Master Thesis
presented by

Karina Chiñas Fuentes

Thesis Supervisors

Prof. Dr. Geert Verdoolaege
PhD Student Joseph Hall

Erasmus Mundus Program on
Nuclear Fusion Science and Engineering Physics

June 5, 2023

FUSION ep
ENGINEERING PHYSICS
Erasmus Mundus Joint Master Degree

European Master of Science in
Nuclear Fusion and Engineering Physics

INVESTIGATING THE DEPENDENCE ON MACHINE SIZE OF THE ENERGY CONFINEMENT IN TOKAMAKS USING DATA-DRIVEN METHODS

Master Thesis
presented by

Karina Chiñas Fuentes

Thesis Supervisors

Prof. Dr. Geert Verdoolaege
PhD Student Joseph Hall

June 5, 2023

UNIVERSITEIT GENT
Universität Stuttgart
instn
uc3m | Universidad Carlos III de Madrid
CTU CZECH TECHNICAL UNIVERSITY IN PRAGUE
UNIVERSITÉ DE LORRAINE
Aix*Marseille université Initiative d'excellence
UNIVERSIDAD COMPLUTENSE MADRID

Declaration in lieu of oath

I am aware that plagiarism is not consistent with academic and research ethics. I declare in lieu of oath that all the work presented in this report, which is not cited in the text or by references, is my own work.

City, country, the

Karina Chiñas Fuentes

Copyright Agreement

I hereby grant the FUSION-EP consortium the non-exclusive right to publish this work.

I declare that this work is free of copyright claims of third parties.

City, country, the

Karina Chiñas Fuentes

ACKNOWLEDGEMENTS

I want to thank …

CONTENTS

ABSTRACT

"In this ideally *half-page* but at most one-page abstract, describe what you have done, why you did it, and what the result was. Many readers of your thesis will read the title and the abstract to decide whether it is worth their time to read any further, so take great care in writing a compelling text. Furthermore, you will be graded on the quality of your abstract."

NOTATION

ACRONYMS AND CONNOTATIONS

ELMs	Edge Localized Modes (also referred as ELMy)
H-mode	High-confinement mode
IAEA	International Atomic Energy Agency
ITER	International Thermonuclear Experimental Reactor
L-mode	Low-confinement mode
LCFS	Last Closed Flux Surface
MHD	Magnetohydrodynamics
OLS	Ordinary Least Squares
Shot	Single experimental run or discharge of a tokamak device
Subset	Reduction in number of rows, when referring to a dataset

DATABASES

DB2	Global H-confinement mode, version 2.8 (also as DB2.8 or DB2P8)
DB5	Updated DB5, version 5.2.3
STDB5	Standard criteria applied to DB5 and ELMy subset
new_DB5	Subset of STDB5, without DB2 observations
decreasing_DB5	Subset of new_DB5
small_ds	Subset of decreasing_DB5, without DB2 observations
big_ds	Subset of decreasing_DB5, without DB2 observations
clean_DB5	Modified STDB5, with a treatment in missing values and reduction of columns

GENERAL MATHEMATICS

$\mathcal{O}(\cdot)$	Big-O notation, denoting the complexity of an algorithm
∇	Vector differential nabla-operator
$\mathbf{I}_n$	$n \times n$ identity matrix
^{T}	Transpose
$\mathbb{E}[\cdot]$	Expected value of one input $\cdot$
$k(\mathbf{x}, \mathbf{x}')$	kernel function of two inputs $\mathbf{x}$, $\mathbf{x}'$
Cov[$\mathbf{x}$, $\mathbf{x}$']	Covariance of two inputs $\mathbf{x}$, $\mathbf{x}'$

$\mathrm{Var}[\cdot]$	Variance of one input $\cdot$
M	Total number of columns in dataset
n	Number of rows/observations in a dataset
$\mathbf{X}$	$n \times (M+1)$ matrix representing a database
x_i	i-th row of $\mathbf{X}$
$\mathbf{y}$	Target variable, an $n-$dimensional column vector
$\boldsymbol{\beta}$	Parametric vector, an $(M+1)-$dimensional column vector
h_i	Leverage of the i-th observation
r_i	Studentized residual of the i-th observation
$\boldsymbol{\varepsilon}$	Gaussian noise, an $n-$dimensional column vector
E	Entropy of a dataset
S_{ij}	Similarity between the x_i and x_j observations
$Z \sim \mathcal{N}(0,1)$	Random Variable Z follows a normal distribution, with mean 0 and a standard deviation 1

TOKAMAK-RELATED AND PHYSICS SYMBOLS

μ_0	Vacuum magnetic permeability	$4\pi \times 10^7$ H/m
q	Elementary charge	$1.602176634{\times}10^{19}$ C
$\bar{n}_e$	Averaged electron density	$10^{19}\mathrm{m}^{-3}$
κ_a	Elongation of the LCFS	m^3
a	Minor radius of the tokamak	m
R_{geo}	Major radius of the tokamak	m
P_{fus}	Fusion power	MW
$P_{l,th}$	Thermal power lost due to the transport through the LCFS	MW
P_α	Energy carried by the alpha particles contributing to D-T fuel	MW
P_Ω	Ohmic heating	MW
P_{aux}	Auxiliary heating power	MW
P_{rad}	radiated power	MW
B_θ	Poloidal magnetic field	T
B_t	Toroidal magnetic field	T
I_p	Plasma current	MA
$<\sigma\nu>$	Highest fusion reaction rate	$\mathrm{m}^3\mathrm{s}^{-1}$
$\tau_{E,th}$	Energy confinement time	s

$\hat{T}$	Total averaged temperature of the plasma	eV
ω_c	Ion cyclotron frequency	Hz
M_{eff}	Effective atomic mass of the plasma	amu
β_t	Plasma pressure normalized to B_t $(= 2\mu_0 \bar{n}_e \hat{T}/B_t)$	-
ϵ	Inverse aspect ratio $(= a/R_{geo})$	-
γ_{rad}	Parametrization constant for defining radiative losses $\in [0, 1]$	-
ν_*	Ion collision frequency normalized to bounce frequency of trapped particles.	-
ρ_*	Ion gyroradius normalized to a.	-
Q	Fusion Gain $(= \hat{P}_{fus}/P_{aux})$	-
q_{95}	Plasma safety factor $(= \epsilon B_t/B_\theta)$ at the 95% poloidal flux surface	-
α_x	Regression coefficient of the x variable in the scaling law for $\tau_{E,th}$	-
χ_x	Regression coefficient of the x variable in the scaling law for $\omega_i \cdot \tau_{E,th}$	-

I

INTRODUCTION

Around 1919, Rutherford conducted numerous experiments at the University of Manchester, which resulted in the first artificial nuclear reaction performed on Earth. Later, scientists started dreaming of utilizing this energy for electricity production. Initially, linear devices of magnetic confinement were studied, but quickly discarded due to their instabilities. Then, toroidal devices were analyzed. It was noted that a strong twisted magnetic field along the torus is crucial for circular confinement to avoid charge accumulation, which terminates confinement in this configuration. The creation of these particular field lines can be achieved in two devices: tokamaks and stellarators. Stellarators make use of external coils, some of which revolve helically around the plasma. In tokamaks, there is an induced net toroidal plasma current [1], [2]; Figure 1.1 shows the difference between these devices.

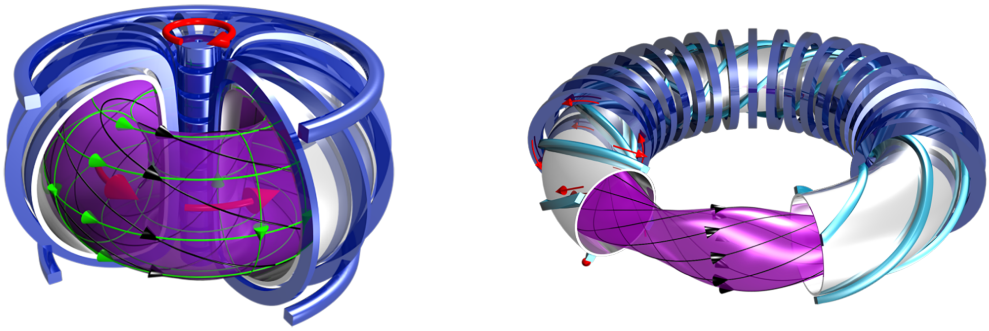

Figure 1.1: In both images, the toroidal coils are shown in dark blue, the red arrows represent a current, and the black arrows represent the required rotational transform of the magnetic field lines. LEFT: tokamak showing the poloidal magnetic field lines, as green arrows, resulting from the presence of the induced current. RIGHT: stellarator having helical coils, in light blue [2]. Images from [3].

Nowadays, tokamaks are the popular option for fusion research due to their simple geometry. The International Thermonuclear Experimental Reactor (ITER) is a megaproject composed of 35 nations dedicated to investigating tokamak physics and helping to be one step closer to employing fusion energy for peaceful usage [4].

In 1991, the International Atomic Energy Agency (IAEA) released a document indicating the conceptual design of ITER, where they mentioned two ways of predicting the energy confinement time: by codes modelling the transport in the plasma and by constructing an empirical scaling law, with the latter being the preferred option [5]. Today, this method remains the dominant approach when investigating the energy confinement time of any tokamak, mainly because a direct estimate of this variable is still unavailable, even with the significant advancements in the field of theoretical physics in tokamaks [6], [7].

1.1 Energy Confinement Time

Since the plasma experiences a decrease in turbulent transport and improved confinement is achieved during the high-confinement mode (H-mode) this technique is expected to be the nominal inductive operation for ITER; with edge localized modes (ELMs) – small periodic bursts of energy and particles ejected from the plasma that impact the plasma-facing components [8], [9].

The expression of the energy confinement time $\tau_{E,th}$ [s] results from an empirical standard power law conformed by eight plasma parameters: plasma current I_p [MA], toroidal magnetic field B_t [T], central line-averaged electron density $\bar{n}_e$ [$\cdot 10^{19}$ m $^{-3}$], thermal power lost due to the transport through the last closed flux surface (LCFS) $P_{l,th}$ [MW], major radius R_{geo} [m], elongation of the LCFS $\kappa_a = V/(2\pi R_{geo}\pi a^2) = b/a$ (with V [m^3] being the plasma volume inside the LCFS, a [m] the minor radius of the tokamak, and $2b$ [m] the height between the upper and lower plasma edge), inverse aspect ratio $\epsilon = a/R_{geo}$, and the effective atomic mass of the plasma M_{eff} [8]; see Figure 1.2 for a visual description of the plasma geometrical parameters. The scaling law of the energy confinement time is:

$$\tau_{E,th} = \alpha_0 \cdot I_p^{\alpha_I} \cdot B_t^{\alpha_B} \cdot \bar{n}_e^{\alpha_n} \cdot P_{l,th}^{\alpha_P} \cdot R_{geo}^{\alpha_R} \cdot \kappa_a^{\alpha_\kappa} \cdot \epsilon^{\alpha_\epsilon} \cdot M_{eff}^{\alpha_M}. \tag{1.1}$$

Each accompanying coefficient α_x is estimated by applying the logarithm to Eq. (1.1) and then using the ordinary least squares (OLS) regression technique. There is another version that considers the triangularity δ, another geometrical parameter that characterizes the plasma cross-section of the LCFS; however, I will not consider this parameter as it turns out to have a weak dependence for the dataset of interest [6].

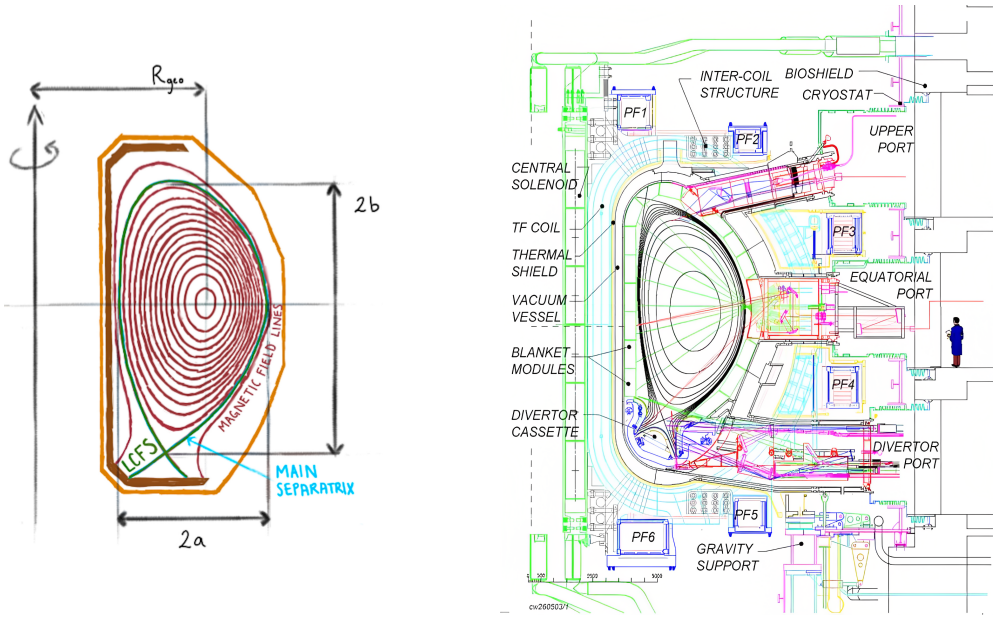

Figure 1.2: LEFT: A drawing of a simple poloidal cross-section of a tokamak is used to depict its geometrical parameters; open field lines ending in an open divertor. RIGHT: ITER poloidal cross-section is shown to illustrate a more complex plasma configuration. Imagre from [10]. It is worth noting that there is the possibility of a second separatrix at the top and that the open field lines could end in a closed divertor.

The data used to estimate the regression coefficients has been collected since 1989, by the H-mode Database Working Group. This data is referred to as the Global H-mode Confinement Database and, since 2001, it follows a framework established by the International Tokamak Physics Activity (ITPA).

Version 2.8, ELMy subset, of this database (DB2.8 [1]) was utilized to estimate ITER's physics and its design. DB2.8 is characterized by 9 devices: ASDEX, ASDEX-UPGRADE (AUG), Alcator C-MOD, DIII-D, JET, JFT-2M, JT-60U, PBXM, and PDX. It contains 1310 data points [6]. When computing the OLS to Eq. (1.1) using DB2.8, one obtains the following regression parameters

$$\tau_{E,th} = 0.05 \cdot I_p^{0.78} \cdot B_t^{0.32} \cdot \bar{n}_e^{0.43} \cdot P_{l,th}^{-0.67} \cdot R_{geo}^{2.22} \cdot \kappa_a^{0.39} \cdot \epsilon^{0.58} \cdot M_{eff}^{0.18}. \tag{1.2}$$

Nowadays, one can work with the updated version of the global H-mode confinement database DB5 version 2.3, released in 2021. This database is characterized by having 14,153 observations of 19 different devices: ASDEX, AUG, Alcator C-MOD, COMPASS, DIII-D, JET, JFT-2M, JT-60U, MAST, NSTX, PBXM, PDX, START, T-10, TCV, TDEV, TEXTOR, TFTR, TUMAN-3M [11]. When considering the ELMy subset of DB5 with the standard criteria (STDB5), composed of 6,252 observations and 18 devices (without TUMAN-3M), one obtains that the regression parameters of Eq. (1.1) are the following:

$$\tau_{E,th} = 0.079 \cdot I_p^{1.07} \cdot B_t^{0.16} \cdot \bar{n}_e^{0.19} \cdot P_{l,th}^{-0.69} \cdot R_{geo}^{1.59} \cdot \kappa_a^{0.44} \cdot \epsilon^{0.23} \cdot M_{eff}^{0.23}. \tag{1.3}$$

The standard criterion is the same applied to previous databases and it is characterized by shots having [6]:

(i) no pellet fueling,
(ii) no strong internal transport barriers,
(iii) no excessive MHD activity near the β_t-limit,
(iv) steady energy content,
(v) limited radiative power,
(vi) limited fast particle energy content, and
(vii) minimum safety factor ($q_{95} \geq 2.5$).

Considering this, when comparing Eq. (1.2) with Eq. (1.3) one can notice that, except for $P_{l,th}$, all the regression parameters change considerably. Two quick observations can be done from this: (i) the new information suggests that the energy confinement time for the ITER scenario decreases, and (ii) the major radius of a tokamak is not as influential, on the energy confinement time, as the DB2.8 suggests. The latter can be of particular interest since it has been estimated that the cost of a machine scales with $B_t^2 \cdot R_{geo}^3$ [12].

A similar technique to obtain information on the energy confinement time is with the scaling law of the dimensionless physics variables. This implies the application of the Buckingham Π-Theorem, which states that a function that expresses a physical law has the property of generalized homogeneity, meaning that it does not depend on the units of measurement [13]. The method was first applied to tokamak physics by Kadomtsev and a perpective on anomalous transport described by the Fokker-Planck equation was implemented by Connor and Taylor [6], [14]. The result is the following:

$$\omega_i \cdot \tau_{E,th} = B_t^{\chi_B} \cdot \rho_*^{\chi_\rho} \cdot \beta_t^{\chi_\beta} \cdot \nu_*^{\chi_\nu} \cdot q_{95}^{\chi_q} \cdot \kappa_a^{\chi_\kappa} \cdot \epsilon^{\chi_\epsilon} \cdot M_{eff}^{\chi_M} \tag{1.4}$$

where dimensionless is achieved through the multiplication of the ion cyclotron frequency $\omega_i = qB_t/M_{eff}$ [6], [14]. Here, $\rho_* = \rho_i/a$ is the ion gyroradius ρ_i normalized to the minor radius, ν_* is the ion collision frequency ν_{ii},

[1] For ease of reading, `this font` will be used to refer to databases only.

$$\nu_* = \nu_{ii}\left(\frac{m_i \cdot R_{geo}^3}{q\hat{T}a^3}\right)\cdot q_{95}R_{geo},$$

normalized to the bounce frequency of the trapped particles, q_{95} is the safety factor at the 95% poloidal flux surface,

$$q_{95} = \frac{2\pi}{\mu_0}\cdot\frac{a^2\kappa_a B_t}{R_{geo}I_p};$$

where I_p is in A, and m_i is the ion mass. By considering the Kadomtsev constraint, one can get the dimensionles regression parameters χ_x, from the engineering regression paramters α_x as [14]:

$$\begin{aligned}
\chi_B &= 0, & \chi_q &= \frac{\alpha_R + 3 - 4\alpha_I + 3\alpha_P - 2\alpha_n}{5(1+\alpha_P)},\\
\chi_\rho &= \frac{2(-3\alpha_R + 3\alpha_I - 9\alpha_P + \alpha_n)}{5(1+\alpha_P)}, & \chi_\kappa &= \frac{\alpha_k + \alpha_P}{1+\alpha_P},\\
\chi_\beta &= \frac{5 + 5\alpha_B - 4\alpha_R + \alpha_I + 3\alpha_P + 8\alpha_n}{5(1+\alpha_P)}, & \chi_\epsilon &= \frac{2\alpha_\epsilon - 3\alpha_R + \alpha_I - 5\alpha_P + 2\alpha_n}{2(1+\alpha_P)},\\
\chi_\nu &= \frac{-\alpha_R - \alpha_I - 3\alpha_P + 2\alpha_n}{5(1+\alpha_P)}, & \chi_M &= \frac{5\alpha_M + 3\alpha_R + 3\alpha_I + 4\alpha_P - \alpha_n}{5(1+\alpha_P)},
\end{aligned} \tag{1.5}$$

which facilitates the shift from the engineering scaling law to the dimensionless scaling law.

