that the distance of some features, e.g. *Surface Area* and

*Cluster Shade* (except for group C), has a decreasing trend; in fact, in these cases their value in the *Onset_Dataset* is higher than that present in the *Last_Dataset*. This shows a greater heterogeneity in the data at the beginning of the trial, while at the end all mice, on average, have a more uniform treatment response.
On the contrary, for instance *Cluster Tendency*, shows the opposite behavior; in fact, it presents more data homogeneity at onset than in last time instant.
It has to be specified that these distances have been evaluated including outliers; these points have not been removed from the dataset in order not to alter the dataset and have a total view of the studied group.

# Conclusion

The aim of this study is to analyze longitudinal radiomic features, obtained from a mice population, subject to injection of glioblastoma multiforme (GBM) cells.
These mice, divided in four groups, were cured with three different therapy: the first groups was the control one, the second with epigenetic drugs, the third with both epigenetic drugs and immunotherapy, and the fourth with immunotherapy only.
The former analysis, implemented with Kaplan-Meier curves and presented in Figures 3.2 and 3.3, was made with the aim to understand the survival probability of each groups, so that it can be compared to the further radiomic analysis.
Subsequently, using Lasso regression, an ODE (1.8) for each selected radiomic feature was studied and estimated in order to investigate the existence of differences between groups. Plots of this preliminary study are reported in Figures 3.6 and 3.7.
These curves were studied and compared to the values of the features for single subjects contained in *Onset_Dataset* and *Last_Dataset*. The former includes values of the first MRI for each mouse, the latter contains values of the last MRI for each mouse. These analysis were useful in order to understand intra-group variability. Results are presented in Figure 3.8 and 3.9
To explore this further, distances between these points and the estimated curve were calculated and reported in Table 3.3.
In conclusion, starting from results obtained from Kaplan-Meier curves, group B and C seems to have the highest survival probability along timeline, followed by group D, while group A has the lowest one.
Those results have been partially validated by the radiomics analysis. In fact, Group A has a shorter temporal interval compared to the other three, indicating that subjects in the control group have a shorter follow-up period rather than a difference in survival. Groups B, C, and D exhibit similar patterns, as both B and C receive epigenetic drug treatment, while both C and D undergo immunotherapy and seems to develop the tumor later in time. A set of relevant features has been chosen among all the 1130 available, in order to observe and to study the tumor growth and behavior for each group. We have selected non filtered features to ease the future biological interpretation of the results.
These features study not only tumor growth, but also its composition, since glioblastoma multifome (GBM) is characterized by the growth of cysts and necrotic areas inside of it.

Results, shown in Section 3.3 seems to confirm that groups B and C had the best response to treatment. More precisely, the use of the epigenetic drug has a clear impact on the survival of the subjects while the combination of the drug and the immunotherapy appears to further delay the onset of cancer. By comparing the approximate curves shown in Figures 3.6 and 3.7, but also the distribution of data at the onset and last time as presented in Figures 3.8 and 3.9, it can be confirmed that groups B and C have a similar behavior. In fact comparing the curves, it can be noticed that in almost all the analyzed features the respective $\tilde{\mathbf{X}}$ have a similar trend. Distances and number of outliers reported in Tables 3.3 and 3.2, respectively, were helpful in order to understand mice performance compared with the average performance estimated by $\dot{\mathbf{X}}$ and $\tilde{\mathbf{X}}$.
Summing up all this information, it can be stated that epigenetic drug, to which both groups were subjected, causes a better response than immunotherapy.
The next step, which will be implemented in the coming months, will involve a more in-depth study of the subjects' internal organs so that biologists can understand, if they are present, what side effects the treatments have on the mice body.

