Modelling and Predicting Acute Ischaemic Stroke Outcomes

Tiago dos Santos
MSc. in Bioinformatics and Computational Biology
Departamento de Ciência de Computadores
2022

Orientador
Luís F. Maia, Centro Hospitalar Universitário da Universidade do Porto
Instituto de Ciências Biomédicas Abel Salazar

Coorientadores
Rui Magalhães, Centro Hospitalar Universitário da Universidade do Porto
Instituto de Ciências Biomédicas Abel Salazar
Pedro G. Ferreira, Departamento de Ciência de Computadores
Universidade do Porto

U.PORTO

FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

Todas as correções determinadas pelo júri, e só essas, foram efetuadas.

O Presidente do Júri,

Porto, ______/______/________

UNIVERSIDADE DO PORTO, PORTUGAL

MASTERS THESIS

Modelling and Predicting Acute Ischaemic Stroke Outcomes

Author:
Tiago DOS SANTOS

Supervisor:
Luís F. MAIA

Co-supervisors:
Rui MAGALHÃES
Pedro G. FERREIRA

A thesis submitted in fulfilment of the requirements
for the degree of MSc. in Bioinformatics and Computational Biology

at the

Faculdade de Ciências da Universidade do Porto
Departamento de Ciência de Computadores

December 12, 2022

"Robot Analysing a Brain in Surrealist Style." produced with DALL-E Flow generative model [1]

" 'All models are wrong but some are useful.' is often attributed to George Box and is used to describe the inherent limitations of all models. All models are an approximation of reality and therefore can never be completely accurate. However, some models can be more useful than others. This is often determined by how well the model captures the important aspects of the system it is trying to represent and how well it can be used to make predictions. "

– GPT-3 [2] explanation of George E.P. Box famous citation in *Science and Statistics & Robustness in Statistics*

Declaração de Honra

Eu, Tiago Filipe dos Santos, inscrito no Mestrado em Bioinformática e Biologia Computacional da Faculdade de Ciências da Universidade do Porto declaro, nos termos do disposto na alínea a) do artigo 14.º do Código Ético de Conduta Académica da U.Porto, que o conteúdo da presente dissertação reflete as perspetivas, o trabalho de investigação e as minhas interpretações no momento da sua entrega.

Ao entregar esta dissertação, declaro, ainda, que a mesma é resultado do meu próprio trabalho de investigação e contém contributos que não foram utilizados previamente noutros trabalhos apresentados a esta ou outra instituição.

Mais declaro que todas as referências a outros autores respeitam escrupulosamente as regras da atribuição, encontrando-se devidamente citadas no corpo do texto e identificadas na secção de referências bibliográficas. Não são divulgados na presente dissertação quaisquer conteúdos cuja reprodução esteja vedada por direitos de autor.

Tenho consciência de que a prática de plágio e auto-plágio constitui um ilícito académico.

Tiago Filipe dos Santos

São João da Madeira, 30 de Setembro

Acknowledgements

Culmination of ideas that propel a person is only possible due to the many people that reduced roadblocks and, for a reason or another, helped gather the willpower to finish daunting tasks. In this brief page to keep them in memory, I would like to leave my thanks to the ones that marked this work the most positively, naming them and their contributions.

First, I would like to thank my family for all the encouragement, for helping me try to reach the high standards in which they regard me, as well as for their constant kindness and support, and Aparecida, for her sweet and understanding nature, even when I was unable to dedicate the time deserved.

I would like to thank Professor and MD Luís Maia for making BioStroke datasets available for this work, for accepting my proposal of extending the project with computer vision components and a greater focus on deep learning, for allowing enough freedom to pursue topics that caught my interest, and for introducing me to many people in the hospital that I had the pleasure to meet and befriend. To Professor Rui Magalhães, I must thank the freedom to pursue the work the way I saw fit and his words of encouragement. I thank Professor Pedro Ferreira, for his availability to supervise, and personal interest shown while mentoring this work and for his students during my time as his student.

I would also like to show my thanks to MD João Pedro Filipe for his time and useful tips on neuroimaging and SECTRA system usage, and for his easy going nature, as well as to MD Ricardo Varela for his welcoming nature and for never being anything but friendly even when I had to occupy his desk to compile the imaging dataset. MD António Costa, MD Joana Fernandes and MD Rui Lopes, I thank for their extensive work compiling the data and typing in missing information in the biomarkers' dataset. I could not forget to name PhD candidate Alexandre Dias, which I thank for sharing his work with me, a valuable guidance to this thesis, and for all the encouragement, tips, and friendliness demonstrated since we met.

Lastly, I would like to thank everyone that contributed to the datasets used in this work. I do not know who all of you are nor did I have the pleasure to get to know all of you, but I would like to thank your effort, and I hope you are adequately named in more significant venues than this.

UNIVERSIDADE DO PORTO, PORTUGAL

Abstract

Faculdade de Ciências da Universidade do Porto

Departamento de Ciência de Computadores

MSc. in Bioinformatics and Computational Biology

Modelling and Predicting Acute Ischaemic Stroke Outcomes

by Tiago DOS SANTOS

Acute ischaemic stroke (AIS) is the second most common cause of death and leading cause of long-term disability worldwide. Fast AIS patient diagnosis and stratification for treatment are fundamental to improve outcomes and reduce disability and lethality. Although it is a thoroughly researched topic, the heterogeneity of human cohorts and brain complexity makes it difficult to create guidelines and models with wide applicability for clinical decision-making in this field. This project used the comprehensive dataset of a prospective cohort of AIS patients submitted to thrombectomy in a Portuguese Comprehensive Stroke Center by applying machine learning and computer vision techniques to develop predictive models on thrombectomy outcomes, using demographic, clinical, biochemical biomarkers and raw imaging data. The goal is to create several models that are optimized for each phase of the patients' admission: at admission to hospital, after neuroimaging and blood work-up and at discharge. This is achieved by selecting the best information available from the datasets provided for this study, trying a vast array of modelling strategies and hyperparameters and creating mixed models for that purpose. For each admission phase and data available, one model is proposed.

The overall best model found was the 3D convolutional neural network (CNN) with basic fine-tuning on a publicly available lung CT scan dataset, which achieved $\overline{AUC_{BasicTransferL}} = 0.97 \pm 0.05$ on validation. However, its generalization capacity may be worse than other models, considering is has shown overfitting in validation curves and only achieved $AUC_{BasicTransferL} = 0.58$ and $F1\text{-}weighted_{BasicTransferL} = 0.66$ on the test set. At admission to hospital, the hyperparametrized logistic regression trained on the strictly selected feature clinical dataset is recommended, achieving

$med(AUC_{LR\_Clin0h\_FS}) = 0.84 \pm 0.07$. When neuroimaging data is available, the abovementioned CNN should be used. At this stage, if hemispheric contrast imaging biomarker is integrated in neuroimaging software, it should be used in conjunction with clinical data, on the proposed Light Gradient Boosting Machine-based model with strict feature selection, which achieved $med(AUC_{LGBM\_Clin0hCA\_FS}) = 0.86 \pm 0.08$ and its result probabilities compared with the ones from the 3D CNN. On the follow-up phase, the 3D CNN results should be compared with the ones from the SVM trained on non-augmented clinical and biomarkers measured at follow-up is proposed, measuring $med(AUC_{SVC\_ClinBiom24h}) = 0.88 \pm 0.08$.

Imaging methods have shown relevance to AIS treatment outcomes modelling, especially when convolutional neural networks are used directly in imaging data. The imaging biomarker studied in this thesis, hemispheric contrast, has shown relevance to modelling, but it did not provide statistically significantly better predictive capacity.

UNIVERSIDADE DO PORTO, PORTUGAL

Resumo

Faculdade de Ciências da Universidade do Porto
Departamento de Ciência de Computadores

Mestrado em Bioinformática e Biologia Computacional

Modelação e Prognosticação de Acidentes Vasculares Cerebrais Isquémicos Agudos com métodos de Aprendizagem de Máquina e Visão Computacional

por Tiago DOS SANTOS

Acidentes vasculares cerebrais (AVCs) são a segunda causa de morte mais comum em todo o mundo e a principal causa de incapacidade a longo prazo. O diagnóstico e estratificação rápidas dos doentes com AVC é fundamental para melhorar prognósticos, e, consequentemente, reduzir a incapacidade e letalidade associadas. Embora seja um tema exaustivamente investigado, a heterogeneidade dos coortes humanos e a complexidade cerebral cria dificuldades na criação de directrizes e modelos com ampla aplicabilidade na tomada de decisões clínicas neste campo. Neste projeto foi utilizado um conjunto de dados abrangente de uma coorte prospetiva de pacientes com AVC que foram submetidos para trombectomia num Centro de Referência de Intervenção na Doença Cerebrovascular. Nesse conjunto de dados foram utilizando métodos de aprendizagem de máquina e de visão computacional para desenvolver modelos preditivos dos resultados de trombectomias, utilizando dados demográficos, clínicos, de biomarcadores bioquímicos e dados de imagem em bruto. O objectivo é criar modelos optimizados para cada fase da admissão dos pacientes: à chegada ao hospital, depois da aquisição inicial de dados de neuroimagiologia e análises laboratoriais, e na alta do paciente. Isto é conseguido seleccionando a melhor informação disponível dos conjuntos de dados fornecidos para este estudo, e criando modelos mistos para esse fim. Para cada fase da admissão é proposto um modelo.

O melhor modelo encontrado foi uma rede convolucional 3D com simples retreino em dados do de TACs pulmonares, tendo conseguido $\overline{AUC_{BasicTransferL}} = 0.97 \pm 0.05$ na validação. No entanto, os resultados no conjunto de teste foram marcadamente piores que outros modelos produzidos, tendo em conta as curvas de validação e por ter conseguido

apenas $AUC_{BasicTransferL\_test} = 0.58$ e $F1\text{-}weighted_{BasicTransferL\_test} = 0.66$. Na admissão ao hospital, uma regressão logística hiperparametrizada e treinada apenas num conjunto de dados clínico estritamente seleccionado é recomendada, tendo atingindo $med(AUC_{LR\_Clin0h\_FS}) = 0.84 \pm 0.07$. Assim que os dados de neuroimagiologia estiverem disponíveis a rede convolutional anteriormente descrita deverá ser utilizada. Se o cálculo do marcador de contraste hemisférico estiver integrado no software de captura e processamento, este deve ser utilizado em conjunto com os dados clínicos base, utilizando o modelo proposto based em Light Gradient Boosting Machine com seleção estrita de variáveis, e que atingiu $med(AUC_{LGBM\_Clin0hCA\_FS}) = 0.86 \pm 0.08$ e as probabilidades de saída comparadas com as da rede convolucional. Na fase de seguimento, a rede convolucional é proposta em conjunto com uma Support Vector Machine treinada sem aumentação de dados em biomarcadores clínicos e biomarcadores medidos no seguimento, com $med(AUC_{SVC\_ClinBiom24h}) = 0.88 \pm 0.08$, uma vez que a espera por análise clínicas adicionais é inconsequente nesta fase da admissão dos pacientes. Apesar do biomarcador de imagem, contraste hemisférico, ter apresentado relevância nos modelos estudados, não permitiu melhorar a capacidade preditiva significativamente.

Os métodos de imagem demonstraram relevância na modelação de prógnosticos de tratamento de acidentes vasculares cerebrais isquémicos agudos, especialmente quando redes neurais convolucionais são utilizadas diretamente em dados de imagem. O biomarcador de imagem estudado nesta tese, contraste hemisférico, mostrou relevância para a modelação, mas não permitiu modelos com capacidade de previsão estatisticamente significativamente superior.

Contents

List of Figures

List of Tables

Glossary

0h	refers to at admission data
24h	refers to follow-up data
a.k.a.	also known as
acc.	accuracy
AF	Activation Function
AI	Artificial Intelligence
AIS	Acute Ischaemic Stroke
ANN	Artificial Neural Network
ASPECTS	Alberta Stroke Program Early CT Score
AUC	Area Under the (ROC) Curve
AUG	refers to models with augmentations
Biom	refers to the full blood counts and biochemical biomarkers dataset
CA	refers to the use of hemispheric contrast
CF	Confounding Factor
CHUP	Centro Hospitalar do Porto
CI	Confidence Interval
CL	Convolutional Layer
Clin	refers to dss. and models using the clinical dataset
CNN	Convolutional Neural Network
CT	Computed Tomography
CTA	CT Angiography
CV	Computer Vision
d.v.	Dependent Variable
DB	Decision Boundary
DL	Deep Learning

ds.	dataset
DT	Decision Tree
e.g.	for example
EIC	Early Ischaemic Change
ENN	Edited Nearest Neighbours
FBC	Full Blood Counts
FN	False Negatives
FP	False Positives
FPR	False Positive Rate
FS	Feature Selection
FWER	Family-Wise Error Rate
GAN	Generative Adversarial Network
GML	Generalized Machine Learning
GNBC	Gaussian Naïve-Bayes classifier
GS	Grid Search
HC	Hemispheric Contrast
hp.	hyperparameter
HP	Hughes Phenomenon
i.e.	that is
i.v.	Independent Variable
IA	Intravenous Alteplase
IB	Imaging Biomarker
ILR	Initial Learning Rate
IM	Image Normalizer
INR	International Normalized Ratio
INR	Prothrombin time and International Normalized Ratio
IP	Image Processing
IS	Ischaemic Stroke
k-NN	k-Nearest Neighbours
LDA	Linear Discriminant Analysis
LGBM	Light Gradient Boost Machine
LR	Logistic Regression
MCA	Middle Cerebral Artery

ML	Machine Learning
MLP	Multi-Layer Perceptron
MMC	Maximum Margin Classifier
MRI	Magnetic Resonance Imaging
mRS	modified Rankin Score
NA	Missing Value
NAS	Neural Architecture Search
NCCT	Non-Contrast Enhanced Computed Tomography
NLP	Natural Language Processing
NLR	Neutrophil Lymphocyte Ratio
NN	Neural Network
OS	Operating System
p.adj.	Adjusted P value
PCA	Principal Component Analysis
pCT	Perfusion Computational Tomography
prep.	preprocessing
QDA	Quadratic Discriminant Analysis
RF	Random Forests
RL	Reinforcement Learning
SOP	Service-Object Pair
stat. sig.	Statistically Significant
Std.	Standard Deviation
SVC	Support Vector Machine Classifier
SVM	Support Vector Machine
TL	Tomek Links
TN	True Negatives
TP	True Positives
TPR	True Positive Rate
UID	Unique Identifier
UP	Universidade do Porto
w.r.t.	with reference to

Chapter 1

Introduction

1.1 Thesis motivation

According to the World Health Organization (WHO), stroke events are the second cause of natural death in adults worldwide, only surpassed by ischaemic heart disease. They are particularly relevant in high-income countries, where they are more predominant and have been increasing their preponderance in the last few decades [3]. Stroke is a neurological deficit attributed to an acute focal injury of the central nervous system (CNS) by a vascular cause [4]. Ischaemic Strokes (ISs) are defined as any neurological dysfunction stemming from an ischaemic event in the CNS. Ischaemic stroke can be acute or transient, differing on the condition's volatility [5]. This study focus specifically on Acute Ischaemic Stroke (AIS) events given that any patient arriving to a hospital and is diagnosed with an IS is treated promptly, proceeding to thrombolysis or mechanical thrombectomy interventions, where it is assumed the IS would not resolve itself in due time [6], intending to provide a tool to forecast the outcome of an AIS.

When an AIS is diagnosed, two main clinical interventions are available for treatment. Intravenous tissue plasminogen activator, IV-rtPA , a technique approved nearly two decades ago for AIS treatment. There is a narrow time window for effective administration — less than 4.5 hours —, and there are several contraindications to its use. Mechanical thrombectomies are usually applied as endovascular approaches to recanalization [7, 8]. State-of-the-art reviews state that both techniques are particularly effective when used in tandem [7], although, around one-third of patients who survive go on to live with long-term disability [9, 10].

1.2 Significance of this Dissertation

Being able to make an accurate prognosis on the treatment outcomes is invaluable to make appropriate treatment decisions after risk-benefit analysis and helps the clinicians to prepare the patient's family for the expected outcomes [11]. As such prognostic risk scores that use patient characteristics to predict functional outcomes in AIS patients are of increasing importance for aiding clinical decisions in stroke management [12], considering the most common treatment options have non-negligible associated risks [13–15], assessment of cost-effective procedures helps resource management for the health unit and the patient [16]. However, pre-treatment information is scarce, usually consisting on a report on the patient's behaviour before admission, a subset of blood work data and imaging data done upon admission [11]. Several formulas and algorithms exist to provide an objective outcome prediction. Most commonly, the National Institute Health Stroke Scale (NIHSS) is used for neurological evaluation, succeeding the modified Rankin Scale (mRS), used for neurological evaluation [17], and the ASPECT score is the most commonly used scoring method for brain imagery [18].

1.3 Dissertation Goals

This thesis intends to improve the selection of patients with AIS for thrombectomy by creating a clinical decision tool to predict individual positive outcomes — i.e., mRS lower than three — calibrated on consecutive stroke patient cohort from a Comprehensive Stroke Center, by creating and selecting machine learning (ML) models adapted to various data availability profiles.

To do so, for each patients admission phase — data collected at hospital admission, former information complemented with clinical analysis and neuroimaging information, and post-thrombectomy follow-up —, the recommended model is ascertained by statistical model comparison.

In a first phase of the project, models are meant to be created using the available tabular data, recurring to a wide selection of modelling strategies. Missing Values (NAs) imputation and data augmentation techniques should be explored to preserve data and improve predictive results.

On a second phase, imaging data is processed and analysed by Neural Networks (NNs) specialized in imaging data, convolutional neural networks (CNNs) and computer vision (CV) techniques. Data augmentation, Network Architecture Search (NAS) and transfer learning should be explored to improve results. The best model represents the proposed model to analyse imaging data in isolation.

On a third phase, merging tabular data and imaging information is possible. Combining selected features from the tabular datasets (dss.) and remodelling allows the creation of a model that accounts demographic, clinical, FBC data, biochemical analyses and imaging data from the patient. These models should be the best models, considering they account for all the data available.

Selected models should only be considered for the demographic analysed and if the data required by each model is complete.

Chapter 2

Background

2.1 Stroke

Stroke is defined as ″**rapidly developed clinical signs of focal (or global) disturbance of cerebral function, lasting more than 24 hours or leading to death, with no apparent cause other than of vascular origin**″ [19]. Stroke is a medical emergency that is a major cause of death and disability, corresponding to approximately 9.2% of the total mortality in Portugal [20], and leaves many patients permanently disabled. Mortality rates of stroke patients aged between 18 and 80 are around 8%, with a standard deviation of approximately 3% [21], and it is the most impacting disease in the Portuguese population [22], stroke treatment costs represent a substantial financial burden on society, should all patients receive treatment.

2.2 BioStroke Study

BioStroke is a prospective study, including patients admitted between January 2019 and March 2020 to the Comprehensive Stroke Centre (CSC) of Centro Hospitalar Universitário do Porto (CHUP). Data retrieved in this project includes complete neurological evaluation, brain neuroimaging and longitudinal blood sampling for biomarker assessment at arrival and follow-ups. Considering there is **no early clinical test that can assess adequately patient's predisposition to a stroke, BioStroke objective is to discover stroke-related biomarkers** [23].

2.3 Disability and Stroke Severity Measures

Hospitalized stroke patients usually incur in some sort of disability, even when receiving successful treatment, so, having an **objective measure** of disability is useful in the clinical context to assess the proper remedial actions and evaluate treatment options against their outcomes.

The **mRS is a clinician-reported measure of disability** usually applied to evaluate stroke patient outcomes, with its usefulness widely supported by literature [24]. mRS six categories describing patients level of impairment are described in table A.1.

NIHSS is another medical assisted diagnostic tool used to **measure stroke severity**, as it quantifies stroke impairment more objectively. It sums the scores of several evaluations on level of consciousness, gaze, hemianopia, facial palsy, motor arm, motor leg, ataxia, sensory perception, aphasia, dysarthria, and neglect. This scale ranks from 0 to 42 and **higher values correspond to greater impairment** [25].

Alberta Stroke Program Early CT Score (ASPECTS) is a neuroimaging based score, calculated by subtracting points for each of 10 distinct Middle Cerebral Arthery (MCA) areas with possible hypoattenuation, **starting with 10 points. One point is subtracted for each hypoattenuated area** and a completed infarct leads to an ASPECTS score of 0. ASPECTS scores between 0 to 7 imply bad outcomes and 8 to 10 usually good outcomes [26]. Common practice recommends that only patients with an ASPECTS of **6 and above should be considered for endovascular treatment** [27]. Given ASPECTS interpretation on a Non-Contrast CT (NCCT) brain scan of AIS patients is challenging and variable, even between stroke experts, automated tool for AIS predictions are desired in clinical practice and research in order to reduce human subjective assessment [27]. These tools are usually Machine Learning (ML)-based programs that vary their performance depending on the training data quality and modelling choices used [27].

2.4 Clinical Imaging Exams

When an AIS patient is admitted, imaging exams are often performed to observe the brain structure and evaluate the lesion according to its location, extension, and aetiology of the lesion. Imaging methods are non-invasive, the most common being NCCT, CT Angiography (CTA), Perfusion CT (pCT) and Magnetic Resonance Imaging (MRI). Although they often can be used to ascertain the same information, each method has its strengths and weaknesses [28]. In this introduction, only NCCT is going to be described since the other imaging methods were not modelled during this thesis.

NCCT is the most **common and simple** imaging method. It improves regular radiographies by enabling the acquisition of a 3D structure, and it allows the identification of any anatomical structures. Their main disadvantage is patient's radiation is much higher than other available options for the same purpose. NCCT is the usual imaging exam, **conducted when a patient is admitted to the hospital**, as a **fast and cheap assessment option**, since it **does not require preparation, nor injection of chemicals** with potential side effects or masking effects [28]. It allows ruling out haemorrhage before thrombolytic therapy, very early signs of ischaemic changes, and hyperdense vessel signs, as well as to review previous infarction areas [28, 29].

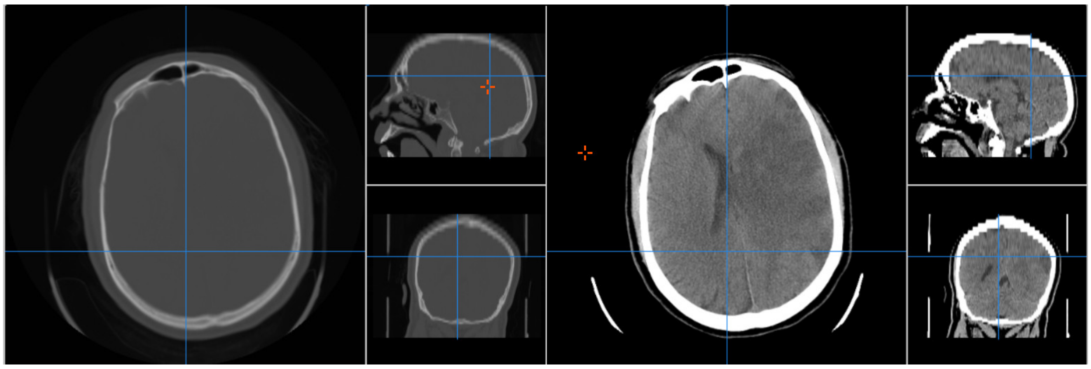


FIGURE 2.1: Orthogonal projections of a NCCT viewed on Mango [30] without proper window, on the left; and with appropriate window for brain features, on the right side.

A CT scan results in a **volume composed by voxels**. These are three-dimensional representations of multiple two-dimensional reconstructions stacked together. These planes — i.e., **slices** — are usually the way that CT scans are presented, and the experts slide through various slices to assess the features of interest. The distance between planes is varied, but it is usually between 2.5mm and 5mm. Since the array of **radiodensities** captured is much higher than the usual colour resolution of the media in which the CT scans are presented, the expert needs to define minimum and maximum radiodensities to be presented, so the image shows appropriate contrasts for interpretation [28, 29, 31]. Radiodensities are measured in **Hounsfield Units (HU)**, and the usual windows for brain tissue analysis are between **0 HU and 80 HU**, and a narrower window can be used when stroke is expected, such as: 20 HU up to 60 HU (W:40, L:40) or 28 HU up to 36 HU (W:8, L:32) [32, 33]. The information contained in a CT scan is three-dimensional, so, modern techniques enable its visualization as a 3D volume using radiodensities reconstruction [28, 29].

2.5 Medical Imaging Data formats

In the context of neuroimaging, a few formats need to be understood, since they are prevalent in this field. **NIfTI, DICOM, Minc, and Analyze** are specialized medical imaging file formats still commonly used in various contexts. Although, for the most part, they are **image encoding formats with some extra metadata**. They encode data in a very precise way and with predefined metadata encodings, — the characteristic that makes them unique. As such, most common imaging programs and libraries cannot interface with them directly and specialized tools are required to manipulate these files, as well as to extract information or convert relevant part to more amenable formats. In this section the most common formats used in this project are going to be summarily described.

2.5.1 Digital Imaging and Communications in Medicine (DICOM)

DICOM is an international standard for medical images and associated metadata, first published in 1993. The standard aims at providing the quality definition that allows imaging files a proper clinical use. Most imaging devices in fields as diverse as odontology to radiotherapy use this standard, being one of the **most widely deployed** healthcare messaging standards used in the world. DICOM standard is recognized by the International Organization for Standardization as the ISO 12052 standard. [34]. This standard includes both the file format definition as well standards for transmission, and storage of clinical data, by defining a data dictionary, data structures, file formats, client and server services, workflow, and compression [34].

DICOM files consist of imaging files with encapsulated **metadata associated to the patient, capture device, image acquisition parameters**, and optionally various other file parameters and clinically relevant data fields. They usually have `.dcm` **file extension**, though not mandatory, as evidenced by SECTRA's file structure. DICOM files contain a file header portion, a metadata portion, and a single Service-Object Pair (SOP) instance. The header is made up of a 128 byte preamble, followed by the characters DICM, all uppercase. The preamble, sometimes used for proprietary data, must contain all zeroes if it is not filled in [35]. DICOM format uses a **separate file for each slice**, so a 3D scan —

such as any CT scan or MRI — uses a series of files to represent the scan, each with its own Unique Identifier (UID). Since each file contains embedded metadata with the patient's and hospital information, anonymizing DICOM data requires thorough handling on all files [35].

DICOMs use a hierarchical structure to its metadata, which includes four main levels: patient, study — the imaging procedure to be performed at a certain date —, series — each part of a study, whether multiple acquisitions, or a series of acquisitions, as in a CT scan —, and instances — corresponding to each individual slice in a series, i.e., the individual files [35].

2.5.2 Neuroimaging Informatics Technology Initiative (NIFTI)

NifTI is a neuroimaging file format, created by an NIH group in early 2000s as an evolution of the formats ANALYZE 7.5 — one of two widespread formats at the time in the research field, the other being Misc [36]. At the moment, it consists in **two sub-formats NIfTI-1 and NIfTI-2** file formats, both usually gathered under the same file extension `.nii`. The main difference between both is that **NIfTI-2** format updates NIfTI-1 to **allow more data** to be stored. NIfTI files are usually more **common in imaging informatics for neuroscience and neuroradiology research**, while **DICOM files are more common in the clinical practice**. Given their widespread usage, several tools exist to convert these two formats, so that image processing pipelines can be readily used with any of the abovementioned formats [36]. Opposed to DICOM files, **NifTI files can contain all slices in a study**, and they can keep order information on each slice. This makes the ds. handling convenient since all slices pertaining a scan are aggregated in a single file, with common metadata present in that file. Although it does not have as many standard metadata fields, the most important pertain acquisition, patient and bureaucratic parameters existing in DICOM files can be encoded in NifTI files [36, 37].

2.5.3 Imaging Data preprocessing

Imaging data formats such as the ones mentioned earlier are able to carry large amounts of personally identifiable patient information, both because they contain a comprehensive set of acquisition metadata that usually is filled in automatically by the software saving imaging data, and because the **high-resolution three-dimensional data can enable facial reconstruction that allows personal identification** [38]. To comply with general privacy legislation, respect patients in the study and comply with the ethical committee considerations, all data should be **stripped of all personally identifiable information** when being analysed [38]. **Metadata can be stripped before acquisition**, and should any relevant metadata persist on the ds., it should be **discarded automatically through software** [39].

After anonymization, the files should be **converted to NifTI**, which is much more convenient when manipulating large collections of CT scans —, and was done with tools such as `dcm2niix` [38]. During this process, one must ensure the data was **converted correctly**, either for elastic distortions in the scans — such as gantry and tilt, which can be usually corrected using the previously mentioned packages —, either by dynamic range compression or off-setting — where values should be within -1024 and 3071 HU [38]. **CT scan types** should not be mixed together, and images with added contrast — such as angiograms — should be treated separately and accounting for the type of contrasting agent used [38]. **Bias field corrections** are usually applied to MRIs, but it can be tentatively used on CT scans and tested as an extra parameter for modelling, since CT scans do not suffer from coil effects such as heating [38], and they can potentially improve light image distortions. In this work, all CT scans with **metal objects induced distortions** too strong for gradient based corrections were omitted. Then, after bias and distortion corrections, **further anonymize** the image by removing body parts not relevant to the study, such as the face, neck, and entire bone structure, making models more resistant to these **confounding factors** [38, 40]. Finally, to remove variability and confounding factors due to size, position or orientation; CT scans should be **registered against a template**. The registration process isolates the brain or skull in each CT scan and applies transformation to it, so it approximates more closely the brain in a template [38].

2.6 Artificial intelligence

One of the greatest advances in computer sciences of the 21st century was the achievement of enough computational power to make widespread use of **Neural Networks (NNs)** [41]. With this advent, these models, mathematically outlined in the 60's, saw booming research and great innovation came to this type of models. Given their great capacity to model arbitrary functions, they allowed researchers to explore them in more scenarios than modelling less structured tabular data [42]. So far, most **Computer Vision (CV)** tasks were done from first principles, using logic-driven algorithms instead of **data-driven algorithms** [43]. Although this process allows more easily explainable and interpretable algorithms that have a more deterministic approach to the tasks they are given, **logic-driven algorithms** are difficult to create and maintain, they require greater human ingenuity and time to develop, and they can be more difficult to adapt to new tasks. Neural Network (NN) algorithms and other types of ML algorithms are data-driven, as an algorithm that designs itself as an approximation to the function that describes the data fed to it, instead of an algorithm designed to replicate a function thought by the developer [44]. These data-driven algorithms, since their **logic is not explicitly coded by a developer, are more commonly associated to Artificial Intelligence (AI)** programs. The main advantage of data-driven algorithms is to exchange human research time by computer runtime, since they can be obtained by brute forcing large amounts of hypothesis/models/calculations [44]. For them to generalize well, they usually require more data, although, fuelled by consumer level internet access inception, new digitized data is created exponentially over time, regarding all aspects of existence.

2.6.1 Machine Learning and Data-driven algorithms

Data-driven algorithms can usually be classified by the way they learn from the data: **supervised learning** — when the developer provides an interpretation to the data, in the form of annotations or a Dependent Variable (d.v.) field —, **unsupervised learning** — when the developer does not provide any information regarding what is meant to be learned from the data, and the algorithm learns relations within the ds. through data distribution and distances —, and **reinforcement learning (RL)** — usually applied to learning actions, by using a learning agent that interacts with its environments through trial and error until it meets a success condition, reinforcing the path that led the agent to that condition [44]. Sometimes, a fourth type of data-driven learning algorithms is considered. **Semi-supervised algorithms**, are in essence pipelines that deal with partially annotated data. In this type of learning, unsupervised learning is applied, but the clusters and associations in the data become labelled based on their association with existing annotated data. Then those samples can be fed again to a supervised learning task to improve its performance [44, 45]. The main issue with this approach is the annotations lack standards of supervised data, where they are based on ground truths, at least by data collection or engineering teams standards, an issue that can amplify classification biases [45]. Not all Artificial Intelligence (AI) data-driven algorithms are NN but much of today's state-of-the-art CV is. The currently controversial and popular GPT-3 [2], DALL-E 2 [46], LAION-based models [47] —such as Stable Diffusion —, Google's LaMDA [81], or AlphaFold [49] are all NNs able to solve a wide array problems not conceivable by the mainstream just a couple of years ago. This versatility is what brings interest to this topic, and the reason why a Deep Learning (DL) approach was attempted on this project. All the previous NN-based models are also examples of supervised learning or semi-supervised learning — where gigantic dss. with annotated data were fed to these models of which, modern trends suggest that even greater amounts of data are needed to fully utilize each deep learning architecture to its full potential [50]. **Supervised models attempt to learn the function that minimizes the difference between modelled calculation of all Independent Variables (i.vs.) and the ground truth** [51].

Machine learning (ML) is a sub-type of AI and data-driven algorithms where structured or unstructured data is used to make predictions, usually as **classification** or **regression** problems [52].

2.6.2 Models description

Several **ML strategies** exist to learn from annotated data through supervised learning. These are encoded as algorithms with different calculation methods, assumptions regarding data, balances between data distribution assumptions and the modelled function — the **model's bias** —, and the way it adapts to differences in the ds. — the **model's variance** [52]. In this subsection, a brief description of each strategy used in this thesis follows.

One of the most ubiquitous ML tools is the **Logistic Regression (LR)**. As a special case of **generalized linear models**, **LR** assumes the d.v. is modulated by linear i.vs., but unlike **linear regression**, **LR** is designed for **classification** problems, where the d.v. is the positive class probability [51]. Like **linear regression**, it also assumes no collinearity between i.vs. — although it is still robust under non-ideal circumstances —, and it requires cases count to be greater than feature count [51]. Considering a **Decision Boundary (DB)** set at $p = 0.5$, can be found for LR by solving the sigmoid function:

$$f(x) = \frac{1}{1+e^{-x}}\,, \tag{2.1}$$

where x is a regressive input function. One **LR advantage** over other modelling strategies is it does not assume i.vs. are normally distributed, and its d.v. distribution is expected to follow the Bernoulli distribution [51, 53]. One important **LR limitation** is its **linear**

DB assumption, which makes it unable to model more complex interactions between variables [53, 54].

Also assuming a linear DB, is the **Linear Discriminant Analysis (LDA)**. This classifier, unlike LR, assumes that observations come from a Gaussian distribution and the covariance matrix for all classes to be classified is identical. When those assumptions are not met, it provides unreliable results. LDA improves on the **Bayesian Optimal Classifier** [54].

Another model used, derived from LDA is **Quadratic Discriminant Analysis (QDA)**. It evolves LDA formulation by accepting a different covariance matrices for each class, allowing more complex Decision Boundaries (DBs) to be defined. When that condition happens — the most usual circumstance — the DB is expressed by a quadric formula, the reason for this modelling strategy name [54]. When **QDA** covariance matrices are all assumed to be diagonal, it is equivalent to Gaussian Naïve-Bayes classifier (GNBC) [55].

GNBC is a classifier modelling strategy that assigns class labels to problem instances, where those are classified as proportions of the outcome to be predicted, based only on the data available on the ds., as approximations to the population's proportion. After decomposing the ds. and doing those estimates, a maximum likelihood evaluates classes' combination influence using **Bayes probabilities** as defined by the conditional probability given by:

$$p(C_k \mid x) = \frac{p(C_k)\, p(x \mid C_k)}{p(x)}, \tag{2.2}$$

where C_k is the set of Independent Variable (i.v.) classes and x is the vector with class representations of every feature in the observation. **GNBC** assumes that each of these features is independent of the others, so the conditional probability $p(C_k \mid x)$, where x is input vector. The main advantage of GNBC is that in dss. where features and classes are indeed independent, it can perform well with small training sets [54].

Decision Trees (DTs) are a very different type of modelling strategy to the ones presented above. It defines models as a branched function, creating a hierarchical structure with the most important classification choices in the upper levels — the ones that reduce entropy the most — and refines the model at each extra level with choices that enable a further reduction of the overall classification entropy. Entropy metric is defined as

$$D = -\sum_{k=1}^{K} \hat{p}_{mk} \log \hat{p}_{mk}\,, \tag{2.3}$$

while the Gini Index, on the other hand, is defined by

$$G = \sum_{k=1}^{K} \hat{p}_{mk} \left(1 - \hat{p}_{mk}\right), \tag{2.4}$$

where D is entropy, G is the Gini Indez, K is the total number of classes of the outcome, and $\hat{p}_{mk}$ is the proportion of class k observations in node m of the tree. It is preferable in **DTs** and associated methods to use the more common **classification error rate** due to its sensitivity. The main advantages of DTs is that they work well with limited data, they are among the most **explainable and interpretable** models, and they can map **very complex DBs** [54]. However, given **Decision Tree (DT)** models **high variance**, especially when configured to overfit, models are unstable, varying significantly with the underlying subset of data in use, they give more preponderance to categorical features with more classes, and models with many features can have structures empirically hard to understand [54].

The main drawbacks of DTs can be overcome with ensembles. **Ensembles or ensemble methods** are models that use the output of learning models' groups to produce their predictions. In theory, the errors of individual models are diluted through majority voting, and the errors of each learner can be emphasized by the next in the ensemble, so those errors are avoided. One major **ensemble methods drawback** is the obfuscation of individual learners, so the models created are opaque and, therefore, less explainable and interpretable.

Random Forests (RF) is one popular ensemble method, using as learners **DTs**. They use bootstrapping for each of their individual trees, and an ensemble process known as bagging, which is mathematically defined as

$$\hat{f}_{\text{bag}}(x) = \frac{1}{B} \sum_{b=1}^{B} \hat{f}^{*b}(x), \tag{2.5}$$

where $*b$ is the number of bootstrapped dss., and B is the total number of bootstrapped dss.. In **bagging**, learners are grown independently, but RF enhances the randomization of data available to each learner, by randomly subsetting features at each split of each DT [54]. Bagging allows DTs to use the entire ds. on each learner but having different perspectives on it, and feature subsetting avoids overrepresentation of features with many categories, making these models usually **perform better than DTs** [54].

The other ensemble methods used in this thesis are **Adaptative Boosting** — better known as **AdaBoost** —, **Extreme Gradient Boosting** — better known as **XGBoost** — and

Light Gradient Boost Machine (LGBM). Unlike RF, they all use **boosting**, which is an ensemble of weak learners — i.e., models intentionally underfit — where learner are grown in succession, gives more weight to the misclassified samples [54]. **Boosting** is known for its top performance in many dss., and **AdaBoost** became notorious in early data science competitive scenarios [56]. It has less hyperparameters (hps.) so, it is easier to configure than previous methods, and like all boosting methods, usually performs much better than individual learners. As disadvantages, it is sensitive to noise, and it may become more easily biased by irrelevant features than competing methods [57, 58]. **XGBoost** improves on AdaBoost by adding automatic Feature Selection (FS), individual tree penalization, proportional leaf nodes' shrinking, a better method for solving the optimization problem — Newton's method —, and it can take advantage of parallelized resources [57, 59]. Finally, **LGBM** is a competing method to XGBoost. Though it is technically similar — despite differences in the method for split finding, and the ability to deal with categorical variable without prior preprocessing —, it can leverage GPU resources, and several heuristics, lead to much faster execution than **XGBoost** with equivalent results [60].

Another important modelling strategy experimented on during this thesis is **Support Vector Machines (SVMs)** . Typically, used as classifiers, but, due to their algorithm exponential complexity, best suited for moderately sized dss.. They are classifiers **robust to outliers** that have become known for their speed and **high performance** in classification, especially in applications where **NN** used to be applied. Their robustness to outliers stems from their base principle: **SVMs** and **Maximum Margin Classifiers (MMCs)** try to find their DB by finding the furthest points from each classes' centroid that are the closest to other classes' boundaries — known as **support vectors** [54]. Given that often there is an intersection between the volumes where samples from multiples classes occur, so the MMCs cannot provide a solution. **SVMs consider a soft margin**, where points of various overlapped classes occur, and each of those points weights on the DB. In this way, outliers, having a boundary that only considers the points within the margin. This can be mathematically described as:

$$\underset{\beta_0,\beta_1,\ldots,\beta_p,\epsilon_1,\ldots,\epsilon_n,M}{\text{maximize}} \quad M \tag{2.6}$$

$$\begin{aligned} &\text{subject to } \sum_{j=1}^{p} \beta_j^2 = 1, \\ y_i\left(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}\right) &\geq M\left(1 - \epsilon_i\right), \\ \epsilon_i &\geq 0, \sum_{i=1}^{n} \epsilon_i \leq C, \end{aligned} \tag{2.7}$$

where M is the margin's width, C is an arbitrary tuning parameter, and ϵ_i is the slack variable, controlling the number of points to intrude in opposite side of the DB [54]. SVMs have a rigid assumption regarding their DB shape — which would be linear in the regular case —, but they are more versatile than other methods given that **kernel transformations** can be used to adapt to different DB shapes. Considering an **SVM kernel with no transformations** is defined by:

$$K(x_i, x_j) = \langle x, x' \rangle, \tag{2.8}$$

where K is the kernel transformation, $\langle\rangle$ is the inner product space operation, and x' is the derived feature set, in this case, without transformation, which produces linear DBs. **Quadratic boundaries** can be achieved by using:

$$K(x_i, x_j) = e^{-\frac{\|x_i - x_j\|^2}{2\sigma^2}}, \tag{2.9}$$

or any other **polynomial DB** by using:

$$K(x_i, x_j) = (\gamma x_i^T x_j + r)^d, \tag{2.10}$$

where σ, γ and r are kernel parameters and d is the polynomial transformation degree [54]. SVMs become popular due to their versatility, allowing them to model complex CV and Natural Language Processing (NLP) tasks — especially to their robustness while modelling highly-dimensional dss. —, and Support Vector Machine (SVM) derivative methods are still used since they achieve results comparable to NNs while having less parametrization effort, performing faster training and inference on moderately sized dss., [61] but tend to have worse generalizability than NNs for the same volumes of data [54].

2.6.3 Artificial Neural Networks

Artificial Neural Networks (ANNs), most commonly referred as **Neural Networks (NN)**, are one type of ML model inspired by its biological counterpart. Given their ability to model complex feature iteration and DBs, they are often used as **alternatives to SVMs for**

CV and NLP tasks, and historically, their popularity increased or dwindled depending on computing power and relative execution time to SVMs [54]. All types of **Artificial Neural Network (ANN) mimic biological brains by emulating an aggregate of individual units** — artificial neurons or perceptrons —, but unlike natural neural networks where the start portion is hard to assess, these networks have a start layer, **input layer**, several ordered layers to which information is passed through, **hidden layers**, and a final layer where the results are expressed and evaluated, **output layer** [52, 54]. The most common layers in ANNs pass signals to all units of the next layer, **forward propagation**, and adapt their function based on ground truth feedback, **the back propagation mechanism** [52]. These layered unit connections emulate the synapses and can connect to other parts of the network depending on the **architectural type** used. Some ANN formulations can pass information in different orders as an attention mechanism — as in most NLP focused **Long Short Term Memory (LSTM) ANN** —, they can pass information to reconstruction layers several layers after the subsequent one — also in some types of generative models —, they can feed the calculation of an aggregate function to be feed to earlier layers — as in **relational neural networks** (RNN) — and various other architectural choices made to improve performance and allow even greater flexibility from this modelling strategy.

While in **biological NNs** the signal is transmitted biochemically, either as an electric signal mediated by ionic transfers or via more chemical signals — neurotransmitters received by G-protein-coupled receptors —, in ANN the signal is transferred as numbers, where the values are mediated in each of the graphs units — also called neurons or perceptrons — and via their edge weights — equivalent to synaptic impulse response [52, 62, 63].

Considering ANNs are human made, their internals are well-defined. On a **feed-forward pass**, signal passed to a single output is given by:

$$f(x) = \beta_0 + \sum_{k=1}^{K} \beta_k h_k(x), \tag{2.11}$$

where K is the number of activations, X is the input tensor — or previous layer tensor —, h_k is the hidden layer function, β_0 is the usual bias neuron added to each layer, and β_k are the weights of every other neuron in the layer [54]. The **hidden layer function** for each neuron is given by:

$$h(k) = g\left(w_{k0} + \sum_{j=1}^{p} w_{kj} X_j\right), \tag{2.12}$$

where $g(z)$ is an **Activation Function (AF)**, w is the edge weight, k is the layer sorting number, X is the tensor, and j is the sorting number of the neuron within the layer [54]. Activation Functions (AFs) control which signals pass to the following layer [52]. **Linear activation**, $g(z) = z$, is rarely used, since negative uncontrolled biases usually leads to vanishing gradients — a condition where the signal flow stops and no further learning is possible. The most commonly used AFs are the **sigmoid function** [54], defined as

$$g(z) = \frac{1}{1+e^{-z}} \,, \tag{2.13}$$

and the **Recurrent Linear Unit (ReLU)** function, defined as

$$g(z) = max(0.0, z) \,, \tag{2.14}$$

although many other AFs are available [55]. By limiting values between 0 and 1, the **sigmoid function** avoids drastic adjustments to weights that avoid progressively finding a local minimum for the gradient — the function intended to be minimized to find the best solution. The **ReLU** function avoids gradients to be transformed in 0 values in deep NNs — deep as in having many layers [52]. The **step function**, is the simplest AF defined as:

$$g(z) = \begin{cases} 1 & \text{if } z \in A \\ 0 & \text{if } z \notin A \end{cases} \,, \tag{2.15}$$

where A is an *a priori* defined interval [55]. Other functions, such as **hyperbolic tangent**, work a middle ground between the step function or as small variations of other more common AFs [54].

Hidden layers are defined to acquire representations of the data at different scales, or in special cases to emulate logical circuits that allow the design of special modelling function [64].

2.6.4 Automated Machine Learning (AutoML)

The ML preparation process is complex, even with all modern tools that speed up the process, it requires a fair amount of technical knowledge, and it is very time-consuming to tweak and test all possible combinations to provide the best model, given any ds.. Making a robust model requires: data preparation, record selection, NAs inference, column data type detection, column intent detection – relevant for transformations, e.g., target/label, stratification field, free text field –, feature engineering and selection, task detection, hyperparametrization, metrics selection, data leakage detection, ensemble creation and final model selection [65]. In extreme cases, it can extend to meta-learning, transfer learning, pipeline section within time, memory and complexity constrains, results' analysis and creation of user interfaces and adequate visualizations [66, 67]. In this context, automatizing the process of creating a model, **automated machine learning (AutoML)**, even if at a cost of great computational time becomes highly desirable, given that technical expertise and experts time to dedicate to each problem is limited [65, 68, 69].

Such systems have been incepted partially in one form or another from ML early days, but it was with **AutoWEKA** that automated ML began by providing automation of the combined algorithm selection and hyperparameter (hp.) selection of 39 classification algorithms using Bayesian optimization algorithms to find solutions within the defined time constrains [68]. This type of automated ML is often referred to as **Generalized Machine Learning (GML)** [65]. **Auto-SkLearn** continued the work started by **WEKA** [67], and applied the same concepts to the **SciKit-Learn library** [70, 71], a modern ML library, also used in this thesis for modelling all models but the ones requiring deep networks with custom features — i.e., Convolutional Neural Networks (CNNs) —. **Auto-SkLearn**, another GML tool, improves on Auto-WEKA by identifying instantiations of ML frameworks that perform well on a new ds. and starting **Bayesian optimization** with them; by automatically constructing ensembles of the models considered by the search space strategy; and by designing a highly parametrized ML framework from high-performing classifiers and preprocessors implemented in `scikit-learn` [67]. A development of this library, released as **Auto-SkLearn 2.0**, improves on the first version by improving modelling on big ds. performance, and by introducing the **Portfolio Successive Halving (PoSH)**, which reallocates more resources to promising pipelines to meet the goals within strict time constrains, and by automatizing optimization policy selection [72]. Both Auto-SkLearn 1.0 and 2.0 have the disadvantage of **not being available for Windows** OŞes. While these libraries automatize the search for the best models with conventional ML purposes, and they use some simple NN, they are not the most suited for search for complex NN architectures or deep NN exploration. Based on PyTorch, a mature and widely used library for advanced deep learning, **Auto-PyTorch focuses only on automating models based on NNs** [73], a subset of automated ML often called AutoDL [65]. As such it focuses on tools that better parametrize this types of models — searching for new architectural designs based on existing architectures —, it thrives where DL large amounts of data with complex relations with the d.v.. Other tools of interest is **AutoGluon**, an automated GML tool that also has advanced DL features, and is much more versatile than the previous one since it **can handle image data, time series and multimodal models**, that combine simple two-dimensional inputs with tabular data [74]. **H2O AutoML** is another freely available automated GML tool with limited DL capabilities — a part of H2O open-source library [75] —, and formal reviews have highlighted its excellent performance and efficiency in tabular data [65].

2.6.5 Convolutional Neural Networks (CNN)

CNNs are specially designed feed-forward NNs — i.e., without loops or recurrence — that account for the spacial relation of data in the tensor they are encoded in, and for that reason, they are useful while modelling all sorts of imaging data. They do so by learning *convolutional filters* [52] instead of single weights. **Convolution operations** can be expressed as

$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau)g(t-\tau)\,d\tau\,, \tag{2.16}$$

where the continuous function f is the main function and g is the **filter**, both usually represented as discretized functions in the form of tensors, the symbol $*$ is the **convolution's representation**, and τ is the positional representation where the filter should be applied. For strictly discrete functions — as it is the case in Convolutional Neural Networks (CNNs) —, the function can be expressed as:

$$(f * g)[n] = \sum_{m=-\infty}^{\infty} f[m]g[n-m], \tag{2.17}$$

where n are positions to be calculated and m is the positional representation of the filter [76]. These filters are smaller tensors that, through the convolution operation, produce a resulting tensor with distinct results. **Filters are commonly used in computer vision (CV)** tasks and image processing software has popularized many of these, such as the blurring with the Gaussian filtering, edge detection using the Laplacian and Sobel filters, or even with an aggregation of filters — such as the difference of Gaussians [77]. In the CNNs' case, each convolutional filter is designed for each part of the input [52]. Individual convolution calculations can be considered a summarized image feature, so although CNNs deal with complex data, no laborious feature extraction is to be conducted beforehand, for them to produce results [52].