Various intriguing analyses can be conducted on the scaling law of the energy confinement time with the updated database. For instance, one can focus on the correlation among variables and how this affects the interpretability of the results. OLS characterizes by assuming that there is no multicollinearity in the variables and that their uncertainty is negligible with respect to the response variable [15]. Yet, this is not the case for the global H-mode confinement database. Furthermore, work has been done to obtain the regressor parameters α_x using a method that does not neglect the uncertainty in the predictor variables [6]. Others have investigated the energy confinement time with non-linear or non-power scaling forms [16], [17]. However, this project focuses on the dominant causes influencing the decrease of α_R by using data analysis and machine learning algorithms. The aim is to use these algorithms to inspect all the columns available in DB5 and determine if any of them provide insight into explaining the change in α_R.

The proposed workflow involves three steps: first, identifying a subset of records corresponding to shots that contribute to the decrease in α_R; second, applying data-driven algorithms to this subset to determine common characteristics; and third, analyzing the results of step two for interpretability in terms of tokamak physics. Figure 1.3 illustrates this idea.

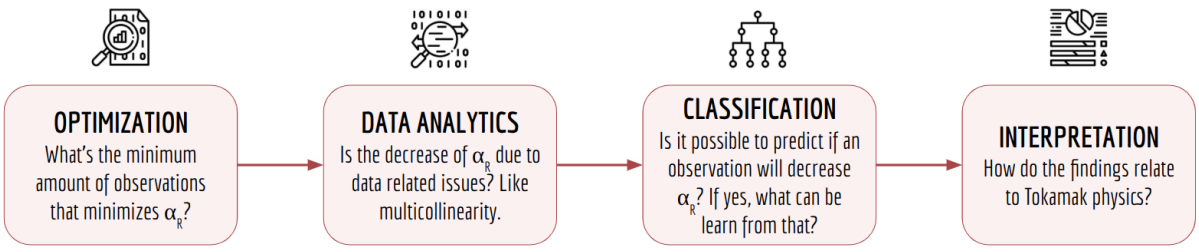

Figure 1.3: Proposed workflow to study the change of the regression parameter α_R in the updated global H-mode confinement database, standard-ELMy subset.

The structure of this report is outlined as follows: Chapter 2 provides an overview of the analyses conducted to identify the minimum subset of variables that contribute to the greatest reduction in α_R. Additionally, this chapter encompasses various regression diagnostics employed to characterize different datasets. Chapter 3 focuses on the classification analyses, including the underlying theory, and examines the variables that are most influential in the algorithms' learning process. This chapter also incorporates the information obtained from Chapter 2. Chapter 4 presents a data-analysis approach to investigate the key findings from Chapter 3 and establish connections with tokamak physics. Finally, Chapter 5 summarizes the main conclusions derived from this project. Appendix A shows the theory needed to understand statements made in Chapter 3.

II

INFLUENTIAL POINTS AND DATA ANALYTICS

This section aims to derive an effective method that identifies the observations that have a substantial influence on α_R. Previously, various algorithms, such as gradient descent, Markov chains, and simulated annealing, were implemented by [18]. The results showed subset sizes ranging from 880 to 1700 with α_R values between 1.1 and 1.2. Nonetheless, instead of subjecting the dataset to complex optimization algorithms, I decided to first assess the impact of influential points and outliers on the new registers in STDB5 using conventional regression diagnostics.

2.1 Regression Diagnostics

Diagnostics techniques make it possible to estimate the influential data and sources of collinearity that may be present in a dataset. However, I will first focus on detecting influential observations. It is common to say that a point greatly influences all regression parameters if it has both high residual and high leverage, as explained shortly [19].

Consider a dataset $\mathbf{X}$ in the form of an $n\times(M+1)$ matrix[1], where n is the total number of observations and M is the total number of variables used to predict the target variable $\mathbf{y}$, an n-dimensional column-vector. OLS is a regression model that assumes the target variable,

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \tag{2.1}$$

is a parametric function dependent on arbitrary parameters $\boldsymbol{\beta}$, a $(M+1)$-dimensional column-vector, and Gaussian noise $\boldsymbol{\varepsilon}$, an n-dimensional column-vector often referred to as irreducible error [20]. With OLS, one can predict the target variable $\hat{\mathbf{y}}$ with the optimized parameters $\hat{\boldsymbol{\beta}}$ such that $\hat{\mathbf{y}} - \mathbf{y} = \boldsymbol{\varepsilon}$ is as small as possible. Viewed as a linear algebra problem, one can find the closest vector to $\mathbf{y}$ using Euclidean distances. The solution to this problem is the orthogonal projection of $\mathbf{y}$ onto a subspace $\mathcal{R}^n$ spanned by the columns of $\mathbf{X}$ [20]. Therefore,

$$\hat{\boldsymbol{\beta}} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{y}. \tag{2.2}$$

Now, Eq. (2.2) can be used to estimate the regression parameters α_R when applying the logarithm to Eq. (1.1); namely,

$$\begin{aligned}\log(\tau_{E,th}) = \log(\alpha_0) &+ \alpha_I\cdot\log(I_P) + \alpha_B\cdot\log(B_t) +\\ \alpha_n\cdot\log(\bar{n}_e) &+ \alpha_P\cdot\log(P_{l,th}) + \alpha_R\cdot\log(R_{geo}) +\\ \alpha_\kappa\cdot\log(\kappa_a) &+ \alpha_\epsilon\cdot\log(\epsilon) + \alpha_M\cdot\log(M_{eff});\end{aligned} \tag{2.3}$$

[1] The +1 is to account for the intercept α_0 in Eq. (1.1)

thus, $\boldsymbol{\beta} = [\log(\alpha_0), \alpha_I, \ldots, \alpha_M]^T$. Having this into account, it is possible to assess the leverage and residual of each point. Leverage is a measure dependent on the mean of a predictor variable and how far an observation is from that mean. In other words, an observation will have high leverage if it is unusually far from the rest observations [21]. Mathematically, the leverage h_i of the i-th observation is

$$h_i \equiv \mathbf{x}_i \left(\mathbf{X}^T \mathbf{X}\right)^{-1} \mathbf{x}_i^T, \tag{2.4}$$

with $\mathbf{x}_i$ denoting the i-th row-vector of the dataset $\mathbf{X}$ [19]. It is common to state that h_i is high if $h_i > (2M+2)/n$ [21]. The residual is a measure that states how much is a prediction different to the actual target value. There are various ways to estimate this, but here the studentized deleted residual r_i is utilized. The idea is to obtain $\hat{\mathbf{y}}_{i(-i)}$, which denotes the prediction on the i-th observable when this one is removed from the data $\mathbf{X}$ when computing Eq. (2.2); then,

$$r_i \equiv \frac{1}{s_i}\left(\mathbf{y}_i - \hat{\mathbf{y}}_{i(-i)}\right), \tag{2.5}$$

where s_i^2 is the estimated variance of $\mathbf{y}_i - \hat{\mathbf{y}}_{i(-i)}$ [22]; with this, it is common to see that the i-th observation is said to have high residual if $|r_i| > 3$; however, one might want to be more strict and reduce the threshold to a more appropriate value to what is being observed with the data [21].

One can also get an insight into the influential points over specific variables by computing the DFBETAS. The idea is to look at the change in regression coefficients when the i-th row is removed and compare the j-column with the original dataset [19], [23]. The estimate is defined as

$$\text{DFBETAS}_{ij} \equiv \frac{\beta_j - \beta_{j(-i)}}{s_i \sqrt{\mathbf{X}^T \mathbf{X}^{-1}_{jj}}}; \tag{2.6}$$

and, it is common to consider that the i-th point is influential on the j-variable if $\text{DFBETAS}_{ij} > 2/\sqrt{n}$; however, this is not a perfect threshold and one can adjust it according to the data [23]. Now, it is useful to have a look at Figure 2.1 showing the Venn Diagram of the databases that will be worked through the project.

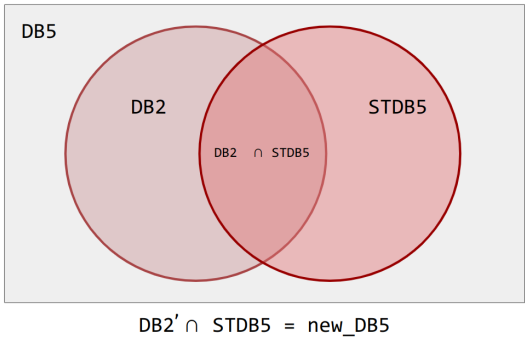

Figure 2.1: Venn Diagram showing the naming of the databases for an easier interpretation.

To determine which observations will undergo regression diagnostics, one new point at a time was selected from `new_DB5` and added to `DB2` to then perform OLS. With this, it is possible to identify the individual points that resulted in a decrease in α_R. The result is shown in Figure 2.2.

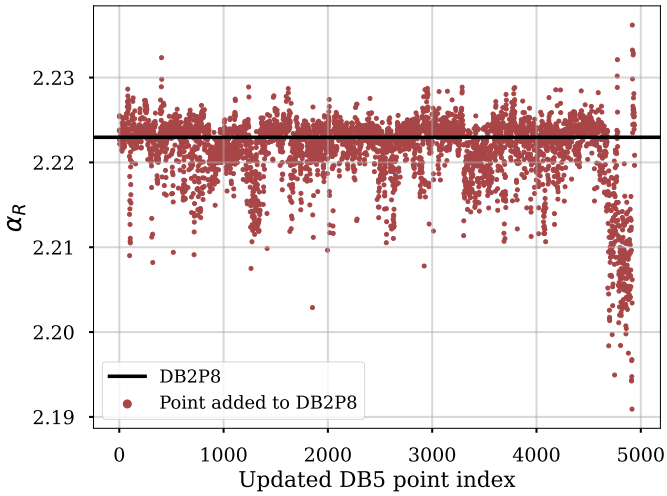

Figure 2.2: Variation of α_R given the individual presence of a point in `new_DB5`. The black line shows the value of α_R when only `DB2` is considered. This black line is referred to as *baseline*. The points below the baseline are referred to as `decreased_DB5`. When taking only the `decreased_DB5` with DB2, $\alpha_R = 1.05$.

One can thus focus on the observations that are below the baseline; I will refer to these points as `decreased_DB5`. This database is characterized by 2,544 observations conformed by 12 devices: START, NSTX, MAST, JET, D3D, AUG, AUG with ITER-like-walls (AUGW), JET with ITER-like-walls (JETILW), JT60U, TFTR, COMPASS, and CMOD; with shots done from 1989 to 2017. Figure 2.3 shows the results of applying the regression diagnostics to `decreased_DB5` along with `DB2`.

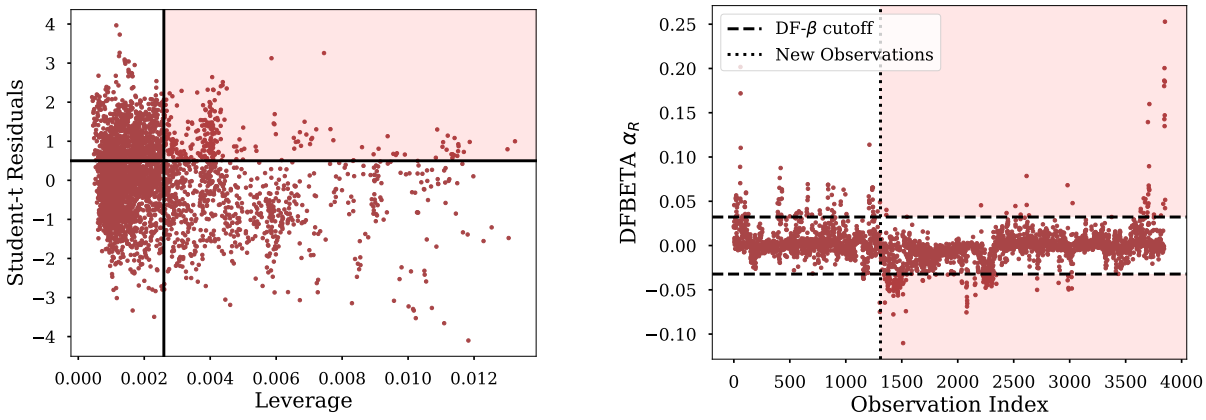

Figure 2.3: LEFT: Influential points to all regressors shown in the pink box according to the concept discussed above. The horizontal and vertical black lines show the cutoff of the Student-t residuals and leverage, respectively. RIGHT: Influential points to α_R shown in the pink areas according to the DFBETAs. `DB2` along with `decreased_DB5` were studied. See [24] for code.

In this section, we observed that influential points, as defined earlier, can play a key role in determining the observations that increase α_R. To illustrate this, we evaluated the OLS model on two subsets of data points based on their positions in the leverage-residual plot (Figure 2.3, left): one in the pink area and the other in the white area; the resulting values of α_R were 1.86 and 1.03, respectively. Similarly, when we analyzed the DFBETAs plot (Figure 2.3, right), the values of α_R for the pink and white areas were 1.46 and 1.15, respectively. These results suggest that conventional regression diagnostics may not be entirely reliable in identifying the points that have a significant impact on the reduction of α_R in the confinement time scaling law, especially when using DFBETAs.

2.2 Random Sampling in Decreasing Subset

Another attempted approach, which proved to be the most efficient in terms of both computational resources and achieving the research goal, is random sampling. The idea behind this method is to add the points in `decreasing_DB5` to DB2 in a cumulative manner, without removing the previously added points. This routine allows observing the impact of point grouping on the value of α_R. Figure 2.4 presents the result of this procedure, as well as a close-up view of it.

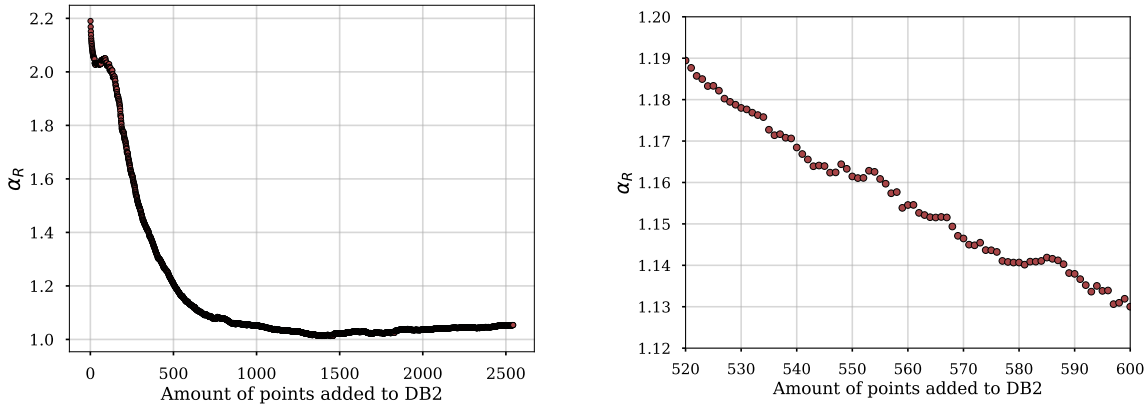

Figure 2.4: LEFT: Change in α_R due to the cumulative behavior of points in `decreasing_DB5`. RIGHT: close-up of the left figure showing that although the added points individually decrease α_R, collectively they increase it.

It is possible to observe that some of the points increase α_R although the overall trending is the reduction of it. From this, one can label the points that increase and decrease α_R in `decreasing_DB5` and then: (i) do a random sampling for all points in the dataset with a higher tendency in picking the decreasing points, and (ii) do a random sampling only of the decreasing points within `decreasing_DB5` to then subject them to OLS, along with DB2. The sampling is performed multiple times, with different sampling subsets, each subset contains unique observations; this is known as bootstrap without replacement [20]. Algorithm 1 depicts the general idea of what was done for both situations.

Algorithm 1 Random sampling to get representative subsets of influential points.

Require: $seeds$ = list with 500 different seeds.
  for $seed$ in $seeds$ **do**
    for $subset\_size$ in range(1, $length(dataset)$) **do**
      $subset \leftarrow$ sampling(`decreased_DB5`, with $random\_seed = seed$)
      $data \leftarrow subset +$ DB2
      $coefficients \leftarrow$ OLS($data$)
      $alpha\_R \leftarrow coefficients$[position = 5]
      plot($alpha\_R$ vs $subset\_size$)

This approach allows for the generation of representative groups that provide different estimates of α_R based on the subset size. Figure 2.5 shows the resulting groups, along with a close-up of the points of interest after implementing Algorithm 1.

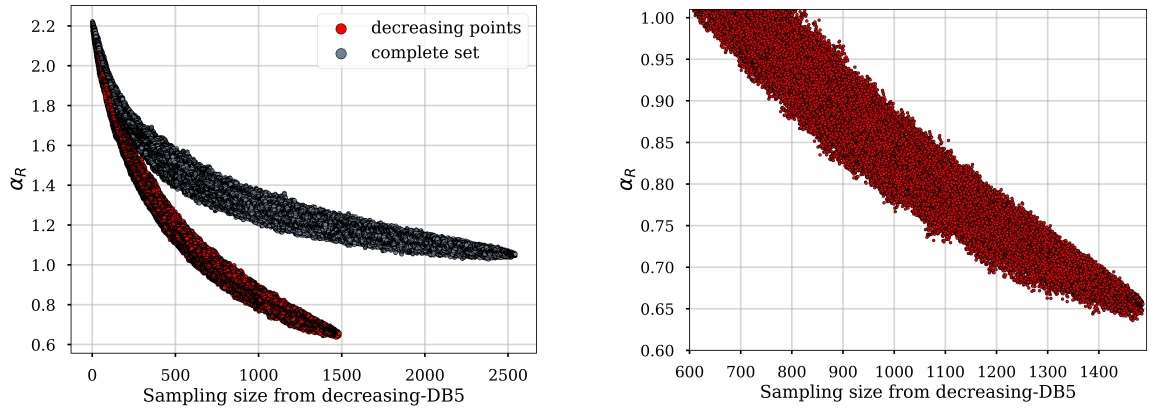

Figure 2.5: LEFT: Change of α_R when sampling different subset-sizes from `decreasing_DB5`. Each of the subsets is added to DB2 to then compute OLS and obtain the respective α_R. RIGHT: close-up to the points of interest: $\alpha_R < 1$.

One can quickly appreciate that the random sampling of the decreasing points in `decreasing_DB5` established smaller values of α_R with smaller subset sizes compared with the random sampling on all points in the dataset. Some of the results that might be of interest to study are shown in Table 2.1

Table 2.1: Characteristic results of random sampling from the decreasing points in `decreasing_DB5`.