# Appendix A

# Features considered

[2]

| Category | Feature Name |
|---|---|
| First Order | Energy |
| | Total Energy |
| | Entropy |
| | $10^{th}$ percentile |
| | $90^{th}$ percentile |
| | Mean |
| | Median |
| | Interquartile Range |
| | Robust Mean Absolute Deviation |
| | Root Mean Squared |
| | Skewness |
| | Kurtosis |
| | Variance |
| | Uniformity |
| Neighboring Gray Tone Difference Matrix (NGTDM) | Coarness |
| | Contrast |
| | Busyness |
| | Complexity |
| | Strength |
| Shaped-based (2D) | Mesh Surface |
| | Pixel Surface |
| | Perimeter |
| | Perimeter Surface Ratio |
| | Sphericity |
| | Maximum Diameter |
| | Major Axis Length |
| | Minor Axis Length |
| | Elongation |

| **Category** | **Feature Name** |
|---|---|
| Gray Level Co-Occurence Matrix (GLCM) | Autocorrelation |
| | Joint Average |
| | Cluster Prominence |
| | Cluster Shade |
| | Cluster Tendency |
| | Contrast |
| | Difference Average |
| | Difference Entropy |
| | Difference Variance |
| | Joint Energy |
| | Joint Entropy |
| | Informational Measure of Correlation (Imc1) |
| | Maximal Correlation Coefficient (MCC) |
| | Inverse Difference Moment Normalized (Idn) |
| | Inverse Variance |
| | Maximum Probabilty |
| | Sum Entropy |
| | Sum Squares |
| Gray Level Size Zone Matrix (GLSZM) | Small Area Empashis |
| | Large Area Emphasis |
| | Gray Level Non Uniformity |
| | Gray Level Non Uniformity Normalized |
| | Size Zone Non Uniformity |
| | Size Zone Non Uniformity Normalized |
| | Zone Percentage |
| | Gray Level Variance |
| | Zone Variance |
| | Zone Entropy |
| | Low Gray Level Zone Emphasis |
| | High Gray Level Zone Emphasis |
| | Small Area Low Gray Level Emphasis |
| | Small Area High Gray Level Emphasis |
| | Large Area Low Gray Level Emphasis |
| | Large Area High Gray Level Emphasis |

| **Category** | **Feature Name** |
|---|---|
| Gray Level Dependence Matrix (GLDM) | Small Dependence Emphasis |
| | Large Dependence Emphasis |
| | Gray Level Non Uniformity |
| | Dependence Non Uniformity |
| | Dependence Non Uniformity Normalized |
| | Gray Level Variance |
| | Dependence Variance |
| | Dependence Entropy |
| | Low Gray Level Emphasis |
| | High Gray Level Emphasis |
| | Small Dependence Low Gray Level Emphasis |
| | Small Dependence High Gray Level Emphasis |
| | Large Dependence Low Gray Level Emphasis |
| | Large Dependence High Gray Level Emphasis |
| Gray Level Run Length Matrix (GLRLM) | Short Run Emphasis |
| | Long Run Emphasis |
| | Gray Level Non Uniformity |
| | Gray Level Non Uniformity |
| | Gray Level Non Uniformity Normalized |
| | Run Length Non Uniformity |
| | Run Length Non Uniformity Normalized |
| | Run Percentage |
| | Gray Level Variance |
| | Run Variance |
| | Run Entropy |
| | Low Gray Level Run Emphasis |
| | High Gray Level Run Emphasis |
| | Short Run Low Gray Level Emphasis |
| | Short Run High Gray Level Emphasis |
| | Long Run Low Gray Level Emphasis |
| | Long Run High Gray Level Emphasis |

# Appendix B

# Estimated coefficients

Denoted the functions that form the basis as follows:

- $f_0$ = model.intercept_
- $f_1(x) = x$
- $f_2(x) = x^2$
- $f_3(x) = x^3$
- $f_4(x) = x^4$
- $f_5(x) = \sin(x)$
- $f_6(x) = \cos(x)$
- $f_7(x) = \sin(2x)$
- $f_8(x) = \cos(2x)$
- $f_9(x) = \sin(3x)$
- $f_{10}(x) = \cos(3x)$