In a **typical CNN architecture**, four basic components are included: a **local receptive field**, **shared weights**, **pooling** and **fully connected (FC) layers**, as in regular ANN.

Several Convolutional Layer (CL)s are stacked with pooling layers and one or more fully connected layers at the end of the network to form a deep CNN architecture [78]. Two-dimensional CNN extract spatial features from input data while one-dimensional CNN extract spectral features. By extracting both spectral and spatial features simultaneously from the input volume, 3D CNNs can take advantage of both 1D and 2D CNNs architectures. In **3D CNN**, an extra dimension is added to the mathematical formulation of 2D CNN as a CL can be mathematically defined by:

$$x^{\ell}_{i,j,k} = \sum_a \sum_b \sum_c \omega_{a,b,c} y^{\ell-1}_{(i+a)(j+b)(k+c)} + \beta^{\ell}, \tag{2.18}$$

where $x^{\ell}_{i,j,k}$ is an input x as a **three-dimensional tensor** of size $i \times j \times k$ where i, j and k are iterators for each dimension, and ℓ is the hidden layer ordinal position of said CL, a, b and c are the numbered parameters for each iterator, $\omega_{a,b,c}$ is the weight w.r.t. the pertinent iterator, β is the bias for that level, and the output function $y^{\ell}_{(i+a)(j+b)(k+c)}$ corresponds to $f(x^{\ell}_{(i+a)(j+b)(k+c)})$ [52, 78].

Although the convolution operation in CLs condenses information regarding a kernel sized tensor of points, it is repeated almost as many times — or as many times if padding is used —, so the number of parameters passed to forward layers can remain the same, which is usually ineffective in terms of computing resources. **Pooling layers** are used to reduce dimensionality of previous layers, and they correspond to a filter that usually calculates either the average or maximum value with its confines and outputs it as a single value. The entire tensor is recurred with this operation, using a step value — called stride — that denotes how many cells are advanced on each iteration. The pooling layer new dimensions are given by:

$$(i^{\ell}, j^{\ell}, k^{\ell}) = \frac{(i^{\ell-1}, j^{\ell-1}, k^{\ell-1}) - (\pi_i, \pi_j, \pi_k) + (1,1,1)}{\sigma}, \tag{2.19}$$

where π is the defined **pool**, and σ is the **stride** [52, 79].

Regularization layers are usually needed considering CNNs high potential for overfitting any given data. **Regularization** allows the network to be less sensitive to sampling noise in the data — especially relevant when data is limited —; therefore, reducing overfitting and improving models generalizability [52]. This can be achieved with **dropout layers**. These layers copy to themselves the previous layer, but *drop* — i.e., set to 0 —, an arbitrary proportion of random units at each step of the training process; consequently, reducing the importance of some units on some steps, to make their effect temporarily less important in the training process [52, 79].

Batch normalization is another important component in Convolutional Neural Network (CNN) — and NN in general. Batch normalization normalizes values from a layer — as codified in its respective tensor — by their mean and variance values. This is important because NNs can converge faster when values are centred around zero, given AFs usually assume the data fed to them has a Gaussian distribution [52, 78]. This allows setting up higher Initial Learning Rate (ILR)s which further speed-ups finding convergence close to a global minimum [78].

Although other modelling strategies, such as SVMs, can be applied with success in imaging data [80], and NN are usually much **less interpretable** — i.e., understand the relation between input data and output predictions — and explainable — understand how the model processes information, and what each of its trained parameters means — than other ML alternatives, they can be parametrized to become more flexible to input data, and therefore, to be able to adequately model more complex problems. This is one of the reasons complex tasks involving images usually use DL models extensively, as the recent advances from OpenAI with DALL-E 2 [46] or Google Parti [48]. This flexibility has to do with the fact **NNs behave as logical circuits** [64] — hard to decode logical circuits nonetheless — and can be hyperparametrized by architecture engineering to emulated virtually any function [82, 83].

2.6.6 Training

After the **CNN's architecture** is defined, the model is trained with the typical NN procedure. This is an iterative process using one of various types of **gradient descent** [52]. The gradient is the first derivative of function that operates in a tensor of at least two dimensions [84]. The objective of **gradient descent optimizers** is to minimize a cost function, evaluating the change in the tensor that represents the cost function and, depending on the cardinality and magnitude of the calculated loss, adjust via back-propagation an appropriate correction to the weights and biases of all former layers before processing the next step in the training process [52]. 3D CNNs most commonly used **loss function** is **categorical cross-entropy (CCE)** [78], since Mean Squared Error (MSE) is mostly used in regression analysis. CCE can be calculated as:

$$CCE = -\Sigma t_i log(s_i) \; , \tag{2.20}$$

where t_i is the iterator corresponding to the ground truth and s_i is the predicted probability vector [52]. **Gradient descent optimizers** use **heuristics** to find their optimum value, since usually the gradient is not perfectly convex, the training set size may not fit into memory and the parameters defined for the search — such as ILR — are usually approximations to the perfect settings to be defined, which would make the problem of ascertaining a direction to be taken an intractable problem [52]. Calculating the loss function for images is a memory intensive process, so, usually this is done in **mini-batches** — i.e., subsets of the whole training ds. that, as a whole, can fit into memory available in the equipment to be used during training —, a heuristic referred to **Stochastic Gradient Descent (SGD)**. Considering that stochastic nature, proper parametrization is key for good performance, and batch size should be as large as memory permits and the ILR should be selected with care, since large values can constantly pass over global minimums and excessively small values make the algorithm take too long to reach a good solution [52, 78]. Another popular optimizer is **Adam** that evolves the concept of SGD by adding **adaptative momentum**, the reason for its name. Momentum makes the gradient descent vector to have influence in subsequent updates — therefore similar to momentum in physics —, and it is able to update momentum at the end of each epoch. This is particularly important for extremely noisy gradients [85], as it is the case when modelling imaging data.

While training, each neuron uses convolutional filters that can take inputs shared with other neurons, as defined by

$$I * K_{x,y} = \Sigma_{i=-1}^{1} \Sigma_{j=-1}^{1} I_{x+i,y+j} * K_{S+i,S+j} \; , \tag{2.21}$$

where I is the input tensor, x, y are input coordinates, K the filter, i, j offsets to the filter's centre, and S the filter's size minus one [86], if stride is smaller than S. Given imaging data dimensions, each neuron learns very localized local data from the previous layer, that, when combined with enough data and adequate data augmentations, enables **shift invariant representations of the data** [54].

ANNs can continuously improve their adjustment to the data until they converge, and since they can map out extremely complex functions, they can converge when all training data is perfectly adjusted. Usually when models are tested in validation or test sets, they perform considerably worse, implying overfitting [54]. A model should be trained while it improves validation and test scores, and stop training it when this is no longer the case. **Early stopping** mechanisms can control this in various models, such as NNs and DTs, acting as a regularization technique [52]. Early stopping is an imposed condition for the model to stop trying to minimize the loss function once the validation metrics do not improve more than a set value, or for a certain amount or training epochs [79].

Transfer learning is an ML method where an algorithm's learning product can be used to warmstart the learning of a second related task, accelerating it. This happens because parts of the network may be trained to discern common partial patterns [87].

2.6.7 Metrics

Metrics are needed to evaluate each model, to assess the fitness of a model to a function, and to compare performance with previous models [88]. The **F1-score** keeps this balance as a harmonic mean, as defined by:

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{2.22}$$

When applied to segmentation task and calculated as a measure of pixel overlap of all samples, the same formula is referred to as **Sørensen–Dice coefficient** or **Dice similarity coefficient** (DSC). The **weighted F1-score** gives weight proportional to the predicted class proportion. Given it is more important to correctly evaluate cases excluded from treatment, and there is slightly more cases with bad outcomes, weighted averaging is more adequate than macro averaging, which balances class representation by increasing minority class weight [88].

Accuracy (ACC) represents the proximity of the measurement results to the true value. It is defined as

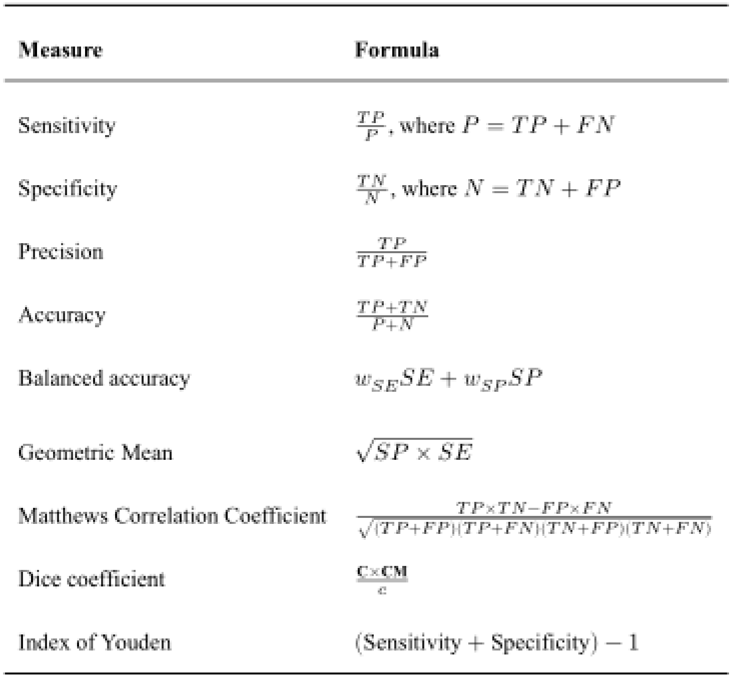
$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.23}$$

Balanced accuracy is more adequate when class imbalance is present and extends the concept of accuracy by averaging recall across predicted classes [89], so when possible weighted version of it and F1-score were used.

Calculation of all relevant binary classification metrics can be consulted in table 2.1.

While absolute metrics indicate how the models perform in a given test set, they do not show how the models fail to classify some samples. Models can have rigid assumptions — as the models with linear boundaries previously mentioned —, and therefore they are

TABLE 2.1: List of main classification metrics and their calculation method. Note that specificity is also known as True Positive Rate (TPR), and sensitivity is also known as True Negative Rate (TNR). Table retrieved from Pereira *et al.* [90].



Measure	**Formula**
Sensitivity	$\frac{TP}{P}$, where $P = TP + FN$
Specificity	$\frac{TN}{N}$, where $N = TN + FP$
Precision	$\frac{TP}{TP+FP}$
Accuracy	$\frac{TP+TN}{P+N}$
Balanced accuracy	$w_{SE}SE + w_{SP}SP$
Geometric Mean	$\sqrt{SP \times SE}$
Matthews Correlation Coefficient	$\frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$
Dice coefficient	$\frac{\mathbf{C} \times \mathbf{CM}}{c}$
Index of Youden	(Sensitivity + Specificity) $-$ 1

less prone to overadjust their function to the ds. they encounter — they have a high bias. This can be an advantage, since when modelling one wants to obtain the base principles that distinguish multiple class, so high bias models have a lower tendency to **overadjust to noise in the data, a condition denominated overfitting**. On the other hand, supposing the distinction between classes is better represented by a quadratic function — a more complex DBs —, a linear boundary does not capture well enough the complexity required by a good model. It underfits the data [54]. **Every time a model is parametrized to reduce overfitting, by making the model have more rigid assumptions or by simplifying its output, it increases its underfitting potential**. If a model is parametrized to reduce its underfitting to the data — e.g., by increasing the depth of a decision tree —, it is more prune to capture noise in the data, and, therefore, overfit. Overfitting makes the model change its DBs more markedly with different observations, a characteristic called model's variance. To achieve the best results one has to evaluate which bias-variance trade-off is best suited for the data at hand [54].

Regularization methods such as dropouts, data augmentation, and early stopping can improve a model's generalizability without seriously impacting training performance [52].

Chapter 3

State of the Art

3.1 Outcomes' Predictive Modelling

Prognostic risk scores using patient's characteristics to predict functional outcomes in AIS patients are of increasing importance for clinical management decisions [12], since common treatment options — such as thrombectomy — have non-negligible associated risks [13, 14, 91]; assessment of cost-effective procedures helps health unit and the patient's resource management [16]; and, the patient and their family appreciate informed predictions on treatment outcomes [92]. There are numerous previous studies on this field, but few have modelled AIS outcomes for CHUP reality. Commonly used AIS outcome models include **ASTRAL** [93], a useful model since it uses only six features readily available in any patient's admission: age, NIH Stroke Scale, time from onset to admission, range of visual fields, acute glucose, and level of consciousness; **DRAGON** score [94], used to predict 3-month outcomes for patients engaging in thrombolysis; **THRIVE** score [95] uses age, NIHSS, and the previous history of atrial fibrillation, hypertension and diabetes mellitus to predict the 3-month outcome. Other widely used publicly available models include **MR PREDICT** [96], which includes **ASPECTS** information and other radiological information, or have been extended to include it, such as the **MT-DRAGON** [97], **mHIAT** [98], **HIAT 2** [99], or **mTHRIVE** [98]. Based on the data collected on the patient, the stage of patient's admission, and predictive accuracy, one of these models may be preferred to others. However, all these models were created on foreign ds., that may have factors influencing their cohorts not accounted for in the model, and, depending on the model characteristics and differences between the training cohort and the one where the model is applied to, may require calibration for optimal applicability, since poorly calibrated algorithms can be misleading and potentially harmful for clinical decision-making [100]. Examples of AIS outcome modelling **calibrated for the Portuguese reality**, include the study from Monteiro *et al.* 2018 [101], which creates predictive models for AIS outcomes, considering good outcomes as $mRS \leq 2$ — like this thesis, and comparing common ML strategies with ASTRAL, THRIVE, and DRAGON — public state-of-the-art models to predict AIS outcomes. A recent biomarkers' study incepted in Centro Hospitalar do Porto (CHUP) by the neurology team assesses biomarker association with AIS outcomes, and models they present as an outcome predictor [102]. These previous studies and resulting models use a **limited array of predictor features**, which is convenient for widespread usage, but may miss important information for accurate inferences. The more features required, the less generalizable the model becomes. In this thesis, to allow a malleable number of features collected, models were created with different feature sets and their performance compared among them, so importance of specific data collection for outcome prediction can be learnt, and only then, statistical feature selection was performed. However, added features incur in potential costs, data collection time and data insertion time, so **parsimony** should be considered in the neurology domain, and was taken into account when selecting models. These improved models may help reduce cost and speed up decision-making processes, if certain clinical analysis are shown not to improve inference. Yet, this may reveal new chemical biomarkers and Imaging Biomarker (IB)s of importance for an accurate prediction, which may suggest their inclusion in standard clinical protocol.

3.2 Classification on 3D Medical Imaging

Cui *et al.* implemented in 2021 a 3D CNN classifier based on Inception NN using **diffusion weighted images**, which they denominated as **DeepSymNet-3D-CNN** [103]. Cui *et al.* adapted the Inception architecture by replacing its 2D convolutional and pooling layers by their 3D counterparts, and each volume was divided in halves and analysed in the first portion of the CNN as a single half, with each side having separate weights [103], since the vast majority of strokes present stroke lateralization [104] and because in conjunction with lateralization information, allows the network to discern between halves with lesions and strokes without lesions. With only 190 cases, Cui *et al.* achieved accuracy of 0.85 and AUC scores of 0.86 ± 0.04 [103]. Their NN fared better than a simple conversion of **MedicalNet** to the classification task — a transfer learning tool designed to help implement medical segmentation using 3D models pre-trained on large dss. [105] , which only achieved 0.71 ± 0.05. This paper shows the importance of **proper weight initialization** to achieve good results, suggests that analysing **brains as two separate hemispheres** improves performance, the benefits of Inception architectural considerations and **proper regularization techniques** [103]. As of February 2021, Lo *et al.* published findings on a 2D CNN model trained with 1244 NCCT slices and using AlexNet pre-trained weights it achieved AlexNet **achieved an accuracy of 0.97, a sensitivity of 0.98, and a specificity of 0.96** on lesion detection, demonstrating once more the importance of proper weight initialization [106]. Ertl *et al.* in 2022 has shown the importance of using **adequate radiodensities windowing** while training medical CNNs, and they were to achieve predictive 0.89 accuracy and 0.93 F1-scores on the presence of brain hyperdensities in post-thrombectomy NCCT trained on 241 samples [107].

3.3 Segmentation on 3D Medical Imaging

Considering that 2D dss. and applications are older, more widespread and more modest in computing requirements, much of CV research from the past decade using DL is done on 2D CNNs.

Lesion segmentation is a relevant application, but AIS feature extraction is challenging due to the complex characterization of an ischaemic lesion and possibly small relative size. **DeepMedic** is a state-of-the-art tool to help in the training and creation of 3D CNN, especially designed for medical imaging segmentation [108]. This tool was successfully used to win several international neuroimaging competitions such as **"Ischemic Stroke Lesion Segmentation Challenge" (ISLES)** [109], or the **"Multimodal Brain Tumor Segmentation Challenge"** [108, 110]. However, these tools that simplify the creation of robust models, still rely on large quantities of high quality annotated data to train their dss..

Other projects solved this problem with **attention mechanisms** and **curriculum based systems**. **Attention mechanisms** save weight update information for each element of the sample, and portions with higher weights get further attention. This weighting is used in subsequent NN updates to focus more resources on regions of a sample — in this case, spatial attention selects data points in the image [111]. **Curriculum based learning** is a way to select subsets of the training data to be learnt by the NN, starting by training the network on easier to classify samples before moving to increasingly harder, usually done by analysing how samples' classification confidence in an initial training and using that score as the sorting function of samples for the full NN [112]. By creating an **attention guided curriculum learning pipeline** where the tumour is roughly segmented and creates the samples curriculum, then the segmentation results are passed to a second NN that dilates the segmented area, and finally the results are passed to another segmentation network that refines the dilated segmented area using the curriculum order, Zhou *et al.* improved Dice scores, reduce training times and the overall number of parameters on their innovative NN architecture, which they referred to as **One-Pass Multi-Task Networks with Guided Attention** [113]. By combining 2D, 2.5D and 3D samples, Milletari *et al.* achieved better segmentation results on a brain tumour segmentation challenge with Dice scores of 0.91 [114]. Sato *et al.* also used an efficient **anomaly detection** strategy relevant to this thesis. They used **autoencoders** to produce normal brains and detect infarction areas by comparing erased portions with their reconstruction. This method achieved a sensitivity of 0.68 and a specificity of 0.88 [78]. Another work by Olli *et al.* using CTA to segment stroke lesions achieved Dice scores of 0.61, and has shown improved performance when lesion lateralization information was included in the model [115].

3.4 Thesis framing

Models using both human decoded Imaging Biomarkers (IBs) and without them have been found to model AIS outcomes and have been referred in their own section. Using raw imaging data both for feature extraction and direct modelling is a more recent development. Only in August 2022, Tolhuisen *et al.* published on Diagnostics an AIS outcome prediction method using autoencoders on diffusion weighted imaging [116], similar to the method Sato *et al.* created, as described on a literature review [78]. It is the study that most closely relates to the CV work done on this thesis, but it is meant to be a post-treatment model, using considerably different topologies, and it is not trained on cheaper and most readily available NCCT scans. Although many of these papers used methods that can be replicated on personal computers with commercial GPUs, most state-of-the-art research referred previously was done in dedicated clusters.

Chapter 4

Material

4.1 Data description

Clinical, biomarker and imaging data provided for this thesis were collected under the **BioStroke Project** for ML purposes. It is composed by three main dss.: one with clinical and bureaucratic data, one with biomarkers data and one with neuroimaging data. The imaging database was provided by CHUP, being compiled by the thesis' author under the guidance of João Pedro Filipe and Ricardo Varela.

Clinical data included demographic, co-morbidities, prior medication, clinical evaluation at arrival — including NIH Stroke Scale and modified Rankin Scale —, and information on patient's admission. This ds. corresponds to a cohort of 274 patients with large vessel occlusion, from a total pool of 563 stroke code activations. After triaging relevant records, **152 cases were kept**, as shown in figure 4.1. The **biomarkers datasets** also refers to the same cohort which was selected as in the previous ds. and all entries were used the study's primary key. It contains **FBC data, clotting report, and biochemical analysis of biomarkers relevant to the thrombectomy**, all of them taken on patients' admission and after thrombectomy was performed. A custom OCR implementation based on `tabula` library was produce to help with manual data digitization, but the dataset was delivered for analysis with all records from the selected cohort due to time constraints. These dss. are described in table A.4, A.5 and A.6. There were **79 records on admission** and **126 cases on follow-up**.

The **neuroimaging dataset** was composed by radiographies in **DICOM format** from each patient of the study, stripped of all personally identifiable metadata. At admission, either to the first hospital, or at arrival in the reference hospital that performed

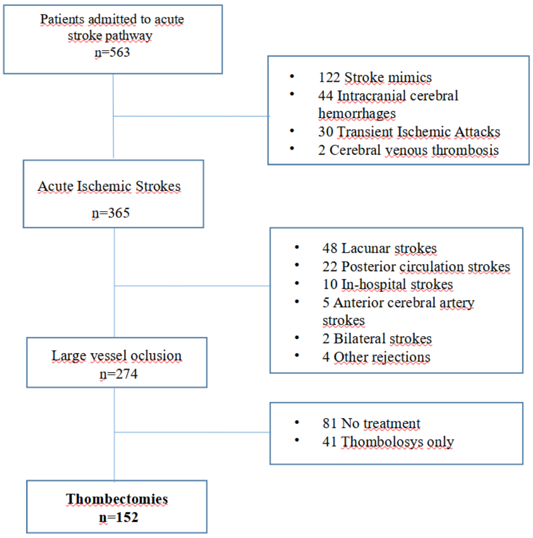


FIGURE 4.1: Cases-cohort selection flowchart from the existing BioStroke dataset.

the thrombectomy, NCCT scans of the patient, and CTAs were taken, and, if available, pCTs as well. Along with this imaging data, follow-up data was also retrieved by adding follow-up NCCTs taken after thrombectomy. Neuroimaging data was retrieved manually from the SECTRA front-end.

Unlike the clinical and biomarkers ds., for the imaging ds. only the relevant records were retrieved — for the 152 patients selected for thrombectomy —, which included **174 NCCT before thrombectomy**, and **141 NCCTs after thrombectomy**. CTAs and pCTs were preprocessed and QC'ed but not used during this thesis. Only pre-thrombectomy NCCTs were analysed during this thesis.

4.2 Computing environment

4.2.1 Hardware

For most programming tasks without a need for raw computing power, a laptop equipped with 16GB of system RAM, an AMD Ryzen 7 3550H CPU, and an NVIDIA GeForce GTX 1650 Mobile with 4096 MB GPU was used. Running in tandem in long experiments — especially for DL and CV tasks exploration and training tasks —, a laptop equipped with a 16GB of system RAM, Intel Core i7-10870H CPU, and an NVIDIA GeForce RTX 3070 GPU with 8192 GB of VRAM was used. Occasionally, while private laptops were busy on long operations, Google Colab cloud services on GPU mode was used to test function tweaks or test online code on synthetic or public dss..

4.2.2 R programming language

For most **data exploration, data preprocessing and data wrangling operations**, R language was used, since many packages are well-developed in this field, and their libraries enable fast data exploration and data manipulation without excessive code burden. R version in use was 4.0.3 [117], while the integrated development environment used was R Studio version 1.4.1717 [118]. Non-standard libraries required for the analysis mentioned or implied in this thesis are:

- DataExplorer [119]
- GGally [120]
- ggplot2 [121]
- ggpubr [122]
- ggthemes [123]
- gtsummary [124]
- lubridate [125]
- MASS [126]
- NAGuideR [127]
- readr [128]
- readxl [129]
- rpart [130]
- rpart.plot [131]
- rstatix [132]
- tidyr [133]
- tidyverse [134]

4.2.3 Python programming language

The main **modelling tasks were conducted in Python**, because most state-of-the-art and up-to-date libraries **for ML, deep learning (DL) and computer vision (CV)** had no adequate alternatives on R. Python ML state-of-the-art libraries are coded on for this language, such as Sci-Kit Learn [55], Keras [79], PyTorch, and all derivative libraries that use them as a basis, such as Project MONAI [135], AutoSkLearn [72], AutoPyTorch [73] and AutoGluon [74], all of them greatly expanding modelling possibilities and accelerating the workflow significantly. Depending on the required libraries for a specific modelling task or program to be used, versions of Python [136] ranging from 3.6.0 to 3.8.5 were used, mostly interfaced through **Jupyter notebooks** [137]. Several libraries and different Python version in use for various tasks are incompatible, so management of several environments was done through Anaconda version 2.4.0 [138]. Most coding is operating system agnostic, but, at the time of writing, some notebooks include packages only available for Linux, most notably hyper-parameter tuning libraries, AutoML libraries and some libraries that enable GPU optimized operations. The list required to produce all the Python code created for this thesis is:

- autorank [139]
- autosklearn [72]
- cuda [140]
- imblearn [141]
- joblib [142]
- keras [79]
- keras-tuner [143]
- LightGBM [60]
- matplotlib [144]
- nibabel [145]
- numpy [146]
- OpenCV [147]
- pandas [148]
- RAPIDS [149]
- scikit-posthocs [150]
- scipy [151]
- seaborn [152]
- sklearn [70]
- talos [153]
- tensorflow [154]
- tensorboard [154]
- volumentations [155]
- xgboost [59]

4.2.4 Other requirements

Linux is required for neuroimaging preprocessing and for some notebooks that explored libraries exclusively on Linux, such as autosklearn [72], or **RAPIDS** [149] — a library that leverages GPU resources for computing intensive operations —, and **FSLTools**, a versatile neuroimaging command line processing tool [156].

Chapter 5

Methods

5.1 Data preprocessing

5.1.1 Clinical Dataset

The ds. contained 274 cases. **152 were selected by only including patients that received treatment** — which excluded 81 patients — **and that treatment included thrombectomy** — which filtered a further 41, as seen in figure 4.1. Fields with high proportion of missing values (NAs), derived features, or no variability were removed. Some fields with non-validated clinically data were also removed. NIHSS after treatment and at discharge as after thrombectomy variables were also removed. Remaining **NAs have to be imputed or discarded** by case removal before modelling, so they do not crash some algorithms, nor are sub-optimally imputed by them. Missing times related to onset were imputed by calculating the median time difference between recorded times on arrival to first hospital and reported onset times. All field with separate time and date information were combined into a single field containing date and time information, removing formerly separate fields. After this triage, all fields with no variability and binary fields with less than three cases in the minority class were removed, since they do not have enough samples to generalize well, and would likely increase models' overfitting. Two NAs in other medication were imputed by creating a DT using only previous co-morbidities and the patient's medication data. After this, the ds. contained no NAs for the selected of 152 cases. All remaining intermediate derived fields were removed. Fields that could bias multiple comparison corrections were removed from statistical analysis. The final ds. contains a slight class imbalance with 87 cases classified as bad outcomes, 57.2%, and 52 cases classified as good outcomes, 41.8%.

5.1.2 Biomarkers Dataset

Biomarkers ds. was retrieved from BioStroke's patients blood work and biochemical analysis conducted upon admission and patient's follow-up. On this ds., part of the preprocessing was done directly on the provided spreadsheet, given **this database was still in work-in-progress state**. After clean up and standardization, the database was saved as new spreadsheet that was used in a formal pre-processing pipeline, done in R. In the biomarkers ds., **only biomarkers data is considered relevant**, so hospital of arrival information was removed as well as dates and times, except FBC and biochemistry timestamps. Finally, mRS outcome was added from the clinical ds. by cross-referencing data between datasets. The high number of NAs makes this ds. challenging for modelling, given imputations on NAs can heavily distort the data — e.g., by replacing NAs by zeros when random values are missing at random. However, with such a great number of NAs, more sophisticated imputation techniques also fail, since they do not have enough data to create good enough models for imputation. To solve this issue, compromises had to be accepted, but several efficient imputation techniques were tested to keep as much data as possible, and keep imputation bias minimal.

First, **FBC quantification** often lacked either the percent value of a blood cell type or its raw quantification, but leucocytes quantification was usually present if the patient had data available. First, using the formula:

$$NLL\_ratio = \frac{[Neutrophils] + [Lymphocytes]}{[Leucocytes]}\ , \tag{5.1}$$

leucocyte data was QC'ed for each FBC set — at admission and at follow-up — and the global average, $\overline{NLL\_ratio}$, was calculated. With this indicator, leucocytes, neutrophils and lymphocytes were QC'ed and absurd or missing values with full component information were corrected with their sum. The remaining erroneous or missing leucocytes' quantification usually had neutrophil and leucocyte quantifications, important for calculating an important clinical biomarker, the **Neutrophil to Leucocyte Ratio (NLR)**. Considering there are other leucocyte types in the blood — e.g., Monocytes, Eosinophils, and Basophils —, $[Leucocytes]$ was imputed using:

$$Leucocytes = \frac{[Neutrophils] + [Lymphocytes]}{\overline{NLL\_ratio}} \tag{5.2}$$

Then all NAs in the remaining component concentrations were imputed as:

$$[Leucocyte\_type] = \frac{TLC \times LTP}{100} \,, \tag{5.3}$$

where TLC is total leucocytes count, LTP is the specific leucocyte type percentage. This NAs calculation was followed by missing LTP imputations with

$$Leucocyte\_type\% = \frac{[LTC]}{TLC \times LTP} \times 100 \,, \tag{5.4}$$

where LTC is each leucocyte type count.

All Neutrophil-Lymphocyte Ratios (NLRs) were then recalculated, correcting or imputing any missing record. The modelling d.v., mRS at discharge, was binarized with the outcome criteria as $mRS_{at\_discharge} \leq 2$ set to 0. FBC and biochemistry timestamps are paired with no difference between both fields, so they were imputed using the value filled in. When that was not possible, the average difference between FBC timestamps at arrival and follow-up, 28.9 hours, was used to make adequate imputations.

The ds. was **split between admission and follow-up data** to improve missing data profiles. B-type Natriuretic Peptide PROmotor Hormone was removed from Biom0h and Biom24h, as well as Homocysteine from the Biom24h, since they still had nearly 90% NAs. At this point, rows with no biomarkers' data were removed leaving **Biom0h with 79 samples** — of which 22 complete records — and **Biom24h with 126** — of which 64 complete records. NAs profiles can be seen in figures A.5 and A.6.

At this point, and before adding imputations via modelling — which can add distortions —, descriptive and statistical analysis was done on both dss.. Data analysis is shown in figures 6.8 and 6.11. To salvage most records with valid information for modelling and to keep most of the ds., **NAGuideR** [127] was used to apply and test several strategies to impute all remaining data in the dss.. NAGuideR was configured to not make data transformations, discard fields nor samples based on their percentage of NAs, nor their coefficient of variation, to **keep as much original data as possible**. The following imputations were done: zeros (`zero`), overall minimum value (`min`), column median (`rowmedian` [157]), row median (`rowmedian` [157]), and deterministic minimal value (`mindet` [158]); global structured methods such as: singular value decomposition (`svdmethod` [159]), and Bayesian principal component analysis (`bpca` [160]); and with the following local similarity approaches: k-nearest neighbour (`knnmethod` [159]), quantile regression (qrilc [158]), iterative robust model (`irm` [161]), and random forest model (`MissForest`). Imputed dss. were evaluated with the available classic criteria only — Normalized Root Mean Squared Error (NRMSE), NRMSE-based sum of ranks (SOR), Procrustes Sum of Squared errors (PSS), and Average Correlation Coefficient between Original value and Imputed value (ACC_OI). **MissForest** (a.k.a., RF on NAs, as a Random Forests-based method [162]) was shown to be the **best for all metrics but ACC_OI**, in which it ranked as the second-best imputation strategy. Imputations with variable media (`rowmin`) were the second-best set of imputations in all criteria, and all metrics ranked worse than MissForest imputation. Both these four complete dss. — via imputations — were used to model the biomarkers ds..

5.1.3 Neuroimaging Dataset

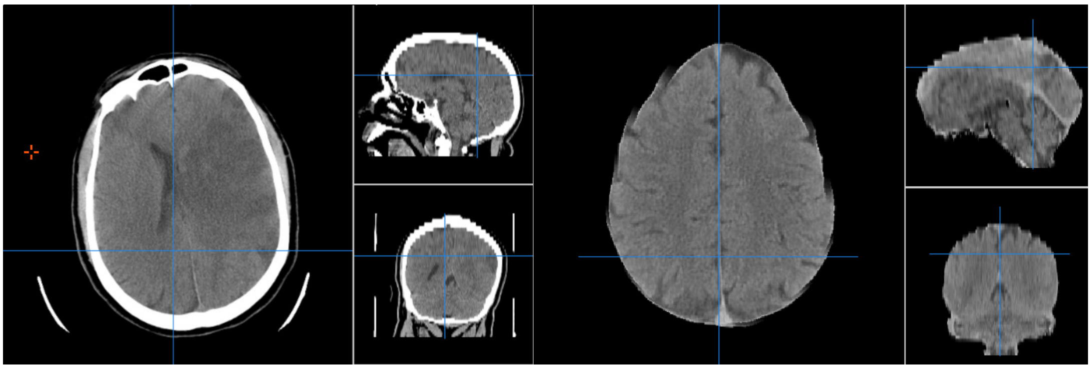


FIGURE 5.1: Orthogonal projections of an NCCT without registration as viewed on Mango [30], on the left; and the same CT scan after the registration and deskulling process, on the right.

Before further preprocessing, all CT scans in DICOM format were reviewed, confirming the absence of personally identifiable metadata, and remove CT scans that fell out of the scope of work, have low quality, or were repetitions of the exam at the same timestamp. Files were then **converted from DICOM format to NifTI format**, to reduce image size, make file structure human-readable and to make individual CT scans easier to interact with other available tools. NIFTIs were named with the BioStroke patient key and metadata regarding time of capture, as well as CT type description — i.e., NCCT, CTA or pCT —, if available in the metadata. As recommended [38], **files were defaced, deskulled and registered** using a custom pipeline based on **FMRIB Software Library** — and neuroimaging library specialized in MRI analysis, but that can be configured to work with CT scans [156] — as exemplified in figure 5.1. Before deskulling, the **radiodensities dynamic range was clipped** using an extended brain window. Registration was done using one

BioStroke sample with no interferences, full brain capture, good positioning and high resolution NCCT. That sample was further cropped and realigned to have perfect alignment with natural axis. Registration process normalized position, orientation, rotation and size in all three axis. The model for registration used elastic local adjustments, so that each brain region's shape and size variation is minimal, but the process did not show perfect results in this regard. Finally, after registering all files were verified one by one to confirm the registration process did not cause unreasonable distortions to the original CT, and after assuring the CT type is the one in the description, they were stored as three separate dss.: NCCT ds., CTA ds., and pCT ds..

5.2 Statistical Analysis

On all tabular dss., descriptive and statistical analysis was conducted to better illustrate data, assess the cohort limitations, and range of application of the proposed models. This analysis was automated with the help of `gtsummary` [124]. Each feature data types were defined manually since `gtsummary` selects tests based on continuous, categorical, and logical variables, as well as the number of independent classes to consider — in our case two outcomes. **Parametric or non-parametric tests are selected automatically**, depending on the feature distribution characterization. For **logical and categorical features**, the percentage of true on each class is shown, and the test performed is the **Pearson's Chi-squared test**; for **categorical variables where the Chi-squared test individual predictions are below 5, Fisher's exact test with Freeman-Halton extension was used**; for **continuous variables**, features were described by median and Inter Quartile Range (IQR) — given that no variable tested positive on the Shapiro-Wilk's test for normal distributions —, and a **Wilcoxon rank sum test**, better known as Mann-Whitney's test, was performed [163]. The number of tests conducted on each ds. creates the multiple comparison problem, where it is likely that some Statistically Significant (stat. sig.) tests at a significance level of 5% are significant only due to chance [164], so, **adjusted p-values were calculated using Holm-Bonferroni correction**, as a uniformly more powerful alternative to the more common Bonferroni *post hoc* test [165]. The *post hoc* test was applied only to groups of variables inside the same ds. and time, that is, a group with all the clinical variables at arrival and the imaging biomarkers, another group with biomarkers analysis ds. at arrival; and the last group with biomarkers ds. on follow-up.

5.3 Clinical Data Modelling

5.3.1 Data preparation

After preprocessing clinical data and preliminary data analysis through an R pipeline, the triaged and processed data was loaded into a Python notebook. Preliminarily, data to be fed to models was QC'ed by verifying NAs. After confirming the ds. was complete, a principal component analysis (PCA) was conducted to evaluate the best FS strategy. For each set of chosen clinical data features, data was loaded into a Python framework, classifying each feature as: category, boolean integer, or float point fields. Numeric variables did not show relevant outliers, so all cases were kept, as seen in tables 6.1, 6.2 and 6.3. For feature transformation purposes, age, mRS before event, NIH Stroke Scale, time difference in minutes from onset to the first hospital, to the first CT and to the second hospital were considered **numerical variables and were rescaled using a `MinMaxScaler` algorithm**. Boolean variables were binarized and categorical variables were converted with dummy encoded by parametrizing `OneHotEncoding`. Despite irrelevant variables and collinearity issues with the data, most variables were added to the modelling in a first stage, since some models handle better than others these issues.

5.3.2 Testing

The ds. was split in training, validation and test sets. Using the definition provided by Brian Ripley in 'Pattern Recognition for Machine Learning'.

> – Training set: A set of examples used for learning, that is to fit the parameters of the classifier.
>
> – Validation set: A set of examples used to tune the parameters of a classifier, for example to choose the number of hidden units in a neural network.
>
> – Test set: A set of examples used only to assess the performance of a fully-specified classifier. [166]

From the initial ds. 20% samples were randomly selected for the test set. For model selection, **10-fold stratified repeated cross-validation was used**. Non-nested cross-validation was used, so the model metrics shown for cross-validation are the ones obtained while looking for the best model. Given the model selection is optimizing for a metric, the metrics obtained this way are optimistic. One solution would be to use nested cross-validation or accept the performance on a separate ds.. Given that on average the difference between nested and non-nested cross-validation is below 2 percent [167], and search and evaluation time increases exponentially, nested cross-validation was not used. The metric selected for search was weighted F1-score, while the intended testing and comparison metric is the AUC score. Finally, the parametrized models were evaluated with **ROC curves on the test** set to assess their general performance is within error confidence intervals obtained in cross-validated validation scores.

5.3.3 Model selection

An array of classifiers strategies of interest was chosen, including **linear based methods** such as LR, Linear Discriminant Analysis (LDA), and linear SVM; **decision-trees based methods** such as single DTs, Random Forests (RF), AdaBoost, XGBoost (XGBM) and Light Gradient Boosting Machine (LightGBM) classifiers; **Bayesian methods** represented by the GNBC — for testing purposes only, given several variables do not follow the normal distribution —; **lazy learning methods** represented by k-Nearest Neighbours (k-NN) strategy; **SVMs** with different kernel transformations; **Quadratic Discriminant Analysis (QDA)**; and **NNs** represented with the fully connected networks, the Multi-Layer Perceptron (MLP) available on Sci-Kit Learn. Before training, a global random generator seed was set and for each model, the hp. was also internally added.

Several **manually tailored Grid Searches (GSes)** were conducted guided by previously found hps., where each time a range limit value was selected, the next Grid Search (GS) would be adjusted so the extreme value would become a median value in the next GS, following Bayesian search principles. After finding the main models for each modelling strategy, modelling strategies were compared by the average metrics calculated on cross-validated results.

5.3.4 Performance metrics and statistical model comparison

During model selection phase, weighted-F1 score was chosen for model selection. The positive class was selected as the bad outcome — the ones rejected for thrombectomy. Accuracy, balanced accuracy, and Area Under the (ROC) Curve (AUC) where also calculated and checked to assure the model performed consistently in all relevant metrics. All metrics were calculated and stored for each cross-validation set. A **classification report**, which calculates precision, recall and F1-score, macro- and weighted-averaging separated by predicted class was also performed.

Statistical model comparison was done by performing a **Friedman's test** on the list of AUC cross-validated scores from the best model of each modelling strategy. **AUC score was the main model evaluation metric.** It is calculated from the Receiver-Operator Curves (ROC) and has the advantage of summarizing those curves under a single metric [54]. When this test rejected the null hypothesis, **multiple models' comparison was done with a Nemenyi *post hoc* test** [168], conducted on the same AUC scores matrix. Models that distance themselves less than the critical distance were considered not stat. sig. different and were grouped together. The group of models that performed the best was used in further analysis, after the initial survey on the base clinical ds..

5.3.5 Automated Machine Learning (AutoML)

The clinical ds. was passed to **AutoSkLearn**, testing both version's strategies: **the base version** — using meta-learning to warm-start the Bayesian optimization, followed by ensemble creation —, and version 2.0 strategy, which expands the first version with Portfolio of Successive Halvings — a way to select test groups of models with increasing resources, such as number of samples, iterators, etc. —, early stopping unpromising tests, and by automatically selecting the search policy based on the information learned from the ds.. The models were run for the same time as the total time of the last manually constructed GS and **compared against all base models** preceding it.

5.3.6 Feature Selection (FS)

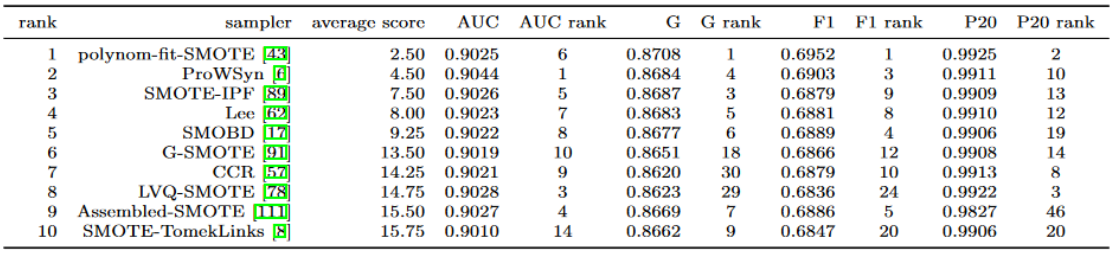
A study on each tabular ds. was conducted after the best models were found. Considering the top two or three hyperparametrized models were selected. For each of them the **Hughes Phenomenon (HP)** [169] was studied, **recurring the model's training with increasing number of features**, selected by their **ANOVA F-values** between label/feature for classification tasks. The previous pipelines were rerun for every ds., trimming the features by the value recommended by FS analysis, and when metrics improved, these new models were considered final.

5.4 Tabular Data Augmentation

This work involves the processing of three different dss., two tabular and one 3D image-based, so different methods were applied. For tabular data, synthetic data can be produced with a special interpolation method, usually applied to dss. with high class imbalance, the Synthetic Minority Over-sampling Technique, better known as SMOTE [170]. Several SMOTE variants, which improve on special types of dss., and try to better generalize SMOTE fields of application were test [171].

TABLE 5.1: The top performer oversamplers ranked by the combination of all scores. Besides the combined ranking, the aggregated values of the measures and the corresponding ranks are also reported. Table retrieved from Kovacs *et al.* 2019 [171].

.



rank	sampler	average score	AUC	AUC rank	G	G rank	F1	F1 rank	P20	P20 rank
1	polynom-fit-SMOTE [43]	2.50	0.9025	6	0.8708	1	0.6952	1	0.9925	2
2	ProWSyn [6]	4.50	0.9044	1	0.8684	4	0.6903	3	0.9911	10
3	SMOTE-IPF [89]	7.50	0.9026	5	0.8687	3	0.6879	9	0.9909	13
4	Lee [62]	8.00	0.9023	7	0.8683	5	0.6881	8	0.9910	12
5	SMOBD [17]	9.25	0.9022	8	0.8677	6	0.6889	4	0.9906	19
6	G-SMOTE [91]	13.50	0.9019	10	0.8651	18	0.6866	12	0.9908	14
7	CCR [57]	14.25	0.9021	9	0.8620	30	0.6879	10	0.9913	8
8	LVQ-SMOTE [78]	14.75	0.9028	3	0.8623	29	0.6836	24	0.9922	3
9	Assembled-SMOTE [111]	15.50	0.9027	4	0.8669	7	0.6886	5	0.9827	46
10	SMOTE-TomekLinks [8]	15.75	0.9010	14	0.8662	9	0.6847	20	0.9906	20

In a first step, a **comprehensive review on SMOTE variants** was made [141, 172, 173]. This review has shown the most reliable method for minority oversampling and data augmentation is polynomial-fit-SMOTE [174], as shown in table 5.1 [171]. `polynomial-fit-SMOTE` is incompatible with existing `conda` environments, so, a study using **SMOTE variants and undersampling methods available in the** `imbalanced-learn` **library was done**. For standardization, the most suitable candidate from a combination of oversampling and undersampling methods was found by **creating pipelines with various oversampling and undersampling techniques in various sorting orders** and calculating the mean True Positive Rate (TPR) and mean False Positive Rate (FPR) with which the AUC score was calculated, the best performing combination was tested by just balancing the dss. and by oversampling the ds. up to 200 samples per class. For every relevant tabular ds. combination, SMOTE variants data balancing techniques and augmentation were analysed.

5.5 Biomarkers Data Modelling

Part of the search for successful strategies was narrowed down, to save computing time. After collecting both relevant dss. and their statistical analysis, the dss. were passed to the modelling pipeline, similar to the one used for clinical data, but without oversampling or undersampling and including only a search for part of the modelling strategies that perform the best within the first group. Over- and undersampling techniques, as well as augmentation were not used since they did not show improved results over the best models. In these cases, **only the best performing modelling strategies on the clinical ds.** — i.e., logistic regression; boosting represented by Extreme Gradient Boosting Machine (XGBM) and the faster LightGBM; and SVMs with various kernel types — **were applied**. RF and AdaBoost were not used since they have a boosting strategy sufficiently similar to XGBM with equivalent results, but having much longer runtimes than the nearly equivalent **LGBM**, which allows modelling time to be used in more experiments. To find the best models, the **same methodology used in the clinical ds. was used on four biomarkers dss.:** biomarkers at admission dataset (Biom0h) with Random Forest (RF) imputations, Biom0h with features median imputations, biomarkers at admission dataset (Biom24h) with RF imputations, Biom24h with features median imputations. After the GS on each ds. the selected models were compared with the top three clinical models, referred to as 'base models'.

5.6 Imaging Data Modelling

The imaging data was treated with CV methods for IB extraction and convolutional neural networks (CNNs).

5.6.1 Hemispheric Contrast (HC)

In a first phase, an **imaging contrast biomarker was created**, as the contrast between two hemispheres. This contrast was calculated on a registered deskulled brain, where the **registration template is perfectly aligned, centred and symmetrical in the volume space**. Considering the radiointensities input tensor, I, and the normalized radiodensities image input tensor, $I^{'}$, **Hemispheric Contrast (HC)** is calculated as:

$$f(I^{'}) = 2 \times \frac{\Sigma_{i=1}^{\frac{x}{2}} \Sigma_{j=1}^{y} \Sigma_{k=1}^{z} \left(I^{'}_{i,j,k} \right) - \Sigma_{m=\frac{x}{2}+1}^{x} \Sigma_{l=1}^{y} \Sigma_{n=1}^{z} \left(I^{'}_{l,m,n} \right)}{x \times y \times z}, \tag{5.5}$$

where i, j, k, l, m and n are imaging volume coordinates, x, y and z are imaging volume dimensions of $I^{'}$. **Normalized intensities tensor,** $I^{'}$, is calculated as:

$$I^{'} = \frac{I - I_{max}}{I_{max} - I_{min}} \tag{5.6}$$

HC and its absolute value was calculated for all samples. These two features were added to the processed clinical ds., statistically analysed, and passed to the formerly described modelling pipeline and compared with the best base models.

5.6.2 Convolutional neural networks (CNNs)

Modelling started with a **Keras example of a 3D CNN for classification**. The architecture initially explored had four 3D CLs, each of them followed by a pooling layer — `MaxPooling3D`, which returns a single maximum value in each $2 \times 2 \times 2$ grid —, followed by normalization of batch values — `BatchNormalization` —. After the previously described sets, the layers are again reduced and flattened with `GlobalAveragePooling3D` and passed to a last fully connected dense layer — i.e., `Dense`. The general initial architecture can be seen in figure 5.2. All layers had the standard **Recursive Linear Unit (ReLU) as AF**.

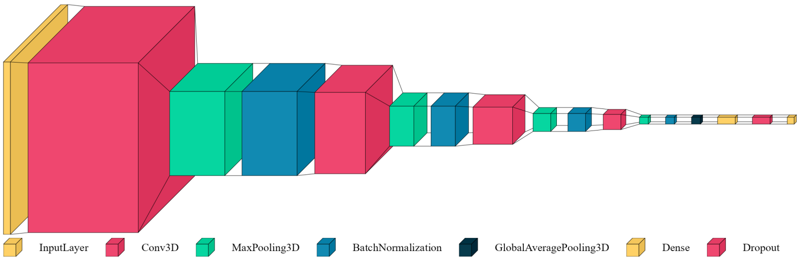


FIGURE 5.2: Diagram representing the initial 3D CNN architecture.

The model used a **modest initial learning rate (ILR),** $lr = 0.0001$, and a slow exponential decay of weights over 10.000 steps as a regularization method, to avoid overfitting. This was changed later since the model has shown no learning in the first untweaked experiments. The model uses **early stopping with patience**, using 15 epochs to allow the network to surpass local minima and a larger margin for final experiments. Patience concept in NN design means the NN continues training for n more epochs after the stopping condition is met. If in those epochs, validation metric improves again, the patience counter is reset and the network continues for more n epochs, counting from the new best. Accuracy was used as the validation metric. Models were saved only when validation metrics improve, using Keras checkpoints functions.

