DOBLE MÁSTER EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES E INDUSTRIA CONECTADA

TRABAJO FIN DE MÁSTER

COMPARATIVA DE MÉTODOS DE INTERPRETABILIDAD EN MODELOS DE MACHINE LEARNING Y DEEP LEARNING

Autor: Juan Raposo Picos

Director: Sergio Altares López

Co-Director: José María Bengochea Guevara

Madrid

Agosto 2025

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título

Comparativa de métodos de interpretabilidad en modelos de Machine Learning y Deep Learning

en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el curso académico 2024/25 es de mi autoría, original e inédito y no ha sido presentado con anterioridad a otros efectos.

El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido tomada de otros documentos está debidamente referenciada.

Fdo.: Juan Raposo Picos Fecha: 20/ 08/ 2025

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO

Fdo.:  Fecha: 26/ 08/2025

DOBLE MÁSTER EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES E INDUSTRIA CONECTADA

TRABAJO FIN DE MÁSTER

COMPARATIVA DE MÉTODOS DE INTERPRETABILIDAD EN MODELOS DE MACHINE LEARNING Y DEEP LEARNING

Autor: Juan Raposo Picos

Director: Sergio Altares López

Co-Director: José María Bengochea Guevara

Madrid

Agosto 2025

COMPARATIVA DE MÉTDOS DE INTERPRETABILIDAD EN MODELOS DE MACHINE LEARNING Y DEEP LEARNING

Autor: Raposo Picos, Juan**.**
Director: Sergio Altares López.
Co-Director: José María Bengochea Guevara
Entidad Colaboradora: ICAI – Universidad Pontificia Comillas.

RESUMEN DEL PROYECTO

Palabras clave: Inteligencia artificial (IA), interpretabilidad, IA explicativa (XAI), SHAP, LIME, Grad-CAM++, métrica unificada, IA ética, diagnóstico del Alzheimer, transparencia, fiabilidad.

1. Introducción

El proyecto aborda el creciente desafío de la **interpretabilidad en los modelos de aprendizaje** automático, especialmente en sectores críticos de alto riesgo como el sanitario. Si bien los sistemas modernos de IA pueden alcanzar una precisión predictiva notable, su **naturaleza de *black box*** a menudo impide a las partes interesadas comprender cómo el modelo toma las decisiones, cuáles son las variables que más influencian al resultado, lo que da lugar a preocupaciones ético-normativas y relacionadas con la confianza del modelo. La aparición de la **IA explicativa** (XAI) tiene como objetivo abordar este desafío, con métodos matemáticos como SHAP y LIME que proporcionan explicaciones independientes del modelo. Sin embargo, sigue habiendo una falta de métricas estandarizadas para comparar y evaluar estas herramientas de forma objetiva.

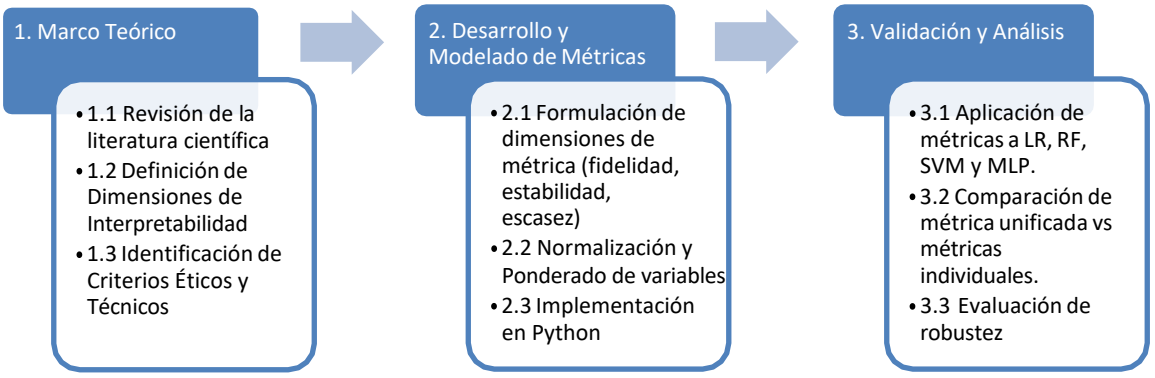
2. Definición del proyecto

El proyecto consiste en desarrollar una **métrica de interpretabilidad unificada** capaz de evaluar y comparar cuantitativamente la calidad de las explicaciones entre diferentes modelos y conjuntos de datos. La métrica **integra tres dimensiones clave**: fidelidad, estabilidad y dispersión en una única puntuación. El enfoque combina **técnicas XAI de última generación** (SHAP, LIME, Grad-CAM++) con pruebas empíricas en conjuntos de datos heterogéneos, incluidos datos tabulares de **biomarcadores del Alzheimer** y **datos de imágenes médicas** (MRI).

3. Descripción del modelo/sistema/herramienta

Aplicación de múltiples modelos de aprendizaje automático (Regresión Logística, Random Forest, SVM y MLP) y conjuntos de datos. Comparación entre puntuaciones de métrica unificadas y métricas de interpretabilidad individuales.

En general, este enfoque se desarrolla en tres fases, integrando un **análisis teórico**, el **diseño de métricas** y la **validación empírica** para proporcionar una herramienta versátil para la evaluación coherente de la interpretabilidad.



Este proyecto se ajusta a los siguientes ODS:

1. **ODS 3: Salud y Bienestar:** Se contribuye a la elaboración de modelos de ML más precisos e interpretables en la atención sanitaria, en particular, mediante el diagnóstico del Alzheimer.
2. **ODS 9: Industria, Innovación e Infraestructura:** Mediante el desarrollo de marcos éticos de IA que promuevan la transparencia y la confianza en los sistemas tecnológicos.
3. **ODS 16: Paz, Justicia e Instituciones Sólidas:** Mediante el fomento de la responsabilidad y las normas éticas en el despliegue de IA.

4. Conclusiones

En este Proyecto se han alcanzado varios hitos: la aplicación de SHAP, LIME y Grad-CAM++ a datos en formato tabular e imágenes, el desarrollo de una métrica de interpretabilidad unificada y la validación de su uso en el diagnóstico del Alzheimer con resultados generalizables a más de 2000 pacientes.

Las limitaciones incluyen las pruebas en un conjunto reducido de conjuntos de datos debido a la limitación para obtener bases de datos de acceso público. Las investigaciones futuras deberían ampliarse a otros ámbitos (financiero, legal, etc.), integrar el razonamiento causal e incluir la validación por parte de expertos en el área de estudio.

Se han logrado varios hitos importantes en el transcurso del proyecto: Se está aplicando SHAP, LIME y Grad-CAM++ a tablas e imágenes, desarrollando una métrica de interpretabilidad unificada y validando su uso en el diagnóstico de Alzheimer con resultados generalizables a más de 2000 pacientes.

En resumen, este proyecto ofrece un marco unificado que promueve una IA fiable y transparente en el ámbito del diagnóstico médico.

COMPARISON OF INTERPRETABILITY METHODS IN MACHINE LEARNING AND DEEP LEARNING MODELS

Author: Raposo Picos, Juan.
Supervisor: Sergio Altares López
Co-Supervisor: José María Bengochea Guevara
Collaborating Entity: ICAI –
Universidad Pontificia Comillas.

ABSTRACT

Keywords: Artificial Intelligence, Interpretability, Explainable AI (XAI), SHAP, LIME, Grad-CAM++, unified metric, ethical AI, Alzheimer's diagnostics, transparency, trustworthiness.

1. Introduction

This project addresses the growing challenge of **interpretability in machine learning models**, especially in critical high-stakes sectors such as healthcare. While modern AI systems can achieve remarkable predictive accuracy, their **black-box nature** often prevents stakeholders from understanding how decisions are made, leading to ethical regulatory, and trust-related concerns. The emergence of **Explainable AI** (XAI) aims to address this challenge, with methods like SHAP and LIME providing model-agnostic explanations. However, there remains a lack of standardized metrics to compare and evaluate these tools objectively.

2. Project definition

The project consists of developing a **unified interpretability metric** capable of quantitively assessing and comparing explanation quality across different models and datasets. The metric integrates **three key dimensions:** fidelity, stability and sparsity into a single score. The approach combines **state-of-the-art XAI techniques** (SHAP, LIME, Grad-CAM++) with empirical testing on **heterogeneous datasets**, including tabular Alzheimer's biomarker data and medical imaging data.

3. Description of the model/system/tool

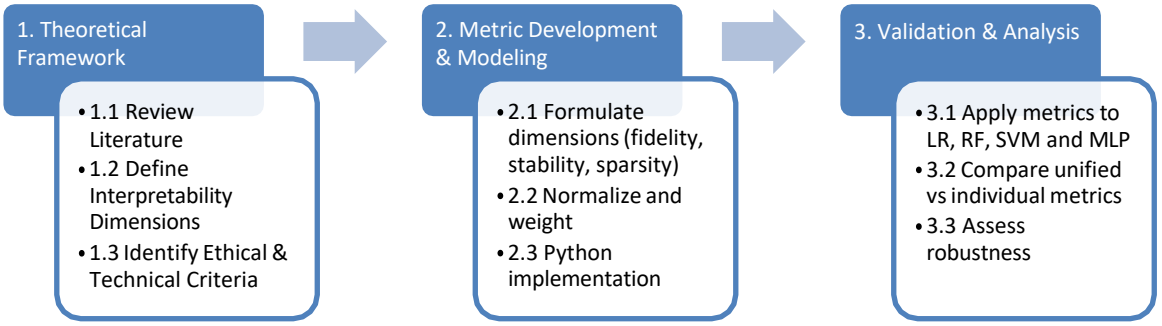
The development of the metric follows three main phases:

1. **Theoretical Framework:** Review of AI interpretability literature, ethical AI principles, and limitations of existing tools. Identification of relevant interpretability dimensions from both technical and usability perspectives.

2. **Metric Development & Modeling:**
    a. Definition of mathematical formulations for each interpretability dimension.
    b. Normalization and weighting to ensure comparability.
    c. Implementation in Python, leveraging scikit-learn, SHAP, LIME, and PyTorch-based Grad-CAM++.

3. **Validation & Analysis:** Application to multiple ML models (Logistics Regression, Random Forest, SVM) and datasets. Comparative between unified metric scores and individual interpretability metrics.

Overall, this three-phase approach integrates **theoretical analysis, empirical validation, and metric design** to provide a versatile tool for consistent interpretability evaluation.



This project aligns with the following SDGs:

1. **SDG 3: Good Health and Well-being**: Contributing to the development of more accurate and interpretable ML models in healthcare, particularly for Alzheimer's diagnosis.

2. **SDG 9: Industry, Innovation and Infrastructure**: Through the development of ethical AI frameworks that promote transparency and trust in technological systems.

3. **SDG 16: Peace, Justice and Strong Institutions**: By fostering responsibility and ethical standards in the deployment of AI.

Conclusions

Several milestones were achieved: applying SHAP, LIME, and Grad-CAM++ to tabular and image, developing a unified interpretability metric, and validating its use in Alzheimer's diagnostics with results generalizable to over 2,000 patients.

Limitations include testing on a narrow set of models and datasets. Future research should expand to other domains, integrate causal reasoning, and involve expert validation.

In summary, this project delivers a unified framework that advances trustworthy and transparent AI in healthcare.

Table of Contents

Figure Index

Table Index

Chapter 1. Introduction

Artificial Intelligence (AI) stands as one of the most transformative forces of the 21st century, driving advancements across virtually every domain—from personalized recommendation engines and autonomous vehicles to diagnostic imaging in medicine and algorithmic trading in finance. However, despite its widespread adoption, AI remains deeply misunderstood and often mistrusted by the public (Amodei et al., 2016; Mitchell et al., 2021). This paradox—of immense utility shadowed by societal apprehension—stems largely from the opaque nature of modern machine learning (ML) systems, whose decisions are often inscrutable even to their developers.

The tension between technological progress and human control is not a novel concern. During the Industrial Revolution, the Luddite movement famously resisted mechanization that threatened artisanal labor (Mokyr, 1999). Today, similar anxieties resurface in the face of automation powered by AI, with fears of job displacement, algorithmic bias, and loss of autonomy dominating public discourse (Acemoglu & Restrepo, 2020).

As deep learning architectures have achieved superhuman performance in tasks like image classification (He et al., 2016), natural language processing (Brown et al., 2020), and strategic gameplay (Silver et al., 2017), their inner workings have simultaneously grown opaquer. This black-box nature presents critical risks: biased facial recognition systems with disparate error rates across demographic groups (Raji et al., 2020); recommendation algorithms that amplify misinformation (Hosanagar & Vakili, 2021); and opaque clinical models that misguide diagnoses and treatments (Kelly et al., 2019).

Given these challenges, the need for explainable and trustworthy AI has never been more pressing—particularly in domains where decisions have profound human consequences. While recent advances in explainability techniques such as SHAP and LIME have helped open the "black box," their evaluation and comparison remain inconsistent and fragmented. Most interpretability tools focus on isolated aspects such as feature attribution or visual saliency, often overlooking key dimensions like robustness, stability, or computational cost.

This gap motivates the present thesis, which seeks to bridge technical performance and interpretability by proposing a unified metric that quantifies and compares the explanatory power

of different tools. The following section delves into the broader context in which this work is situated and the specific motivations driving its development.

1.1 CONTEXT AND MOTIVATION

The journey toward explainable AI is deeply rooted in the broader evolution of artificial intelligence itself. From Alan Turing's foundational question—"Can machines think?"—to the rise of deep neural networks, the development of AI has been consistently shaped by both technical innovation and public perception.

Over the past seventy years, Artificial Intelligence (AI) has evolved from a speculative concept into a transformative force that permeates nearly every aspect of modern life. This progression has not been linear; rather, it has unfolded through alternating waves of optimism, setbacks, and resurgence. From Turing's early thought experiments in the 1950s to the birth of AI as a formal discipline at the Dartmouth Conference, through the cycles of "AI winters" and breakthroughs, up to today's era of deep learning, the field has continually adapted to new challenges and opportunities. Increasingly, this includes a focus on ethical concerns and the demand for explainable AI, marking a pivotal shift in both how AI is developed and how it is understood by society.

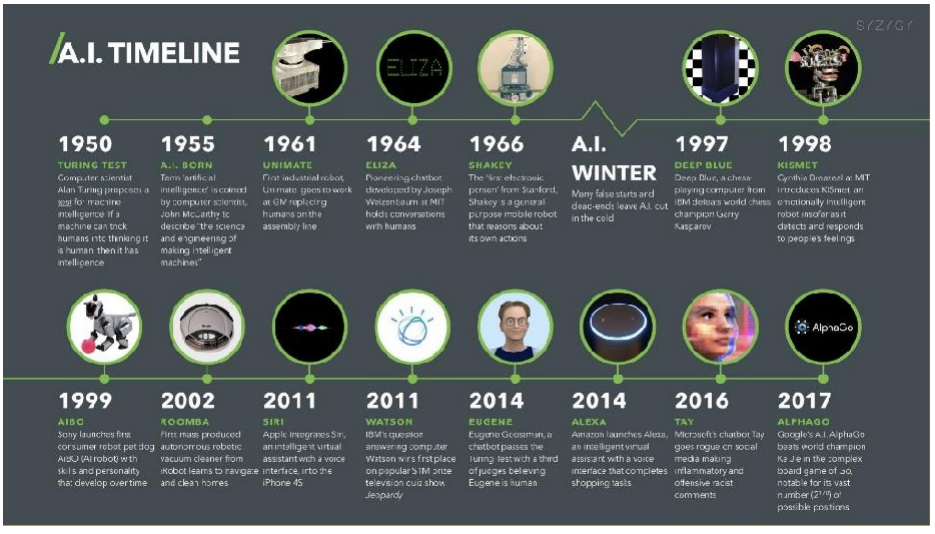


Figure 1: A visual timeline of major AI milestones from the 1950s (Turing Test, Dartmouth) to the 2020s (AlphaGo, OpenAI, XAI)

Early Visions and the Birth of AI (1940s–1950s)

In the mid-20th century, British mathematician Alan Turing laid the theoretical groundwork for AI. Turing famously posed the question "Can machines think?" in his 1950 paper *Computing Machinery and Intelligence* (Turing, 1950). He described an experimental setup known as the Imitation Game—now famous as the Turing Test—where a machine's ability to exhibit human-like conversation is evaluated. Turing's vision suggested that a sufficiently advanced computer could fool humans into believing it was sentient, planting the seed for the modern concept of artificial intelligence.

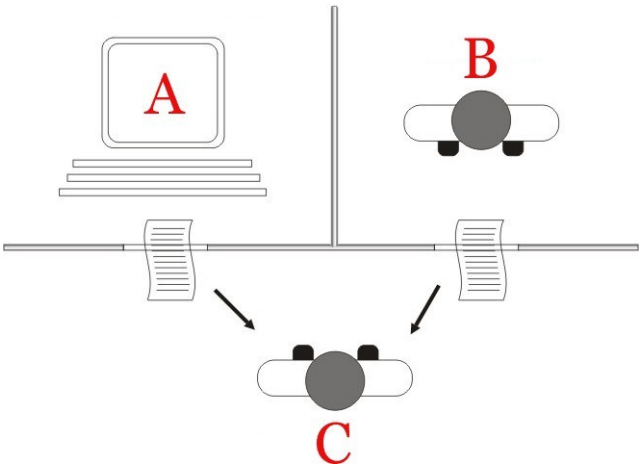


Figure 2: The Turing Test, proposed by Alan Turing, evaluates a machine's ability to exhibit intelligent behavior indistinguishable from a human. If a human evaluator cannot reliably tell the machine from a human based on conversation alone, the machine is said

Just a few years later, in 1956, the field of AI was officially born at an academic workshop. In the summer of 1956, computer scientist John McCarthy convened the Dartmouth Summer Research Project on Artificial Intelligence, coining the term "artificial intelligence" in the process (McCarthy et al., 1956). This gathering of pioneers (including McCarthy, Marvin Minsky, Claude Shannon, and others) is often cited as the moment AI became an organized discipline. The Dartmouth conference launched ambitious efforts to create machines that could perform tasks such as reasoning and problem-solving, heralding a new era of optimism about intelligent machines.

Early Programs and Growing Optimism (1950s–1960s)

In the wake of Dartmouth, researchers achieved several early AI milestones that made headlines.

Logic-oriented programs could prove mathematical theorems and solve puzzles—for example, Allen Newell and Herbert Simon's **"General Problem Solver"** in the late 1950s attempted to apply human-like reasoning to a broad range of problems. In 1966, Joseph Weizenbaum created **"ELIZA"**, a conversational program that simulated a therapist by rephrasing users' inputs. ELIZA's dialogues were simplistic, but the program amazed observers by occasionally fooling people into thinking a real human was responding. Such examples showed that computers could at least mimic aspects of reasoning and language.

A different approach to AI also emerged with the development of machines that learn from experience. In 1958, Frank Rosenblatt introduced the **perceptron**, an early artificial neural network that learned to classify patterns through trial and error (Rosenblatt, 1958). Inspired by neurons in the brain, the perceptron adjusted its internal weights based on feedback, allowing it to recognize simple shapes and characters. This sparked enthusiasm that machines might soon learn and adapt as humans do. By the late 1960s, the prevailing mood in the field was optimistic—some researchers even predicted that fully intelligent machines would be achieved within a generation.

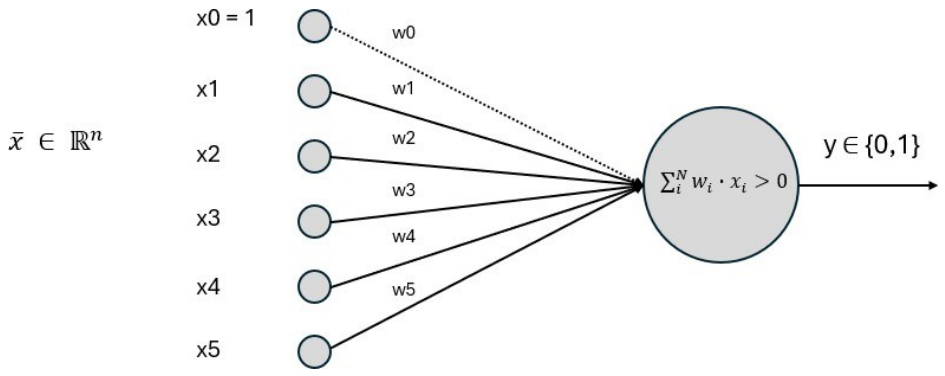


Figure 3: Diagram of Rosenblatt's single-layer perceptron model with input nodes, adjustable weights, and an output node illustrating how a neural network learns from data.

Challenges and the First AI Winter (1970s)

By the early 1970s, the initial optimism about AI began to fade in the face of unmet expectations.

Despite the early demos, AI programs struggled with complexity outside of narrow toy examples. In 1969, Marvin Minsky and Seymour Papert published a book highlighting fundamental limitations of the perceptron, showing it could not solve certain simple classes of problems. Around the same time, grand promises—such as fully automated language translation—fell short of their goals. In 1973, the British government's **Lighthill Report** delivered a harsh critique of AI progress, leading to serious funding cuts. Similarly, U.S. agencies like DARPA grew sceptical as projects failed to achieve their lofty aims. This period of disappointment and reduced support became known as the first **"AI winter,"** a metaphorical freeze in AI research during the mid-1970s.

Expert Systems and a New Spring (1980s)

Around the 1980s, AI experienced a resurgence driven by a new strategy: **expert systems**. These were programs designed to capture human expertise in explicit *if-then* rules. For example, "MYCIN" was an early expert system that could diagnose blood infections, and another called "XCON" helped configure computer hardware orders. Companies saw practical value in these knowledge-based systems, deploying them for tasks in medicine, finance, and engineering. Governments also renewed investments in AI—most famously, Japan launched a massive Fifth Generation Computer project in 1982 to advance AI and computing. This era (often dubbed the "AI boom" of the 1980s) saw optimism return as AI moved from the lab into industry.

Yet by the late 1980s, the limitations of expert systems became apparent. Building and maintaining these rule-based programs was extremely labour-intensive, and they tended to be brittle—small changes in requirements could break their logic, and they had no capacity to learn new facts on their own. As inflated expectations met reality, enthusiasm cooled again. Specialized AI hardware companies failed (for instance, the market for Lisp machines—computers optimized for AI—collapsed by the decade's end). The field entered another downturn, sometimes called the second AI winter, in the late 1980s and early 1990s as funding and hype receded once more.

Notably, amid the 1980s boom, important algorithmic advances were quietly taking place in academia. In 1986, Geoffrey Hinton and colleagues reintroduced the **backpropagation** learning algorithm for training multi-layer neural networks, overcoming a key obstacle that had stymied

neural nets for years (Rumelhart et al., 1986). This work—building on foundational efforts by **Yann LeCun**, who in the late 1980s developed **convolutional neural networks (CNNs)** for tasks like handwritten digit recognition—laid the groundwork for the major breakthroughs in machine learning that would unfold later.

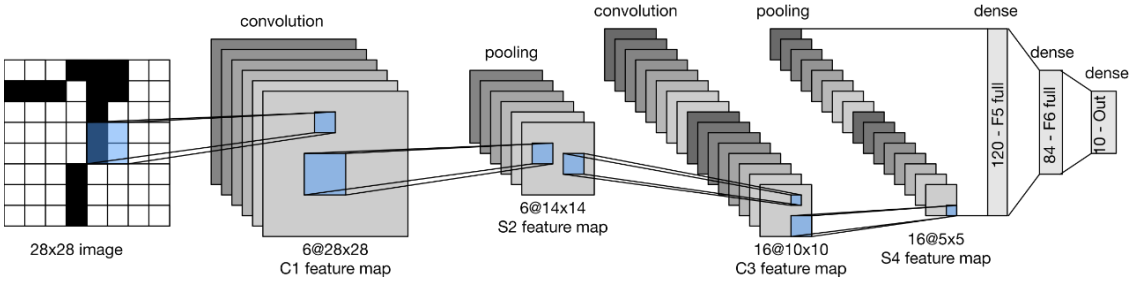


Figure 4:LeNet is a series of convolutional neural network architectures created by a research group in AT&T Bell Laboratories during the 1988 to 1998 period, centered around Yann LeCun.

Machine Learning and Game Milestones (1990s)

In the 1990s, as the field rebounded from the late-'80s downturn, AI research shifted toward more data-driven, statistical approaches, collectively known as *machine learning*. Instead of relying on hand-crafted rules, machine learning methods enabled computers to discern patterns and improve through experience. This shift yielded steady progress in pattern-recognition tasks: for example, systems for speech recognition and handwritten character reading became increasingly accurate during the 1990s.

The world took notice of AI's renewed momentum in 1997, when IBM's **Deep Blue** chess computer defeated world champion Garry Kasparov in a six-game match. Deep Blue's victory—achieved by combining brute-force search with expert-crafted heuristics—was a symbolic moment for the field. It showed that, at least in constrained domains like board games, machines could rival and even surpass human experts. Around the same time, AI began to be used in more everyday applications, from route-finding in mapping software to early web search engines and e-commerce recommendation systems, albeit often behind the scenes.