	Smallest Subset for $\alpha_R < 1$	Smallest α_R for subset <1400	Smallest α_R found
α_R	0.9998	0.6439	0.6357
Subset size	618	1388	1466
Observations decreasing α_R	9.88%	22.20%	23.45%
Dataset Name	`small_ds`	-	`big_ds`

Now, the question is: out of all these dataset subsets of STDB5, which one will provide the most informative insights into the factors causing the decrease in α_R? If I were to consider only the subsets obtained by random sampling of the decreasing points in `decreasing_DB5` with an $\alpha_R < 1$, I would need to study a total of 382,495 subsets. However, to streamline the analysis, I will focus initially on the smallest values of α_R and the smallest subset sizes for which $\alpha_R < 1$, since this research aims to discern which columns can yield information into what is decreasing α_R.

2.3 Multicollinearity

One encounters a multicollinearity[2] issue when a regression algorithm, such as OLS, struggles to distinguish the effects of one variable from another on the target variable [25]. This issue arises due to significant inter-correlations among the predictor variables [26]. Strong multicollinearity has two main effects: an increase in the standard errors of each coefficient in the model and numerical instability [19], [25]. These consequences might lead to unstable parameter estimation, unreliable models, and weak predictive ability [27]. In other words, the presence of low or moderate multicollinearity does not necessarily imply an issue. However, evaluating the extent of multicollinearity in a dataset will determine whether the model effectively explains how the target variable (in our case, $\tau_{E,th}$) is influenced by each variable

[2] Also referred to as *ill conditioning* in numerical analysis [19].

[25] (I_p, B_t, $\bar{n}_e$, $P_{l,th}$, R_{geo}, κ_a, ϵ, and M_{eff}). To determine the presence of multicollinearity in the diverse available datasets, I will consider the variance inflation factor (VIF), condition index, and variance decomposition.

Variance Inflation Factor (VIF)
The method consists in computing the coefficient of determination R_k^2 of the $k-$th parameter when regressing it against the remaining explanatory variables [19], [27]. Generally speaking the coefficient of determination R^2 is computed as [19]

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2}, \tag{2.7}$$

where $\bar{y}$ is the mean value of the target vector. All the other symbols remain with the same interpretation as discussed above. The VIF associated to the $k-$th regressor parameter is therefore [27]

$$VIF_k = \frac{1}{1 - R_k^2}. \tag{2.8}$$

In the ideal case, $VIF_k = 1$, indicating that there is no collinearity between the $k-$th regressor parameter and the other variables in the model. When $VIF_k \geq 10$; then, on might have a multicollinearity issue [27]. However, one of the weaknesses of this procedure is that it does not effectively distinguish several coexisting near dependencies [19].

Condition Index
To compute the condition index of all regressor parameters in $\mathbf{X}$ it is necessary to (i) scale $\mathbf{X}$ to have equal (unit) column length and (ii) perform the singular-value decomposition to the scaled (non-centered) matrix [19]. The decomposition is performed as $\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^T$; where[3] $\mathbf{U}^T\mathbf{U} = \mathbf{V}^T\mathbf{V} = \mathbf{I}_M$, and $\mathbf{D}$ contains the nonegative *sigular values, or principal components,* $\boldsymbol{\mu}$ and it is diagonal [19]. The obtained singular values are the squareroots of the eigenvalues of the covariance matrix; thus, they provide information on the amount of variance explained by each principal component in the data [19]. Furthermore, they are ordered based on the amount of variance they explain. When a high condition number (not to be confused with the condition index) is observed, variables associated with higher principal components are considered of interest when explaining multicollinearity patterns within that dataset [19], [28]. The condition number of the matrix $\mathbf{X}$ is computed as

$$c(\mathbf{X}) = \frac{max(\boldsymbol{\mu})}{min(\boldsymbol{\mu})} \geq 1. \tag{2.9}$$

The condition index η_k of the $k-$regressor parameter is obtained as [19]

$$\eta_k = \frac{max(\boldsymbol{\mu})}{\mu_k} \quad k = 1, \ldots, M. \tag{2.10}$$

[3] The shape the elements are: $\overset{n\times M}{\mathbf{X}} = \overset{n\times n}{\mathbf{U}} \quad \overset{n\times M}{\mathbf{D}} \quad \overset{M\times M}{\mathbf{V}^T}$. In this context, M also accounts for the intercept; hence, $M = 9$ when assessing the scaling law of the energy confinement time.

Notice that the largest condition index is the condition number of the matrix $\mathbf{X}$. These two values are of great interest when assessing multicollinearity in a dataset, because if $c(\mathbf{X}) \sim 5$ or 10; then, there might be weak dependencies in a dataset. If $c(\mathbf{X}) \sim 30$ or 100; then, there might be moderate to strong dependencies in a dataset. Furthermore, one can also think that there are as many near dependencies as high condition indexes [19].

Variance Decomposition and Associated Π Matrix

Over the presence of high condition indexes, it is recommended to perform the variance decomposition over the variance-covariance matrix $\mathbf{V}(\hat{\boldsymbol{\beta}})$ of the least-squares estimator $\hat{\boldsymbol{\beta}}$ to estimate the damage caused by multicollinearity over the regression estimates [19]. This is done by decomposing $\mathbf{V}(\hat{\boldsymbol{\beta}})$ as [19], [28]

$$\mathbf{V}(\hat{\boldsymbol{\beta}}) = \sigma^2 \left[\left(\mathbf{U}\mathbf{D}\mathbf{V}^T \right)^T \left(\mathbf{U}\mathbf{D}\mathbf{V}^T \right) \right]^{-1} = \sigma^2 \mathbf{V}\mathbf{D}^{-2}\mathbf{V}^T. \tag{2.11}$$

For the k-th component of $\hat{\boldsymbol{\beta}}$,

$$Var[\beta_k] = \sigma^2 \sum_j \frac{V_{kj}^2}{\mu_j^2}. \tag{2.12}$$

Here, σ^2 is the common variance of the components of $\boldsymbol{\varepsilon}$ in Eq. (2.1), μ_j's are the singular values, and V_{kj} is the (k, j) element in $\mathbf{V}$ [19]. Eq. (2.12) allows identifying unusually high proportions of variance between two or more coefficients by having the singular values in the denominator. From this reasoning, it follows the idea of defining the $k, j-$th *variance-decomposition proportion* Π_{jk} as the proportion of variance corresponding to the k-th regressor, related by the $j-$th component of its decomposition in Eq. (2.12) [19]. This influence is observed through the $\mathbf{\Pi}$ matrix, which is calculated as

$$\Pi_{jk} = \frac{\phi_{kj}}{\sum_{j=1}^{M} \phi_{kj}}, \tag{2.13}$$

where $\phi_{kj} = V_{kj}^2/\mu_j^2$. The ideal case, where no multicollinearity exists in the dataset, would behave as, $\Pi_{ij} = \delta_{ij}$, a Kronecker delta. In other words, the $Var[\beta_k]$ would not be affected by near dependency among the columns of $\mathbf{X}$ [19]. The degradation of a regression estimate due to multicollinearity is evinced when a singular value μ_j associates with two or more coefficients with considerable proportions. The regressors that isolate a singular value indicate that they do not have a collinear depedence with respect to others [19].

2.3.1 Assessing Dependencies in Datasets

In this subsection, I will present the results of conducting the aforementioned multicollinearity tests on four datasets: `DB2`, `DB5`, `small_ds`, and `big_ds`. For a comprehensive analysis of `small_ds` and `big_ds`, these datasets have been divided into two subsets: decreasing and unaffected. Each subset contains the respective observations for its specific case. The corresponding labels, and visual representation, for each case, are displayed in Figure 2.6.

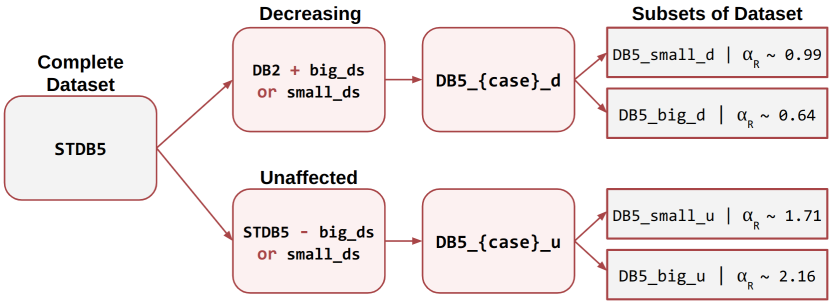

Figure 2.6: Diagram showing the split of `small_ds` and `big_ds` into creating two datasets containing the observations that decrease or does not affect α_R. Notice that all subsets contain all observations of **DB2** in them.

The obtained VIF values for all regressors, categorized by dataset, are presented in Figure 2.7 (left). According to this metric, it is evident that in all cases, the plasma current is a significant contributor to collinearity, followed by the outer radius. When the plasma current is excluded from the dataset, all variables exhibit low VIF values. However, the variable with the highest VIF value shifts from the major radius to the thermal power loss. See Figure 2.7 (right).

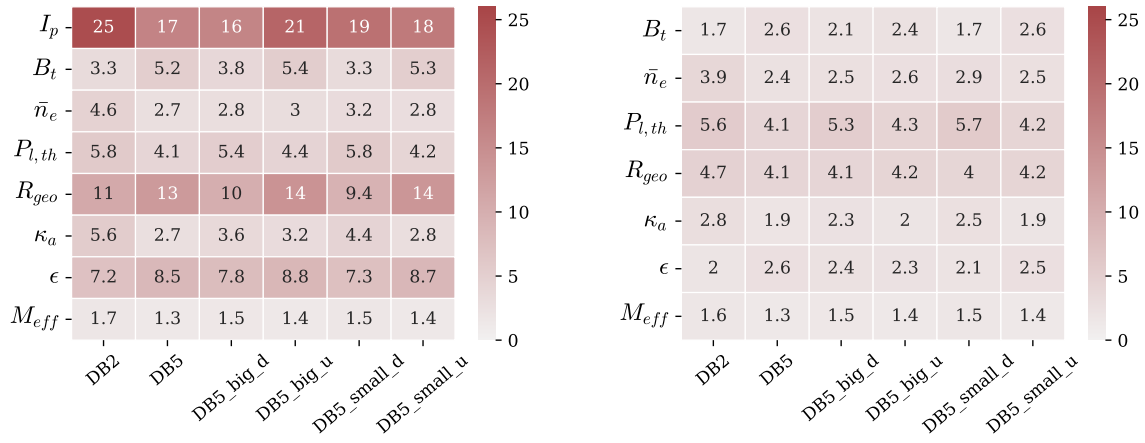

Figure 2.7: LEFT: VIF scores of all regressors. RIGHT: VIF scores of regressors after removing the plasma current. It is considered that a high $VIF \geq 10$.

Furthermore, the condition number and the Π matrix was computed for all datasets. All cases showed a high condition number, meaning that all of them are likely to show multicollinearity issues. Moreover, all datasets showed three condition indexes ranging [10,20], one condition index ranging [25,35], and one (also their condition number) ranging [46, 58].

Figure 2.1 shows the results implemented to DB2 and DB5. In both datasets, the effective mass showed an isolated principal component, meaning that this variable is not collinear with the others. Almost the same behaviour is observed for the intercept, except that in DB5, its variance is almost split in half by two principal components. All other variances share various principal components. However, the singular value μ_8 accounts for more than 60% of $Var(I_p)$, $Var(B_t)$ and $Var(\epsilon)$, with the latter being 99% in both DB2 and DB5.

Figure 2.9 show the results for the two subsets resulting from `big_ds`. In this subset, the variance of the effective mass is isolated to mostly one principal component. A similar pattern is observed for μ_8, where it explains a significant portion of the variance in I_p, B_t, $\bar{n}_e$, $P_{l,th}$, R_{geo}, κ_a, and primarily ϵ. Notably, in the unaffected dataset compared to the decreasing dataset, the influence of μ_8 increases.

Figure 2.10 presents the results of the analyses conducted on `small_ds`. A similar pattern is observed, although the increase in influence for μ_8 is not as significant as in the previous case. However, it is worth noting that the percentage of variance explained by κ_a drops from approximately 30% in the decreasing dataset to around 15% in the unaffected dataset.

Overall, similar patterns and behaviours are observed across all datasets, including the total number of high η_k present in the data.

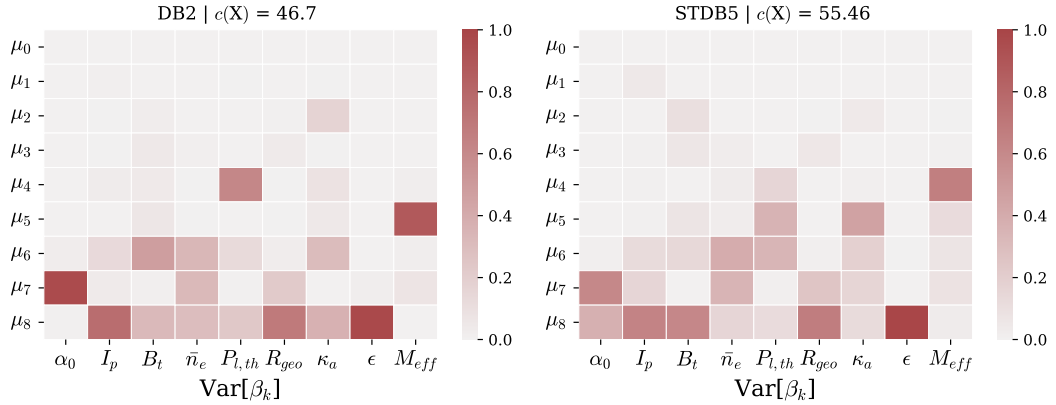

Figure 2.8: Π matrix for DB2 (left) and DB5 (right) with their corresponding condition number $c(\mathbf{X})$, for all variances associated to the regressor variables in Eq. (2.1).

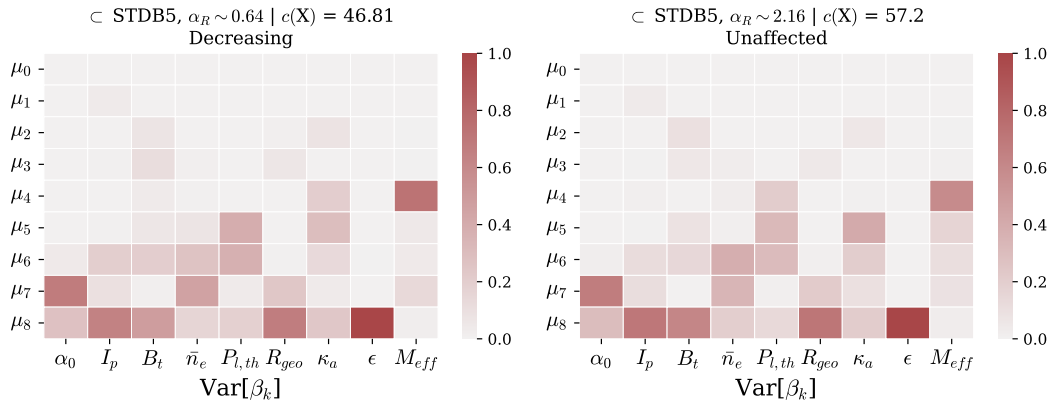

Figure 2.9: Π matrix for observations that decrease α_R (left) and does not affect α_R (right), based on `big_ds` with their corresponding condition number $c(\mathbf{X})$, for all variances associated to the regressor variables in Eq. (2.1).

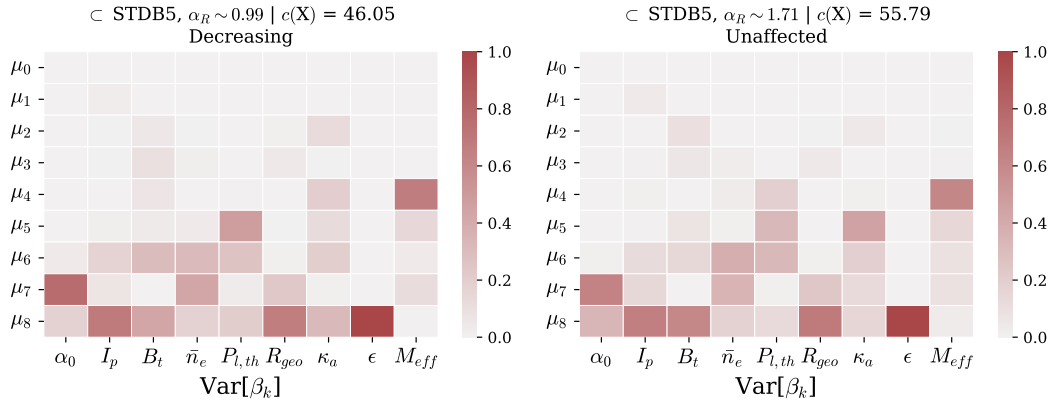

Figure 2.10: Π matrix for observations that decrease α_R (left) and does not affect α_R (right), based on `small_ds` with their corresponding condition number $c(\mathbf{X})$, for all variances associated to the regressor variables in Eq. (2.1).

2.4 Model Comparison

Multivariate models tend to be updated when more data is available. One can use statistical tests to compare models and assess the performance of the multivariate model given the new data, removal or addition of variables [29]. In this section, I will use the F-test to compare nested models; these are models where one is a special case of the other, for instance, a reduced model from a full model [30]. Here, our full model is described by Eq. (2.1) in the log-space, and the reduced model is the removal of at least one of its variables. To compute the F-statistic, one needs the sum of square error (SSE), being

$$SSE = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2, \tag{2.14}$$

one can then compute its value for the specific case to obtain

$$\Delta SSE = SSE_r - SSE_f. \tag{2.15}$$

Here, the subscripts r, f denote the reduced and the full model, respectively. The mean square error (MSE) is the SSE divided by the number of observations n in the dataset. The F-statistic, or F-value, is then

$$F = \frac{1}{\Delta M} \cdot \frac{\Delta SSE}{MSE_f}, \tag{2.16}$$

where ΔM is the number of coefficients being tested, also referred to as the numerator degrees of freedom ν_1. The denominator degrees of freedom is $\nu_2 = n - M$. These two are important when computing the P-value [30]. The P-value, denoted as $Pr(F_{\nu_1,\nu_2} \geq F)$, represents the probability of observing the obtained F-value based on the F-distribution with degrees of freedom ν_1, and ν_2. Usually, it is considered that having $F \geq 2.5$, or $P \geq 0.05$, is enough to consider that the reduced model is statistically significantly different than the full model and that it performs better. This type of analysis is also referred to as ANOVA (analysis of variance) test [31]. Table 2.2 show the F- and P-values for DB2, and STDB5. Table 2.3 show the same results for the two subsets that follow from `small_ds`, and `big_ds`, as explained in Figure 2.6.

Table 2.2: ANOVA test applied to DB2, and STDB5.