In **3D CNN first versions**, the data was fed to **training without rotations, then with a basic axial axis rotation augmentation**, and, finally, several other augmentations were introduced with the package `volumentations` library. Imaging data augmentations will be further discussed on their own section.

After initial training, and augmentation introduction, improvements were attempted using Neural Architecture Search (NAS) via the more simple `talos` — limited to several advanced types of random search algorithms —, and the more versatile `keras-tuner` — with advanced search space algorithms such as Bayesian Optimization and Hyperband —, since more efficient NAS are needed for longer training times that 3D CNNs incur.

The **search space** included number of CLs sets — i.e., CL, associated pooling, and associated normalization —, filter and kernel size of each CL, AF used in each CL — from a selection of ReLU, Exponential Linear Unit (ELU), Scaled ELU (SELU), Hyperbolic Tangent (TanH), and sigmoid function —, existence of pooling layers and their type — pooling by average or maximum —, dense layer size, and dropout existence and proportion. This model used exponential weight decay, so initial learning decay, number of decay steps, and decay rate were hps. also tested.

To run all models with just 8GB of VRAM batch sizes were adjusted, and **CUDA unified memory** was used. CUDA unified memory implies longer training times when VRAM is exceeded but allows models to run with hardware that would be unable to do so. Despite these optimizations some experiments ran shorter than expected, but a **system of checkpoints saving all models tested by the search algorithm**, allows continuing the search, or salvaging the best models.

5.6.3 Training and Model Evaluation

Validation performance metrics on CNN models were substantially different from the ones chosen for tabular data for operational reasons. First, CNNs can continuously improve their adjustment to training data, so, the training of CNNs continues until a stop condition — usually based on validation metrics — stops improving. This is similar to creating a model at each epoch, with former model weights. The consequence is the model trained and chosen cannot include validation data, as the ML methods used for tabular data, which refit the models to include both training and validation data on the model after model selection was completed. The base models and DL library chosen was TensorFlow with Keras high level code, which does not have encoded libraries to use F1-weighted scores during training, so **validation metrics search the epoch with the highest weighted accuracy**. The models' fitting function, **loss**, was defined as binary cross-entropy. The base 3D CNN **optimizer** was maintained, **Adam**. **Batch size** was set to two to comply with memory constraints. Base **ILRs** were tested depending on the model's configuration.

When **relevant models** were found, generalization capacity needs to be **evaluated on previously unseen data**. Considering data scarcity and computing time required to develop CNNs, several development versions were done tracking validation results before finding the **final architectures to be tested**. The imaging ds. was divided in the typical training, validation and test before training. After fitting the model and evaluating on the test set, **stratified cross-validation on the training and validation set was performed**, assessing **accuracy, balanced accuracy, weighted F1 and AUC scores**, so an apples-to-apples comparison could be done. ROCs and classification reports — which include True Positives (TP), True Negatives (TN), False Positives (FP), False Negatives (FN), precision and recall — were obtained on the test set.

5.6.4 Neural Network Architecture Search (NAS)

To try to improve on base architecture metrics, a better performing one for this ds. was searched. First, since initial implementations used more strict early stop parameters, $patience = 15$, to account for low ILRs this value was increased to 30 epochs. The base template converted inputs to squared tensor by rotating by 90 degrees and cropping, which cut the volumes in their frontal and occipital parts, destroying potentially useful information. This behaviour was changed, removing the rotation, and adding adequate padding before resizing to the intended dimensions. The cropped dimensions were changed to $width = 130$, $height = 130$ and $depth = 52$, so that after processing by convolution tensors would be multiples of 128, therefore benefiting from TPU acceleration. The exhaustive former search procedure, GS, was intractable for this NN. `keras-tuner` **was selected for NAS** By running first Hyperband search, initial test performance can be tested on hundreds of hp. combinations, defaulting each parameter on the base CNN model. The best between the base model and the model found by HyperBand can then be used as the baseline for Bayesian Optimization, a slower and more dependent on initialization search heuristic. The search space of those Bayesian Optimization searches is described in the following code snippet:

```
# configure model search space
ilr           = hp.Float(’learning_rate’,
            min_value = 1e-5, max_value = 1e-1,
            sampling  = ’log’, default   = 1e-3)
decay_steps   = hp.Int(’decay_steps’,
            min_value=1e4, max_value=1e6,
            sampling=’log’, default = 1e5)
decay_rate    = hp.Float(’decay_rate’,
            min_value=0.93, max_value=0.99,
            step=0.1, default = 0.96)
lr_schedule = keras.optimizers.schedules.ExponentialDecay(
            initial_learning_rate = ilr,
            decay_steps           = decay_steps,
            decay_rate            = decay_rate,
            staircase             = True
            )

nr_conv_layers= hp.Int(’number_of_layers’,
            min_value = 1, max_value = 5, default = 4)
filter_s_1    = hp.Choice(’filter_s_1’, [64, 128, 256], default = 128)
kernel_s      = hp.Choice(’kernel_s’, [3, 5], default = 3)
activation_t  = hp.Choice(’activation’,
            [’relu’, ’elu’, ’selu’, ’tanh’, ’sigmoid’],
            default = ’selu’)
type_pooling  = hp.Choice(’type_pooling’, [’max’, ’avg’], default = ’avg’)
maxp_1        = hp.Choice(’maxp_1’, [1, 2], default = 2)

with hp.conditional_scope("number_of_layers", [2, 3, 4, 5]):
      filter_s_2    = hp.Choice(’filter_s_2’, [64, 128, 256], default = 128)
      maxp_2        = hp.Choice(’maxp_2’, [1, 2], default = 2)

with hp.conditional_scope("number_of_layers", [3, 4, 5]):
      filter_s_3    = hp.Choice(’filter_s_3’, [128, 256, 512], default = 256)
      maxp_3        = hp.Choice(’maxp_3’, [1, 2], default = 2)

with hp.conditional_scope("number_of_layers", [4, 5]):
      filter_s_4    = hp.Choice(’filter_s_4’, [256, 512, 1024], default = 512)
      maxp_4        = hp.Choice(’maxp_4’, [1, 2], default = 2)

dense_u_5     = hp.Choice(’dense_u_5’, [256, 512, 1024], default = 512)

dropout       = hp.Float(’dropout’,
      min_value = 0.0, max_value = 0.5,
      step = 0.1, default = 0.1)
```

LISTING 5.1: Python code snippet describing keras-tuner search space parameters

5.6.5 Imaging Data Augmentation

Augmentations were also used to improve CNN performance. In this case, the data augmentations are related to image transformations, in this case, adapted to 3D voxel based spaces. In CV tasks, synthesizing images from the ds. bearing the same meaning but different characteristics, such as contrasts, colours, axial flipping, rotations, minor deformations, and added noise allows deep neural networks to learn the various forms a single definition — i.e., sample — can take, and avoids overfitting by improving the definition of the function that describes the class to be classified [175].

Five transformations were used in this project, namely: **axial view rotations, alterations to global brightness, and flipping in all planes**. Probabilities of application were assigned to each augmentation, so there was a chance the original data was fed to the NN, and so that transformation do not affected all the images. Contrast changes, scaling, and noise addition were removed due to memory leaks in all available volumentations library implementations tested, only perceivable in longer training sessions. Elastic deformations were not implemented due to their excessive processing time. All these transformations were select based on their usefulness, computational resources required and implementation difficulty, being applied randomly to each training example, with adjusted probabilities of occurrence and diverse parameters, to avoid distorting each non-synthetic training sample excessively. These augmentation sets are commonly used in biomedical data, and many libraries specialized in this field implement them, such as project MONAI [135], `volumentions`, and `volumentations-3d` [155, 176], as well in several well-known data science competitions [109, 177].

5.6.6 Transfer learning

In a first part, the best architecture found in earlier sections was **trained on the original template model ds., MosMedData**, a publicly available ds. of chest CT scans with COVID-19 related findings. After training the chosen CNN on this data, model's weights were saved. **On a second session, the model was fine-tuned** to the task of classifying thrombectomy outcomes by running another training session on the weights resulting from the model's training in lung NCCTs. **On a third session**, model weights from MosMedTraing were loaded into the same architecture, but **the last part of the network was removed: a pooling layer, a fully connected dense layer, a dropout layer and the output layer**. Then they were readded as non-initialized trainable layers. Training was done on these layers, using the inner model in inference mode. Then, the **entire model was unfrozen and retrained with a ILR one order of magnitude lower** than the one selected previously. Finally, it was evaluated and compared by the **cross-validation** procedure used previously.

5.7 Mixed Models Modelling

Mixed models are models that combine more than one of the former dss. — clinical, biomarkers and imaging data. **The first mixed models created combined the clinical ds. with the biomarkers ds.**. To minimize underfitting, FS based **multivariate LR analysis was done, considering as threshold the maximum p-value that includes sex and age** from the clinical ds., as seen in table A.25. Using the same FS threshold the analysis was repeated for the biomarkers ds. at admission — table A.26 —, and at follow-up — table A.27. **All dss. were subset using the same threshold and two new dss. were created**, by running an inner merge with the clinical subset of features against each subset of biomarker features with RF imputations. The ds. with admission features was called `ClinBiom0h`, and the one with follow-up features was called `ClinBiom24h`. These new dss. were them passed to the same modelling pipeline used to model the original biomarkers dss., searching the best hp. for each modelling strategy and compared to base models. The best mixed models were also trained on augmented data to try further improvements. **Later mixed models combine IB information with the previous dss.**, but only for the best models selected on each phase. This was achieved by merging HC and absolute HC with the dss. required by each model, through an inner merge.

Chapter 6

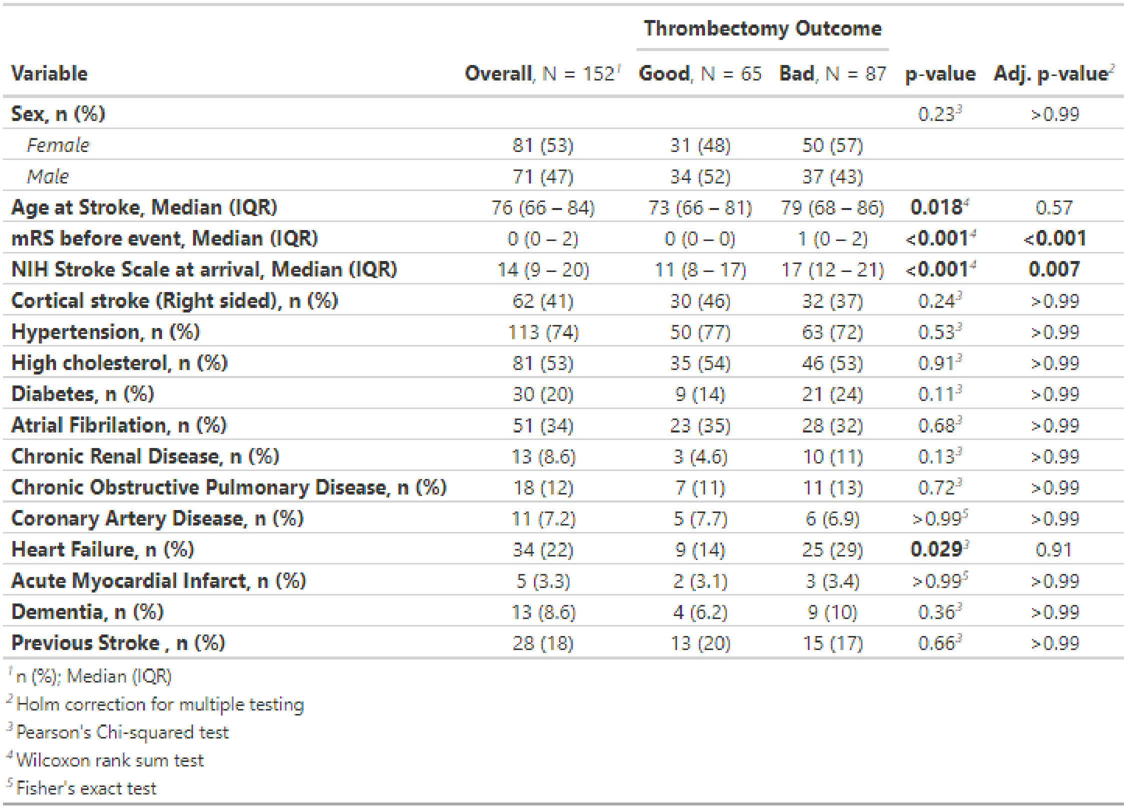
Results

In this section, to simplify **dss. and models description they are often referred by abbreviations** related to their characteristics. **Clin** refers to dss. and models using the clinical ds.. **Biom** refers to the FBC and biochemical biomarkers ds.. **CA** refers to the use of hemispheric contrast (HC), an imaging biomarker explained in its own section. **0h** refers to at admission data and **24h** refers to follow-up data. **FS** is used each time strict feature selection is used. **AUG** refers to models with augmentations. To exemplify, a model referred as **ClinBiom24h AUG FS** refers to a model created with clinical and biomarkers data at follow-up using data augmentation and that has strict model selection in its pre-processing pipeline.

6.1 Clinical Data

The sociodemographic profile of the cohort, is shown in table 6.1 is focused on senior population with a median age of 76 y.o. without statistically significant (stat. sig.) sex imbalance. Age, suggests being a more relevant variable, where good outcomes had lower median aged subjects, but again, p-values after adjustment to multiple comparison are not significant, $p.adj. = 0.57$. mRS before the event is stat. sig. to better outcomes, $p.adj. < 0.001$. **Good outcomes had lower mRS before the event than bad outcomes**, mostly composed by patients with individuals with no previous impairment as evaluated by mRS, $IQR_{mRSprv} = [0 - 0]$. **Lower level of impairment after AIS, as evaluated by the NIHSS score at arrival to hospital also had statistically better outcomes**, $p.adj. = 0.007$. Although, 89.5% of the patients admitted had at least one previous evaluated condition,

TABLE 6.1: Demographic and medical history data descriptive and statistical summary.



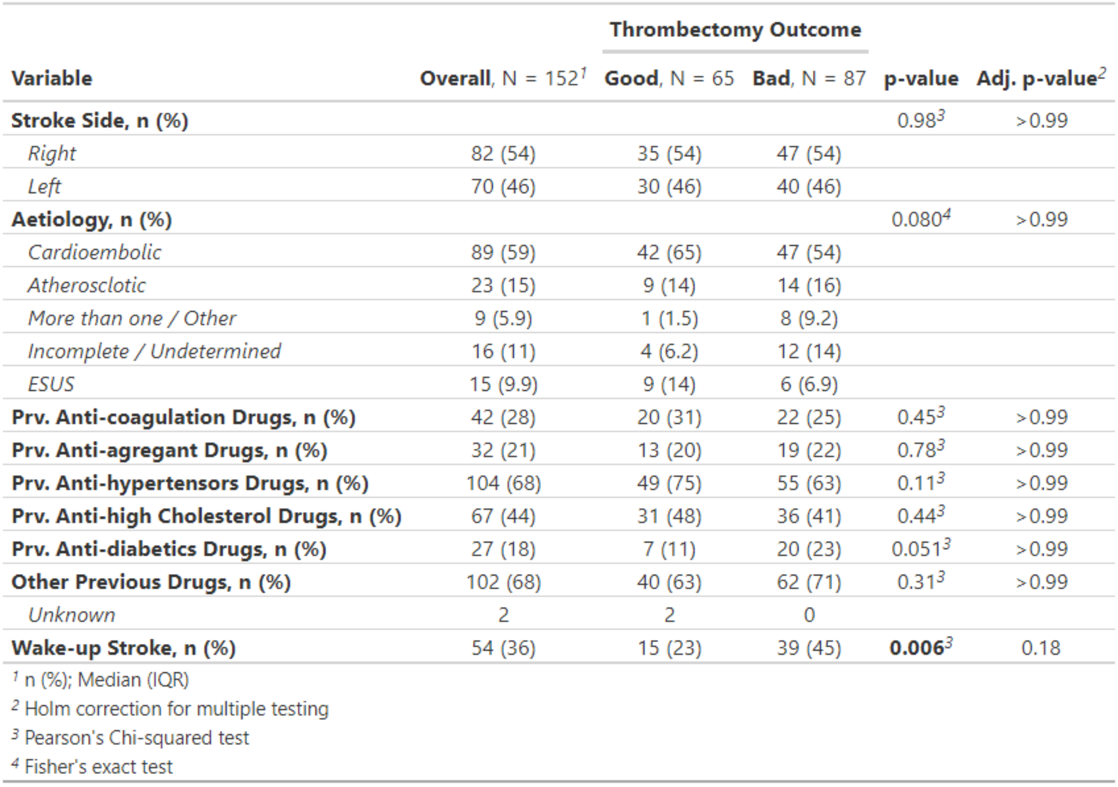
		Thrombectomy Outcome			
Variable	**Overall**, N = 152[1]	**Good**, N = 65	**Bad**, N = 87	**p-value**	**Adj. p-value**[2]
Sex, n (%)				0.23[3]	>0.99
Female	81 (53)	31 (48)	50 (57)		
Male	71 (47)	34 (52)	37 (43)		
Age at Stroke, Median (IQR)	76 (66 – 84)	73 (66 – 81)	79 (68 – 86)	**0.018**[4]	0.57
mRS before event, Median (IQR)	0 (0 – 2)	0 (0 – 0)	1 (0 – 2)	**<0.001**[4]	**<0.001**
NIH Stroke Scale at arrival, Median (IQR)	14 (9 – 20)	11 (8 – 17)	17 (12 – 21)	**<0.001**[4]	**0.007**
Cortical stroke (Right sided), n (%)	62 (41)	30 (46)	32 (37)	0.24[3]	>0.99
Hypertension, n (%)	113 (74)	50 (77)	63 (72)	0.53[3]	>0.99
High cholesterol, n (%)	81 (53)	35 (54)	46 (53)	0.91[3]	>0.99
Diabetes, n (%)	30 (20)	9 (14)	21 (24)	0.11[3]	>0.99
Atrial Fibrilation, n (%)	51 (34)	23 (35)	28 (32)	0.68[3]	>0.99
Chronic Renal Disease, n (%)	13 (8.6)	3 (4.6)	10 (11)	0.13[3]	>0.99
Chronic Obstructive Pulmonary Disease, n (%)	18 (12)	7 (11)	11 (13)	0.72[3]	>0.99
Coronary Artery Disease, n (%)	11 (7.2)	5 (7.7)	6 (6.9)	>0.99[5]	>0.99
Heart Failure, n (%)	34 (22)	9 (14)	25 (29)	**0.029**[3]	0.91
Acute Myocardial Infarct, n (%)	5 (3.3)	2 (3.1)	3 (3.4)	>0.99[5]	>0.99
Dementia, n (%)	13 (8.6)	4 (6.2)	9 (10)	0.36[3]	>0.99
Previous Stroke , n (%)	28 (18)	13 (20)	15 (17)	0.66[3]	>0.99

[1] n (%); Median (IQR)
[2] Holm correction for multiple testing
[3] Pearson's Chi-squared test
[4] Wilcoxon rank sum test
[5] Fisher's exact test

individual conditions evaluated on this study were not stat. sig. for thrombectomy outcome, $p.adj. \geq 0.91$.

Only patients with lateralized strokes were selected for this thesis cohort, and there is a slight prevalence to AIS on the right hemisphere. Neither side of the AIS nor its aetiology are stat. sig. to thrombectomy outcome, $p.adj > 0.99$. No previous medication group had significantly statistical relevance to the outcomes, $p.adj. > 0.99$ for all groups. One relevant variable on this table is the analysis of stroke evidence at patients' wake up — before calling the emergency services. Although shown not stat. sig., $p.adj. = 0.18$, it was considered highly significant without multiple test correction, $p\text{-}value = 0.006$, and it was the third most significant variable in this ds.. **Bad outcomes have a higher proportion of strokes at wake up.** Emergency episodes logistic data shows no stat. sig. variable, nor suggests relevant features from unadjusted p-values, not even for times from onset, as shown in table 6.3.

TABLE 6.2: Diagnosis and medication history data descriptive and statistical summary.



		Thrombectomy Outcome			
Variable	**Overall**, N = 152[1]	**Good**, N = 65	**Bad**, N = 87	**p-value**	**Adj. p-value**[2]
Stroke Side, n (%)				0.98[3]	>0.99
Right	82 (54)	35 (54)	47 (54)		
Left	70 (46)	30 (46)	40 (46)		
Aetiology, n (%)				0.080[4]	>0.99
Cardioembolic	89 (59)	42 (65)	47 (54)		
Atherosclotic	23 (15)	9 (14)	14 (16)		
More than one / Other	9 (5.9)	1 (1.5)	8 (9.2)		
Incomplete / Undetermined	16 (11)	4 (6.2)	12 (14)		
ESUS	15 (9.9)	9 (14)	6 (6.9)		
Prv. Anti-coagulation Drugs, n (%)	42 (28)	20 (31)	22 (25)	0.45[3]	>0.99
Prv. Anti-agregant Drugs, n (%)	32 (21)	13 (20)	19 (22)	0.78[3]	>0.99
Prv. Anti-hypertensors Drugs, n (%)	104 (68)	49 (75)	55 (63)	0.11[3]	>0.99
Prv. Anti-high Cholesterol Drugs, n (%)	67 (44)	31 (48)	36 (41)	0.44[3]	>0.99
Prv. Anti-diabetics Drugs, n (%)	27 (18)	7 (11)	20 (23)	0.051[3]	>0.99
Other Previous Drugs, n (%)	102 (68)	40 (63)	62 (71)	0.31[3]	>0.99
Unknown	2	2	0		
Wake-up Stroke, n (%)	54 (36)	15 (23)	39 (45)	**0.006**[3]	0.18

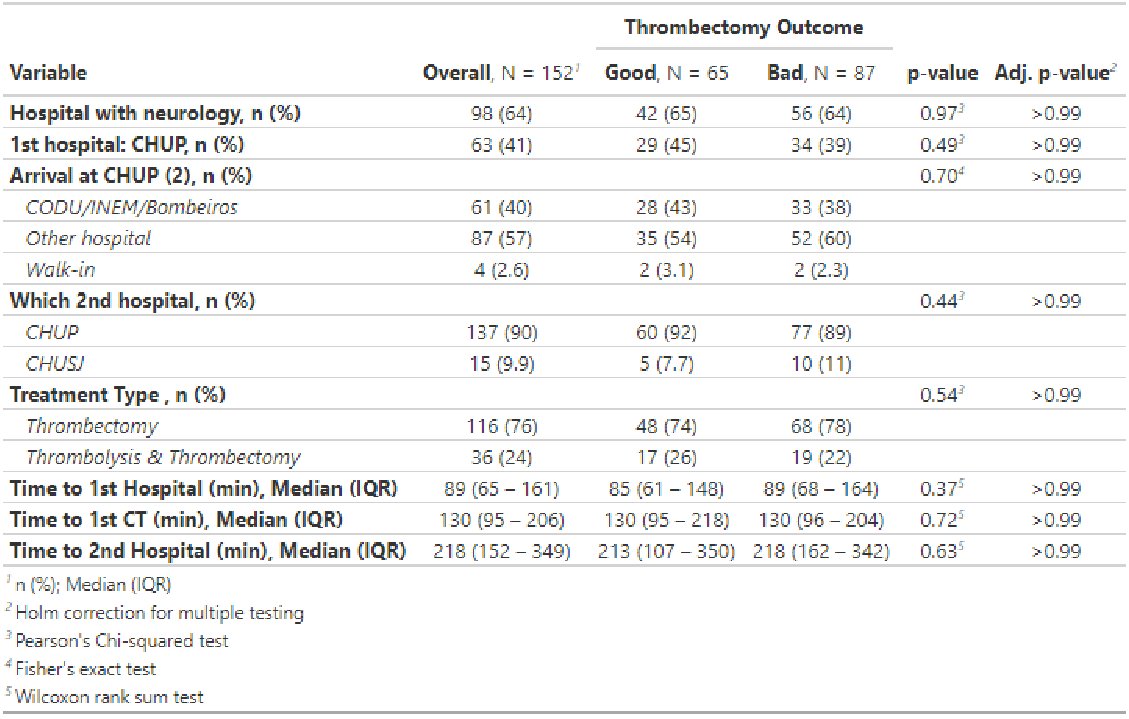
[1] n (%); Median (IQR)

[2] Holm correction for multiple testing

[3] Pearson's Chi-squared test

[4] Fisher's exact test

TABLE 6.3: Emergency episodes logistic data descriptive and statistical summary.



		Thrombectomy Outcome			
Variable	**Overall**, N = 152[1]	**Good**, N = 65	**Bad**, N = 87	**p-value**	**Adj. p-value**[2]
Hospital with neurology, n (%)	98 (64)	42 (65)	56 (64)	0.97[3]	>0.99
1st hospital: CHUP, n (%)	63 (41)	29 (45)	34 (39)	0.49[3]	>0.99
Arrival at CHUP (2), n (%)				0.70[4]	>0.99
CODU/INEM/Bombeiros	61 (40)	28 (43)	33 (38)		
Other hospital	87 (57)	35 (54)	52 (60)		
Walk-in	4 (2.6)	2 (3.1)	2 (2.3)		
Which 2nd hospital, n (%)				0.44[3]	>0.99
CHUP	137 (90)	60 (92)	77 (89)		
CHUSJ	15 (9.9)	5 (7.7)	10 (11)		
Treatment Type , n (%)				0.54[3]	>0.99
Thrombectomy	116 (76)	48 (74)	68 (78)		
Thrombolysis & Thrombectomy	36 (24)	17 (26)	19 (22)		
Time to 1st Hospital (min), Median (IQR)	89 (65 – 161)	85 (61 – 148)	89 (68 – 164)	0.37[5]	>0.99
Time to 1st CT (min), Median (IQR)	130 (95 – 206)	130 (95 – 218)	130 (96 – 204)	0.72[5]	>0.99
Time to 2nd Hospital (min), Median (IQR)	218 (152 – 349)	213 (107 – 350)	218 (162 – 342)	0.63[5]	>0.99

[1] n (%); Median (IQR)

[2] Holm correction for multiple testing

[3] Pearson's Chi-squared test

[4] Fisher's exact test

[5] Wilcoxon rank sum test

6.1.1 Model Selection

In the first stage, several more limited GSes were performed, adjusting the search parameters when they selected the limits of a range, which evaluated in total approximately 7000 models. The **best parameters found for each model** after cross-validated evaluation on this first phase can be seen in table 6.4. Logistic regression (LR) has performed better using the ℓ_1 regularization, which is consistent with the number of categorical variables, tree-based methods such as DT, RF, AdaBoost, XGBoost and LightGBM do not agree in the criterion for selecting branches, but all the best tree-based models were shallow, 2-4 levels deep, and they included many features, 21-22 features in DT and RF. k-NN used a moderate number of neighbours for interpolation, 7, and the SVM classifier kernel with the best performance was a Radial Based Function (RBF), — selected over polynomial kernels, suggesting a DB close described as a quadratic function. NN selected a funnel shape, resembling EfficientNet principles, even when the grid allowed more units in all hidden layers, and the best performing NN used the unusual identity function as the AF.

TABLE 6.4: Clinical Data best hyperparameters per modelling strategy.

Model	Parameters
LogisticRegression	{’classifier__C’: 1, ’classifier__max_iter’: 100, ’classifier__penalty’: ’l1’, ’classifier__solver’: ’saga’}
DecisionTreeClassifier	{’classifier__criterion’: ’gini’, ’classifier__max_depth’: 4, ’classifier__max_features’: 19, ’classifier__min_samples_leaf’: 22}
RandomForestClassifier	{’classifier__criterion’: ’gini’, ’classifier__max_depth’: 4, ’classifier__max_features’: 33, ’classifier__min_samples_leaf’: 11}
AdaBoostClassifier	{’classifier__base_estimator’: DecisionTreeClassifier(max_depth=2), ’classifier__learning_rate’: 0.05}
XGBClassifier	{’classifier__booster’: ’dart’, ’classifier__learning_rate’: 0.1, ’classifier__max_depth’: 4, ’classifier__min_child_weight’: 4}
LGBMClassifier	{’classifier__boosting_type’: ’gbdt’, ’classifier__learning_rate’: 0.1, ’classifier__max_depth’: 4, ’classifier__min_child_weight’: 4}
KNeighborsClassifier	{’classifier__n_neighbors’: 7, ’classifier__weights’: ’distance’}
QuadraticDiscriminantAnalysis	{’classifier__reg_param’: 1, ’classifier__tol’: 1e-07}
LinearDiscriminantAnalysis	{’classifier__shrinkage’: ’auto’, ’classifier__solver’: ’lsqr’, ’classifier__tol’: 1e-05}
CategoricalNB	{’classifier__alpha’: 1e-05, ’classifier__fit_prior’: True}
LinearSVC	{’classifier__C’: 1, ’classifier__loss’: ’hinge’, ’classifier__penalty’: ’l2’}
SVC	{’classifier__C’: 100, ’classifier__coef0’: 0, ’classifier__degree’: 2, ’classifier__gamma’: ’auto’, ’classifier__kernel’: ’poly’}
MLPClassifier	{’classifier__activation’: ’relu’, ’classifier__alpha’: 5e-06, ’classifier__hidden_layer_sizes’: (32, 16, 16, 16), ’classifier__learning_rate’: ’constant’, ’classifier__shuffle’: True, ’classifier__solver’: ’lbfgs’}

Cross-validated metrics analysis shows the best performing model depends on the metric to be considered, in table 6.5. The modelling strategy that achieved the best results in validation by F1-score was a DT, $F1\text{-}weighted = 0.73 \pm 0.15$, followed two other DT-based methods, RF and XGBM, both with $F1\text{-}weighted = 0.72 \pm 0.12$, and by LR with $F1\text{-}weighted = 0.71 \pm 0.14$. However, for our intended metric, the AUC scores, LR has shown the best scores with $AUC = 0.80 \pm 0.13$, only matched by the RF and XGBM classifier, both with $AUC = 0.78 \pm 0.13$. When one considers secondary metrics such as accuracy and balanced accuracy, two models stand out: DT with $Acc. = 0.74 \pm 0.13$ and $Bal.Acc. = 0.73 \pm 0.13$ and XGBM with $Acc. = 0.73 \pm 0.12$ and $Bal.Acc. = 0.72 \pm 0.12$, although LR and DT classifier were still among the top performing models, with $Acc. = 0.71 \pm 0.13$ and $Bal.Acc. = 0.71 \pm 0.13$, and $Acc. = 0.73 \pm 0.11$ and $Bal.Acc. = 0.71 \pm 0.12$ respectively.

TABLE 6.5: Best clinical models on each modelling strategy compared by their cross-validated metrics.

Model	Val. Acc.	Std.	Val. Bal.Acc.	Std.	Val. AUC	Std.	Val. F1-Weighted	Std.
LogisticRegression	0.71	0.13	0.71	0.13	0.80	0.13	0.71	0.14
DecisionTreeClassifier	0.74	0.13	0.72	0.14	0.70	0.15	0.73	0.15
RandomForestClassifier	0.73	0.11	0.71	0.12	0.78	0.13	0.72	0.12
AdaBoostClassifier	0.69	0.13	0.68	0.13	0.76	0.15	0.68	0.14
XGBClassifier	0.73	0.12	0.72	0.13	0.78	0.13	0.73	0.13
LGBMClassifier	0.72	0.12	0.71	0.12	0.78	0.13	0.71	0.12
KNeighborsClassifier	0.58	0.15	0.57	0.15	0.56	0.18	0.57	0.15
QuadraticDiscriminantAnalysis	0.59	0.14	0.55	0.14	0.61	0.17	0.55	0.15
LinearDiscriminantAnalysis	0.68	0.14	0.67	0.14	0.73	0.15	0.67	0.14
CategoricalNB	nan	nan	nan	nan	nan	nan	nan	nan
LinearSVC	0.68	0.13	0.67	0.13	0.72	0.14	0.68	0.13
SVC	0.66	0.13	0.65	0.13	0.69	0.15	0.65	0.13
MLPClassifier	0.66	0.12	0.65	0.13	0.67	0.15	0.65	0.13

Not all populations of cross-validated metrics had metrics in a normal distribution, so the Friedman's test was used to compare metrics populations, showing at least one model is stat. sig. different from other models, $p < 0.001$. However, measured cross-validated median AUC scores have shown overlaps on most models, as, for example, XGBM, LGBM, and RF are all within LR 95% Confidence Intervals (CIs), $AUC_{LR} = [0.76 - 0.84]$ A.2. Conducting the *post hoc* Nemenyi's *post hoc* test 6.1, it was possible to confirm that **LR was not stat. sig. different from XGBM, LGBM, RF, and AdaBoost but it was also shown to be different from all other models.** Several other groups of samples appear, and models from the top-tier group have other non-significant differences with tertiary models, such as the LDA and Linear Support Vector Machine Classifier (SVC).

Model comparison analysis using AUC scores tested on the test set shows if validation metrics are overly optimistic or pessimistic. Considering the test set only contains 31 samples, and nested-CV was not used, comparisons between validation metrics and test metrics are limited to comparing validation 95% CI A.2 with single sample values from the test metric 6.2. The best model in the test set was AdaBoost with $AUC_{AdaBoost_test} = 0.90$, which exceed the upper Confidence Interval (CI) in the validation set, $AUC_{AdaBoost_val} = [0.72 - 0.80]$. Considering the small number of samples, this effect is highly influenced by stochastic changes, and tests with different random seed have shown considerably different results. Linear SVC and SVC with kernel trick underperform in the test set but all remaining models all performed within their validation CIs as can be confirmed be seen comparing the values from A.2 with 6.2.

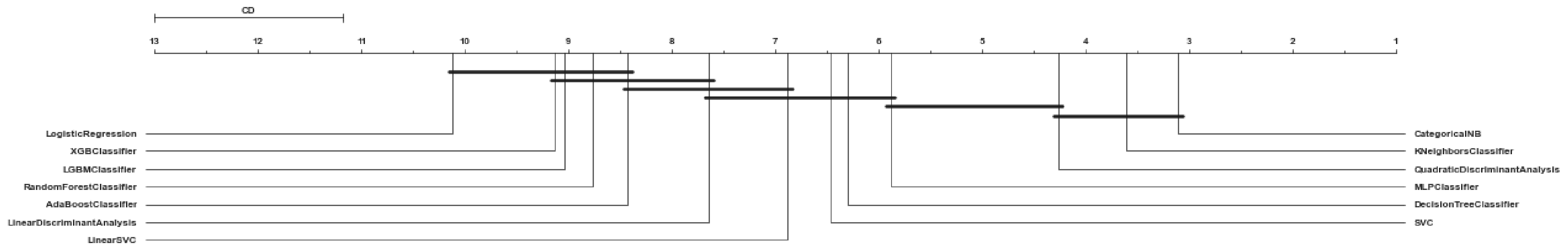


FIGURE 6.1: Nemenyi's plot comparing initial hyperparametrized models, trained on clinical data.

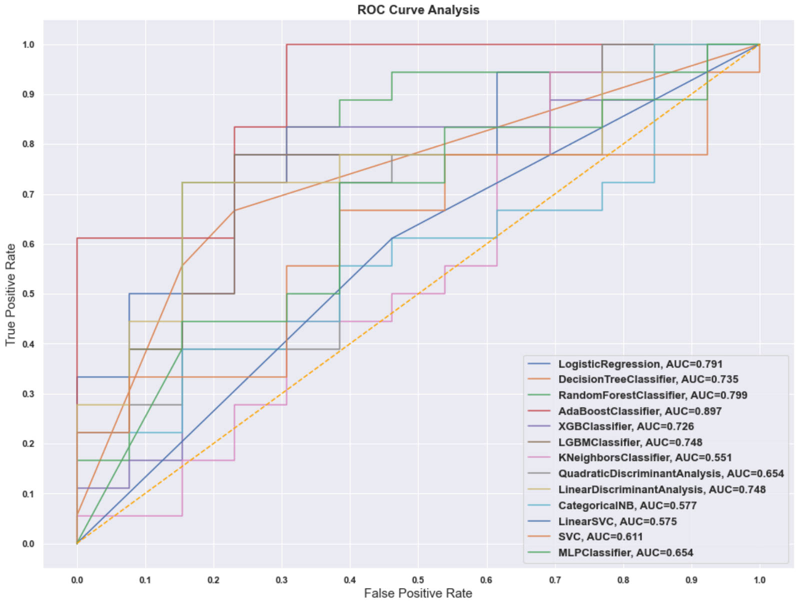


FIGURE 6.2: Receiver-Operating Curves for initial hyperparametrized models.

6.1.2 Automated Machine Learning on Clin0h (AutoML)

AutoML with AutoSkLearn 1.0 running for 10425 s tested 433 models and find the best model with $\overline{F1_{AutoSkLearn1.0_{val}}} = 0.77$, and 95% CIs $AUC_{AutoSkLearn1.0_{val}} = [0.73 - 0.80]$. This model is non-stat. sig. different from other top performing models, as indicated by the Nemenyi's critical distance shown in plot A.2. Evaluation on the test set has shown a value below the estimated CI, $AUC_{AutoSkLearn1.0_{val}} = 0.72$, suggesting poorer generalization ability than former top models.

AutoML with AutoSkLearn 2.0, also running for 10425 s tested many more models, 4002 in total, and it found a better model than AutoSkLearn 1.0 with $\overline{F1_{AutoSkLearn2.0_{val}}} = 0.78$ on validation, but considerably worse in the intended metric with $\overline{AUC_{AutoSkLearn2.0_{val}}} = 0.71$ and 95% CIs $AUC_{AutoSkLearn2.0_{val}} = [0.67 - 0.74]$. This suggests the models chosen by AutoSkLearn 2.0 was worse than the one selected by AutoSkLearn 1.0. Nemenyi's critical distances show that this model is statistically different from the one selected by version 1.0 strategy, and all the models within the best models group, as seen in figure A.2 and table A.3.

6.1.3 Data Augmentation

Comparing with the base models, **SMOTE followed by Tomek Links (TL) was unable to improve** the best modelling strategy on this ds., **but achieved the highest augmented AUC score** (0.78 ± 0.14) in this study, confirming findings from other authors regarding the **SMOTE-TomekLinks** combination, so it was used in subsequent work, as shown in table A.12. This study shows **Edited Nearest Neighbours (ENN)** is too aggressive in this ds. and its sample removal technique causes all models to have lower performance. Results of TL are not always positive, and some algorithms seem to perform worse with TL undersampling. This may be due stochastic effects on a mildly unbalanced ds., considering that all undersampling and oversampling techniques had mixed results, dependent on which model was being applied. If one wants **to maximize accuracy**, the best algorithms were a combination between undersampling with **TomekLinks, followed by oversampling with BorderlineSMOTE or SVMSMOTE, followed again by TomekLinks**.

The results in hyperparametrized models were not improved, with drops in average metric scores in the best performing model — LR dropped from $\overline{AUC} = 0.80$ to $\overline{AUC} = 0.77$, while XGBM and LGBM remained with $\overline{AUC} = 0.78$. Only the previously underfit model, Linear SVC, has shown improvement going from $\overline{AUC} = 0.72$ to $\overline{AUC} = 0.75$, as seen by comparing the data in table 6.5 with table 6.6. The randomness of variations in scores is confirmed by the Nemenyi's *post hoc* test shown in figure A.12. All models trained on SMOTE'd data underperformed the base LR although not significantly. Data augmentation did not bring significant improvements to tabular dss..

TABLE 6.6: Best augmented clinical models cross-validated metrics.

Model	Val. Acc.	Std.	Val. Bal.Acc.	Std.	Val. AUC	Std.	Val. F1-Weighted	Std.
LGBMClassifier (TL)	0.72	0.12	0.72	0.13	0.78	0.13	0.72	0.13
XGBClassifier (TL)	0.73	0.12	0.73	0.12	0.78	0.13	0.72	0.12
LinearSVC (TL)	0.68	0.14	0.68	0.15	0.75	0.15	0.68	0.15
SVC (TL)	0.68	0.12	0.67	0.13	0.70	0.17	0.67	0.13
LogisticRegression (TL)	0.70	0.13	0.70	0.13	0.78	0.14	0.69	0.13

Studies on oversampling each class to 200 samples — an 163% in the number of sample — was unable to offer better results, as seen in table 6.7 and by the Nemenyi's critical distances, shown in figure A.13. The results in table 6.7 are using non-parametric descriptor because some cross-validates metric set did not follow a normal distribution, which makes `autorank` automatically calculate the table differently. This makes it impossible to compare values directly, but the trends shown in previous analysis are maintained.

All models trained in the ds. with 400 samples did not perform better than LR model trained on the base ds. — having worse central tendency metrics — although XGBM and LGBM trained on augmented data have shown non-stat. sig. different from base LR model.

TABLE 6.7: SMOTE-TomekLinks augmented to 400 samples AUC metrics compared to base models.

	meanrank	median	mad	ci_lower	ci_upper	effect_size	magnitude
SVC (Augmented)	2.45	0.71	0.13	0.60	0.78	0.00	negligible
LinearSVC (Augmented)	3.98	0.76	0.10	0.68	0.83	-0.24	small
Random Forests (Base)	4.51	0.77	0.08	0.71	0.86	-0.35	small
LogisticRegression (Augmented)	4.72	0.77	0.11	0.71	0.85	-0.34	small
XGBoost (Base)	4.74	0.80	0.08	0.72	0.86	-0.52	medium
LGBMClassifier (Augmented)	4.84	0.78	0.08	0.74	0.86	-0.41	small
XGBClassifier (Augmented)	4.92	0.77	0.08	0.72	0.80	-0.34	small
Logistic Regression (Base)	5.83	0.80	0.08	0.74	0.88	-0.52	medium

6.2 Biomarkers Data

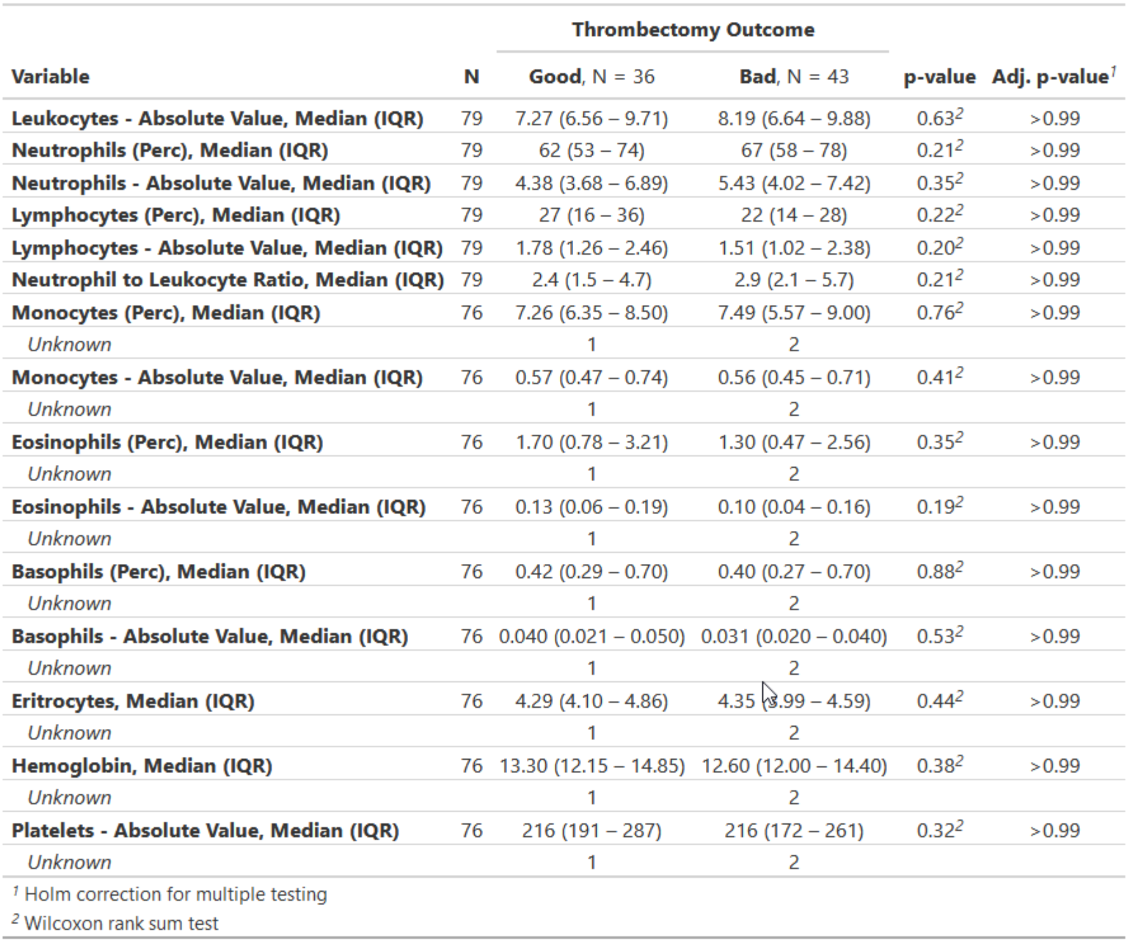
Considering biomarkers ds. was split between data collected on patient's admission and on patient's follow-up, analysis is done separately.

6.2.1 Biomarkers at admission (Biom0h)

Although clinical analysis have been conducted on every patient, this ds. is limited to 79 cases. Within the retrieved data, FBC data is mostly complete with only three cases with NAs in some quantification — i.e., monocytes, eosinophils, basophils, erythrocytes, platelets and haemoglobin. In all admitted patients. Neutrophil Lymphocyte Ratio (NLR), an important inflammation and tumour biomarker, was not considered stat. sig. to thrombectomy outcomes, $p.adj. > 0.99$. Indeed, **no FBC biomarker was considered significant to the outcomes, even without multiple comparison correction**, with the best unadjusted p-value was $p\text{-}value = 0.19$, as table 6.8 shows.

Table 6.9 shows biochemical biomarkers on follow-up and no biochemical biomarker has a good correlation with thrombectomy outcome, $p.adj. \geq 0.76$. Many biomarkers tested are indicators of inflammation, hepatic diseases, diabetes and other conditions that hinder the patients health and influence available treatment choices, so although they are essential for proper treatment, they have little relation with recovery from thrombectomy outcomes, as the statistical tests suggest. Considering how small this ds. is — with

TABLE 6.8: Hemogram descriptive and statistical analysis on Biom0h.



		Thrombectomy Outcome			
Variable	**N**	**Good**, N = 36	**Bad**, N = 43	**p-value**	**Adj. p-value**[1]
Leukocytes - Absolute Value, Median (IQR)	79	7.27 (6.56 – 9.71)	8.19 (6.64 – 9.88)	0.63[2]	>0.99
Neutrophils (Perc), Median (IQR)	79	62 (53 – 74)	67 (58 – 78)	0.21[2]	>0.99
Neutrophils - Absolute Value, Median (IQR)	79	4.38 (3.68 – 6.89)	5.43 (4.02 – 7.42)	0.35[2]	>0.99
Lymphocytes (Perc), Median (IQR)	79	27 (16 – 36)	22 (14 – 28)	0.22[2]	>0.99
Lymphocytes - Absolute Value, Median (IQR)	79	1.78 (1.26 – 2.46)	1.51 (1.02 – 2.38)	0.20[2]	>0.99
Neutrophil to Leukocyte Ratio, Median (IQR)	79	2.4 (1.5 – 4.7)	2.9 (2.1 – 5.7)	0.21[2]	>0.99
Monocytes (Perc), Median (IQR)	76	7.26 (6.35 – 8.50)	7.49 (5.57 – 9.00)	0.76[2]	>0.99
Unknown		1	2		
Monocytes - Absolute Value, Median (IQR)	76	0.57 (0.47 – 0.74)	0.56 (0.45 – 0.71)	0.41[2]	>0.99
Unknown		1	2		
Eosinophils (Perc), Median (IQR)	76	1.70 (0.78 – 3.21)	1.30 (0.47 – 2.56)	0.35[2]	>0.99
Unknown		1	2		
Eosinophils - Absolute Value, Median (IQR)	76	0.13 (0.06 – 0.19)	0.10 (0.04 – 0.16)	0.19[2]	>0.99
Unknown		1	2		
Basophils (Perc), Median (IQR)	76	0.42 (0.29 – 0.70)	0.40 (0.27 – 0.70)	0.88[2]	>0.99
Unknown		1	2		
Basophils - Absolute Value, Median (IQR)	76	0.040 (0.021 – 0.050)	0.031 (0.020 – 0.040)	0.53[2]	>0.99
Unknown		1	2		
Eritrocytes, Median (IQR)	76	4.29 (4.10 – 4.86)	4.35 (3.99 – 4.59)	0.44[2]	>0.99
Unknown		1	2		
Hemoglobin, Median (IQR)	76	13.30 (12.15 – 14.85)	12.60 (12.00 – 14.40)	0.38[2]	>0.99
Unknown		1	2		
Platelets - Absolute Value, Median (IQR)	76	216 (191 – 287)	216 (172 – 261)	0.32[2]	>0.99
Unknown		1	2		

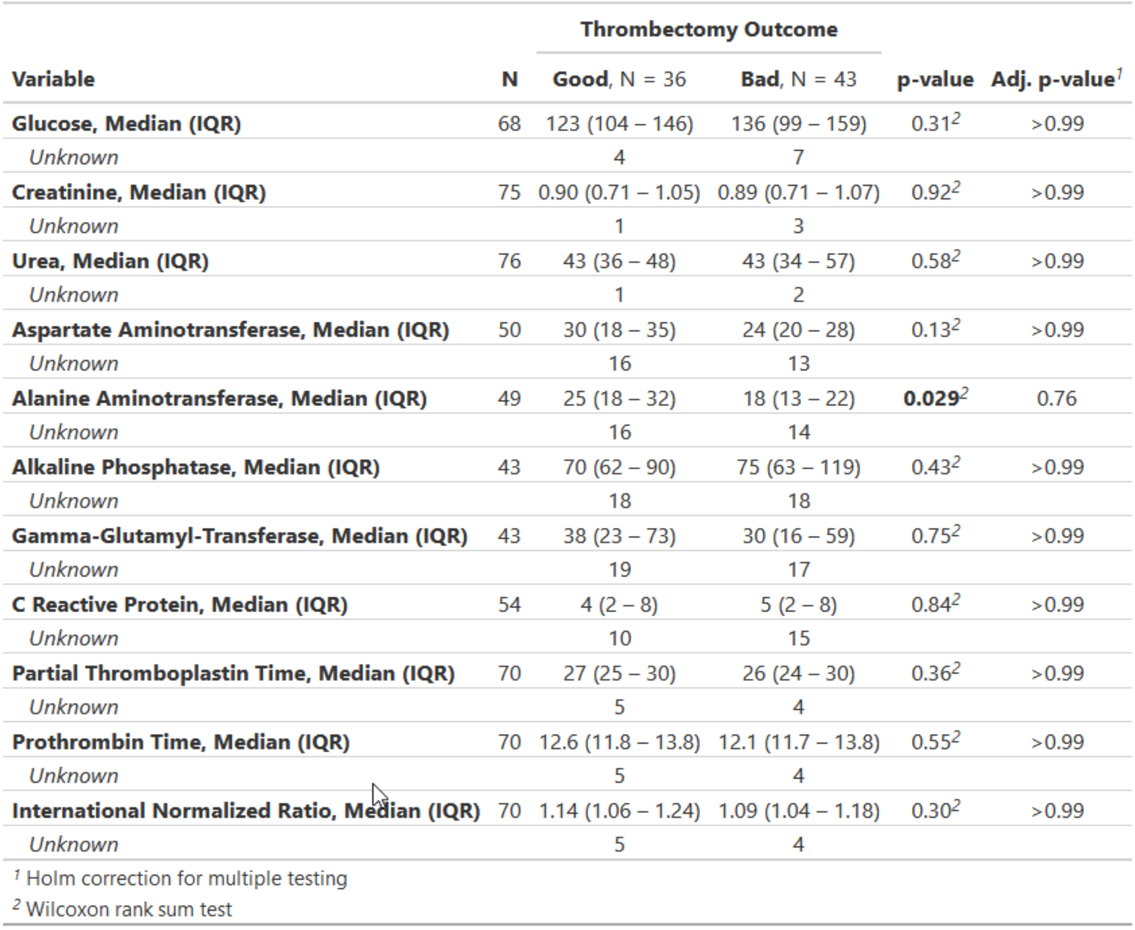
[1] Holm correction for multiple testing
[2] Wilcoxon rank sum test

some biomarkers having down to 17 samples in one class, as it is the case of Gamma-Glutamyl-Transferase — effects may be revealed in larger dss., and p-values without multiple comparisons corrections can be used as indicators for further research. Only Alanine Aminotransferase (ALA) — a biomarker to diagnose liver damage — was stat. sig. without Family-Wise Error Rate (FWER) corrections, $p\text{-}value = 0.03$.