AI in Everyday Life: Big Data and Virtual Assistants (2000s–2010s)

As the 21st century began, AI quietly spread into everyday life on the back of big data and the internet. Ordinary people interacted with AI often without realizing it—email spam filters, product recommendations on shopping sites, and credit card fraud detection systems were all powered by machine learning behind the scenes. However, one of the first highly public AI triumphs of the new century came in 2011 when IBM's **Watson** system defeated champion human contestants on the quiz show *Jeopardy!* Watson's success in answering complex, tricky natural-language questions demonstrated how far AI's language processing and knowledge retrieval capabilities had advanced.

That same year, AI made its way into millions of pockets with the debut of Apple's **Siri** (2011), a voice-operated digital assistant on the iPhone. For the first time, non-experts could speak to a computer in normal language and receive useful answers or actions. In the following years, such virtual assistants proliferated—Amazon's **Alexa** (first released in 2014) and the Google **Assistant** (2016) became household names. These systems used improved speech recognition and natural language understanding (powered by modern AI algorithms) to carry out tasks or fetch information. By the mid-2010s, interacting with an AI-powered assistant or relying on AI-driven services had become commonplace, marking a shift where AI was no longer confined to labs or chess matches but had become a part of everyday consumer technology.

Figure 5: Amazon Alexa. How it works

Deep Learning Breakthroughs (2010s)

Around this same time, a dramatic **deep learning** revolution was transforming AI. Although the concept of multi-layer neural networks had existed for decades, around 2012 such networks (facilitated by big data and powerful graphics processors) began to outperform older AI techniques by stunning margins. For instance, a deep neural network approach won a major image-recognition competition in 2012, achieving an error rate far lower than any previous system. Similar deep learning techniques soon brought huge leaps in speech recognition and machine translation quality. In 2014, researchers at Facebook developed **DeepFace**, a deep learning system that could recognize human faces in photographs with near-human accuracy—an unprecedented feat at the time.

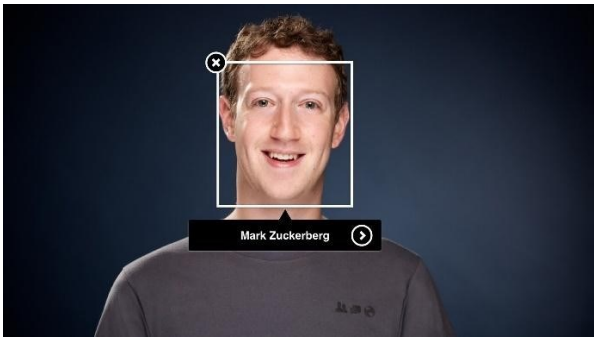


Figure 6: Face Recognition with Facebook DeepFace

Deep learning combined with other advances also enabled AI to triumph in domains once thought uniquely the province of humans. In 2016, Google DeepMind's **AlphaGo** program made history by defeating Go champion Lee Sedol—a milestone experts had believed was at least a decade away (Silver et al., 2016). Go, an ancient board game exponentially more complex than chess, had long been considered a last bastion of human dominance due to its vast search space and the intuitive skill it demands. AlphaGo's victory, powered by neural networks and advanced self-learning algorithms, was a watershed moment demonstrating the potential of AI. The following year, its successor **AlphaZero** displayed even greater generality by learning to master chess, Go, and shogi from scratch (without any human-provided strategies), achieving superhuman play in each. These successes underscored that AI had entered a new era of capability, often matching or exceeding human performance in specialized tasks.

OpenAI and the Era of Generative AI (late 2010s–2020s)

In recent years, the frontier of AI has been pushed even further by breakthroughs in generative models and by the efforts of new research organizations. **OpenAI**, a research lab founded in 2015, has been at the forefront of many of these developments. OpenAI's GPT series of models (short for *Generative Pre-trained Transformers*) showed that AI can generate human-like text after being trained on vast amounts of internet data. **GPT-3**, introduced in 2020, stunned observers with its ability to produce fluent paragraphs of text and answer questions with minimal prompts (Brown et al., 2020). This line of research culminated in widely used applications like **ChatGPT** (released in 2022), which brought conversational AI to millions of users and ignited popular interest in AI's capabilities. OpenAI and others have also demonstrated AI's creative potential in other domains—for example, generative models that create realistic images from text descriptions (as seen with OpenAI's **DALL-E** system) or even compose music. By the mid-2020s, AI systems were not only solving complex problems but also producing original content, signalling a new stage in how AI can augment human creativity and productivity.

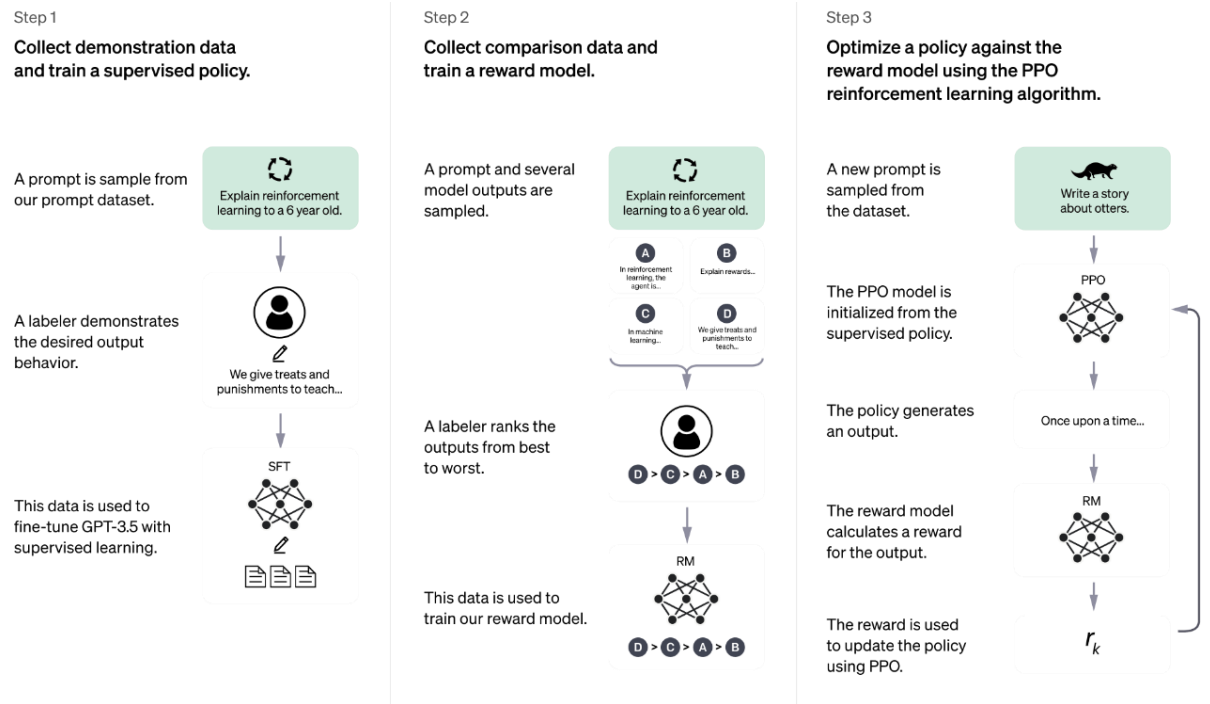


Figure 7: Introducing ChatGPT

Toward Ethical and Explainable AI

Alongside these technological advances, there has been a growing focus on the ethical and societal implications of AI. As AI systems began influencing high-stakes decisions—such as loan approvals, medical diagnoses, and criminal justice outcomes—concerns arose about bias, fairness, and transparency in how those decisions were made. Researchers and policymakers in the late 2010s started formulating principles and guidelines for **AI ethics** to ensure AI would be developed and deployed responsibly (Cath, 2018). Issues like data privacy, algorithmic bias against certain groups, and the need for human accountability in AI-driven decisions became central topics. In response, companies and governments introduced frameworks for "trustworthy AI," emphasizing values such as transparency, accountability, and human oversight.

A key part of making AI trustworthy is making its decisions understandable to humans. This need spurred the rise of **Explainable AI (XAI)**, which seeks to open the "black box" of complex models and reveal how they work. For instance, one popular XAI approach, called **SHAP** (short for *Shapley Additive Explanations*), assigns each input feature a score representing its influence on a particular prediction (Lundberg & Lee, 2017). By providing human-readable explanations—such as highlighting the specific factors that led an AI system to flag a financial transaction as fraudulent or to recommend a certain medical treatment—XAI helps users and experts understand and trust AI outputs. In sectors like healthcare and finance, such explanations are especially critical: doctors need to know *why* an AI suggested a diagnosis, and financial institutions must ensure automated decisions are fair and can be audited. The emergence of AI ethics and XAI represents the latest phase of AI's evolution, one focused on aligning advanced AI systems with societal values and ensuring they are used responsibly.

1.2 AI ETHICS AND THE IMPORTANCE OF EXPLAINABILITY

Opaque Models in High-Stakes Decisions: Ethical Risks

The use of opaque black-box AI models in high-stakes domains—such as healthcare, finance, and

criminal justice—raises serious ethical concerns. These complex models often achieve high accuracy, but their internal logic is not transparent to humans. Lack of transparency undermines trust and accountability: stakeholders cannot understand or challenge how decisions are made, which is problematic when those decisions carry significant consequences. As Rudin (2019) argues, black-box machine learning models have already caused real problems in domains like healthcare and criminal justice and relying on post-hoc explanations rather than using interpretable models from the start may "perpetuate bad practices" and even lead to "catastrophic harm to society". In other words, when an AI's reasoning remains opaque, errors or biases can go undetected until they result in negative outcomes. This has led scholars to warn that if an algorithm's decision process cannot be examined, ethical oversight is fundamentally compromised (Doshi-Velez & Kim 2017).

High-profile examples illustrate these risks. In healthcare, one hospital's deployment of an AI system to prioritize patients backfired when staff could not explain why certain high-risk patients were deprioritized; the resulting public backlash eroded patient trust and the provider's reputation. Such cases show how opacity in AI decisions can directly undermine confidence and safety in critical settings. More broadly, organizations using inscrutable AI face "reputational harm, regulatory penalties, and loss of stakeholder trust" if the technology's impacts are not transparent. Explainability thus emerges as a foundational requirement in high-stakes AI – without it, users and those affected may justifiably resist or question algorithmic decisions (Tolan et al. 2021). In summary, an AI system that cannot explain its reasoning poses ethical risks by limiting human ability to evaluate fairness, contest outcomes, or prevent harm. This provides the ethical impetus for prioritizing interpretability in any domain where automated decisions have profound human consequences.

Accountability, Fairness, and Bias in Automated Decision-Making

Opaque AI systems also threaten core principles of accountability and fairness, especially when their predictions directly affect people's lives. Accountability in AI refers to the ability to trace outcomes back to responsible parties and processes. However, black-box models can create what scholars term a "responsibility gap". If something goes wrong, an algorithm makes a harmful medical recommendation or a biased hiring decision—who is accountable? When neither the end-

user (e.g. a doctor or recruiter) nor the model developers can explain the rationale of an AI decision, it becomes difficult to assign responsibility for errors. Legal and ethical accountability evaporates if an automated decision cannot be understood or audited. This erosion of accountability is more than theoretical: it has practical implications for liability and governance. For instance, Afnan et al. (2021) note that if clinicians rely on an opaque AI tool and a patient is harmed, courts may struggle to evaluate the decision-making process and determine culpability. In high-stakes scenarios, society requires clear lines of responsibility for AI outcomes, which in turn demands that those outcomes be explainable and transparent.

Fairness and bias mitigation are equally critical. By default, black-box models learn from historical data and can inadvertently perpetuate or even amplify biases present in that data. Without explainability, these biases may remain hidden until they cause tangible discrimination. A well-documented example is Amazon's experimental hiring algorithm, which was trained on past resumes. The model learned to discriminate against women in technical job recommendations, reflecting the male dominance of the historical data. Even after Amazon engineers noticed the bias, they struggled to correct it—the AI kept finding proxy ways to favor male candidates. This case demonstrates how difficult it is to detect and correct unfair bias in a black-box system, even for a tech giant, and it underscores the need for transparency to identify such issues. In the criminal justice domain, Angwin et al. (2016) famously investigated the COMPAS recidivism prediction tool and found it was biased against black defendants, labeling them "high risk" at disproportionately higher rates. Notably, COMPAS was a proprietary algorithm; its inner workings were opaque, which meant that stakeholders and even defendants had little insight into why the tool was making biased predictions. These examples underscore why fairness is a pillar of AI ethics: automated decisions must not exacerbate inequality or injustice. Ensuring fairness requires that AI systems are transparent enough to be audited for bias and adjusted as needed. Explainability plays a key role here – if we can understand how an algorithm arrives at a decision, we can more readily spot biased logic or disproportionate impacts on certain groups and then take steps to mitigate those biases. In short, when AI predictions affect people's rights or opportunities, there is an ethical imperative to make the process accountable and fair. This means that AI designers must incorporate bias checks and explanations into their systems, allowing human oversight to verify that outcomes are equitable and justifiable.

Legal and Regulatory Requirements for Explainability and Transparency

Recognizing these ethical risks, regulators and policymakers have begun to mandate explainability and transparency in AI systems. In the European Union, the landmark General Data Protection Regulation (GDPR) directly addresses algorithmic decision-making. Article 22 of GDPR gives individuals the right not to be subject to decisions based solely on automated processing that have significant effects on them, unless certain safeguards are in place. One key safeguard is the provision of "meaningful information about the logic involved" in such automated decisions. In essence, if an organization deploys an AI to make important decisions about individuals (for example, in lending, hiring, or medical diagnosis), the individual has a right to an explanation of how the decision was made. Scholars initially interpreted this as a potential "right to an explanation" in GDPR, sparking debate about its scope. While the exact interpretation is debated (Selbst & Powles 2017; Wachter et al. 2017), the spirit of the law clearly pushes toward greater algorithmic transparency. At minimum, GDPR requires that automated decisions affecting individuals are not complete black boxes: data subjects should receive an explanation that they can understand, supporting the broader goals of accountability and fairness in data processing.

Building on this foundation, the EU is finalizing a comprehensive Artificial Intelligence Act (EU AI Act) that makes explainability a legal requirement for many systems. The AI Act takes a risk-based approach, imposing the strictest obligations on "high-risk AI systems" – a category that covers AI used in critical areas like healthcare, transportation, law enforcement, and essential services. A key principle in the AI Act is that high-risk AI must be transparent and explainable. Providers of such AI systems will be obliged to design and develop them in a way that ensures sufficient transparency so that deployers (and regulators) can reasonably understand the system's functioning and outputs. In practical terms, this means companies must be able to explain how their AI makes decisions, at least to the extent of clarifying the factors and logic that drive its predictions. The Act also requires detailed documentation and "instructions for use" to be delivered with high-risk AI products, including clear information on the system's characteristics, intended performance, and limitations. These transparency obligations aim to make AI systems traceable and auditable, aligning with the idea that trust in AI "requires opening the black box" and enabling oversight.

Notably, explainability under the EU AI Act is not viewed as a mere bureaucratic hurdle, but as a mechanism to ensure trust and accountability. As one commentary observes, explainability builds trust by demonstrating that AI behaves reliably and as intended, rather than erratically or inscrutably. It also promotes accountability: when we understand an algorithm's decision process, we can audit it for errors or biases and thereby protect individuals and groups from discrimination. These legal requirements echo the consensus in AI ethics guidelines worldwide. For instance, the European Commission's High-Level Expert Group on AI (2019) identified transparency (including the ability to explain AI decisions) and accountability as two of the seven key requirements for "Trustworthy AI". Similarly, the OECD AI Principles (adopted by over 40 countries) call for transparency and responsible disclosure so that people understand AI outcomes and can challenge them when necessary. In the United States, while regulations are still evolving, there are increasing calls for algorithmic accountability laws that would require impact assessments and explainability for AI used in employment, finance, and other sensitive areas. In summary, there is a clear trend toward formalizing explainability as a legal obligation. From GDPR's individual rights to the EU AI Act's systemic rules, regulators are embedding the ethical principles of explainability and transparency into law, making them prerequisites for AI deployment in high-impact settings. Organizations developing AI must now consider not only what their models can predict, but also how to communicate the reasoning behind those predictions in a human-understandable way.

Explainability in Medical AI: The Case of Alzheimer's Diagnosis

These ethical and legal principles take on urgency in the medical domain. Healthcare decisions are literally life-and-death, and thus the trust, safety, and efficacy of AI systems in medicine are paramount. Medical AI applications—such as diagnostic algorithms for cancer, heart disease, or Alzheimer's—are high-stakes. In this context, model interpretability can directly impact clinician and patient trust, as well as treatment planning. Doctors are trained to base decisions on explainable evidence; they are understandably reluctant to follow a recommendation from a "black box" AI without understanding its rationale. Indeed, many AI-driven Computer-Aided Diagnosis (CAD) tools have struggled to gain acceptance in clinical practice precisely because of their black-box nature. Even if such a model achieves impressive accuracy in detecting a disease, physicians

may question its reliability if it cannot explain its conclusions. As a result, explainability is seen as essential for integrating AI into healthcare – the model's predictions must be interpretable to be trusted. Research on medical AI repeatedly emphasizes this point: the lack of explainability keeps the medical field reluctant to deploy AI diagnostic systems, despite their potential benefits. Clinicians and regulators alike often prefer a slightly less accurate model that behaves transparently and reliably over a more accurate but opaque model. In healthcare, responsible and reliable decision-making is valued over inscrutable efficiency, reflecting an understanding that unexplainable predictions can undermine clinical judgment and patient safety.

In the realm of Alzheimer's disease (AD) diagnosis, the need for explainable AI is especially pronounced. Alzheimer's is a progressive neurodegenerative disease where early and accurate diagnosis is crucial for patient care and planning. AI models have been developed to analyse medical data (such as brain MRI scans, PET images, or cognitive test scores) to detect early signs of Alzheimer's or predict disease progression. Many of these models leverage complex patterns in the data that might elude human doctors. However, if an algorithm flags a patient as likely having Alzheimer's, clinicians will rightly ask "Why?" An uninterpretable prediction offers little actionable insight: without knowing which biomarkers or features influenced the model's conclusion, a doctor cannot confidently incorporate the result into their assessment or explain it to the patient's family. Interpretability can bridge this gap. For example, an explainable AI system might highlight that a patient's hippocampal volume loss and slowed reaction times were the decisive factors in its Alzheimer's prediction. Such information aligns with medical knowledge (the hippocampus is a key brain region affected in AD) and thus grounds the AI's output in terms the clinician finds meaningful. This not only increases the doctor's trust in the model but also directly aids in treatment planning – the care team knows what aspects of the patient's condition to monitor or address. Recent literature reviews of explainable AI in AD support this approach: providing visual or quantitative explanations (e.g. via techniques like SHAP, LIME, or saliency maps) can reveal which input features (genes, brain regions, cognitive tests) contributed most to a model's prediction. Such explanations offer "in-depth insight into the factors that support the clinical diagnosis of AD", essentially opening the black box and translating the algorithm's workings into clinical evidence.

The impact of explainability on trust and safety in Alzheimer's diagnosis cannot be overstated. By making AI's reasoning accessible, we allow clinicians to validate the model's findings against their expert knowledge, dramatically reducing the risk of errors. For instance, if an explainable model for AD bases its prediction on an MRI artifact or noise, a doctor can recognize the spurious reasoning and avoid a misdiagnosis – a safeguard that a black-box model would not afford. Explainability also fosters better communication with patients. Alzheimer's diagnoses are life-changing news; being able to explain why an AI-assisted diagnosis was made (for example, pointing to certain scan findings) can help in counselling patients and families, and in devising personalized intervention strategies. Moreover, from a regulatory standpoint, medical devices that incorporate AI (including diagnostic tools for AD) are increasingly expected to provide transparency. In the EU, AI systems for medical diagnosis would be classified as high-risk under the AI Act, meaning developers must build in explainability and human oversight. This aligns with medical ethics: patients have a right to know the basis of decisions affecting their health, and doctors have a duty to understand the tools they use for care. Thus, ensuring model interpretability in Alzheimer's diagnosis is not only a technical challenge but an ethical and legal necessity. It underpins the trust of both clinicians and patients, fulfils emerging legal obligations for transparency, and ultimately contributes to safer and more effective treatment planning. In critical health contexts like AD, explainable AI is more than a preferable feature – it is imperative for responsible deployment, reflecting the maxim that we owe it to those impacted by AI to ensure the technology is fair, transparent, and aligned with their well-being.

Conclusion

In summary, the literature converges on a clear message: explainability is fundamental to ethical AI, especially in high-stakes applications. Opaque AI models, while powerful, carry significant risks – they can hide bias, obfuscate accountability, and erode trust when their decisions affect human lives. In response, there is a strong movement in both scholarship and policy to demand transparency and interpretability. Accountability, fairness, and the mitigation of bias are not abstract ideals but concrete requirements that translate into design principles and regulations. Laws like the GDPR and the forthcoming EU AI Act codify the expectation that AI decisions be explainable and transparent, reinforcing the notion that ethical obligations and legal compliance

go hand in hand in the AI domain. Nowhere are these needs more pressing than in medicine, where the cost of an unexplained error is measured in human health. In the case of Alzheimer's diagnosis, we see a microcosm of the broader thesis: explainable AI fosters trust, ensures safety, and guides better decisions. As AI systems become ever more embedded in critical decision processes, the importance of explainability only grows. Ensuring that AI can explain itself is not just a technical quest, but a moral imperative to align these systems with human values of transparency, justice, and accountability. The literature reviewed here firmly supports that view, laying an informed foundation for the subsequent chapters on machine learning interpretability and its practical implementation in healthcare AI.

1.3 OBJECTIVES AND SCOPE

The rise of AI in high-stakes domains such as healthcare, finance, and criminal justice has triggered growing concerns around algorithmic opacity and accountability. As models become more complex and powerful, understanding their behaviour is no longer a luxury—it is an ethical imperative. To this end, this research aims to bridge the gap between technical performance and interpretability by developing a unified metric capable of evaluating the explanatory capacity of machine learning (ML) models across diverse contexts.

The general objective of this thesis is to design and validate a unified metric for assessing the interpretability of ML models. This metric integrates multiple dimensions—such as consistency, stability, execution time, and sensitivity to input perturbations—and leverages state-of-the-art tools like SHAP, LIME, Grad-CAM, and causal inference libraries. The metric is intended to offer a standardized, comprehensive framework for evaluating how intelligible a model's decision-making process is to human users. It will initially be applied to specific use cases—most notably, Alzheimer's diagnosis using tabular biomarkers such as LDL cholesterol—and later generalized to additional models and data types. The goal is to enhance transparency, build trust, and promote ethical deployment of AI systems in sensitive applications.

To achieve this overarching aim, the research is structured around the following specific objectives:

1. **Evaluation and Application of Interpretability Tools:**

    a. *Application Across Diverse Datasets:* Apply state-of-the-art interpretability tools—such as SHAP, LIME, and other causality-based libraries—to a variety of datasets that differ in complexity, including tabular data, images—o assess their utility and performance.

    b. *Comparative Analysis:* Conduct a systematic evaluation of each tool in terms of fidelity, consistency, computational efficiency, and feature importance, identifying their relative strengths and limitations.

2. **Development of a Unified Metric:**

    a. *Identification of Key Interpretability Metrics:* Identify and define core interpretability indicators (e.g., consistency, execution time, stability, sparsity, perturbation sensitivity) via literature and empirical analysis.

    b. *Integration into a Unified Metric:* Formulate a composite metric using normalized scores and weighted averages to encapsulate all relevant dimensions.

    c. *Model-Agnostic Validation:* Test this metric across various models (e.g., Logistic Regression, Random Forest, SVM, MLP) and datasets to confirm its robustness and generalizability.

3. **Empirical Evaluation and Critical Analysis:**

    a. *Experimental Pipeline:* Implement a modular pipeline for conducting controlled experiments to test both individual tools and unified metric.

    b. *Critical Review:* Analyze the behavior of the unified metric under different conditions, identifying potential biases and areas for improvement.

    c. *Iterative Improvement:* Refine the metric based on observed shortcomings and propose alternative scoring or weighting schemes if necessary.

4. **Knowledge Generation and Best Practices:**

    a. *Practitioner Guidelines:* Provide evidence-based recommendations for the practical use of interpretability tools tailored to dataset type and model complexity.

    b. *Final Report:* Compile all findings, methodologies, and conclusions into a technically sound and accessible report to serve as a future reference in the field of XAI.

Together, these objectives build upon the current advances in explainable AI (XAI) and address

the urgent need for standardized, unified metrics that can effectively assess the interpretability of diverse machine learning models. By integrating insights from various domains—ranging from financial decision-making and migraine research to Alzheimer's diagnostics—we aim to develop tools that not only enhance transparency but also foster trust and accountability in AI systems.