	DB2		**DB5**	
Removed	**F**	**Pr(>F)**	**F**	**Pr(>F)**
I_p	961.58	0.0	4809.57	0.00
B_t	138.99	0.0	25.79	0.00
$\bar{ne}$	494.28	0.0	332.97	0.00
Pl, th	2674.97	0.0	10024.60	0.00
R_{geo}	2022.43	0.0	3221.36	0.00
κ_a	94.67	0.0	94.70	0.00
ϵ	131.14	0.0	2.96	0.09
M_{eff}	36.51	0.0	157.27	0.00

Table 2.3: ANOVA test applied to subsets of STDB5 that result in cases where observations decrease or do not affect α_R w.r.t. α_R in DB2.

	Decreasing $\alpha_R \sim 0.64$		Unaffected $\alpha_R \sim 2.16$		Decreasing $\alpha_R \sim 0.99$		Unaffected $\alpha_R \sim 1.71$	
Removed	**F**	**Pr(>F)**	**F**	**Pr(>F)**	**F**	**Pr(>F)**	**F**	**Pr(>F)**
I_p	4165.11	0.00	2370.15	0.0	2149.90	0.000	3757.01	0.0
B_t	1.011	0.32	300.46	0.0	18.89	0.000	69.15	0.0
$\bar{n}_e$	85.69	0.00	1971.23	0.0	1.23	0.269	752.88	0.0
$P_{l,th}$	2017.40	0.00	14937.77	0.0	1609.65	0.000	11206.25	0.0
R_{geo}	294.38	0.00	7450.69	0.0	471.16	0.000	4404.12	0.0
κ_a	0.002	0.97	350.36	0.0	10.45	0.001	144.63	0.0
ϵ	136.93	0.00	507.59	0.0	11.20	0.001	77.07	0.0
M_{eff}	25.50	0.00	182.73	0.0	15.86	0.000	179.70	0.0

The results indicate that, as anticipated, no variable should be eliminated when using the DB2 dataset. However, for STDB5, it may be acceptable to remove the inverse aspect ratio without significantly degrading the model's performance. For the other subsets of STDB5, it is observed that removing the elongation and toroidal magnetic field may not have a negative impact on the model when utilizing DB5_big_d. In the case of DB5_small_d, only the averaged-line electron density should be considered for removal. For the remaining unaffected datasets, all variables should be retained. Furthermore, when the elongation is removed from the subset with $\alpha_R \sim 0.64$, the value of $Pr(> F)$ increases from 0.32 to 0.42. When both κ_a and B_t are removed from the same subset, all the other variables maintain a $Pr(> F)$ value of 0. It is interesting to note that none of the subsets suggests that the removal of the major radius may not degrade the model.

In this chapter, I have demonstrated how bootstrapping sampling can be used to find representative subsets that decrease α_R, without the need for complex optimization algorithms. Conventional regression diagnostics were also implemented but failed to discern the observations that reduced α_R. The smallest α_R found through sampling was 0.6357, with a subset size of 1466. It is worth noting that the order of points being added to DB5 from decreasing_DB5, to create Figure 2.4 (left), does not influence the overall result, but it does change the shape of the plot. Furthermore, I have examined the presence of multicollinearity among different subsets and investigated whether it is possible to remove a variable from the model presented in Eq. (2.1) without detrimental effects, given the updated observations. The analyses revealed that R_{geo} plays a crucial role in estimating $\tau_{E,th}$, and it was found that multicollinearity may not be solely responsible for the decrease in α_R when utilizing STDB5. This is evident from the fact that the division of the dataset into decreasing and unaffected subsets did not exhibit a significant reduction in multicollinearity; this absence of variability within the split datasets emphasizes the need to explore potential physical factors contributing to the decrease in α_R.

III

MACHINE LEARNING ALGORITHMS

Machine learning refers to computer programs that learn and make decisions based on data, regardless of the field of application; this sets it apart from traditional data analysis [20]. Machine learning can be used to guide research when mathematical expressions are not available or when data is limited. For instance, in drug discovery, these algorithms are used to identify molecules with specific properties, such as binding affinity to a target protein, even when the underlying mechanism of action is not well understood [32]. Astronomy [33] and neuroscience [34] are other fields where machine learning is implemented to recognize patterns and correlations in data, leading to new areas of research or refinement of existing theories. In this chapter, I will use supervised machine learning algorithms to investigate the causes that lead to a decrease in α_R. This will be based on the subsets obtained in the previous chapter.

3.1 Feature Selection

One of the challenges when implementing any machine learning algorithm is feature selection, the process of deciding which columns of a dataset should be used so that the algorithm can learn better and faster from them. Moreover, identifying and extracting the right features becomes more complex as the number of columns in the dataset increases, since adding or removing them could reduce the algorithm's learning quality [20]. To address this issue, I have (i) identified which of all the variables available in DB5 can help understand why there is a reduction in α_R, (ii) utilized the concept of entropy in information theory to identify which of these variables contain more information, (iii) considered relevant research to decide on a set of variables that will yield more insight into the reduction of α_R, and (iv) studied the different resultant subsets for multicollinearity.

3.1.1 All Variables of Interest

In DB5, there are 192 columns containing information on the tokamaks, including divertor and wall materials, plasma characteristics, heating mechanisms, instabilities, impurities, power, and more [11]. However, since I am using the standard version, not all of the columns have complete values, and not all of the columns are relevant. Tables 3.1 to 3.7 show the description [11] of 47 variables that I have considered to be interesting for this project, based on the knowledge I have gained through this master programme. In all tables, **bold features** represent the variables provided by the research group, not found in [11], and underline features are categorical data.

To clean STDB5, I have applied the pipeline shown in Figure 3.1. The cleaned dataset, which includes the 47 variables of interest and all rows from STDB5, will be referred to as `clean_DB5`. In subsequent references to `small_ds` or `big_ds` (see Table 2.1), their cleaned versions will be used.

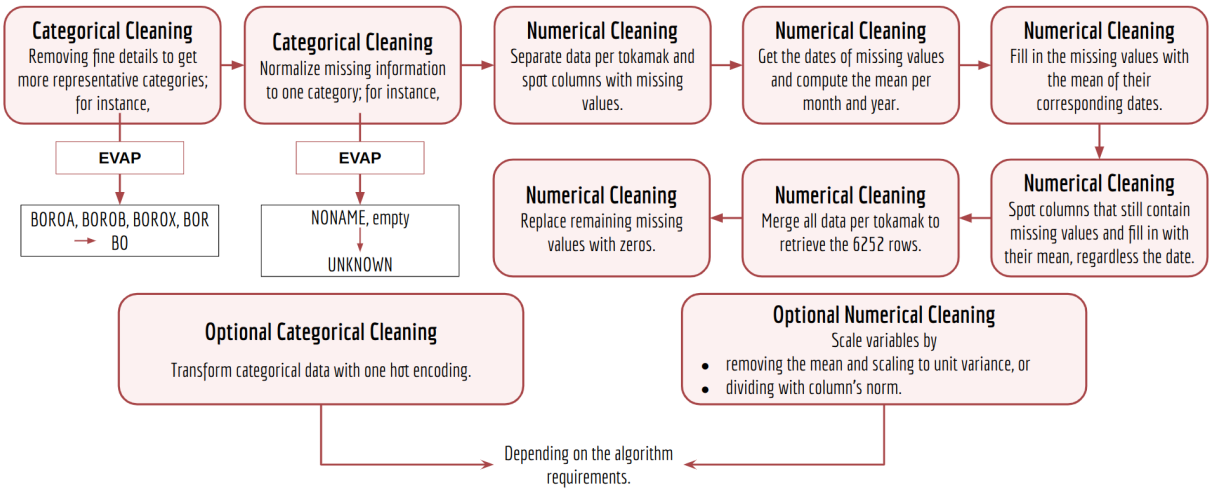

Figure 3.1: Pipeline showing the criteria to clean STDB5. The one-hot encoding creates a binary vector, for each category in the column, with 1 if the observation has that observation and 0 otherwise [35].

Table 3.1: Variables of interest regarding **plasma characteristics**.

Feature	Description	Feature	Description
BEIMHD	Beta Shafranov	MEFF	Effective atomic mass
BETASTAR	Plasma pressure normalised to B_t	**NUSTAR**	normalised Ion collision frequency
CONFIG	Plasma configuration	PREMAG	Flag for startup: with or without pre-magnetization current
DWDIA	Time rate of change of the total plasma stored energy	Q95	Plasma safety factor at the 95% poloidal flux surface
EPS	Inverse aspect ratio	**RHOSTAR**	normalised Ion Gyroradius
IP	Plasma current	TORQ	Torque on plasma due to NBI
KAREA	Plasma elongation	VOL	Plasma volume
LCOULOMB	Coulomb Logarithm $\log_e(\Lambda) = 30.9 - \log_e(\bar{n}_e^{0.5}/\hat{T})$	WMHD	Total plasma energy (MHD equilibrium)

Table 3.2: Variables of interest regarding **heating**.

Feature	Description	Feature	Description
AUXHEAT	Type of auxiliary heating	PELLET	Pellet material if injected
ECHMODE	Mode of ECRH waves	PICRH	ICRH power absorbed by the plasma
ENBI	Neutral beam energy weighted by power	PNBI	Total NBI power minus shine through
ICSCHEME	ICRH heating scheme	POHM	Total Ohmic Power
PECRH	ECRH power absorbed by the plasma	PRAD	Total radiated power as measured by Bolometer

Table 3.3: Variables of interest regarding **impurities**.

Feature	Description
EVAP	Evaporated material to cover the vessel
ZEFF	Line average plasma effective charge, from Bremsstrahlung
ZEFFNEO	Plasma effective charge, from neo-classical resistivity

Table 3.4: Variables of interest regarding **tokamaks' characteristics**.

Feature	Description	Feature	Description
AMIN	Horizontal plasma minor radius	LIMMAT	Limiters' material
BT	Vacuum toroidal magnetic field at R_{geo}	TOK	Tokamak's name
DIVMAT	Material of divertor tiles	WALMAT	Walls' material

Table 3.5: Variables of interest regarding **power loss** and **ELMs**.

Feature	Description	Feature	Description
BEIMHD	NBI power that is lost from the plasma through charge exchange and unconfined orbits	ELMTYPE	Type of ELMs
PLTH	Estimated loss power corrected for charge exchange and unconfined orbit losses	ELMFREQ	ELM frequency

Table 3.6: Variables of interest regarding **temperature**.

Feature	Description
TAV	Total volume average temperature
TEV	Total volume averaged electron temperature
TIV	Total volume averaged ion temperature

Table 3.7: Variables of interest regarding **particles**.

Feature	Description	Feature	Description
NEL	Central line average electron density	**OMEGACYCL**	Ion Cyclotron Frequency $\omega_i = qB_t/M_{eff}$
NESOL	Electron density in scrape-off layer	WFFORM	Total fast ion energy due to NBI
WFICFORM	Total fast ion energy due to ICRH estimated from approximate formula		

While all these variables may contain valuable information, it does not necessarily mean that an algorithm will perform optimally when utilising all of them. In fact, there are instances where it is desirable to reduce the number of features and only include the most representative ones. This process is often referred to as feature engineering. Additionally, it has been observed that as the number of features in a model increases, the number of required observations grows exponentially in order to achieve satisfactory performance. This phenomenon is commonly known as the "curse of dimensionality" [36]. Therefore, it becomes crucial to identify a subset of the relevant features that can potentially reduce the model's complexity and improve its overall performance.

3.1.2 Entropy Variables

By considering features as random variables, it is possible to use the concept of entropy in information theory to estimate the amount of order or information in a feature [37]. The entropy of a random variable Z is defined as [38]:

$$E(Z) = -\sum_{z} p(Z = z) \cdot \log(p(Z = z)). \tag{3.1}$$

Here, $p(Z = z)$ represents the prior probability of z. The units of entropy depend on the logarithm being used, if it is $\log_2$ then the units are bits, which represent the number of bits[1] required to characterize the random event [40]. However, there is no specific expression for the probabilities of all the features of interest, it is possible to approximate Eq. (3.1) using the similarity of observation, which directly depends on the distance between them [37]. To estimate the similarity between two observations S_{ij} for numerical data, euclidean distance is implemented as

$$S_{ij} = \exp\left(-\gamma \cdot D_{ij}\right), \quad D_{ij} = \left[\sum_{k=1}^{M} \left(\frac{x_{ik} - x_{jk}}{max(F_k) - min(F_k)}\right)^2\right]^{1/2}, \tag{3.2}$$

where D_{ij} being a $n \times n$ matrix. Here, F_k represents the column vector belonging to the k-th feature, out of the total M features. And, x_{ik} (or x_{jk}) is the i-th (or j-th) row, and the k-th column in $\mathbf{X}$. Finally, γ is a parameter which can be tuned according to the problem at hand. It is common to set $\gamma = 0.5$ [37]. For categorical features, the similarity is computed through the Hamming distance[2] as [37]:

$$S_{ij} = \frac{1}{M} \sum_{k=1}^{M} \delta_{ij}(x^k); \quad \text{with } \delta_{ij}(x^k) = \begin{cases} 1, \text{ if } x_i^k = x_j^k \\ 0, \text{ if } x_i^k \neq x_j^k \end{cases}. \tag{3.3}$$

From this, it is possible to evaluate the entropy of a dataset[3] as [37]:

$$E = -\sum_{\substack{i,j=1 \\ i \neq j}}^{N} \Big[S_{ij} \log(S_{ij}) + \left(1 - S_{ij}\right) \log(1 - S_{ij})\Big]. \tag{3.4}$$

To estimate the importance of a feature's presence in a dataset in terms of information, one can compute the entropy of the dataset with the missing k-th feature E_{-F_k} and compare this instance to another one missing a different feature. For example, if $E_{-F_1} > E_{-F_2}$, one can say that feature 1 is more important than feature 2, as the former imposes more structure in the database [37]. The idea shown in Algorithm 2 to rank the feature importance of `small_ds` and `big_ds`, was implemented.

Algorithm 2 Ranking features' importance based on entropy.

Require: $entropy\_features$ = empty onde-dimensional array of size M
  for $feature$ in columns(`clean_dataset`) **do**
    $data \leftarrow$ `clean_dataset` without $feature$
    $entropy\_k \leftarrow get\_entropy(data)$
    $entropy\_features$.append($entropy\_k$)
  $ordered\_features \leftarrow$ sort_values($entropy\_features$)

[1] If $\log_e$, units are nats. If $\log_{10}$, ban. And, if $10\log_{10}$, deciban (db) [39].

[2] In case the reader is interested, one can simply compute $\delta_{ij}(x^k)$ in Python with:
np.frompyfunc(lambda x,y: x==y, 2, 1).reduce(np.array(np.meshgrid(F_k, F_k)))

[3] In the case of $S_{ij} = 0$, it is considered that $0 \log(0) = 0$.

The result of this method is shown in Figure 3.2. Notice that the entropy associated with the dataset for all variables, with one missing column, results in high values of bits. This is just a reflection that the dataset itself is extraordinarily disordered with high levels of uncertainty [41]. Moreover, the information contributed by the categorical data is less than the numerical data, this can also be observed in Figure 3.2, along with Table 3.8. When implemented in Python, the result is the following when printing the ordered features from the most important to the less important.

```
>>> print(ordered_features,)

    ['WFICFORM', 'WFFORM', 'RHOSTAR', 'ZEFFNEO', 'DWDIA', 'BETASTAR', 'POHM', 'NEL',
    ↪  'NUSTAR', 'EPS', 'TAV', 'PFLOSS', 'WMHD', 'Q95', 'MEFF', 'PLTH', 'LCOULOMB',
    ↪  'OMEGACYCL', 'KAREA', 'PICRH', 'TIV', 'PRAD', 'TEV', 'PNBI', 'PECRH',
    ↪  'ELMFREQ', 'IP', 'ENBI', 'AMIN', 'ZEFF', 'TORQ', 'NESOL', 'BT', 'BEIMHD',
    ↪  'VOL', 'TOK', 'WALMAT', 'EVAP', 'PREMAG', 'LIMMAT', 'DIVMAT', 'ELMTYPE',
    ↪  'ECHMODE', 'ICSCHEME', 'PELLET', 'CONFIG', 'AUXHEAT']
```

Meaning that the less important in providing information, is the type of auxiliary heating used during the shot, and the most important is the total fast one energy due to ICRH. It is also interesting to note that most of the categorical data do not contribute significantly to providing information. From this, we can then take the first 16 most important variables in terms of entropy, and subject that subset to analysis. In this case, the *entropy variables* will be: WFICFORM, WFFORM, ZEFFNEO, RHOSTAR, DWDIA, BETASTAR, POHM, NEL, WMHD, TAV, NUSTAR, EPS, PFLOSS, Q95, PLTH, and LCOULOMB.

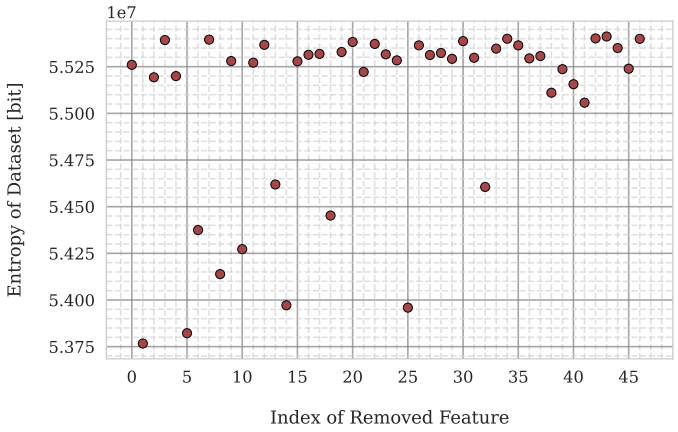

Figure 3.2: Result of applying Algorithm 2 to the 47 variables of interest in `big_DB5`; the same output resulted when applying it to `small_DB5`, despite having 848 fewer observations which may have made it more ordered. Table 3.8 shows the labels of the index of the removed feature.

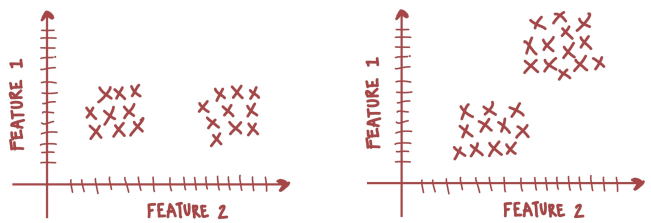

Figure 3.3: Simple drawing showing the comparison between two situations to understand feature dependence for finding ordered groups.

Note that the presence of certain variables can overshadow or enhance the importance of others. For example, a single feature can contain ordered groups by itself, as shown in Figure 3.3 (left), while other features may need to be combined to obtain an ordered group, as shown in Figure 3.3 (right).

While the Entropy method considers dependence on other features, it does not evaluate the best subset of features. Due to the many possible feature subsets that can be assessed from the 47 variables of interest, alternative methods have been implemented to identify relevant subsets that provide insights into the factors affecting the decrease in α_R.

Table 3.8: Numbering of variables in Figure 3.2.