6.2.1.1 Model Selection

Considering NAs profile shown in image A.5, this ds. required imputations to be applied extensively. Since imputation methods heavily influence the results, a study using **NAGuideR** was conducted on this ds. to assess which imputation method would be preferable. The ranked NAGuideR results of the **eleven imputation algorithms**, in table 6.10, shows that **three out of four metrics consider RF imputation, *rf*, the best algorithm for this ds.**, followed by imputation by median value per biomarker, *rowmedian* — note the table was transposed for NAGuideR processing. Quantile regression, *iqr*, was

TABLE 6.9: Biochemical analysis descriptive and statistical analysis on Biom0h.



		Thrombectomy Outcome			
Variable	**N**	**Good**, N = 36	**Bad**, N = 43	**p-value**	**Adj. p-value**[1]
Glucose, Median (IQR)	68	123 (104 – 146)	136 (99 – 159)	0.31[2]	>0.99
Unknown		4	7		
Creatinine, Median (IQR)	75	0.90 (0.71 – 1.05)	0.89 (0.71 – 1.07)	0.92[2]	>0.99
Unknown		1	3		
Urea, Median (IQR)	76	43 (36 – 48)	43 (34 – 57)	0.58[2]	>0.99
Unknown		1	2		
Aspartate Aminotransferase, Median (IQR)	50	30 (18 – 35)	24 (20 – 28)	0.13[2]	>0.99
Unknown		16	13		
Alanine Aminotransferase, Median (IQR)	49	25 (18 – 32)	18 (13 – 22)	**0.029**[2]	0.76
Unknown		16	14		
Alkaline Phosphatase, Median (IQR)	43	70 (62 – 90)	75 (63 – 119)	0.43[2]	>0.99
Unknown		18	18		
Gamma-Glutamyl-Transferase, Median (IQR)	43	38 (23 – 73)	30 (16 – 59)	0.75[2]	>0.99
Unknown		19	17		
C Reactive Protein, Median (IQR)	54	4 (2 – 8)	5 (2 – 8)	0.84[2]	>0.99
Unknown		10	15		
Partial Thromboplastin Time, Median (IQR)	70	27 (25 – 30)	26 (24 – 30)	0.36[2]	>0.99
Unknown		5	4		
Prothrombin Time, Median (IQR)	70	12.6 (11.8 – 13.8)	12.1 (11.7 – 13.8)	0.55[2]	>0.99
Unknown		5	4		
International Normalized Ratio, Median (IQR)	70	1.14 (1.06 – 1.24)	1.09 (1.04 – 1.18)	0.30[2]	>0.99
Unknown		5	4		

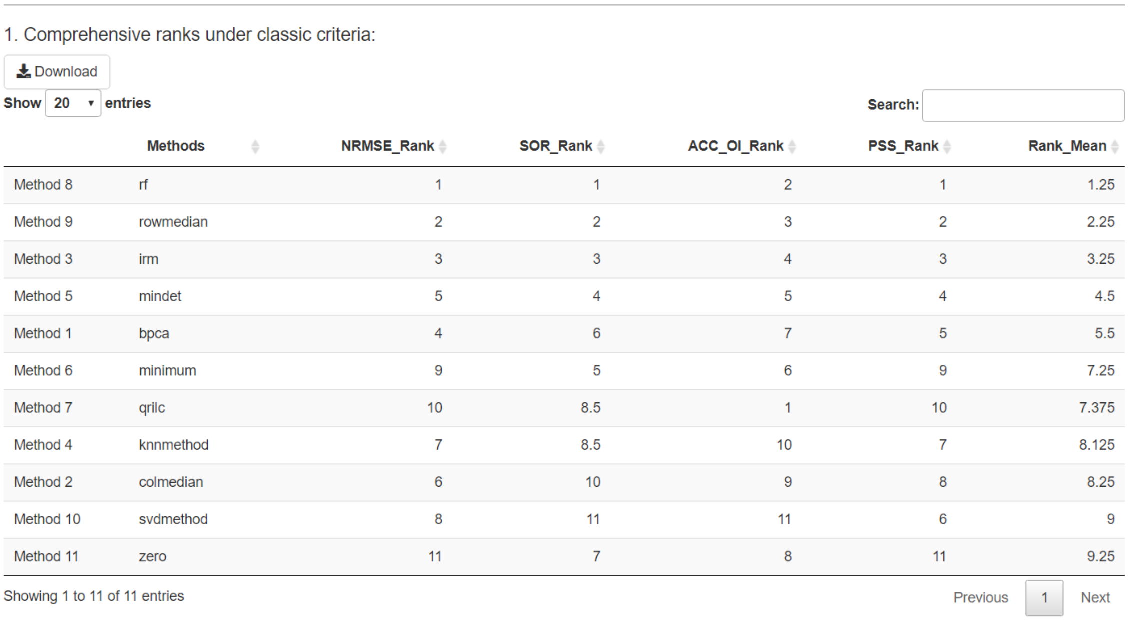
[1] Holm correction for multiple testing

[2] Wilcoxon rank sum test

the only method that surpasses both above-mentioned methods on Average Correlation Coefficient between Original value and Imputed value (ACC_OI).

Tables A.7 and A.8 with individual ranking metrics show **NAGuideR misclassifies *iqr* as the best imputation method in relation to ACC_OI**, since its average correlation coefficient is the one closest to 0, showing little to no correlation between predicted and expected value, which makes it the worst performing method by this metric. This was possibly due to a bug in one internal function, leading it to overvalue negative coefficients. As such, only the dss. imputed with the two best imputation methods were modelled, modelling the second one for evaluation of imputation method influence. Although studies on NAs imputations have come to the conclusion that RF is the best performing method for most dss. [127, 178], within the selected group of methods chosen in this thesis, this was not shown. None of the selected methods is particularly adequate for data missing not at random, which does not seem to be the case, since clinical exam values indicate when tested values are below detection threshold — e.g., some cells in the ds.

TABLE 6.10: NAGuideR imputation methods ranking on Biomarkers at admission dataset.



1. Comprehensive ranks under classic criteria:

	Methods	NRMSE_Rank	SOR_Rank	ACC_OI_Rank	PSS_Rank	Rank_Mean
Method 8	rf	1	1	2	1	1.25
Method 9	rowmedian	2	2	3	2	2.25
Method 3	irm	3	3	4	3	3.25
Method 5	mindet	5	4	5	4	4.5
Method 1	bpca	4	6	7	5	5.5
Method 6	minimum	9	5	6	9	7.25
Method 7	qrilc	10	8.5	1	10	7.375
Method 4	knnmethod	7	8.5	10	7	8.125
Method 2	colmedian	6	10	9	8	8.25
Method 10	svdmethod	8	11	11	6	9
Method 11	zero	11	7	8	11	9.25

Showing 1 to 11 of 11 entries

contained < 0.001 before preprocessing. In this study, these entries were converted during preprocessing to the numeric form by replacing by the minimum detection value.

Comparison among all models seems to suggest that **both imputation methods offer similar results**, with some modelling strategies on the same ds. performing better with row median, while other modelling strategies do not, as seen in figure A.9. Given how distinct both strategies and values are, it can be assumed **both imputation methods are suboptimal**. However, both allow modelling without discarding excessive amounts of data, and the best XGBM-based models achieve median AUC scores of 0.79, which is among the best performances recorded, in table A.10. These models still underperform the best LR-based model on the clinical ds. due to their wider confidence interval, although not significantly as seen by the Nemenyi's plot in figure A.11, suggesting biomarkers models have greater variance than their clinical counterparts, probably due to inadequate imputation methods and the number of imputed values in each cross-validated set.

Testing the models against **the test set shows these underperform the null model**, with AUC scores below 0.50. While models based on the ds. with RF imputations performed slightly better than the models that used the ds. with row median imputations, the low values obtained confirm the lack of correlation suggested by statistical tests, with the extra bias introduced by multiple imputations.

6.2.2 Biomarkers on follow-up (Biom24h))

This subset of the original biomarkers ds. preserved 126 cases, with only three cases with missing FBC information. Biochemical information on follow-up for the triaged biomarkers was present in at least 78 cases per biomarker with more than 100 data points on eight biomarkers. There is a mild class imbalance, with 57.9% of the bad outcome cases and 42.1% good outcome cases.

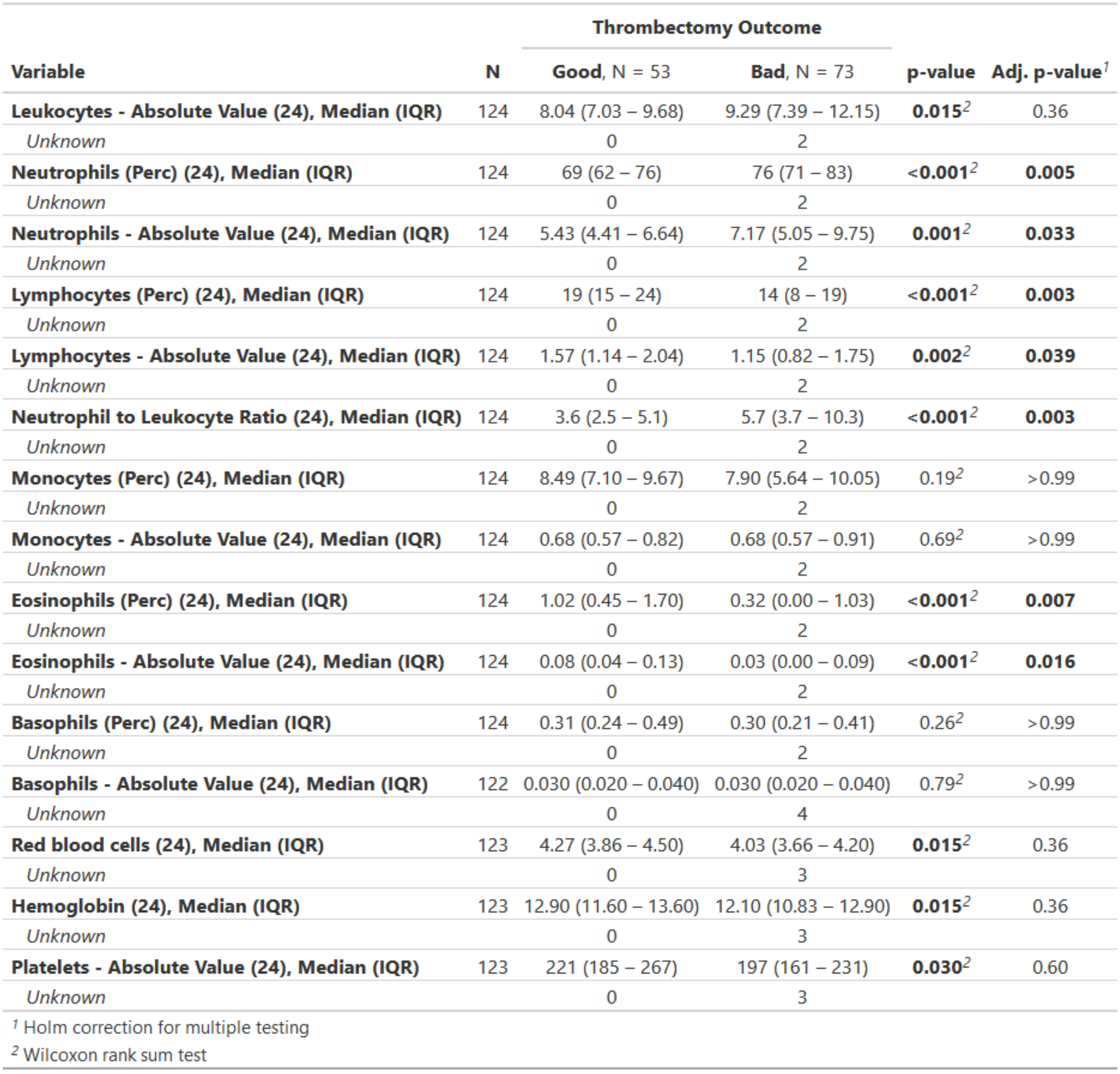
6.2.2.1 Descriptive and Statistical Analysis

Tables 6.11 and 6.12 show Biom24h ds. descriptive and statistical analysis. Unlike in the analysis of at admission ds., **stat. sig. biomarkers were found**, which is expected considering that these biomarkers should correlate to surgical success and therefore, patients' health condition after a thrombectomy.

Some blood cell quantifications are stat. sig. while no biochemical biomarkers has statistical significance after multiple comparisons corrections, as shown in tables 6.11 and 6.12. While total leucocyte count does not have stat. sig. relation with thrombectomy outcomes, $p.adj. = 0.36$, unadjusted values suggest it is an important variable with $p\text{-}value = 0.02$. Leucocyte type quantifications, but, where shown to be more relevant, especially NLR related quantifications, with neutrophil and lymphocytes proportion, as well as NLR itself being all highly stat. sig. with $p.adj. < 0.01$, with neutrophil and lymphocyte quantifications being stat. sig., $p.adj. < 0.05$. All these quantifiers are known to be associated with ischaemic events and revascularization outcomes, particularly neutrophil higher counts and high NLR, while lymphocyte counts are usually more associated with the inflammation, and they sharply decline when acute kidney and liver damage occurs, further increasing NLR [179]. **Bad outcomes have higher neutrophils concentrations and proportions, as well as higher NLR**, than good outcomes, while the opposite is true for lymphocytes, as table 6.11 shows.

Eosinophils were also considered stat. sig., $p.adj._{Eosinophil_{\%}} = 0.007$ and $p.adj._{Eosinophil_{\%}} = 0.02$. Although this specific blood cell type has no clear correlation with AIS when hyper-eosinophilic syndrome is not present — a type of eosinophils' hyperregulation when infection by metazoans occurs —, the correlation between eosinophils and AIS outcomes is well documented [180, 181].

TABLE 6.11: Full blood count descriptive and statistical analysis on Biom24h.



		Thrombectomy Outcome			
Variable	**N**	**Good**, N = 53	**Bad**, N = 73	**p-value**	**Adj. p-value**[1]
Leukocytes - Absolute Value (24), Median (IQR)	124	8.04 (7.03 – 9.68)	9.29 (7.39 – 12.15)	**0.015**[2]	0.36
Unknown		0	2		
Neutrophils (Perc) (24), Median (IQR)	124	69 (62 – 76)	76 (71 – 83)	**<0.001**[2]	**0.005**
Unknown		0	2		
Neutrophils - Absolute Value (24), Median (IQR)	124	5.43 (4.41 – 6.64)	7.17 (5.05 – 9.75)	**0.001**[2]	**0.033**
Unknown		0	2		
Lymphocytes (Perc) (24), Median (IQR)	124	19 (15 – 24)	14 (8 – 19)	**<0.001**[2]	**0.003**
Unknown		0	2		
Lymphocytes - Absolute Value (24), Median (IQR)	124	1.57 (1.14 – 2.04)	1.15 (0.82 – 1.75)	**0.002**[2]	**0.039**
Unknown		0	2		
Neutrophil to Leukocyte Ratio (24), Median (IQR)	124	3.6 (2.5 – 5.1)	5.7 (3.7 – 10.3)	**<0.001**[2]	**0.003**
Unknown		0	2		
Monocytes (Perc) (24), Median (IQR)	124	8.49 (7.10 – 9.67)	7.90 (5.64 – 10.05)	0.19[2]	>0.99
Unknown		0	2		
Monocytes - Absolute Value (24), Median (IQR)	124	0.68 (0.57 – 0.82)	0.68 (0.57 – 0.91)	0.69[2]	>0.99
Unknown		0	2		
Eosinophils (Perc) (24), Median (IQR)	124	1.02 (0.45 – 1.70)	0.32 (0.00 – 1.03)	**<0.001**[2]	**0.007**
Unknown		0	2		
Eosinophils - Absolute Value (24), Median (IQR)	124	0.08 (0.04 – 0.13)	0.03 (0.00 – 0.09)	**<0.001**[2]	**0.016**
Unknown		0	2		
Basophils (Perc) (24), Median (IQR)	124	0.31 (0.24 – 0.49)	0.30 (0.21 – 0.41)	0.26[2]	>0.99
Unknown		0	2		
Basophils - Absolute Value (24), Median (IQR)	122	0.030 (0.020 – 0.040)	0.030 (0.020 – 0.040)	0.79[2]	>0.99
Unknown		0	4		
Red blood cells (24), Median (IQR)	123	4.27 (3.86 – 4.50)	4.03 (3.66 – 4.20)	**0.015**[2]	0.36
Unknown		0	3		
Hemoglobin (24), Median (IQR)	123	12.90 (11.60 – 13.60)	12.10 (10.83 – 12.90)	**0.015**[2]	0.36
Unknown		0	3		
Platelets - Absolute Value (24), Median (IQR)	123	221 (185 – 267)	197 (161 – 231)	**0.030**[2]	0.60
Unknown		0	3		

[1] Holm correction for multiple testing

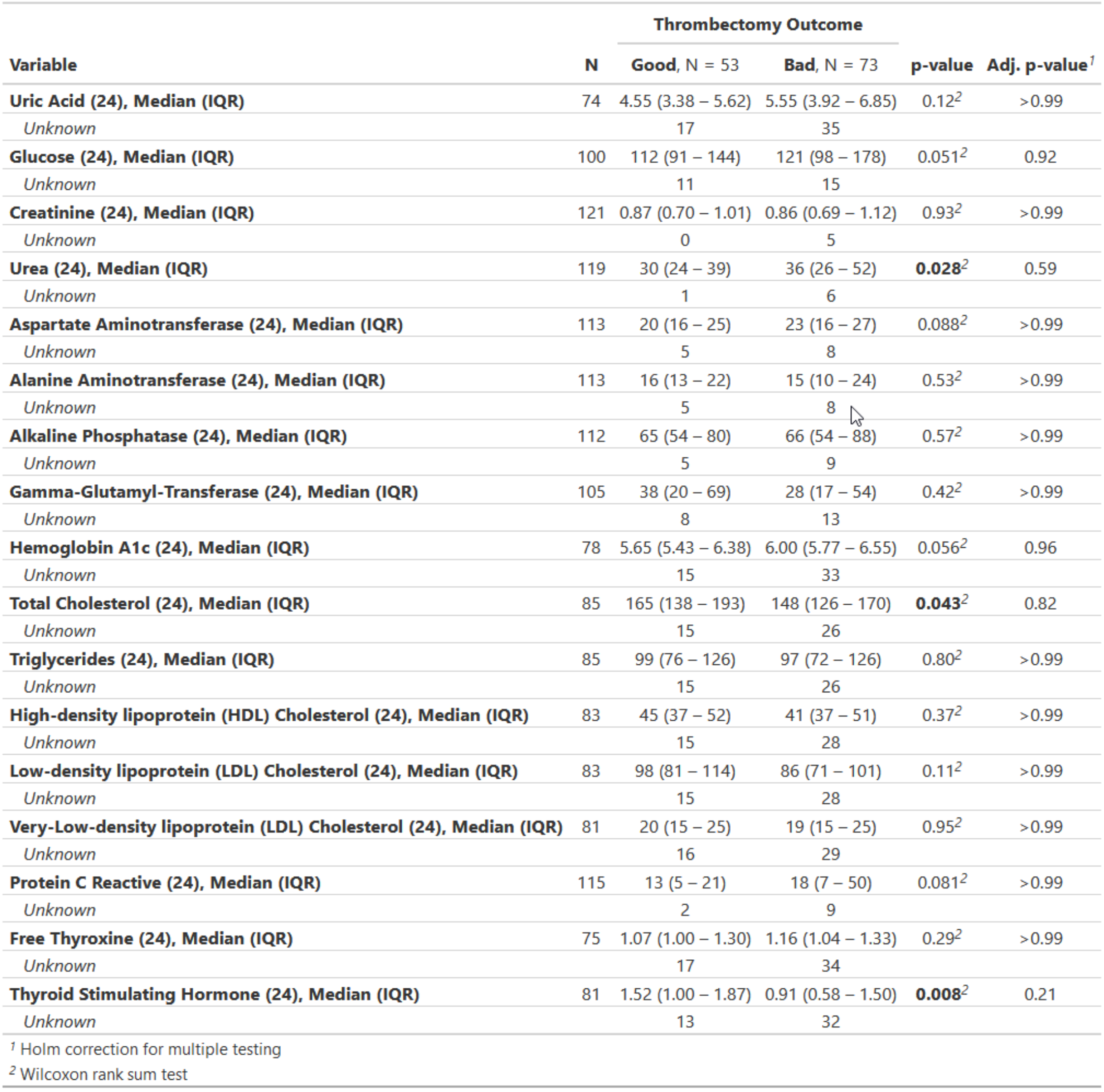
[2] Wilcoxon rank sum test

Basophils and monocytes have shown no statistical significant correlation, despite being recruited by neutrophils and; therefore, being associated with their increased presence. Basophils, given their primary role in foreign bodies controls, such as allergic compounds, and in inflammatory events, do not have an obvious relation with AIS. Their relation with heparin regulation could influence clotting events or the recovery from a clotting event, but their recorded values nearly identical in both outcome classes, both in Biom0h and in Biom24h.

The three remaining FBC quantification — i.e., red blood cells, platelets and haemoglobin — were also not stat. sig. after multiple comparison corrections, but this all non-adjusted tests have rejected the null hypothesis. Erythrocyte and haemoglobin quantifications are important to assess anaemia, but no strong correlation exists with Ischaemic

Stroke (IS) and their outcomes. In this study, slightly lower values of both were observed in patients with poorer outcomes. Post-thrombectomy quantification are about 10% lower in patients with bad outcomes.

TABLE 6.12: Biochemical analysis descriptive and statistical analysis on Biom24h.



		Thrombectomy Outcome			
Variable	**N**	**Good**, N = 53	**Bad**, N = 73	**p-value**	**Adj. p-value**[1]
Uric Acid (24), Median (IQR)	74	4.55 (3.38 – 5.62)	5.55 (3.92 – 6.85)	0.12[2]	>0.99
Unknown		17	35		
Glucose (24), Median (IQR)	100	112 (91 – 144)	121 (98 – 178)	0.051[2]	0.92
Unknown		11	15		
Creatinine (24), Median (IQR)	121	0.87 (0.70 – 1.01)	0.86 (0.69 – 1.12)	0.93[2]	>0.99
Unknown		0	5		
Urea (24), Median (IQR)	119	30 (24 – 39)	36 (26 – 52)	**0.028**[2]	0.59
Unknown		1	6		
Aspartate Aminotransferase (24), Median (IQR)	113	20 (16 – 25)	23 (16 – 27)	0.088[2]	>0.99
Unknown		5	8		
Alanine Aminotransferase (24), Median (IQR)	113	16 (13 – 22)	15 (10 – 24)	0.53[2]	>0.99
Unknown		5	8		
Alkaline Phosphatase (24), Median (IQR)	112	65 (54 – 80)	66 (54 – 88)	0.57[2]	>0.99
Unknown		5	9		
Gamma-Glutamyl-Transferase (24), Median (IQR)	105	38 (20 – 69)	28 (17 – 54)	0.42[2]	>0.99
Unknown		8	13		
Hemoglobin A1c (24), Median (IQR)	78	5.65 (5.43 – 6.38)	6.00 (5.77 – 6.55)	0.056[2]	0.96
Unknown		15	33		
Total Cholesterol (24), Median (IQR)	85	165 (138 – 193)	148 (126 – 170)	**0.043**[2]	0.82
Unknown		15	26		
Triglycerides (24), Median (IQR)	85	99 (76 – 126)	97 (72 – 126)	0.80[2]	>0.99
Unknown		15	26		
High-density lipoprotein (HDL) Cholesterol (24), Median (IQR)	83	45 (37 – 52)	41 (37 – 51)	0.37[2]	>0.99
Unknown		15	28		
Low-density lipoprotein (LDL) Cholesterol (24), Median (IQR)	83	98 (81 – 114)	86 (71 – 101)	0.11[2]	>0.99
Unknown		15	28		
Very-Low-density lipoprotein (LDL) Cholesterol (24), Median (IQR)	81	20 (15 – 25)	19 (15 – 25)	0.95[2]	>0.99
Unknown		16	29		
Protein C Reactive (24), Median (IQR)	115	13 (5 – 21)	18 (7 – 50)	0.081[2]	>0.99
Unknown		2	9		
Free Thyroxine (24), Median (IQR)	75	1.07 (1.00 – 1.30)	1.16 (1.04 – 1.33)	0.29[2]	>0.99
Unknown		17	34		
Thyroid Stimulating Hormone (24), Median (IQR)	81	1.52 (1.00 – 1.87)	0.91 (0.58 – 1.50)	**0.008**[2]	0.21
Unknown		13	32		

[1] Holm correction for multiple testing

[2] Wilcoxon rank sum test

Table 6.12 shows descriptive and statistical analysis on biochemical biomarkers. While **no biochemical biomarker tested on follow-up has shown statistically relevant correlations to outcomes**, three biomarkers show statistical significance when this correction is not done. Urea concentration on follow-up has about 20% higher values on patients with bad outcomes. This effect was not observed in admission, suggesting kidney function has changed between both states, possibly due to anaesthetics reaction, which would explain its prevalence in patients with bad outcomes. Various cholesterol analysis were done on the follow-up but only total cholesterol was suggested as relevant before Holm's correction, showing about 10% lower values in patients with bad outcomes. Lastly, Thyroid Stimulating Hormone (TSH) has shown roughly 50% higher values in patients with good outcomes. Considering TSH is responsible for upregulating the metabolism of almost all living tissues, low values indicate low metabolism and consequently slower recovery from thrombectomy and other ailments the patient may have.

6.2.2.2 Model Selection

This ds. required even further imputations, 554 in total, as seen by NAs profile in figure A.5 and the exclusion of some biomarkers from analysis, as detailed in the methods section. The imputation method analysis done on the previous ds. was used on the Biom24h, so, modelling analysis was also done with complete cases after imputation with RF and row median. Results in table 6.13 confirm that using **row median ds. allowed slight better results on all modelling strategies**, with gains of up to 2%. The **modelling strategy with the best results in both dss. was XGBM**, $AUC_{XGBM\_RM} = 0.76 \pm 0.15$ and $AUC_{XGBM\_RM} = 0.74 \pm 0.15$, closely followed by LGBM, $AUC_{LGBM\_RM} = 0.72 \pm 0.16$ and $AUC_{LGBM\_RF} = 0.72 \pm 0.15$. XGBM and LGBM follow the same trend in other metrics, achieving the best performance in all standards. Non-parametric central tendency analysis shown in table A.10 also confirms these findings, with XGBM achieving median AUC scores above 0.79.

TABLE 6.13: Biom24h-based models metrics table including both imputation methods.

Model	Val. Acc.	Std.	Val. Bal.Acc.	Std.	Val. AUC	Std.	Val. F1-Weighted	Std.
LGBMClassifier (24h RF)	0.66	0.13	0.66	0.13	0.72	0.16	0.66	0.14
XGBClassifier (24h RF)	0.68	0.13	0.66	0.13	0.74	0.15	0.67	0.14
LinearSVC (24h RF)	0.63	0.14	0.61	0.14	0.67	0.16	0.62	0.15
SVC (24h RF)	0.64	0.15	0.61	0.15	0.67	0.16	0.62	0.16
LogisticRegression (24h RF)	0.63	0.14	0.61	0.14	0.68	0.16	0.61	0.15
LGBMClassifier (24h RM)	0.68	0.13	0.67	0.13	0.72	0.16	0.67	0.13
XGBClassifier (24h RM)	0.71	0.13	0.69	0.14	0.76	0.15	0.70	0.14
LinearSVC (24h RM)	0.64	0.14	0.63	0.15	0.70	0.16	0.63	0.14
SVC (24h RM)	0.66	0.14	0.62	0.14	0.68	0.16	0.63	0.15
LogisticRegression (24h RM)	0.64	0.15	0.61	0.15	0.68	0.16	0.62	0.16

Even though these summary statistics suggest a difference in performance between imputation methods, formal statistical model comparison achieved by Nemenyi's critical distances, shown in figure A.9, suggests the improvements seen in some models with a specific imputation strategy are casual, and not stat. sig.. **The best performing models**, by their ranked metrics are models trained on follow-up metrics, with only SVC with kernel transformation trained on Biom0h model **not showing significant statistical difference with the first group**. In this ds., models with high bias and linear DBs— such as LR and Linear SVC seem to underperform more complex models. When compared with the best models from the clinical ds., both XGBM models show better performance than LGBM, with XGBM not being stat. sig. different from the best model found so far, a LR trained on the clinical ds., as shown in figure A.11.

ROC curves on the test set also show that Biom24h models generalize better than Biom0h models, with all AUCs above the null model line, in figure 6.3. Unlike during validation, XGBM and LGBM models performed considerably worse than the other tested modelling strategies, achieving AUCs around 0.60, while high bias LR achieved $AUC_{LR\_RF} = 0.73$ and $AUC_{LR\_RM} = 0.78$. Although the test set analysis is not cross-validated, the values are below CIs, so these **XGBM and LGBM models show signs of overfitting** and should be considered carefully.

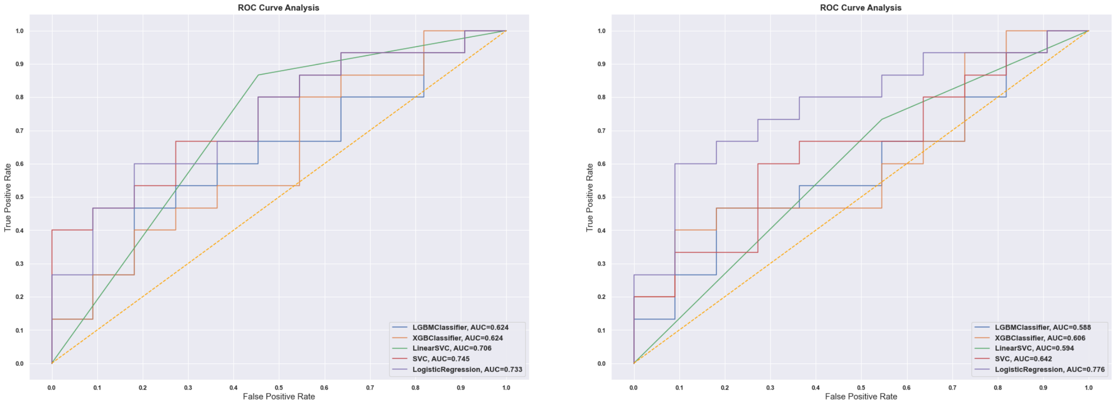


FIGURE 6.3: ROC curves for models trained on follow-up biomarkers dataset with MissingForest imputations, on the left, and with row median imputations, on the right.

6.3 Hughes Phenomenon (HP) on each dataset

HP —- also known as peaking phenomenon —, is the observation that **most modelling strategies only benefit for a certain amount of features** — signal features —, which past that point adding more features to a model does not improve performance and can even degrade it. Considering how light FS was done on tabular datasets, Hughes analysis was relevant to understand if by reducing the number of features, the model can be improved.

6.3.1 HP on Clinical Data at Admission (Clin0h FS)

The clinical ds. benefited from FS the most by trimming to only ten features since all modelling strategies had been initially tested on it. **All models gained increased performance** reflected in most metrics, but hyperparametrized LRs improved with FS was unable to surpass the previous best A.14. Ten features were selected on the clinical ds. considering peaking performance: age; previous modified Rankin Score (mRS); NIHSS at admission; history of chronic renal disease, heart failure or AIS; if the AIS is pro-thrombotic; and if the symptoms were detected at patient's wake-up. With these features, a maximum $\overline{med(AUC)} = 0.82$ was achieved. Since reparametrization on selected features did not show improvements, the LR model selected from HP analysis was reused, by training it on the average peak, $k = 10$, achieving $med(AUC) = 0.84$ and $CI_{AUC} = [0.77 - 0.91]$, values that made it stat. sig. from all previous models. The balancing and augmentation technique studies as best performing, SMOTE-TomekLinks as documented on appendix section A.4, has performed worse with $med(AUC) = 0.82$, but it was the only model from this group statistically similar according to Nemenyi' critical distances, as seen in figure A.16.

6.3.2 HP on Biomarkers Data at Admission (Biom0h FS)

Biom0h A.15 and Biom24h, A.16 have shown the same phenomenon but **did not improve with FS**, neither on the best model nor on adjusting underfit models. **Ten features were selected on Biom0h** considering peaking performance: neutrophils and eosinophils percentage, as well as eosinophils, haemoglobin, glycose, aspartate aminotransferase, alanine aminotransferase, alkaline phosphatase, C-reative protein concentrations, and partial thromboplastin time. With these features the models achieved a maximum $\overline{med(AUC)} = 0.70$. However, greater variability in these models makes the mean scores stay at more modest values.

6.3.3 HP on Biomarkers Data at Follow-up (Biom24h FS)

22 features were selected on Biom24h, considering peaking performance: leucocytes, neutrophils, lymphocytes, eosinophils, erythrocytes, platelets, haemoglobin, glycose, urea, aspartate transferase, alanine transferase, total cholesterol, LDL cholesterol, VLDL cholesterol, C-reactive protein, and tiroid-stimulating hormone concentrations; neutrophils, lymphocytes, monocytes, eosinophils, basophils, A1c haemoglobin percentages; and NLR. With these features the models achieved a maximum $\overline{med(AUC)} = 0.79$.

Considerations on HP study on ClinCA0h were done on a later stage, its description can be consulted in ClinCA0h section and appendices. Figure 6.4 summarizes HP in this ds., **ClinCA0h** Nemenyi's plot in figure A.20 and AUCs comparison tables in table A.17.

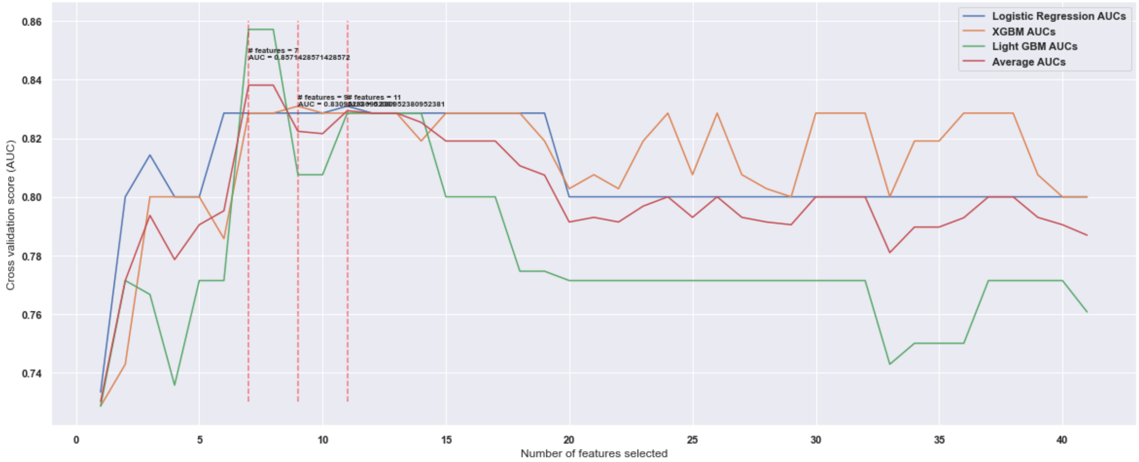


FIGURE 6.4: Hughes phenomenon analysis on the two top performing mixed clinical model with hemispheric contrast.

6.4 Imaging Data

6.4.1 Hemispheric Contrast (HC)

Some distribution differences exist in absolute HC, as shown in figure 6.5.

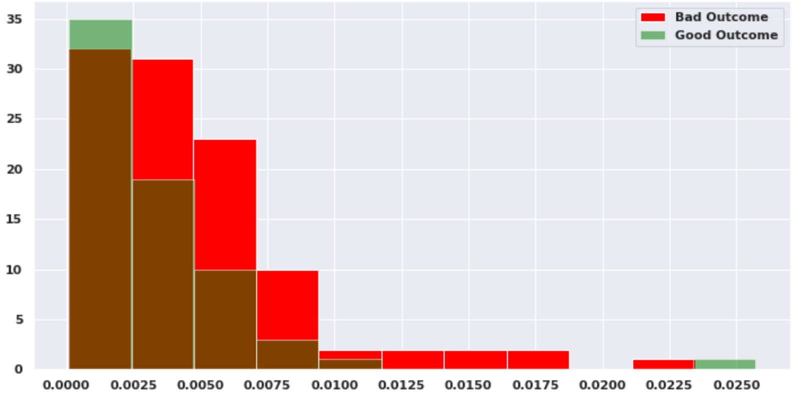
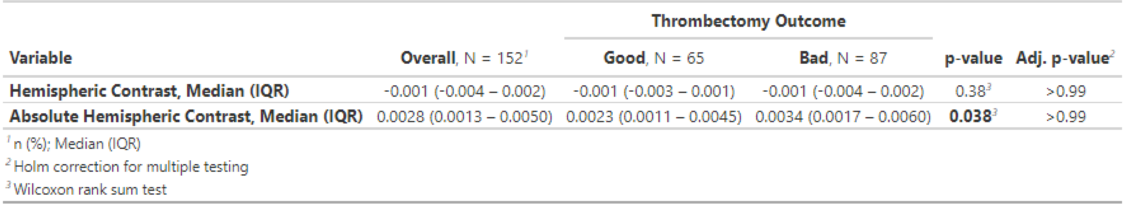


FIGURE 6.5: Absolute hemispheric contrast histogram.

Both **HC and absolute HC, when tested together with all clinical variables do not show stat. sig. correlation with outcomes**, but **absolute HC is stat. sig. when multiple comparison corrections are not performed**, $p\text{-}value = 0.04$, while signed HC stays non-SS with $p\text{-}value = 0.38$. For this reason, all further modelling work continued disregarding signed **HC**. Bad outcomes had higher medium absolute HC, as shown in table 6.14.

TABLE 6.14: Imaging biomarkers descriptive and statistical analysis.



		Thrombectomy Outcome			
Variable	**Overall**, N = 152[1]	**Good**, N = 65	**Bad**, N = 87	**p-value**	**Adj. p-value**[2]
Hemispheric Contrast, Median (IQR)	-0.001 (-0.004 – 0.002)	-0.001 (-0.003 – 0.001)	-0.001 (-0.004 – 0.002)	0.38[3]	>0.99
Absolute Hemispheric Contrast, Median (IQR)	0.0028 (0.0013 – 0.0050)	0.0023 (0.0011 – 0.0045)	0.0034 (0.0017 – 0.0060)	**0.038**[3]	>0.99

[1] n (%); Median (IQR)
[2] Holm correction for multiple testing
[3] Wilcoxon rank sum test

Absolute HC alone is not enough to produce good models, as seen in table 6.15, but better results were achieved as complementary feature. Those can be found in Mixed Models section within ClinCA0h subsection and ClinBiom24h with Feature Selection (FS) and Hemispheric Contrast (HC) subsections.

TABLE 6.15: Absolute HC only models.

Model	**Val. Acc.**	**Std.**	**Val. Bal.Acc.**	**Std.**	**Val. AUC**	**Std.**	**Val. F1-Weighted**	**Std.**
LGBMClassifier	0.60	0.03	0.50	0.00	0.63	0.12	0.46	0.04
XGBClassifier	0.60	0.13	0.58	0.14	0.60	0.14	0.59	0.13
LinearSVC	0.60	0.03	0.50	0.00	0.63	0.14	0.46	0.04
SVC	0.55	0.08	0.49	0.06	0.60	0.16	0.47	0.08
LogisticRegression	0.60	0.11	0.56	0.11	0.63	0.14	0.58	0.11

6.4.2 3D CNNs

The base model with patience hp. defined to 15 epochs has shown no signs of learning, with training ending before validation loss and accuracy converged and with its values swinging wildly, as seen in 6.6. This may be due to the low ILR and patience defined.

The classification report shows the model classified all samples as having poor outcomes. Subsequent runs were adapted to learn faster — by increasing the ILR —, and train for longer — by increasing `patience` setting. Despite these bad results, the model achieved cross-validated $\overline{AUC} = 0.69 \pm 0.12$, but a $\overline{F1\_weighted} = 0.62 \pm 0.17$, since the

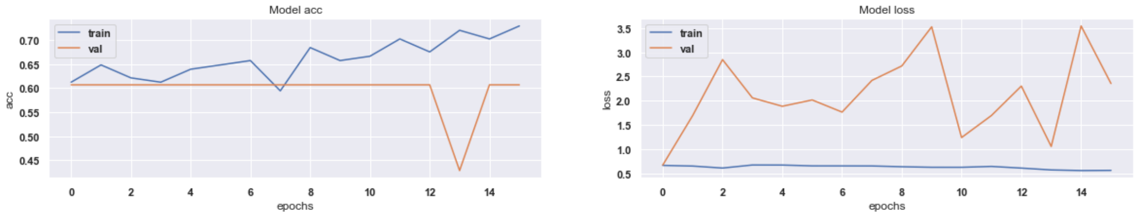


FIGURE 6.6: Training and validation accuracy and loss evolution through 3D CNN base model training with no augmentation and patience is set to 15 epochs.

regularizing effect of F1-weighted during model selection was not present in CNN individual training sessions, as seen in tables 6.16, A.19, and A.20, but test accuracy results just match class proportions, as seen in table 6.17.

TABLE 6.16: CNNs cross-validated metrics comparison.

Model	Val. Acc.	Std.	Val. Bal.Acc.	Std.	Val. AUC	Std.	Val. F1-Weighted	Std.
3D CNN (NO AUGs, Patience = 15)	0.69	0.12	0.63	0.15	0.75	0.20	0.62	0.17
3D CNN (NO AUGs, Patience = 30)	0.76	0.14	0.73	0.16	0.87	0.13	0.73	0.17
3D CNN (NO AUGs, Patience = 100, ILR=0.01)	0.57	0.12	0.51	0.12	0.60	0.17	0.51	0.12
3D CNN (NO AUGs, Patience = 100, ILR=0.001)	0.80	0.13	0.76	0.16	0.92	0.11	0.77	0.17
3D CNN (NO AUGs, Patience = 100, ILR=0.0001)	0.89	0.09	0.88	0.11	0.96	0.07	0.89	0.10
3D CNN (NO AUGs, Patience = 200, ILR=0.0001)	0.91	0.09	0.91	0.10	0.97	0.07	0.91	0.09

Classification reports from models with **patience increased to 30, 100 and 200 epochs show no meaningful improvement in the model's generalization results**, although models trained for longer started making more balanced predictions. Increasing patience from 100 to 200 improves validation metrics marginally, but it does not achieve better generalizability, as seen in table 6.17 and 6.18, **suggesting the models are just overfitting past a certain point**. These test results were observed also with varying ILRs ranging from 0.0001 to 0.01, although cross-validation shows learning during training with low ILRs is possible, but not with higher values such as 0.01, as seen in plots 6.7, 6.8, and 6.9.

TABLE 6.17: Base 3D CNN model without augmentations and patience set to 15, on the left side, and patience set to 30 classification reports, on the right side.

	precision	recall	f1-score	support
0	0.00	0.00	0.00	14
1	0.60	1.00	0.75	21
accuracy			0.60	35
macro avg	0.30	0.50	0.37	35
weighted avg	0.36	0.60	0.45	35

	precision	**recall**	**f1-score**	**support**
0	0.50	0.14	0.22	14
1	0.61	0.90	0.73	21
accuracy			0.60	35
macro avg	0.56	0.52	0.48	35
weighted avg	0.57	0.60	0.53	35

TABLE 6.18: Base 3D CNN model without augmentations and patience set to 100, on the left side, and patience set to 200 classification reports, on the right side.

	precision	recall	f1-score	support		precision	recall	f1-score	support
0	0.67	0.14	0.24	14	**0**	0.33	0.21	0.26	14
1	0.62	0.95	0.75	21	**1**	0.58	0.71	0.64	21
accuracy			0.63	35	**accuracy**			0.51	35
macro avg	0.65	0.55	0.50	35	**macro avg**	0.46	0.46	0.45	35
weighted avg	0.64	0.63	0.55	35	**weighted avg**	0.48	0.51	0.49	35

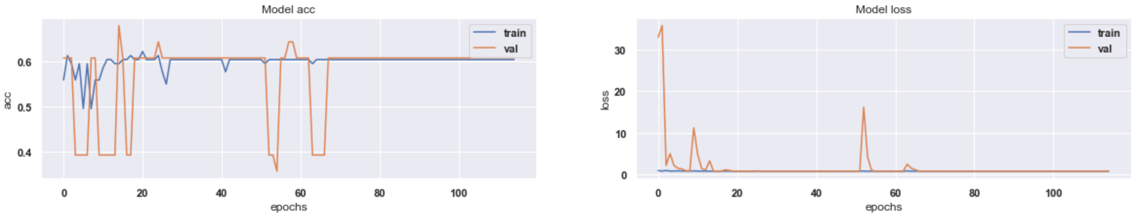


FIGURE 6.7: Training and validation accuracy and loss evolution through 3D CNN base model training with no augmentation and patience is set to 100 epochs and initial learning rate set to 0.01.

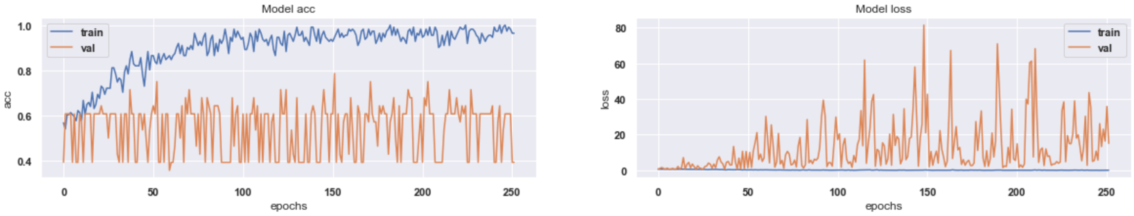


FIGURE 6.8: Training and validation accuracy and loss evolution through 3D CNN base model training with no augmentation and patience is set to 100 epochs and initial learning rate set to 0.001.

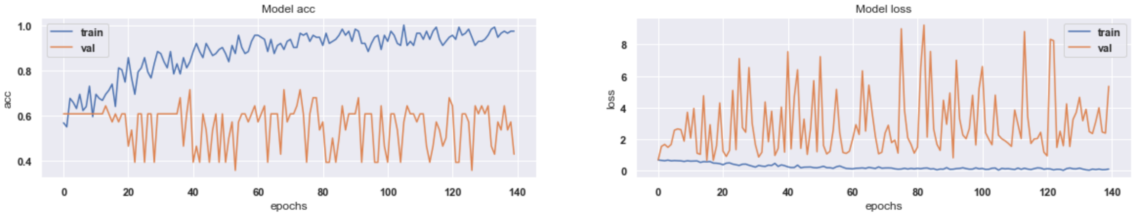


FIGURE 6.9: Training and validation accuracy and loss evolution through 3D CNN base model training with no augmentation and patience is set to 100 epochs and initial learning rate set to 0.0001.