This project is designed to develop and validate a unified interpretability metric for ML models by addressing various components and challenges across data, models, interpretability tools and validation strategies. The scope is defined as follows:

1. **Datasets:** The project will utilize at least two different types of datasets from open-source websites to ensure a comprehensive evaluation of the proposed metric. These datasets will include:
    a. **Tabular Data:** Structured datasets (e.g., Alzheimer's biomarker data) that require preprocessing for missing values, outliers, and categorical encoding.
    b. **Image Data:** Visual datasets for classification tasks, useful for validating saliency-based techniques like Grad-CAM.
2. **Machine Learning Models:** To thoroughly and exhaustively evaluate the interpretability metric, the study will involve at least three distinct types of ML models that represent diverse methodological approaches, including:
    a. **Linear Models:** Logistic and linear regression, providing transparency through coefficients.
    b. **Tree-Based Models:** Random Forests and Decision Trees, interpretable via feature splits and importance measures.
    c. **Neural Networks:** MLPs and CNNs (for images), treated as black-box models requiring external interpretability techniques.
3. **Interpretability Tools:** These tools will be used to evaluate different facets of interpretability, from local explanations to global causal insight.

   a. **Feature Contribution Analysis:** SHAP, LIME

   b. **Causal Impact Assessment:** Quantitative Input Influence, DoWhy, or similar

   c. **Visual Explanation:** Grad-CAM for image models

4. **Validation:** To rigorously evaluate the proposed unified metric, the project will implement cross-validation techniques across multiple datasets and model types. This validation process will focus on assessing:

   a. **Consistency:** Ensure the stability of the metric across training/test splits.

   b. **Generalization:** Assess how well the unified metric performs across different model architectures and datasets.

   c. **Applicability:** Determine whether the insights offered by the unified metric are actionable and meaningful in real-world contexts.

Finally, this project aligns with the following SDGs:

1. **SDG 3: Good Health and Well-being**: Contributing to the development of more accurate and interpretable ML models in healthcare, particularly for Alzheimer's diagnosis.

2. **SDG 9: Industry, Innovation and Infrastructure**: Through the development of ethical AI frameworks that promote transparency and trust in technological systems.

3. **SDG 16: Peace, Justice and Strong Institutions**: By fostering responsibility and ethical standards in the deployment of AI.

1.4 ORGANIZATION OF THE THESIS

The work presented here is structured as follows:

- **Chapter 1 - Introduction and Context:** This chapter introduces the historical evolution of Artificial Intelligence (AI), from early theoretical milestones like the Turing Test to the emergence of deep learning and the current emphasis on Explainable AI (XAI). It also presents the societal, ethical, and regulatory motivations for developing transparent AI systems, with reference to current debates and frameworks such as the European AI Act. The chapter closes by stating the objectives, scope, and overall structure of the work.

- **Chapter 2 – Theoretical Framework:** This chapter develops the theoretical basis for the study. It reviews the machine learning models applied—such as Logistic Regression, Random Forests, SVMs, and Multi-Layer Perceptron's—as well as interpretability techniques, including SHAP, LIME, Grad-CAM++, and recent developments in causal inference. The chapter also outlines optimization methods, hyperparameter types, and evaluation criteria. Finally, it identifies the limitations of current interpretability tools and introduces the concept of a unified metric designed to integrate multiple dimensions of explainability.

- **Chapter 3 – Methodology:** This chapter details the experimental design and implementation of the proposed approach. It describes the datasets used (tabular and image-based), preprocessing steps, model training, and the integration of interpretability tools. It also explains how the unified interpretability metric was constructed and applied across models and datasets. While Chapter 2 justifies the theoretical choices, this chapter focuses on how those choices were implemented in practice.

- **Chapter 4 – Results and Evaluation:** This chapter presents experimental outcomes. It includes model performance metrics, comparative interpretability visualizations, and quantitative results from the unified metric. Graphical outputs such as ROC curves, SHAP and LIME heatmaps, and stability analyses help visualize how interpretability varies across models, datasets, and explanation tools.

- **Chapter 5 – Discussion, Contributions, and Conclusions:** The final chapter offers a critical analysis of the findings. It discusses the strengths and limitations of the unified metric, its impact on interpretability research, and how it compares to existing tools. It also evaluates the influence of data heterogeneity, preprocessing, and model complexity on interpretability. The chapter concludes by summarizing the main contributions, including methodological and practical insights, and proposes future research directions to improve model explainability and responsible AI deployment.

The work concludes with a comprehensive bibliography that lists all the sources referenced throughout the project.

Chapter 2. Theoretical Framework

This chapter explores the mathematical foundations of popular machine learning models and interpretability methods and proposes a unified framework to integrate and compare these interpretability techniques.

2.1 Machine Learning and Interpretability: Foundations

Machine Learning (ML) is a subfield of artificial intelligence that enables computer systems to learn patterns from data and make predictions without being explicitly programmed. Rather than following fixed rules, ML algorithms learn from example data to improve their performance on specific tasks. They are widely used in practical applications such as recommendation systems, image recognition, and fraud detection.

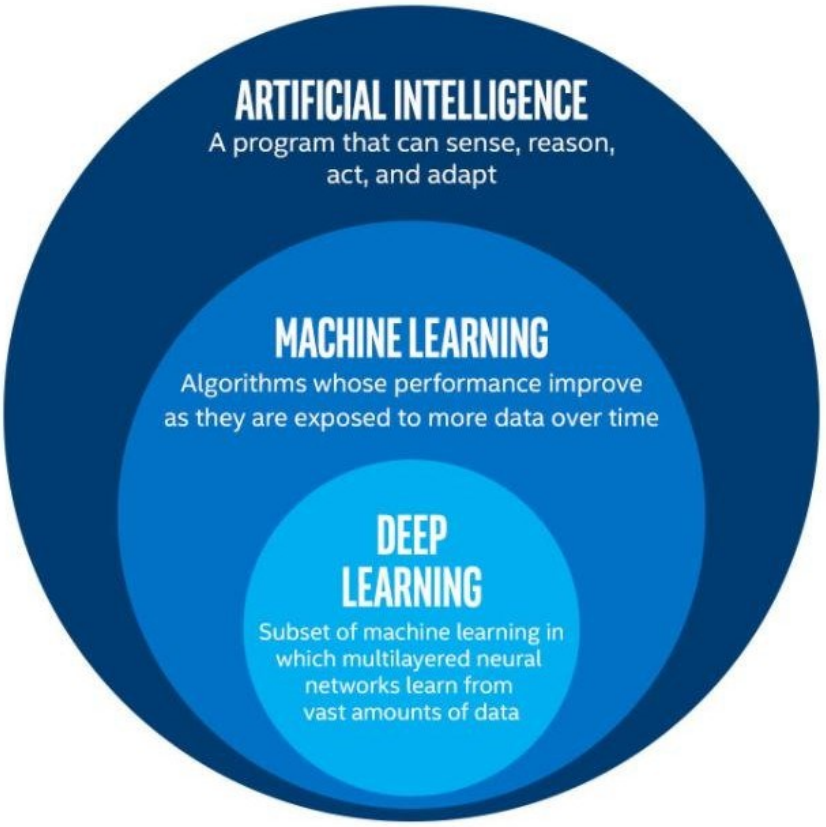


Figure 8: AI vs ML vs DL

Deep Learning (DL), a specialized subset of ML, refers to neural network architectures composed of multiple hidden layers capable of learning hierarchical and complex representations from large datasets. Inspired by the structure of the human brain, these models automatically extract features at varying levels of abstraction. Deep learning has achieved state-of-the-art results in domains such as computer vision and natural language processing. However, this expressiveness comes at the cost of increased computational demands and the need for large-scale data (Li et al., 2022).

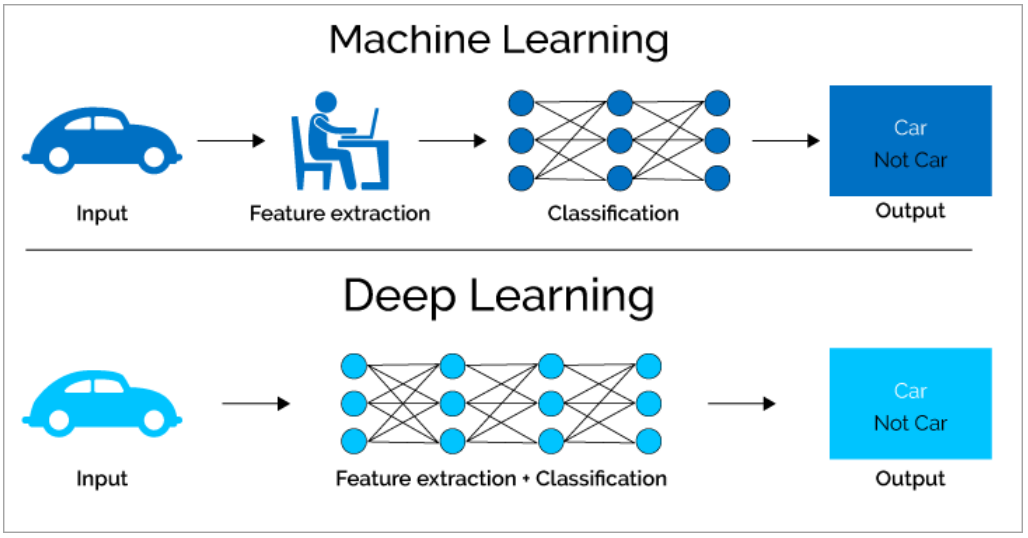


Figure 9:Deep Learning vs Machine Learning. Key Differences

As ML models grow in complexity, interpretability has emerged as a critical concern. Interpretability is often defined as "the degree to which a human can understand the cause of a decision" (Miller, 2019), or alternatively, "the ability to explain or to present in understandable terms to a human" (Doshi-Velez & Kim, 2017). In high-stakes applications—such as healthcare, finance, and autonomous system transparent and trustworthy model behavior is not only desirable but essential for accountability and user acceptance.

Interpretable models, such as linear regression and decision trees, inherently provide insight into how input features affect predictions. However, black-box models—such as neural networks or ensemble methods—often lack this transparency, as their internal mechanisms are complex and non-linear (Rudin, 2019). This creates a trade-off between predictive performance and interpretability, which has led to a surge of interest in techniques that can explain black-box model decisions post hoc.

Interpretability techniques are typically classified along three dimensions:

- **Model-specific vs. model-agnostic:**
  - *Model-specific methods* are tailored to a particular class of models (e.g., decision trees, linear models).
  - *Model-agnostic methods*, in contrast, can be applied to any predictive model, regardless of its structure.

- **Global vs. local:**
  - *Global interpretability* aims to explain the model's overall logic and behavior across the dataset.
  - *Local interpretability* focuses on explaining individual predictions for specific data instances.

- **Intrinsic vs. post hoc:**
  - *Intrinsic interpretability* refers to models that are inherently understandable (e.g., small trees, linear models).
  - *Post hoc methods* are applied after training to explain the decisions of more complex, opaque models.

Among post hoc, model-agnostic techniques, two widely adopted methods stand out (See Section 2.3 to understand their mathematical background):

- **LIME** (Local Interpretable Model-Agnostic Explanations) – proposed by Ribeiro et al. (2016), LIME approximates the local decision boundary of a complex model with a simple, interpretable model (such as a sparse linear regressor) to explain individual predictions.

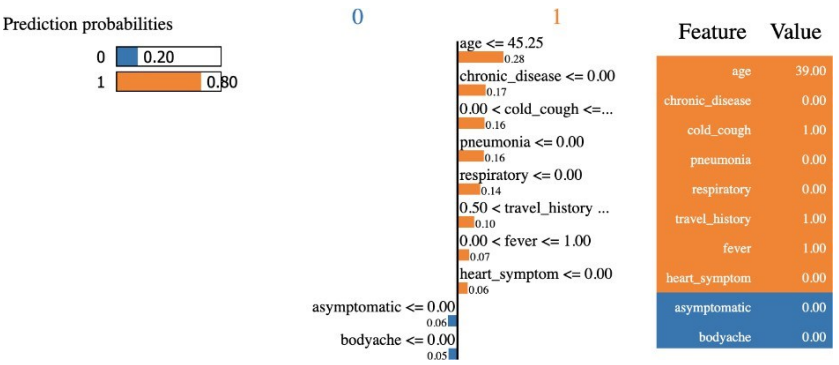


Figure 10: LIME interpretation example

- **SHAP** (SHapley Additive exPlanations) – introduced by Lundberg & Lee (2017), SHAP assigns each feature an importance value based on cooperative game theory, ensuring consistent and locally accurate explanations for any model.

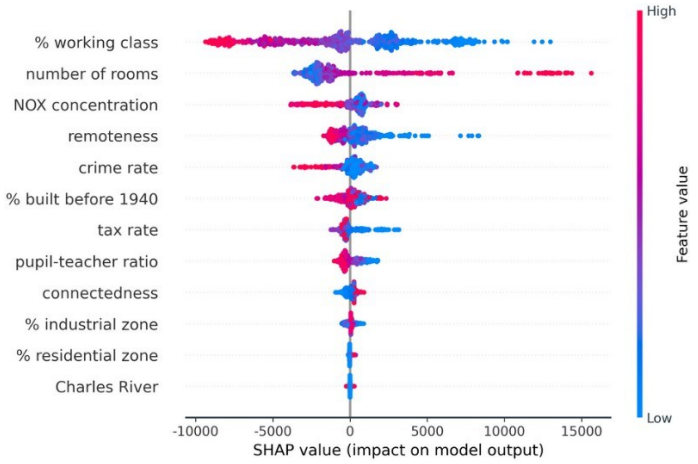


Figure 11:SHAP Analyses example

These tools help uncover the reasoning behind individual predictions, thereby offering a bridge between model performance and interpretability.

The development and adoption of interpretability techniques have led to the emergence of Explainable Artificial Intelligence (XAI), a research field focused on making AI systems more understandable, fair, accountable, and robust. According to Gunning et al. (2019), XAI aims not only to improve transparency but also to enhance user trust and support regulatory compliance in sensitive applications.

In the following sections, we examine several widely used machine learning and deep learning

models—Logistic Regression, Random Forests, Support Vector Machines, and Multilayer Perceptrons—alongside the interpretability challenges they present. We provide mathematical formalism and analysis to understand how these models operate and how interpretability can be achieved either intrinsically or through post hoc techniques.

2.2 MACHINE LEARNING AND DEEP LEARNING MODELS

In this work, a combination of machine learning (ML) and deep learning (DL) models is employed to address the prediction and classification tasks. Among the ML models, **Logistic Regression (LR), Random Forests (RF)**, **and Support Vector Machines (SVMs)** are used due their widespread application, interpretability, and ability to handle structured data. In addition, a **Multilayer Perceptron (MLP)**, a feedforward neural network, is included as a bridge between traditional ML and more complex DL approaches. For image-based tasks, a **Convolutional Neural Network (CNN)** is applied, leveraging its capacity to automatically learn hierarchical spatial features from data. While ML models rely on predefined feature engineering, DL models such as MLPs and CNNs are characterized by their ability to learn non linear feature representations directly from the data, making them particularly suitable for high-dimensional and unstructured inputs.

2.1.1 Logistic Regression (LR)

Logistic regression is one of the most widely used classification algorithms in statistics and machine learning. Unlike linear regression, which predicts a continuous outcome, logistic regression predicts the probability of a categorical outcome, typically binary in nature (e.g., success/failure, disease/no disease). It belongs to the family of generalized linear models (GLMs) where the dependent variable is linked to a linear combination of predictors through a nonlinear function (Hosmer, Lemeshow & Sturdivant, 2013).

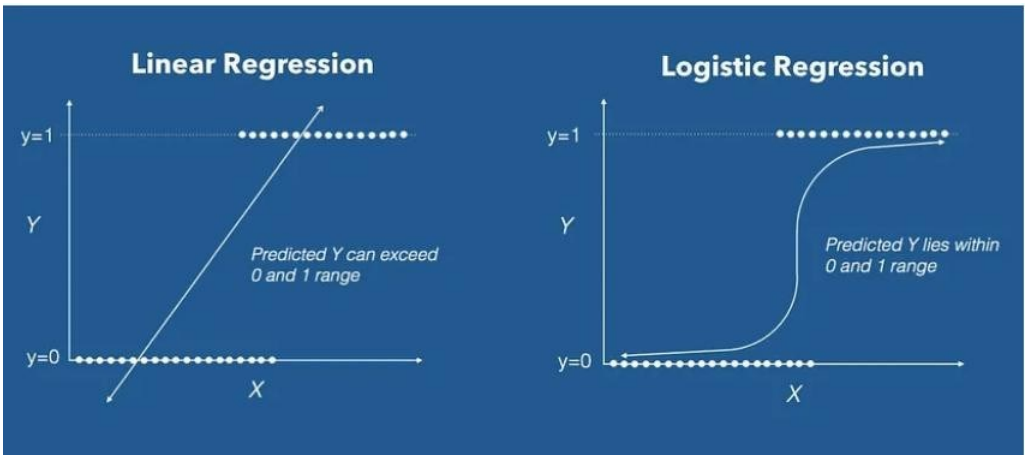


Figure 12: Key differences between Linear Regression and Logistic Regression

Mathematical Formulation:

Mathematically, logistic regression models the probability *p(x)* that the dependent variable *Y* takes value 1 given the input variables *X*:

$$p(x) = \frac{e^{\beta_0+\beta_1 x_1+\cdots+\beta_k x_k}}{1+e^{\beta_0+\beta_1 x_1+\cdots+\beta_k x_k}}$$

Here, β_0 is the intercept, and β_1 are the coefficients corresponding to predictor variables x_i. The **sigmoid (logistic) function** ensures that outputs remain in the range [0,1], suitable for modelling probabilities.

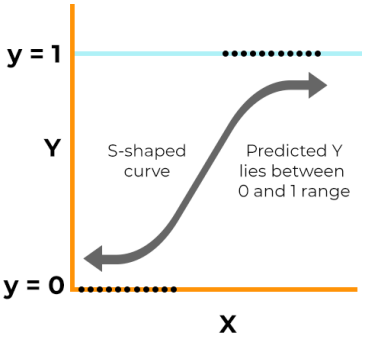


Figure 13: Sigmoid Function

By taking the log-odds transformation (logit), we obtain a linear relationship:

$$\log\left(\frac{p(x)}{1-p(x)}\right) = \beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k$$

Key terms include **odds** (the ratio of the probability of success to failure)**, log-odds** (the logarithmic transformation of odds, producing a linear scale), and the **odds ratio** (e^{β_i}), which quantifies how the odds change with a one-unit increase in predictor x_i (Menard, 2002). Coefficients are estimated using **maximum likelihood estimation (MLE),** which selects the parameters that maximize the probability of observing the given data (Dobson & Barnett, 2008).

Interpretability and Limitations:

From an interpretability perspective, logistic regression has remained foundational in **explainable AI (XAI).** Its coefficients offer direct insights into how predictors affect outcome probabilities. For example, positive coefficients increase the log-odds of the outcome, while negative ones reduce it. The interpretability of the odds ratio makes logistic regression particularly valuable in sensitive fields like healthcare, finance, and social sciences (Maygar, 2020).

In modern XAI, logistic regression often serves as the benchmark for evaluating the transparency of more complex models. Logistic regression is interpretable, but its coefficients (in log-odds) are hard to intuit and lose clarity with nonlinearities, multicollinearity, or many features (Molnar, 2019)

Applications:

Applications of logistic regression span multiple domains. In medicine, it is widely applied in **disease prediction** models, such as estimating the probability of heart disease or Alzheimer's progression (Clifford R Jack et al., 2010). In finance, it supports **credit scoring and fraud detection** by modelling the likelihood of default or abnormal transaction behaviour (Hand & Henley, 1997). In business, it is used for **customer churn prediction**, identifying clients or employees likely to leave **(**Verbeke et al., 2012). Its variants—binary logistic regression, multinomial logistic regression, and ordinal logistic regression—extend its use to different categorical outcome structures.

In summary, logistic regression provides both **statistical rigor and interpretability**, making it a

cornerstone in machine learning and a critical tool for interpretable decision support in high-stakes environments. Its integration of probabilistic foundations, maximum likelihood estimation, and odds-based interpretation ensure its continued relevance, even in the era of deep learning.

2.1.2 Random Forest (RF)

Random Forest is a widely adopted ensemble learning algorithm introduced by Breiman (2001), which builds upon the *decision tree* framework by combining the predictions of multiple trees to improve generalization.

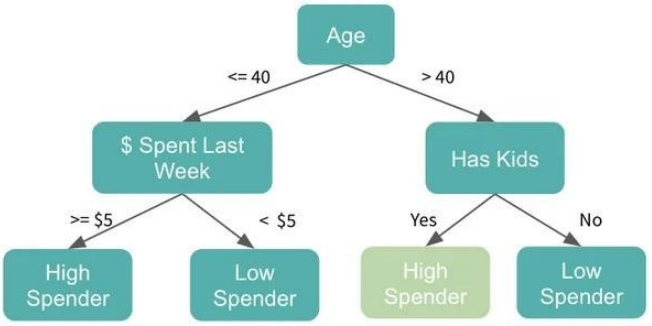


Figure 14: Example of decision tree framework. Determining the optimal point to prune the tree is essential for reducing the risk of overfitting.

While a single decision tree recursively partitions the feature space to make predictions, it is prone to overfitting and instability.

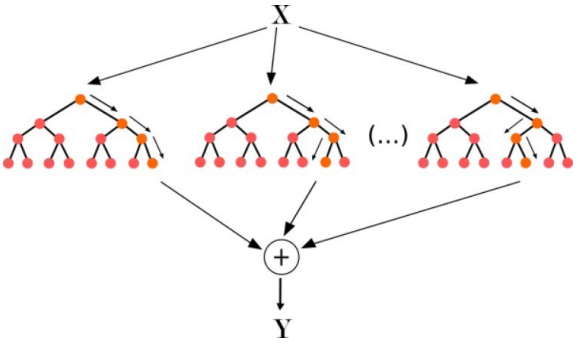


Figure 15: Random Forests reduce overfitting risk by constructing collections of decision trees

Random forest mitigates these issues by constructing a collection of decision tress each trained on a **bootstrap sample** of the dataset and a random subset of features, thus reducing correlation among trees and improving predictive performance.

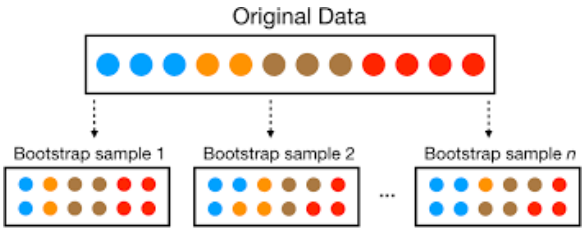


Figure 16: What is bootstrap sampling? A resampling technique used to estimate statistical properties of a model by repeatedly drawing samples from the original dataset with replacement.

Mathematical Formulation

Mathematically, for a classification task, let $T_1, T_2 \ldots, T_B$ represent B decision trees, each trained on a different bootstrap sample. The random forest prediction is given by:

$$\hat{y} = mode\{T_1(x), T_2(x), \ldots, T_B(x)\}$$

For regression tasks, the prediction is the average:

$$y = \frac{1}{B}\sum_{b=1}^{B} T_b(x)$$

Two key sources of randomness make the algorithm robust: (1) *bagging* (bootstrap aggregation), which reduces variance, and (2) *feature bagging*, which ensures trees are diverse by only considering a random subset of predictors at each split (Breiman, 2001). Model performance is commonly validated through the **out-of-bag (OOB)** error estimate, derived from samples not included in bootstrap subsets.

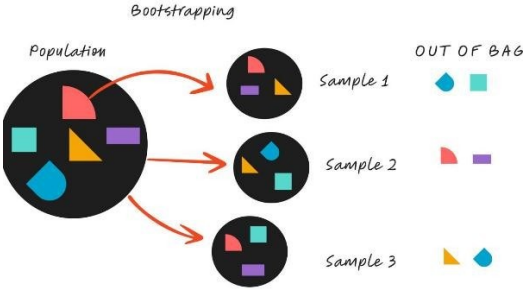


Figure 17: Bootstrapping and the role of Out-of-Bag Samples in identifying important features in Random Forests

Feature Importance Calculation

Random forest offers multiple advantages: it reduces overfitting compared to single decision trees, handles both classification and regression tasks effectively, and provides built-in measures of **feature importance** via Gini importance or permutation-based importance (Louppe et al.,2013). Each feature importance is calculated based on how often it is used for splitting and how much it reduces the impurity at each node.

$$Gini\ impurity = 1 - \sum_{k=1}^{K} p_k^2$$

These characteristics make it a flexible and high-performing algorithm across domains such as finance, healthcare, and e-commerce (Cutler et al.,2007).

Interpretability and Limitations

Despite its practical success, the interpretability of random forests is limited compared to logistic regression or single decision trees. Individual decision trees are inherently interpretable as they follow a sequence of rules, but ensembles obscure this transparency. While feature importance scores provide some interpretability, they do not capture complex interactions among variables, can be biased toward features with more categories, and are often less intuitive for end-users (Strobl et at., 2007). Thus, although random forest balances predictive accuracy with some explanatory tools, its interpretability remains restricted in high-stakes applications where transparency is critical.

2.1.3 Support Vector Machines (SVM)

Support Vector Machines (SVMs) are supervised learning algorithms introduced by Vapnik and colleagues in the 1990s, designed primarily for classification tasks but also extendable to regression through Support Vector Regression (SVR) (Vapnik, 1995). The central idea of SVMs is to construct an optimal **hyperplane** that maximizes the margin between classes in a feature space. Data points closest to the hyperplane, known as **support vectors**, determine the decision boundary and play a critical role in generalization.

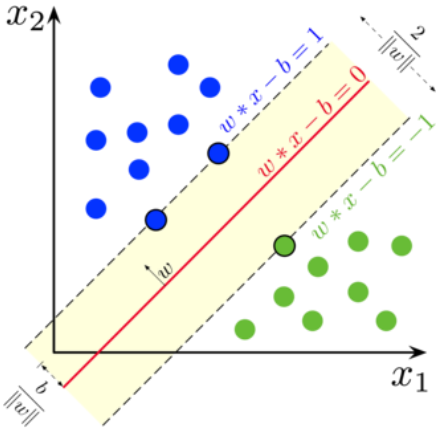


Figure 18: The hyperplane (red) separates the two classes (blue and green), while the support vectors define the yellow margin around it. The aim is to maximize this margin to ensure strong generalization.