0 - AMIN	12 - ELMTYPE	24 - PECRH	36 - Q95
1 - AUXHEAT	13 - ENBI	25 - PELLET	37 - RHOSTAR
2 - BEIMHD	14 - EPS	26 - PFLOSS	38 - TAV
3 - BETASTAR	15 - EVAP	27 - PICRH	39 - TEV
4 - BT	16 - ICSCHEME	28 - PLTH	40 - TIV
5 - CONFIG	17 - IP	29 - PNBI	41 - TOK
6 - DIVMAT	18 - KAREA	30 - POHM	42 - TORQ
7 - DWDIA	19 - LCOULOMB	31 - PRAD	43 - VOL
8 - ECHMODE	20 - LIMMAT	32 - PREMAG	44 - WALMAT
9 - ELMFREQ	21 - MEFF	33 - Q95	45 - WFFORM
10 - ELMTYPE	22 - NEL	34 - RHOSTAR	46 - WMHD
11 - ENBI	23 - NESOL	35 - TAV	

3.1.3 Research Variables

During the last year of my master, I attended a two-week workshop, at CEA Cadarache, that required me and my team to estimate the toroidal magnetic field B_t [T] and the radius R_{geo} [m] of a tokamak given the target fusion power P_{fus} [MW] and fusion gain Q. To solve this issue, it is possible to create a system of non-linear equations [42]; such as:

$$R_{geo}^3 B_t^4 = \frac{P_{fus} C_\beta^2 q_{95}^2}{C_{fus} C_I^2 \kappa_a \epsilon^4 \beta_N^2}, \tag{3.5}$$

and

$$R_{geo}^{-\gamma_R} \cdot B_t^{-\gamma_B} = C \cdot \Gamma(Q) \cdot \alpha_0 \cdot P_{fus}^{\alpha_P} \cdot M_{eff}^{\alpha_M} \cdot \kappa_a^{\alpha_\kappa} \cdot \epsilon^{\gamma_\epsilon} \cdot n_N^{\alpha_n} \cdot q_{95}^{-\gamma_I} \cdot \beta_N, \tag{3.6}$$

where $\Gamma(Q)$, β_N, n_N, γ_R, γ_B and the constants C, C_I, C_{fus}, C_β are defined through the derivation (see Appendix A) as properties of the tokamak are established. The main takeaway of this research is that one can get an expression where the energy confinement time $\tau_{E,th}$ scales with four engineering variables: line average electron density $\bar{n}_e$, temperature of the plasma $\hat{T}$, outer radius R_{geo}, and toroidal magnetic field B_t; and, $\omega_c \cdot \tau_{E,th}$ with: normalised ion gyroradius ρ_*, normalised ion collision frequency ν_*, and normalised plasma beta β_t [12]. Therefore, I have decided that the *research variables* that might yield

information into the reduction of α_R are: NEL ($\bar{n}_e$), TAV ($\hat{T}$), BT (B_t), RHOSTAR (ρ_*), NUSTAR (ν_*), and BETASTAR (β_t)[4].

3.1.4 Low Multicollinearity Variables

To assess the amount of multicollinearity in the variables of interest, I have implemented the same techniques explained in the previous chapter to `clean_DB5`. The result is shown in Figure 3.4.

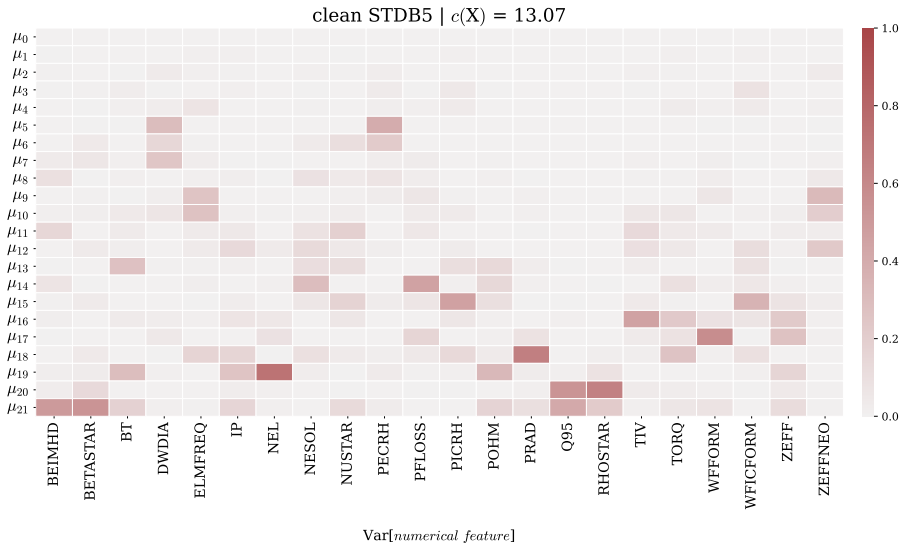

Figure 3.4: Π matrix and condition number for all numerical features of interest.

The VIF of PLTH and PNBI were 4200 and 3600, respectively. To derive a subset of variables with low multicollinearity, I systematically removed one variable at a time and recalculated the condition number along with the VIF of each variable until they stopped being too high. To reach a condition number of 13.07 and a maximum VIF of 14, the following variables were removed: PLTH, AMIN, LCOULOMB, KAREA, EPS, MEFF, TAV, ENBI, PNBI, OMEGACYCL, TEV, VOL, WMHD. Hence, the resultant variables with low multicollinearity are:

```
>>> print(low_multicollinearity_features,)

    ['BEIMHD', 'BETASTAR', 'BT', 'DWDIA', 'ELMFREQ', 'IP', 'NEL', 'NESOL', 'NUSTAR',
    ↪  'PECRH', 'PFLOSS', 'PICRH', 'POHM', 'PRAD', 'Q95', 'RHOSTAR', 'TIV', 'TORQ',
    ↪  'WFFORM', 'WFICFORM', 'ZEFF', 'ZEFFNEO']
```

made of 22 numerical features.

3.2 Classification

As previously mentioned, machine learning algorithms are trained to learn from data in order to make predictions or decisions. If the algorithm predicts numerical data, it is known as regression, such as OLS. On the other hand, if the algorithm predicts classes or categorical data, it is known as classification.

[4] Although $\beta_N \neq \beta_t$, I will focus on β_t since it is easier to interpret in terms of data analysis for tokamak physics and it directly relates to β_N (see Appendix A).

Both regression and classification are forms of supervised machine learning, meaning that the data must include an output variable column **y** for the algorithm to learn from the input columns **X**. The goal of the algorithm is to learn how the input relates to the output to correctly predict **y** for new inputs that it has never seen [20]. Given this information, it is possible to use the results from Chapter 2 to add a new column to `clean_DB5` containing the labels that do not affect or decrease α_R, labelled as 0 and 1 respectively. Observations belonging to either `small_ds` or `big_ds` (depending on the case being studied) will be labelled with 1.

There are numerous algorithms available for classification, some of which may or may not be well-suited for the problem at hand. Certain algorithms, referred to as "data-hungry" require large amounts of data and tend to perform well in fields such as social media, where a vast amount of data is generated daily. However, these algorithms can be challenging to implement in academic research as acquiring more data is typically expensive or unattainable [43], [44]. On the other hand, there are algorithms that perform exceptionally well with a small amount of data because they require the inverse of a certain matrix, which can be computationally expensive [20].

3.2.1 Imbalanced Data

For this project, I have adopted the step-by-step framework proposed by [45], which is designed to handle imbalanced data classification problems. The framework focuses on selecting a classifier that performs well in predicting the minority class, which in our case is the class that decreases α_R. The author suggests first focusing on choosing the correct metric and then using that to test various algorithms.

Various metrics evaluate different aspects of the algorithm. For example, a particular metric can enhance the performance in predicting all classes, the majority class, or the minority class; selecting the wrong metric means selecting an incorrect algorithm for solving a specific problem [45]. Since we need to predict two labels, 0 and 1, we have a binary classification problem; various metrics for this type of problem can be obtained through a tool known as "confusion matrix", see Figure 3.5. Table 3.9 shows some of the most common metrics that can be obtained through a confusion matrix.

To use a confusion matrix, the data must be split into two sections: one for training the algorithm, and the other for validating its performance on unseen observations. Typically, the validation dataset comprises 20% to 50% of the complete dataset, depending on the amount of available data [20]. For this project, I created a validation dataset by stratified random[5] sampling 30% of `clean_DB5`, with its corresponding labels.

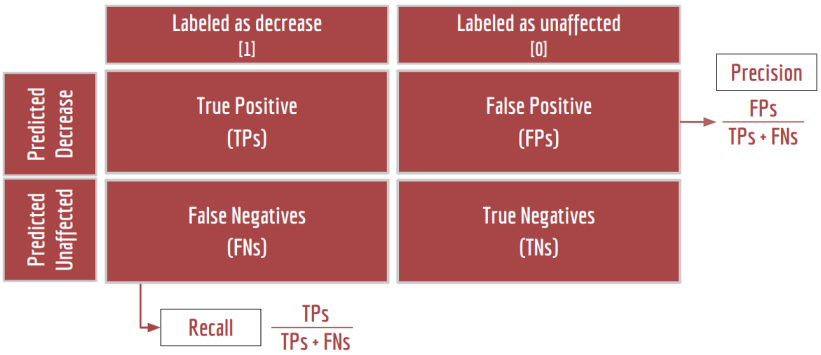

Figure 3.5: Confusion matrix for a binary class and two metrics. This is a tool to obtain different metrics to evaluate an algorithm, here showing for precision and recall. The minority class is referred to as positive. See Table 3.9 for more metrics.

[5]This means that there is the same amount of classes in the training dataset as in the validation dataset.

It is important to mention that all classifiers make use of a threshold, whose values $\in [0,1]$, to make decisions or predictions. The choice on this threshold directly impacts the performance of the algorithm, if it is too high, the classifier will almost never predict the positive class, and vice versa [20].

Table 3.9: Some of the obtainable metrics from a confusion matrix [20], [45]. Remember that n is the total number of observations in a dataset.

Metric	Formula	Description
Misclassification Rate	$\frac{FNs+FPs}{n}$	Fraction of predictions being incorrect
Accuracy	$\frac{TNs+TPs}{n}$	Complement of misclassification rate
Precision	$\frac{TPs}{TPs+FPs}$	Fraction of predicted positives actually being positive
Recall	$\frac{TPs}{TPs+FNs}$	Fraction of actual positives correctly predicted
Fall-out	$\frac{FPs}{FPs+TNs}$	Probability of false alarm
Specificity	$\frac{TPs}{FPs+TNs}$	Compliment of fall-out
False discovery rate	$\frac{FPs}{FPs+TPs}$	Fraction of incorrect positive predictions
False negative rate	$\frac{FNs}{FNs+TPs}$	Fraction of actual positive incorrectly classified
False omission rate	$\frac{FNs}{FNs+FPs}$	Fraction of incorrect negative relative to tall incorrect classifications
Prevalence	$\frac{FNs+TPs}{n}$	Proportion of actual positive instances in the dataset
F_1-score	$\frac{2 \cdot precision \cdot recall}{precision+recall}$	Harmonic mean of precision and recall
F_β-score	$\frac{(1+\beta^2) \cdot precision \cdot recall}{\beta^2(precision+recall)}$	Used to account that recall is β-times as important as precision

When dealing with imbalanced or asymmetric problems, it is recommended to focus on metrics such as F_1-score, precision, and recall [45]. Imbalanced data refers to a dataset with more instances of a particular class than others. An asymmetric problem arises when a false negative is more serious than a false positive, or vice versa. For example, consider the diagnosis of a rare disease. In both situations, training an algorithm with accuracy or misclassification rate would lead to disastrous results [20]. As mentioned in Table 3.9, F_1-score is the harmonic mean of the precision and recall curve, which means it is a metric that represents the performance on the algorithm predicting the minority class in a single metric [20]. Another way to merge these two metrics is through the precision-recall curve (PRC), which is used to optimize the threshold of a classifier [45]. Figure 3.6 shows how to interpret these curves.

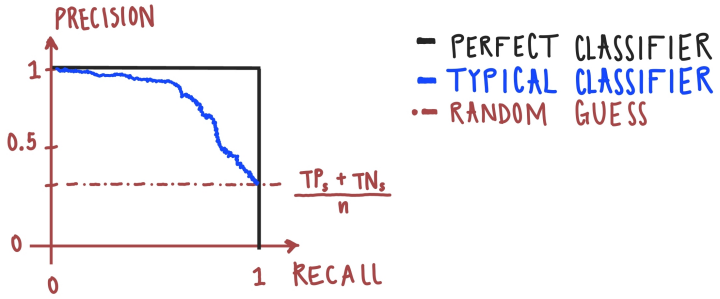

Figure 3.6: Drawing of a precision-recall curve (PRC) for three classifiers, as the threshold of the classifier decreases.The closer the area under the curve is to one, the better the classifier.

After having decided on the metric, now it is time to revise different classifiers and decide which one is worth our attention. For a quick comparison, I have decided to consider

(i) a linear parametric model: logistic regression,
(ii) a non-linear probabilistic model: Gaussian process,
(iii) a non-parametric ensemble: random forest.

These three algorithms share the property of not requiring a large amount of data to perform well, but there are significant differences between each of them. Algorithm (i) differs from (ii) and (iii) by modelling the output as a linear combination of weighted inputs. Both (ii) and (iii) can handle linear and non-linear combinations of the inputs. The main difference between (ii) and (iii) is that (ii) is a probabilistic model that assumes the observations are drawn from a multivariate normal distribution, whereas (iii) makes no strong assumptions[6] about the data and makes decisions based on information gain [20].

One can quickly implement these algorithms using scikit-learn[7], a Python library designed to execute various machine learning algorithms [47]. For a quick comparison, I have implemented these three models using the default settings of scikit-learn[8] and all variables of interest with one-hot encoding applied to categorical data and scaling numerical values. results of their PRC are shown in Figure 3.7 when considering `small_ds` and Figure 3.8 when considering `big_ds`. Some of the metrics for all algorithms are shown in Table **??**.

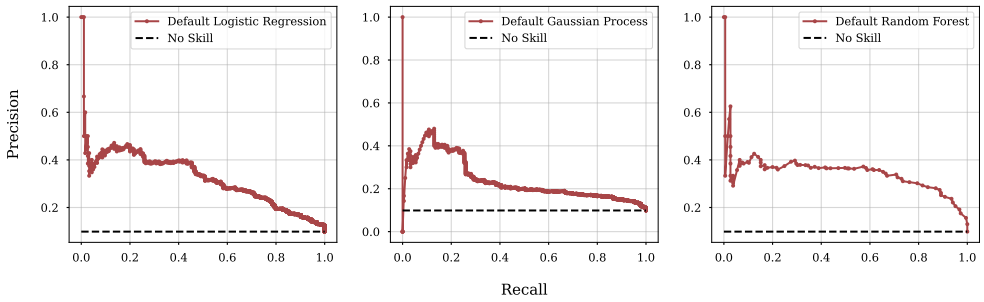

Figure 3.7: PRCs of all variables of interest and default settings in scikit-learn for logistic regression, Gaussian process and random forest; utilizing `small_ds` to label observations. A model with no skill is shown with dashed lines, which means it is a model that classifies based on random guesses. In order, the F_1-score obtained per model were: 0.16, 0.31, and 0.26.

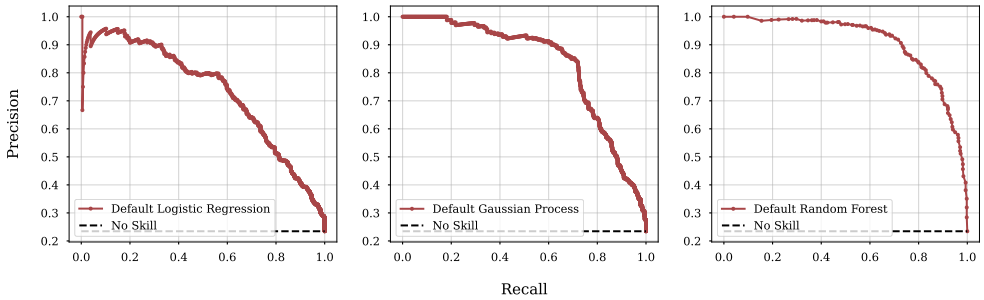

Figure 3.8: PRCs of all variables of interest and default settings in scikit-learn for logistic regression, Gaussian process and random forest; utilizing `big_ds` to label observations. A model with no skill is shown with dashed lines, which means it is a model that classifies based on random guesses. In order, the F_1-score obtained per model were: 0.66, 0.77, and 0.81.

[6] In general, all machine learning algorithms assume that the input data is independent and identically distributed, this is known as the i.i.d. assumption.

[7] This library uses CPU power, a library that implements GPU with Nvidia graphic cards is cuML [46].

[8] Version 1.2.2

It is possible to see that the worst performance is on the linear classifier, regardless of the type of dataset being used. On the other hand, Gaussian process and random forest perform appropriately for `big_ds`, only; this means that `small_ds` is a case of severely imbalanced data and might require an algorithm that specializes in this type of situation, like the one-class classifier [45]. For now, I will focus on the analysis of `big_ds` using Gaussian process (GP) and random forest (RF).

3.2.2 Gaussian Process and Random Forest

GP and RF are two machine learning algorithms that can be used both for regression and classification. The main difference between these two methods is that one uses a stochastic approach, while the other uses information gain of various models for making decisions [20]. I will explain in more detail the difference between them.

Gaussian Process
A collection of finite random variables whose joint distribution is a Gaussian distribution can be considered a Gaussian process. In other words, it is a method for modelling data using multivariate Gaussian distributions [48]. Figure 3.9 shows the case of a multivariate Gaussian distribution for two features contained in an arbitrary vector $\vec{r}$, with mean $\vec{\mu}$ and covariance matrix $\vec{\Sigma}$.

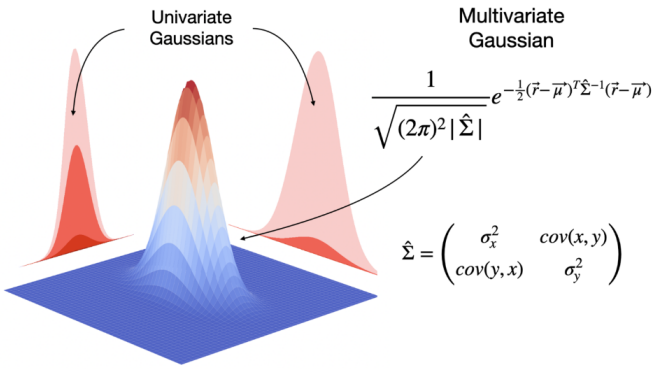

Figure 3.9: . Illustration of a bivariate Gaussian distribution. Image from: [49].