## B.1 Group A

| Base Functions | $f_0$ | $f_1(x)$ | $f_2(x)$ | $f_3(x)$ | $f_4(x)$ | $f_5(x)$ | $f_6(x)$ | $f_7(x)$ | $f_8(x)$ | $f_9(x)$ | $f_{10}(x)$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $10^{th}$ Percentile | $10^{-3}$ | 0 | 0 | 0 | $10^{-5}$ | 0 | 0 | 0 | 0 | 0 | 0 |
| $90^{th}$ Percentile | $10^{-1}$ | 0 | 0 | $-10^{-3}$ | $10^{4}$ | 0 | 0 | 0 | 0 | 0 | 0 |
| Autocorrelation | $10^{4}$ | $-10^{0}$ | $10^{-4}$ | $10^{-8}$ | $10^{-12}$ | $-10^{3}$ | $10^{3}$ | $10^{2}$ | 0 | $-10^{3}$ | $10^{3}$ |
| Cluster Shade | $10^{4}$ | $10^{-2}$ | $10^{-5}$ | $10^{-10}$ | $-10^{-14}$ | $-10^{-4}$ | $-10^{-4}$ | $-10^{-4}$ | $-10^{-4}$ | $-10^{-4}$ | $-10^{-4}$ |
| Cluster Tendency | $10^{0}$ | $10^{-1}$ | $10^{-4}$ | $10^{-7}$ | $10^{-10}$ | $-10^{1}$ | $10^{2}$ | 0 | 0 | $-10^{1}$ | $10^{2}$ |
| Complexity | $10^{5}$ | $-10^{1}$ | $10^{-4}$ | $10^{-9}$ | $10^{-13}$ | $-10^{3}$ | $10^{3}$ | 0 | $10^{3}$ | $10^{3}$ | 0 |
| Contrast | $-10^{2}$ | $10^{0}$ | $-10^{-3}$ | $-10^{-6}$ | $-10^{-9}$ | $-10^{2}$ | 0 | $10^{1}$ | $-10^{2}$ | $10^{1}$ | $-10^{3}$ |
| Correlation | $10^{-2}$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | $10^{-3}$ |
| Difference Entropy | $10^{0}$ | $-10^{-1}$ | $-10^{-4}$ | $-10^{-4}$ | $-10^{-5}$ | $-10^{0}$ | $10^{-1}$ | $-10^{-1}$ | $-10^{-1}$ | $-10^{-1}$ | $-10^{-1}$ |
| Elongation | $10^{-2}$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | $10^{-2}$ |
| Kurtosis | $-10^{-2}$ | 0 | 0 | 0 | $10^{-4}$ | 0 | 0 | 0 | 0 | 0 | 0 |
| Mean | $10^{-1}$ | 0 | 0 | $-10^{-3}$ | $-10^{-4}$ | 0 | 0 | 0 | 0 | $10^{-1}$ | $10^{-2}$ |
| Skewness | $10^{-2}$ | 0 | 0 | 0 | $10^{-2}$ | 0 | 0 | 0 | 0 | $-10^{-2}$ | $-10^{-2}$ |
| Surface Area | $10^{2}$ | $-10^{-1}$ | $-10^{-2}$ | $10^{-5}$ | $10^{-6}$ | 0 | $-10^{1}$ | $-10^{2}$ | $10^{1}$ | $-10^{1}$ | $-10^{0}$ |
| Uniformity | $-10^{-3}$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | $10^{-2}$ | 0 |