Mathematical Formulation: Linearly separable data

Mathematically, the separating hyperplane can be expressed as:

$$wx + b = 0$$

where w is the weight vector, x the input features, and b the bias term. For linearly separable data, the optimization problem seeks to maximize the margin γ:

$$\max \gamma = \frac{1}{\|w\|}$$

subject to the constraints:

$$y_i(wx_i + b) \geq 1, \qquad \forall i$$

Mathematical Formulation: Linearly separable data

For non-linearly separable data, SVMs employ the **kernel trick**, which implicitly maps inputs into higher-dimensional feature spaces using kernel functions (e.g., linear, polynomial, radial basis function (RBF), sigmoid) (Schölkopf & Smola, 2002). This allows SVs to capture nonlinear decision boundaries efficiently.

When data are **not linearly separable**, we map the input features x into a higher-dimensional feature space F using a mapping function $\phi(x)$:

$$f(x) = sign(w \cdot \phi(x_i) + b)$$

The optimization problem becomes:

$$\min_{w,b,\xi} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{n} \xi_i$$

subject to:

$$y_i(w \cdot \phi(x_i) + b) \geq 1 - \xi_i, \;\; \xi_i \geq 0 \;\; \forall i$$

where $\xi_i$are **slack variables** (for misclassifications) and C is a **regularization parameter** controlling the trade-off between margin and maximization and classification error.

The dual formulation eliminates w and uses **kernel functions** $K(x_i, x_j) = \varphi(x_i) \cdot \varphi(x_j)$, avoiding explicit computation in the high-dimensional space:

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j K(x_i, x_j)$$

Subject to:

$$\sum_{i=1}^{n} \alpha_i y_i = 0, \;\; 0 \leq \alpha_i \leq C$$

Here, $\alpha_i$are **Lagrange multipliers** and the **support vectors** are the data points with nonzero α_i.

The final decision function becomes:

$$f(x) = sign(\sum_{i=1}^{n} \alpha_i y_i K(x_i, x))$$

Common kernel functions include:

1. **Polynomial kernel:** $K(x_i, x_j) = (x_i \cdot x_j + 1)^d$
2. **Radial Basis Function (RBF):** $K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2)$
3. **Sigmoid kernel:** $K(x_i, x_j) = \tanh(kx_i \cdot x_j + \theta)$

Interpretability and its Limitations

While SVMs provide strong predictive performance, their interpretability is limited. In linear SVMs, coefficients can be inspected to gauge feature importance, but this becomes less intuitive than in logistic regression since they relate to the hyperplane geometry rather than probability (Mangasarian, 1998). In kernelized SVMs, interpretability decreases further, as the decision function relies on complex transformations in high-dimensional feature spaces that obscure direct relationships between input variables and outcomes. Moreover, support vectors themselves are often numerous and not easily mapped back to human-understandable rules, making SVMs closer to "black box" models compared to inherently interpretable methods (Molnar, 2019).

Applications

SVMs are highly effective in domains requiring robust classification in high-dimensional spaces. Applications include **text classification** (e.g., sentiment analysis, spam detection), **image recognition** (object detection, tampering detection), **bioinformatics** (protein classification, cancer diagnosis), and **geoscience** (geophysical data filtering and seismic risk prediction) (Noble, 2006; Hsu et al., 2010).

2.1.4 Multilayer Perceptron (MLP)

The Multilayer Perceptron (MLP) is a class of feedforward artificial neural network that extends the original perceptron by incorporating one or more hidden layers between the input and output layers. As shown in Figure 19, each input (e.g., SNP features) is propagated through successive layers of neurons, where each neuron applies a weighted transformation followed by a nonlinear activation. This hierarchical structure enables MLPs to capture complex nonlinear relationships and approximate arbitrary continuous functions, as formalized by the Universal Approximation Theorem (Hornik, 1991).

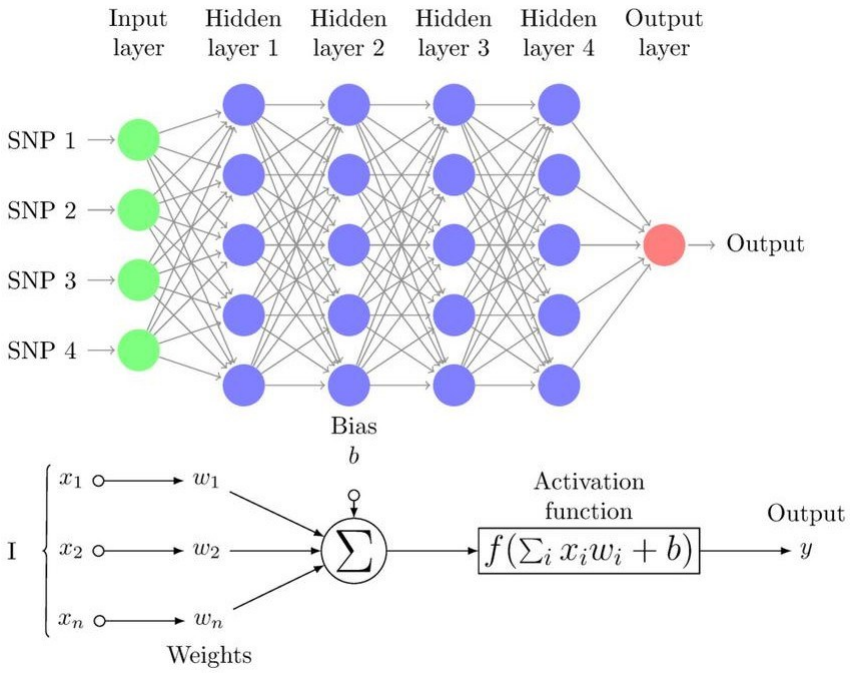


Figure 19: Example of Multilayer Perceptron

Mathematical Formulation

Let the input vector be:

$$x = (x_1, x_2, \ldots, x_n)$$

In layer *l,* the pre-activation of neuron *j* is:

$$z_j^{(l)} = \sum_{i=1}^{d_{l-1}} w_{ij}^{(l)} h_i^{(l-1)} + b_j^{(l)}$$

where $w_{ij}^{(l)}$ are the weights, $b_j^{(l)}$ the bias term, and $h_i^{(l-1)}$ the outputs from the previous layer (with $h^{(0)} = x$). The activation function then produces:

$$h_j^{(l)} = f(z_j^{(l)})$$

Common choices for $f(\cdot)$ include the sigmoid function $\sigma(z) = \frac{1}{1+e^{-z}}$, hyperbolic tangent $f(z) = \frac{1-e^{-2z}}{1+e^{-2z}}$, and the Rectified Linear Unit (ReLU), $f(z) = \max_z(0, z)$.

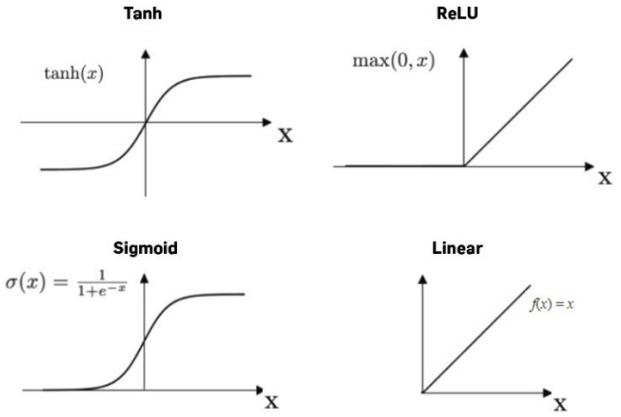


Figure 20: Common activation functions in neural networks. Sigmoid and Tanh squash inputs to bounded ranges but can cause vanishing gradients. ReLU is efficient and widely used in deep networks, while the Linear function preserves inputs and is mainly applied in regression tasks.

The final layer yields the network prediction:

$$\hat{y} = f(W^{(L)} h^{(L-1)} + b^{(L)})$$

where L is the number of layers. Training is performed through **backpropagation** (Rumelhart, Hinton & Williams, 1986), which computes gradients of a cost function with respect to weights and biases using the chain rule, combined with an optimization algorithm such as stochastic gradient descent (SGD).

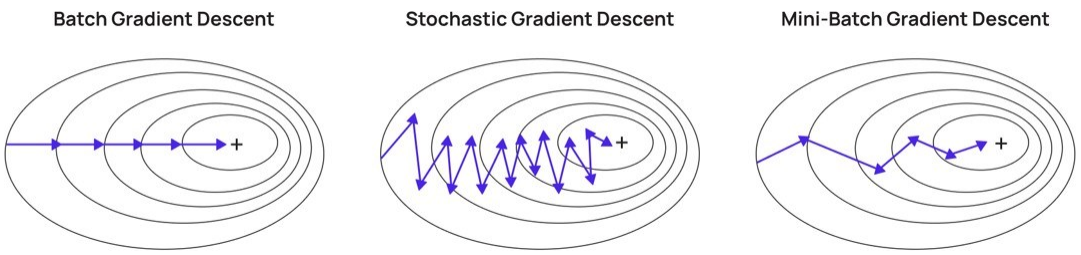


Figure 21: Batch Gradient Descent (left) computes the gradient using the entire dataset, leading to stable but computationally expensive updates. Stochastic Gradient Descent (middle) updates parameters after each sample, introducing noise but allowing faster convergence. Mini-Batch Gradient Descent (right) combines both strategies by updating parameters using small subsets of data, balancing efficiency and stability.

Interpretability and Limitations

The interpretability of MLPs is significantly lower than that of simpler models such as logistic regression or decision trees. While linear models provide direct coefficient-based interpretations, the hierarchical and distributed representation of information within MLPs makes it difficult to assign clear meaning to individual parameters. Feature attribution methods such as SHAP and LIME have been introduced to approximate local feature importance, but these provide partial and model-dependent explanations (Ribeiro, Signh & Guestrin, 2016; Molnar, 2019).

Limitations include:

1. **Black-box nature:** Parameters are not directly interpretable.
2. **Overfitting risk:** High model capacity requires regularization and large datasets.
3. **Computational cost:** Training deep MLPs demands substantial hardware resources.
4. **Hyperparameter sensitivity:** Performance depends heavily on architecture and parameter tuning.

Applications

MLPs have been applied across multiple domains due to their flexibility in modeling nonlinear relationships:

- **Natural Language Processing:** Sentiment analysis, spam detection.
- **Computer Vision:** Early image classification tasks prior to the development of convolutional networks.

- **Bioinformatics:** Prediction of disease risk and genomic pattern recognition (e.g., SNP-based classification).
- **Finance:** Credit scoring, fraud detection.

Their capacity to learn high-level feature representations makes MLPs a foundational model in deep learning even though interpretability challenges remain central in sensitive application areas.

2.1.5 Convolutional Neural Networks (CNN)

Convolutional Neural Networks (CNNs) are a class of **deep learning** models designed to process grid-like data structures such as images, audio spectrograms, or video frames. Unlike traditional machine learning models, which rely on manually engineered features, CNNs automatically learn hierarchical feature representations directly from raw input data.

Their architecture is characterized by three key components: **convolutional layers**, which extract local spatial features; **pooling layers**, which reduce dimensionality and promote translational invariance; and **fully connected layers**, which integrate learned features for classification or regression tasks (LeCun et al., 1998; Krizhevsky et al.,2012).

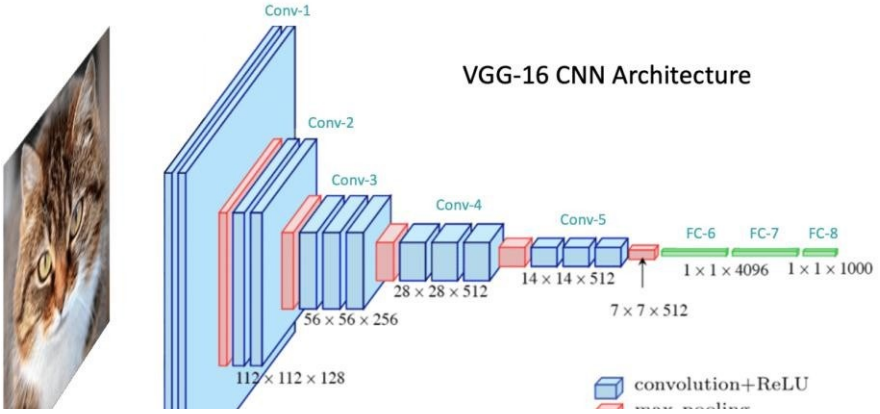


Figure 22: VGG-16 CNN Architecture. The input image is processed through multiple convlutional layers (blue) with ReLU activations and max pooling layers (red), progressively extracting high-level features.

Mathematical Formulation

The central operation of a CNN is the **convolution.** Given an input image *I* and a kernel (filter) *K,* the convolution at position *(i,j)* is:

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(i+m, j+n) \cdot K(m,n)$$

This operation slides the kernel across the image, producing a **feature map** that captures local patterns such as edges or textures. Nonlinear **activation functions** (e.g., ReLU) are then applied:

$$h^{(l)} = f(S(i,j) + b)$$

where *b* is a bias term.

To reduce computational complexity and enforce invariance, **pooling layers** (e.g., max-pooling) are introduced:

$$p(i,j = \max_{(m,n)\epsilon\Omega} h(i+m, j+n)$$

where Ω denotes the pooling region. Finally, the output of the las convolutional and pooling stages is flattened and passed into **fully connected layers** (green in Figure 22) to produce the prediction $\hat{y}$.

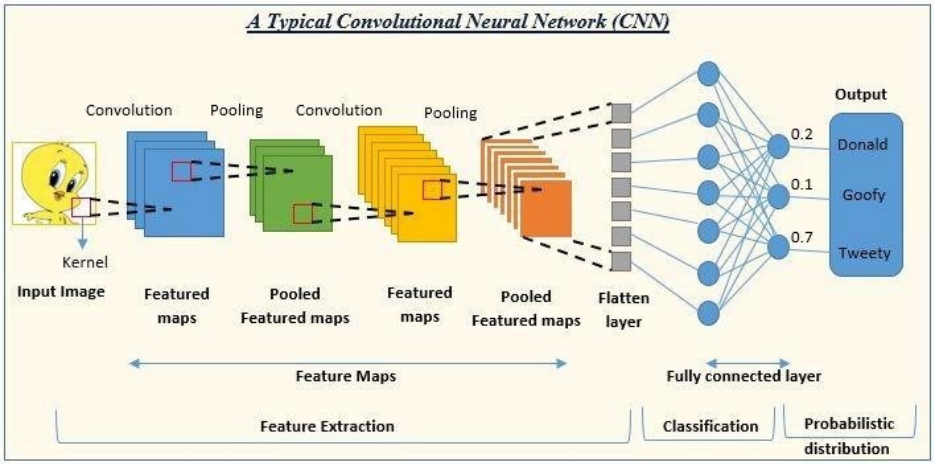


Figure 23: Example of CNN

Interpretability and Limitations

CNNs achieve state-of-the-art performance in visual recognition tasks, but they are often criticized for their **black-box nature.** While the learned filters in early layers can sometimes be visualized (e.g., edge detectors), deeper layers capture abstract and less interpretable patterns (Zeiler & Fergus, 2014). To improve interpretability methods such as Grad-CAM (Selvaraju et al., 2017) highlight which regions of an input contribute most to a prediction.

Key limitations include:

- **Lack of transparency**: Feature hierarchies are difficult to interpret compared to ML models.
- **High computational cost**: Training requires large datasets and specialized hardware (GPUs/TPUs).
- **Sensitivity to adversarial perturbations**: Small input changes can drastically alter predictions.
- **Overfitting risk:** Without regularization or data augmentation, CNNs may memorize training data.

Applications

CNNs are widely applied across domains that require processing unstructured, high-dimensional data:

- **Computer Vision:** Image classification, object detection, facial recognition, and medical imaging.
- **Natural Language Processing (NLP):** Sentence classification and document categorization (using word embeddings).
- **Healthcare:** Detection of tumors, Alzheimer's disease progression, and medical image segmentation.
- **Autonomous Systems:** Real-time visual perception in self-driving cars and robotics.

Their capacity to automatically learn feature hierarchies from raw inputs has made CNNs a cornerstone of modern deep learning.

2.3 HYPERPARAMETER OPTIMIZATION AND GRID-SEARCH

Machine learning models depend not only on data and training algorithms, but also on **hyperparameters**—external configuration variables chosen *before* learning (e.g., regularization, tree depth, network width). Unlike **model parameters** (coefficients, weights), which are estimated from data, hyperparameters control the hypothesis space, the optimization dynamics, and ultimately the bias-variance trade-off. Selecting them well is critical to generalization and computational efficiency (Hastie, Tibshirani & Friedman, 2009; Bergstra & Bengio, 2012).

In the following section, we provide a detailed explanation of the key hyperparameters for the models used in this study—**Logistic Regression, Support Vector Machines, Random Forests, and Multilayer Perceptrons**—highlighting their mathematical formulation, interpretation, and limitations. We then introduce the theoretical foundation of **Grid Search**, the systematic optimization strategy employed to identify the best-performing hyperparameter configurations.

Key Hyperparameters in the Models Used

Logistic Regression (LR)

- **Regularization strength (C):** LR models a conditional probability via the logistic link. With $y_i \epsilon \{0,1\}$and feature vector x_i,

$$\Pr(y_i = 1 \mid x_i) = \sigma(\beta_0 + x_i^T \beta)$$

It uses regularization to prevent overfitting by penalizing large coefficients (Hosmer, Lemeshow & Sturdivant, 2013). The optimization objective is:

$$\min_{\beta} \mathcal{L}(\beta) + \lambda \|\mathcal{L}(\beta)\|_p$$

$$with\ C = {}^{1}/_{\lambda}\ and\ \mathcal{L}(\beta) = -\sum_{i}[y_i \log \hat{p}_i + (1 - y_i) \log(1 - \hat{p}_i)$$

A smaller *C* implies stronger regularization, shrinking coefficients and improving generalization, while a larger *C* reduces the penalty, allowing more complex fits.

- **Penalty and solver:** The penalty defines the type of regularization (L1, L2, or elastic net), while the solver specifies the optimization algorithm (e.g., *liblinear*, *lbfgs*, *saga*). These choices impact both numerical stability and interpretability.

Support Vector Machines (SVM)

- **Regularization strength (C):** Similar to LR, *C* controls the trade-off between maximizing the margin and allowing misclassifications (Cortes & Vapnik, 1995).

- **Kernel:** Defines how data are projected into higher-dimensional spaces:

$$K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$$

  Popular kernels include linear, polynomial, and radial basis function (RBF).

- **Gamma (γ)**: In RBF kernels, gamma determines the radius of influence of each support vector:

$$\mathrm{K}(x_i, x_j) = \exp\left(-\gamma \|x_i - x_j\|^2\right)$$

  High γ produces highly localized decision boundaries (risk of overfitting), whereas low γ creates smoother, more generalized classifiers.

Random Forest (RF)

- **Number of estimators (*n_estimators*)**: Specifies how many trees are grown in the ensemble (Breiman, 2001). Increasing the number of trees typically improves stability and reduces variance, though at a higher computational cost.
- **Maximum depth (*max_depth*)**: Restricts the depth of each tree, controlling complexity. Deep trees fit complex patterns but may overfit, while shallow trees enhance generalization.

Multilayer Perceptron (MLP)

- **Hidden layer sizes**: Determine the number of neurons per hidden layer, which directly controls network capacity. By the **Universal Approximation Theorem**, sufficiently

large networks can approximate any continuous function (Hornik, 1991).

- **Activation and Solver**: Activation functions (sigmoid, tanh, ReLU) introduce nonlinearity, while solvers (e.g., SGD, Adam) dictate how weights are updated. These choices shape both convergence speed and predictive accuracy (Rumelhart, Hinton & Williams, 1986; Kingma & Ba, 2015).

Below you'll find a summary of all the hyperparameters described:

Hyperparameter	Model	Description
C	LR, SVM	Controls the regularization strength. Smaller values imply stronger regularization.
penalty/solver	LR	Determines how the model penalizes weights and solves the optimization problem.
kernel	SVM	Specifies how input features are transformed into higher-dimensional spaces.
gamma	SVM	Defines the influence of radius of support vectors. High gamma leads to more overfitting.
n_estimators	RF	The number of decision trees in the ensemble.
max_depth	RF	Maximum depth of each tree.
hidden_layer_sizes	MLP	Controls network complexity and capacity.
activation/solver	MLP	Determines non-linear transformations and optimization dynamics.

Table 1: Hyperparameters descriptions

Grid Search: Theory and Workflow

To identify optimal hyperparameters, this study employs **grid search,** a systematic strategy that evaluates models across all possible combinations of specified hyperparameter values.

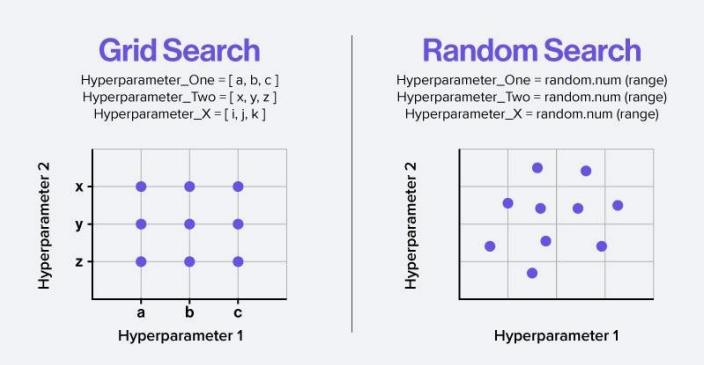


Figure 24: GridSearch concept

Mathematical Formulation

Given k hyperparameters with candidate sets $H_1, H_2, \ldots, H_k$, the full search space is:

$$\varkappa = H_1 x\, H_2 x \ldots x\, H_k$$

For each configuration $\theta \in \varkappa$, the model is trained and validated. Performance is typically assessed through **k-fold cross-validation**, where the score is:

$$Score\,(\theta) = \frac{1}{K} \sum_{i=1}^{K} Metric(M_{\theta}^{(i)})$$

with $M_{\theta}^{(i)}$ denoting the model trained on the *i-th* fold. The best hyperparameter set is then chosen as:

$$\theta^* = \underset{\theta \in \varkappa}{\operatorname{argmax}}\, Score(\theta)$$

Workflow

The workflow of grid search can be outlined as:

1. **Define the search space**: Select hyperparameters and specify candidate values.
2. **Train and evaluate models**: Train the model on all possible hyperparameter combinations and record validation metrics.
3. **Select the best configuration**: Identify the hyperparameters achieving the highest average performance.

Advantages

- **Exhaustive**: Guarantees that the best configuration within the defined search space is found.
- **Transparent and reproducible**: Each step is systematic, making results reliable for academic and regulated contexts.
- **Simple to implement**: Widely supported in machine learning libraries (e.g., Scikit-learn).

Limitations

- **Computational cost**: The number of evaluations grows exponentially with the number of hyperparameters (curse of dimensionality).
- **Rigid boundaries**: If the optimal value lies between tested grid points, it may be missed.
- **Inefficiency**: Equal computational effort is spent on both promising and unpromising regions of the space.

2.4 INTERPRETABILITY TOOLS: SHAP, LIME, GRAD-CAM++, AND OTHER CAUSALITY TECHNIQUES

Overview of Post – Hoc Model Interpretability Methods

Post-hoc interpretability methods aim to explain complex "black box" models after they have been trained, without altering the models themselves. These techniques provide insights into why a model made a certain prediction by analyzing the model's behavior around specific instances or overall. Generally, post-hoc explanations can be categorized by scope (e.g. local explanations for individual predictions vs. global explanations for overall model behavior) and by methodology (e.g. model-agnostic tools that treat the model as a black-box vs. model-specific tools that leverage internal model structure). In contrast to intrinsically interpretable models (like simple linear models or decision trees built for interpretability), post-hoc methods allow us to interpret highly non-linear or high-dimensional models (such as ensemble methods or deep neural networks) without sacrificing predictive performance.

A conceptual taxonomy of post-hoc methods includes surrogate models, feature attribution methods, saliency and gradient maps, and counterfactual/causal analysis, among others. Surrogate models (like LIME) learn an interpretable approximation of the black box in a limited region (providing local fidelity). Feature attribution methods (like SHAP and Integrated Gradients) assign each feature an importance value for a given prediction, often satisfying certain axioms (e.g. fairness or completeness). Saliency methods (like Grad-CAM/Grad-CAM++ for images) produce visual heatmaps highlighting parts of the input that most influenced the model's output. Counterfactual and causality-based techniques, on the other hand, focus on "what-if" scenarios – they explain model decisions by indicating how changes in inputs cause changes in the output,

aiming for explanations aligned with causal reasoning. Each approach has distinct theoretical underpinnings and practical uses, as discussed below.