6.4.3 3D CNNs Data Augmentation

In a first approach, ILRs were tested without cross-validation to see how the training loss progressed. Figures 6.7, 6.8 and 6.9, show that defining $ILR = 0.0001$ enables significantly lower training loss values than other magnitudes values, but it also achieves visible improvements in training starting on earlier epochs, as seen comparing 6.8 with 6.9. This seems to indicate the **extra variability added by the augmentations requires slower learning** for the network to be able to adjust to this dynamically produced ds., not overajusting to initial features. Considering these results, further experiments used $ILR = 0.0001$.

TABLE 6.19: CNNs cross-validated metrics comparison.

Model	Val. Acc.	Std.	Val. Bal.Acc.	Std.	Val. AUC	Std.	Val. F1-Weighted	Std.
3D CNN (NO AUGs, Patience = 200, ILR=0.0001)	0.91	0.09	0.91	0.10	0.97	0.07	0.91	0.09
3D CNN (AUGs Rot., Patience = 200, ILR=0.0001)	0.73	0.13	0.68	0.15	0.82	0.17	0.69	0.17
3D CNN (AUGs Vol., Patience = 200, ILR=0.0001)	0.71	0.08	0.66	0.09	0.80	0.10	0.67	0.10

As seen in tables 6.19, the model with simple rotations and ILR $= 0.0001$ was unable to produce significantly better results than the worse model measured, a 3D CNN with no data augmentation and $patience = 30$, inducing that when using data augmentation, training epochs should be increased, so the model maintains its validation performance and increases its generalization capacity, but increasing patience to 200 epochs produced the same test results, so the same poor generalization results were achieved, with cross-validation metrics only improving slightly. Adding **more complex augmentations** with the `volumentations` package — and another increase in sample variability — **did not improve results**, neither during validation — where accuracy and F1-weighted scores were equivalent to a null model —, nor on test scores where the model was unable to improve upon any of the previous models.

6.4.4 3D CNNs Network Architecture Search (NAS) Results

The test run with HyperBand lasted for 23h 12m 32s before crashing without possibility to continue from that point. During that period 307 trials were conducted in the first Hyperband phase, recurring three epochs each, and finding a model that achieved 0.67 validation loss. NAS suggested hps. are shown in table 6.20.

The model suggested by HyperBand has shown no signs of learning during its training, as shown by figure 6.10.

TABLE 6.20: 3D CNN architecture hyperparameters suggested by Hyperband.

Hyperparameter	learning_rate	decay_steps	decay_rate	number_of_layers	filter_s_1	kernel_s	activation	type_pooling	maxp_1
Best Value So Far	0.00011319	1.52e+05	0.93	5	64	3	selu	avg	2
Hyperparameter	**filter_s_2**	**maxp_2**	**filter_s_3**	**maxp_3**	**filter_s_4**	**maxp_4**	**dense_u_5**	**dropout**	
Best Value So Far	128	2	512	2	256	2	256	0.10	

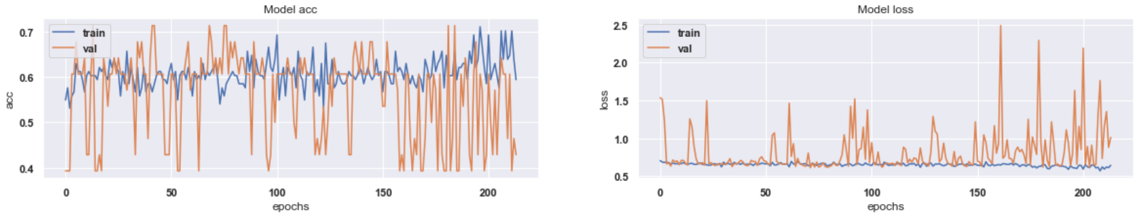


FIGURE 6.10: Training and validation accuracy and loss evolution through HyperBand parametrized 3D CNN training with full augmentation set, set to $patience = 200$ and $ILR = 0.0001$.

Considering the same training epochs and data augmentations as the models analysed in the previous section, the model suggested by HyperBand has decreased AUC performance, with only $\overline{AUC} = 0.76$, as seen in table 6.16, and all other metrics with similarly bad results. The model was unable to distinguish itself statistically from other 3D CNNs when comparing F1-weighted scores, as seen in figure 6.13.

TABLE 6.21: 3D CNN architecture hyperparameters suggested by Bayesian Optimization.

Hyperparameter	learning_rate	decay_steps	decay_rate	number_of_layers	filter_s_1	kernel_s	activation	type_pooling
Best Value So Far	3.60e-06	17770	0.93	2	64	5	relu	max
Hyperparameter	**maxp_1**	**filter_s_2**	**maxp_2**	**filter_s_3**	**maxp_3**	**dense_u_5**	**dropout**	
Best Value So Far	2	256	2	None	None	1024	0.20	

The test run with **Bayesian Optimization** lasted for 22h 59m 19s before crashing without possibility to continue from that point. During that period 7 full trials were conducted, recurring approximately 600 epochs each, and finding a model that achieved 0.73 validation weighted accuracy. NAS suggested hps. are shown in table 6.21, not much can be learnt from them considering its **even lower metrics compared to the baseline**, as shown in table 6.22.

TABLE 6.22: CNNs cross-validated metrics comparison.

Model	Val. Acc.	Std.	Val. Bal.Acc.	Std.	Val. AUC	Std.	Val. F1-Weighted	Std.
3D CNN (AUGs Vol., Patience = 200, ILR=0.0001)	0.71	0.08	0.66	0.09	0.80	0.10	0.67	0.10
3D CNN (AUGs Vol., Patience = 200, Hyperband)	0.68	0.11	0.63	0.12	0.76	0.13	0.65	0.12
3D CNN (AUGs Vol., Patience = 200, BayesOpt)	0.65	0.08	0.61	0.08	0.69	0.11	0.62	0.09

6.4.5 3D CNNs with Transfer Learning

The best 3D CNN model architecture so far was the template 3D CNN architecture when trained for 200 epochs with a $ILR = 0.0001$, so this was the architecture used for transfer learning. The tests were done comparing to the model trained with $patience = 100$ due to time constraints and the small difference in metrics this incurs, considering overfitting was detected much earlier on the training process.

TABLE 6.23: CNNs cross-validated metrics comparison.

Model	Val. Acc.	Std.	Val. Bal.Acc.	Std.	Val. AUC	Std.	Val. F1-Weighted	Std.
3D CNN (NO AUGs, Patience = 100, ILR=0.001)	0.80	0.13	0.76	0.16	0.92	0.11	0.77	0.17
3D CNN (NO AUGs, Patience = 100, ILR=0.0001)	0.89	0.09	0.88	0.11	0.96	0.07	0.89	0.10
3D CNN (NO AUGs, Pat.=100, BasicTransferL)	0.94	0.09	0.93	0.10	0.97	0.05	0.94	0.09
3D CNN (No AUGs, Pat.=100, FullTranferL)	0.93	0.08	0.92	0.10	0.94	0.08	0.92	0.11

Basic transfer learning via simple fine-tuning, BasicTransferL, improved all metrics over the best 3D CNN architecture so far, while Training the final dense layers separately before fine-tuning, **FullTranferL**, made all metrics decreased compared to simple fine-tuning, as seen in table 6.23. Considering Nemenyi's critical distances based on AUCs, models with transfer learning differentiate themselves from most models, but they are not stat. sig. different among themselves and the model without transfer learning, as seen in 6.13.

TABLE 6.24: 3D CNN trained with basic transfer learning, on the left side, and with full transfer learning, on the right side.

	precision	recall	f1-score	support
0.0	0.71	0.36	0.48	14
1.0	0.68	0.90	0.78	21
accuracy			0.69	35
macro avg	0.70	0.63	0.63	35
weighted avg	0.69	0.69	0.66	35

	precision	recall	f1-score	support
0.0	0.60	0.21	0.32	14
1.0	0.63	0.90	0.75	21
accuracy			0.63	35
macro avg	0.62	0.56	0.53	35
weighted avg	0.62	0.63	0.57	35

Despite excellent validation results for both methods, the models achieved **values below CIs on the test set, although BasicTransferL still fared better**, as seen on 6.24. As expected, transfer learning made training faster, as denoted by achieving top training accuracy markedly earlier, although starting to overfit earlier, as seen by comparing training and validation curves from transfer learning models 6.11 and 6.12 with the model without transfer learning, in figure 6.9.

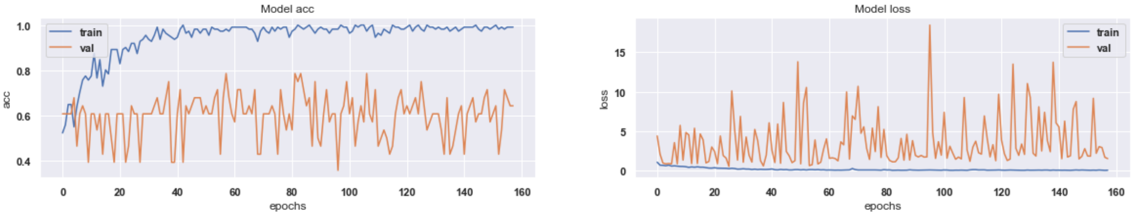


FIGURE 6.11: Training and validation accuracy and loss evolution through 3D CNN basic fine-tuning, BasicTransferL.

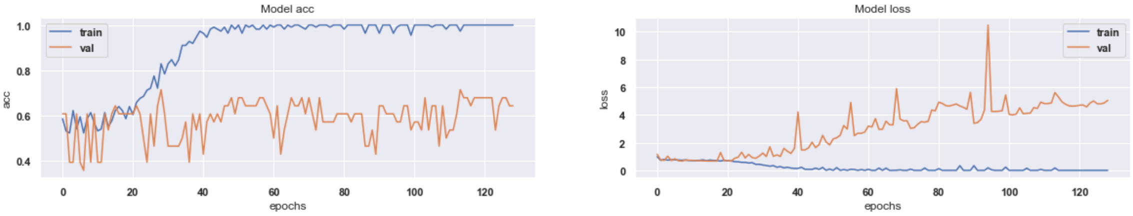


FIGURE 6.12: Training and validation accuracy and loss evolution through 3D CNN with fine-tuning after dense layer retraining, FullTransferL.

6.5 Mixed Models

6.5.1 Clinical data and Selected Biomarkers at Admission (ClinBiom0h) Model Selection

ClinBiom0h included all variables with p-values in bold present in tables A.25 and A.26, summing up 29 features. ClinBiom0h models achieved better results than Biom0h models, but no better than the best base clinical models, with all **ClinBiom0h** models achieving $\overline{AUC} < 0.72$. Models trained on augmented data performed better than their non-augmented counterparts with $\overline{AUC}$ increases ranging from 2% to 6%. Once more, **XGBM** achieved the best result $\overline{AUC} = 0.72 \pm 0.16$ with **LGBM** following closely $\overline{AUC} = 0.72 \pm 0.17$ and showing more consistent results in augmented and non-augmented data.

Despite measured metrics differences, w.r.t. AUC scores, only four models were statistically different from the best clinical models: both SVC models without kernel transformation, LR on non-augmented data and XGBM on non-augmented data, as shown in figure A.22. The poor results of SVC on this ds. are due to underfitting, steaming probably because convergence was not achieved within the set maximum of iteration, since SVC on non-augmented data classified all cases as having poor outcomes.

Figure 6.14 shows ClinBiom0h models' ROCs on non-augmented and augmented test data respectively. AUCs on the test set show both

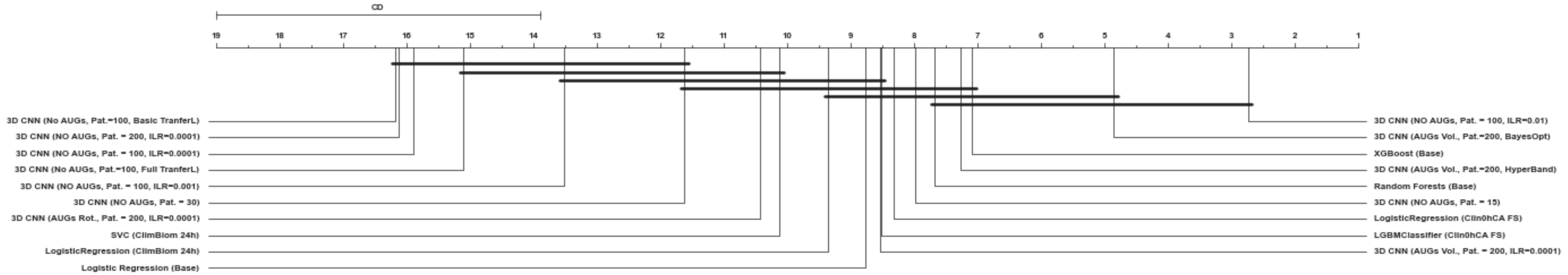


FIGURE 6.13: Nemenyi's plot comparing 3D CNN models through AUC scores.

TABLE 6.25: Table comparing best ClinBiom0h tabular models against augmented models and clinical dataset models.

	meanrank	mean	std	ci_lower	ci_upper	effect_size	magnitude
SVC (ClinBiom 0h)	3.79	0.57	0.16	0.49	0.65	0.00	negligible
LogisticRegression (ClinBiom 0h)	4.44	0.60	0.12	0.54	0.66	-0.18	negligible
SVC (ClinBiom 0h AUG)	5.36	0.62	0.21	0.51	0.71	-0.22	small
XGBClassifier (ClinBiom 0h)	5.77	0.65	0.14	0.59	0.72	-0.54	medium
LinearSVC (ClinBiom 0h)	6.02	0.67	0.17	0.59	0.75	-0.57	medium
LogisticRegression (ClinBiom 0h AUG)	6.03	0.65	0.23	0.54	0.75	-0.37	small
LinearSVC (ClinBiom 0h AUG)	6.64	0.69	0.24	0.58	0.80	-0.56	medium
LGBMClassifier (ClinBiom 0h)	6.78	0.68	0.15	0.61	0.75	-0.70	medium
LGBMClassifier (ClinBiom 0h AUG)	6.78	0.71	0.18	0.63	0.79	-0.83	large
XGBClassifier (ClinBiom 0h AUG)	7.16	0.72	0.16	0.65	0.80	-0.94	large
XGBoost (Base)	7.86	0.78	0.11	0.73	0.82	-1.50	large
Random Forests (Base)	7.93	0.78	0.11	0.73	0.83	-1.50	large
Logistic Regression (Base)	8.50	0.79	0.14	0.73	0.85	-1.45	large

LGBM and XGBM generalize well, with AUCs above CIs, LR has shown consistent performance with the cross-validated scores and **SVM-based models underperformed on the test set**, with SVC on non-augmented data showing the same AUC as the null model, while the one trained on augmented data showing a performance far worse than the null model.

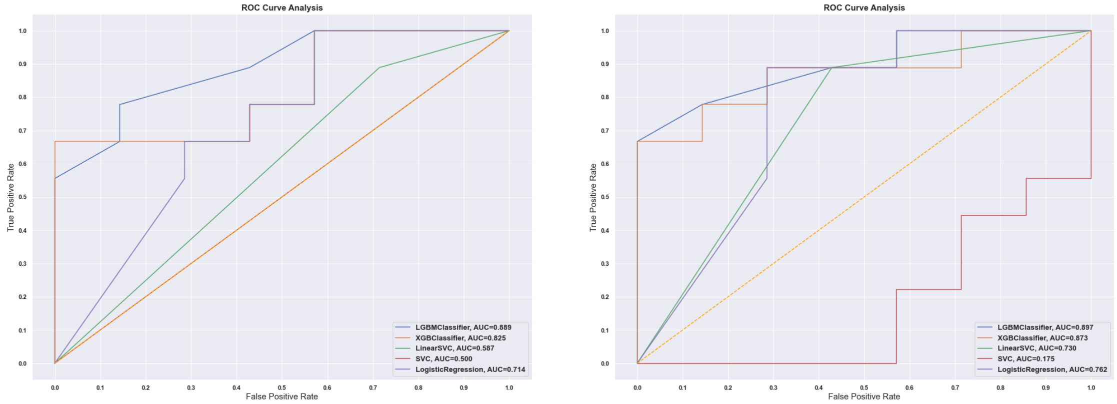


FIGURE 6.14: ROCs comparing ClinBiom0h models without augmentation, on the left, and with augmentation on the right.

6.5.2 Clinical data and Selected Biom24h (ClinBiom24h) Model Selection

ClinBiom24h included all variables with p-values in bold present in tables A.25 and A.27, summing up 43 features. Table 6.26 details how models performed based on their AUC scores. **ClinBiom24h models also achieved better results than models with Biom24h, and, were also better than the best base clinical models**, unlike ClinBiom0h. **The best ClinBiom24h model is a SVM with kernel transformation model on non-augmented**

data, which achieved $med(AUC) = 0.88$, nearly a 10% improvement over the best clinical model. This group did not show clear improvements with data augmentation and only LGBM and Linear SVC have shown noticeable improvements in their median AUC values. Metrics differences among top performing models were small, with CIs strongly overlapping in the maximum upper CI being achieved by seven different models. These results suggest there were too many noisy features in the ClinBiom0h for SVC with kernel transformations to properly fit.

TABLE 6.26: Table comparing the best ClinBiom24h tabular models against augmented models and clinical dataset models.

	meanrank	median	mad	ci_lower	ci_upper	effect_size	magnitude
Random Forests (Base)	6.10	0.77	0.08	0.71	0.86	0.00	negligible
XGBClassifier (ClinBiom 24h)	6.19	0.79	0.08	0.71	0.88	-0.16	negligible
XGBoost (Base)	6.23	0.80	0.08	0.71	0.86	-0.22	small
LGBMClassifier (ClinBiom 24h)	6.39	0.81	0.10	0.74	0.88	-0.30	small
XGBClassifier (ClinBiom 24h AUG)	6.46	0.79	0.12	0.71	0.92	-0.13	negligible
LogisticRegression (ClinBiom 24h)	7.04	0.82	0.10	0.71	0.92	-0.33	small
Logistic Regression (Base)	7.20	0.80	0.08	0.74	0.88	-0.22	small
LogisticRegression (ClinBiom 24h AUG)	7.24	0.81	0.10	0.75	0.92	-0.29	small
LinearSVC (ClinBiom 24h AUG)	7.31	0.81	0.11	0.75	0.92	-0.28	small
SVC (ClinBiom 24h AUG)	7.51	0.83	0.08	0.75	0.88	-0.49	small
LinearSVC (ClinBiom 24h)	7.58	0.83	0.08	0.75	0.92	-0.49	small
LGBMClassifier (ClinBiom 24h AUG)	7.61	0.83	0.08	0.75	0.92	-0.49	small
SVC (ClinBiom 24h)	8.14	0.88	0.08	0.79	0.92	-0.83	large

These new models were stat. sig. different from the two base models, base RF and XGBM. Although metrics have improved considerably, there is still no stat. sig. from base **LR** the overall best clinical model, as shown in figure A.23.

Figures 6.15 and 6.16 show ClinBiom24h models' ROCs on non-augmented and augmented test data respectively. All models but Linear SVC trained without data augmentation have dropped AUC scores on the test set, but only **SVC with kernel transformation** AUC shows clear signs of overfitting, considering its value dropped below its validation CI, $AUC_{SVC\_val} = [0.79 - 0.92]$ and $AUC_{SVC\_AUC\_val} = [0.75 - 0.88]$, with $AUC_{SVC\_test} = 0.74$ and $AUC_{SVC\_AUG\_test} = 0.70$.

6.5.3 Clinical models with Hemispheric Contrast (HC) Imaging Biomarker (IM) (ClinCA0h) and ClinCA0h with Feature Selection (ClinCA0h FS)

Most clinical models benefited from absolute HC addition but statistic comparison as shown no improvement over the best baseline LR, figure A.30.

While conducting HP analysis, figure 6.4, **absolute HC appeared as the fifth most important variable** by its ANOVA F-value, **preceded only by mRS before event, NIHSS at admission to hospital, history of heart failure**, and **if the AIS was detected while**

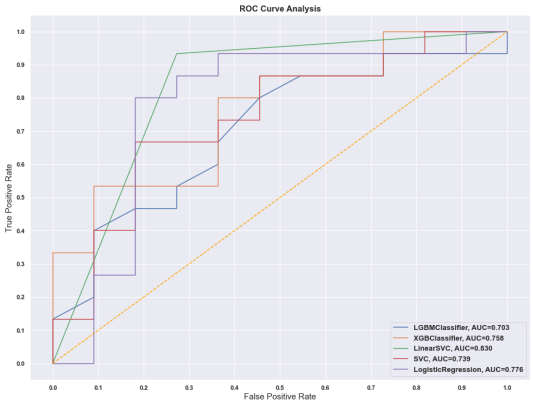


FIGURE 6.15: ROCs comparing ClinBiom24h models.

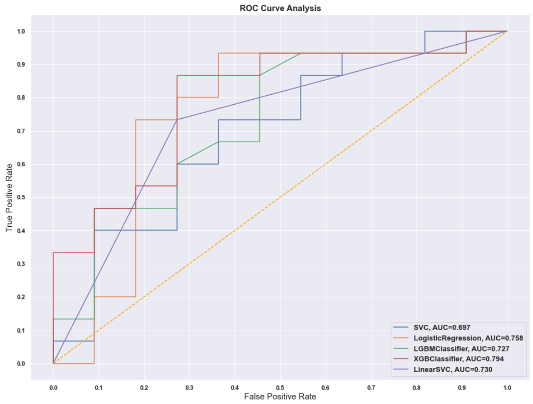


FIGURE 6.16: ROCs comparing ClinBiom24h models trained on augmented data.

waking up. The best median for all models cross-validate AUCs was achieved adding previous intake of anti-hypertensors and knowing if AIS' aetiology was prothrombotic, also allowing LGBM to achieve $med(AUC) = 0.86 \pm 0.08$. The highest LR AUC was achieved with 11 variables, $med(AUC) = 0.83 \pm 0.08$, adding **patient's age, history of chronic renal disease, AIS and previous intake of medication for diabetes**.

F1-score regularizing effect while conducting GS makes the model continue to peak at $\overline{AUC} = 0.80$, with no model surpassing LR on the clinical ds. after renewed GS. As such, the original FS model is selected due to its better final metrics, table A.17, although better metrics are not stat. sig. different from the best previous models A.20. It should be noted both SVCs have improved the most with FS, showing they were assigning too much weight to irrelevant features.

6.5.4 ClinBiom24h with Feature Selection (FS) and Hemipheric Contrast (HC)

ClinBiom24h FS did not improve central tendency nor dispersion metrics over the base ClinBiom24h, $med(AUC_{ClinBiom24hFS}) = 0.87 \pm 0.09$, **although this LGBM achieves it with only three variables: mRS before event, NIHSS and patient's age**. Adding HC at admission (**ClinBiom24hCA0h FS**) does not improve the best model either, since absolute HC is the seventh variable to be selected; therefore creating a model with more requirements, and because metrics further decrease, $med(AUC_{ClinBiom24hCA0hFS}) = 0.83$, as can be seen in table A.18. Despite lower metrics, these models are not stat. sig. from **the best model found so far for post-thrombectomy data, an SVC with kernel transformation trained on ClinBiom24h**, as seen in figure A.21.

Chapter 7

Discussion

7.1 Clinical Models

The variables mostly correlated to outcome were the clinical scores for patients functional evaluation: mRS before event, and NIHSS at admission. These make sense since incoming patients with previous AIS damage have a worse baseline to recover from, and any subsequent damage stacks-up. While conducting FS, it was possible to verify that FWER corrections were too aggressive on these ds. since models continued to improve peak scores up to 16 variables — dummy classes included —, which suggests extra information from those variables is relevant to the outcome, and, therefore, correlated in some way, with ANOVA F-values sorting variables in a way similar to unadjusted p-values. Although neurology's motto is "Time is brain", and the ds. contained detailed timing information from incident to thrombectomy, time differences were not successful indicators, mostly due to two factors: patient's specific tolerance to ischaemia differences, and the bias created by imputing event times for cases with symptoms while waking up. A related boolean variable, AIS at wake-up (`WkUp`), was among the most relevant variables.

Clinical models provide a baseline for modelling, considering the ds. is composed by information acquired once the patient arrives to hospital, and much of it can be automatically obtained through the patient's health records. Baseline models trained on this ds. had good predictive capacity, with **LR dominating the first modelling phase**. This can be due to the high bias that this modelling strategy has, conforming with linear DBs, ℓ_1 regularization technique, and the ability to deal moderately well with collinear variables. **Initial FS was minimal** to allow evaluating modelling strategies on their ability to deal with collinearity, with models that integrate good FS methods — such as tree-based models — to differentiate positively. This strategy was sub-optimal, considering **HP improved metrics while reducing model complexity and training time**. Multicollinearity is a particular problem for some modelling strategies, particularly **NNs**, **LDA** and **SVM** — which are sensitive to this phenomenon —, and those strategies could have had better results, if multicollinear variables were handled more strictly beforehand.

Minority class balancing and data augmentation were not effective on the best model trained on this ds., the hyperparametrized LR, but data augmentation has helped **XGBM** and **LGBM** come closer to the original **LR**, which suggests the regularizing effect of data augmentation was more effective in models with high variance, probably because they were overfitting noisy features.

7.2 Automated Machine Learning (AutoML)

The limited test with **AutoML** has shown that within the same computational search time, a **manually constructed search space is more effective at finding top performing models**, faster classifiers are created, and they are generally easier to explain. However, **the experiment conducted did not factor in the time spent on repeated experiments, tweaking search spaces, nor development time** orders of magnitude greater than AutoSkLearn runtime. Should those be accounted for, it is possible that better results were found with AutoML. The experiment extended to AutoPyTorch, and H2O, but no good way was found to obtain individual cross-validation scores so formal statistical model comparison could be done, — the reason these experiments not being detailed —, but, preliminary tests have returned promising results with $\overline{AUC_{AutoPyTorch}} = 0.78$. **AutoML was shown to be a good way to produce robust models** when developer time can be exchanged by extra computing resources and processing time; models interpretability and explainability is not key; and inference time and performance can be suboptimal. The research on AutoML models enabled the discovery of more efficient models selection strategies — later used on **CNNs** NAS —, ensemble methods inner workings and importance — later used on multi-model voting systems —, as well as the **importance of the exploration-exploitation dilemma** — a concept transversal to all research, and an AutoML pillar. Should the objective of this thesis not be to produce the best models while developing a deeper understanding on ML and CV with focus on DL methods applied to neurosciences, dedicating more computing resources and research time to AutoML methods could be a sound strategy to find top models. That research would shift to spending more time trying to **explore and explain AutoML** created models than to produce them.

7.3 Biomarker Models

Statistical analysis on **ClinBiom0h** has difficulty predicting modelling outcomes with this ds., which was confirmed by the **poor results on this ds.**. This suggests the **main biochemical characteristics used in routine clinical practice have little to no bearing in the patients' outcome**, nor are conducted for that purpose. However, these biochemical analysis were done to diagnose underlying patient's conditions for anaesthesiology guidance, not for prospective values for modelling AIS outcomes. Notwithstanding, given how important these are to the overall patient's health, analysing them as fitting within the normal range for the patients sex and age, or use them as a reference to calculate the variation with follow-ups, could bring engineered features with better predictive potential. This analysis was not yet done, due to time constrains.

Models trained only on pre-thrombectomy data missed some important variables, considering **models based on the same biomarkers at different phases had very different results**. **Biom24h-based models matched the performance of the base clinical models** with AUC scores around 0.80, while **Biom0h** had less predictive ability than a null model. This indicates that some **surgery related information is important for outcomes prediction**. Considering the clinical analysis done shortly after surgery, it is expected that **inflammation and apoptosis-related biomarkers collected have some degree of correlation with the reperfusion achieved** and how the brain tissues are reacting to treatment; therefore, acting as a collinear variable to surgical success, as was shown while modelling with such biomarkers. This clinical success is related to the level of reperfusion achieved on the affected artery, which is often evaluated with the **modified Thrombolysis in Cerebral Infarction (mTICI)**, and, in fact, several studies show its importance in outcome modelling [182, 183].

Biom24h had more variables with relation to the outcome, especially FBC information. AST, as an aminotransferase is usually associated to ALA in liver damage, but since AST occurs in other organs, **high concentrations may also be indicative of myocardial infarction** — or causes of extensive bodily damage, such as extensive burns, acute pancreatitis or acute haemolytic anaemia —, a possible cause of brain infarction without localized brain ischaemia [184]. **Prothrombin time** and **International Normalized Ratio (INR)**

are coagulation related biomarkers, and are usually tested before admission [184]. Due to this they have a direct relation with thrombosis in case of below reference INR values, one common form of IS, or haemorrhagic strokes in case of exceedingly high INR values [185]. As previously stated, these values also have **no bearing in patients outcome when measured before the thrombectomy** takes place since they possibly only reveal the possibility for bodily damage to occur, and **after thrombectomy coagulation is regulated by prescribed medication; therefore not affecting the patients' outcome**. All these biomarkers are relevant clinically, but, considering this study's subjects advanced age, the range of possible values affecting these biomarkers is too narrow, — $IQR = [2 - 8]$, their variation within the selected cohort may be too narrow to produce perceivable effects on outcomes.

Biomarkers modelling was constrained by two main aspects. Cases arriving from other hospitals did not have available data, reducing considerably this subset of data. With **only 79 cases** to model and 26 features, **overfitting to noisy features** is more relevant. As in the clinical models, **after reducing the number of features to 10, better results were achieved** in all models.

While **studying HP** with AUC scores, all models improved significantly in raw metrics but only **Clin0h LR improved stat. sig. over all other models**. It should be noted the regularizing effect of using a different search space objective, F1-weighted score, made the former top hps. not to be selected because they would have lower total F1-score despite maximizing AUC. This made most models maintain or lower their AUC scores after GS. Trying to optimize models for a single objective — as it is common in data science competition, such as Kaggle and ISLES — can prove useful to maximize scores across the board, so the hps. used on HP were used to retrain models on the optimal number of features and considered the best.

Several features selected are **mathematically coupled** [186] — e.g., **percentages and concentrations** of the same biomarker, NLR and features relating to lymphocytes and neutrophils — which shows that the automated FS using ANOVA F-values accounts poorly for this type of variable interaction that causes multicollinearity problems. These problems are more easily solved when selecting variables manually, so, **combining automated FS with manual tweaking may bear better results**. Also, **analysing which variables are responsible for increases in cross-validated peak performance in HP analysis and selecting only those for new models may prove to be a sound strategy** for model improvement.

7.4 Imaging Models

Imaging models achieved the best overall results on validation. Transfer learning was effective on improving model performance, and the models with **the best validation and test results among 3D CNNs were achieved by fine-tuning a model without data augmentation**, and doing it in considerably less epochs than the one used by most other advanced models tested, even when considering retraining times. The model with basic fine-tuning, **BasicTransferL**, can be considered the best model overall. However, **further cross-validation of this model on unseen data is recommended**, considering the disparity between cross-validated results and test results. Test results are a single point metric, so they are greatly influenced by stochastic factors, and comparing directly validation results from DL methods with validation results from other ML strategies is tricky, because the patience mechanism cherry-picks the best epoch within numerous epochs and metrics evolution variation should also be taken into account.

Image data augmentation techniques can theoretically improve generalization results, but within the maximum set training epochs, this was not observed, neither in validation nor in testing. **Samples registration is meant to reduce confounding factors and variability to be modelled**, so it is likely that augmentations have to be better designed for registered data, especially because overfitting was suggested by training and validation loss plots, the difference between different lesion severities is very subtle, and successful augmentations documented on other classifications tasks use larger models trained for longer. **Despite having used longer training sessions than many experiments without data augmentation, it has not been enough to account for the extra variability**. It should also be noted that mirroring images in the axial plane causes hemispheres to be interpreted as being switched and lesion impacts are different depending on the hemisphere affected, which may result in distinct outcome categorization [187]. Since transfer learning was done after data augmentation analysis, showing much faster convergence — but also diminishing returns on increased training time —, it would be interesting to also test data augmentation on the transfer learning pipeline and assess generalization results.

Considering the low number of epochs in NAS trials with HyperBand and that various manually hyperparametrized models have only shown learning with dozens to a hundred epochs, **it is unlikely that Hyperband was able to select models appropriately**.

With so few epochs, the assumption the models' performance at early epochs reflects the overall relative model performance may be flawed, especially considering the amount of input parameters and model's expected complexity. Despite these limitations, **Hyperband suggested a deep model similar to the base model**, with five hidden layers but changed some activation unity **replaced ReLU activation by SeLU, and average pooling was selected instead of maximum pooling**, so testing these parameters on the base architecture may improve 3D CNN results. **Bayesian Optimization was more fitting to the task since it fully tests each architecture, but the low number of trials has been unable to find a distinctly better architecture**. Using this NAS strategy requires a significant number of trails to find improvements over the base architecture, which was already chosen for dss. such as the one used.

7.5 Mixed Models

Tabular mixed models had varying results. **The best ClinBiom0h model** — a XGBM trained on augmented data — was not statistically different from the baseline models, but it has shown considerably lower metrics than all the best clinical only models, suggesting that Biom0h added to the models were mostly multicollinear or noisy features, and reinforcing the notion the biochemical state of the patient before surgery has little influence on its outcome, since it reflects the critical state that took the patient to the hospital. On the other hand, **several ClinBiom24h models outperformed all clinical only models but the FS'ed ones**, demonstrating that despite most biomarkers being the same as in admission, **the 'updated' information is relevant to determine patients' recovery**.

Model-dependent results of data augmentation on tabular data suggest that either the chosen data augmentation algorithm was suboptimal — despite the study done on the subject —, or for mildly imbalanced data, the regularizing effects of this strategy achieve meagre improvements that get diluted by other more significant modelling adjustments. It was noticeable that the best models during validation — SVM with kernel transformation and LGBM — underperformed during testing, while modelling strategies with more modest results — linear SVM and LR — performed consistently with validation CIs, suggesting increased robustness on models with high DB bias.

The FS for mixed models used statistics for initial data reduction before the merge, so the most likely noisy features were not included on mixed dss.. This is likely the cause for posterior FS while analysing the HP not being able to produce relevant improvements.

On the **Clin0hCA mixed model**, it was shown that adding **the IB enables models to maintain equivalent performance with fewer features**. If this IB is calculated by the CT scanner software without extra delays to emergency procedure, this may increase operational performance since fewer variables have to be confirmed by the operator, with **Clin0hCA** being less prone to extreme results due to noise on the extra features.

7.6 Transferability and generalization

The **analysed cohort** refers to very specific subjects with specific characteristics: **a senior population, with little to no previous AIS sequels, being affected by an AIS in the cerebral middle artery with infarction in the carotid territory**. Considering the cohort represents the general statistics for patients being conducted for thrombectomy, the models trained on these dss. will be **useful for the vast majority of incoming patients**, but care should be taken to not use the models in cases considerably different from this cohort parameters.

Test AUCs have been analysed in each section, and **they mostly fit within validation CIs**, but it was noticeable that strict FS had a positive impact in test scores, making most models increase test scores and achieve values closer to validation mean AUCs. This reaffirms FS regularizing effect, and its usefulness in AIS modelling. **The most important tabular features selected for clinical models** are similar to the ones used by state-of-the-art AIS outcomes models: **age** — used in all well-known models —, pre-stroke functional status (**mRS before event**) — used by Dragon, FSV, PLAN and SOAR —, co-morbidities, such as previous **renal chronic disease, heart failure, previous AIS** — used by iScore, PLAN, SPI, S-TPI and THRIVE —, stroke severity (**NIHSS at admission**) — used by most well-known AIS predictive models but SOAR and SPI. Imaging findings, such as **absolute HC** were not known to be specifically used by any model, but common tools like e-ASPECTS use imaging findings, as well as, models such as Dragon, SNARL, and S-TPI [188].

No publicly available models were applied to these dss. and comparing single point measures taken on different dss. from references does not provide an accurate assessment, but if AUCs' CIs were calculated, they could be compared statistically [189]. Regardless, the best 3D CNN results of this thesis compare positively with the values encountered in referenced studies, but not its test results. Once again, the comparison is limited, since different dss. and measuring methodologies were used in each work.

Chapter 8

Conclusions

8.1 General conclusions

During this work several modelling strategies were used. Various ML methods to select and improve model performance were applied, studying the various parts of the process to optimize results as much as possible. DL was used mostly for CV tasks, since on tabular data NN have shown severe overfitting, mostly because multicollinearity is poorly handled by this algorithms group.

The overall best model found was the 3D CNN with basic fine-tuning on MosMedData, which achieved $\overline{AUC_{BasicTransferL}} = 0.97 \pm 0.05$ on validation, although its generalization capacity may be worse than other models, considering all 3D CNN show strong signs of overfitting in the training/validation plots, and this model only achieved $\overline{AUC_{BasicTransferL}} = 0.58$ and $\overline{F1\text{-}weighted_{BasicTransferL}} = 0.66$ on the test set.

For each admission phase and data available, one model is proposed. **On arrival, the hyperparametrized LR trained on the strictly selected features clinical ds. is recommended**. The moderate AUC values achieved by this model suggest caution since AUC scores on the order of 80% is only considered good in many low consequence setting [190]. The medical field is no such area, where $AUC > 0.95$ is expected for models to be considered for use [191]. It should however be noted using a metric for model selection different from the one used in validation and test evaluation makes measurements more conservative. F1-weighted score assures the models selected have a good compromise between precision and recall, with slightly more relevance given to the positive class, i.e., the one assuming the decision to not perform surgical treatment with the prediction of a poor outcome. Should neuroimaging data be available on decision time, the 3D CNN with simple fine-tuning is recommended, considering its validation results. However, comparing results with the previous model is advisable, since direct metrics comparison among 3D CNN results and other ML methods is not straight forward. If hemispheric contrast imaging biomarker is integrated in neuroimaging software, recurring to **the proposed ClinCA0h FS model for this second pre-surgical phase and its results compared with the ones from the 3D CNN**. These models might be further improved by automatizing the comparison by using a voting system that selects the highest probability between the best 3D CNN and one of other above-mentioned models.

For the follow-up phase, **the pre-treatment 3D CNN results should be compared to the SVM with kernel transformation trained on non-augmented ClinBiom24h mixed model**. The mixed SVM model alone achieved $med(AUC) = 0.88 \pm 0.08$, a considerably higher median than all clinical models — even the one improved by HP analysis. Despite better metrics, they have greater variability, and therefore are **not stat. sig. different from the best model trained on Clin0h**. Waiting on further FBC and biochemical analysis is inconsequential in this stage of the patient's admission, so, there is no reason to not use extra information for modelling. Follow-up models might also benefit from a voting system that automatizes model comparison.

Imaging methods have shown relevance to AIS treatment outcomes modelling, especially when CNNs are used directly in imaging data. The imaging biomarker studied in this thesis, **hemispheric contrast, has shown relevance to modelling**, but it did not provide statistically significantly better predictive capacity.

Despite neuroimaging-based models improvements on predictive capacity, they take longer to preprocess the data and make inferences, which is an extra step over radioimaging acquisition, something not desirable in the emergency settings where these models can be used as an assisted diagnosis tool for treatment selection. **Absolute HC biomarker has shown high relevance to modelling**, and it can be further improved with other deterministic IBs with excellent AIS predictive capacity — e.g., angular second moment, contrast, entropy, correlation, sum of squares, difference entropy, inverse difference moment, inertia, cluster prominence and shade, energy, homogeneity, dissimilarity and difference in variance [192] —, which can further improve predict outcomes, both on pre- and post- thrombolysis and thrombectomy imaging data. ASPECTS is a well-known imaging biomarker with a body of evidence on AIS prediction, usually used in conjunction with other variables [193]. ASPECTS alone achieved $AUC_{LR\_ASPECTS} = 0.75$ while predicting 3-month outcomes in a 200 patients study [194].

There are some **limitation in this study worth mentioning**. In this work, although cross-validated validation metrics achieved outstanding results, single point test metrics diverged enough to warrant caution. Selected models have shown robust results even with the small dataset that was available. Training the selected models on larger datasets, should existing cohort data collection efforts continue, will make future models more robust, especially if more cases with minority characteristics or attributes not contemplated in this study are present — e.g., AIS cases in middle-aged individuals, or AIS events in different arteries and territories. Increasing the amount of training data is key to train robust CNNs. Comparison with other studies may be biased, since this study used data on medical discharge, while most AIS outcomes studies model outcomes over a 90-day period.

During this work, while assessing HP, it was also demonstrated **increased number of features does not imply better model performance**, due to multicollinearity, mathematical coupling and noisy features with little to no relation to the outcome.

The IB calculated during this thesis has shown good predictive results, being the fifth most relevant feature for outcome prediction. As studied on at admission NCCTs, it was unable to consistently improve models containing it, and its simple inclusion was not enough to improve the best models. However, as a novel improvement, **absolute HC enables models with fewer features for equivalent performance**. If its value is confirmed with large scale studies, this may be suitable to replace more subjective clinical variables.

Imaging data alone is so relevant that it produced top results through 3D CNNs. DL models are difficult to interpret and explain, so despite their impressive results, they usually do not expand overall knowledge on the domain they are applied to and their black-box design is not reassuring to users. This thesis has shown promising strategies to further improve this 3D CNN performance and generalization capacity, so future work with more data and computational resources is bound to improve these models.

8.2 Future work

First and foremost, creating a multimodal implementation that would enable merging tabular data with imaging data would likely provide more robust results, and would ease the burden of running multiple models at one stage and compare manually their results. Tabular data models might be improved with feature engineering, exploring massive number of derived features with tools such as `featuretools`, and then selecting the best feature with the methods already explored or via recursive feature elimination. AutoML was lightly explored during this thesis. Within the search time provided, its results were unable to match human guided search spaces and FS, but this is a constantly expanding ML field. Several other tools explored in a non-structured way during this thesis may outperform existing models, since preliminary exploration has shown promising results with AutoPyTorch. Testing these tools with higher budgets will help study how improvements scale with the budget and if the best models found are the best possible models for the provided dss.. BioStroke has fuelled the discovery of several biomarkers associated with AIS not assessed in routine analysis, so integrating information on those biomarkers in future cohorts can show improvements on these predictive models. Integrating more reliable thrombectomy evaluation features such as mTICI, or selecting cases for the training ds. where full recanalization was achieved, is likely to help the models perform better, and calibrate posterior probabilities with publicly accessible ones [195]. Recanalization variable success implies HC can be even more valuable as a post-thrombectomy outcome predictor, a feature not tested during this thesis. Exploring existing AIS IB as the ones referred by Hema Rajini *et al.* [192] is a deterministic CV task that can provide models that offer insights as neuroimaging data is collected. Assessing damage to each functional area and extracting that information as a biomarker after proper segmentation can provide derived imaging features useful for several neuroimaging tasks — including precise thrombectomy outcome prediction. Scoring systems such as e-ASPECTs can be expanded, accounting for more precise quantification, more separate regions to be analysed, or account for the relevance of the area affected to perceived dysfunction. Finally, to use models in high-consequence scenarios and improve them, they need to be interpretable. Tools as SHapley Additive exPlanationns or ELI5 can help understand the inner workings of black-box models such as the LGBM, XGBM and NNs, and bring insights that can help improve those models. Image data augmentations on properly registered data were unable to produce improved models, possibly indicating the importance of registration as a time and energy saving method on this kind of dss. and can be investigated further. Further exploring NAS, with more appropriate equipment may improve results and find architectures able to generalize better with little data, since filter size on this project may have been a limiting factor. Transfer learning can be further improved by using 3D versions of ResNet or ImageNet for neuroimaging modelling, since they enable learning from models with much more data, parameters and training time. Alternatively chaining imaging related modelling studies from other medical domains may improve overall results, since there is less likelihood of negative knowledge transfers.

8.3 Ethics and Data Protection

The dss. for this study were provided under a non-disclosure agreement, so it will not be made publicly available. Taking into account medical records' sensitive nature, all data was anonymized before being passed to the author, by **removing personally identifiable information** and changing the keys of each record to BioStroke study specific identification keys. The imaging analysis extension was approved by CHUP Ethical Committee before the author had access to the data. It was downloaded in CHUP computers, **stripped of all personal identifiable information on download**, and **further anonymized by deskulling**. No data from BioStroke was passed to Google Colab for processing.

8.4 Code Availability

The code from this study can be consulted on `https://github.com/TiagoSantos81/AISOutcomes` private repository, accessible through request.

Appendix A

Appendix

A.1 Stroke Disability and Severity Scales

TABLE A.1: Modified Rankin Score table [24].

Grade	Patient's Description
0	Without symptoms
1	Without significant disability despite symptoms and able to carry out all usual duties and activities
2	Only has a slight disability in which it is unable to perform all previous activities
3	Requires some help, but it is able to walk without assistance
4	Unable to walk without assistance and unable to attend to own bodily needs without assistance
5	The patient is bedridden, incontinent and requiring constant nursing care and attention
6	Death

A.2 Clinical Data Appendix

TABLE A.2: Best clinical models AUC results comparison table.

	meanrank	mean	std	ci_lower	ci_upper	effect_size	magnitude
CategoricalNB	3.10	0.55	0.18	0.49	0.60	0.00	negligible
KNeighborsClassifier	3.60	0.56	0.18	0.51	0.62	-0.09	negligible
QuadraticDiscriminantAnalysis	4.26	0.61	0.17	0.56	0.66	-0.35	small
MLPClassifier	5.88	0.67	0.15	0.63	0.72	-0.76	medium
DecisionTreeClassifier	6.30	0.70	0.15	0.65	0.74	-0.91	large
SVC	6.47	0.69	0.15	0.64	0.74	-0.86	large
LinearSVC	6.88	0.72	0.14	0.67	0.76	-1.04	large
LinearDiscriminantAnalysis	7.64	0.73	0.16	0.68	0.78	-1.10	large
AdaBoostClassifier	8.42	0.76	0.15	0.71	0.80	-1.27	large
RandomForestClassifier	8.76	0.78	0.13	0.74	0.81	-1.46	large
LGBMClassifier	9.04	0.78	0.14	0.74	0.82	-1.44	large
XGBClassifier	9.13	0.78	0.13	0.74	0.82	-1.49	large
LogisticRegression	10.12	0.80	0.13	0.76	0.84	-1.56	large

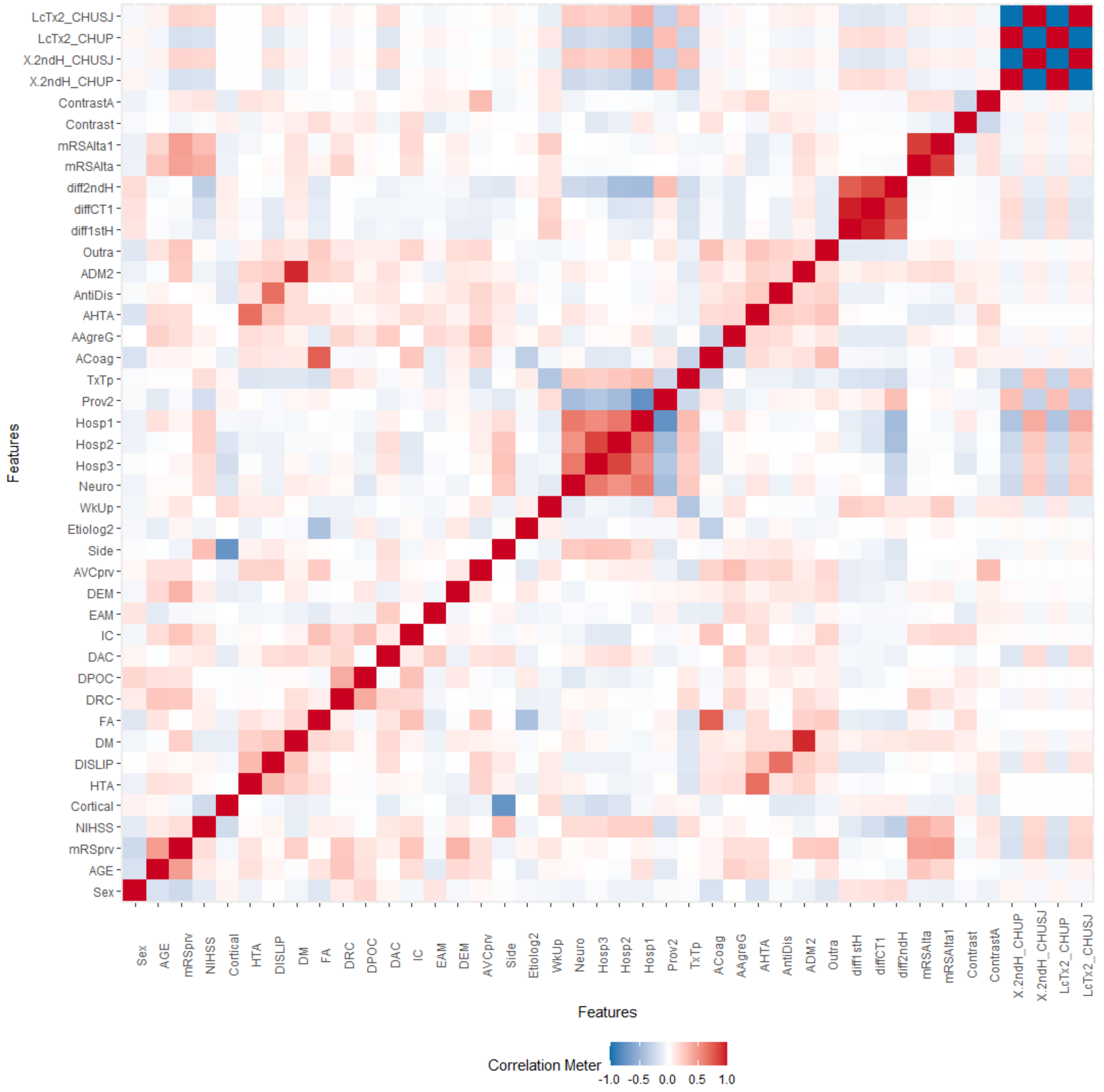


FIGURE A.1: Pearson's correlation heatmap accounting all variables and dummy variables in the clinical dataset.