## B.2 Group B

| Base Functions | $f_0$ | $f_1(x)$ | $f_2(x)$ | $f_3(x)$ | $f_4(x)$ | $f_5(x)$ | $f_6(x)$ | $f_7(x)$ | $f_8(x)$ | $f_9(x)$ | $f_{10}(x)$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $10^{th}$ Percentile | $-10^{-2}$ | 0 | 0 | $-10^{-3}$ | $10^{-3}$ | 0 | 0 | $-10^{-2}$ | $-10^{-1}$ | $10^{-2}$ | $10^{-3}$ |
| $90^{th}$ Percentile | $10^{0}$ | 0 | 0 | $-10^{-2}$ | $10^{3}$ | 0 | 0 | 0 | 0 | 0 | 0 |
| Autocorrelation | $10^{2}$ | $-10^{0}$ | $10^{-5}$ | $10^{-8}$ | $10^{-12}$ | $10^{3}$ | $-10^{3}$ | $10^{3}$ | $-10^{3}$ | $10^{3}$ | $-10^{3}$ |
| Cluster Shade | $10^{4}$ | $10^{0}$ | $-10^{-3}$ | $-10^{-8}$ | $10^{-12}$ | $10^{4}$ | $-10^{4}$ | $10^{4}$ | $10^{4}$ | $-10^{4}$ | $10^{4}$ |
| Cluster Tendency | $10^{3}$ | $-10^{0}$ | $10^{-3}$ | $10^{-7}$ | $-10^{-9}$ | $-10^{1}$ | $-10^{0}$ | $-10^{-1}$ | $10^{1}$ | $10^{1}$ | $-10^{2}$ |
| Complexity | $10^{4}$ | $-10^{0}$ | $10^{-4}$ | $10^{-10}$ | $-10^{-14}$ | $-10^{3}$ | $10^{3}$ | $-10^{3}$ | $10^{3}$ | $10^{3}$ | $10^{3}$ |
| Contrast | $-10^{3}$ | $10^{1}$ | $-10^{-2}$ | $-10^{-6}$ | $10^{-8}$ | $-10^{2}$ | $10^{1}$ | $10^{2}$ | $10^{2}$ | $-10^{2}$ | $10^{2}$ |
| Correlation | $10^{-3}$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | $10^{-6}$ |
| Difference Entropy | $10^{-4}$ | 0 | 0 | $10^{-5}$ | $-10^{-7}$ | 0 | 0 | 0 | 0 | 0 | 0 |
| Elongation | $-10^{-4}$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | $10^{-6}$ | 0 |
| Kurtosis | $-10^{-1}$ | 0 | $-10^{-3}$ | $10^{-4}$ | $10^{-4}$ | 0 | 0 | $10^{-1}$ | $10^{-1}$ | $10^{-1}$ | $10^{-2}$ |
| Mean | $10^{-1}$ | 0 | $-10^{-2}$ | $-10^{-4}$ | $10^{-4}$ | 0 | 0 | 0 | $10^{-2}$ | $10^{-3}$ | $10^{-2}$ |
| Skewness | $10^{-2}$ | 0 | 0 | 0 | $-10^{-3}$ | 0 | 0 | $10^{-3}$ | 0 | $10^{-3}$ | 0 |
| Surface Area | $-10^{1}$ | $10^{0}$ | $-10^{-2}$ | $-10^{-7}$ | $10^{-6}$ | $-10^{1}$ | $10^{1}$ | $10^{0}$ | $-10^{1}$ | $-10^{1}$ | $-10^{1}$ |
| Uniformity | $10^{-5}$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | $-10^{-3}$ | 0 |