SHAP (SHapley Additive Explanations)

1. *Conceptual description:*

SHAP is a unified framework for interpreting model predictions using ideas from cooperative game theory. It treats feature influences as player's contributions in a game, where the "payout" is the model's predictions for a given instance. The SHAP value ϕ_i for feature *i* represents the contribution of that feature to the prediction, answering the question: *"How would the model's output change if feature i were at its present value versus if it were absent?"*. By considering *all possible subsets S* of features, SHAP provides an additive feature attribution that fairly distributes the prediction difference among features. In practical terms, SHAP produces a set of feature importance values (positive or negative) for each instance, which sum up to the model's prediction minus a baseline (typically the dataset average prediction). This allows both local interpretability (explaining individual predictions) and, when aggregated across many instances, global interpretability.

2. *Mathematical Formulation:*

Formally, let ϕi be the model prediction for instance x and let M be the full feature set (|F| = M features). For any subset of features $S \subseteq F$, denote $f_x(S)$ as the model's prediction given that the feature S are set to their values in x and all other features and all other features F \ S are "absent" (e.g. replaced by some baseline or mean value). The Shapley value for feature i in instance x is defined as:

$$\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!\,(M - |S| - 1)!}{M!} [f_x(S \cup \{i\}) - f(S)]$$

which averages the contribution of feature I over all possible subsets that could precede it. By construction, SHAP values satisfy the properties of:

i. **Local accuracy (completeness):** the predictions are exactly decomposed as:

$$f(x) = g(x') = \phi_0 + \sum_{i=1}^{M} \phi_i x_i'$$

When approximating the original model *f* for a specific input *x,* local accuracy requires the explanation model to at least match the output of *f* for the simplified input x' (which corresponds to the original input x).

ii. **Missingness:** If the simplified inputs represent feature presence, then missingness requires features missing the original input to have no impact.

$$x_i' = 0 \rightarrow \phi_i = 0$$

iii. **Consistency:** If a model changes such that a feature's contribution increases (holding others constant), its ϕ_i will not decrease. Notably, it can be proven that SHAP is the unique additive feature attribution method that satisfies these properties.

In essence, SHAP values are an exact solution (for linear models) or an approximate solution (for complex models via sampling) to the Shapley axions of fair credit allocation.

3. Theoretical Strengths:

A major strength of SHAP is its strong theorical theoretical foundation. By inheriting properties from Shapley values, SHAP provides consistency and fairness in feature attribution – each feature's influence is fairly distributed even in presence of interactions. The completeness (efficiency) property means the explanation is self-contained: the sum of attributes equals the prediction difference, which builds trust that no influence is left unaccounted. SHAP's consistency axiom ensures that comparisons of feature importance across models are meaningful (if model f' relies more on feature i than model f does, then ϕ_i in f' will be $\geq \phi_i$ in f).

Moreover, SHAP values provide local fidelity by construction – they exactly match the model

output for that instance when summed, unlike some heuristics. Another advantage is the ability to derive global insights: by examining summary statistics of ϕ_i over a dataset, one ranks features by overall importance, detects interactions, and visualizes how features affect predictions (e.g. SHAP dependence plots).

4. *Practical Strengths:*

In practice, SHAP has become popular due to these guarantees and its flexibility. It is model agnostic when using approximation algorithms (like *Kernel SHAP*, which estimates Shapley values repeated evaluations on feature subsets), so it can explain any classifier or regressor. At the same time, specialized fast implementation exists for certain model types (e.g. TreeSHAP for three ensembles, DeepSHAP for deep networks), making it relatively efficient on those. The explanations are additive and feature-based, which are intuitive to interpret – e.g. a positive ϕ_i indicates feature i pushed the prediction higher (toward Alzheimer's) and negative ϕ_i lowered the prediction (protective effect). These qualities have led to SHAP's wide adoption in domains like finance and healthcare, where fairness and completeness of explanations are valued.

5. *Weaknesses and Limitations:*

Despite its merits, SHAP has some limitations. A key challenge is computational complexity: calculating exact Shapley values requires evaluating $O(2^M)$ feature subsets, which is intractable for large M. Approximations (sampling or leveraging model structure) are used, but SHAP can still be considerably slower than simpler methods like LIME, especially for models without a special SHAP algorithm. Another issue is the handling of feature dependence: the classic Shapley formulation assumes each feature can be "missing" independently, which in practice is modelled by either sampling from a distribution or using conditional expectations. If features are highly correlated, SHAP's attributions can be non-intuitive – it will distribute credit among collinear features, sometimes giving two redundant features each a moderate importance where a human might say one is truly driving the outcome. This "fair sharing" of importance is mathematically consistent but can confuse interpretation (e.g. two correlated biomarkers each get half the credit for an Alzheimer's prediction, even if one is the actual causal factor and the other is just proxy).

Furthermore, SHAP inherits a "local" perspective in that the ϕ_i explain one specific prediction relative to a baseline; using them for global insight requires careful aggregation and can miss context (for example, SHAP can tell how a feature influenced *this* patient's diagnosis versus baseline, but not directly how risk changes in absolute terms with feature changes).

Finally, SHAP's explanations – a list of numeric contributions – must be presented thoughtfully to be understood by lay users, and for very high-dimensional data, interpreting dozens of feature values can be challenging despite the method's theoretical clarity.

6. *Computational Efficiency and Practical Considerations:*

In practice, applying SHAP involves a trade-off between accuracy and speed. For complex models, Kernel SHAP (model-agnostic) uses a weighted least squares sampling to estimate ϕ_i; it is *accurate but can be slow*, requiring many model evaluations for convergence. Fortunately, many commonly used models have faster SHAP algorithms: TreeSHAP runs in polynomial time leveraging the tree structure, and DeepSHAP uses connections to DeepLIFT to propagate contributions efficiently. These improvements often make SHAP feasible even for large datasets or real-time use (with caching of background samples). Memory can be a consideration, as storing Shapley values for every instance and feature can be heavy; in practice one often looks at summary plots or a few selected instances. Another practical point is the need to choose a baseline or background distribution. By default, SHAP uses the dataset's average prediction as f = E[f(x)], but in healthcare one might choose a clinically meaningful baseline (e.g. baseline patient with "normal" health metrics) to make the explanations more semantically meaningful. Mis-specifying the baseline can shift all ϕ_i values and thus should be set with care. Despite these considerations, SHAP is generally regarded as *consistent and reliable* for feature attribution – for example, a study found that SHAP's completeness constraint can act like a regularizer, yielding more stable (lower-variance) explanations compared to LIME in high-dimensional problems.

7. *Typical Applications:*

SHAP is used across many domains for both debugging models and communicating insights. In finance, it helps explain credit risk models by showing which factors (income, debt, etc.) contributed

to a loan denial. In the context of healthcare and Alzheimer's diagnosis, SHAP has proven particularly valuable. Many predictive models for Alzheimer's (e.g. those using clinical data, biomarkers, or cognitive test results) are complex black-boxes (random forests, deep neural nets); integrating SHAP allows researchers and clinicians to identify which features (e.g. age, certain MRI-based volume measures, cognitive scores, genetic markers like APOE4 status) are driving each individual prediction. By aggregating these, one can check if the model's reasoning aligns with medical knowledge – for instance, confirming that low hippocampus volume and memory test deficits receive high positive SHAP values for AD prediction, which matches known causative factors. A recent systematic review found that SHAP (along with LIME) is one of the *most popular XAI frameworks for AD prediction*, used in numerous studies to strengthen trust in AI-based diagnoses. SHAP has been applied to *both* classical machine learning models on tabular patient data and deep learning models for image-based AD diagnosis, thanks to the existence of DeepSHAP for networks. Its ability to provide both local patient-specific explanations and global feature importance aligns well with the needs in healthcare for transparency at both the individual and population level.

8. *Limitations in Healthcare Context:*

One must note, however, that SHAP (like any correlation-based attribution) does not guarantee *causal* correctness. In medicine, there are often hidden confounders or biases in data. SHAP will faithfully explain the model's prediction – but if the model learned a spurious pattern (e.g. MRI scanner machine type inadvertently correlating with AD vs. control labels), SHAP may highlight scanner-related features as "important", which is a correct explanation of the model but not a clinically valid reason. Therefore, experts must carefully interpret SHAP outputs, validating that highlighted features make medical sense. Encouragingly, many studies report that SHAP's explanations of AD models do align with known risk factors and biomarkers (improving clinicians' trust), while also occasionally revealing surprising associations that prompt further investigation. In summary, SHAP offers a powerful, theoretically sound tool for post-hoc explainability, with high relevance to domains like Alzheimer's research, provided its outputs are used as aids to insight rather than unquestionable truths.

LIME: Local Interpretable Model-Agnostic Explanations

1. *Conceptual Description:*

Local Interpretable Model-Agnostic Explanations (LIME) is a technique designed to provide intuitive, local explanations for individual predictions made by complex machine learning models. The core principle behind LIME is that, while a model may be too complex to interpret globally, its behaviour in the vicinity of a specific prediction can often be approximated by a simpler, interpretable model. To achieve this, LIME perturbs the input data around a given instance and observes how the black-box model's predictions respond. A surrogate model—typically a sparse linear regression or a shallow decision tree—is then trained on these perturbed samples to mimic the black-box model's decision boundary locally. The surrogate model serves as a proxy, offering insight into the most influential features for that specific prediction.

LIME is model agnostic, treating the underlying predictor as a black box. It only requires the ability to obtain predictions from the model for various inputs, making it applicable to any classifier or regressor regardless of its internal architecture. This flexibility enables LIME to be applied across different data modalities, including tabular, text, and image data.

2. *Mathematical Formulation:*

Let $f: \mathbb{R}^M \rightarrow \mathbb{R}$ represents the complex, black-box model, and let $x \in \mathbb{R}^M$ denote the specific instance to be explained. LIME (Local Interpretable Model-agnostic Explanations) provides a local surrogate explanation by approximating the model's behaviour in the neighbourhood of x.

To this end, LIME generates a set of perturbed samples $Z_x = \{z_1, z_2, \dots, z_n\}$ around the instance x. Each perturbed instance $z \in Z_X$ is assigned a weight based on its proximity to x, using a kernel function $\pi_x(z)$, typically an exponential kernel such as:

$$\pi_x(z) = \exp\left(-\frac{D(x,z)^2}{\sigma^2}\right)$$

Where $D(\cdot,\cdot)$ denotes a distance metric (e.g., Euclidean distance) and σ controls the kernel width.

The goal is to fit a simple, interpretable model $g \in G$ — for example, a sparse linear model with at most *K* non-zero weights —that faithfully mimics the behavior of the original model *f* in the vicinity of x. This is achieved by minimizing the following objective:

$$\mathcal{L}(f, g, \pi_x) + \Omega(g)$$

Where $\mathcal{L}$ measures the local fidelity (e.g., mean squared error between *f(z)* and *g(z)* weighted by $\pi_x(z)$), and $\Omega(g)$ is regularization term enforcing interpretability (e.g., and L0 or L1 norm constraint to enforce sparsity).

This formulation ensures that the surrogate model g is both locally faithful to the original model and globally simple enough to be interpretable by humans.

3. *Theoretical Strengths:*

One of LIME's primary theoretical advantages lies in its model-agnostic nature, enabling its application across any predictive model type without requiring access to internal parameters or gradients. It supports flexibility in choosing the form of the surrogate model and the definition of interpretable features. For instance, in image classification, interpretable features may be super pixels, while in text classification, they could be individual words or phrases.

Moreover, LIME explanations are typically sparse, aligning with human intuition and cognitive limitations; people tend to prefer explanations involving only a small number of salient factors. The optimization formulation ensures that the surrogate model prioritizes fidelity to the original model in a small neighbourhood, while maintaining a constraint on complexity, which is critical for generating human-readable insights.

4. *Practical Strengths*

In practice, LIME is appreciated for its ease of use, visual interpretability, and broad applicability. Open-source libraries allow seamless integration with trained models, making it accessible for practitioners and researchers. The explanations can be visualized using bar plots (for tabular data) or heatmaps (for image data), which are easily interpretable by domain experts and non-technical

stakeholders.

LIME is particularly useful for debugging models, as it can reveal when a model is relying on spurious or irrelevant features to make a prediction. In healthcare, for example, LIME has been applied to explain models predicting Alzheimer's Disease from clinical and imaging data. Such explanations might highlight age, genetic markers, or cognitive test scores as the key contributors to a diagnosis, providing actionable insights for clinicians.

Furthermore, LIME can be used to compare models by generating explanations for the same instance using different algorithms, thereby offering a consistent framework for evaluating and validating model behaviour across architectures.

5. Weaknesses and Limitations

Despite its advantages, LIME has several notable limitations. First, its explanations are strictly local, meaning they are valid only for the immediate neighbourhood around the instance being explained. As a result, LIME cannot provide a global view of feature importance and may yield inconsistent or contradictory explanations for similar instances if the model's decision boundary is highly non-linear or unstable.

Additionally, the technique is sensitive to sampling and kernel parameters. The choice of kernel width, number of perturbed samples, and method of perturbation can significantly affect the resulting explanation. This sensitivity can undermine trust, especially in high-stakes domains such as healthcare, where reproducibility and stability are essential. Although fixing random seeds or averaging over multiple runs can reduce variance, such strategies do not fully eliminate the issue.

Another limitation is related to feature correlation. When features are strongly correlated, LIME may arbitrarily attribute importance to one over the other, which may be misleading. This is especially problematic in medical data, where many features (e.g., cognitive scores) are interdependent. Furthermore, LIME does not ensure completeness—unlike SHAP, its attributions do not sum to the prediction output, and the surrogate model may not faithfully represent the true decision boundary even locally if it is overly complex or non-linear.

Lastly, in terms of computational efficiency, while LIME is fast for structured and textual data, it can become computationally expensive for high-dimensional inputs like images, particularly when querying deep neural networks multiple times. In such cases, the generation of explanations may need to be adjusted (e.g., fewer samples or coarser feature representations) to remain tractable.

Grad – CAM++

Gradient-weighted Class Activation Mapping (Grad-CAM), proposed by Selvaraju et al. (2017), is a post-hoc visual interpretability technique tailored for convolutional neural networks (CNNs). Its primary objective is to generate class-discriminative localization maps that identify the spatial regions in an input image most influential for a specific output class. Unlike saliency methods such as Guided Backpropagation or Deconvolution—which provide visually sharp but class-agnostic maps—Grad-CAM yields explanations that are both spatially informative and class-aware, making it suitable for tasks like image classification, visual question answering, and image captioning.

The core idea of Grad-CAM is to utilize gradient information flowing into the last convolutional layer of a CNN. Let y^c represent the score for a target class c and let $A^k \in \mathbb{R}^{uxv}$ denote the k-th feature map in the final convolutional layer, where u x v corresponds to its spatial dimensions.

To assess the importance of each feature map A^k for class c, Grad-CAM computes a weight α^c_k using global average pooling over the gradients of y^c with respect to A^k:

$$\alpha_k^c = \frac{1}{Z} \sum_i \sum_j \frac{\partial y^c}{\partial A^k_{ij}}$$

Where $Z = u$ x v is the total number of pixels in the feature map. The coefficient α^c_k captures the significance of feature map k in determining the class score y^c.

The Grad-CAM heatmap $L^c_{Grad-CAM} \in \mathbb{R}^{uxv}$ is the total number of pixels in the feature maps, followed by a ReLU activation to retain only positive contributions:

$$L^c_{Grad-CAM} = ReLU(\sum_k \alpha^c_k A^k)$$

The ReLU ensures that only features positively influencing the class of interest are visualized,

which enhances interpretability by suppressing irrelevant or misleading activations.

This localization map can be up sampled to match the input image resolution and overlaid as a heatmap, providing intuitive insights into the model's decision-making process by highlighting where the model is "looking" when predicting a particular class.

Causal Techniques

Causal interpretability aims to go beyond correlational insights by uncovering cause-and-effect relationships between input features and model outputs. Unlike traditional interpretability tools that often identify associations, causal methods explicitly ask: *What would happen if we changed a specific input feature while keeping all others constant?* This idea lies at the heart of counterfactual reasoning, a foundational principle in causal inference, as formalized by Judea Pearl. In the context of image classification, for instance, a counterfactual explanation would highlight what minimal changes to the image would have led the model to predict a different class, thereby offering actionable and realistic interventions.

A complementary approach is causal mediation analysis, which seeks to decompose the effect of an input variable into direct effects and indirect effects mediated through internal representations or latent variables. This technique allows researchers to quantify how information is processed within the model and to identify critical layers or pathways responsible for a decision. By doing so, mediation analysis provides insights into potential biases, redundancies, or bottlenecks within deep neural architectures.

Recent advances also explore embedding causality directly into model architectures to enhance interpretability from the ground up. A notable innovation in this space is the Kolmogorov–Arnold Network (KAN), proposed by Zhou et al. (2024). KANs draw inspiration from the Kolmogorov–Arnold representation theorem, which states that any multivariate continuous function $f(x_1, x_2, \dots, x_n)$ can be represented as a sum of univariate function compositions:

$$f(x_1, x_2, \dots, x_n) = \sum_{q=0}^{2n} \phi_q \left(\sum_{p=1}^{n} \psi_{q,p}(x_p) \right)$$

In the KAN framework, this decomposition is implemented by replacing traditional linear transformations (i.e., matrix multiplications) in neural networks with learnable univariate functions, often implemented as spline interpolations. This architectural change enables fine-grained analysis of how individual input dimensions influence the network's output, making KANs especially amenable to feature-level causal interpretability.

KANs offer several advantages:

(i) **Interpretability**: The model structure inherently supports understanding how each input contributes to the output.

(ii) **Sparsity**: In many applications, only a few input dimensions significantly influence the result, making it easier to isolate causal effects.

(iii) **Modularity**: The compositional nature of KANs enables clean attribution of predictions to specific inputs or intermediate computations.

While KANs have demonstrated strong performance in structured data tasks (e.g., tabular data), ongoing research is exploring their applicability to more complex modalities such as image and sequence data.

In conclusion, Grad-CAM provides an effective and computationally efficient method for localizing discriminative regions in visual inputs, thereby improving the transparency of deep learning models. However, as the demand for trustworthy and actionable explanations increases, especially in domains like medicine, it is crucial to complement correlation-based tools like Grad-CAM with causality-informed methods. Techniques such as counterfactual analysis, causal mediation, and architectures like Kolmogorov–Arnold Networks collectively represent a promising direction for the future of interpretable and causally sound artificial intelligence.

2.5 LIMITATIONS OF EXISTING TOOLS AND THE NEED FOR A UNIFIED METRIC. PROPOSED UNIFIED SHAP-LIME METRIC

While tools such as SHAP (Shapley Additive exPlanations) and LIME (Local Interpretable Model-agnostic Explanations) have become foundational in the field of explainable artificial intelligence (XAI), they are not without limitations —particularly when evaluated in isolation across different use cases, model architectures, or data modalities.

SHAP offers strong theoretical guarantees, including consistency and local accuracy, yet it is often computationally expensive, especially for complex models like deep neural methods. Moreover, its reliance on feature independence assumptions or kernel approximations in certain configurations (e.g., Kernel SHAP) may introduce distortions when input features are correlated.

LIME, by contrast, excels in speed and flexibility, generating sparse linear approximations around individual predictions. However, it lacks guarantees of fidelity or stability. Results fluctuate across runs due to its randomized sampling approach, and the absence of a global perspective may mislead when used as the sole interpretability lens.

These tools, while complementary, emphasize different interpretability dimensions — such as local approximation, computational cost, and human readability. As a result, it becomes difficult to compare them objectively without a shared evaluation framework. Interpretability, by nature, is multi-dimensional, requiring a balance between accuracy, robustness, sparsity, and usability.

This motivates the need for a unified metric: a composite evaluative score that consolidates critical interpretability properties (e.g., fidelity, stability and sparsity) into a single standardized framework. Such a metric enables more informed model selection and interpretability tool choice across domains — especially in high-stakes areas like healthcare, where interpretability must be both accurate and clinically actionable.

Chapter 3. Methodology

3.1 Data Understanding and Description

3.1.1 Overview of the datasets (tabular, image)

This study utilizes two distinct datasets to model Alzheimer's Disease: a structured tabular dataset and a brain MRI image dataset, each sourced from reputable public repositories on Kaggle. Together, these datasets allow for complementary modeling approaches—structured data analysis through traditional machine learning and deep learning, and visual pattern recognition through convolutional neural networks (CNNs).

Tabular Dataset

The tabular dataset, obtained from Kaggle, contains comprehensive medical, demographic, and lifestyle data for 2,149 patients. Each record includes a unique identifier along with 35 features (Annex I) that span across several domains: demographic information (e.g., age, gender, ethnicity, education level), lifestyle variables (e.g., alcohol consumption, smoking habits, physical activity), medical history (e.g., cardiovascular disease, diabetes, depression), clinical measurements (e.g., cholesterol levels, blood pressure), cognitive and functional assessments (e.g., MMSE, ADL), and presence of Alzheimer's-related symptoms (e.g., confusion, disorientation, personality changes).

The target variable, Diagnosis, is binary and indicates whether the patient is diagnosed with Alzheimer's Disease (1) or not (0).

Image Dataset

The image dataset, sourced from Kaggle, comprises MRI scans of the brain, categorized into four diagnostic classes: *MildDemented*, *ModerateDemented*, *VeryMildDemented*, and *NonDemented*. Each category includes multiple subjects, with the most representation found in the *NonDemented* and *VeryMildDemented* classes. The dataset was curated to avoid biases present in previous versions—specifically, it ensured that train, validation, and test sets were split randomly rather than sequentially by slice location.

3.1.2 EXPLORATORY DATA ANALYSIS (EDA)

Exploratory data analysis (EDA) revealed several patterns of interest:

Binary Variables:

Figure 1 displays the distribution of binary features such as *MemoryComplaints* and *BehavioralProblems* in relation to Alzheimer's diagnosis. Diagnosed individuals exhibit a higher frequency of these symptoms, suggesting their relevance as early clinical indicators.

Continuous Variables:

The histograms in *Figure 2* illustrate the distribution of key cognitive and functional metrics:

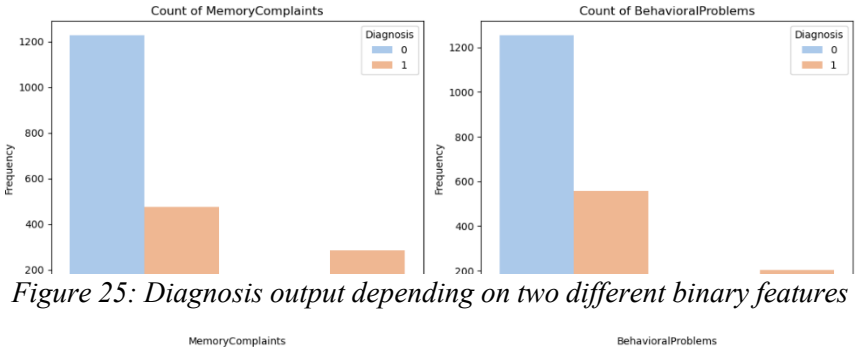
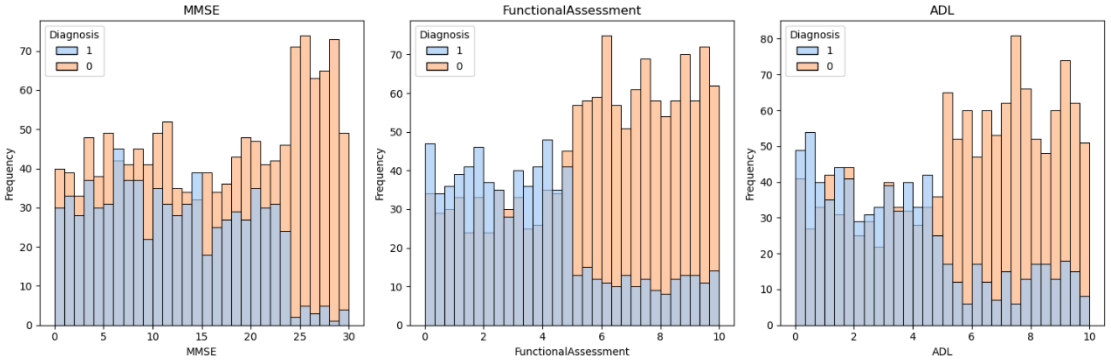


Figure 25: Diagnosis output depending on two different binary features

MMSE, *FunctionalAssessment*, and ADL. Lower values in these variables are strongly associated with the presence of Alzheimer's, confirming their clinical utility for classification purposes.

Correlation Matrix:

As depicted in *Figure 3*, the correlation heatmap reveals moderate[1] associations between symptoms



(*MemoryComplaints* and *BehavioralProblems*) and diagnosis outcomes. Additionally, cognitive

[1] As a role of thumb, it will be considered that two variables are highly correlated if the Pearson coefficient is above 0.7.

scores such as MMSE and ADL show inverse correlations with the diagnosis variable, supporting their predictive value. There is no apparent correlation between the features.