TABLE A.3: AUC comparison table with both AutoSkLearn algorithms tested.

	meanrank	mean	std	ci_lower	ci_upper	effect_size	magnitude
CategoricalNB	3.42	0.55	0.18	0.49	0.60	0.00	negligible
KNeighborsClassifier	3.90	0.56	0.18	0.51	0.62	-0.09	negligible
QuadraticDiscriminantAnalysis	4.68	0.61	0.17	0.56	0.66	-0.35	small
MLPClassifier	6.56	0.67	0.15	0.63	0.72	-0.76	medium
DecisionTreeClassifier	7.10	0.70	0.15	0.65	0.74	-0.91	large
SVC	7.21	0.69	0.15	0.64	0.74	-0.86	large
Auto-SkLearn 2.0	7.72	0.71	0.13	0.67	0.74	-1.03	large
LinearSVC	7.80	0.72	0.14	0.67	0.76	-1.04	large
LinearDiscriminantAnalysis	8.61	0.73	0.16	0.68	0.78	-1.10	large
Auto-SkLearn 1.0	9.44	0.75	0.13	0.71	0.79	-1.31	large
AdaBoostClassifier	9.56	0.76	0.15	0.71	0.80	-1.27	large
RandomForestClassifier	10.03	0.78	0.13	0.74	0.82	-1.46	large
LGBMClassifier	10.36	0.78	0.14	0.74	0.82	-1.44	large
XGBClassifier	10.45	0.78	0.13	0.74	0.82	-1.49	large
LogisticRegression	11.55	0.80	0.13	0.76	0.84	-1.56	large

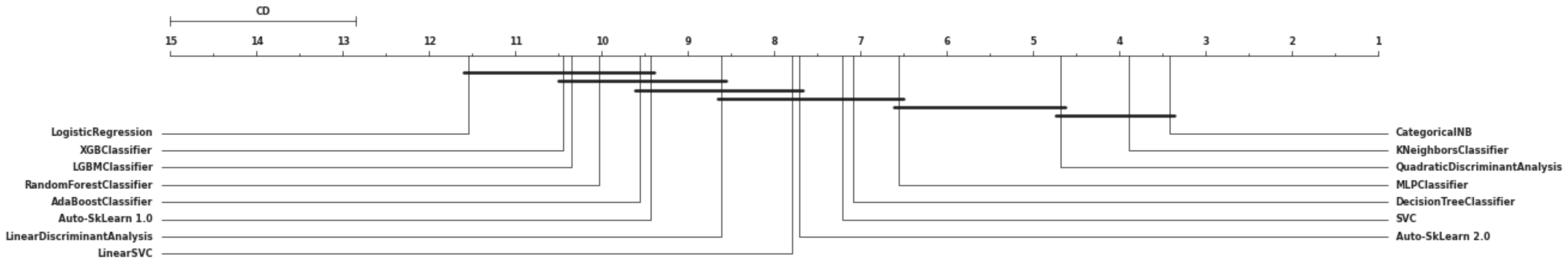


FIGURE A.2: Nemenyi plot comparing AutoML models with all the best described so far.

A.3 Biomarkers Data Appendix

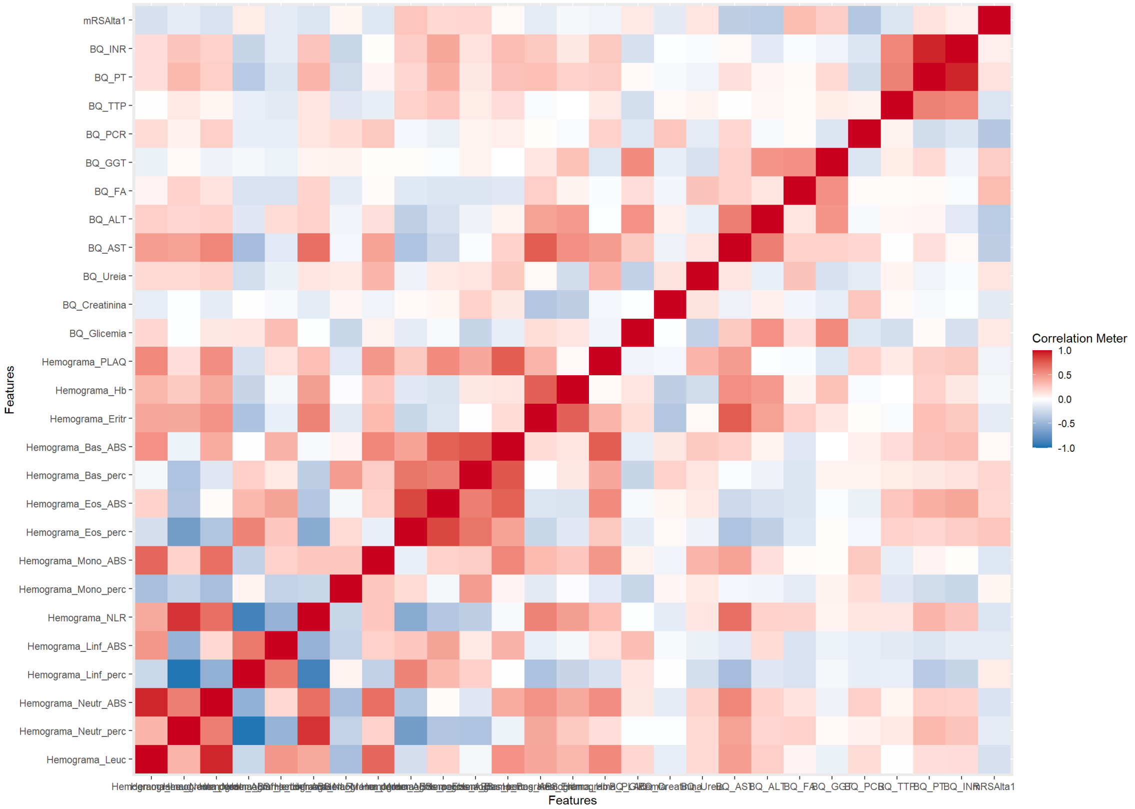


FIGURE A.3: Pearson's correlation heatmap accounting all variables and dummies in the biomarkers at admission dataset.

TABLE A.4: Biomarkers ds. patients' logistic fields description.

Field Name 0h	Field Name 24h	Description
Ref.Biostroke	Ref.Biostroke	BioStroke Key
1EXT		External Patient? 1 – True, 0 – False
Onde		Which External Hospital
Data AVC Sintomas		1st Symptoms Date
Data 1º Hospital		Hospital Admission Date
Hora 1º Hospital		Hospital Admission Hour
Data 1º TAC		1st CT Scan Date
Hora 1º TAC		1st CT Scan Hour
2EXT	3EXT	Acute phase clinical analysis conducted in CHUP? 0 - True, 1 – False

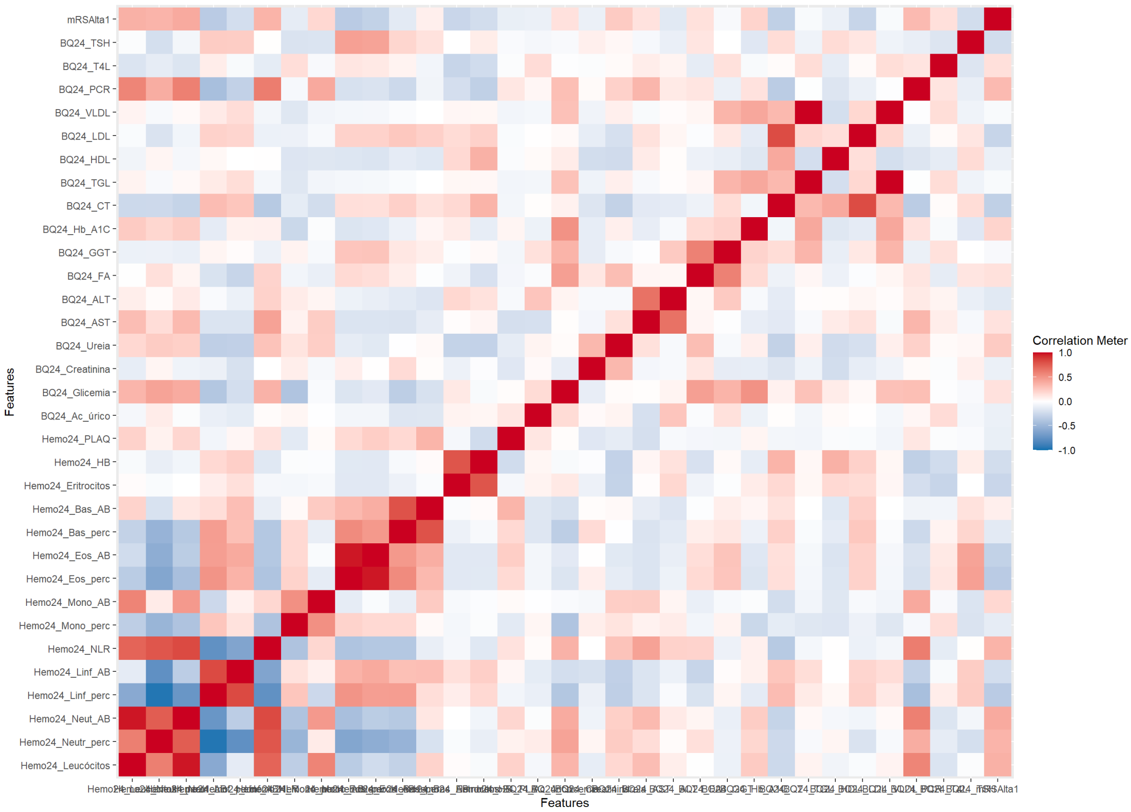


FIGURE A.4: Pearson's correlation heatmap accounting all variables and dummies in the biomarkers at follow-up dataset.

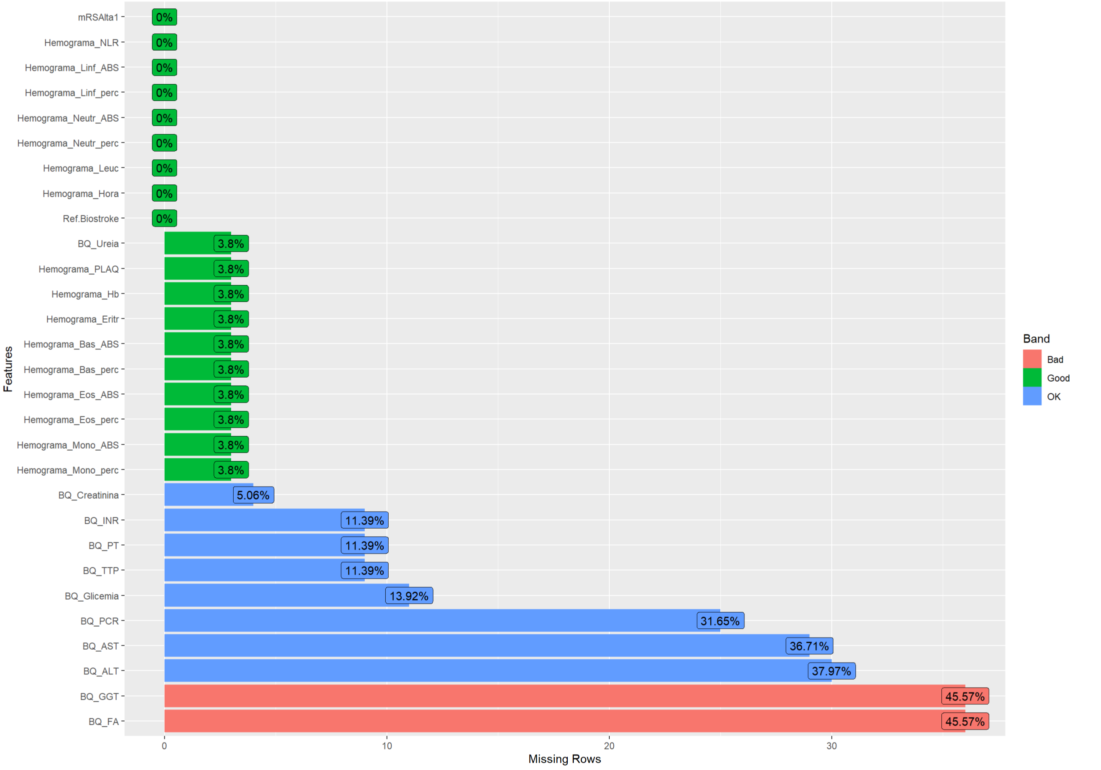


FIGURE A.5: At admission biomarker dataset missing data profile after calculations.

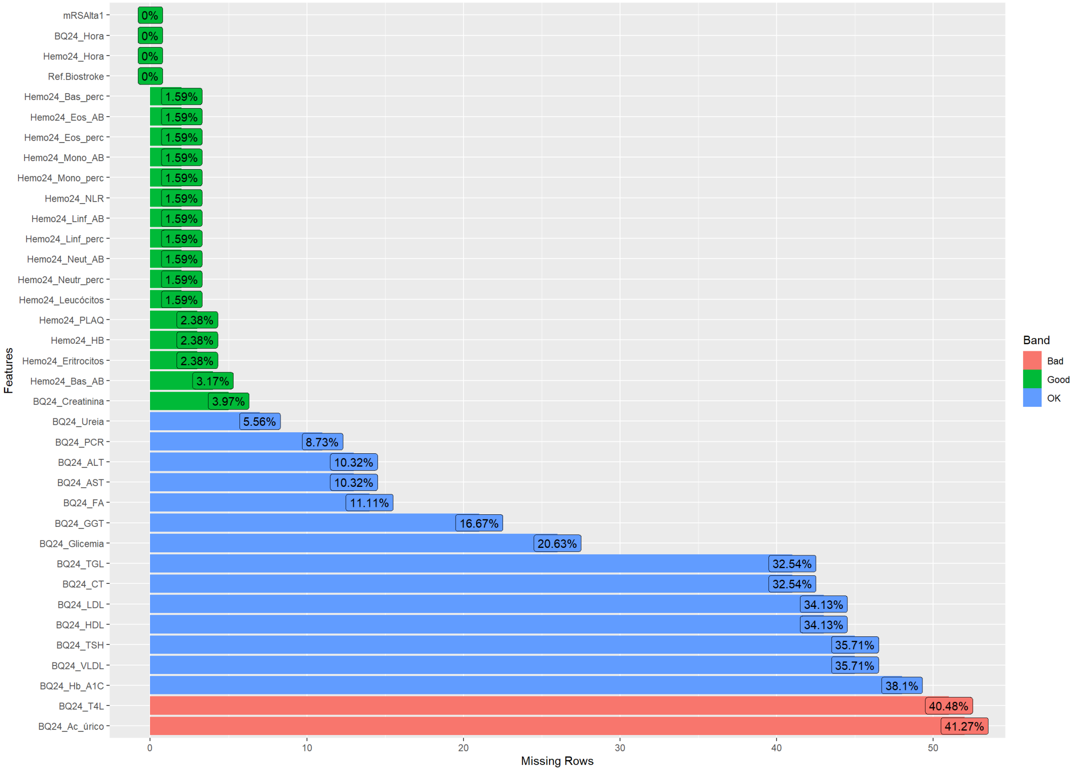
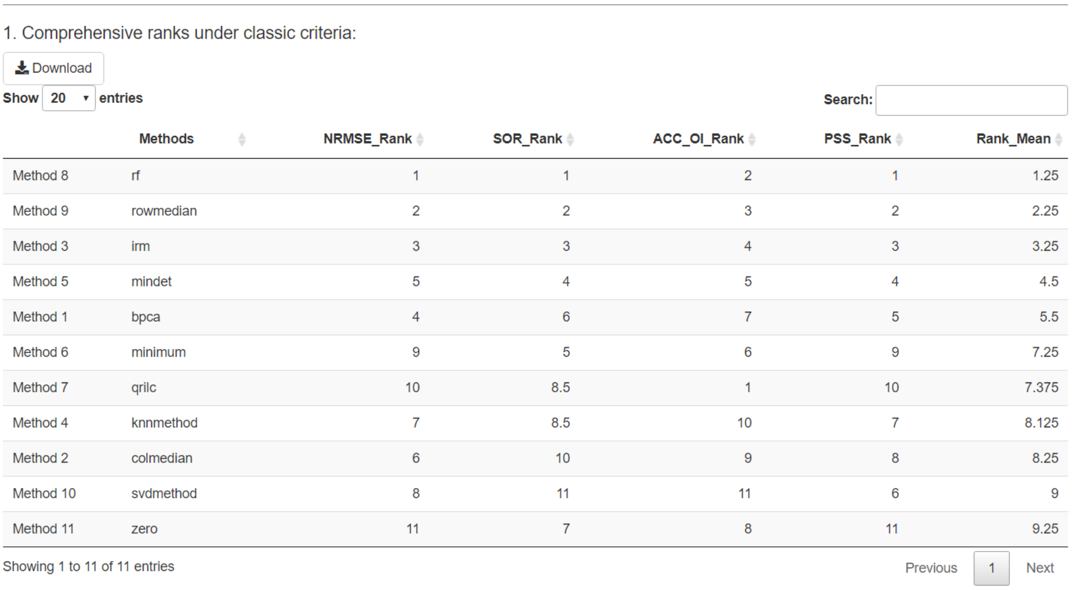


FIGURE A.6: At follow-up biomarker dataset missing data profile after calculations.


1. Comprehensive ranks under classic criteria:

	Methods	NRMSE_Rank	SOR_Rank	ACC_OI_Rank	PSS_Rank	Rank_Mean
Method 8	rf	1	1	2	1	1.25
Method 9	rowmedian	2	2	3	2	2.25
Method 3	irm	3	3	4	3	3.25
Method 5	mindet	5	4	5	4	4.5
Method 1	bpca	4	6	7	5	5.5
Method 6	minimum	9	5	6	9	7.25
Method 7	qrilc	10	8.5	1	10	7.375
Method 4	knnmethod	7	8.5	10	7	8.125
Method 2	colmedian	6	10	9	8	8.25
Method 10	svdmethod	8	11	11	6	9
Method 11	zero	11	7	8	11	9.25

Showing 1 to 11 of 11 entries

FIGURE A.7: NAGuideR imputations evaluation on admission ds..

TABLE A.5: Biomarkers ds. patients' FBC related fields description.

Field Name 0h	Field Name 24h	Description
Hemograma_Data	Hemo24_Data	FBC Date
Hemograma_Hora	Hemo24_Hora	FBC Hour
Hemograma_Leuc	Hemo24_Leucócitos	Leucocytes ($10^3/\mu$L)
Hemograma_Neutr(%)	Hemo24_Neutr(%)	Neutrophils (%)
Hemograma_Neutr ABS	Hemo24_Neut AB	Neutrophils ($10^3/\mu$L)
Hemograma_Linf (%)	Hemo24_Linf (%)	Lymphocytes (%)
Hemograma_Linf ABS	Hemo24_Linf AB	Lymphocytes ($10^3/\mu$L)
Hemograma_NLR	Hemo24_NLR	Neutrophils-Leucocytes Ratio (NLR)
Hemograma_Mono (%)	Hemo24_Mono (%)	Monocytes (%)
Hemograma_Mono ABS	Hemo24_Mono AB	Monocytes ($10^3/\mu$L)
Hemograma_Eos (%)	Hemo24_Eos (%)	Eosinophils (%)
Hemograma_Eos ABS	Hemo24_Eos AB	Eosinophils ($10^3/\mu$L)
Hemograma_Bas (%)	Hemo24_Bas (%)	Basophils (%)
Hemograma_Bas ABS	Hemo24_Bas AB	Basophils ($10^3/\mu$L)
Hemograma_Eritr	Hemo24_Eritrocitos	Erythrocytes ($10\hat{6}/\mu$L)
Hemograma_Hb	Hemo24_HB	Haemoglobin (g/dL)
Hemograma_PLAQ	Hemo24_PLAQ	Platelets ($10^3/\mu$L)

TABLE A.6: Biomarkers ds. patients' biochemical related fields description.

Field Name 0h	Field Name 24h	Description
BQ_Data	BQ24_Data	
BQ_Hora	BQ24_Hora	
BQ_Glicemia	BQ24_Glicemia	Glycose (mg/dL)
BQ_Creatinina	BQ24_Creatinina	Creatinine (mg/dL)
BQ_Ureia	BQ24_Ureia	Urea (mg/dL)
BQ_AST	BQ24_AST	Aspartate aminotransferase (U/L a 37°)
BQ_ALT	BQ24_ALT	Alanine Aminotransferase (U/L a 37°)
BQ_FA	BQ24_FA	Alkaline Phosphatasis (U/L a 37°)
BQ_GGT	BQ24_GGT	Gamma-Glutamyl-Transferase (U/L a 37°)
BQ_PCR	BQ24_PCR	C-reactive protein (mg/L)
BQ_Pro-BNP	BQ24_Pro-BNP	B-type Natriuretic Peptide PROmotor Hormone (pg/mL)
BQ_TTP		Partial Thromboplastin Time (s)
BQ_PT		Prothrombin Time (s)
BQ_INR		International Normalised Ratio
	BQ24_Ac úrico	Uric Acid (mg/dL)
	BQ24_Hb A1C	Haemoglobin A1c (%)
	BQ24_CT	Total Cholesterol (mg/dL)
	BQ24_TGL	Triglycerides (mg/dL)
	BQ24_HDL	High-density lipoprotein (HDL) cholesterol (mg/dL)
	BQ24_LDL	Low-density lipoprotein (LDL) cholesterol (mg/dL)
	BQ24_VLDL	Very-Low-density lipoprotein (VLDL) cholesterol (mg/dL)
	BQ24_T4L	Free T4 - Free Thyroxine (ng/dL)
	BQ24_TSH	Tiroid-Stimulating Hormone (ng/dL)
	BQ24_Homocisteína	Homocysteine (μUI/mL)

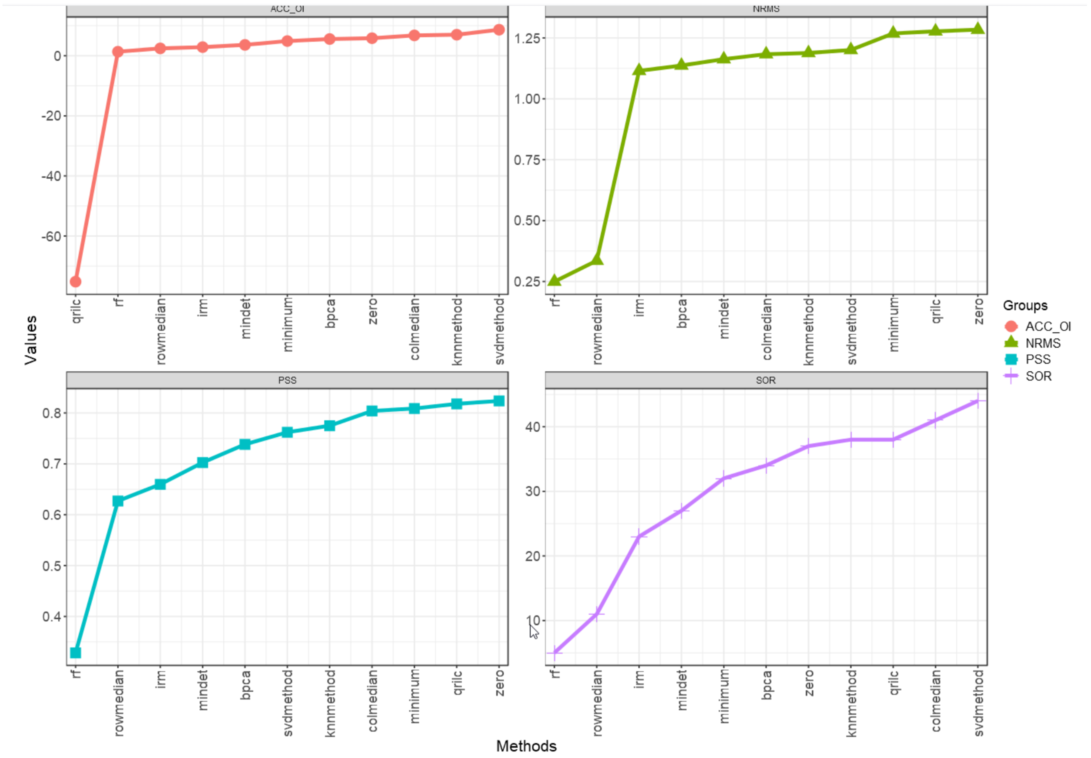
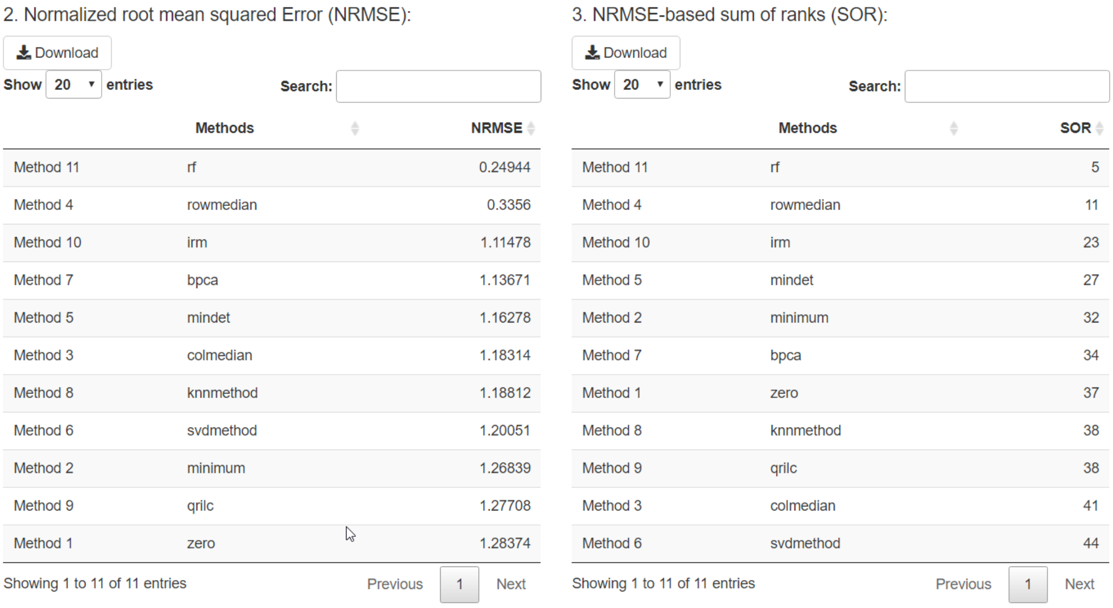


FIGURE A.8: NAGuideR imputations metrics plots on admission ds..

TABLE A.7: Normalized Root Mean Squared Error (NRMSR) and NRMSE-based Sum Of Ranks (SOR).



2. Normalized root mean squared Error (NRMSE):

Download

Show 20 entries Search:

	Methods	NRMSE
Method 11	rf	0.24944
Method 4	rowmedian	0.3356
Method 10	irm	1.11478
Method 7	bpca	1.13671
Method 5	mindet	1.16278
Method 3	colmedian	1.18314
Method 8	knnmethod	1.18812
Method 6	svdmethod	1.20051
Method 2	minimum	1.26839
Method 9	qrilc	1.27708
Method 1	zero	1.28374

Showing 1 to 11 of 11 entries Previous 1 Next

3. NRMSE-based sum of ranks (SOR):

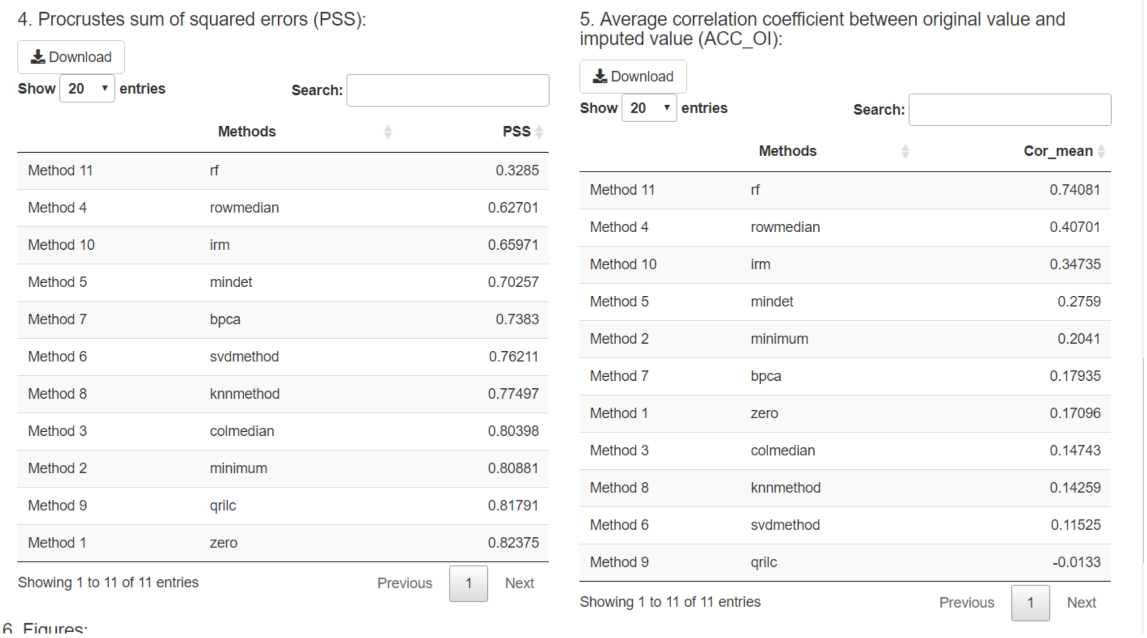
Download

Show 20 entries Search:

	Methods	SOR
Method 11	rf	5
Method 4	rowmedian	11
Method 10	irm	23
Method 5	mindet	27
Method 2	minimum	32
Method 7	bpca	34
Method 1	zero	37
Method 8	knnmethod	38
Method 9	qrilc	38
Method 3	colmedian	41
Method 6	svdmethod	44

Showing 1 to 11 of 11 entries Previous 1 Next

TABLE A.8: Procustes sum of squared errors (PSS) and ACC_OI tables for the selected models.



4. Procrustes sum of squared errors (PSS):

	Methods	PSS
Method 11	rf	0.3285
Method 4	rowmedian	0.62701
Method 10	irm	0.65971
Method 5	mindet	0.70257
Method 7	bpca	0.7383
Method 6	svdmethod	0.76211
Method 8	knnmethod	0.77497
Method 3	colmedian	0.80398
Method 2	minimum	0.80881
Method 9	qrilc	0.81791
Method 1	zero	0.82375

Showing 1 to 11 of 11 entries

6. Figures:

5. Average correlation coefficient between original value and imputed value (ACC_OI):

	Methods	Cor_mean
Method 11	rf	0.74081
Method 4	rowmedian	0.40701
Method 10	irm	0.34735
Method 5	mindet	0.2759
Method 2	minimum	0.2041
Method 7	bpca	0.17935
Method 1	zero	0.17096
Method 3	colmedian	0.14743
Method 8	knnmethod	0.14259
Method 6	svdmethod	0.11525
Method 9	qrilc	-0.0133

Showing 1 to 11 of 11 entries

TABLE A.9: Best hypermaters for models trained on biomarkers datasets.

Model	Parameters
LGBMClassifier (0h RF)	{’classifier__boosting_type’: ’gbdt’, ’classifier__learning_rate’: 0.1, ’classifier__max_depth’: 2, ’classifier__min_child_weight’: 2}
XGBClassifier (0h RF)	{’classifier__booster’: ’gblinear’, ’classifier__learning_rate’: 0.1, ’classifier__max_depth’: 2, ’classifier__min_child_weight’: 2}
LinearSVC (0h RF)	{’classifier__C’: 10, ’classifier__loss’: ’hinge’, ’classifier__penalty’: ’l2’}
SVC (0h RF)	{’classifier__C’: 100, ’classifier__coef0’: -1, ’classifier__degree’: 3, ’classifier__gamma’: ’auto’, ’classifier__kernel’: ’poly’}
LogisticRegression (0h RF)	{’classifier__C’: 0.01, ’classifier__max_iter’: 400, ’classifier__penalty’: ’none’, ’classifier__solver’: ’saga’}
LGBMClassifier (0h RM)	{’classifier__boosting_type’: ’gbdt’, ’classifier__learning_rate’: 0.05, ’classifier__max_depth’: 2, ’classifier__min_child_weight’: 2}
XGBClassifier (0h RM)	{’classifier__booster’: ’gblinear’, ’classifier__learning_rate’: 0.1, ’classifier__max_depth’: 4, ’classifier__min_child_weight’: 8}
LinearSVC (0h RM)	{’classifier__C’: 1, ’classifier__loss’: ’hinge’, ’classifier__penalty’: ’l2’}
SVC (0h RM)	{’classifier__C’: 100, ’classifier__coef0’: 0, ’classifier__degree’: 2, ’classifier__gamma’: ’auto’, ’classifier__kernel’: ’poly’}
LogisticRegression (0h RM)	{’classifier__C’: 20, ’classifier__max_iter’: 100, ’classifier__penalty’: ’l2’, ’classifier__solver’: ’sag’}
LGBMClassifier (24h RF)	{’classifier__boosting_type’: ’gbdt’, ’classifier__learning_rate’: 0.1, ’classifier__max_depth’: 4, ’classifier__min_child_weight’: 2}
XGBClassifier (24h RF)	{’classifier__booster’: ’gbtree’, ’classifier__learning_rate’: 0.1, ’classifier__max_depth’: 6, ’classifier__min_child_weight’: 2}
LinearSVC (24h RF)	{’classifier__C’: 0.5, ’classifier__loss’: ’hinge’, ’classifier__penalty’: ’l2’}
SVC (24h RF)	{’classifier__C’: 10, ’classifier__coef0’: -1, ’classifier__degree’: 3, ’classifier__gamma’: ’auto’, ’classifier__kernel’: ’poly’}
LogisticRegression (24h RF)	{’classifier__C’: 1, ’classifier__max_iter’: 100, ’classifier__penalty’: ’l2’, ’classifier__solver’: ’newton-cg’}
LGBMClassifier (24h RM)	{’classifier__boosting_type’: ’gbdt’, ’classifier__learning_rate’: 0.1, ’classifier__max_depth’: 2, ’classifier__min_child_weight’: 2}
XGBClassifier (24h RM)	{’classifier__booster’: ’gbtree’, ’classifier__learning_rate’: 0.05, ’classifier__max_depth’: 4, ’classifier__min_child_weight’: 2}
LinearSVC (24h RM)	{’classifier__C’: 100, ’classifier__loss’: ’hinge’, ’classifier__penalty’: ’l2’}
SVC (24h RM)	{’classifier__C’: 0.5, ’classifier__coef0’: 0, ’classifier__degree’: 2, ’classifier__gamma’: ’scale’, ’classifier__kernel’: ’poly’}
LogisticRegression (24h RM)	{’classifier__C’: 1, ’classifier__max_iter’: 100, ’classifier__penalty’: ’l2’, ’classifier__solver’: ’liblinear’}

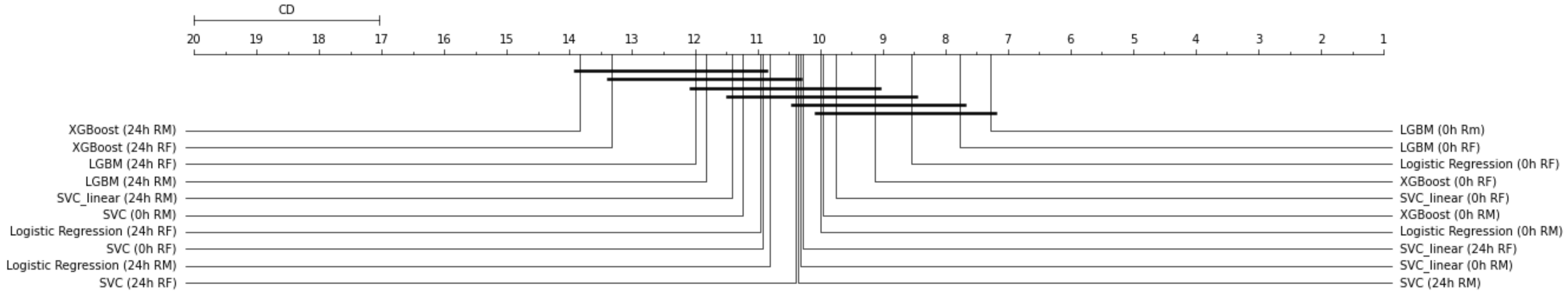


FIGURE A.9: Nemenyi's plot of all best biomarkers models with both imputation methods and both subsets.

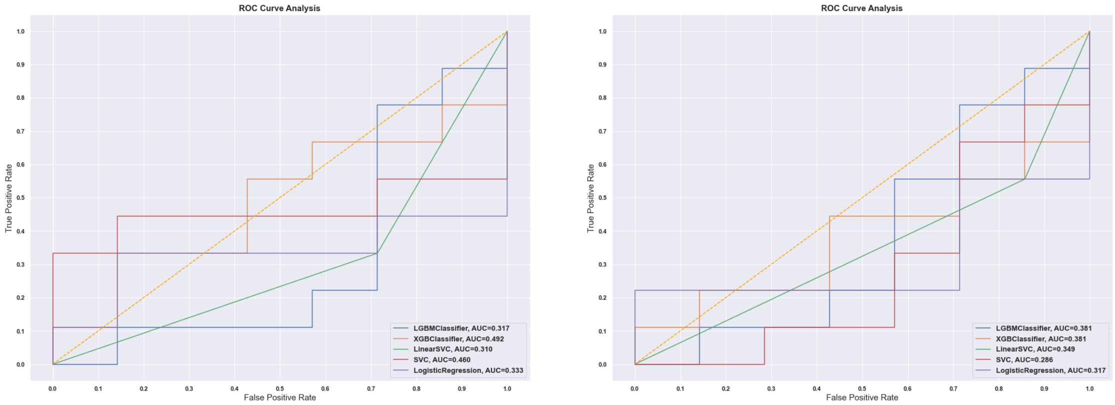


FIGURE A.10: ROC curves of best Biom0h models with RF imputation, tested on the test set, on the left, and with row median imputations, on the right.

TABLE A.10: Best biomarkers models AUC scores and ranking.

	meanrank	**median**	**mad**	**ci_lower**	**ci_upper**	**effect_size**	**magnitude**
LGBM (0h Rm)	7.27	0.56	0.14	0.44	0.67	0.00	negligible
LGBM (0h RF)	7.76	0.58	0.17	0.44	0.75	-0.12	negligible
Logistic Regression (0h RF)	8.54	0.58	0.17	0.44	0.75	-0.12	negligible
XGBoost (0h RF)	9.12	0.58	0.19	0.44	0.78	-0.11	negligible
SVC_linear (0h RF)	9.76	0.67	0.17	0.50	0.78	-0.49	small
XGBoost (0h RM)	9.95	0.67	0.17	0.50	0.78	-0.49	small
Logistic Regression (0h RM)	10.00	0.67	0.17	0.56	0.78	-0.49	small
SVC_linear (24h RF)	10.28	0.67	0.12	0.58	0.79	-0.57	medium
SVC_linear (0h RM)	10.31	0.67	0.17	0.50	0.83	-0.49	small
SVC (24h RM)	10.35	0.67	0.11	0.58	0.76	-0.60	medium
SVC (24h RF)	10.38	0.67	0.12	0.58	0.79	-0.57	medium
Logistic Regression (24h RM)	10.80	0.67	0.12	0.62	0.80	-0.57	medium
SVC (0h RF)	10.92	0.67	0.19	0.56	0.83	-0.45	small
Logistic Regression (24h RF)	10.94	0.67	0.12	0.58	0.79	-0.57	medium
SVC (0h RM)	11.23	0.67	0.17	0.56	0.88	-0.49	small
SVC_linear (24h RM)	11.40	0.68	0.10	0.62	0.79	-0.70	medium
LGBM (24h RM)	11.82	0.71	0.12	0.62	0.83	-0.78	medium
LGBM (24h RF)	11.99	0.75	0.11	0.64	0.83	-1.05	large
XGBoost (24h RF)	13.32	0.75	0.12	0.67	0.88	-0.99	large
XGBoost (24h RM)	13.84	0.79	0.12	0.67	0.88	-1.20	large

TABLE A.11: Best biomarkers models and clinical models AUC scores table.

	meanrank	**median**	**mad**	**ci_lower**	**ci_upper**	**effect_size**	**magnitude**
LGBM (24h RM)	3.22	0.71	0.12	0.62	0.79	0.00	negligible
LGBM (24h RF)	3.51	0.75	0.11	0.64	0.83	-0.24	small
Random Forests (Base)	4.10	0.77	0.08	0.71	0.86	-0.40	small
XGBoost (24h RF)	4.11	0.75	0.12	0.67	0.84	-0.22	small
XGBoost (Base)	4.14	0.80	0.08	0.72	0.86	-0.58	medium
XGBoost (24h RM)	4.26	0.79	0.12	0.68	0.88	-0.45	small
Logistic Regression (Base)	4.66	0.80	0.08	0.74	0.88	-0.58	medium

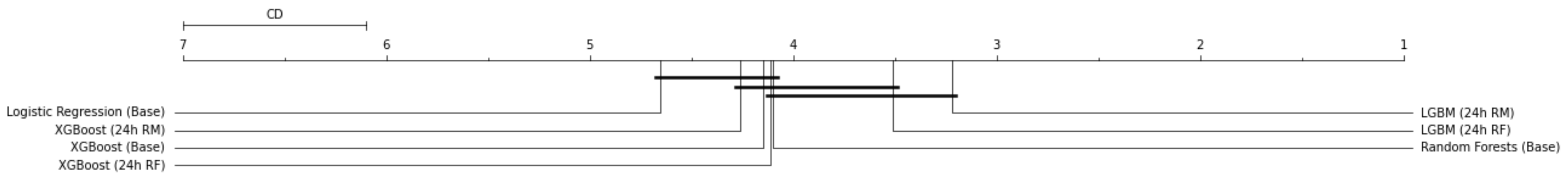


FIGURE A.11: Nemenyi's plot comparing best base biomarkers models against the best clinical dataset models.

A.4 Tabular Data Augmentation

TABLE A.12: SMOTE results on most successful models comparison table.

Model	Mean AUC		Std.
LogisticRegression	0.78	±	0.13
LGBMClassifier	0.76	±	0.13
XGBClassifier	0.74	±	0.14
SMOTE-LogisticRegression	0.78	±	0.14
SMOTE-LGBMClassifier	0.75	±	0.14
SMOTE-XGBClassifier	0.74	±	0.14
BorderlineSMOTE-LogisticRegression	0.77	±	0.14
BorderlineSMOTE-LGBMClassifier	0.76	±	0.13
BorderlineSMOTE-XGBClassifier	0.74	±	0.14
SVMSMOTE-LogisticRegression	0.77	±	0.14
SVMSMOTE-LGBMClassifier	0.75	±	0.14
SVMSMOTE-XGBClassifier	0.74	±	0.14
KMeansSMOTE-LogisticRegression	0.77	±	0.13
KMeansSMOTE-LGBMClassifier	0.75	±	0.13
KMeansSMOTE-XGBClassifier	0.75	±	0.13
TL-LogisticRegression	0.77	±	0.14
TL-LGBMClassifier	0.77	±	0.13
TL-XGBClassifier	0.76	±	0.14
SMOTE-TL-LogisticRegression	0.78	±	0.14
TL-SMOTE-LogisticRegression	0.76	±	0.13
TL-SMOTE-TL-LogisticRegression	0.77	±	0.13
SMOTE-TL-LGBMClassifier	0.75	±	0.13
TL-SMOTE-LGBMClassifier	0.76	±	0.13
TL-SMOTE-TL-LGBMClassifier	0.77	±	0.13
SMOTE-TL-XGBClassifier	0.75	±	0.14
TL-SMOTE-XGBClassifier	0.76	±	0.13
TL-SMOTE-TL-XGBClassifier	0.75	±	0.14
BorderlineSMOTE-TL-LogisticRegression	0.77	±	0.14
TL-BorderlineSMOTE-LogisticRegression	0.77	±	0.14
TL-BorderlineSMOTE-TL-LogisticRegression	0.77	±	0.14
BorderlineSMOTE-TL-LGBMClassifier	0.76	±	0.14
TL-BorderlineSMOTE-LGBMClassifier	0.76	±	0.13
TL-BorderlineSMOTE-TL-LGBMClassifier	0.76	±	0.13
BorderlineSMOTE-TL-XGBClassifier	0.75	±	0.14
TL-BorderlineSMOTE-XGBClassifier	0.75	±	0.14
TL-BorderlineSMOTE-TL-XGBClassifier	0.76	±	0.14
SVMSMOTE-TL-LogisticRegression	0.76	±	0.14
TL-SVMSMOTE-LogisticRegression	0.77	±	0.14
TL-SVMSMOTE-TL-LogisticRegression	0.77	±	0.14
SVMSMOTE-TL-LGBMClassifier	0.76	±	0.13
TL-SVMSMOTE-LGBMClassifier	0.76	±	0.13
TL-SVMSMOTE-TL-LGBMClassifier	0.76	±	0.14
SVMSMOTE-TL-XGBClassifier	0.75	±	0.14
TL-SVMSMOTE-XGBClassifier	0.75	±	0.14
TL-SVMSMOTE-TL-XGBClassifier	0.75	±	0.14
KMeansSMOTE-TL-LogisticRegression	0.77	±	0.14
TL-KMeansSMOTE-LogisticRegression	0.76	±	0.13
TL-KMeansSMOTE-TL-LogisticRegression	0.76	±	0.13
KMeansSMOTE-TL-LGBMClassifier	0.76	±	0.14
TL-KMeansSMOTE-LGBMClassifier	0.76	±	0.14
TL-KMeansSMOTE-TL-LGBMClassifier	0.76	±	0.13
KMeansSMOTE-TL-XGBClassifier	0.75	±	0.14
TL-KMeansSMOTE-XGBClassifier	0.75	±	0.13
TL-KMeansSMOTE-TL-XGBClassifier	0.75	±	0.14
ENN-LogisticRegression	0.58	±	0.14
ENN-LGBMClassifier	0.50	±	0.00
ENN-XGBClassifier	0.68	±	0.14
SMOTE-ENN-LogisticRegression	0.63	±	0.14
ENN-SMOTE-LogisticRegression	0.59	±	0.14
ENN-SMOTE-ENN-LogisticRegression	0.52	±	0.13
SMOTE-ENN-LGBMClassifier	0.57	±	0.16
ENN-SMOTE-LGBMClassifier	0.50	±	0.06
ENN-SMOTE-ENN-LGBMClassifier	0.50	±	0.00
SMOTE-ENN-XGBClassifier	0.68	±	0.13
ENN-SMOTE-XGBClassifier	0.65	±	0.15
ENN-SMOTE-ENN-XGBClassifier	0.56	±	0.16
BorderlineSMOTE-ENN-LogisticRegression	0.64	±	0.14
ENN-BorderlineSMOTE-LogisticRegression	0.59	±	0.14
ENN-BorderlineSMOTE-ENN-LogisticRegression	0.51	±	0.14
BorderlineSMOTE-ENN-LGBMClassifier	0.56	±	0.16
ENN-BorderlineSMOTE-LGBMClassifier	0.50	±	0.06
ENN-BorderlineSMOTE-ENN-LGBMClassifier	0.50	±	0.00
BorderlineSMOTE-ENN-XGBClassifier	0.69	±	0.14
ENN-BorderlineSMOTE-XGBClassifier	0.65	±	0.14
ENN-BorderlineSMOTE-ENN-XGBClassifier	0.56	±	0.16
SVMSMOTE-ENN-LogisticRegression	0.63	±	0.15
ENN-SVMSMOTE-LogisticRegression	0.58	±	0.14
ENN-SVMSMOTE-ENN-LogisticRegression	0.50	±	0.14
SVMSMOTE-ENN-LGBMClassifier	0.54	±	0.12
ENN-SVMSMOTE-LGBMClassifier	0.50	±	0.02
ENN-SVMSMOTE-ENN-LGBMClassifier	0.50	±	0.00
SVMSMOTE-ENN-XGBClassifier	0.68	±	0.14
ENN-SVMSMOTE-XGBClassifier	0.65	±	0.16
ENN-SVMSMOTE-ENN-XGBClassifier	0.54	±	0.16
KMeansSMOTE-ENN-LogisticRegression	0.59	±	0.16
ENN-KMeansSMOTE-LogisticRegression	0.58	±	0.13
ENN-KMeansSMOTE-ENN-LogisticRegression	0.48	±	0.13
KMeansSMOTE-ENN-LGBMClassifier	0.56	±	0.15
ENN-KMeansSMOTE-LGBMClassifier	0.49	±	0.05
ENN-KMeansSMOTE-ENN-LGBMClassifier	0.50	±	0.00
KMeansSMOTE-ENN-XGBClassifier	0.68	±	0.15
ENN-KMeansSMOTE-XGBClassifier	0.63	±	0.15
ENN-KMeansSMOTE-ENN-XGBClassifier	0.51	±	0.16

TABLE A.13: Best augmented clinical models cross-validated metrics.