## B.3 Group C

| Base Functions | $f_0$ | $f_1(x)$ | $f_2(x)$ | $f_3(x)$ | $f_4(x)$ | $f_5(x)$ | $f_6(x)$ | $f_7(x)$ | $f_8(x)$ | $f_9(x)$ | $f_{10}(x)$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $10^{th}$ Percentile | $10^{-1}$ | 0 | $-10^{-1}$ | $10^{-4}$ | $10^{-3}$ | 0 | 0 | $10^{-1}$ | 0 | 0 | $-10^{-2}$ |
| $90^{th}$ Percentile | $10^{0}$ | 0 | 0 | $-10^{-2}$ | $10^{-3}$ | 0 | 0 | 0 | 0 | 0 | 0 |
| Autocorrelation | $10^{4}$ | $-10^{1}$ | $10^{-4}$ | $10^{-7}$ | $10^{-11}$ | $-10^{2}$ | $-10^{2}$ | $10^{3}$ | $10^{3}$ | $10^{3}$ | $10^{3}$ |
| Cluster Shade | $10^{3}$ | $10^{-1}$ | $10^{-4}$ | $10^{-8}$ | $-10^{-13}$ | $10^{3}$ | $10^{4}$ | $-10^{4}$ | $-10^{4}$ | $-10^{4}$ | $10^{4}$ |
| Cluster Tendency | $10^{3}$ | $-10^{0}$ | $10^{-4}$ | $10^{-8}$ | $10^{-11}$ | $-10^{2}$ | $-10^{2}$ | 0 | $-10^{2}$ | $-10^{2}$ | $10^{1}$ |
| Complexity | $-10^{5}$ | $10^{0}$ | $-10^{-5}$ | $-10^{10}$ | $10^{-14}$ | $-10^{4}$ | $-10^{5}$ | $-10^{4}$ | $-10^{5}$ | $-10^{4}$ | $-10^{4}$ |
| Contrast | $-10^{2}$ | $10^{0}$ | $10^{-3}$ | $-10^{-6}$ | $-10^{-9}$ | $10^{2}$ | $-10^{1}$ | $-10^{2}$ | $10^{2}$ | $10^{2}$ | $-10^{2}$ |
| Correlation | $10^{-2}$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | $-10^{-2}$ | $10^{-2}$ |
| Difference Entropy | $-10^{-2}$ | 0 | 0 | $10^{-3}$ | $-10^{-4}$ | 0 | 0 | 0 | 0 | $10^{-2}$ | 0 |
| Elongation | $-10^{-3}$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | $10^{-6}$ |
| Kurtosis | $10^{-1}$ | 0 | 0 | $-10^{-3}$ | $-10^{-4}$ | 0 | 0 | 0 | 0 | $-10^{-2}$ | $10^{-2}$ |
| Mean | $10^{-1}$ | 0 | 0 | $-10^{-2}$ | $10^{-3}$ | 0 | 0 | 0 | 0 | 0 | 0 |
| Skewness | $10^{-3}$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | $10^{-3}$ | 0 |
| Surface Area | $10^{1}$ | $-10^{0}$ | $10^{-2}$ | $10^{-5}$ | $-10^{-7}$ | $-10^{-1}$ | 0 | $-10^{1}$ | $-10^{1}$ | $10^{0}$ | $-10^{1}$ |
| Uniformity | $10^{-3}$ | 0 | 0 | 0 | 0 | 0 | 0 | $-10^{-3}$ | 0 | $-10^{-2}$ | 0 |