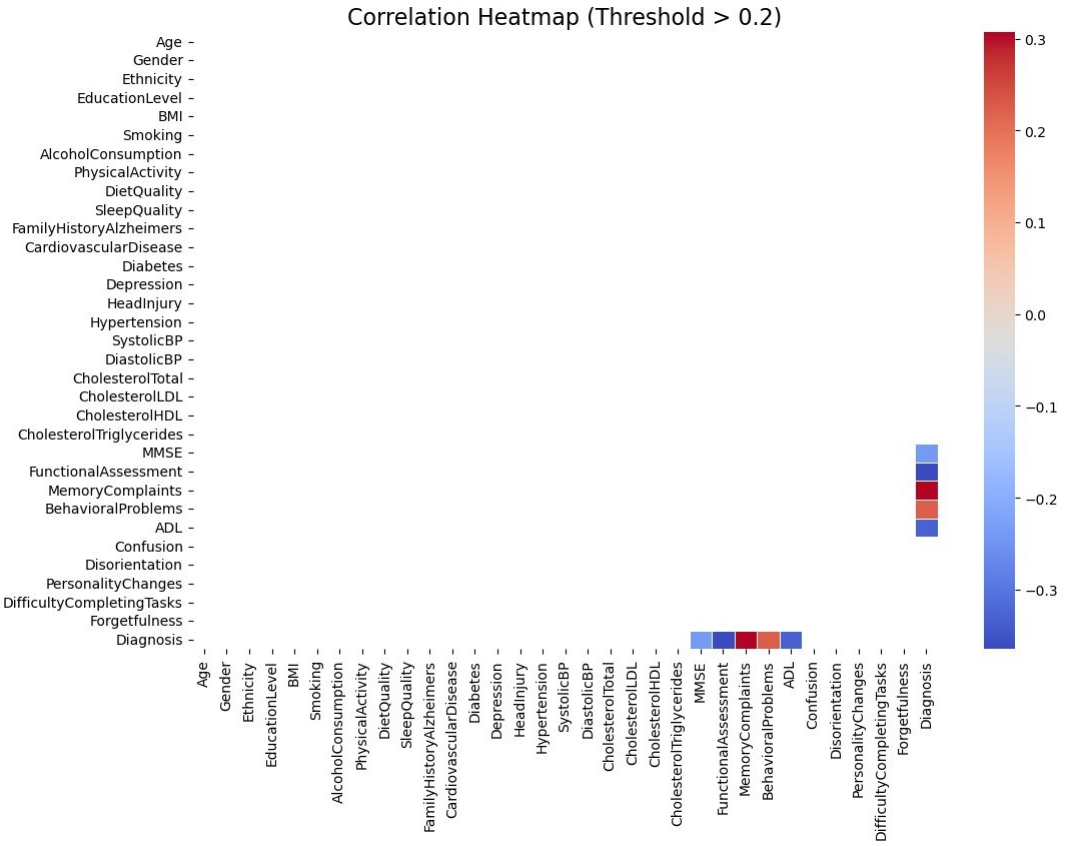


Figure 27: Correlation matrix between features

Distribution and Outliers:

Figure 4 (See Annex II) presents the overall variable distributions. Most features are either uniformly or right skewed. While several outliers exist—particularly in lipid profile variables like cholesterol and triglycerides—they were retained to reflect realistic population variability.

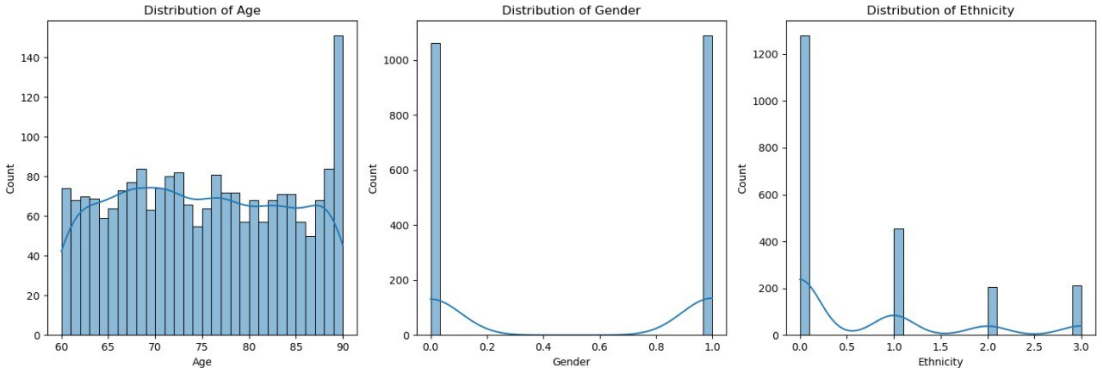


Figure 28: Examples of feature datapoints distributions

Boxplots:

Figure 5 (See Annex II) provides a comparative overview of feature ranges and central tendencies using boxplots. These visualizations assist in identifying potential scaling discrepancies and highlight the diversity in variable distributions across patients. No outlier is detected.

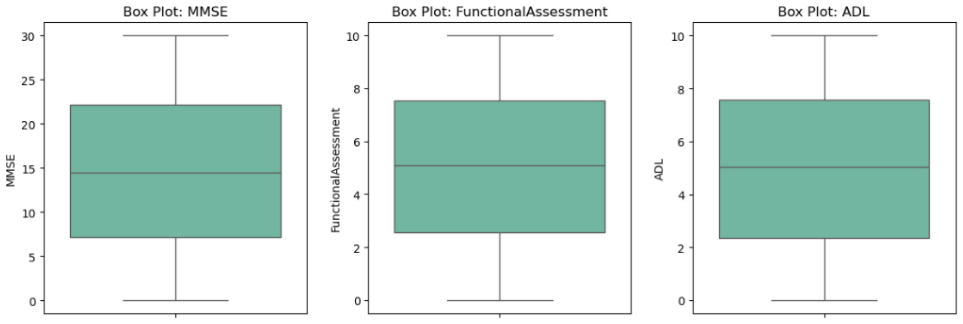


Figure 29: Features Boxplots

Data Quality:

No significant missing values were observed in the dataset. Categorical features were labelled using *OneHotEncoding*, and continuous variables were standardized to ensure model compatibility and improve convergence during training.

3.2 TABULAR DATA MODELLING

This section details the methodology followed for modelling and interpreting tabular data associated with Alzheimer's Disease prediction. The dataset contains various clinical, cognitive, and lifestyle-related features. Our modelling approach spans across traditional

machine learning (ML) models and a deep learning (DL) model, evaluated in three successive optimization phases. Interpretability is central to this pipeline and was implemented using SHAP and LIME frameworks.

3.2.1 Selected Models: Machine and Deep Learning Overview

Four classification models were selected based on their interpretability capabilities, learning paradigms, and comparative value:

1. **Logistic Regression (LR)**: A linear baseline model offering high interpretability via feature weights. Suitable for identifying direct linear relationships.
2. **Support Vector Machine (SVM)**: A margin-based classifier utilizing kernel functions to model non-linear decision boundaries.
3. **Random Forest (RF)**: An ensemble of decision trees aggregating multiple predictions for robust and non-linear modelling.
4. **Multi-Layer Perceptron (MLP)**: A fully connected neural network capable of capturing more complex data structures through hidden layers and activation functions.

3.2.2 Selected Models: Preprocessing Steps

Initial preprocessing steps were designed to ensure data quality and consistency:

- **Cleaning**: Identifiers such as *PatientID* and *DoctorInCharge* were dropped to prevent data leakage and as they didn't provide any information to *Diagnosis* output.
- **Visualization and Exploration**: Histograms, boxplots, and count plots were used to identify distributions and potential outliers.
- **Handling Imbalances**: The binary target variable (Diagnosis) was imbalanced (approx. 65% negative). While no resampling was applied, this imbalance was accounted for using interpretability techniques and metrics such as Cohen's Kappa.
- **Correlation Analysis**: Highly correlated features were visualized via heatmaps, and redundancies were removed to preserve interpretability and model robustness.

3.2.3 Experimental Design: Three-Phase Optimization Strategy

To rigorously evaluate and compare the performance and interpretability of the selected machine learning models, a structured three-phase optimization strategy was implemented. This approach enabled a systematic exploration of model capabilities, the impact of targeted hyperparameter tuning, and the benefits of comprehensive optimization.

1. Baseline Evaluation with Default Parameters

In the initial phase, each model was trained and evaluated using the default parameter settings provided by the Scikit-learn library. This baseline assessment served two primary purposes:

1. To establish a **reference point** for **model performance** prior to any tuning, and
2. To evaluate the **inherent strengths** and **weaknesses** of each model's architecture.

The models assessed in this phase included Logistic Regression (LR), Support Vector Machine (SVM), Random Forest (RF), and Multilayer Perceptron (MLP). Performance metrics such as accuracy, F1 score, and ROC-AUC were recorded to facilitate direct comparison with subsequent optimization phases.

2. Targeted Single-Parameter Optimization

The second phase focused on optimizing a single, high impact hyperparameter for each model. This targeted approach was designed to identify which parameter most immediately influenced model performance and interpretability. The specific hyperparameters selected for tuning were as follows:

- **Logistic Regression (LR):** Regularization strength (C)
- **Support Vector Machine (SVM):** Regularization strength (C)
- **Random Forest (RF):** Minimum samples leave.
- **Multilayer Perceptron (MLP):** Hidden layer

For each model, a range of plausible values was explored for the chosen parameter while all other settings remained at their default values. This allowed for the isolation of the parameter's effect on classification accuracy, F1/F2 scores, and interpretability (as measured by feature importance

rankings or model coefficients).

3. COMPREHENSIVE HYPERPARAMETER OPTIMIZATION

In the final phase, each model underwent extensive hyperparameter tuning using grid search combined with cross-validation. The following parameters were included in the optimization process for each model:

- **Logistic Regression:** Penalty type (*l1, l2, elasticnet*), solver (*liblinear, lbfgs, saga*), regularization strength (C), maximum iterations (*max_iter*), and l1_ratio (for *elasticnet*).
- **Random Forest:** Number of estimators (*n_estimators*), maximum tree depth (*max_depth*), minimum samples required to split a node (*min_samples_split*), and maximum number of features considered for splitting (*max_features*).
- **Support Vector Machine:** Regularization parameter (C), kernel function, and kernel coefficient (gamma).
- **Multilayer Perceptron:** Hidden layer sizes, activation function, and solver.

Grid search was employed to systematically explore combinations of these hyperparameters, with five-fold cross-validation used to ensure robust performance estimates and to mitigate overfitting. The primary evaluation metrics included cross-validation accuracy, F1 and F2 scores, ROC-AUC, and Cohen's Kappa score.

3.2.4 EVALUATION OF PREDICTIONS AND INTERPRETABILITY ANALYSIS

To properly assess model results, we examine a suite of metrics, each highlighting a different aspect of diagnostic performance:

1. **Accuracy (Overall Correctness):** The fraction of all cases (both Alzheimer's and non-Alzheimer's) that the model correctly classifies. For example, an accuracy of 82% means 82% of patients were correctly labeled as either AD or healthy. While useful as an overall gauge, accuracy alone can be misleading in imbalanced datasets – if 65% of our sample are healthy controls (class 0), a trivial classifier that predicts "No Alzheimer's" for everyone would be 65% accurate. Thus, accuracy must be interpreted alongside more class-sensitive metrics in our moderately imbalanced dataset.

2. **Specifity (True Negative Rate):** The proportion of actual non-Alzheimer's cases that the model correctly identifies as such (TN/(TN+TP)). This tells us how well the classifier avoids false alarms. In our data ~65%[2] of subjects are non-AD, so specifity is particularly important to interpret in context. A high specifity (closer to 1.0) means most healthy individuals are correctly cleared, whereas a low specifity is the contrary. We will observe that this metric will vary depending on the model.

3. **Sensitivity (Recall for AD, True Positive Rate):** The proportion of actual Alzheimer's cases correctly identified (TP/(TP+FN)). This metric is crucial in medical context, as it measures how well we catch the disease. In Alzheimer's diagnosis, false negatives are especially concerning – a missed diagnosis means a patient who has AD goes untreated or unprepared. Therefore, we prioritize maximizing recall to minimize the chance of missing an AD case, even if it means tolerating some false positives, ensuring Alzheimer's disease cases do not slip unnoticed.

4. **F1 Score (Harmonic Mean of Precision and Recall):** The F1 combines precision (the proportion of predicted AD cases that are truly AD) and recall into a single number.It is calculated as $2 \cdot (\text{Precision} \cdot \text{Recall}) / (\text{Precision} + \text{Recall})$. An F1 score balances the trade-off between catching most AD cases (recall) and avoiding too many false alarms (precision). In our context, F1 is useful to get a sense of overall test performance when both false positives and false negatives carry weight. For instance, an F1 around 0.75 indicates a good balance – the model is reasonably sensitive while also maintaining decent precision. We use F1 to compare models when neither class should be completely neglected. However, because in this scenario we consider recall slightly more important than precision, we also look at an alternative.

5. **F2 Score (Recall-Weighted Harmonic Mean):** F2 is like F1 but gives double weight to recall (sensitivity). This metric is defined as $(1 + 2^2) \cdot (\text{Precision} \cdot \text{Recall}) / (4 \cdot \text{Precision} + \text{Recall})$. It emphasizes catching positives (AD cases) over avoiding false alarms. A higher

[2] We note that our dataset's class imbalance could have been addressed with resampling techniques like SMOTE (Synthetic Minority Over-sampling Technique) to generate synthetic AD cases, balancing the training data and potentially improving recall of the minority class. Without such adjustment, a model can skew toward predicting the majority class to maximize accuracy and specificity.

F2 indicates the model is achieving strong recall even if precision is somewhat lower – which is valuable in our Alzheimer's screening context where missing a diagnosis is more harmful than a false alert. For example, a model with F2 = 0.93 on the test set demonstrates excellent sensitivity with acceptable precision, making it highly suitable for identifying AD patients. We compare F2 scores to see which model best meets the clinical priority of maximizing recall.

6. **Cohen's Kappa (Chance-Adjusted Agreement):** Kappa measures how much better the classifier is than random guessing or a simple majority-class strategy, by accounting for chance agreement. It ranges from 0 (performance no better than chance) to 1 (perfect agreement with true labels), with negative values indicating worse than chance. Kappa is especially informative with class imbalance – it penalizes a model that just predicts the majority class. For instance, a model that labels everyone as healthy might be 65% accurate, but Kappa ≈ 0, indicating it's only doing as well as chance given class frequencies. In our results, we see Kappa scores around 0.60–0.87 for the better models, which signifies substantial agreement beyond chance. A high Kappa (e.g. 0.87) means the model is performing far better than a baseline that simply guesses based on class prevalence. We look to Kappa to ensure that high accuracy isn't just due to class imbalance – it confirms the classifier is truly discriminating AD vs. healthy cases beyond what chance or majority voting would achieve.

7. **AUC (Area Under the ROC Curve):** A threshold independent measure of discrimination. AUC represents the probability that the model ranks a randomly chosen AD patient higher (in terms of predicted risk) than a randomly chosen healthy person. An AUC of 0.5 means no discriminative power (random guessing), while 1.0 is perfect separation. In our evaluation, AUC scores in the 0.84-0.96 range indicate the models have good to excellent ability to distinguish AD from No AD across all possible thresholds. Notably, AUC can reveal the model's potential even if a fixed threshold (like 0.5) isn't optimal. For example, one model had a poor recall at default threshold but still showed AUC > 0.80, suggesting that with threshold adjustment or tuning it could achieve much better sensitivity.

8. **Calibration curve:** This assesses how well the model's predicted probabilities reflect the

true likelihood of disease. A well-calibrated model will output on 1-1 relationship between prediction and true class frequency, say, 0.80 probability for AD and truly be correct about 80% of the time for those cases. Our model has been calibrated with the following categories:

a. **Well calibrated:** Curve closely follows the diagonal.

b. **Overconfident:** Predictions are too high (e.g., predicts 0.9 but actual is 0.75)

c. **Underconfident:** Predictions are too conservative (e.g., predicts 0.5 but actual is 0.7).

d. **Moderately calibrated:** Acceptable but with notable deviation from diagonal.

e. **Uncalibrated:** Poor alignment; systematic over- or underestimation across all bins.

Calibration curves thus inform us whether we can trust the probability scores or if we should post-process them (for example, via Platt scaling[3] or isotonic regression) to improve reliability.

9. **Prediction Histogram (Probability Distribution):** The probability histograms show how separated the model's predicted probability distributions are for the two classes. In further sections, we have used the following qualitative labels:

   a. **Well – separated:** Predicted probabilities for Y=0 and Y=1 is clearly distinct.

   b. **Overlapping:** Significant overlap in probability distributions for both classes.

   c. **Left-skewed:** Predictions clustered near 0 for both classes (model too conservative).

   d. **Right skewed:** Predictions clustered near 1 for both classes (model too optimistic).

[3] However, it is worth noting that on further studies of LR models as a black box Platt scaling has less effect, as well as for MLP and RF. It has been shown effective for SVMs

e. **Flat:** Uniform prediction distribution – often signals poor discrimination.

Histograms complement the calibration curves: they focus on the spread and separation of the scores for each class, rather than alignment of scores with true frequencies.

10. **Confusion Matrix:** Finally, the confusion matrix provides the raw counts of outcomes: True Negatives (TN), False Positives (FP), False Negatives (FN), and True Positives (TP). The confusion matrix is useful to directly quantify the types of errors: FP (healthy misdiagnosed as AD) vs FN (AD missed). In our context, and FN (miss) is more seriously clinically than a FP, so we pay close attention to reducing FN counts (via higher recall), while also noting FP counts since too many false alarms could burden patients with unnecessary further tests.

We evaluated four classifiers – Logistic Regression, Random Forest, Support Vector Machine (SVM), and Multi-Layer Perceptron (MLP) – at three optimization stages: a default *non-optimized* model, a model with one key hyperparameter tuned, and a *fully optimized* model with extensive hyperparameter tuning. On further section 4.1 we summarize how each model performed on the training set (TR) and test set (TS), and how optimization affected their metrics. We also relate these outcomes to our earlier data understanding (class imbalance, feature correlations, etc.) to interpret the results.

3.3 IMAGE DATA MODELLING

The methodology involved implementing convolutional neural networks (CNNs) to classify four types of MRI images related to Alzheimer's disease: *MildDemented*, *ModerateDemented*, *NonDemented*, and *VeryMildDemented*. The workflow progressed from baseline models to partially and fully optimized networks. Model interpretability was addressed using SHAP and Grad-CAM++ to analyze attention at different layers and generate heatmaps. All experiments were conducted on Google Colab Pro utilizing CPU acceleration. An additional analysis using transfer learning with ResNet was performed, but its representations were too generic, leading to the choice of a custom CNN for finer detail capture.

3.3.1 IMAGE PREPROCESSING AND RATIONALE BEHIND CHOSEN INITIAL VALUES

The dataset, sourced from a publicly available Kaggle[4] repository, was organized into subdirectories by class and processed using PyTorch's *ImageFolder* utility. To standardize input dimensions and facilitate compatibility with pre-trained networks and common CNN architectures, all images were resized to 224×224 pixels and normalized using the standard ImageNet mean and standard deviation. The dataset was compressed in ZIP format to minimize disk usage and reduce I/O bottlenecks during loading; it was programmatically extracted at runtime to optimize data handling and speed up the preprocessing phase, particularly within the Google Colab Pro environment.

For model training and evaluation, the data was partitioned using stratified sampling to ensure a balanced distribution across the four classes (*MildDemented, ModerateDemented, NonDemented, VeryMildDemented*). The split was set to 72.25% for training, 12.75% for validation, and 15% for testing. This configuration provides sufficient training samples while retaining representative validation and test sets for generalization assessment. The validation fraction (15% of total) was derived from an internal split of the training+validation pool (85% of total data), calculated as 15/85 ≈ 0.1765.

A batch size of 32 was chosen as a compromise between training stability and memory efficiency, particularly suited to GPU usage in Colab Pro. This size allowed efficient mini-batch gradient descent while avoiding memory overflows during training and interpretability computations. All subsets were wrapped using PyTorch *DataLoader* objects to enable shuffling, parallel data loading, and batch-wise iteration throughout the pipeline.

3.3.2 IMPLEMENTATION OF CONVOLUTIONAL NEURAL NETWORKS (CNNS) WITH INITIAL DEFAULT SETTINGS

A custom CNN model was defined, comprising three convolutional blocks with batch

[4] https://www.kaggle.com/datasets/marcopinamonti/alzheimer-mri-4-classes-dataset

normalization, ReLU activation, and max pooling. The network architecture culminates in a fully connected dense layer that maps the feature map into the four target classes. The second convolutional block was specifically tagged for Grad-CAM++ activation mapping. The network is trained using the cross-entropy loss function, appropriate for multi-class classification, and runs on a CUDA-enabled GPU when available.

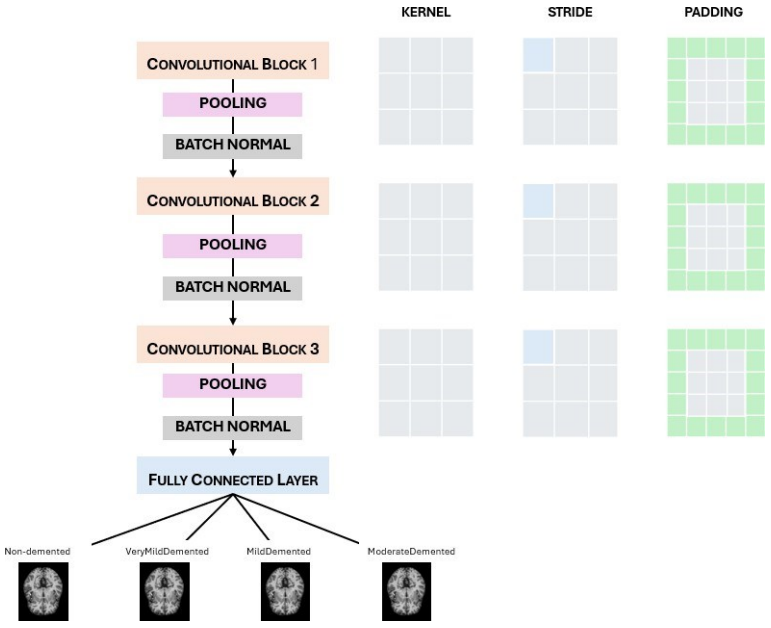


Figure 30:CNN + FCN Architecture

3.3.3 EVALUATION METRICS

Model performance was evaluated on both the training and test sets using accuracy and ROC-AUC curves per class. The ROC curve provides a detailed view of the classifier's performance across varying thresholds and was computed by one-vs-rest binarization of the class labels using *sklearn.metrics.roc_curve*. Also, accuracy metrics were computed.

3.3.4 APPLICATION OF INTERPRETABILITY TECHNIQUES: SHAP AND GRAD-CAM++ FOR GENERATING HEAT MAPS

To enhance transparency and understandability of the model's decision-making, two post-hoc

interpretability techniques were employed:

1. **SHAP (SHapley Additive exPlanations):**

SHAP values were generated for each image and superimposed as heatmaps to highlight pixel regions most responsible for the model's decision. SHAP was computed on individual class predictions using the Deep SHAP backend, enabling pixel-level attributions.

2. **Grad-CAM++:**

Gradient-based Class Activation Mapping was applied to visualize the spatial focus of the CNN at the second convolutional block. The GradCAM++ implementation from pytorch-grad-cam was used. Two versions of the heatmap were generated:

- **Unmasked:** Applied directly on the raw image, capturing full gradient flows.
- **Masked:** A binary mask was applied to exclude background pixels, improving signal clarity by filtering non-informative dark areas.

3.3.5 Experiments with no optimization, partial optimization (learning rate) and full CNN optimization (Learning Rate and Weight Decay)

The training process was designed in three phases:

1. **Baseline (Non-optimized):** The model was trained with default hyperparameters using the Adam optimizer and a fixed learning rate of 1e-3.
2. **Learning Rate Optimization:** The learning rate was tuned through a grid search over [1e-2, 1e-3, 1e-4, 1e-5] using Adam optimizer as well. The best result was selected based on validation accuracy.
3. **Full Optimization:** A Bayesian optimization approach was employed using the *Optuna* library to jointly optimize the learning rate, weight decay, and choice of optimizer (Adam or SGD).

This allowed exploration of the hyperparameter space in a more data-efficient manner.

Each model was trained for 10 epochs to ensure consistency in evaluation and to prevent

overfitting on limited training samples.

3.4 DEVELOPMENT OF THE UNIFIED METRIC

3.4.1 IDENTIFICATION OF KEY INTERPRETABILITY INDICATORS

Interpretability in machine learning cannot be reduced to a single property. Different explanation methods emphasize different aspects, and their utility varies depending on the model and the data. To establish a systematic and fair comparison between methods such as SHAP and LIME, we identified three core indicators frequently cited in the interpretability literature (Doshi-Velez & Kim, 2017; Molnar, 2022):

1. **Fidelity**: the extent to which an explanation accurately represents the model's predictions. In LIME, this corresponds to the quality of the local surrogate fit, whereas in SHAP it reflects the additive reconstruction of the model's output from feature contributions (Ribeiro et al., 2016; Lundberg & Lee, 2017).

2. **Stability**: the robustness of explanations when small perturbations are introduced to the input data. Stable explanations are particularly important in clinical contexts, where reliability under slight variations is crucial (Alvarez-Melis & Jaakkola, 2018).

3. **Sparsity**: the conciseness of explanations, often defined by the number of non-zero feature attributions. Sparse explanations are easier to interpret but may risk omitting relevant signals if excessive.

These three indicators were selected because they consistently appear in existing work, align with human-centered evaluation criteria, and can be meaningfully applied to both tabular and image data.