GP is a collection of random variables representing the value of a real process $f(\mathbf{x})$, at location $\mathbf{x}$, which is fully defined by its mean function $m(\mathbf{x})$ and covariance function $k(\mathbf{x}, \mathbf{x}')$, expressed as [20], [50]

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')), \tag{3.7}$$

where

$$\begin{aligned} m(\mathbf{x}) &= \mathbb{E}[f(\mathbf{x})] \\ k(\mathbf{x}, \mathbf{x}') &= \mathbb{E}[(f(\mathbf{x}) - m(\mathbf{x})) \cdot (f(\mathbf{x}') - m(\mathbf{x}'))] = \mathrm{Cov}[f(\mathbf{x}), f(\mathbf{x}')], \end{aligned} \tag{3.8}$$

with $\mathbf{x}'$ denoting a different observation.It is common to set the mean $m(\mathbf{x})$ to zero, as it is possible to add an extra term in the covariance function to represent its uncertainty [51], and to treat the covariance function as kernels [20]. The heart of GPs lies in the kernel and not in $f(x)$ per se. The latter is more of a tool that allows deriving the path for inference. In fact, the problem of learning GPs is the problem of

learning the optimal properties of the kernel. This is because the kernel states the relationship of given data to new data [50], [51].

Formally speaking, a kernel is a positive-definite function of two inputs $\mathbf{x}$, $\mathbf{x}'$ that are Euclidean vectors; but, they could also represent categorical inputs, graphs, images, or text. Another way to interpret a kernel is by treating them as mathematical objects specifying the similarity of a function evaluated on different objects, as shown in Eq. (3.8) [51]. By adding or multiplying kernels with different characteristics, it is possible to create a new kernel that encompasses all of these properties, examples of this are shown in Figure 3.10.

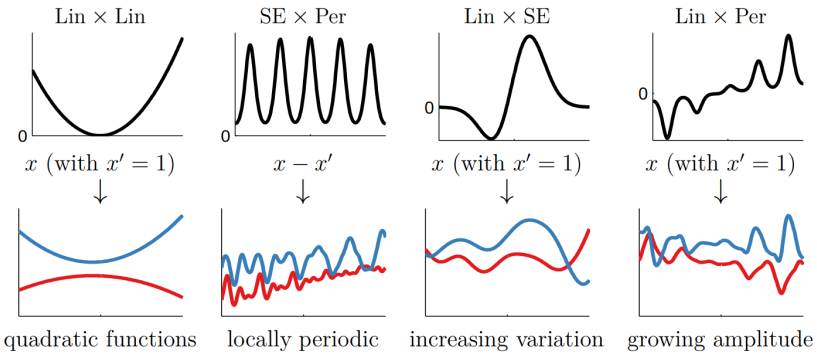

Figure 3.10: Some of the most common kernels, linear (Lin), squared exponential (SE), and periodic (Per); combined to form a new kernel. The top row of plots represents $k(\mathbf{x}, \mathbf{x}')$ and the bottom plots are two samples of a $\mathcal{GP}$ with the specified kernels above; this type of sampling is known as sampling the *prior* [48].
Image from: Fig. 2.2 in [51].

Once the mean and the covariance function have been specified, one proceeds to compute the predictive distribution $p(\mathbf{y}'|\mathbf{x}', \mathbf{X}, \mathbf{y})$ for the unknown target variable $\mathbf{y}'$, given the new input $\mathbf{x}'$ and the data used to train the algorithm $\mathbf{X}$, with their corresponding labels $\mathbf{y}$; namely,

$$p(\mathbf{y}' \mid \mathbf{x}', \mathbf{X}, \mathbf{y}) = \mathcal{N}(\mathbf{y}'|\mu_*, \Sigma_*), \tag{3.9}$$

which represents the posterior after observing the data and turns to be a GP too. Here,

$$\begin{aligned} \mu_* &= \mathbf{k}_* K^{-1} \mathbf{y} \\ \Sigma_* &= k(\mathbf{x}', \mathbf{x}') - \mathbf{k}_* K^{-1} \mathbf{k}_*^T, \end{aligned} \tag{3.10}$$

where $K_{ij} = k(\mathbf{x}_i', \mathbf{x}_j')$ is the covariance matrix, and $\mathbf{k}_*$ is the row-vector $\mathbf{k}_* = (k(\mathbf{x}_1, \mathbf{x}') \ \ldots \ k(\mathbf{x}_n, \mathbf{x}'))$. Then, one finds the optimal hyperparameters of the kernel by maximizing the marginal likelihood $p(\mathbf{y}|\mathbf{X})$, also Gaussian, through an optimization algorithm, like gradient descent. Broadly speaking, this is the general procedure to implement GPs for regression; an algorithm with a complexity $\mathcal{O}(n^3)$ [48]. Nevertheless, when working GPs for classification, not everything is analytically tractable since the labels are discrete and cannot have a Gaussian likelihood. In this situation, one "squashes" the output of a regression model into a class probability through what is known as a response function; for instance,

$$p(y = 1 \mid \mathbf{x}) = \sigma(y f(\mathbf{x})), \text{ and, } \ p(y = 0|\mathbf{x}) = 1 - \sigma(y f(\mathbf{x})), \tag{3.11}$$

where $f \sim \mathcal{GP}(0, k(\mathbf{x}, \mathbf{x}'))$ and $\sigma(z)$ can be the sigmoid function $\sigma(z) = [1 + \exp(-z)]^{-1}$ or the cumulative density function of a standard normal distribution $\sigma(z) = \Phi(z) = \int_{-\infty}^{z} \mathcal{N}(x|0, 1)dx$, known as the probit

regression [50], [52]; this depends on the problem at hand. Here, we take the sigmoid function. After defining the model through the link function, one can obtain the unnormalised posterior as Eq. (15.33) in [52]:

$$\ell(f) = \log p(y \mid f) + \log p(f \mid X) = \log p(y \mid f) - \frac{1}{2} f^T K^{-1} f - \frac{1}{2} \log |K| - \frac{n}{2} \log 2\pi, \tag{3.12}$$

which can be normalised and analytically approximated by Laplace approximation [50], [52], [53]. One then gets the posterior predictive distribution as Eq. (15.48) in [52],

$$p(f_* \mid \mathbf{x}_*, \mathbf{X}, \mathbf{y}) = \mathcal{N}\left(\mathbb{E}[f_*], \mathrm{Var}[f_*]\right),$$

where $\mathbf{x}_*$ is the best point used for f; and $f_* = f(\mathbf{x}_*)$. Then, the hyperparameters of the kernel are estimated with a gradient-based optimization algorithm, such that the marginal likelihood is maximized [53]; namely, Eq. (15.51) in [52],

$$\log p(y \mid \mathbf{X}) \approx \log p(y \mid \hat{f}) - \frac{1}{2} \hat{f}^T K^{-1} \hat{f} - \frac{1}{2} \log |K| - \frac{1}{2} \log \left| K^{-1} - \nabla\nabla \log p(y \mid f) \right|, \tag{3.13}$$

where $\hat{f}$ is the maximum a posteriori estimation algorithm[9] (MAP) applied to f. These steps are applied to obtain the GP classifier method, shown in Algorithm 3; where $k_{**} = k(\mathbf{x}_*, \mathbf{x}_*)$, $\mathbf{I}_n$ is the $n \times n$ identity matrix, and Cholesky refers to the Cholesky decomposition, used to efficiently inverse matrices [50], [52].

Algorithm 3 Gaussian Process Classifier, based in Algorithm 15.2 in [52]. The algorithm I have implemented is *GaussianProcessClassifier()* of scikit-learn [55], which also follows this structure.

Require: MAP using iteratively reweighted least squares.
  $f \leftarrow 0$
  repeat
    $W \leftarrow -\nabla\nabla \log p(y \mid f)$
    $B \leftarrow \mathbf{I}_n + W^{1/2} K W^{1/2}$
    $L \leftarrow \mathrm{Cholesky}(B)$
    $b \leftarrow W f + \nabla \log p(y \mid f)$
    $a \leftarrow b - W^{1/2} L^T \backslash (L \backslash (W^{1/2} K b))$
    $f \leftarrow K a$
  until convergence
  $\log p(y \mid X) \leftarrow \log p(y \mid f) - \frac{1}{2} a^T - \sum_i \log L_{ii}$
  % Perform prediction
  $\mathbb{E}[f_*] \leftarrow \mathbf{k}_*^T \nabla log p(y \mid f)$
  $v \leftarrow L \backslash \left(W^{1/2} \mathbf{k}_* \right)$
  $\mathrm{Var}[f_*] \leftarrow k_{**} - v^T v$
  $p(y_* = 1) \leftarrow \int \sigma(z) \mathcal{N}(z \mid \mathbb{E}[f_*], \mathrm{Var}[f_*]) dz$ =0

The algorithm takes $\mathcal{O}(n^3)$ for fitting, and $\mathcal{O}(n^2 n')$ for prediction, where n' is the total number of new observations to be classified [52]. Extensive details on the derivation of this algorithm are found on pp. 525-528 [52], and pp. 33-48 [50].

[9] A probabilistic framework to estimate probability densities. A good description of MAP is found in [54].

Random Forest

Ensemble learning is a method that involves constructing a new model by combining multiple basic models, each trained in slightly different ways. By doing so, the ensemble leverages the different outputs provided by the models to define the input-output relationship. [20]. RF is an instance of ensemble learning by using decision trees with different randomized samples.

A decision tree is a rule-based model that structures a graph referred to as a tree. Various disjoint regions are created to divide the input space, with each region having a fixed value to predict the output [20]. A simple tree for a classification problem is shown in Figure 3.11 using two numerical features.

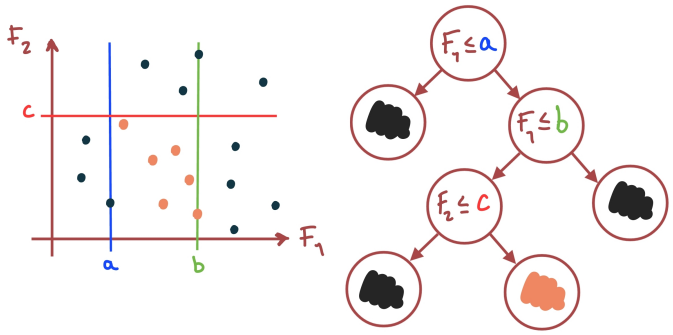

Figure 3.11: Simple drawing illustrating a plot for two numerical features and their corresponding binary decision tree. Here, a, b, and c represent real numbers creating decision boundaries, shown as lines in the plot. The top node of a tree is called the *root node*, and its subsequent branches are referred to as child-internal nodes. Nodes with a condition are known as *parent nodes*, while the nodes that follow them are their *child nodes*. A node that does not have any further branches is called a *leaf node*, which leads to a classification outcome [20]. Drawing is inspired by [56].

It is possible that observations may overlap with the decision boundary, as shown in Figure 3.11 for b; in such situations, it can be assigned to the majority class in its vicinity, which corresponds to regions with a higher risk of misclassification. Furthermore, it is worth mentioning that if the number of leaf nodes is equal to the number of observations in the data set; then, it is said that the tree is fully grown, meaning that the model mimics the training data and it has lost its capability of generalisation, this is known as *overfitting* the model [20]. To prevent this, one can set a stopping criterion.

The challenge in decision trees lies in determining the constants that establish optimal decision boundaries for a given dataset, considering the infinite number of available options. The recommended approach for deriving these constants is by employing *greedy algorithms* [20]. Specifically, the splitting rules are generated sequentially, one at a time, starting from the root node. This process is known as the recursive binary splitting algorithm [20], which has been implemented in various versions, including CART (classification and regression trees) [57], ID3 (Iterative Dichotomiser 3) [58], and C4.5 [59].

To create a binary decision tree, for a binary classification problem, one begins by designing the root node; this implies iterating over all features and assessing which ones give the best split. There are three different criteria to measure the quality of a split through their amount of *impurity*; hence, the lower the impurity, the better [20], [60]. The different metrics are shown in Figure 3.12. Mathematically, each of the criteria for computing the impurity per node $i(N)$ is defined as [20], [60]

- misclassification error:

$$i(N) = 1 - max(\pi_1, 1 - \pi_1), \tag{3.14}$$

- Gini index:

$$i(N) = 2\pi_1 (1 - \pi_1), \tag{3.15}$$

- and, entropy:

$$i(N) = -\pi_1 \log(\pi_1) - (1 - \pi_1)\log(1 - \pi_1). \tag{3.16}$$

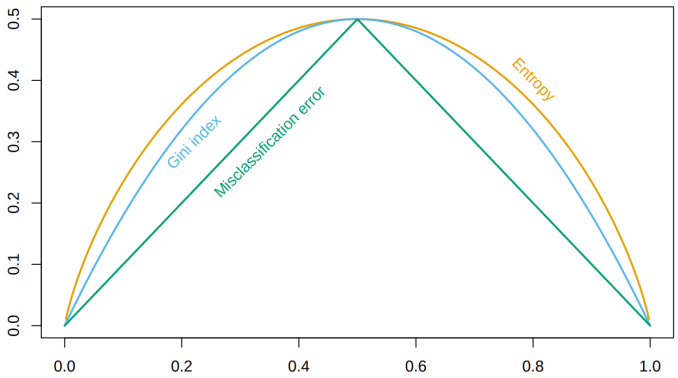

Figure 3.12: Measure of node impurity $i(N)$, vertical axis, versus the proportion of the first class π_1 in a given region denoted by a split, horizontal axis. Image from: Fig. 9.3 in [61].

Once the optimal root node has been determined, the subsequent splits are constructed based on the selected criteria and the preceding node, until the stopping criterion is met. The general idea of how to implement a binary decision tree classifier is shown in Algorithm 4.

Algorithm 4 Basic principles in a Decision Tree Classifier

Require: Stopping condition $stopping\_cond$
Require: Training dataset $\mathcal{D}$ and New observations $\mathcal{D}'$
  Model Construction:
  Split data considering all observations in $\mathcal{D}$
  Compute quality of split according to the selected impurity metric
  while $stopping\_cond$ is not fulfilled and further splits are possible **do**
    Go through all possible splits considering previous ones
    Pick the split that minimizes impurity of the node
    Check for $stopping\_cond$
  Store tree conditions in T
  Prediction:
    for each row in $\mathcal{D}'$ **do**
      Traverse tree T based on the conditions
      Get prediction for the row

One drawback of decision trees is their high sensitivity to the training data, even when the tree is not fully grown. To address this issue, an improvement can be achieved by creating multiple trees trained on the same dataset but with the introduction of randomness. These ensembles of randomised trees are known as random forests.

To create a random forest, a common practice is to start with a specified number of trees, typically ranging from 50 to 1000. The number of trees corresponds to the number of bootstrap samples, with replacement, that will be drawn from the dataset. Each subset generated through bootstrapping is then used to train an individual tree. Subsequently, a new observation can be passed through all the trees, and the final classification is determined by the majority vote across the ensemble. This process is referred to as aggregation. The overall procedure, combining bootstrapping and aggregation, is known as bagging. It is worth noting that other models can be constructed following the same reasoning by utilising different base models [20], [60].

For an ensemble to outperform a single model, two key requirements must be met: (i) each classifier within the ensemble should be uncorrelated, and (ii) the individual classifiers should have an error probability below 0.5 [60].

An additional source of randomness can be introduced to enhance the decorrelation among trees. Instead of considering different tree sizes while using all features, one can fully grow each tree but utilise different randomized subsets of features to develop them. It is worth noting that the latter requires less computational power than the former [60], but the optimal implementation depends on the dataset at hand; for instance, if there are several features with similar information, it is advisable to implement the latter technique.

Random Forest Algorithm

Algorithm Tuning
In order for an algorithm to achieve optimal performance, it is crucial to tune its hyperparameters according to the specific dataset. The advantageous aspect of Gaussian Process is that it automatically tunes its hyperparameters while calculating the maximum likelihood [48]. Conversely, the hyperparameters of Random Forests, such as the number of trees to be created or the number of features to be considered per split, are tuned using a grid search methodology [20]. A grid search involves exploring a range of different values for the hyperparameters and evaluating the algorithm's performance for each combination. The hyperparameters that yield the best performance are then selected as the optimal choices [20], [48].

3.2.3 Feature Importance

There is a way to assess the most significant features for the algorithm to learn. The idea is based on randomly shuffling the values of a single feature and subjecting the modified dataset to the model. The performance of the model is then evaluated using a specific metric, such as precision or recall. If the model's performance significantly worsens compared to the original dataset, it is considered that the respective feature played an important role in learning [62]. It is important to note that if there are two correlated features and one of them is randomly shuffled, the model can still access their properties through the other correlated variable, thereby masking the importance of the shuffled feature [63].

Feature Importance Algorithm

Entropy with Low Multicollinearity Variables
Before presenting the results of the feature importance for each algorithm and subset of features, I would like to mention that I have examined the multicollinearity in the entropy variables. I found that it yielded a value of $c(\mathbf{X}) = 45.07$. Consequently, I strategically eliminated the features with high collinearity and low importance when analysed with an RF classifier. The resulting set of features demonstrates low multicollinearity and high importance. These features include WFFORM, RHOSTAR, DWDIA, BETASTAR, POHM, NEL, NUSTAR, and PLTH. Collectively, they exhibit a multicollinearity value of $c(\mathbf{X}) = 7.36$. Henceforth, this subset of features will be referred to as the *entropy with low multicollinearity*. It is worth mentioning that the research variables have $c(\mathbf{X}) = 6.97$.

3.2.4 Results

Up to this point, there are two algorithms, Gaussian process and random forest, that can be used to assess three different subsets of variables: research, entropy with low multicollinearity, and low multicollinearity. Consequently, we have a total of six different models to examine the variables that may provide

insights into the reduction of α_R in STDB5. Additionally, I have added the categorical features to the entropy with low multicollinearity and research features for a more complete analysis.

Figure 3.13 depicts the PRCs for the three cases utilizing Gaussian Process, while Figure 3.14 illustrates the PRCs for tuned Random Forests.

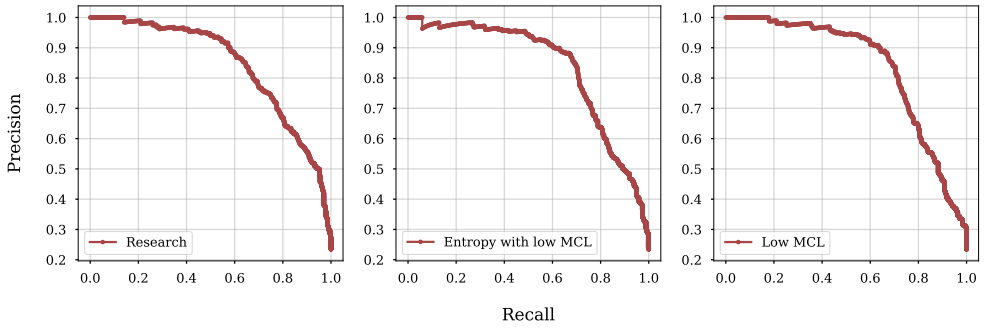

Figure 3.13: PRCs for GP classifiers with three different subsets of features being used. In order, the F_1-score obtained per model were: 0.74, 0.76, and 0.77.