## B.4 Group D

| Base Functions | $f_0$ | $f_1(x)$ | $f_2(x)$ | $f_3(x)$ | $f_4(x)$ | $f_5(x)$ | $f_6(x)$ | $f_7(x)$ | $f_8(x)$ | $f_9(x)$ | $f_{10}(x)$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $10^{th}$ Percentile | $-10^{-1}$ | 0 | 0 | $10^{-3}$ | $-10^{-3}$ | 0 | 0 | 0 | 0 | 0 | 10 |
| $90^{th}$ Percentile | $-10^{-1}$ | 0 | $10^{-3}$ | $10^{-4}$ | $10^{-5}$ | 0 | 0 | $10^{-1}$ | $-10^{-1}$ | $-10^{-1}$ | $10^{-2}$ |
| Autocorrelation | $-10^{3}$ | $10^{0}$ | $10^{-4}$ | $-10^{-8}$ | $-10^{-11}$ | $-10^{4}$ | $-10^{3}$ | $-10^{3}$ | $-10^{3}$ | $10^{3}$ | $10^{3}$ |
| Cluster Shade | $10^{4}$ | $-10^{0}$ | $-10^{-4}$ | $10^{-9}$ | $-10^{-13}$ | $-10^{4}$ | $-10^{4}$ | $-10^{4}$ | $10^{4}$ | $10^{3}$ | $-10^{4}$ |
| Cluster Tendency | $10^{3}$ | $-10^{0}$ | $-10^{-4}$ | $-10^{-7}$ | $-10^{-10}$ | $-10^{2}$ | $10^{1}$ | $10^{1}$ | $10^{2}$ | $10^{0}$ | $-10^{2}$ |
| Correlation | $-10^{-1}$ | $10^{-1}$ | $-10^{-2}$ | $10^{-1}$ | $10^{-1}$ | $10^{-1}$ | $10^{-1}$ | $10^{-1}$ | $10^{-2}$ | $10^{-1}$ | $10^{-2}$ |
| Complexity | $-10^{4}$ | $10^{0}$ | $-10^{-5}$ | $-10^{-9}$ | $-10^{-14}$ | $10^{4}$ | $-10^{4}$ | $-10^{4}$ | $10^{4}$ | $10^{4}$ | $-10^{2}$ |
| Contrast | $-10^{3}$ | $10^{0}$ | $-10^{-2}$ | $-10^{-5}$ | $10^{-7}$ | $10^{3}$ | $-10^{3}$ | $10^{3}$ | $-10^{3}$ | $10^{3}$ | $10^{3}$ |
| Difference Entropy | $10^{-2}$ | 0 | 0 | $-10^{-3}$ | $10^{-4}$ | 0 | 0 | 0 | 0 | $-10^{-2}$ | 0 |
| Elongation | $-10^{-3}$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | $10^{-5}$ |
| Kurtosis | $10^{-1}$ | 0 | $-10^{-2}$ | 0 | $10^{-4}$ | 0 | 0 | $-10^{-1}$ | $-10^{-1}$ | $-10^{-1}$ | 0 |
| Mean | $-10^{-2}$ | 0 | 0 | 0 | $10^{-5}$ | 0 | 0 | 0 | 0 | 0 | 0 |
| Skewness | $10^{-2}$ | 0 | 0 | $-10^{-2}$ | $-10^{-2}$ | 0 | 0 | 0 | $10^{-3}$ | $10^{-2}$ | 0 |
| Surface Area | $10^{2}$ | $-10^{0}$ | $10^{-2}$ | $-10^{-5}$ | $-10^{-7}$ | $-10^{2}$ | $10^{2}$ | $10^{1}$ | $10^{2}$ | $10^{2}$ | $-10^{2}$ |
| Uniformity | $10^{-3}$ | 0 | 0 | 0 | 0 | 0 | 0 | $-10^{-3}$ | 0 | $-10^{-2}$ | 0 |

# Bibliography


[1] Fenoglio D-Gangemi R Tosi A Parodi A Banelli B Rigo V Mastracci L Grillo F Cereghetti A Tastanova A Ghosh A Sallustio F Emionite L Daga A Altosole T Filaci G Rosato A Levesque M Maio M Pfeffer U Croce M; EPigenetic Immune-oncology Consortium Airc (EPICA) consortium. Amaro A, Reggiani F. **Guadecitabine increases response to combined anti-CTLA-4 and anti-PD-1 treatment in mouse melanoma in vivo by controlling T-cells, myeloid derived suppressor and NK cells**. *J Exp Clin Cancer Res*, 2023. doi: doi:10.1186/s13046-023-02628-x.

[2] Isabella Cama, Valentina Candiani, Luca Roccatagliata, Pietro Fiaschi, Giacomo Rebella, Martina Resaz, Michele Piana, and Cristina Campi. Segmentation agreement and the reliability of radiomics features. ***Advances in Computational Science and Engineering***, 1(2):202–217, 2023. doi: 10.3934/acse.2023009. URL https://www.aimsciences.org/article/id/649d0fa91778ab0a9522bc76.

[3] Brian de Silva, Kathleen Champion, Markus Quade, Jean-Christophe Loiseau, J. Kutz, and Steven Brunton. **PySINDy: A Python package for the sparse identification of nonlinear dynamical systems from data**. *Journal of Open Source Software*, 5(49):2104, 2020. doi: 10.21105/joss.02104. URL https://doi.org/10.21105/joss.02104.

[4] Kishore J Goel MK, Khanna P. **Understanding survival analysis: Kaplan-Meier estimate.** *Int J Ayurveda Res.*, 2010. doi: doi:10.4103/0974-7788.76794.