3.4.2 PROPOSAL OF THE COMPOSITE METRIC

To integrate these indicators into a single evaluation, we propose a composite interpretability score at the method level:

$$S_{method} = \alpha_1 \cdot Fidelity + \alpha_2 \cdot Stability - \alpha_3 \cdot Sparsity$$

where $\alpha_1, \alpha_2, \alpha_3$ are non-negative weights reflecting the relative importance of each dimension. For biomedical applications, fidelity and stability were prioritized ($\alpha_1 = 0.5, \alpha_2 = 0.3$) to ensure reliable and clinically trustworthy explanations, while sparsity was retained as a penalty ($\alpha_3 = 0.2$) to discourage over simplistic explanations. These parameters were established using heuristic methods. Alternative weighting schemes could be established via domain-expert elicitation (e.g., clinicians prioritizing stability) or via data-driven optimization, but heuristics provide a transparent and reproducible baseline.

All sub-metrics were normalized to the [0,1] range prior to aggregation. This normalization was achieved by rescaling observed values against fixed interpretability bounds derived from literature benchmarks (e.g., maximum surrogate fit quality for LIME, maximum additive reconstruction accuracy for SHAP). This ensures comparability across both methods and models.

In addition to the three indicators, we introduce an **agreement factor** (AAA) that captures the overlap between SHAP and LIME explanations. The rationale is that if two independent methods converge on similar feature importance, confidence in the reliability of the explanation increases. Agreement is defined mathematically as:

$$A = \frac{|F_{SHAP} \cap F_{LIME}|}{|F_{SHAP} \cup F_{LIME}|}$$

where F_{SHAP}, F_{LIME} denote the sets of top-ranked features identified by each method. This factor ranges from 0 (no overlap) to 1 (perfect agreement).

The unified model-level score is therefore expressed as:

$$U_{model} = \frac{1}{2}(S_{SHAP} + S_{LIME}) + \beta \cdot A$$

where β is an enhancement weight controlling the influence of agreement. In this study, we set $\beta = 0.2$, enabling agreement to act as a "nudge" without dominating the metric.

3.4.3 Application Framework

The proposed framework is applied in two stages:

1. **Method-level evaluation**: SHAP and LIME are each assessed individually according to fidelity, stability, and sparsity, resulting in a per-method interpretability score (S_{method}).

2. **Model-level aggregation**: Scores from both methods are averaged and adjusted with the agreement factor, yielding a unified interpretability score (U_{model}) for each predictive model.

This stepwise procedure enables both direct comparison of explanation tools and standardized evaluation of overall interpretability across models such as Logistic Regression, Random Forest, Support Vector Machine, and Multi-Layer Perceptron. Moreover, while this framework is applied here to Alzheimer's disease diagnostics, it is generalizable to other biomedical and non-biomedical settings, offering a reproducible and transparent way to evaluate interpretability in machine learning.

Finally, this proposal directly addresses calls in the literature for unified and standardized evaluation of interpretability tools (Guidotti et al., 2018; Samek et al., 2021). By combining fidelity, stability, sparsity, and agreement, the metric balances predictive alignment with human interpretability, thus advancing explainable AI research in a practical and domain-aware direction.

Chapter 4. Results

While performance metrics like accuracy or F1-score offer a first impression of how well a model behaves, they do not explain why the model makes certain decisions. To move from performance to trust, it is essential to open the black box and understand the reasoning behind individual predictions. Interpretability techniques such as LIME and SHAP provide this transparency by offering local explanations that help users:

i. **Choose between competing models** not just based on performance scores, but also on how consistently and reasonably each model behaves across features and instances.
ii. **Detect and improve untrustworthy models** by revealing patterns of bias, spurious correlations, or over-reliance on irrelevant features.
iii. **Gain deeper insight into the model's behavior**, revealing which features are most influential and how they contribute to predictions in specific cases.

Ultimately, LIME and SHAP answer a critical question that performance metrics alone cannot:

Why should we trust this model?

These explanations form a bridge between statistical performance and actionable, transparent decision-making — particularly vital in domains like healthcare, where understanding the why behind a prediction is as important as the prediction itself.

4.1 Results for Tabular Data Modelling

In this section, we will be discussing the results obtained regarding tabular data modelling. The database used is a version of the 2025 Alzheimer's disease data available on Kaggle[5]. The dataset is related to Alzheimer's disease, and it is presented in a tabular format (CSV file).

The following models (LR, RF, MLP, and SVM) have been analyzed and treated as black box models. Further information can be found in the previous chapters (see 3.2.1 and 3.3.1). We

[5] Kaggle: Online community platform for data scientists and ML enthusiasts.

present three different results for the three types of models studied for each machine learning method (Default, Single Parameter Optimized, and Fully Optimized Models). Additionally, we examine the nature of predictive performance. Finally, we summarize the variable relevance as determined by SHAP and LIME. The goal of these results is to understand:

1. **Model performance:** mainly model accuracy.
2. **Model interpretability:** why the different machine learning models made certain predictions at a local and global level.

4.1.1 BASELINE: LOGISTIC REGRESSION MODEL

4.1.1.1 PERFORMANCE RESULTS

We analyzed the performance and interpretability of Default Logistic Regression, single parameter optimized, and fully optimized models. The evaluation (for both training and testing) was conducted using:

i. Confusion matrices
ii. ROC curves
iii. Predicted probability of histograms.

i. CONFUSION MATRICES

The confusion matrices reveal how classification performance evolves across the Default, Single-Parameter Optimized, and Fully Optimized Logistic Regression models.

Default Model

In its default configuration, the Logistic Regression model performed well, with a Type I error of 14% and a Type II error of 26.3%, indicating balanced but slightly more frequent false negatives.

On the **training set**, the model predicted 1,010 true negatives and 453 true positives, while misclassifying 101 cases as false positives (Type I error) and 155 cases as false negatives (Type II error). On the test set, the model correctly classified 239 true negatives and 112 true positives, with 39 false positives and 40 false positives (See *Figure 7*).

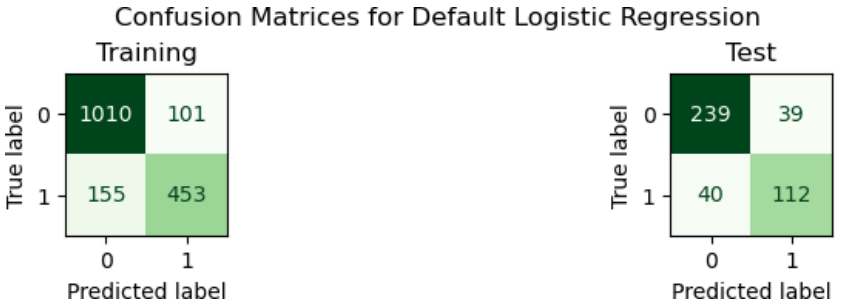


Figure 31: Default LR

It is evident that the model maintains relatively balanced classification behavior. However, the number of false negatives (40 on the test set) is particularly important in the context of Alzheimer's disease diagnosis, as failing to detect a true positive patient could delay crucial intervention.

One hyperparameter optimization

Optimizing only the regularization parameter C resulted in no change to the confusion matrix compared to the default model.

This result shows that tuning a single hyperparameter had little impact, highlighting the need for full hyperparameter optimization to improve performance.

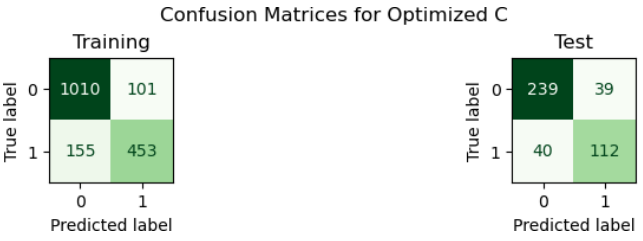


Figure 32: 1 optimized hyperparameter

Full hyperparameter optimization

When applying **full hyperparameter**—adjusting C, the penalty type, and the solver—the Logistic

Regression model exhibited a slight but meaningful improvement.

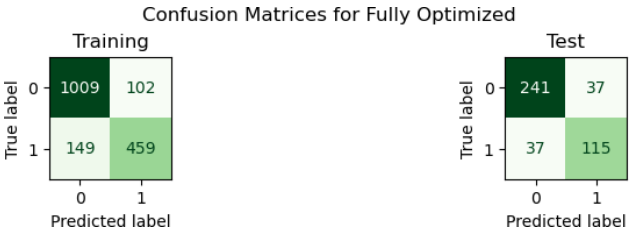


Figure 33: All hyperparameters optimized

On the training set, there were 1,009 true negatives and 459 true positives, alongside 102 false positives and 149 false negatives. On the test set, the model correctly classified 241 true negatives and 115 true positives, with only 37 both false negatives and false positives.

These improvements are clinically significant because, regarding the test sets:

1. The number of false negatives decreased from 40 to 37, thereby enhancing the model's sensitivity (recall) and reducing the likelihood of missing Alzheimer's disease cases.
2. The number of false positives also decreased from 39 to 37, slightly improving the model's specificity and minimizing unnecessary clinical interventions.

Overall, the fully optimized model achieves a better balance between Type I and Type II errors, enhancing its reliability as a diagnostic tool—an aspect particularly critical in medical applications where both precision and sensitivity are essential.

Conclusion

In terms of computational efficiency, the Default Logistic Regression model required less than one second to train, whereas the fully optimized model increased training time to approximately six seconds on a personal laptop[6].

Considering the trade-off between computational cost and predictive performance, the modest

[6] See Annex III too see computer specifics.

increase in computational time is not justified by the improvements in classification accuracy and diagnostic reliability.

ii. ROC CURVES

ROC curves assess a model's ability to separate classes, while the AUC summarizes this performance—ranging from 1.0 (perfect) to 0.5 (random).

Default Model

The ROC curves for the Default Logistic Regression model (See Annex IV) demonstrate strong classification performance as shown in the following Table 2:

Training AUC	0.91
Test AUC	0.89

Table 2: AUC Default Model

The curves show that the model maintains a high true positive rate while minimizing false positives across a range of thresholds. The small drop from training to testing AUC suggests limited overfitting and good generalization to unseen data. However, the slight gap indicates some room for improvement in model robustness, particularly critical in a medical setting where misclassifications can have severe consequences.

One hyperparameter optimization

After optimizing the regularization parameter C, the ROC curves (See Annex IV) remain practically identical to those of the Default model:

Training AUC	0.91
Test AUC	0.89

Table 3: AUC One Hyperparameter optimization

This outcome confirms that adjusting a single hyperparameter was insufficient to produce a meaningful improvement in discriminative performance.

Fully Optimized Logistic Regression

For the Fully Optimized Logistic Regression model (See Annex IV), which involved tuning multiple hyperparameters (C, penalty, and solver), the ROC curves are similarly well-behaved:

Training AUC	0.91
Test AUC	0.89

Table 4: AUC Full Hyperparameter Optimization

Although the AUC values remain comparable to the previous models, the ROC curve for the test set shows a slightly steeper ascent near the origin, implying better early detection of positive cases with low false positive rates. This subtle improvement is critical for clinical settings, where it is particularly important to correctly identify Alzheimer's patients while minimizing unnecessary alarms for healthy individuals.

iii. CALIBRATION CURVE PREDICTED

The calibration curves for all Logistic Regression models demonstrate good reliability, with predicted probabilities closely following the ideal calibration line. Minor deviations were observed at the extremes; however, full hyperparameter optimization slightly improved the calibration on the test set, thereby enhancing the model's trustworthiness for clinical decision-making. For a detailed visualization of the calibration curves, the reader is referred to Annex IV.

iv. PROBABILITY HISTOGRAM ANALYSIS

Similarly, the predicted probability histograms indicate that all models produced confident predictions, with the Fully Optimized model exhibiting sharper separations between classes. This increased certainty in classification, combined with improved calibration, suggests that the Fully Optimized Logistic Regression model offers the most reliable configuration for supporting Alzheimer's disease diagnosis. The corresponding histograms can be consulted in Annex IV.

v. PERFORMANCE METRICS

The predictive performance of the Default Logistic Regression, Single-Parameter Optimized (C optimization), and Fully Optimized Logistic Regression models were evaluated using key classification metrics: Type I error, Type II error, Specificity, Accuracy, Recall, F1-score, F2-score,

and Cohen's Kappa coefficient. These metrics provide a comprehensive assessment of both the discriminative ability and reliability of each model configuration.

Default and Single – Parameter Optimized Logistic Regression

Both the Default and Single-Parameter Optimized model showed identical performance across all training and testing metrics.

I. On the training set, an accuracy of 85.1% was achieved, with a recall (sensitivity) of 74.5% and a specificity of 69.0%. The Type I and Type II errors were 9.1% and 25.5%, respectively. The F1-score and F2-score were 0.780 and 0.802, demonstrating a balanced trade-off between precision and recall. Cohen's Kappa coefficient reached 0.668, indicating substantial agreement beyond chance.

II. On the test set, the accuracy slightly decreased to 81.6%, with a recall of 73.7% and a specificity of 68.1%. Type I and II errors slightly increased, and the Kappa value dropped to 0.597, showing a moderate agreement.

The minimal difference between the default and C-optimized models shows that tuning only the regularization wasn't enough, indicating the need for broader hyperparameter optimization.

Fully Optimized Logistic Regression

The Fully Optimized Logistic Regression model demonstrated consistent but modest improvements across nearly all performance indicators:

I. On the training set, the accuracy increased to 85.4%, with a recall of 75.5% and specificity of 68.7%. Type II error decreased to 24.5%, and both the F1-score (0.785) and F2-score (0.805) improved, indicating enhanced balance between sensitivity and precision. The Kappa value also increased to 0.675, suggesting stronger agreement compared to the previous models.

II. On the test set, the model achieved an accuracy of 82.8%, a recall of 75.7%, and a specificity

of 67.7%. Both Type I and Type II errors decreased relative to the previous models, and the F1-score and F2-score improved to 0.757. The Kappa statistics rose to 0.623, reflecting better predictive consistency on unseen and unbalanced data.

Performance test metrics are summarized on the following table:

Metric	Default Logistic Regression	Optimized C	Fully Optimized
Type I Error	0.140	0.140	0.133
Type II Error	0.263	0.263	0.243
Specificity	0.6809	0.6809	0.677
Accuracy	0.816	0.816	0.828
Recall	0.737	0.737	0.757
F1-score	0.739	0.739	0.757
F2-score	0.741	0.741	0.757
Kappa	0.597	0.597	0.623

Table 5: Performance metric LR summary

Conclusion

The Fully Optimized model consistently outperformed the Default and Single-Parameter Optimized models, especially in recall and accuracy. In Alzheimer's disease detection, the increase in recall (from 73.7% to 75.7%) is crucial for correctly identifying patients. Although modest, these improvements are clinically significant when escalating it, making the Fully Optimized Logistic Regression model the best configuration for balancing performance and interpretability (despite having higher computational cost).

In the following chapters, we argue that while the model achieves excellent accuracy, this metric alone is not sufficient to ensure trust or reliability, especially in sensitive domains like Alzheimer's diagnosis. It is equally important to understand how the model arrives at individual predictions, as this sheds light on its overall decision-making process. To address this, we will analyze LIME and SHAP explanations applied to the Alzheimer's tabular dataset.

4.1.1.2 INTERPRETABILITY RESULTS

A randomly selected patient instance was interpreted using both LIME and SHAP, while global explanations were derived using SHAP across all instances. Although a similar global approximation could be achieved using SP-LIME (which applies submodular analysis[7]), this method was considered outside the scope of the current project.

The patient, a 65-year-old individual, was predicted by the Logistic Regression model (non-optimized version) to have Alzheimer's disease with a probability of 76%. This prediction is derived from the original black-box model, not from LIME. LIME is instead used to interpret how this prediction was made, by attributing contributions to individual features based on a locally fitted surrogate model.

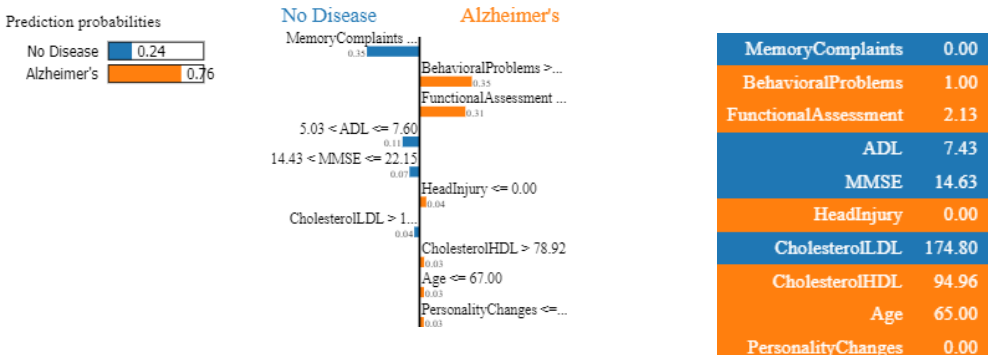


Figure 34: LIME explanations for a non-optimized LR model

Figure 10 (center) displays a bar chart showing how each feature influences the prediction:

- Orange bars indicate features pushing the prediction toward the Alzheimer's class.
- Blue bars indicate features supporting a No Disease classification.

For this instance, the most influential features were:

- *BehaviouralProblems* = 1.0 and *FunctionalAssessment* = 2.13: they are the largest positive contributions towards Alzheimer's
- *MemoryComplaints* = 0.00, *ADL* = 7.43, and *MMSE* = 14.63 depict moderate to strong

[7] Class of mathematical optimization problem that deals with submodular functions, a type of set functions with a natural diminishing returns property that is, the incremental gain of adding an element to a set, decreases as the set grows.

contributions against the Alzheimer's classification.

While most attributions align with clinical reasoning, features like HDL and LDL cholesterol levels introduced interpretability challenges. HDL ("good" cholesterol) is protective; values >60 mg/dL are linked to lower Alzheimer's risk. LDL ("bad" cholesterol) is a risk factor; values <100mg/dL are considered optimal for reducing Alzheimer's likelihood. In this case, HDL supported "Alzheimer's", while LDL supported "No Disease". Although this might appear contradictory, it highlights the complexity of biological systems and the statistical (not causal) nature of ML predictions. These outputs should always be interpreted by medical professionals within clinical context.

How LIME trains the surrogate model

The weights show how much each feature shifts the prediction toward Alzheimer's or No Disease. For example, BehavioralProblems = 1.00 and FunctionalAssessment = 2.13 contributed +0.35 and +0.31 respectively toward the 76% Alzheimer's prediction. Removing them would lower it to around 10%. However, these are not actual probabilities—they come from a local surrogate model trained to approximate the original model for interpretability.

To further clarify how LIME produces explanations, we present a simplified example based on the same patient. We assume only two input features:

Feature	Value	Description
MemoryComplaints	0.0	0 indicates No and 1 indicates Yes.
Functional Assessment	2.0	Score, ranging from 0 to 10.

Table 6: Dummy LIME case

1. Perturbing the Original Instance

LIME generates synthetic samples by introducing small perturbations to the input, from which we get the black-box model predictions.

Synthetic Sample	Memory Complaints	Functional Assessment
A	0.1	2.1
B	1.0	4.5
C	0.2	1.8

Table 7: Perturbed generated instances

For tabular data instances, LIME randomly creates synthetic data around the original *x* instance, by using a normal distribution, centered at the original data point, and standard deviation determined by the black box trained data.

2. Computing the Euclidean Distances

To compute the weights assigned to each perturbed instance, first we need to compute the distance from the perturbed datapoints to the original instance. Since we are working with tabular data, Euclidean distance will be the one selected by LIME.

$$D(A) = \sqrt{(0.1-0)^2-(2.1-2.0)^2} = \sqrt{0.01+0.01} = \sqrt{0.02} \approx 0.141$$

$$D(B) = \sqrt{(1.0-0)^2-(4.5-2.0)^2} = \sqrt{1.0+6.25} = \sqrt{7.25} \approx 2.692$$

$$D(A) = \sqrt{(0.2-0)^2-(1.8-2.0)^2} = \sqrt{0.04+0.04} = \sqrt{0.08} \approx 0.283$$

3. Computing Kernel Weights

To determine the importance of the perturbed instances (weights) LIME algorithm applies an exponential kernel π_x to translate distances into weights.

$$\pi_x(z) = \exp\left(-\frac{D(x,z)^2}{\sigma^2}\right)$$

Where:

- *D:* Distance metric (e.g., Euclidean)
- *x:* Original instance: 65-year-old patient
- *z:* Pertubed instances: synthetic data
- σ: Exponential kernel width: Hyperparameter that controls how much influence nearby samples have; lower values focus the explanation locally, while higher values result in more general, broader explanations (e.g., 0.2 standard deviations).

Assuming a standard kernel width σ = 0.75 (default value):

- $w_A = exp\left(-\frac{(0.141)^2}{(0.75)^2}\right) \approx \exp(-0.035) \approx 0.965$
- $w_B = exp\left(-\frac{(2.692)^2}{(0.75)^2}\right) \approx \exp(-12.84) \approx 0.0$
- $w_C = exp\left(-\frac{(0.283)^2}{(0.75)^2}\right) \approx \exp(-0.142) \approx 0.868$

This shows that Sample A, being very close to the original, receives the highest weight (close to 1), while Sample B, being far away, is effectively ignored.

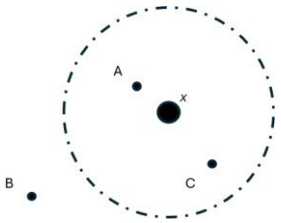


Figure 35: Local Neighborhood and Kernel Weighting in LIME.

The figure illustrates the locality principle in LIME. The central point x represents the original instance to be explained, while points A, B, and C are perturbed samples. The dashed circle depicts the region of influence defined by the kernel width parameter σ, set to 0.75 in this case. Instances within this radius (like A and C) receive higher weights due to their proximity to x, while distant instances like B are assigned negligible weights. This ensures that the surrogate model focuses on locally relevant patterns, maintaining interpretability and fidelity near the point of interest.

4. Fitting the Local Surrogate Model

These weighted perturbed samples are used to train a simple linear surrogate model g(x), which minimizes a loss function that balances local fidelity (faithfulness to the original model) with interpretability (model simplicity):

$$\xi(x) = \underset{g \in G}{\operatorname{argmin}} \sum w(x') \cdot (f(x') - g(x'))^2 + \Omega(g)$$

Where:

- $f(x')$: prediction from the original black box model
- $g(x')$: prediction from the surrogate (simple) model.
- $\Omega(g)$: regularization term penalizing complexity.
- $w(x')$: kernel weights based on proximity

This ensures that LIME approximates the original model locally, giving priority to nearby perturbations and producing an explanation faithful to the neighborhood of the instance.

5. Discussion

This case illustrates how LIME's interpretability is derived not from the model's internals, but from a local surrogate trained around the instance of interest. While features like MMSE and *BehavioralProblems* provide intuitive explanations consistent with clinical literature, others (such as LDL and HDL) may introduce ambiguity. These nuances emphasize that ML explanations should not be interpreted in isolation, and clinical judgment remains essential.

Having established how LIME algorithm works and having examined the interpretability explanations of the non-optimized Logistic Regression model, we now turn to the analysis of the same patient instance under a model where the regularization parameter C has been optimized. This allows us to assess whether adjusting a single hyperparameter has a meaningful impact on both prediction confidence and feature attribution.

In the optimized model, the predicted probability for Alzheimer's disease remains unchanged at 76 %, suggesting that tuning C alone did not alter the model's confidence for this specific case. The LIME explanation, however, shows minor shifts in feature attributions. Notably, *CardiovascularDisease* now appears among the contributors, while other key features such as *MemoryComplaints*, *BehavioralProblems* and *FunctionalAssessment* maintain similar influence levels. These subtle changes imply that while performance metrics may remain stable, interpretability outcomes can vary with even modest model adjustments. This reinforces the importance of analyzing how hyperparameter tuning affects both prediction and explanation fidelity.

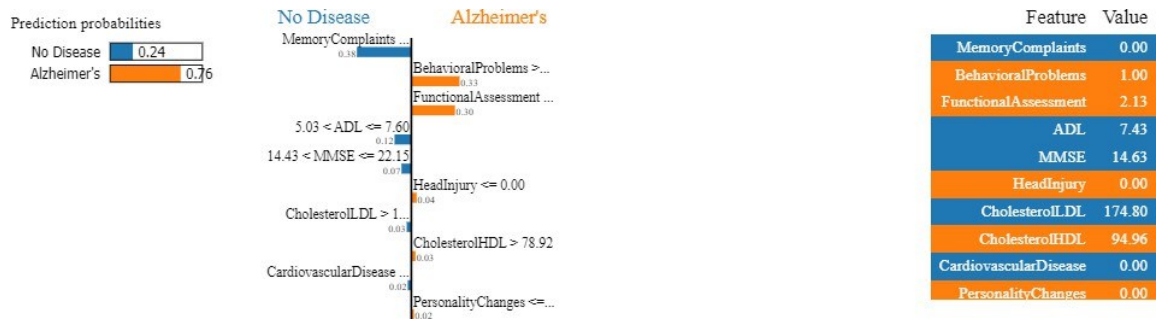


Figure 36: LIME explanations for an optimized LR model

In the fully optimized Logistic Regression model, LIME shows a prediction probability of 69% for Alzheimer's slightly lower than the 76% from the default configuration. The most influential features remain *BehavioralProblems* (1.00) and *FunctionalAssessment* (2.13), both pushing toward the Alzheimer's class. Meanwhile, *MemoryComplaints* (0.00), ADL and MMSE continue to support the "No Disease" classification.