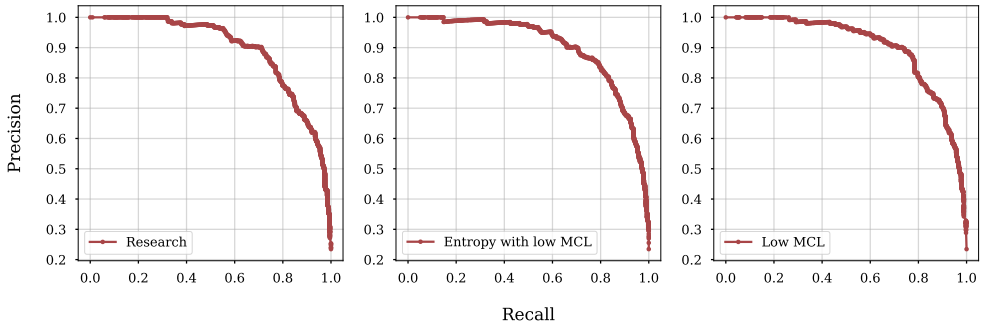

Figure 3.14: PRCs for RF classifiers with three different subsets of features being used. In order, the F_1-score obtained per model were: 0.79, 0.81, and 0.82.

It is evident that RF consistently outperforms GP in all scenarios. Nevertheless, all algorithms effectively learned from the various datasets to make predictions that reduce α_R, exhibiting acceptable performance. The Random Forest model utilizing the subset of variables with low multicollinearity proved to be the most effective. Figure 3.15 displays the resulting first five feature importance for GP, while Figure 3.16 illustrates the feature importance for RF, across the three subsets of variables.

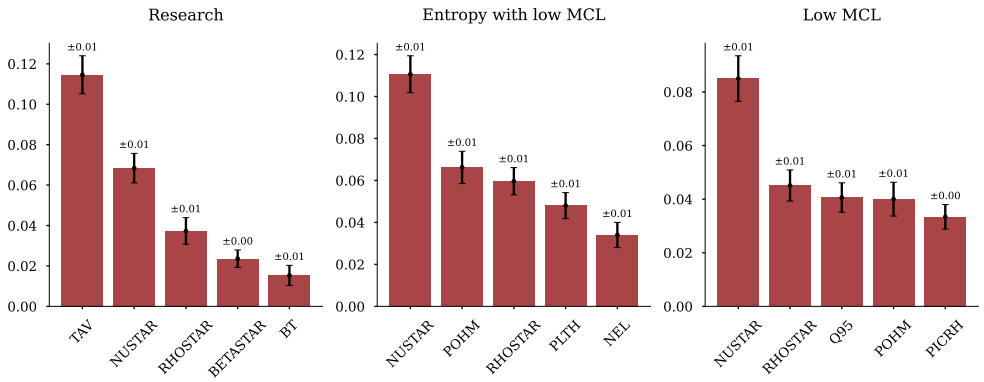

Figure 3.15: Top five most important features in Gaussian Process when utilising three different subsets. The numbers above the bars represent the standard deviation of 200 rounds of shuffling. These standard deviations serve as indicators of the variability in feature importance scores.

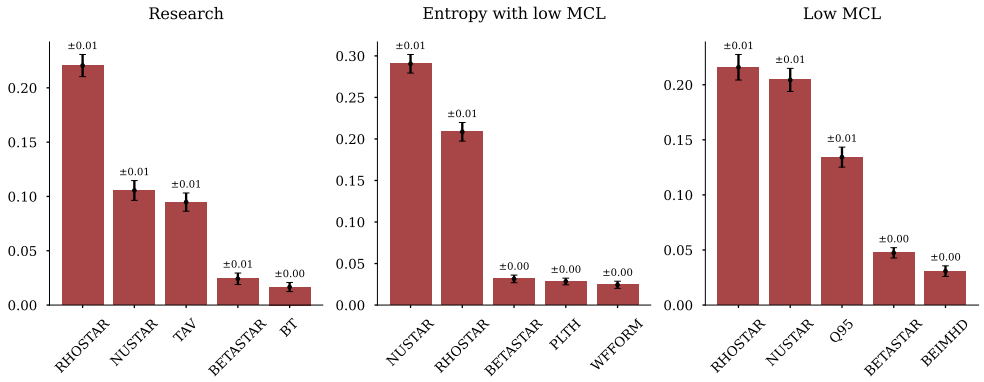

Figure 3.16: Top five most important features in Random Forests when utilising three different subsets. The numbers above the bars represent the standard deviation of 200 rounds of shuffling. These standard deviations serve as indicators of the variability in feature importance scores.

According to all algorithms, the normalised collision frequency and the normalised ion gyroradius emerge as crucial factors in predicting observations that lead to a decrease in α_R. The normalised plasma beta ranks as the third most significant feature. The subsequent important features include the average temperature, safety factor, toroidal magnetic field, and others, which may vary. It is noteworthy that none of the categorical features played a significant role in the learning process of the algorithms. Figure 3.17 show the complete feature performance for the low multicollinearity variables.

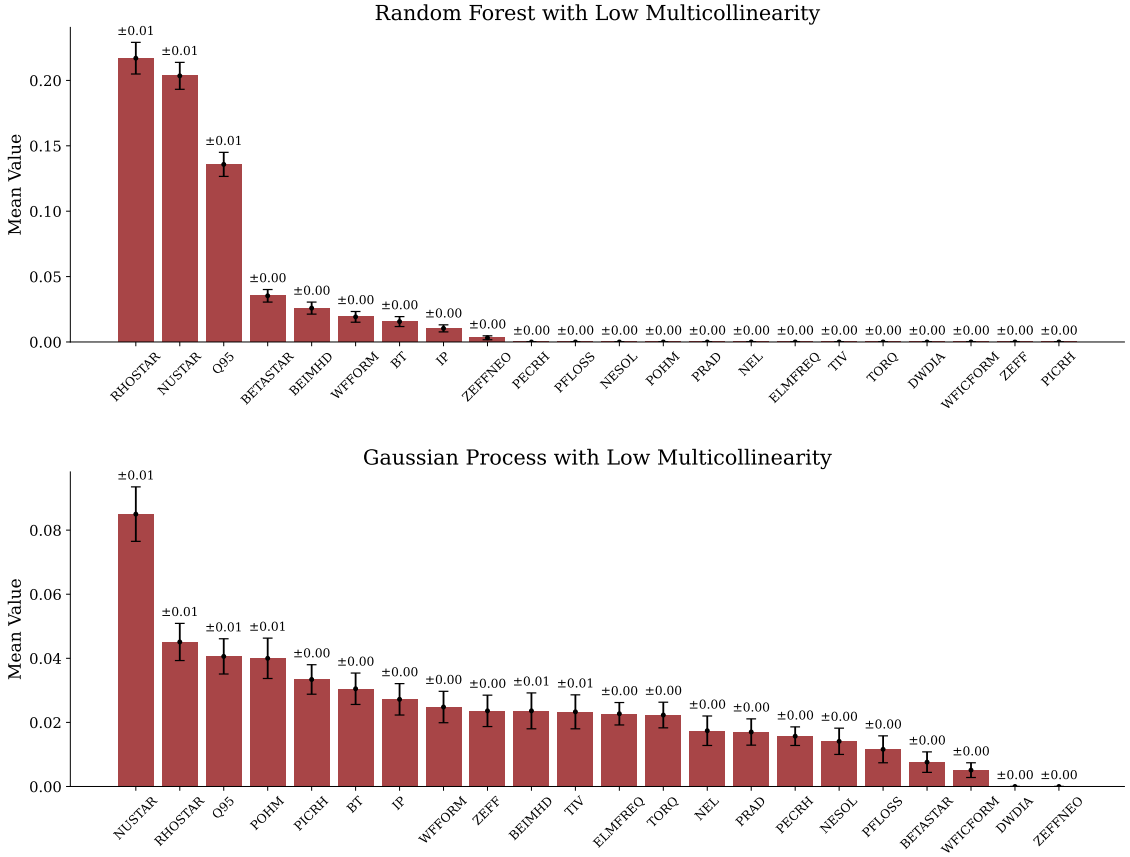

Figure 3.17: Complete feature importance for the best subset of features, for GP (bottom) and RF (top). Both of them show their standard deviation from shuffling the dataset 200 times.

It is interesting to note that GP makes use of most of the features to make predictions, while RF takes less

than half of the subset to outperform GP. In the subsequent chapter, these variables will be thoroughly examined and interpreted to identify commonalities and establish connections with tokamak physics.

3.3 Other Applications in Fusion

Before delving into the tokamak physics interpretation of the most important features, it is worth noting the various novel applications of machine learning in fusion and nuclear technology.

C. Rea Lecture Thursday

IV

TOKAMAK PHYSICS

4.1 Variables of Importance and Interpretation

V

CONCLUSIONS

A

DERIVATION OF RESEARCH VARIABLES

In this chapter, I will explain step by step how to derive Eq. (3.5) and Eq. (3.6) to understand why the *research variables* are of particular interest. These mathematical derivations are heavily based on the ideas and research findings presented in [12], which serves as the primary reference for this section. However, it should be noted that the presented content is not intended to replicate or mimic those in [12]. Rather, this derivation aims to provide a more accessible and reader-friendly approach, particularly for those without a strong background in tokamak physics.

Designing a Tokamak

When having the target fusion power P_{fus} and fusion gain Q, for instance, $P_{fus} = 500$ MW and $Q = 10$, it is possible to figure out the optimal major radius R_{geo} and toroidal magnetic field B_t for best stability. The first thing one should think about is the possible risks that might terminate the plasma confinement; one of them is the Greenwald Limit n_{GW}; which is an sharp operational limit to the electron density that ends in disruption when surpassed [64]. However, recent work done by the Swiss Plasma Center at EPFL has demonstrated that ITER could use twice the amount of Hydrogen than expected without the risk of disruption [65]. The limit is defined as [64]:

$$n_{GW} = \frac{10 \cdot I_p}{\pi a^2}. \tag{A.1}$$

Now, if the length of the poloidal cross-section is defined as $L_\theta = 2\pi a \cdot \sqrt{(1+\kappa_a^2)/2}$ it is possible to approximate the plasma current as [12]:

$$I_p = \frac{1}{\mu_0} L_\theta B_\theta, \tag{A.2}$$

where B_θ is the poloidal magnetic field and can be obtained from the safety factor q_{95}. One can use these definitions, and the fact that $a = \epsilon R_{geo}$, to have an expression for the plasma current in terms of the toroidal magnetic field; namely,

$$I_p = C_I \cdot \frac{\epsilon^2}{q_{95}} \cdot B_t R_{geo}; \text{ with, } C_I = \frac{2\pi}{\mu_0}\sqrt{(1+\kappa_a^2)/2}. \tag{A.3}$$

With this, it is possible to get an expression for the Greenwald Litim in terms of R_{geo} and B_t, instead of I_p and a, such that

$$n_{GW} = \frac{10}{\pi} \cdot \frac{C_i \epsilon^2 B_t R_{geo}/q_{95}}{\epsilon^2 R_{geo}^2} = \frac{10}{\pi} C_I \cdot \frac{B_t}{q_{95} R_{geo}}. \tag{A.4}$$

From this, one gets the normalized density $n_N = \bar{n}_e / n_{GW}$.

Now, let's discuss MHD instabilities. This is a complex topic, but it is sufficient to understand that these instabilities can occur due to perturbations that cause the displacement of the plasma, such as the *Rayleigh–Taylor interchange instability* [66]. If the perturbation propagates perpendicular and slightly parallel to the magnetic field, the plasma will exhibit *ballooning instabilities*, which are pressure-driven instabilities that can result in the loss of plasma confinement [1], [66]. One can reduce the chance of observing MHD instabilities when the tokamak complies with the β-limit, defined as [12]:

$$\beta_{\%} = 100 \cdot \beta_t \leq \beta_m = g \cdot \frac{I_p}{a B_t}, \tag{A.5}$$

where g is a proportionality constant, often considered between 2-4, due to the Troyon limit [67]. From this, the normalized plasma-beta β_N is defined,

$$\beta_{\%} = \beta_N \cdot \frac{I_p}{a B_t} \tag{A.6}$$

such that the stability limit can be read as $\beta_N < g$ [12]. Let us now define the type of species that will be in the tokamak. Figure A.1 shows the highest fusion reaction rate $< \sigma \nu >$ versus the plasma temperature for different elements.

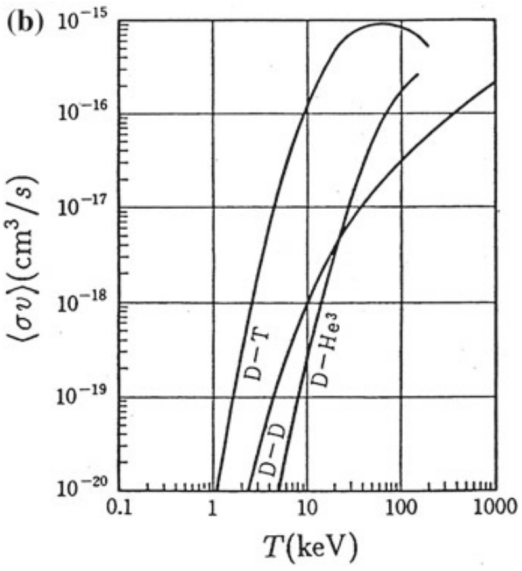

Figure A.1: Feasibility of obtaining fusion power depending on the species in the plasma at a given temperature. The lines shown are for Deuterium-Tritium (D-T), (Deuterium-Deuterium) D-D, and Deuterium-Helium-3 (D-He^3). Image from: Fig. 1.2 (b) in [66].

Since D-T is more likely to undergo fusion at lower temperatures, compared to other fusion fuels, it is a popular option for nuclear fusion research. Nevertheless, Tritium is an expensive and difficult element to

work with [1], which is why some companies prefer to research D-He^3 [68]. ITER will be working with D-T [8]; hence, these are the species that we are considering in this study. Its fusion reaction reads

$$\mathrm{D} + \mathrm{T} \longrightarrow {}^4\mathrm{He}\,(3.56\,\mathrm{MeV}) + \mathrm{n}\,(14.03\,\mathrm{MeV}), \tag{A.7}$$

with a total energy release of $E_{DT} = 17.59$ MeV, per fusion reaction. The ratio of the total energy transferred by the alpha particles $\lambda = 17.50/3.56 \approx 4.94$ [12]. Knowing the density of the deuterium and tritium (n_D and n_T, respectively), one can approximate the P_{fus} through Figure A.1 and

$$P_{fus} = n_D n_T < \sigma\nu >_{DT} E_{DT} V_t, \tag{A.8}$$

where $V_t = 2\pi^2 \kappa_a R_{geo} a^2$ is the volume of the tore [12]. By assuming that $n_D = n_T = \bar{n}_e/2$, replacing the expression of V_t, and approximating $< \sigma\nu >_{DT} \approx 1.1810 \times 10^{-24} \hat{T}\,\mathrm{m^3 s^{-1}}$ through a polynomial fit for temperatures in the 10.3-18.5 keV range [12], one can get that

$$P_{fus} = C_{fus} \kappa \epsilon^2 R_{geo}^3 \left(\bar{n}_e \hat{T}\right)^2, \tag{A.9}$$

with $C_{fus} = \pi^2 \cdot 1.1810 \times 10^{-24} \cdot E_{DT}/2$. To get the expression of P_{fus} in terms of R_{geo} and B_t instead of the plasma pressure $\bar{n}_e \hat{T}$, one can use the definition of $\beta_{\%}$ and β_N. First, we express $\beta_{\%}$ in a convenient form, such as:

$$\beta_{\%} = C_\beta \frac{\bar{n}_e \hat{T}}{B_t^2}; \text{ with } C_\beta = 2 \cdot 100 \cdot 10^3 \cdot q. \tag{A.10}$$

Therefore, $\bar{n}_e \hat{T} = C_\beta^{-1} C_I \beta_N B^2$. Now, we use this to re-express Eq. (A.9) as

$$P_{fus} = \frac{C_{fus} C_I^2}{C_\beta^2} \cdot \frac{\kappa_a \epsilon^4}{q_{95}^2} \beta_N^2 R_{geo}^3 B_t^4. \tag{A.11}$$

We have derived Eq. (3.5). To derive Eq. (3.6), it is necessary to think about the power balance in equilibrium at steady state. This means that the total plasma heating sources P_{hss} is equal to the power loss P_{loss}; namely,

$$P_{hss} = P_\Omega + P_\alpha + P_{aux} = P_{loss} = P_{rad} + P_{l,th}. \tag{A.12}$$

Here, P_Ω is the ohmic heating (often neglected), P_α is the energy carried by the alpha particles that are fully contributing to the D-T fuel, P_{aux} is the auxiliary heating power, P_{rad} is the radiated power, and $P_{l,th}$ is the power lost due to transport through the LCFS [12]. In order to have an explicit expression for the radiative losses, one needs to consider specific characteristics of the tokamak, like the plasma facing components [69], [70]. It is for this reason, that is preferable to parametrize the expression of the radiative losses with a coefficient, $0 \leq \gamma_{rad} \leq 1$, as $P_{rad} = (1 - \gamma_{rad}) P_{hss}$. From this, one gets from the power balance that

$$P_{hss} = (1 - \gamma_{rad})P_{hss} + P_{l,th}, \tag{A.13}$$

meaning,

$$P_{l,th} = \gamma_{rad}\,(P_\Omega + P_\alpha + P_{aux}). \tag{A.14}$$

If one neglects the ohmic heating, and makes use of $P_{fus} = \lambda P_\alpha$ and $Q = P_{fus}/P_{aux}$, then, one obtains the power loss due to transport in the form of

$$P_{l,th} = \gamma_{rad} P_{fus}\left(\frac{1}{Q} + \frac{1}{\lambda}\right). \tag{A.15}$$

It is now time that we make use of the energy confinement time, which is defined as [66]

$$\tau_{E,th} = \frac{1}{P_{l,th}} \cdot 3 V_t \hat{T} \bar{n}_e, \tag{A.16}$$

where it has been assumed that the electron temperature is equal to the ion temperature [12]. By making use of the explicit expression of the torus volume, Eq. (A.9), and Eq. (A.15) one gets

$$\bar{n}_e \hat{T} \tau_{E,th} = 6\pi^2 \left[\gamma_{rad} C_{fus}\left(\frac{1}{\lambda} + \frac{1}{Q}\right)\right]^{-1}. \tag{A.17}$$

We now take the scaling law of the energy confinement time, shown in Eq. (1.1),

$$\tau_{E,th} = \alpha_0 \cdot I_p^{\alpha_I} \cdot B_t^{\alpha_B} \cdot \bar{n}_e^{\alpha_n} \cdot P_{l,th}^{\alpha_P} \cdot R_{geo}^{\alpha_R} \cdot \kappa_a^{\alpha_\kappa} \cdot \epsilon^{\alpha_\epsilon} \cdot M_{eff}^{\alpha_M},$$

to multiply it with $\bar{n}_e \hat{T}$, replace I_p with Eq. (A.3), and $P_{l,th}$ with Eq. (A.15); namely,

$$\bar{n}_e \hat{T} \tau_{E,th} = \alpha_0 \cdot \hat{T} \cdot \bar{n}_e^{\alpha_n + 1} \left(\frac{C_I}{q_{95}} \epsilon^2 B_t R_{geo}\right)^{\alpha_I} \cdot B_t^{\alpha_B} \cdot R_{geo}^{\alpha_R} \cdot \kappa_a^{\alpha_\kappa} \cdot \epsilon^{\alpha_\epsilon} \cdot M_{eff}^{\alpha_M} \cdot \left[\gamma_{rad} P_{fus}\left(\frac{1}{\lambda} + \frac{1}{Q}\right)\right]^{\alpha_P} \tag{A.18}$$

which then can be simplified by replacing $\bar{n}_e$ with n_N, and $\bar{n}_e \hat{T}$ with β_N to then equate with Eq. (A.17); such as:

$$[C \cdot \Gamma(Q)]^{-1} = \alpha_0 \cdot R_{geo}^{\gamma_R} \cdot B_t^{\gamma_B} \cdot P_{fus}^{\alpha_P} \cdot M_{eff}^{\alpha_M} \cdot \kappa_a^{\alpha_\kappa} \cdot \epsilon^{\gamma_\epsilon} \cdot n_N^{\alpha_n} \cdot q_{95}^{-\gamma_I} \cdot \beta_N, \tag{A.19}$$

Which is Eq. (3.6). Here, the exponents γ_x are:

- $\gamma_I = 1 + \alpha_n + \alpha_I$,

- $\gamma_\epsilon = 1 + \alpha_\epsilon + 2\alpha_I$,
- $\gamma_R = \alpha_R + \alpha_I - \alpha_n$,
- $\gamma_B = \alpha_B + \alpha_n + \alpha_I + 2$.