[5] The MathWorks Inc. ***vpasolve - Variable-Precision Arithmetic Solver***, 2024. Available at https://www.mathworks.com/help/symbolic/sym.vpasolve.html.

[6] The MathWorks Inc. **syms - Create symbolic variables and functions**. *MATLAB Documentation*, 2025. URL https://www.mathworks.com/help/symbolic/syms.html.

[7] Alan A. Kaptanoglu, Brian M. de Silva, Urban Fasel, Kadierdan Kaheman, Andy J. Goldschmidt, Jared Callaham, Charles B. Delahunt, Zachary G. Nicolaou, Kathleen Champion, Jean-Christophe Loiseau, J. Nathan Kutz, and

Steven L. Brunton. **PySINDy: A comprehensive Python package for robust sparse system identification**. *Journal of Open Source Software*, 7 (69):3994, 2022. doi: 10.21105/joss.03994. URL https://doi.org/10.21105/joss.03994.

[8] R. Marler and Jasbir Arora. **Survey of Multi-Objective Optimization Methods for Engineering**. *Structural and Multidisciplinary Optimization*, 26: 369–395, 04 2004. doi: 10.1007/s00158-003-0368-6.

[9] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. **Scikit-learn: Machine Learning in Python**. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[10] Enrico Pegoraro. **Statistica per Data Science con R**. 2019.

[11] Jason T. Rich, J. Gail Neely, Randal C. Paniello, Courtney C. J. Voelker, Brian Nussenbaum, and Eric W. Wang. **A practical guide to understanding Kaplan-Meier curves**. *Otolaryngology–Head and Neck Surgery*, 143(3):331–336, 2010. doi: https://doi.org/10.1016/j.otohns.2010.05.007. URL https://aaohnsfjournals.onlinelibrary.wiley.com/doi/abs/10.1016/j.otohns.2010.05.007.

[12] Somasundaran Sandhya. **Comprehending Kaplan–Meier curve**. *Kerala Journal of Ophthalmology*, 2023. doi: 10.4103/kjo.kjo_91_23.

[13] Joshua L. Proctor Steven L. Brunton and J. Nathan Kutz. **Discovering governing equations from data by sparse identification of nonlinear dynamical systems**. 2016.

[14] Jerome Friedman Trevor Hastie, Robert Tibshirani. ***The Elements of Statistical Learning***. Springer New York, NY, 2009. doi: https://doi.org/10.1007/978-0-387-84858-7.

[15] Hector Andres Mejia Vallejo. **Too Many Features? Let's Look at Principal Component Analysis**. *Towards Data Science*, 2023.

[16] Joost J.M. van Griethuysen, Andriy Fedorov, Chintan Parmar, Ahmed Hosny, Nicole Aucoin, Vivek Narayan, Regina G.H. Beets-Tan, Jean-Christophe Fillion-Robin, Steve Pieper, and Hugo J.W.L. Aerts. **Computational Radiomics System to Decode the Radiographic Phenotype**. *Cancer Research*, 77 (21):e104–e107, 10 2017. ISSN 0008-5472. doi: 10.1158/0008-5472.CAN-17-0339. URL https://doi.org/10.1158/0008-5472.CAN-17-0339.

[17] Tanadini-Lang S. Alkadhi H. Beassler B. van Timmeren J., Cester D. **Radiomics in medical imaging—"how-to" guide and critical reflection**. *Insights Imaging*, 2020.

[18] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. **SciPy 1.0: Fundamental Algorithms for ScientificComputing in Python**. *Nature Methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2.

[19] Ying Yu, Chunping Wang, Qiang Fu, Renke Kou, Fuyu Huang, Boxiong Yang, Tingting Yang, and Mingliang Gao. **Techniques and Challenges of Image Segmentation: A Review**. *Electronics*, 12(5), 2023. ISSN 2079-9292. doi: 10.3390/electronics12051199. URL https://www.mdpi.com/2079-9292/12/5/1199.

[20] Vallières M-Löck S Zwanenburg A, Leger S. **Image biomarker standardisation initiative**. *arXiv = 1612.07003*, 2016.