Model	Parameters
LGBMClassifier	{’classifier__boosting_type’: ’gbdt’, ’classifier__learning_rate’: 0.1, ’classifier__max_depth’: 2, ’classifier__min_child_weight’: 4}
XGBClassifier	{’classifier__booster’: ’gbtree’, ’classifier__learning_rate’: 0.05, ’classifier__max_depth’: 4, ’classifier__min_child_weight’: 4}
LinearSVC	{’classifier__C’: 0.5, ’classifier__dual’: False, ’classifier__loss’: ’squared_hinge’, ’classifier__penalty’: ’l1’}
SVC	{’classifier__C’: 100, ’classifier__coef0’: -1, ’classifier__degree’: 2, ’classifier__gamma’: ’auto’, ’classifier__kernel’: ’rbf’}
LogisticRegression	{’classifier__C’: 1, ’classifier__max_iter’: 200, ’classifier__penalty’: ’l1’, ’classifier__solver’: ’saga’}

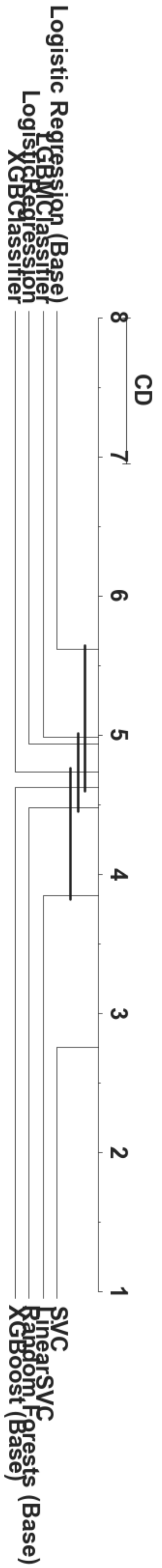


Figure A.12: Clin0h models augmented with TomekLinks-SMOTE-TomekLinks combination compared to base Clin0h models.

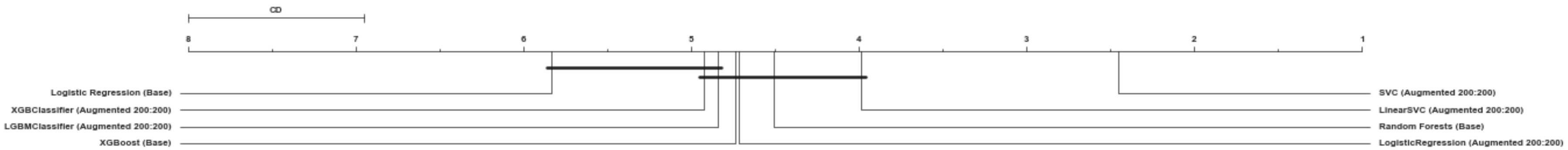


FIGURE A.13: Nemenyi's plot comparing models trained on balanced and augmented data against base models.

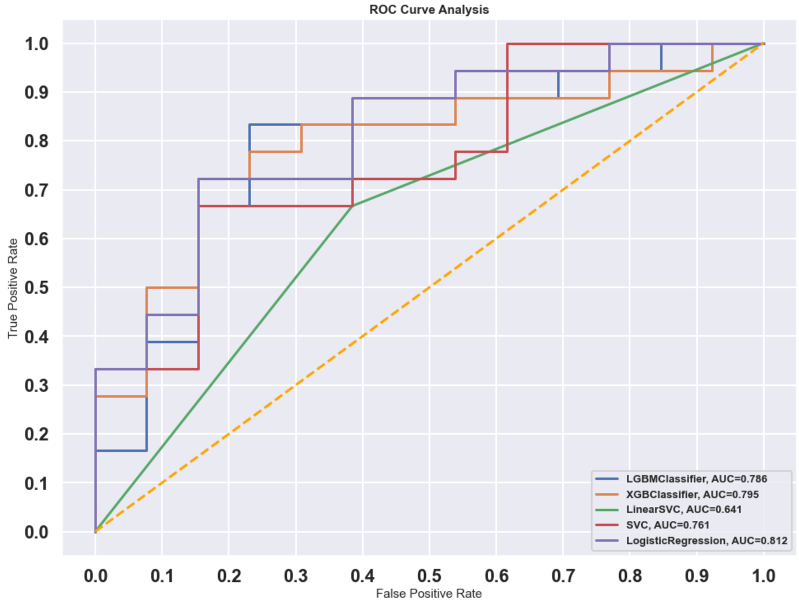


FIGURE A.14: Test set ROC curves for minority class SMOTE augmented models.

A.5 Feature Selection

A.6 Clin0h Feature Selection

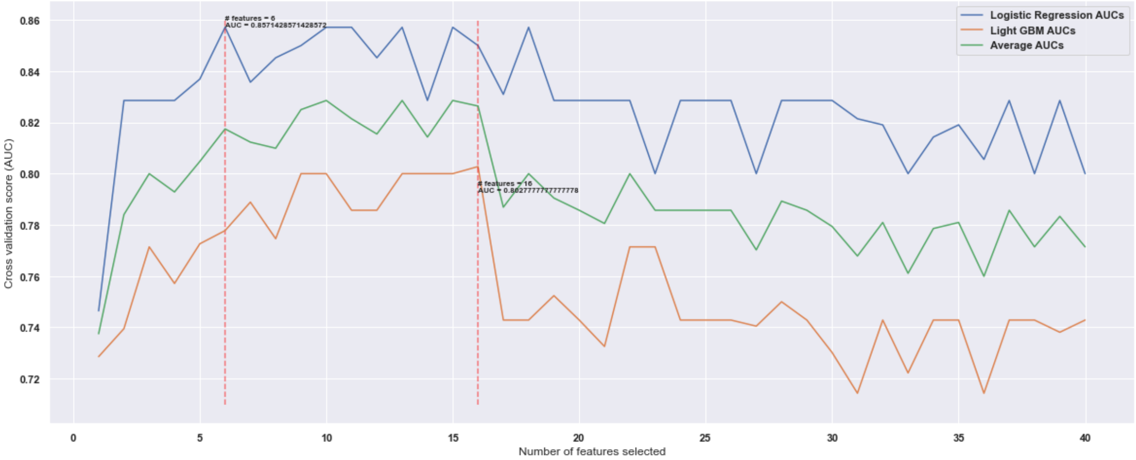


FIGURE A.15: Clinical ds. with strict feature selection peaking phenomenon analysis.

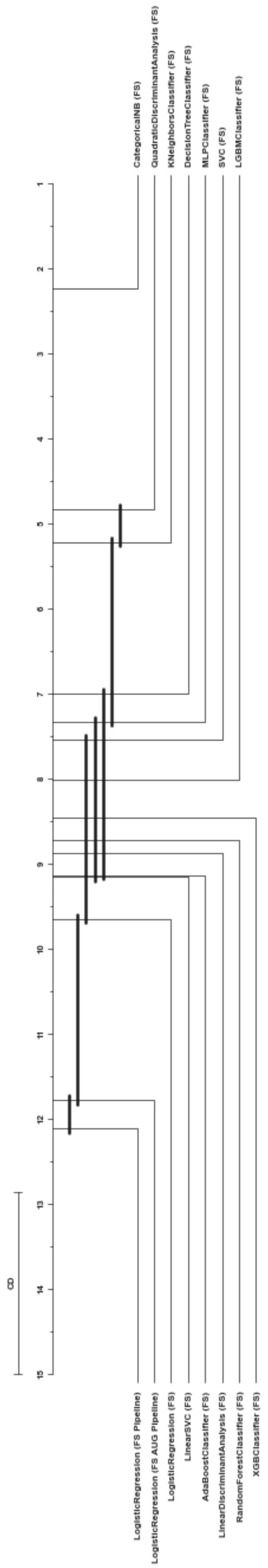


FIGURE A.16: Nemenyi's plot comparing all the best models trained on the ten best clinical features.

TABLE A.14: Clinical ds. with strict FS AUCs comparison table.

	meanrank	median	mad	ci_lower	ci_upper	effect_size	magnitude
CategoricalNB (FS)	2.20	0.57	0.14	0.44	0.68	0.00	negligible
QuadraticDiscriminantAnalysis (FS)	4.81	0.69	0.13	0.57	0.80	-0.57	medium
KNeighborsClassifier (FS)	5.17	0.68	0.14	0.000 0.57	0.80	-0.54	medium
DecisionTreeClassifier (FS)	6.99	0.73	0.08	0.68	0.81	-0.92	large
MLPClassifier (FS)	7.36	0.77	0.12	0.64	0.86	-1.04	large
SVC (FS)	7.52	0.77	0.11	0.66	0.86	-1.04	large
LGBMClassifier (FS)	8.02	0.77	0.08	0.70	0.83	-1.14	large
XGBClassifier (FS)	8.44	0.79	0.08	0.71	0.86	-1.25	large
RandomForestClassifier (FS)	8.70	0.77	0.08	0.71	0.86	-1.14	large
LinearDiscriminantAnalysis (FS)	8.86	0.80	0.08	0.71	0.88	-1.31	large
AdaBoostClassifier (FS)	9.08	0.80	0.08	0.71	0.86	-1.31	large
LinearSVC (FS)	9.12	0.79	0.10	0.71	0.88	-1.19	large
LogisticRegression (FS)	9.58	0.80	0.08	0.74	0.88	-1.31	large
LogisticRegression (FS Pipeline)	12.08	0.84	0.07	0.77	0.91	-1.62	large
LogisticRegression (FS AUG Pipeline)	12.08	0.84	0.07	0.77	0.91	-1.62	large

A.7 Biom0h Feature Selection

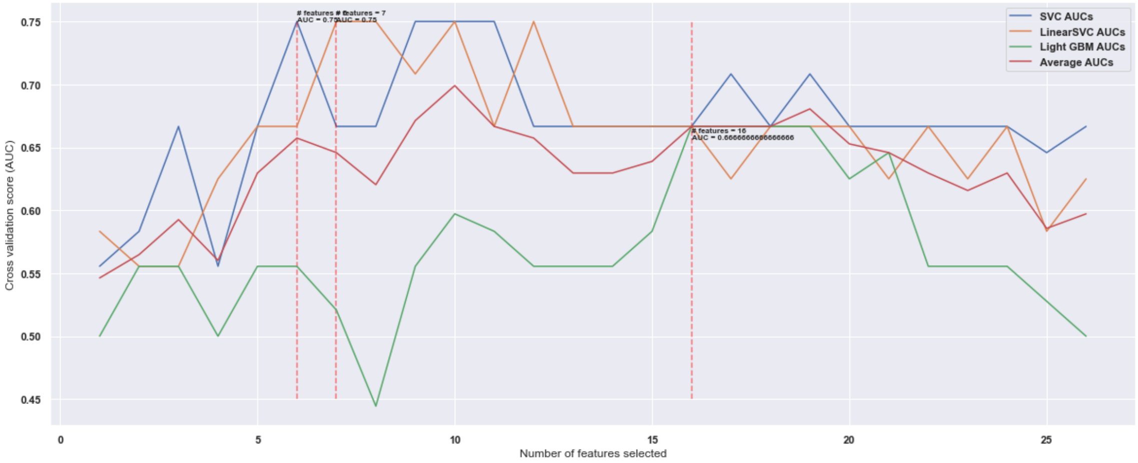


FIGURE A.17: Biom0h with strict feature selection peaking phenomenon analysis.

TABLE A.15: Biom0h RM with strict feature selection AUCs comparison table.

	meanrank	mean	std	ci_lower	ci_upper	effect_size	magnitude
LGBMClassifier	3.01	0.48	0.21	0.43	0.53	0.00	negligible
SVC	3.33	0.48	0.22	0.44	0.53	-0.02	negligible
LinearSVC	3.41	0.54	0.24	0.49	0.59	-0.27	small
LogisticRegression	3.54	0.54	0.25	0.50	0.59	-0.28	small
XGBClassifier	3.80	0.56	0.25	0.51	0.61	-0.36	small
MLPClassifier	3.91	0.57	0.25	0.52	0.62	-0.39	small

A.8 Biom24h Feature Selection

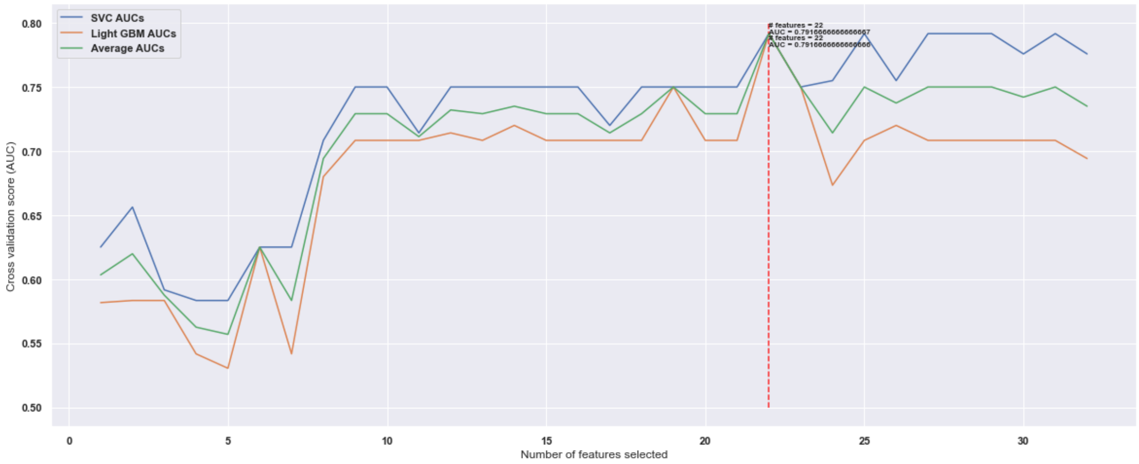


FIGURE A.18: Biom24h with strict feature selection peaking phenomenon analysis.

TABLE A.16: Biom24h with strict feature selection AUCs comparison table.

	meanrank	mean	std	ci_lower	ci_upper	effect_size	magnitude
MLPClassifier	3.10	0.65	0.16	0.62	0.68	0.00	negligible
LogisticRegression	3.14	0.65	0.17	0.62	0.68	-0.01	negligible
LinearSVC	3.18	0.67	0.16	0.64	0.70	-0.12	negligible
SVC	3.56	0.70	0.16	0.66	0.73	-0.28	small
LGBMClassifier	3.77	0.72	0.15	0.69	0.76	-0.46	small
XGBClassifier	4.24	0.75	0.16	0.71	0.78	-0.59	medium

A.9 ClinCA0h Feature Selection

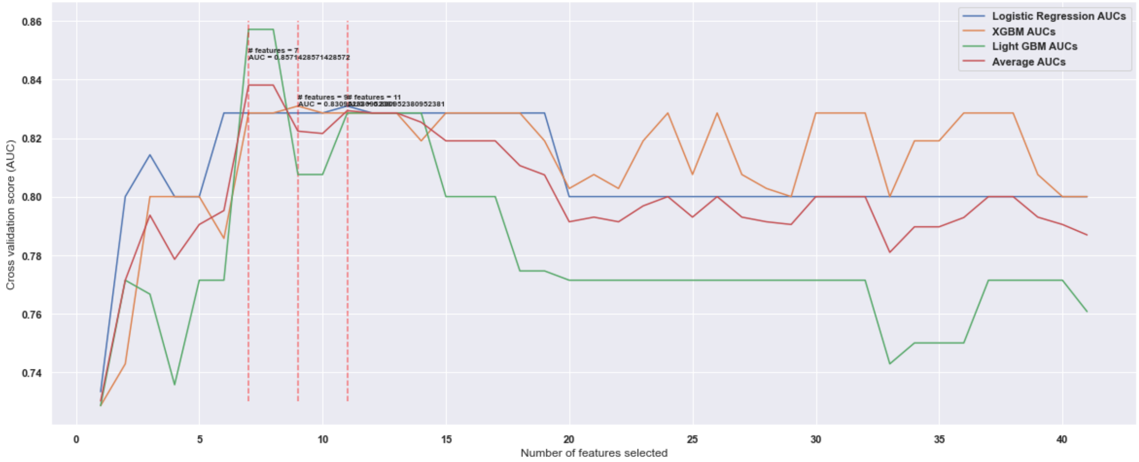


FIGURE A.19

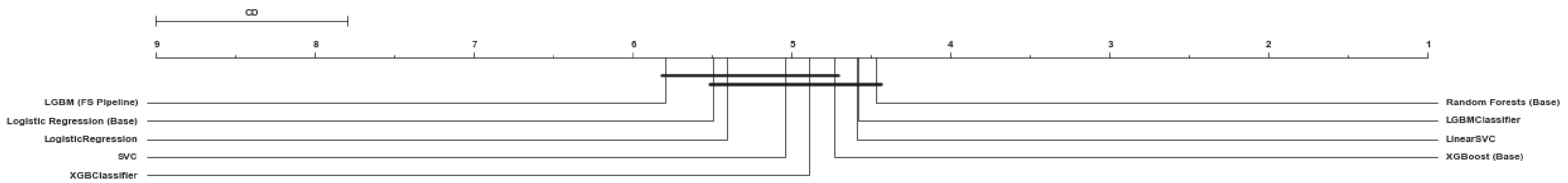


FIGURE A.20: Nemenyi's plot comparing ClinCA0h with strict feature selection models against baselines. LGBM (FS Hughes) model shows the highest median AUC score in the Hugues phenomenon study.

TABLE A.17: ClinCA0h with strict feature selection AUCs comparison table. LGBM (FS Hughes) model shows the highest median AUC score in the Hughes phenomenon study.

	meanrank	median	mad	ci_lower	ci_upper	effect_size	magnitude
Random Forests (Base)	4.47	0.77	0.08	0.71	0.86	0.00	negligible
LGBMClassifier	4.58	0.79	0.09	0.71	0.84	-0.13	negligible
LinearSVC	4.59	0.80	0.08	0.71	0.86	-0.22	small
XGBoost (Base)	4.74	0.80	0.08	0.71	0.86	-0.22	small
XGBClassifier	4.89	0.80	0.11	0.71	0.86	-0.19	negligible
SVC	5.04	0.80	0.07	0.74	0.86	-0.25	small
LogisticRegression	5.40	0.80	0.08	0.74	0.83	-0.25	small
Logistic Regression (Base)	5.50	0.80	0.08	0.74	0.88	-0.22	small
LGBM (FS Hughes)	5.80	0.86	0.08	0.76	0.91	-0.67	medium

A.10 ClinBiom0h

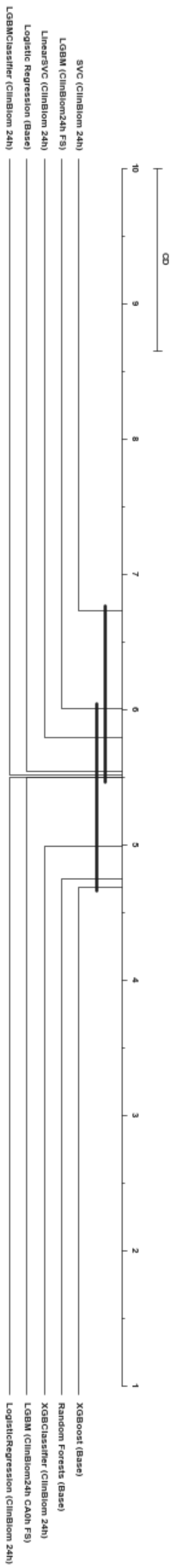


FIGURE A.21: Nemenyi's plot comparison ClinBiom24h based models.

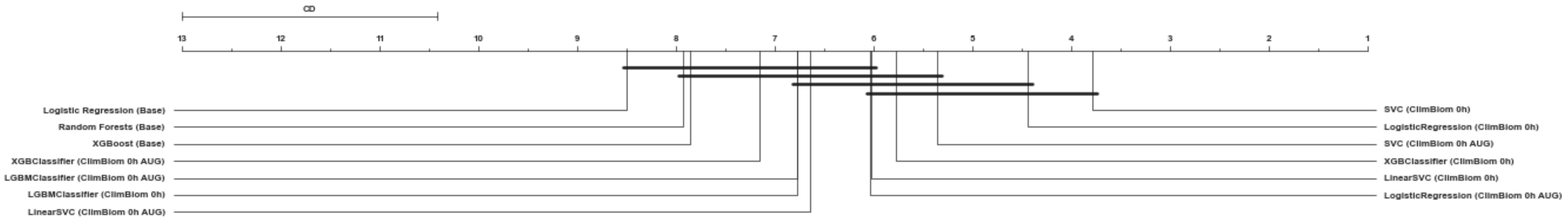


FIGURE A.22: Nemenyi's plot comparing best tabular ClinBiom0h against augmented models and clinical dataset models.

A.11 ClinBiom24h and ClinBiom24hCA0h Feature Selection

TABLE A.18: ClinBiom24h with strict feature selection and with HC AUCs comparison table.

	meanrank	median	mad	ci_lower	ci_upper	effect_size	magnitude
XGBoost (Base)	4.68	0.80	0.08	0.71	0.86	0.00	negligible
Random Forests (Base)	4.75	0.77	0.08	0.71	0.86	0.22	small
XGBClassifier (ClinBiom 24h)	4.99	0.79	0.12	0.68	0.88	0.05	negligible
LGBM (ClinBiom24h HC0h FS)	5.50	0.83	0.08	0.75	0.88	-0.26	small
LogisticRegression (ClinBiom 24h)	5.50	0.80	0.12	0.75	0.92	0.03	negligible
LGBMClassifier (ClinBiom 24h)	5.52	0.82	0.09	0.74	0.88	-0.13	negligible
Logistic Regression (Base)	5.54	0.80	0.08	0.74	0.88	0.00	negligible
LinearSVC (ClinBiom 24h)	5.79	0.80	0.12	0.75	0.92	0.03	negligible
LGBM (ClinBiom24h FS)	6.00	0.87	0.09	0.75	0.92	-0.52	medium
SVC (ClinBiom 24h)	6.73	0.88	0.08	0.79	0.92	-0.60	medium

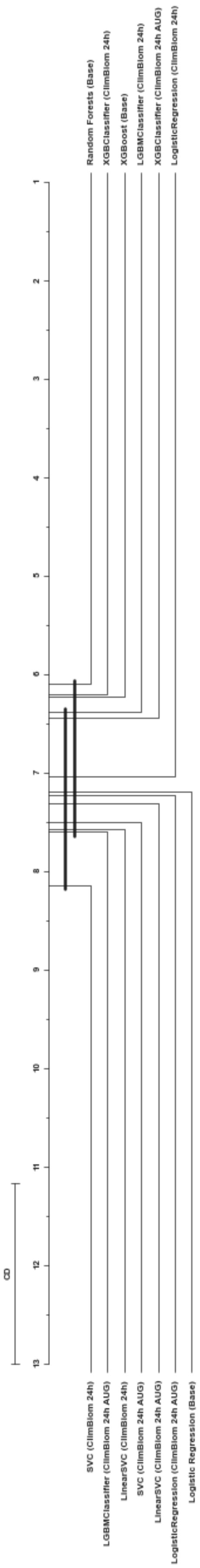


FIGURE A.23: Nemenyi's plot comparing ClinBiom24h models trained on augmented data.

A.12 Imaging Data Appendix

TABLE A.19: Best models AUC score ranks table.

	meanrank	median	mad	ci_lower	ci_upper	effect_size	magnitude
3D CNN (NO AUGs, Pat. = 100, ILR=0.01)	2.73	0.62	0.08	0.40	0.88	0.00	negligible
3D CNN (AUGs Vol., Pat.=200, BayesOpt)	4.85	0.69	0.07	0.54	0.84	-0.65	medium
XGBoost (Base)	7.08	0.74	0.06	0.66	0.94	-1.15	large
3D CNN (AUGs Vol., Pat.=200, HyperBand)	7.27	0.77	0.08	0.60	0.98	-1.30	large
Random Forests (Base)	7.67	0.75	0.08	0.66	0.94	-1.08	large
3D CNN (NO AUGs, Pat. = 15)	7.98	0.82	0.13	0.49	0.98	-1.21	large
LogisticRegression (Clin0hCA FS)	8.32	0.82	0.06	0.63	0.94	-1.84	large
LGBMClassifier (Clin0hCA FS)	8.52	0.80	0.09	0.59	1.00	-1.33	large
3D CNN (AUGs Vol., Pat. = 200, ILR=0.0001)	8.53	0.81	0.05	0.69	0.98	-1.84	large
Logistic Regression (Base)	8.77	0.80	0.07	0.69	0.97	-1.56	large
LogisticRegression (ClinBiom 24h)	9.35	0.83	0.08	0.63	1.00	-1.74	large
SVC (ClinBiom 24h)	10.12	0.88	0.04	0.71	1.00	-2.66	large
3D CNN (AUGs Rot., Pat. = 200, ILR=0.0001)	10.42	0.86	0.09	0.58	1.00	-1.87	large
3D CNN (NO AUGs, Pat. = 30)	11.62	0.93	0.08	0.67	1.00	-2.62	large
3D CNN (NO AUGs, Pat. = 100, ILR=0.001)	13.52	0.97	0.03	0.80	1.00	-3.81	large
3D CNN (No AUGs, Pat.=100, Full TranferL)	15.10	0.99	0.01	0.87	1.00	-4.31	large
3D CNN (NO AUGs, Pat. = 100, ILR=0.0001)	15.88	1.00	0.00	0.90	1.00	-4.47	large
3D CNN (NO AUGs, Pat. = 200, ILR=0.0001)	16.12	1.00	0.00	0.91	1.00	-4.47	large
3D CNN (No AUGs, Pat.=100, Basic TranferL)	16.17	1.00	0.00	0.93	1.00	-4.47	large

TABLE A.20: Best models weighted-F1 score ranks table.

	meanrank	median	mad	ci_lower	ci_upper	effect_size	magnitude
3D CNN (NO AUGs, Pat. = 100, ILR=0.01)	2.88	0.50	0.08	0.42	0.70	0.00	negligible
3D CNN (NO AUGs, Pat. = 15)	6.05	0.59	0.17	0.42	0.86	-0.48	small
3D CNN (AUGs Vol., Pat.=200, HyperBand)	6.37	0.64	0.08	0.51	0.85	-1.29	large
3D CNN (AUGs Vol., Pat. = 200, ILR=0.0001)	6.62	0.64	0.06	0.57	0.78	-1.43	large
XGBoost (Base)	7.45	0.72	0.05	0.60	0.82	-2.30	large
Logistic Regression (Base)	7.55	0.74	0.08	0.57	0.83	-2.17	large
Random Forests (Base)	7.58	0.72	0.07	0.59	0.83	-2.01	large
3D CNN (AUGs Rot., Pat. = 200, ILR=0.0001)	8.33	0.64	0.14	0.51	0.93	-0.88	large
LGBMClassifier (Clin0hCA FS)	8.72	0.75	0.10	0.59	0.92	-1.90	large
LogisticRegression (Clin0hCA FS)	8.73	0.74	0.08	0.60	0.92	-2.17	large
3D CNN (NO AUGs, Pat. = 30)	9.00	0.77	0.15	0.50	0.93	-1.60	large
SVC (ClinBiom 24h)	9.33	0.74	0.06	0.69	0.90	-2.44	large
LogisticRegression (ClinBiom 24h)	9.63	0.78	0.09	0.69	0.90	-2.34	large
3D CNN (NO AUGs, Pat. = 100, ILR=0.001)	10.00	0.84	0.09	0.62	0.93	-2.85	large
3D CNN (NO AUGs, Pat. = 100, ILR=0.0001)	13.65	0.93	0.07	0.76	1.00	-3.89	large
3D CNN (NO AUGs, Pat. = 200, ILR=0.0001)	14.10	0.93	0.07	0.79	1.00	-3.94	large

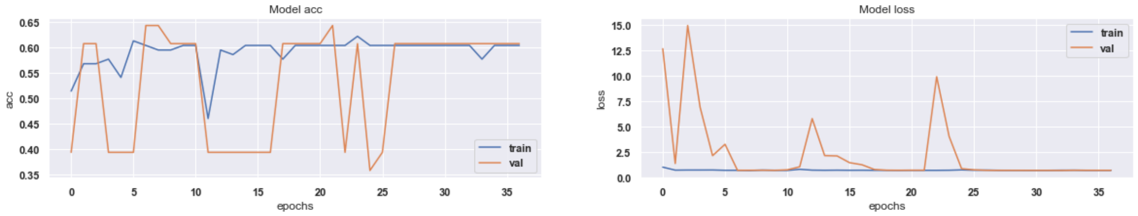


FIGURE A.24: Training and validation accuracy and loss evolution through 3D CNN base model training with simple rotations, set to $patience = 30$ and $ILR = 0.01$.

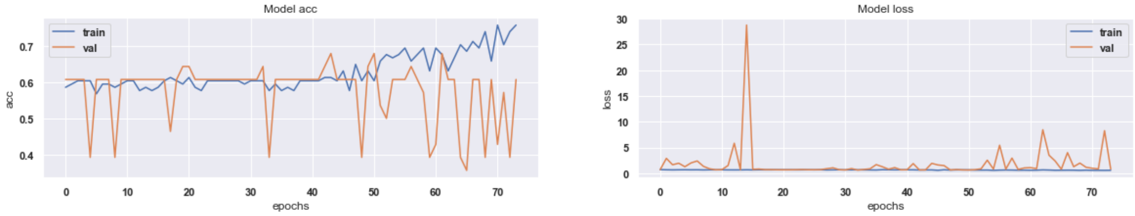


FIGURE A.25: Training and validation accuracy and loss evolution through 3D CNN base model training with simple rotations, set to *patience* = 30 and $ILR = 0.001$.

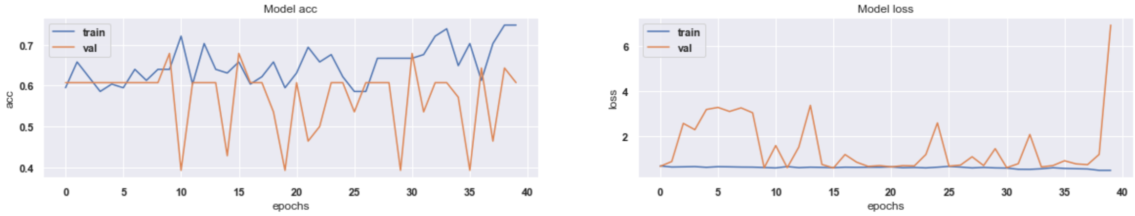


FIGURE A.26: Training and validation accuracy and loss evolution through 3D CNN base model training with simple rotations, set to *patience* = 30 and $ILR = 0.0001$.

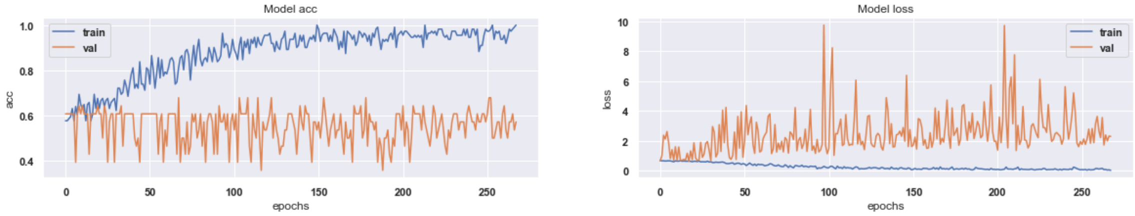


FIGURE A.27: Training and validation accuracy and loss evolution through 3D CNN base model training with simple rotations, set to *patience* = 200 and $ILR = 0.0001$.

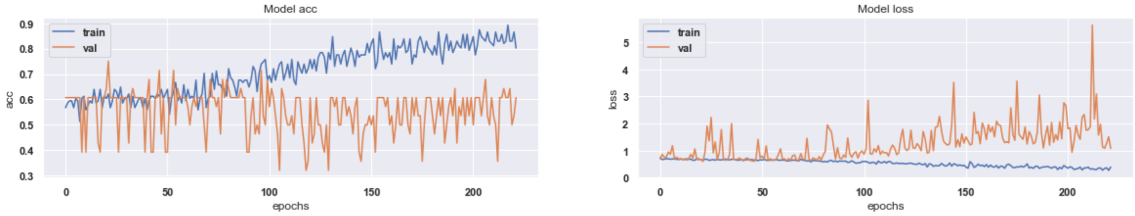


FIGURE A.28: Training and validation accuracy and loss evolution through 3D CNN base model training with full augmentation set, set to *patience* = 200 and $ILR = 0.0001$.

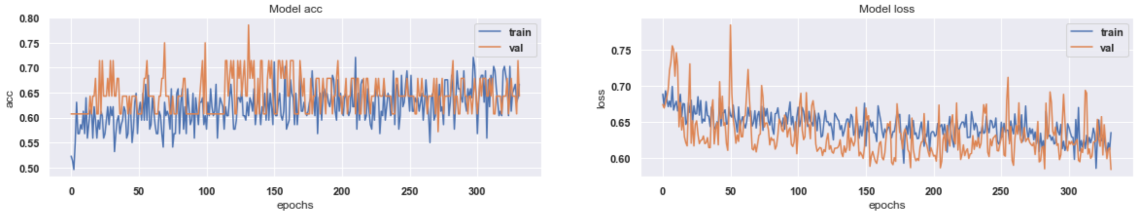


FIGURE A.29: Training and validation accuracy and loss evolution through Bayesian Optimization parametrized 3D CNN training with full augmentation set, set to *patience* = 200 and $ILR = 0.0001$.

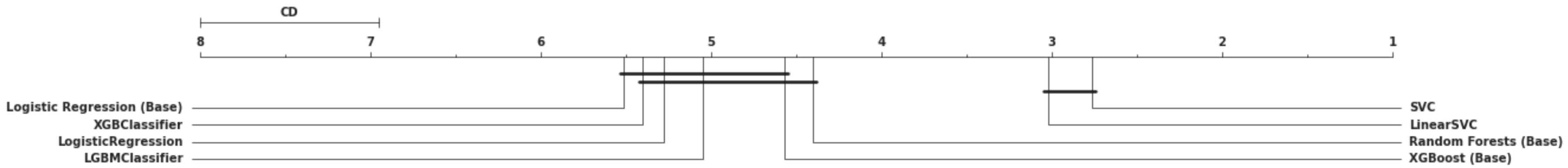


Figure A.30: Nemenyi's plot comparing models with absolute hemispheric contrast against the best base models.

A.13 Mixed Models Data Appendix

A.13.1 Models hyperparameters

TABLE A.21: Best ClinBiom0h hyperparameters.

Model	Parameters
LGBMClassifier	{'classifier__boosting_type': 'gbdt', 'classifier__learning_rate': 0.01, 'classifier__max_depth': 2, 'classifier__min_child_weight': 2}
XGBClassifier	{'classifier__booster': 'gbtree', 'classifier__learning_rate': 0.01, 'classifier__max_depth': 2, 'classifier__min_child_weight': 2}
LinearSVC	{'classifier__C': 0.01, 'classifier__dual': True, 'classifier__loss': 'hinge', 'classifier__penalty': 'l2'}
SVC	{'classifier__C': 0.01, 'classifier__coef0': -1, 'classifier__degree': 2, 'classifier__gamma': 'scale', 'classifier__kernel': 'rbf'}
LogisticRegression	{'classifier__C': 0.01, 'classifier__max_iter': 100, 'classifier__penalty': 'none', 'classifier__solver': 'newton-cg'}

TABLE A.22: Best ClinBiom0h trained on augmented data hyperparameters.

Model	Parameters
LGBMClassifier	{'classifier__boosting_type': 'gbdt', 'classifier__learning_rate': 0.01, 'classifier__max_depth': 2, 'classifier__min_child_weight': 2}
XGBClassifier	{'classifier__booster': 'gbtree', 'classifier__learning_rate': 0.01, 'classifier__max_depth': 2, 'classifier__min_child_weight': 2}
LinearSVC	{'classifier__C': 0.01, 'classifier__dual': True, 'classifier__loss': 'hinge', 'classifier__penalty': 'l2'}
SVC	{'classifier__C': 0.01, 'classifier__coef0': -1, 'classifier__degree': 2, 'classifier__gamma': 'scale', 'classifier__kernel': 'rbf'}
LogisticRegression	{'classifier__C': 0.01, 'classifier__max_iter': 100, 'classifier__penalty': 'none', 'classifier__solver': 'newton-cg'}

TABLE A.23: Best ClinBiom24h data hyperparameters.

Model	Parameters
LGBMClassifier	{'classifier__boosting_type': 'gbdt', 'classifier__learning_rate': 0.01, 'classifier__max_depth': 2, 'classifier__min_child_weight': 8}
XGBClassifier	{'classifier__booster': 'gblinear', 'classifier__learning_rate': 0.01, 'classifier__max_depth': 2, 'classifier__min_child_weight': 12}
LinearSVC	{'classifier__C': 0.01, 'classifier__dual': True, 'classifier__loss': 'squared_hinge', 'classifier__penalty': 'l2'}
SVC	{'classifier__C': 100, 'classifier__coef0': 0, 'classifier__degree': 4, 'classifier__gamma': 'scale', 'classifier__kernel': 'poly'}
LogisticRegression	{'classifier__C': 0.1, 'classifier__max_iter': 100, 'classifier__penalty': 'l2', 'classifier__solver': 'sag'}

TABLE A.24: Best ClinBiom24h trained on augmented data hyperparameters.

Model	Parameters
SVC	{'classifier__C': 10, 'classifier__coef0': 0, 'classifier__degree': 4, 'classifier__gamma': 'scale', 'classifier__kernel': 'poly'}
LogisticRegression	{'classifier__C': 0.01, 'classifier__max_iter': 100, 'classifier__penalty': 'l2', 'classifier__solver': 'sag'}
LGBMClassifier	{'classifier__boosting_type': 'dart', 'classifier__learning_rate': 0.01, 'classifier__max_depth': 2, 'classifier__min_child_weight': 12}
XGBClassifier	{'classifier__booster': 'gbtree', 'classifier__learning_rate': 0.01, 'classifier__max_depth': 2, 'classifier__min_child_weight': 12}
LinearSVC	{'classifier__C': 0.01, 'classifier__dual': True, 'classifier__loss': 'hinge', 'classifier__penalty': 'l2'}

A.13.2 Multivariate Analysis

TABLE A.25: Multivariate LR analysis of the clinical dataset. Variables sorted by their p-value.

	Estimate	Std. Error	z value	Pr($>\lvert z\rvert$)
LcTx2CHUSJ	NA	NA	NA	NA
mRSprv	1.73e+00	5.09e-01	3.40	**0.000678**
AHTA1	-3.01e+00	1.12e+00	-2.68	**0.007364**
NIHSS	1.39e-01	5.21e-02	2.67	**0.007644**
WkUp1	2.02e+00	8.05e-01	2.51	**0.01**
EAM1	3.62e+00	1.85e+00	1.96	**0.05**
ACoag1	-2.22e+00	1.29e+00	-1.72	**0.09**
Cortical1	-1.90e+00	1.13e+00	-1.68	**0.09**
IC1	1.49e+00	9.16e-01	1.63	**0.10**
ADM21	6.03e+00	3.76e+00	1.60	**0.11**
DAC1	-3.04e+00	2.00e+00	-1.52	**0.13**
Neuro1	1.48e+00	1.01e+00	1.46	**0.14**
Side2	-1.63e+00	1.11e+00	-1.46	**0.14**
DISLIP1	1.17e+00	8.93e-01	1.31	**0.19**
DPOC1	-1.51e+00	1.25e+00	-1.21	**0.22**
AntiDis1	-1.05e+00	8.78e-01	-1.20	**0.23**
AGE	2.94e-02	2.63e-02	1.12	**0.26**
DM1	-3.80e+00	3.56e+00	-1.07	**0.28**
Etiology4	-1.11e+00	1.08e+00	-1.03	**0.30**
Etiology5	-1.67e+00	1.69e+00	-0.99	**0.32**
Etiology2	8.54e-01	8.96e-01	0.95	**0.34**
Sex1	-5.86e-01	6.63e-01	-0.88	**0.38**
Hosp31	-1.29e+00	1.48e+00	-0.87	0.38
AVCprv1	-7.40e-01	9.66e-01	-0.77	0.44
(Intercept)	-1.90e+00	2.70e+00	-0.70	0.48
Etiology8	-1.72e+00	2.47e+00	-0.70	0.48
Hosp11	-1.42e+00	2.13e+00	-0.67	0.50
X.2ndHCHUSJ	-7.07e-01	1.19e+00	-0.59	0.55
DEM1	-6.66e-01	1.18e+00	-0.56	0.57
Hosp21	7.57e-01	1.39e+00	0.55	0.58
FA1	5.62e-01	1.14e+00	0.49	0.62
AAgreG1	-3.93e-01	9.17e-01	-0.43	0.67
HTA1	3.95e-01	9.89e-01	0.40	0.69
Etiology6	5.15e-01	1.40e+00	0.37	0.71
diff2ndH	1.19e-03	3.90e-03	0.31	0.76
diffCT1	-2.84e-03	9.94e-03	-0.29	0.77
DRC1	3.29e-01	1.50e+00	0.22	0.83
diff1stH	1.70e-03	8.19e-03	0.21	0.84
TxTp3	1.42e-01	8.29e-01	0.17	0.86
Prov22	-2.59e-01	1.77e+00	-0.15	0.88
Prov23	1.11e-01	1.38e+00	0.08	0.94
Etiology7	1.99e+01	1.25e+03	0.02	0.99
Outra1	9.76e-03	6.68e-01	0.02	0.99
Etiology3	1.65e+01	2.46e+03	0.007	0.99

TABLE A.26: Multivariate LR analysis of biomarkers at admission dataset. Variables sorted by their p-value.

	Estimate	Std. Error	z val	Pr($>\lvert z\rvert$)
BQ_AST	-1.53e-01	7.61e-02	-2.01	**0.04**
BQ_Glicemia	3.10e-02	1.57e-02	1.98	**0.05**
H_NLR	5.64e-01	2.92e-01	1.93	**0.05**
BQ_PCR	5.49e-02	2.93e-02	1.88	**0.06**
BQ_Ureia	4.53e-02	2.95e-02	1.53	**0.12**
BQ_ALT	-1.06e-01	7.20e-02	-1.47	**0.14**
BQ_TTP	-1.13e-01	8.26e-02	-1.37	**0.17**
BQ_FA	1.72e-02	1.77e-02	0.97	**0.33**
BQ_PT	7.03e-01	7.56e-01	0.93	**0.35**
BQ_Creatinina	-6.63e-01	7.17e-01	-0.92	**0.36**
BQ_INR	-6.95e+00	8.23e+00	-0.84	0.40
H_Eritr	-8.80e-01	1.32e+00	-0.66	0.51
H_Linf_perc	-3.75e+00	6.14e+00	-0.61	0.54
H_Neutr_perc	-3.71e+00	6.17e+00	-0.60	0.55
H_Hb	2.42e-01	4.04e-01	0.60	0.55
(Intercept)	3.66e+02	6.13e+02	0.60	0.55
H_Eos_perc	-4.07e+00	6.94e+00	-0.59	0.56
BQ_GGT	7.08e-03	1.21e-02	0.59	0.56
H_Bas_perc	-4.91e+00	9.51e+00	-0.52	0.60
H_Mono_perc	-2.95e+00	5.94e+00	-0.50	0.62
H_Mono_ABS	-2.73e+01	6.02e+01	-0.45	0.65
H_Neutr_ABS	-1.94e+01	6.26e+01	-0.31	0.76
H_Leuc	1.92e+01	6.22e+01	0.31	0.76
H_Linf_ABS	-1.78e+01	6.26e+01	-0.28	0.78
H_Eos_ABS	-2.0e+01	7.11e+01	-0.28	0.78
H_Bas_ABS	2.85e+01	1.13e+02	0.25	0.80
H_PLAQ	-7.55e-04	5.72e-03	-0.13	0.90

TABLE A.27: Multivariate LR analysis of biomarkers at follow-up with variables sorted by their p-value.

	Estimate	Std. Error	z value	Pr($>\lvert z\rvert$)
Hemo24_PLAQ	-0.02	0.007005	-2.99	**0.00279**
BQ24_TGL	0.03	0.02	2.01	**0.04**
Hemo24_Mono_AB	-18.80	9.38	-2.00	**0.04**
BQ24_Creatinina	-2.82	1.59	-1.78	**0.08**
Hemo24_Neutr_perc	-2.29	1.40	-1.63	**0.10**
(Intercept)	225.06	137.95	1.63	**0.10**
BQ24_VLDL	-0.18	0.12	-1.52	**0.13**
Hemo24_Linf_perc	-2.08	1.39	-1.49	**0.14**
BQ24_TSH	-0.64	0.43	-1.48	**0.14**
Hemo24_Eos_perc	-3.44	2.38	-1.45	**0.15**
Hemo24_Bas_AB	118.51	83.39	1.42	**0.16**
BQ24_GGT	-0.008288	0.006081	-1.36	**0.17**
Hemo24_Eritrocitos	-1.49	1.11	-1.34	**0.18**
BQ24_ALT	-0.07	0.05	-1.28	**0.20**
BQ24_T4L	2.46	2.19	1.12	**0.26**
BQ24_Ureia	0.03	0.03	1.08	**0.28**
Hemo24_Bas_perc	-6.97	6.76	-1.03	**0.30**
BQ24_FA	0.01	0.01	1.01	**0.31**
Hemo24_Linf_AB	-8.55	8.49	-1.01	**0.31**
Hemo24_Mono_perc	-1.35	1.41	-0.95	**0.34**
Hemo24_Leucocitos	6.52	7.75	0.84	0.40
BQ24_Hb_A1C	0.38	0.47	0.80	0.42
BQ24_AST	0.04	0.06	0.77	0.44
BQ24_Glicemia	0.005817	0.008022	0.72	0.47
Hemo24_Neut_AB	-5.08	7.73	-0.66	0.51
Hemo24_NLR	-0.09	0.15	-0.58	0.56
BQ24_PCR	0.009024	0.02	0.51	0.61
BQ24_CT	-0.02	0.05	-0.48	0.63
Hemo24_Eos_AB	9.54	22.89	0.42	0.68
BQ24_HDL	0.02	0.06	0.32	0.75
Hemo24_HB	0.07	0.38	0.19	0.85
BQ24_LDL	0.008471	0.05	0.16	0.87
BQ24_Ac_urico	0.01	0.11	0.13	0.90

Bibliography

[1] Jina-Ai, “Jina-ai/dalle-flow: A human-in-the-loop workflow for creating hd images from text.” [Online]. Available: https://github.com/jina-ai/dalle-flow [Cited on page iii.]

[2] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, “Language models are few-shot learners,” 2020. [Online]. Available: https://arxiv.org/abs/2005.14165 [Cited on pages iii and 12.]

[3] W. H. Organization, “Global health estimates 2020: Deaths by cause, age, sex, by country and by region, 2000-2019.” 2020. [Online]. Available: https://www.who.int/data/gho/data/themes/mortality-and-global-health-estimates/ghe-leading-causes-of-death [Cited on page 1.]

[4] R. L. Sacco, S. E. Kasner, J. P. Broderick, L. R. Caplan, J. B. Connors, A. Culebras, M. S. Elkind, M. G. George, A. D. Hamdan, R. T. Higashida, B. L. Hoh, L. S. Janis, C. S. Kase, D. O. Kleindorfer, J.-M. Lee, M. E. Moseley, E. D. Peterson, T. N. Turan, A. L. Valderrama, and H. V. Vinters, “An updated definition of stroke for the 21st century,” *Stroke*, vol. 44, no. 7, pp. 2064–2089, Jul. 2013. [Online]. Available: https://doi.org/10.1161/str.0b013e318296aeca [Cited on page 1.]

[5] M. Correia, M. R. Silva, R. Magalhaes, L. Guimaraes, and M. C. Silva, “Transient ischemic attacks in rural and urban northern portugal,” *Stroke*, vol. 37, no. 1, pp. 50–55, jan 2006. [Online]. Available: https://doi.org/10.1161/01.str.0000195209.26543.8f [Cited on page 1.]

[6] "Stroke treatments," Aug 2022. [Online]. Available: https://www.stroke.org.uk/what-is-stroke/diagnosis-to-discharge/treatment [Cited on page 1.]

[7] P. Khandelwal, D. R. Yavagal, and R. L. Sacco, "Acute ischemic stroke intervention," vol. 67, no. 22, pp. 2631–2644, Jun. 2016. [Online]. Available: https://doi.org/10.1016/j.jacc.2016.03.555 [Cited on page 1.]

[8] Y.-W. Chen, S.-F. Sung, C.-H. Chen, S.-C. Tang, L.-K. Tsai, H.-J. Lin, H.-Y. Huang, H. L. Po, Y. Sun, P.-L. Chen, and et al., "Intravenous thrombolysis administration 3–4.5 h after acute ischemic stroke: A retrospective, multicenter study," *Frontiers in Neurology*, vol. 10, 2019. [Cited on page 1.]

[9] T. Truelsen, B. Piechowski-Jozwiak, R. Bonita, C. Mathers, J. Bogousslavsky, and G. Boysen, "Stroke incidence and prevalence in europe: a review of available data," vol. 13, no. 6, pp. 581–598, Jun. 2006. [Online]. Available: https://doi.org/10.1111/j.1468-1331.2006.01138.x [Cited on page 1.]

[10] G. J. Hankey, K. Jamrozik, R. J. Broadhurst, S. Forbes, and C. S. Anderson, "Long-term disability after first-ever stroke and related prognostic factors in the perth community stroke study, 1989–1990," vol. 33, no. 4, pp. 1034–1040, Apr. 2002. [Online]. Available: https://doi.org/10.1161/01.str.0000012515.66889.24 [Cited on page 1.]

[11] M. Monteiro, A. C. Fonseca, A. T. Freitas, T. P. e Melo, A. P. Francisco, J. M. Ferro, and A. L. Oliveira, "Using machine learning to improve the prediction of functional outcome in ischemic stroke patients," vol. 15, no. 6, pp. 1953–1959, Nov. 2018. [Online]. Available: https://doi.org/10.1109/tcbb.2018.2811471 [Cited on page 2.]

[12] P. J. F. Lucas and A. Abu-Hanna, "Prognostic models in medicine," p. 1–5, 2001. [Online]. Available: http://dx.doi.org/10.1055/s-0038-1634456 [Cited on pages 2 and 27.]

[13] A. N. Mahmoud, A. A. Bavry, and I. Y. Elgendy, "The risk for stroke with aspiration thrombectomy: Procedure or patient related?" p. 1750–1752, Aug 2016. [Online]. Available: http://dx.doi.org/10.1016/j.jcin.2016.06.011 [Cited on pages 2 and 27.]