Moreover,

$$\Gamma(Q) = \left[\gamma_{rad} \cdot \left(\frac{1}{\lambda} + \frac{1}{Q}\right)\right]^{\alpha_p+1}, \text{ and, } C = \left(\frac{10}{\pi}\right)^{\alpha_n} \cdot \frac{C_I^{\gamma_I} \cdot C_{fus}}{6\pi^2 C_\beta}. \tag{A.20}$$

It is possible to refine the analysis by considering the alpha-particle dilution, different ion and electron temperatures, or different density profiles [12]. However, for the purpose of this chapter, this is not necessary. In fact, I have not explained why the research variables are of interest yet; although, the reader might have an idea with the derivation of Eq. (3.6). This becomes clear when considering Eq. (A.18). ...

REFERENCES

[1] A. Köhn-Seemann, *Fusion research*, Wintersemester, Universität Stuttgart, 2022.

[2] D. A. Hartmann, "Stellarators," *Fusion Science and Technology*, vol. 49, no. 2T, pp. 43–55, Feb. 2006. DOI: 10.13182/FST06-A1103.

[3] W. Picot, "Magnetic fusion confinement with tokamaks and stellarators," *Bulletin of the International Atomic Energy Agency*, vol. No. 2/2021, Jun. 2021, IAEA. [Online]. Available: https://www.iaea.org/bulletin/magnetic-fusion-confinement-with-tokamaks-and-stellarators (visited on 05/13/2023).

[4] ITER, "What is ITER?," 2020. [Online]. Available: https://www.iter.org/proj/inafewlines.

[5] IAEA, "ITER conceptual design report," *International Atomic Energy Agency, Vienna*, 1991. [Online]. Available: https://inis.iaea.org/collection/NCLCollectionStore/_Public/22/064/22064645.pdf.

[6] G. Verdoolaege *et al.*, "The updated ITPA global H-mode confinement database: Description and analysis," *Nuclear Fusion*, vol. 61, Jan. 2020. DOI: 10.1088/1741-4326/abdb91.

[7] I. P. E. G. on Confinement, Transport, I. P. E. G. on Confinement Modelling, Database, and I. P. B. Editors, "Chapter 2: Plasma confinement and transport," *Nuclear Fusion*, vol. 39, no. 12, p. 2175, Dec. 1999. DOI: 10.1088/0029-5515/39/12/302. [Online]. Available: https://dx.doi.org/10.1088/0029-5515/39/12/302.

[8] IAEA, *ITER Technical Basis*. Vienna: IAEA, 2002. [Online]. Available: https://www-pub.iaea.org/mtcd/publications/pdf/iter-eda-ds-24.pdf.

[9] A. W. Leonard, "Edge-localized-modes in tokamaks," *Physics of Plasmas*, vol. 21, no. 9, Sep. 2014. DOI: 10.1063/1.4894742. [Online]. Available: https://www.osti.gov/biblio/1352343.

[10] R. Barnsley, "Radiation hardness in ITER diagnostics," 2014, ITER Diagnostics Division. [Online]. Available: https://www.slideserve.com/cybele/radiation-hardness-in-iter-diagnostics.

[11] G. Verdoolaege *et al.*, "The updated ITPA global h-mode confinement database: Description and analysis," *Princeton Plasma Physics Laboratory, Princeton University*, 2021. [Online]. Available: http://arks.princeton.edu/ark:/88435/dsp01m900nx49h.

[12] Y. Sarazin *et al.*, "Impact of scaling laws on tokamak reactor dimensioning," *Nuclear Fusion*, vol. 60, no. 1, p. 016 010, Oct. 2019. DOI: 10.1088/1741-4326/ab48a5. [Online]. Available: https://dx.doi.org/10.1088/1741-4326/ab48a5.

[13] J. Bakarji, J. Callaham, S. L. Brunton, and J. N. Kutz, "Dimensionally consistent learning with buckingham pi," *arXiv preprint arXiv:2202.04643*, 2022. [Online]. Available: https://arxiv.org/abs/2202.04643.

[14] V. Ostuni, "Impact de la composition du plasma et du rapport d'aspect sur les performances globales du tokamak," *Doctoral School of Physics and Material Sciences (Marseille)*, 2022, thèse de doctorat, in partnership with CEA Cadarache (Bouches-du-Rhône). [Online]. Available: https://www.theses.fr/2022AIXM0345.

[15] S.-E. Skjelbred, *Introductory econometrics*, Summer Semester, University of Oslo, 2015. [Online]. Available: https://www.uio.no/studier/emner/sv/oekonomi/ECON4150/v15/lectures/ols-assumptions.pdf.

[16] A. Murari, E. Peluso, P. Gaudio, and M. Gelfusa, "Robust scaling laws for energy confinement time, including radiated fraction, in tokamaks," *Nuclear Fusion*, vol. 57, no. 12, p. 126 017, Sep. 2017. DOI: 10.1088/1741-4326/aa7bb4. [Online]. Available: https://dx.doi.org/10.1088/1741-4326/aa7bb4.

[17] O. Kardaun, "The tortuous route of confinement prediction near operational boundary - improvement of analysis based on iterhdb4/ldb3 database," International Atomic Energy Agency (IAEA), IAEA-CN–149 IAEA-CN–149, 2006, p. 142. [Online]. Available: http://www-pub.iaea.org/MTCD/Meetings/PDFplus/2006/cn149_BookOfAbstracts.pdf (visited on 04/01/2023).

[18] E. Villalobos, B. Bauyrzhan, and L. Steyn, "Scaling law: DB5 vs DB2.8," *Universiteit Gent*, 2022.

[19] D. Belsley, E. Kuh, and R. Welsch, *Regression Diagnostics*. Canada: Wiley-Interscience, 2004.

[20] A. Lindholm, N. Wahlström, F. Lindsten, and T. B. Schön, *Machine Learning - A First Course for Engineers and Scientists*. Cambridge University Press, 2022. [Online]. Available: https://smlbook.org.

[21] F. A. Viernes, "Linear regression models and influential points," *Towards Data Science*, Mar. 2021. [Online]. Available: https://towardsdatascience.com/linear-regression-models-and-influential-points-4ee844adac6d.

[22] C. Shalizi, "Outliers and influential points," *Carnegie Mellon University*, 2015, Statistics Data Science. [Online]. Available: https://www.stat.cmu.edu/~larry/=stat401/lecture-20.pdf.

[23] J. Goldstein-Greenwood, "Detecting influential points in regression with DFBETA(S)," *University of Virginia Library*, Jul. 2022, Research Data Scientist. [Online]. Available: https://data.library.virginia.edu/detecting-influential-points-in-regression-with-dfbetas/.

[24] K. Chiñas-Fuentes, *Regression analysis for decreased db5*, Accessed: 14 April, 2023. [Online]. Available: https://github.com/Chinnasf/Thesis/blob/main/Optimization/Diagnostics/Regression_Analysis_for_Decreased_DB5_LATEX_IMAGES.ipynb.

[25] A. F. Siegel, *Practical Business Statistics*, Eighth. Academic Press, 2022. DOI: https://doi.org/10.1016/C2019-0-00330-5.

[26] N. Shrestha, "Detecting multicollinearity in regression analysis," *American Journal of Applied Mathematics and Statistics*, vol. 8, no. 2, pp. 39–42, 2020. DOI: 10.12691/ajams-8-2-1.

[27] J. Cheng, J. Sun, K. Yao, M. Xu, and Y. Cao, "A variable selection method based on mutual information and variance inflation factor," *Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy*, vol. 268, p. 120 652, 2022. DOI: https://doi.org/10.1016/j.saa.2021.120652. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1386142521012294.

[28] D. Liao and R. Valliant, "Condition indexes and variance decompositions for diagnosing collinearity in linear model analysis of survey data," *Survey Methodology*, vol. 38, pp. 189–202, Jan. 2012.

[29] S. Šašić, T. Veriotti, T. Kotecki, and S. Austin, "Comparing the predictions by nir spectroscopy based multivariate models for distillation fractions of crude oils by f-test," *Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy*, vol. 286, p. 122 023, 2023. DOI: https://doi.org/10.1016/j.saa.2022.122023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1386142522011714.

[30] M. Inlow, *Applied regression and time series analysis, f-tests and nested models*, Rose-Hulman Institute of Technology, Feb. 2014. [Online]. Available: https://www.rose-hulman.edu/class/ma/inlow/Math485/ftests.pdf.

[31] T. Haslwanter, "An introduction to statistics with python, with applications in the life sciences," *Switzerland: Springer International Publishing*, 2016. [Online]. Available: http://www.springer.com/de/book/9783319283159.

[32] I. Wallach, M. Dzamba, and A. Heifets, *Atomnet: A deep convolutional neural network for bioactivity prediction in structure-based drug discovery*, 2015. arXiv: 1510.02855 [cs.LG].

[33] S. Dieleman, K. W. Willett, and J. Dambre, "Rotation-invariant convolutional neural networks for galaxy morphology prediction," *Monthly Notices of the Royal Astronomical Society*, vol. 450, no. 2, pp. 1441–1459, Apr. 2015. DOI: 10.1093/mnras/stv632. eprint: https://academic.oup.com/mnras/article-pdf/450/2/1441/3022697/stv632.pdf. [Online]. Available: https://doi.org/10.1093/mnras/stv632.

[34] C. Davatzikos, Y. Fan, X. Wu, D. Shen, and S. M. Resnick, "Detection of prodromal alzheimer's disease via pattern classification of magnetic resonance imaging," *Neurobiology of aging*, vol. 29, no. 4, pp. 514–523, 2008. DOI: 10.1016/j.neurobiolaging.2006.11.010.

[35] J. Brownlee, *Why one-hot encode data in machine learning?* MachineLearningMastery.com, 2017, July 27. [Online]. Available: https://machinelearningmastery.com/why-one-hot-encode-data-in-machine-learning/.

[36] L. Chen, "Curse of dimensionality," in *Encyclopedia of Database Systems*, L. LIU and M. T. ÖZSU, Eds. Boston, MA: Springer US, 2009, pp. 545–546. DOI: 10.1007/978-0-387-39940-9_133. [Online]. Available: https://doi.org/10.1007/978-0-387-39940-9_133.

[37] M. Dash and H. Liu, "Feature selection for clustering," *Arizona State University*, Jun. 2020. [Online]. Available: https://www.public.asu.edu/~huanliu/papers/pakdd00clu.pdf.

[38] V. B. Vaghela, K. H. Vandra, and N. K. Modi, "Entropy based feature selection for multi-relational naïve bayesian classifier," *Journal of International Technology and Information Management*, vol. 23, no. 1, 2014. DOI: 10.58729/1941-6679.1062. [Online]. Available: https://scholarworks.lib.csusb.edu/jitim/vol23/iss1/2.

[39] D. J. MacKay, *Information theory, inference and learning algorithms*, Fourth. Cambridge university press, 2014. [Online]. Available: http://www.inference.org.uk/mackay/itila/.

[40] J. Brownlee, *A gentle introduction to information entropy - machinelearningmastery.com*, MachineLearningMastery.com website, Retrieved June 1, 2023, 2020. [Online]. Available: https://machinelearningmastery.com/what-is-information-entropy/.

[41] S. Vajapeyam, "Understanding shannon's entropy metric for information," *Arxiv*, Mar. 2014. [Online]. Available: https://doi.org/10.48550/arXiv.1405.2061.

[42] K. Chiñas-Fuentes, N. Das, J. Gallego, D. Kirhetov, L. Steyn, and P. Manas, "Dimensioning of a tokamak," *IRFM CEA Cadarache*, 2023.

[43] A. Adadi, "A survey on data-efficient algorithms in big data era," *Journal of Big Data*, vol. 8, no. 1, p. 24, 2021. DOI: 10.1186/s40537-021-00419-9.

[44] M. S. Rahman and H. Reza, "A systematic review towards big data analytics in social media," *Big Data Mining and Analytics*, vol. 5, no. 3, pp. 228–244, 2022. DOI: 10.26599/BDMA.2022.9020009.

[45] J. Brownlee, *Step-by-step framework for imbalanced classification projects - machinelearningmastery.com*, Accessed May 07, 2023, 2020. [Online]. Available: https://machinelearningmastery.com/framework-for-imbalanced-classification-projects/.

[46] NVIDIA Corporation, "CuML 23.04.00 documentation," RAPIDS Documentation and Resources. [Online]. Available: https://docs.rapids.ai/api/cuml/stable/#.

[47] L. Buitinck *et al.*, "API design for machine learning software: Experiences from the scikit-learn project," in *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, 2013, pp. 108–122.

[48] T. Dhaene, *Machine learning, gaussian processes*, Faculty of Engineering and Architecture. Winter Semester. Lecture 8., Universiteit Gent, 2022.

[49] *Visualizing a multivariate gaussian*, HashPi, Retrieved June 4, 2023, 2021. [Online]. Available: https://www.hashpi.com/visualizing-a-multivariate-gaussian.

[50] C. E. Rasmussen and C. K. Williams, *Gaussian Processes for Machine Learning*. MIT Press, 2006. [Online]. Available: www.GaussianProcess.org/gpml.

[51] D. K. Duvenaud, "Automatic model construction with gaussian processes," Ph.D. dissertation, University of Cambridge, Jun. 2014. [Online]. Available: https://www.cs.toronto.edu/~duvenaud/thesis.pdf.

[52] K. P. Murphy, *Machine learning: a probabilistic perspective*. Cambridge, Massachusetts; London, England: MIT Press, 2012.

[53] D. J. Rosevear and A. de Waal, "Gaussian processes applied to class-imbalanced datasets," in *Proceedings of the 59th Annual Conference of SASA*, 2017, pp. 41–48. [Online]. Available: https://www.researchgate.net/publication/321533851.

[54] J. Brownlee, *A gentle introduction to maximum a posteriori (map) for machine learning - machinelearningmastery.com*, MachineLearningMastery.com, [Accessed: May 25, 2023], Nov. 2019. [Online]. Available: https://machinelearningmastery.com/maximum-a-posteriori-estimation/.

[55] scikit-learn, *scikit-learn.gaussian_process.GaussianProcessClassifier*, Accessed May 15, 2023, 2015. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.gaussian_process.GaussianProcessClassifier.html.

[56] S. Dutta, *Decision Tree Classification Clearly Explained!* YouTube video, Accessed: May 19, 2023, Jan. 2021. [Online]. Available: https://youtu.be/ZVR2Way4nwQ.

[57] L. Breiman, *Classification and Regression Trees*, 1st. Routledge, 1984. [Online]. Available: https://doi.org/10.1201/9781315139470.

[58] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, no. 1, pp. 81–106, 1986. DOI: 10.1007/BF00116251. [Online]. Available: https://doi.org/10.1007/BF00116251.

[59] S. L. Salzberg, "C4.5: Programs for machine learning by j. ross quinlan. morgan kaufmann publishers, inc., 1993," *Machine Learning*, vol. 16, no. 3, pp. 235–240, 1994. DOI: 10.1007/BF00993309. [Online]. Available: https://doi.org/10.1007/BF00993309.

[60] T. Dhaene, *Machine learning, decision trees and random forests*, Faculty of Engineering and Architecture. Winter Semester. Lecture 9., Universiteit Gent, 2022.

[61] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman, *The elements of statistical learning: data mining, inference, and prediction*. Springer, 2009, vol. 2.

[62] L. Breiman, "Random forests," *Machine learning*, vol. 45, pp. 5–32, 2001.

[63] *Permutation feature importance*, scikit-learn website, 2023. [Online]. Available: https://scikit-learn.org/stable/modules/permutation_importance.html.

[64] D. Morozov, "Greenwald density limit and power balance in tokamaks," *Journal of Physics: Conference Series*, vol. 941, p. 012 009, Dec. 2017. DOI: 10.1088/1742-6596/941/1/012009.

[65] ANS-Nuclear-Cafe, "EPFL researchers update fusion's "Greenwald limit"," *NuclearNewswire*, Jun. 2022. [Online]. Available: https://www.ans.org/news/article-4026/epfl-researchers-update-fusions-greenwald-limit/.

[66] K. Miyamoto, *Plasma Physics and Controlled Nuclear Fusion* (Springer Series on Atomic, Optical, and Plasma Physics, 38), eng, 1st ed. Berlin, Heidelberg: Springer Berlin Heidelberg : Imprint: Springer, 2005.

[67] F. Troyon, R. Gruber, H. Saurenmann, S. Semenzato, and S. Succi, "Mhd-limits to plasma confinement," *Plasma Physics and Controlled Fusion*, vol. 26, Jan. 1984. DOI: 10.1088/0741-3335/26/1A/319.

[68] Helion, "Fusion energy," 2022. [Online]. Available: https://www.helionenergy.com/fusion-energy/.

[69] T. Pütterich, E. Fable, R. Dux, M. O'Mullane, R. Neu, and M. Siccinio, "Determination of the tolerable impurity concentrations in a fusion reactor using a consistent set of cooling factors," *Nuclear Fusion*, vol. 59, no. 5, p. 056 013, 2019. DOI: 10.1088/1741-4326/ab0384.

[70] D. Reiter, G. Wolf, and H. Kever, "Burn condition, helium particle confinement and exhaust efficiency*," *Nuclear Fusion*, vol. 30, no. 10, p. 2141, 1990. DOI: 10.1088/0029-5515/30/10/012.