[14] B. C. Campbell, M. D. Hill, M. Rubiera, B. K. Menon, A. Demchuk, G. A. Donnan, D. Roy, J. Thornton, L. Dorado, A. Bonafe, E. I. Levy, H.-C. Diener, M. Hernández-Pérez, V. M. Pereira, J. Blasco, H. Quesada, J. Rempel, R. Jahan, S. M. Davis, B. C. Stouch, P. J. Mitchell, T. G. Jovin, J. L. Saver, and M. Goyal, "Safety and efficacy of solitaire stent thrombectomy," p. 798–806, Mar 2016. [Online]. Available: http://dx.doi.org/10.1161/STROKEAHA.115.012360 [Cited on page 27.]

[15] "Stroke thrombectomy complication management." [Cited on page 2.]

[16] C. G. Patil, E. F. Long, and M. G. Lansberg, "Cost-effectiveness analysis of mechanical thrombectomy in acute ischemic stroke," p. 508–513, Mar 2009. [Online]. Available: http://dx.doi.org/10.3171/2008.8.JNS08133 [Cited on pages 2 and 27.]

[17] B. C. Meyer and P. D. Lyden, "The modified national institutes of health stroke scale: Its time has come," *International Journal of Stroke*, vol. 4, no. 4, p. 267–273, 2009. [Cited on page 2.]

[18] B. K. Menon, V. Puetz, P. Kochar, and A. M. Demchuk, "Aspects and other neuroimaging scores in the triage and prediction of outcome in acute stroke patients," p. 407–423, May 2011. [Online]. Available: http://dx.doi.org/10.1016/j.nic.2011.01.007 [Cited on page 2.]

[19] K. Aho, P. Harmsen, S. Hatano, J. Marquardsen, V. E. Smirnov, and T. Strasser, "Cerebrovascular disease in the community: Results of a who collaborative study," 1980. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2395897/ [Cited on page 5.]

[20] INE, *Instituto Nacional de Estatística - Statistical Yearbook of Portugal: 2021*. INE, 2022. [Cited on page 5.]

[21] S. Silva and M. Gouveia, "Program "via verde do AVC": analysis of the impact on stroke mortality," *Revista Portuguesa de Saúde Pública*, vol. 30, no. 2, pp. 172–179, jul 2012. [Online]. Available: https://doi.org/10.1016/j.rpsp.2012.12.005 [Cited on page 5.]

[22] G. . Diseases and I. Collaborators*, "Global burden of 369 diseases and injuries in 204 countries and territories, 1990–2019: A systematic analysis for the global

burden of disease study 2019," *The Lancet*, vol. 396, no. 10258, pp. 1204–1222, Oct. 2020. [Online]. Available: https://doi.org/10.1016/S0140-6736(20)30925-9 [Cited on page 5.]

[23] M. B. R. Lagarteira, "Stroke laterality: impact on the time to admission and acute treatment," Porto, Portugal, 2021. [Online]. Available: https://hdl.handle.net/10216/134671 [Cited on page 5.]

[24] J. L. Banks and C. A. Marotta, "Outcomes validity and reliability of the modified rankin scale: Implications for stroke clinical trials," *Stroke*, vol. 38, no. 3, pp. 1091–1096, Mar. 2007. [Online]. Available: https://doi.org/10.1161/01.str.0000258355.23810.c6 [Cited on pages xxii, 6, and 95.]

[25] "NIHSS form." [Online]. Available: https://www.stroke.nih.gov/documents/NIH_Stroke_Scale_508C.pdf [Cited on page 6.]

[26] N. Kashani, "Aspects in stroke." [Online]. Available: http://aspectsinstroke.com/ [Cited on page 6.]

[27] S. Nagel, D. Sinha, D. Day, W. Reith, R. Chapot, P. Papanagiotou, E. A. Warburton, P. Guyler, S. Tysoe, K. Fassbender, S. Walter, M. Essig, J. Heidenrich, A. A. Konstas, M. Harrison, M. Papadakis, E. Greveson, O. Joly, S. Gerry, H. Maguire, C. Roffe, J. Hampton-Till, A. M. Buchan, and I. Q. Grunwald, "e-ASPECTS software is non-inferior to neuroradiologists in applying the ASPECT score to computed tomography scans of acute ischemic stroke patients," vol. 12, no. 6, pp. 615–622, Dec. 2016. [Online]. Available: https://doi.org/10.1177/1747493016681020 [Cited on page 6.]

[28] A. P. Jadhav, S. M. Desai, D. S. Liebeskind, and L. R. Wechsler, "Neuroimaging of acute stroke," vol. 38, no. 1, pp. 185–199, Feb. 2020. [Online]. Available: https://doi.org/10.1016/j.ncl.2019.09.004 [Cited on pages 7 and 8.]

[29] L. W. Goldman, "Principles of CT: Multislice CT," *Journal of Nuclear Medicine Technology*, vol. 36, no. 2, pp. 57–68, may 2008. [Online]. Available: https://doi.org/10.2967/jnmt.107.044826 [Cited on pages 7 and 8.]

[30] "Mango." [Online]. Available: https://rii.uthscsa.edu/mango/mango.html [Cited on pages xvii, 7, and 38.]

[31] L. Kobbelt, *Vision, modeling, and visualization 2006: Proceedings, November 22-24, 2006, Aachen, Germany*. Akademische Verlagsgesellschaft, 2006. [Cited on page 7.]

[32] R. A. Zimmerman, W. A. Gibby, and R. F. Carmody, *Neuroimaging : clinical and physical principles*. Springer Science and Business Media, 2000. [Online]. Available: https://doi.org/10.1007/978-1-4612-1152-5 [Cited on page 7.]

[33] Y. Baba, "Windowing (ct): Radiology reference article," Sep 2021. [Online]. Available: https://radiopaedia.org/articles/windowing-ct [Cited on page 7.]

[34] Jan. 2022. [Online]. Available: https://www.dicomstandard.org/ [Cited on page 8.]

[35] S. Nagel, D. Sinha, D. Day, W. Reith, R. Chapot, P. Papanagiotou, E. A. Warburton, P. Guyler, S. Tysoe, K. Fassbender, S. Walter, M. Essig, J. Heidenrich, A. A. Konstas, M. Harrison, M. Papadakis, E. Greveson, O. Joly, S. Gerry, H. Maguire, C. Roffe, J. Hampton-Till, A. M. Buchan, and I. Q. Grunwald, "DICOM PS3.1 2021e - introduction and overview," Jan. 2021. [Online]. Available: https://dicom.nema.org/medical/dicom/current/output/chtml/part10/chapter_7.html [Cited on pages 8 and 9.]

[36] H. Knipe and C. Moore, "NIfTI (file format)," Nov. 2019. [Online]. Available: https://doi.org/10.53347/rid-72562 [Cited on page 9.]

[37] B. Whitcher, V. J. Schmid, and A. Thorton, "Working with the dicom and nifti data standards in r," *Journal of Statistical Software*, vol. 44, no. 6, p. 1–29, 2011. [Online]. Available: https://www.jstatsoft.org/index.php/jss/article/view/v044i06 [Cited on page 9.]

[38] J. Muschelli, "Recommendations for processing head CT data," *Frontiers in Neuroinformatics*, vol. 13, Sep. 2019. [Online]. Available: https://doi.org/10.3389/fninf.2019.00061 [Cited on pages 9, 10, and 38.]

[39] "Gdcm release 2.0." [Online]. Available: http://gdcm.sourceforge.net/wiki/index.php/GDCM_Release_2.0 [Cited on page 10.]

[40] L. Snoek, S. Miletić, and H. S. Scholte, "How to control for confounds in decoding analyses of neuroimaging data," *NeuroImage*, vol. 184, pp. 741–760, jan 2019. [Online]. Available: https://doi.org/10.1016/j.neuroimage.2018.09.074 [Cited on page 10.]

[41] "Convolutional neural network," Jul 2022. [Online]. Available: https://en.wikipedia.org/wiki/Convolutional_neural_network [Cited on page 10.]

[42] A. e. a. Waibel, "Phoneme recognition using time-delay neural networks - acoustics ..." Mar 1989. [Online]. Available: https://www.cs.toronto.edu/~hinton/absps/waibelTDNN.pdf [Cited on page 10.]

[43] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT Press, 2016. [Cited on page 10.]

[44] S. J. Russell and P. Norvig, *Artificial Intelligence*. Prentice Hall, 2009. [Cited on page 11.]

[45] N. Fazakis, S. Karlos, S. Kotsiantis, and K. Sgarbas, "Self-trained lmt for semisupervised learning," p. 1–13, 2016. [Online]. Available: http://dx.doi.org/10.1155/2016/3057481 [Cited on page 11.]

[46] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, "Hierarchical text-conditional image generation with CLIP latents," 2022. [Online]. Available: https://arxiv.org/abs/2204.06125 [Cited on pages 12 and 22.]

[47] C. Schuhmann, R. Vencu, R. Beaumont, R. Kaczmarczyk, C. Mullis, A. Katta, T. Coombes, J. Jitsev, and A. Komatsuzaki, "Laion-400m: Open dataset of clip-filtered 400 million image-text pairs," 2021. [Online]. Available: https://arxiv.org/abs/2111.02114 [Cited on page 12.]

[48] J. Yu, Y. Xu, J. Y. Koh, T. Luong, G. Baid, Z. Wang, V. Vasudevan, A. Ku, Y. Yang, B. K. Ayan, B. Hutchinson, W. Han, Z. Parekh, X. Li, H. Zhang, J. Baldridge, and Y. Wu, "Scaling autoregressive models for content-rich text-to-image generation," 2022. [Online]. Available: https://arxiv.org/abs/2206.10789 [Cited on page 22.]

[49] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, and et al., "Highly accurate protein structure prediction with alphafold," *Nature*, vol. 596, no. 7873, p. 583–589, 2021. [Cited on page 12.]

[50] J. Hoffmann, S. Borgeaud, A. Mensch, E. Buchatskaya, T. Cai, E. Rutherford, D. d. L. Casas, L. A. Hendricks, J. Welbl, A. Clark, T. Hennigan, E. Noland, K. Millican, G. v. d. Driessche, B. Damoc, A. Guy, S. Osindero, K. Simonyan, E. Elsen, J. W. Rae,

O. Vinyals, and L. Sifre, “Training compute-optimal large language models,” 2022. [Online]. Available: https://arxiv.org/abs/2203.15556 [Cited on page 12.]

[51] C. Bishop, *Pattern recognition and machine learning*. Springer Verlag. [Cited on page 12.]

[52] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org. [Cited on pages 12, 17, 18, 20, 21, 22, 23, 24, and 25.]

[53] “Advantages and disadvantages of logistic regression,” Sep 2020. [Online]. Available: https://www.geeksforgeeks.org/advantages-and-disadvantages-of-logistic-regres⟩sion/ [Cited on pages 12 and 13.]

[54] G. James, D. Witten, T. Hastie, and R. Tibshirani, “An introduction to statistical learning,” Jul 2021. [Online]. Available: https://www.statlearning.com/ [Cited on pages 13, 14, 15, 16, 17, 18, 23, 25, and 42.]

[55] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011. [Cited on pages 13, 18, and 33.]

[56] B. Kégl, “The return of adaboost.mh: multi-class hamming trees,” 2013. [Online]. Available: https://arxiv.org/abs/1312.6086 [Cited on page 15.]

[57] D. Gera, “Boosting algorithms: Adaboost, gradient boosting, xgb, light gbm and catboost,” Sep 2020. [Online]. Available: https://medium.com/@divyagera2402/boosting-algorithms-adaboost-gradient-boosting-xgb-light-gbm-and-catboost-e7d2dbc4e4ca [Cited on page 15.]

[58] N. Fazakis, S. Karlos, S. Kotsiantis, and K. Sgarbas, “Self-trained lmt for semisupervised learning,” p. 1–13, 2016. [Online]. Available: http://dx.doi.org/10.1155/2016/3057481 [Cited on page 15.]

[59] T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery*

and Data Mining, ser. KDD '16. New York, NY, USA: ACM, 2016, pp. 785–794. [Online]. Available: http://doi.acm.org/10.1145/2939672.2939785 [Cited on pages 15 and 34.]

[60] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," *Advances in neural information processing systems*, vol. 30, pp. 3146–3154, 2017. [Cited on pages 15 and 34.]

[61] D. S. Jodas, N. Marranghello, A. S. Pereira, and R. C. Guido, "Comparing support vector machines and artificial neural networks in the recognition of steering angle for driving of mobile robots through paths in plantations," p. 240–249, 2013. [Online]. Available: http://dx.doi.org/10.1016/j.procs.2013.05.187 [Cited on page 16.]

[62] L. Jiang, Y. Cao, X. Yin, S. Ni, M. Li, C. Li, Z. Luo, H. Lu, and J. Hu, "A combinatorial method to visualize the neuronal network in the mouse spinal cord: combination of a modified golgi-cox method and synchrotron radiation micro-computed tomography," p. 477–489, Jan 2021. [Online]. Available: http://dx.doi.org/10.1007/s00418-020-01949-8 [Cited on page 17.]

[63] N. Sperelakis, *Cell physiology sourcebook: A molecular approach*. Academic, 2012. [Cited on page 17.]

[64] C. Snyder and S. Vishwanath, "Deep networks as logical circuits: Generalization and interpretation," 2020. [Online]. Available: https://arxiv.org/abs/2003.11619 [Cited on pages 18 and 22.]

[65] L. Ferreira, A. Pilastri, C. M. Martins, P. M. Pires, and P. Cortez, "A comparison of automl tools for machine learning, deep learning and xgboost," in *2021 International Joint Conference on Neural Networks (IJCNN)*, 2021, pp. 1–8. [Cited on pages 19 and 20.]

[66] M. Reif, F. Shafait, and A. Dengel, "Meta-learning for evolutionary parameter optimization of classifiers," *Machine Learning*, vol. 87, no. 3, pp. 357–380, apr 2012. [Online]. Available: https://doi.org/10.1007/s10994-012-5286-7 [Cited on page 19.]

[67] M. Feurer, A. Klein, K. Eggensperger, J. T. Springenberg, M. Blum, and F. Hutter, *Auto-sklearn: Efficient and Robust Automated Machine Learning*. Cham: Springer International Publishing, 2019, pp. 113–134. [Online]. Available: https://doi.org/10.1007/978-3-030-05318-5_6 [Cited on page 19.]

[68] C. Thornton, F. Hutter, H. H. Hoos, and K. Leyton-Brown, "Auto-WEKA: Combined selection and hyperparameter optimization of classification algorithms," in *Proc. of KDD-2013*, 2013, pp. 847–855. [Cited on page 19.]

[69] D. S. C. UMIT, *Automated machine learning: A beginner's Guide to Building Automated Machine Learning Systems ... using automl and python*. PACKT Publishing Limited, 2018. [Cited on page 19.]

[70] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011. [Cited on pages 19 and 34.]

[71] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. VanderPlas, A. Joly, B. Holt, and G. Varoquaux, "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. [Cited on page 19.]

[72] M. Feurer, K. Eggensperger, S. Falkner, M. Lindauer, and F. Hutter, "Auto-Sklearn 2.0: Hands-free AutoML via meta-learning," 2020. [Online]. Available: https://arxiv.org/abs/2007.04074 [Cited on pages 19, 33, and 34.]

[73] L. Zimmer, M. Lindauer, and F. Hutter, "Auto-pytorch tabular: Multi-fidelity metalearning for efficient and robust autodl," *arXiv preprint arXiv:2006.13799*, 2020. [Cited on pages 19 and 33.]

[74] N. Erickson, J. Mueller, A. Shirkov, H. Zhang, P. Larroy, M. Li, and A. Smola, "AutoGluon-tabular: Robust and accurate AutoML for structured data," 2020. [Online]. Available: https://arxiv.org/abs/2003.06505 [Cited on pages 20 and 33.]

[75] H2O.ai, *H2O: Scalable Machine Learning Platform*, 2020, version 3.30.0.6. [Online]. Available: https://github.com/h2oai/h2o-3 [Cited on page 20.]

[76] William H. Press, etc., B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical recipes in pascal (first edition)*. Cambridge, England: Cambridge University Press, Oct. 1989. [Cited on page 20.]

[77] G. R. C. and R. E. Woods, *Digital Image Processing*. Prentice Hall, 2008. [Cited on page 20.]

[78] S. P. Singh, L. Wang, S. Gupta, H. Goli, P. Padmanabhan, and B. Gulyás, “3d deep learning on medical images: A review,” p. 5097, Sep 2020. [Online]. Available: http://dx.doi.org/10.3390/s20185097 [Cited on pages 21, 22, 23, and 30.]

[79] F. Chollet *et al.* (2015) Keras. [Online]. Available: https://github.com/fchollet/keras [Cited on pages 21, 22, 24, 33, and 34.]

[80] Z. Wang, A. R. Childress, J. Wang, and J. A. Detre, “Support vector machine learning-based fMRI data group analysis,” *NeuroImage*, vol. 36, no. 4, pp. 1139–1151, jul 2007. [Online]. Available: https://doi.org/10.1016/j.neuroimage.2007.03.072 [Cited on page 22.]

[81] R. Thoppilan, D. De Freitas, J. Hall, N. Shazeer, A. Kulshreshtha, H.-T. Cheng, A. Jin, T. Bos, L. Baker, Y. Du, Y. Li, H. Lee, H. S. Zheng, A. Ghafouri, M. Menegali, Y. Huang, M. Krikun, D. Lepikhin, J. Qin, D. Chen, Y. Xu, Z. Chen, A. Roberts, M. Bosma, V. Zhao, Y. Zhou, C.-C. Chang, I. Krivokon, W. Rusch, M. Pickett, P. Srinivasan, L. Man, K. Meier-Hellstern, M. R. Morris, T. Doshi, R. D. Santos, T. Duke, J. Soraker, B. Zevenbergen, V. Prabhakaran, M. Diaz, B. Hutchinson, K. Olson, A. Molina, E. Hoffman-John, J. Lee, L. Aroyo, R. Rajakumar, A. Butryna, M. Lamm, V. Kuzmina, J. Fenton, A. Cohen, R. Bernstein, R. Kurzweil, B. Aguera-Arcas, C. Cui, M. Croak, E. Chi, and Q. Le, “LaMDA: Language Models for Dialog Applications,” 2022. [Online]. Available: https://arxiv.org/abs/2201.08239 [Cited on page 12.]

[82] A. Kratsios and E. Bilokopytov, “Non-euclidean universal approximation,” 2020. [Online]. Available: https://arxiv.org/abs/2006.02341 [Cited on page 22.]

[83] D.-X. Zhou, “Universality of deep convolutional neural networks,” 2018. [Online]. Available: https://arxiv.org/abs/1805.10769 [Cited on page 22.]

[84] D. Downing and D. Downing, *Barron’s E-Z Calculus*. Barron’s Educational Series, Inc., 2010. [Cited on page 22.]

[85] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014. [Online]. Available: https://arxiv.org/abs/1412.6980 [Cited on page 23.]

[86] M. Gilg, "Representation of networks of wireless sensors with a grayscale image: Application to routing," p. 31–66, 2017. [Online]. Available: http://dx.doi.org/10.1016/B978-1-78548-274-8.50002-7 [Cited on page 23.]

[87] S. Bozinovski, "Reminder of the first paper on transfer learning in neural networks, 1976," *Informatica*, vol. 44, no. 3, 2020. [Cited on page 24.]

[88] A. Zheng, "Evaluating machine learning models direct textbook," Sep 2015. [Online]. Available: https://www.directtextbook.com/isbn/9781491932445 [Cited on page 24.]

[89] K. H. Brodersen, C. S. Ong, K. E. Stephan, and J. M. Buhmann, "The balanced accuracy and its posterior distribution," in *2010 20th International Conference on Pattern Recognition*. IEEE, Aug. 2010. [Online]. Available: https://doi.org/10.1109/icpr.2010.764 [Cited on page 24.]

[90] A. M. R. d. S. M. Pereira, "Classification methods applied to the diagnosis of obstructive sleep apnea - a comparative study," Oct 2021. [Online]. Available: http://hdl.handle.net/10451/49338 [Cited on page 25.]

[91] S. M. P.-P. E. I. P. T. D. J. K. J. D. H. J. G. P. Mordasini, "Stroke thrombectomy complication management." *Journal of NeuroInterventional Surgery*, 2021. [Cited on page 27.]

[92] S. Anderson and N. J. Marlett, "The language of recovery: How effective communication of information is crucial to restructuring post-stroke life," p. 55–67, Oct 2004. [Online]. Available: http://dx.doi.org/10.1310/NPC4-01YV-P66Q-VM9R [Cited on page 27.]

[93] G. Ntaios, M. Faouzi, J. Ferrari, W. Lang, K. Vemmos, and P. Michel, "An integer-based score to predict functional outcome in acute ischemic stroke: The ASTRAL score," *Neurology*, vol. 78, no. 24, pp. 1916–1922, May 2012. [Online]. Available: https://doi.org/10.1212/wnl.0b013e318259e221 [Cited on page 27.]

[94] D. Strbian, A. Meretoja, F. J. Ahlhelm, J. Pitkaniemi, P. Lyrer, M. Kaste, S. Engelter, and T. Tatlisumak, "Predicting outcome of IV thrombolysis-treated ischemic stroke patients: The DRAGON score," *Neurology*, vol. 78, no. 6, pp. 427–432, Feb.

2012. [Online]. Available: https://doi.org/10.1212/wnl.0b013e318245d2a9 [Cited on page 27.]

[95] A. C. Flint, B. S. Faigeles, S. P. Cullen, H. Kamel, V. A. Rao, R. Gupta, W. S. Smith, P. M. Bath, G. A. Donnan, K. Lees, A. Alexandrov, P. Bath, E. Bluhmki, N. Bornstein, L. Claesson, S. Davis, G. Donnan, H. Diener, M. Fisher, B. Gregson, J. Grotta, W. Hacke, M. Hennerici, M. Hommel, M. Kaste, P. Lyden, J. Marler, K. Muir, R. Sacco, A. Shuaib, P. Teal, N. Wahlgren, S. Warach, and C. W. and, "THRIVE score predicts ischemic stroke outcomes and thrombolytic hemorrhage risk in VISTA," *Stroke*, vol. 44, no. 12, pp. 3365–3369, Dec. 2013. [Online]. Available: https://doi.org/10.1161/strokeaha.113.002794 [Cited on page 27.]

[96] E. Venema, M. J. H. L. Mulder, B. Roozenbeek, J. P. Broderick, S. D. Yeatts, P. Khatri, O. A. Berkhemer, B. J. Emmer, Y. B. W. E. M. Roos, C. B. L. M. Majoie, R. J. van Oostenbrugge, W. H. van Zwam, A. van der Lugt, E. W. Steyerberg, D. W. J. Dippel, and H. F. Lingsma, "Selection of patients for intra-arterial treatment for acute ischaemic stroke: development and validation of a clinical decision tool in two randomised trials," *BMJ*, p. j1710, May 2017. [Online]. Available: https://doi.org/10.1136/bmj.j1710 [Cited on page 27.]

[97] G. Turc, M. Apoil, O. Naggara, D. Calvet, C. Lamy, A. M. Tataru, J.-F. Méder, J.-L. Mas, J.-C. Baron, C. Oppenheim, and E. Touzé, "Magnetic resonance imaging-DRAGON score," *Stroke*, vol. 44, no. 5, pp. 1323–1328, May 2013. [Online]. Available: https://doi.org/10.1161/strokeaha.111.000127 [Cited on page 27.]

[98] C.-W. Ryu, B. M. Kim, H.-G. Kim, J. H. Heo, H. S. Nam, D. J. Kim, and Y. D. Kim, "Optimizing outcome prediction scores in patients undergoing endovascular thrombectomy for large vessel occlusions using collateral grade on computed tomography angiography," *Neurosurgery*, vol. 85, no. 3, pp. 350–358, Jul. 2018. [Online]. Available: https://doi.org/10.1093/neuros/nyy316 [Cited on page 27.]

[99] A. Sarraj, K. Albright, A. D. Barreto, A. K. Boehme, C. W. Sitton, J. Choi, S. L. Lutzker, C.-H. J. Sun, W. Bibars, C. B. Nguyen, O. Mir, F. Vahidy, T.-C. Wu, G. A. Lopez, N. R. Gonzales, R. Edgell, S. Martin-Schild, H. Hallevi, P. R. Chen, M. Dannenbaum, J. L. Saver, D. S. Liebeskind, R. G. Nogueira, R. Gupta, J. C. Grotta, and S. I. Savitz, "Optimizing prediction scores for

poor outcome after intra-arterial therapy in anterior circulation acute ischemic stroke," *Stroke*, vol. 44, no. 12, pp. 3324–3330, Dec. 2013. [Online]. Available: https://doi.org/10.1161/strokeaha.113.001050 [Cited on page 27.]

[100] B. V. Calster, , D. J. McLernon, M. van Smeden, L. Wynants, and E. W. Steyerberg, "Calibration: the achilles heel of predictive analytics," *BMC Medicine*, vol. 17, no. 1, Dec. 2019. [Online]. Available: https://doi.org/10.1186/s12916-019-1466-7 [Cited on page 28.]

[101] M. Monteiro, A. C. Fonseca, A. T. Freitas, T. P. e Melo, A. P. Francisco, J. M. Ferro, and A. L. Oliveira, "Using machine learning to improve the prediction of functional outcome in ischemic stroke patients," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 15, no. 6, pp. 1953–1959, Nov. 2018. [Online]. Available: https://doi.org/10.1109/tcbb.2018.2811471 [Cited on page 28.]

[102] M. Correia, I. Silva, D. Gabriel, J. Simrén, A. Carneiro, S. Ribeiro, H. M. Dória, R. Varela, A. Aires, K. Minta, R. Antunes, R. Felgueiras, P. Castro, K. Blenow, R. Magalhães, H. Zetterberg, and L. F. Maia, "Early plasma biomarker dynamic profiles are associated with acute ischemic stroke outcomes," *European Journal of Neurology*, Feb. 2022. [Online]. Available: https://doi.org/10.1111/ene.15273 [Cited on page 28.]

[103] L. Cui, S. Han, S. Qi, Y. Duan, Y. Kang, and Y. Luo, "Deep symmetric three-dimensional convolutional neural networks for identifying acute ischemic stroke via diffusion-weighted images," *Journal of X-Ray Science and Technology*, vol. 29, no. 4, p. 551–566, 2021. [Cited on pages 28 and 29.]

[104] J. Li, P. Zhang, Y. Liu, W. Chen, X. Yi, and C. Wang, "Stroke lateralization in large hemisphere infarctions: Characteristics, stroke-related complications, and outcomes," *Frontiers in Neurology*, vol. 12, 2021. [Cited on page 28.]

[105] S. Chen, K. Ma, and Y. Zheng, "Med3d: Transfer learning for 3d medical image analysis," *arXiv preprint arXiv:1904.00625*, 2019. [Cited on page 29.]

[106] C.-M. Lo, P.-H. Hung, and D.-T. Lin, "Rapid assessment of acute ischemic stroke by computed tomography using deep convolutional neural networks," *Journal of Digital Imaging*, 2021. [Cited on page 29.]

[107] A. Ertl, A. Franz, B. Schmitz, and M. Braun, "3d cnn-based identification of hyperdensities in cranial non-contrast ct after thrombectomy," *Informatik aktuell*, p. 309–314, 2022. [Cited on page 29.]

[108] K. Kamnitsas, C. Ledig, V. F. Newcombe, J. P. Simpson, A. D. Kane, D. K. Menon, D. Rueckert, and B. Glocker, "Efficient multi-scale 3d cnn with fully connected crf for accurate brain lesion segmentation," p. 61–78, Feb 2017. [Online]. Available: http://dx.doi.org/10.1016/j.media.2016.10.004 [Cited on page 29.]

[109] "Isles challenge 2022 ischemic stroke lesion segmentation." [Online]. Available: https://www.isles-challenge.org/ [Cited on pages 29 and 49.]

[110] "Section for biomedical image analysis (sbia)." [Online]. Available: https://www.med.upenn.edu/sbia/miccai-brats-2018-previous-brats-challenges.html [Cited on page 29.]

[111] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017. [Online]. Available: https://arxiv.org/abs/1706.03762 [Cited on page 29.]

[112] G. Hacohen and D. Weinshall, "On the power of curriculum learning in training deep networks," 2019. [Online]. Available: https://arxiv.org/abs/1904.03626 [Cited on page 30.]

[113] C. Zhou, C. Ding, X. Wang, Z. Lu, and D. Tao, "One-pass multi-task networks with cross-task guided attention for brain tumor segmentation," p. 4516–4529, 2020. [Online]. Available: http://dx.doi.org/10.1109/TIP.2020.2973510 [Cited on page 30.]

[114] F. Milletari, S.-A. Ahmadi, C. Kroll, A. Plate, V. Rozanski, J. Maiostre, J. Levin, O. Dietrich, B. Ertl-Wagner, K. Bötzel, and N. Navab, "Hough-cnn: Deep learning for segmentation of deep brain regions in mri and ultrasound," p. 92–102, Nov 2017. [Online]. Available: http://dx.doi.org/10.1016/j.cviu.2017.04.002 [Cited on page 30.]

[115] O. Öman, T. Mäkelä, E. Salli, S. Savolainen, and M. Kangasniemi, "3d convolutional neural networks applied to ct angiography in the detection of acute ischemic stroke," *European Radiology Experimental*, vol. 3, no. 1, 2019. [Cited on page 30.]

[116] M. L. Tolhuisen, J. W. Hoving, M. S. Koopman, M. Kappelhof, H. van Voorst, A. E. Bruggeman, A. M. Demchuck, D. W. Dippel, B. J. Emmer, S. Bracard, and et al., "Outcome prediction based on automatically extracted infarct core image features in patients with acute ischemic stroke," *Diagnostics*, vol. 12, no. 8, p. 1786, 2022. [Cited on page 30.]

[117] K. Müller, *here: A Simpler Way to Find Your Files*, 2022, r package version 4.0.3. [Online]. Available: https://CRAN.R-project.org/package=here [Cited on page 33.]

[118] RStudio Team, *RStudio: Integrated Development Environment for R*, RStudio, PBC., Boston, MA, 2022. [Online]. Available: http://www.rstudio.com/ [Cited on page 33.]

[119] B. Cui, *DataExplorer*, 2020, https://github.com/boxuancui/DataExplorer. [Cited on page 33.]

[120] B. Schloerke, *GGally*, 2021, https://ggobi.github.io/ggally/, https://cran.r-project.org/web/packages/GGally/GGally.pdf. [Cited on page 33.]

[121] H. Wickham, *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2016. [Online]. Available: https://ggplot2.tidyverse.org [Cited on page 33.]

[122] A. Kassambara, *ggpubr*, 2020, https://cran.r-project.org/web/packages/ggpubr/ggpubr.pdf. [Cited on page 33.]

[123] J. Arnold, *ggthemes*, 2020, https://github.com/jrnold/ggthemes. [Cited on page 33.]

[124] D. D. Sjoberg, K. Whiting, M. Curry, J. A. Lavery, and J. Larmarange, "Reproducible summary tables with the gtsummary package," *The R Journal*, vol. 13, pp. 570–580, 2021. [Online]. Available: https://doi.org/10.32614/RJ-2021-053 [Cited on pages 33 and 39.]

[125] G. Grolemund and H. Wickham, "Dates and times made easy with lubridate," *Journal of Statistical Software*, vol. 40, no. 3, pp. 1–25, 2011. [Online]. Available: https://www.jstatsoft.org/v40/i03/ [Cited on page 33.]

[126] W. N. Venables and B. D. Ripley, *Modern Applied Statistics with S*, 4th ed. New York: Springer, 2002, iSBN 0-387-95457-0. [Online]. Available: https://www.stats.ox.ac.uk/pub/MASS4/ [Cited on page 33.]

[127] S. Wang, W. Li, L. Hu, J. Cheng, H. Yang, and Y. Liu, "NAguideR: performing and prioritizing missing value imputations for consistent bottom-up proteomic analyses," *Nucleic Acids Research*, vol. 48, no. 14, pp. e83–e83, jun 2020. [Online]. Available: https://doi.org/10.1093/nar/gkaa498 [Cited on pages 33, 37, and 61.]

[128] H. Wickham, J. Hester, and J. Bryan, *readr: Read Rectangular Text Data*, 2022, https://readr.tidyverse.org, https://github.com/tidyverse/readr. [Cited on page 33.]

[129] H. Wickham and J. Bryan, *readxl: Read Excel Files*, 2022, https://readxl.tidyverse.org, https://github.com/tidyverse/readxl. [Cited on page 33.]

[130] B. Atkinson, *rpart*, 2021, https://github.com/bethatkinson/rpart. [Cited on page 33.]

[131] S. Milborrow, *rpart.plot*, 2022, http://www.milbo.org/rpart-plot/index.html. [Cited on page 33.]

[132] A. Kassambara, *rstatix: Pipe-Friendly Framework for Basic Statistical Tests*, 2021, r package version 0.7.0. [Online]. Available: https://cloud.r-project.org/web/packages/rstatix/rstatix.pdf [Cited on page 33.]

[133] H. Wickham and M. Girlich, *tidyr: Tidy Messy Data*, 2022, https://tidyr.tidyverse.org, https://github.com/tidyverse/tidyr. [Cited on page 33.]

[134] H. Wickham, M. Averick, J. Bryan, W. Chang, L. D. McGowan, R. François, G. Grolemund, A. Hayes, L. Henry, J. Hester, M. Kuhn, T. L. Pedersen, E. Miller, S. M. Bache, K. Müller, J. Ooms, D. Robinson, D. P. Seidel, V. Spinu, K. Takahashi, D. Vaughan, C. Wilke, K. Woo, and H. Yutani, "Welcome to the tidyverse," *Journal of Open Source Software*, vol. 4, no. 43, p. 1686, 2019. [Cited on page 33.]

[135] MONAI Consortium, "MONAI: Medical open network for ai," 2022. [Online]. Available: https://zenodo.org/record/4323058 [Cited on pages 33 and 49.]

[136] G. Van Rossum and F. L. Drake Jr, *Python reference manual*. Centrum voor Wiskunde en Informatica Amsterdam, 1995. [Cited on page 33.]

[137] T. Kluyver, B. Ragan-Kelley, F. Pérez, B. Granger, M. Bussonnier, J. Frederic, K. Kelley, J. Hamrick, J. Grout, S. Corlay, P. Ivanov, D. Avila, S. Abdalla, and C. Willing, “Jupyter notebooks – a publishing format for reproducible computational workflows,” in *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, F. Loizides and B. Schmidt, Eds. IOS Press, 2016, pp. 87 – 90. [Cited on page 33.]

[138] “Anaconda software distribution,” 2020. [Online]. Available: https://docs.anaconda.com/ [Cited on page 33.]

[139] S. Herbold, “Autorank: A python package for automated ranking of classifiers,” *Journal of Open Source Software*, vol. 5, no. 48, p. 2173, 2020. [Online]. Available: https://doi.org/10.21105/joss.02173 [Cited on page 34.]

[140] NVIDIA, P. Vingelmann, and F. H. Fitzek, “Cuda, release: 10.2.89,” 2020. [Online]. Available: https://developer.nvidia.com/cuda-toolkit [Cited on page 34.]

[141] G. Lemaître, F. Nogueira, and C. K. Aridas, “Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning,” *Journal of Machine Learning Research*, vol. 18, no. 17, pp. 1–5, 2017. [Online]. Available: http://jmlr.org/papers/v18/16-365.html [Cited on pages 34 and 43.]

[142] Joblib Development Team, “Joblib: running python functions as pipeline jobs,” 2020. [Online]. Available: https://joblib.readthedocs.io/ [Cited on page 34.]

[143] T. O’Malley, E. Bursztein, J. Long, F. Chollet, H. Jin, L. Invernizzi *et al.*, “Kerastuner,” https://github.com/keras-team/keras-tuner, 2019. [Cited on page 34.]

[144] J. D. Hunter, “Matplotlib: A 2d graphics environment,” *Computing in science & engineering*, vol. 9, no. 3, pp. 90–95, 2007. [Cited on page 34.]

[145] M. Brett, C. J. Markiewicz, M. Hanke, M.-A. Côté, B. Cipollini, P. McCarthy, D. Jarecka, C. P. Cheng, Y. O. Halchenko, M. Cottaar, E. Larson, S. Ghosh, D. Wassermann, S. Gerhard, G. R. Lee, H.-T. Wang, E. Kastman, J. Kaczmarzyk, R. Guidotti, J. Daniel, O. Duek, A. Rokem, C. Madison, B. Moloney, F. C. Morency, M. Goncalves, R. Markello, C. Riddell, A. Sólon, C. Burns, J. Millman, A. Gramfort, J. Leppäkangas, J. J. van den Bosch, R. D. Vincent, H. Braun,

K. Subramaniam, D. Papadopoulos Orfanos, A. Van, K. J. Gorgolewski, P. R. Raamana, J. Klug, B. N. Nichols, E. M. Baker, S. Hayashi, B. Pinsard, C. Haselgrove, M. Hymers, O. Esteban, S. Koudoro, F. Pérez-García, J. Dockès, N. N. Oosterhof, B. Amirbekian, I. Nimmo-Smith, L. Nguyen, S. Reddigari, S. St-Jean, E. Panfilov, E. Garyfallidis, G. Varoquaux, J. H. Legarreta, K. S. Hahn, L. Waller, O. P. Hinds, B. Fauber, J. Roberts, J.-B. Poline, J. Stutters, K. Jordan, M. Cieslak, M. E. Moreno, T. Hrnčiar, V. Haenel, Y. Schwartz, Z. Baratz, B. C. Darwin, B. Thirion, C. Gauthier, I. Solovey, I. Gonzalez, J. Palasubramaniam, J. Lecher, K. Leinweber, K. Raktivan, M. Calábková, P. Fischer, P. Gervais, S. Gadde, T. Ballinger, T. Roos, V. R. Reddam, and Freec84, "nipy/nibabel:," 2022. [Online]. Available: https://zenodo.org/record/6658382 [Cited on page 34.]

[146] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. Fernández del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, "Array programming with NumPy," *Nature*, vol. 585, p. 357–362, 2020. [Cited on page 34.]

[147] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000. [Cited on page 34.]

[148] W. McKinney *et al.*, "Data structures for statistical computing in python," in *Proceedings of the 9th Python in Science Conference*, vol. 445. Austin, TX, 2010, pp. 51–56. [Cited on page 34.]

[149] R. D. Team, *RAPIDS: Collection of Libraries for End to End GPU Data Science*, 2018. [Online]. Available: https://rapids.ai [Cited on page 34.]

[150] M. Terpilowski, "scikit-posthocs: Pairwise multiple comparison tests in python," *The Journal of Open Source Software*, vol. 4, no. 36, p. 1169, 2019. [Cited on page 34.]

[151] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro,

F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020. [Cited on page 34.]

[152] M. Waskom, O. Botvinnik, D. O'Kane, P. Hobson, S. Lukauskas, D. C. Gemperline, T. Augspurger, Y. Halchenko, J. B. Cole, J. Warmenhoven, J. de Ruiter, C. Pye, S. Hoyer, J. Vanderplas, S. Villalba, G. Kunter, E. Quintero, P. Bachant, M. Martin, K. Meyer, A. Miles, Y. Ram, T. Yarkoni, M. L. Williams, C. Evans, C. Fitzgerald, Brian, C. Fonnesbeck, A. Lee, and A. Qalieh, "mwaskom/seaborn: v0.8.1 (september 2017)," Sep. 2017. [Online]. Available: https://doi.org/10.5281/zenodo.883859 [Cited on page 34.]

[153] Autonomio, "Autonomio/talos: Hyperparameter optimization for tensorflow, keras and pytorch." [Online]. Available: https://github.com/autonomio/talos [Cited on page 34.]

[154] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: A system for large-scale machine learning," in *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, 2016, pp. 265–283. [Cited on page 34.]

[155] R. Solovyev, A. A. Kalinin, and T. Gabruseva, "3d convolutional neural networks for stalled brain capillary detection," *Computers in Biology and Medicine*, vol. 141, p. 105089, 2022. [Cited on pages 34 and 49.]

[156] "Fmrib software library v6.0." [Online]. Available: https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/ [Cited on pages 34 and 38.]

[157] "Misc functions of the department of statistics, probability theory group (formerly: E1071), tu wien [r package e1071 version 1.7-11]," Jun 2022. [Online]. Available: https://cran.r-project.org/web/packages/e1071/index.html [Cited on page 37.]

[158] "A collection of methods for left-censored missing data imputation [r package imputelcmd version 2.1]," Jun 2022. [Online]. Available: https://cran.r-project.org/package=imputeLCMD [Cited on page 37.]

[159] O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, and R. B. Altman, "Missing value estimation methods for DNA

microarrays," *Bioinformatics*, vol. 17, no. 6, pp. 520–525, jun 2001. [Online]. Available: https://doi.org/10.1093/bioinformatics/17.6.520 [Cited on page 37.]

[160] S. Oba, M. a. Sato, I. Takemasa, M. Monden, K. i. Matsubara, and S. Ishii, "A bayesian missing value estimation method for gene expression profile data," *Bioinformatics*, vol. 19, no. 16, pp. 2088–2096, oct 2003. [Online]. Available: https://doi.org/10.1093/bioinformatics/btg287 [Cited on page 37.]

[161] A. Kowarik and M. Templ, "Imputation with the R Package VIM," *Journal of Statistical Software*, vol. 74, no. 7, 2016. [Online]. Available: https://doi.org/10.18637/jss.v074.i07 [Cited on page 37.]

[162] D. J. Stekhoven and P. Bühlmann, "Missforest–non-parametric missing value imputation for mixed-type data." *Bioinformatics (Oxford, England)*, vol. 28, no. 1, pp. 112–8, Jan 2012. [Cited on page 38.]

[163] Ddsjoberg, "Gtsummary/add-p.r at main - ddsjoberg/gtsummary." [Online]. Available: https://github.com/ddsjoberg/gtsummary/blob/main/R/add_p.R [Cited on page 39.]

[164] A. Ross and V. L. Willson, "Basic and advanced statistical tests," 2017. [Cited on page 39.]

[165] S. Holm, "A simple sequentially rejective multiple test procedure. scandinavian journal of statistics," vol. 6, no. 2, 1979. [Online]. Available: http://www.jstor.org/stable/4615733 [Cited on page 39.]

[166] B. Ripley, "Pattern recognition and neural networks," *Cambridge University Press*, p. 394, 1996. [Cited on page 40.]

[167] "Nested versus non-nested cross-validation." [Online]. Available: https://scikit-learn.org/stable/auto_examples/model_selection/plot_nested_cross_validation_iris.html [Cited on page 41.]

[168] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research*, vol. 7, no. 1, pp. 1–30, 2006. [Online]. Available: http://jmlr.org/papers/v7/demsar06a.html [Cited on page 42.]

[169] C. Sima and E. R. Dougherty, “The peaking phenomenon in the presence of feature-selection,” *Pattern Recognition Letters*, vol. 29, no. 11, p. 1667–1674, 2008. [Cited on page 42.]

[170] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “Smote: Synthetic minority over-sampling technique,” *Journal of Artificial Intelligence Research*, vol. 16, p. 321–357, 2002. [Cited on page 43.]

[171] G. Kovács, “An empirical comparison and evaluation of minority oversampling techniques on a large number of imbalanced datasets,” *Applied Soft Computing*, vol. 83, p. 105662, 2019, (IF-2019=4.873). [Cited on pages xxi and 43.]

[172] ——, “smote-variants: a python implementation of 85 minority oversampling techniques,” *Neurocomputing*, vol. 366, pp. 352–354, 2019, (IF-2019=4.07). [Cited on page 43.]

[173] A. Saad Hussein, T. Li, C. W. Yohannese, and K. Bashir, “A-smote: A new preprocessing approach for highly imbalanced datasets by improving smote,” *International Journal of Computational Intelligence Systems*, vol. 12, no. 2, p. 1412, 2019. [Cited on page 43.]

[174] S. Gazzah and N. E. B. Amara, “New oversampling approaches based on polynomial fitting for imbalanced data sets,” in *2008 The Eighth IAPR International Workshop on Document Analysis Systems*, Sept 2008, pp. 677–684. [Cited on page 43.]

[175] C. Shorten and T. M. Khoshgoftaar, “A survey on image data augmentation for deep learning,” *Journal of Big Data*, vol. 6, no. 1, 2019. [Cited on page 49.]

[176] “Volumentations-3d.” [Online]. Available: https://pypi.org/project/volumentations-3D/ [Cited on page 49.]

[177] “Rsna-miccai brain tumor radiogenomic classification.” [Online]. Available: https://www.kaggle.com/competitions/rsna-miccai-brain-tumor-radiogenomic-classification [Cited on page 49.]

[178] L. Jin, Y. Bi, C. Hu, J. Qu, S. Shen, X. Wang, and Y. Tian, “A comparative study of evaluating missing value imputation methods in label-free proteomics,” *Scientific Reports*, vol. 11, no. 1, 2021. [Cited on page 61.]

[179] X. Wang, G. Zhang, X. Jiang, H. Zhu, Z. Lu, and L. Xu, "Neutrophil to lymphocyte ratio in relation to risk of all-cause mortality and cardiovascular events among patients undergoing angiography or cardiac revascularization: A meta-analysis of observational studies," *Atherosclerosis*, vol. 234, no. 1, p. 206–213, 2014. [Cited on page 63.]

[180] J. Wang, L. Ma, T. Lin, S.-J. Li, L.-L. Chen, and D.-Z. Wang, "The significance of eosinophils in predicting the severity of acute ischemic stroke," *Oncotarget*, vol. 8, no. 61, p. 104238–104246, 2017. [Cited on page 63.]

[181] H. Pagram, A. Bivard, L. F. Lincz, and C. Levi, "Peripheral immune cell counts and advanced imaging as biomarkers of stroke outcome," *Cerebrovascular Diseases Extra*, vol. 6, no. 3, p. 120–128, 2016. [Cited on page 63.]

[182] M. Correia, I. Silva, D. Gabriel, J. Simrén, A. Carneiro, S. Ribeiro, H. M. Dória, R. Varela, A. Aires, K. Minta, and et al., "Early plasma biomarker dynamic profiles are associated with acute ischemic stroke outcomes," *European Journal of Neurology*, vol. 29, no. 6, p. 1630–1642, 2022. [Cited on page 83.]

[183] C. Dargazanli, A. Consoli, M. Barral, J. Labreuche, H. Redjem, G. Ciccio, S. Smajda, J. Desilles, G. Taylor, C. Preda, and et al., "Impact of modified tici 3 versus modified tici 2b reperfusion score to predict good outcome following endovascular therapy," *American Journal of Neuroradiology*, vol. 38, no. 1, p. 90–96, 2016. [Cited on page 83.]

[184] [Online]. Available: https://www.medscape.com/ [Cited on pages 83 and 84.]

[185] "gpnotebook." [Online]. Available: https://gpnotebook.com/ [Cited on page 84.]

[186] J. P. ARCHIE, "Mathematic coupling of data," *Annals of Surgery*, vol. 193, no. 3, p. 296–303, 1981. [Cited on page 84.]

[187] M. Ernst, A. Boers, N. Forkert, O. Berkhemer, Y. Roos, D. Dippel, A. van der Lugt, R. van Oostenbrugge, W. van Zwam, E. Vettorazzi, and et al., "Impact of ischemic lesion location on the mrs score in patients with ischemic stroke: A voxel-based approach," *American Journal of Neuroradiology*, vol. 39, no. 11, p. 1989–1994, 2018. [Cited on page 85.]

[188] B. A. Drozdowska, S. Singh, and T. J. Quinn, "Thinking about the future: A review of prognostic scales used in acute stroke," *Frontiers in Neurology*, vol. 10, 2019. [Cited on page 87.]

[189] C. Cortes and M. Mohri, "Confidence intervals for the area under the ROC curve," in *Advances in Neural Information Processing Systems*, L. Saul, Y. Weiss, and L. Bottou, Eds., vol. 17. MIT Press, 2004. [Online]. Available: https://proceedings.neurips.cc/paper/2004/file/a7789ef88d599b8df86bbee632b2994d-Paper.pdf [Cited on page 87.]

[190] D. W. Hosmer, S. Lemeshow, and R. X. Sturdivant, *Applied Logistic Regression*. John Wiley and Sons, 2013. [Cited on page 89.]

[191] Zach, "What is considered a good AUC score?" Sep 2021. [Online]. Available: https://www.statology.org/what-is-a-good-auc-score/ [Cited on page 89.]

[192] N. Hema Rajini and R. Bhavani, "Computer aided detection of ischemic stroke using segmentation and texture features," *Measurement*, vol. 46, no. 6, p. 1865–1874, 2013. [Cited on pages 90 and 92.]

[193] C. Zhang, W. Zhang, Y. Huang, J. Qiu, and Z.-X. Huang, "A dynamic nomogram to predict the 3-month unfavorable outcome of patients with acute ischemic stroke," *Risk Management and Healthcare Policy*, vol. Volume 15, p. 923–934, 2022. [Cited on page 90.]

[194] Z. Cheng, X. Geng, G. B. Rajah, J. Gao, L. Ma, F. Li, H. Du, and Y. Ding, "Nihss consciousness score combined with aspects is a favorable predictor of functional outcome post endovascular recanalization in stroke patients," *Aging and disease*, vol. 12, no. 2, p. 415, 2021. [Cited on page 91.]

[195] W. S. Smith, G. Sung, J. Saver, R. Budzik, G. Duckwiler, D. S. Liebeskind, H. L. Lutsep, M. M. Rymer, R. T. Higashida, S. Starkman, and et al., "Mechanical thrombectomy for acute ischemic stroke," *Stroke*, vol. 39, no. 4, p. 1205–1212, 2008. [Cited on page 92.]