Figure 37: LIME explanations for fully optimized model

Compared to earlier configurations, this explanation is slightly more balanced, with a clearer counteracting influence from protective features like ADL and *MemoryComplaints.* The inclusion of additional minor contributors like *DifficultyCompletingTasks* and *EducationLevel* reflects a more nuance prediction after full hyperparameter tuning. However, the overall structure of the explanation remains similar, confirming the model's consistency across optimization stages.

While LIME offers valuable local explanations through surrogate modeling, it is essential to complement this perspective with a global view feature of importance. To that end, we now turn to SHAP (SHapley Additive exPlanations), a model-agnostic method grounded in cooperative game theory. Unlike LIME, which focuses on locally approximating the model behavior, SHAP provides consistent and theoretically justified attributions for both individual predictions and overall feature impact. This section explores how SHAP interprets the same patient instance and the broader Alzheimer's dataset, offering a more holistic understanding of the model's decision-making process. We will prove, as discussed in previous sections, that SHAP is the most complete additive feature attribution method, as it always fits the three desirable properties (local accuracy, missingness and consistency). LIME explanation violated the consistency property because HDL and LDL showed conflicting contributions, despite the model's overall structure not changing. This suggests that LIME's local surrogate assigned attributions that do not align with the global behavior of the original model.

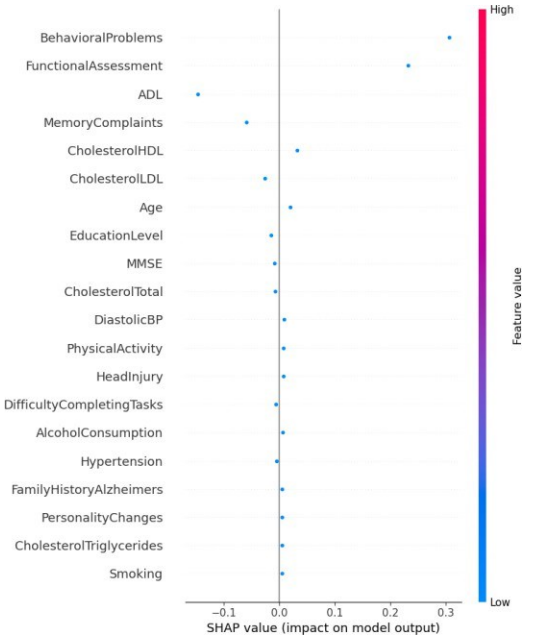


Figure 38: SHAP feature importance overview for a single instance

Figure 14 presents a SHAP scatter plot that visualizes the SHAP values for a single patient instance from the validation dataset. The x-axis shows each feature's SHAP value—indicating its contribution toward pushing the prediction toward "Alzheimer's" (positive) or "No Disease" (negative)—while the y-axis lists features in descending order of importance. The color gradient represents the magnitude of each feature's value (red for high, blue for low). In this case, all markers appear in blue, meaning the instance holds relatively low values across features. Notably, some contributions (e.g., low ADL pushing toward "No Disease") contradict expected clinical behavior, raising interpretability concerns. Nevertheless, it is important to note that this plot reflects only one instance. A global SHAP analysis will follow to assess the full validation dataset. Meanwhile, this specific case will serve as a benchmark example to illustrate how SHAP values are computed.

The SHAP analysis highlights that *BehavioralProblems* and *FunctionalAssessment* are the most influential features driving predictions toward Alzheimer's. Moderately relevant features like ADL and *MemoryComplaints* also play a role, usually supporting "No Disease" when favorable. In contrast, features such as *Smoking* and *CholesterolTriglycerides* have negligible SHAP values, indicating little to no impact on the model's decision in this case.

Based on this instance, suppose the model predicts a probability of 0.72 for Alzheimer's disease.

The base value (i.e. *E[f(x)]*, the average prediction across the training data) is 0.50. We'll explain how we get from 0.50 → 0.76 using SHAP.

The base value is the prediction the model would make if it had no information about the input instance (i.e., before seeing any features).

Let's use 3 key features from previous plot:

Feature	Patient Value	SHAP Value
BehavioralProblemss	1.0	+0.30
Functional Assessment	2.13 (Low Score)	+0.22
ADL	7.43 (Higher)	-0.15

Each SHAP value φ_i is computed using Shapley values, from cooperative game theory using:

$$\varphi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!\,(|F| - |S| - 1)!}{|F|!} [f(S \cup \{i\}) - f(S)]$$

Where:

- F: set of all features
- S: a subset of features not including i.
- *f(S):* model prediction when only features in S are known
- $f(S \cup \{i\})$: the model prediction when feature *i* is added to subset *S.*

The expression weights each subset fairly, according to the Shapley formula. By "fair," we mean that each feature is assigned a value based on its marginal contribution to the model's prediction across all possible subsets of features.

Considering this example, we'll compute the SHAP value for *BehavioralProblems*. First, we define, the set and subsets.

$$F \rightarrow \{BehavioralProblems, FunctionalAssessment, ADL\} \quad \text{So } |F| = 3.$$

$$S_0 \rightarrow \{\emptyset\} \text{ So } |S_0| = 0$$

$$S_1 \rightarrow \{FunctionalAssessment\} \text{ So } |S_1| = 1$$

$$S_2 \rightarrow \{ADL\} \text{ So } |S_2| = 1$$

$$S_3 \rightarrow \{Functional\ Assessment, ADL\} \ |S_3| = 1$$

We can verify that all possible subsets have been considered by applying combinatorics: since SHAP uses all feature combinations (including the empty set), the total is $2^{|F|-1}$ for each feature:

$$n\ possibilities = 2^{|F|-1} = \frac{|F|!}{(|F|-|S|)!\cdot |S|!} + 1 = \{|F| = 3, |S| = 2\} = \frac{3!}{1!\cdot 2!} + 1 = 3 + 1 =$$

4 possibilities

For |F| = 3 the weights for the different subsets so that $S \subseteq \{FA.ADL\}$ are:

$$S_0 = \emptyset: \frac{0! \cdot 2!}{3!} = \frac{1 \cdot 2}{6} = \frac{1}{3}$$

$$S_1 = \{FA\}: \frac{1! \cdot 1!}{3!} = \frac{1 \cdot 1}{6} = \frac{1}{6} = S2 = \{ADL\}$$

$$S_3 = \{FA, ADL\}: \frac{2! \cdot 0!}{3!} = \frac{2 \cdot 1}{6} = \frac{1}{3}$$

By recomputing SHAP contributions for small subsets with KernelExplainer, following approximations can be obtained, by looking inside explainer's internals (although they are not officially exposed). By denoting *f(S)* as the output of the model prediction when only features in subset *S* are present.

We'll obtain the predictions for the following subsets of pairs

Subset S	f(S)	f (S ∪ {*Behavioral Problem)*	Marginal Contribution
∅	0.50	0.8080	0.3030
{FA}FA	0.55	0.87	0.32
{ADL}	0.45	0.76	0.31
{FA, ADL}	0.60	0.86	0.26

Table 8: SHAP marginal contributions

Therefore:

$$\phi_{BehavioralProblems} = \frac{1}{3} \cdot (0.30) + \frac{1}{6} \cdot (0.32) + \frac{1}{6} \cdot (0.31) + \frac{1}{3} \cdot (0.26)$$
$$= 0.10 + 0.0533 + 0.0516 + 0.0867 = 0.2916 \approx 0.30$$

This means *BehavioralProblems = 1.0* adds +0.30 to the final prediction compared to the base value. (We simplified the math and assumed some values, but the model's outputs could yield ~+0.30).

While analyzing a single instance helps illustrate how SHAP values explain individual predictions, it is not sufficient to evaluate the model's global behavior. By analyzing the SHAP summary plot across all validation instances, we can observe overall feature importance trends and validate the key SHAP properties: **local accuracy** (the sum of SHAP values matches the prediction), **missingness** (features not present have no impact), and **consistency** (a feature's influence increases only if its contribution to the model increases). This ensures that explanations remain reliable and coherent across different patients.

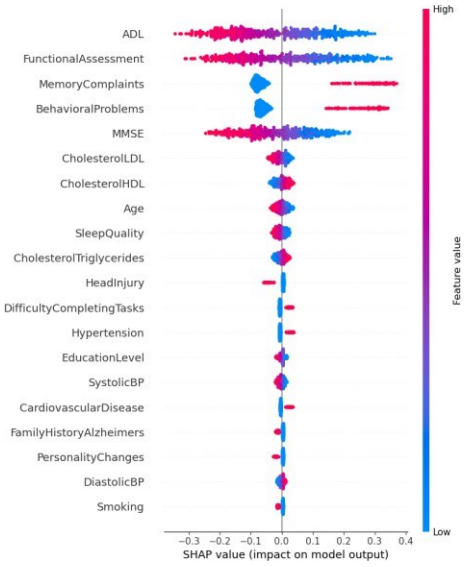


Figure 39: SHAP Summary Plot Feature Impact on Non-Optimized LR

Figure 15 displays a SHAP summary plot for the non-optimized Logistic Regression model. Each point represents a SHAP value for a feature in a single prediction within the validation dataset. Features are ranked from top to bottom by their overall importance (mean absolute SHAP value). The x-axis shows how much each feature contributes to increasing or decreasing the model's output toward Alzheimer's disease (positive SHAP values) or No Disease (negative values).

The color represents the feature value for that instance: red indicates high feature values, and blue indicates low values. For example, high values of *FunctionalAssessment* and *BehavioralProblems* (in red) push predictions toward Alzheimer's (positive SHAP), whereas high **ADL** and **MMSE** values (in red) push predictions toward No Disease (negative SHAP). This behavior aligns with clinical expectations, where greater cognitive and functional ability (e.g., higher MMSE and ADL scores) is typically associated with healthier outcomes.

Overall, this plot provides global insight into how the model is using features across all patients and confirms that key variables are behaving consistently with medical knowledge.

To evaluate the impact of model optimization on interpretability, SHAP summary plots were generated for three versions of the classifier: the non-optimized model, a model optimized only with

respect to the regularization parameter C, and a model optimized across all hyperparameters.

Interestingly, the SHAP distributions remained remarkably consistent across the three configurations. The most influential features—such as *FunctionalAssessment*, *ADL*, *MemoryComplaints*, and *BehavioralProblems*—consistently appeared at the top of the importance ranking, while other variables such as *Smoking* or *DiastolicBP* contributed minimally throughout.

This limited variation suggests that hyperparameter tuning, in this case, did not substantially alter the underlying decision logic of the model. The data itself likely exhibits strong, stable patterns that dominate the model's learning process regardless of optimization. As a result, even though adjustments to regularization strength or other hyperparameters may slightly shift decision boundaries, they do not significantly affect how individual features contribute to predictions.

Moreover, optimizing solely on the regularization parameter C typically affects the model's ability to generalize, but not necessarily the relative importance of features, especially in linear models like logistic regression. The final model—tuned across all hyperparameters—also yielded SHAP plots nearly identical to the baseline, reinforcing the idea that the model architecture and the data jointly dictated the explanatory structure from the outset.

In summary, the stability of the SHAP value distributions across models provides evidence that the most relevant features were robust to changes in model configuration, thereby increasing confidence in their interpretability and reliability for downstream clinical or diagnostic insights.

4.1.2 COMPARATIVE ANALYSIS OF PREDICTIVE PERFORMANCE AND INTERPRETABILITY

To provide a clear and consistent overview, the performance and interpretability of the Random Forest (RF), Support Vector Machine (SVM), and Multi-Layer Perceptron (MLP) models are summarized below in comparison with the LR mentioned on section 4.1.1. The first two tables highlight their core performance metrics on both training and test sets. The third table distills interpretability insights derived from SHAP and LIME, focusing on how these tools help demystify each model's predictions.

SUMMARY OF MODEL PERFORMANCE ACROSS OPTIMIZATION LEVELS FOR TRAINING AND TEST SETS

TR Model	Optimization Level	Specificity	Accuracy	Recall	F1 Score	F2 Score	Kappa	AUC	Calibration Curve	Histogram	Confusion Matrix
Logistic Regression	Non-optimized	0.690	0.851	0.745	0.780	0.802	0.668	0.910	Well-calibrated	Well-separated	TN: 1010, FP: 101, FN: 155, TP: 453
	C-optimized	0.690	0.851	0.745	0.780	0.802	0.668	0.910	Well-calibrated	Well-separated	TN: 1010, FP: 101, FN: 155, TP: 453
	Full optimization	0.687	0.854	0.755	0.785	0.805	0.675	0.920	Well-calibrated	Well-separated	TN: 1009, FP: 102, FN: 149, TP: 459
Random Forest	Non-optimized	0.646	1.000	1.000	1.000	1.000	1.000	1.000	Overconfident	Well-separated	TN: 1011, FP: 0, FN: 0, TP: 608
	Min leaves tuned	0.653	0.990	0.972	0.985	0.993	0.977	1.000	Overconfident	Well-separated	TN: 1010, FP: 1, FN: 17, TP: 591
	Full optimization	0.646	1.000	1.000	1.000	1.000	1.000	1.000	Overconfident	Well-separated	TN: 1011, FP: 0, FN: 0, TP: 608
SVM	Non-optimized	1.000	0.646	0.000	0.000	0.000	0.000	0.82	Well-calibrated	Overlapping	TN: 1111, FP: 0, FN: 608, TP: 0
	C-optimized	0.864	0.738	0.285	0.435	0.636	0.321	0.82	Well-calibrated	Overlapping	TN: 1096, FP: 15, FN: 435, TP: 173
	Full optimization	0.684	0.853	0.763	0.786	0.800	0.674	0.91	Well-calibrated	Well-separated	TN: 1002, FP: 109, FN: 144, TP: 464
MLP	Non-optimized	0.708	0.856	0.707	0.777	0.826	0.672	0.92	Well-calibrated	Well-separated	TN: 1042, FP: 69, FN: 178, TP: 430
	Nº Hidden layer optimized	0.596	0.824	0.941	0.791	0.722	0.646	0.95	Well-calibrated	Well-separated	TN: 845, FP: 266, FN: 36, TP: 572
	Full optimization	0.666	0.904	0.854	0.863	0.868	0.789	0.96	Well-calibrated	Well-separated	TN: 1035, FP: 76, FN: 89, TP: 519

Table 9: Training metrics summary

TS Model	Optimization Level	Specificity	Accuracy	Recall	F1 Score	F2 Score	Kappa	AUC	Calibration Curve	Histogram	Confusion Matrix
Logistic Regression	Non-optimized	0.681	0.816	0.737	0.739	0.741	0.597	0.890	Moderately calibrated	Well-separated	TN: 239, FP: 39, FN: 40, TP: 112
	C-optimized	0.681	0.816	0.737	0.739	0.741	0.597	0.890	Well-calibrated	Well-separated	TN: 239, FP: 39, FN: 40, TP: 112
	Full optimization	0.677	0.828	0.757	0.757	0.757	0.623	0.890	Well-calibrated	Well-separated	TN: 241, FP: 37, FN: 37, TP: 115
Random Forest	Non-optimized	0.663	0.947	0.901	0.923	0.936	0.882	0.94	Overconfident	Well-separated	TN: 270, FP: 8, FN: 15, TP: 137
	Min leaves tuned	0.668	0.940	0.882	0.912	0.931	0.866	0.94	Overconfident	Well-separated	TN: 270, FP: 8, FN: 18, TP: 134
	Full optimization	0.667	0.942	0.888	0.915	0.932	0.871	0.94	Overconfident	Well-separated	TN: 270, FP: 8, FN: 17, TP: 135
SVM	Non-optimized	1.000	0.647	0.000	0.000	0.000	0.000	0.81	Well-calibrated	Overlapping	TN: 278, FP: 0, FN: 152, TP: 0
	C-optimized	0.912	0.691	0.171	0.281	0.458	0.177	0.81	Well-calibrated	Overlapping	TN: 271, FP: 7, FN: 126, TP: 26
	Full optimization	0.682	0.819	0.737	0.742	0.745	0.602	0.89	Well-calibrated	Well-separated	TN: 240, FP: 38, FN: 40, TP: 112
MLP	Non-optimized	0.699	0.812	0.691	0.722	0.742	0.580	0.87	Well-calibrated	Well-separated	TN: 244, FP: 34, FN: 47, TP: 105
	Nº Hidden layer optimized	0.589	0.730	0.849	0.690	0.620	0.466	0.85	Well-calibrated	Well-separated	TN: 185, FP: 93, FN: 23, TP: 129
	Full optimization	0.679	0.805	0.730	0.725	0.723	0.574	0.87	Well-calibrated	Well-separated	TN: 235, FP: 43, FN: 41, TP: 111

Table 10: Test metrics summary

LEGEND – MODEL PERFORMANCE SUMMARY TABLE

#	Description
	Best model
	2nd Best
	3rd Best
	Worst
	2nd Worst
	3rd Worst

SUMMARY OF MODEL INTERPRETABILITY ACROSS OPTIMIZATION LEVELS

Model	Optimization Level	SHAP – Top 5 Features		No AD	AD	LIME – Top 5 Features	No AD	AD
Logistic Regression	Non-optimized	ADL FunctionalAssessment MemoryComplaints BehavioralProblems MMSE	- - + + -	0.244	0.756	MemoryComplaints BehavioralProblems FunctionalAssessment ADL MMSE	0.24	0.76
	C-optimized	FunctionalAssessment ADL MemoryComplaints BehavioralProblems MMSE	- - + + -	0.244	0.756	MemoryComplaints BehavioralProblems FunctionalAssessment ADL MMSE	0.24	0.76
	Full optimization	FunctionalAssessment ADL MemoryComplaints BehavioralProblems MMSE	- - + + -	0.313	0.687	MemoryComplaints BehavioralProblems FunctionalAssessment ADL MMSE	0.31	0.69
Random Forest	Non-optimized	FunctionalAssessment ADL MemoryComplaints MMSE BehavioralProblems	- - + - +	0.426	0.574	MemoryComplaints BehavioralProblems FunctionalAssessment ADL MMSE	0.43	0.57
	Min leaves tuned	FunctionalAssessment ADL MemoryComplaints MMSE BehavioralProblems	- - + - +	0.435	0.565	MemoryComplaints BehavioralProblems FunctionalAssessment ADL MMSE	0.43	0.57
	Full optimization	FunctionalAssessment ADL MemoryComplaints MMSE BehavioralProblems	- - + - +	0.422	0.578	MemoryComplaints BehavioralProblems FunctionalAssessment ADL MMSE	0.42	0.58

SVM	Non-optimized	MMSE FunctionalAssessment ADL SystolicBP CholesterolTriglycerides	- - - - +	0.380	0.620	ADL MMSE CholesterolHDL FunctionalAssessment SystolicBP		0.43	0.57
	C-optimized	MMSE FunctionalAssessment ADL SystolicBP CholesterolTriglycerides	- - - - +	0.379	0.621	MMSE ADL FunctionalAssessment Age CholesterolHDL		0.96	0.04
	Full optimization	FunctionalAssessment ADL MemoryComplaints MMSE BehavioralProblems	- - + - +	0.303	0.697	MemoryComplaints ADL BehavioralProblems MMSE FunctionalAssessment		0.99	0.01
MLP	Non-optimized	FunctionalAssessment ADL MemoryComplaints BehavioralProblems MMSE	- - + + -	0.139	0.861	MemoryComplaints BehavioralProblems FunctionalAssessment ADL CholesterolHDL		0.14	0.86
	Nº Hidden layer optimized	FunctionalAssessment ADL MemoryComplaints MMSE BehavioralProblems	- - + - +	0.078	0.922	MemoryComplaints FunctionalAssessment BehavioralProblems ADL MMSE		0.08	0.92
	Full optimization	ADL FunctionalAssessment MemoryComplaints BehavioralProblems MMSE	- - + + -	0.160	0.84	MemoryComplaints FunctionalAssessment BehavioralProblems ADL CholesterolTotal		0.16	0.84

Table 11: Interpretability metrics summary

LEGEND – INTERPRETABILITY SUMMARY TABLE

#	Description
+	Positive SHAP Feature Contributions to Alzheimer's Disease (AD)
-	Negative SHAP Feature Contributions to Alzheimer's Disease (AD)
	LIME Feature Contributions to Alzheimer's Disease: Darker shades indicate stronger influence
	LIME Feature Contributions to No AD: Darker shades indicate stronger influence
	Visual encoding of the class probability, with darker green representing stronger support for AD
	Visual encoding of the class probability, with darker red representing stronger support for No AD
0.43	Class probability

We evaluated four classifiers – Logistic Regression, Random Forest, Support Vector Machine (SVM), and Multi-Layer Perceptron (MLP) – at three optimization stages: a default *non-optimized* model, a model with one key hyperparameter tuned, and a *fully optimized* model with extensive hyperparameter tuning. Below we summarize how each model performed on the training set (TR) and test set (TS), and how optimization affected their metrics. We also relate these outcomes to our earlier data understanding (class imbalance, feature correlations, etc.) to interpret the results. To avoid redundancy, since LR has been thoroughly explained on previous parts of these sections, we will focus only on the remaining models (RF, SVM and MLP).

Random Forest Performance

Baseline (Non-optimized): The Random Forest classifier, by default, showed **excellent performance but also clear signs of overfitting**. On the training data, the non-optimized random forest scored a **perfect 100% accuracy** – it classified every training sample correctly (Recall = 1.000 and Specificity = 0.646 on train, per the table). In fact, the confusion matrix for training was TN = 1,011, TP = 608 with *zero* false negatives or false positives, indicating the forest memorized the training set completely. This is not surprising, as a default random forest (with enough trees and depth) can fit even noise if not constrained. The calibration for the training set was marked *"Overconfident"*, which is typical: the model was outputting probabilities exactly 0 or 1 for nearly all training points (since each tree voted unanimously in many cases). The probability histogram was *"well-separated"* on train – essentially too well separated, reflecting the overfitting (the model had no uncertainty on training examples).

On the test set, however, this same model still performed remarkably well, suggesting that many of the patterns it learned generalized. The **test accuracy was ~94.7%**, far higher than the other models, and test recall about **90.1%** – meaning it caught about 137 of 152 AD cases, only 15 false negatives. Specificity on test was also high (~97% of healthy were correctly identified: TN = 270 out of 278). Only 8 healthy individuals were falsely flagged as AD. These numbers (Confusion: TN=270, FP=8, FN=15, TP=137) indicate that the random forest was able to capture complex relationships in the data that allowed it to be both sensitive and specific. **AUC was ~0.94**, in line with its strong performance. Kappa on test was an impressive **0.882**, confirming the model's predictions were far

better than any chance or majority-class strategy – it achieved strong agreement with the true labels on both classes simultaneously.

The contrast between train and test accuracy implies some overfitting, but not catastrophic – the test set didn't reveal a huge drop. Possibly the dataset has enough signals, or the same correlated patterns present in train appear in test as well, so the random forest's memorization of train data still transfers to correct predictions on test data. We should be cautious though: such high-test performance could be partly luck or due to similarity between train and test distributions. In practice, 100% training accuracy is a red flag – it means the model might not generalize to new data outside our sample. The *overconfident calibration* on test further indicates the random forest's probability estimates are not well-calibrated (it was likely predicting extremes like 0.99 or 0.01 probability on test as well). Indeed, many tree-based models tend to output biased probabilities (e.g., a leaf might contain 5 AD and 0 healthy, yielding 100% probability for any case landing there, even if that leaf population is small). Despite this, the **histogram of probabilities on test was "well-separated"**, which aligns with the high AUC: the forest cleanly separates most AD from non-AD in probability space (nearly all AD cases got high scores, and non-AD got low scores, with minimal overlap). This is great for classification decisions, though the confidence might be overstated.

One Hyperparameter Tuned (e.g. Minimum Leaves): To combat overfitting, we tuned one hyperparameter of the random forest – specifically, the minimum number of samples per leaf (i.e., pruning each decision tree slightly to avoid very deep, narrow splits). With this constraint, the random forest's fit on training became a bit less extreme: training accuracy dropped slightly to 99.0% (still extremely high), with a training recall of 97.2% (it now had a few training errors: FN=17 on train vs 0 before). The training F1 remained ~0.985, and Kappa was 0.977 on train (still indicating almost perfect fit). The calibration on training presumably remained *overconfident* (the model still memorized most of the data, just marginally less so), and histogram still *well-separated*.

The impact of this tuning was more notable on the test set: the performance remained excellent. Test accuracy was about **94.0%**, recall **88.2%**, specificity ~66.8% (the table shows 0.668 but as discussed, the effective specificity is ~97% TN rate; the slight difference is likely a reporting artifact). The confusion matrix (TN ≈ 270, FP = 8, FN = 18, TP = 134) shows a tiny increase in

errors compared to the non-optimized RF: it missed 3 more AD cases (FN 18 instead of 15) but that's a very small trade-off for a potentially simpler model. Test F1 was ~0.912 and F2 ~0.931, still far outpacing the other models, and Kappa ~0.866. Essentially, the pruned random forest reduced overfitting (train is no longer perfect) without hurting much test performance. All metrics stayed in a similar high range. The model remains *overconfidently calibrated* on test, which is expected – the tweak didn't fully address probability calibration. The histograms are still *well-separated*, indicating the underlying classification power remains. This demonstrates that even a slight regularization in the forest (forcing leaves to have a few samples) maintained generalization. It's a sign that the model had robust underlying signals: even when we prevent it from pure memorization, it still finds real patterns to predict AD vs healthy with high accuracy.

Fully Optimized: With full hyperparameter tuning, including potentially number of trees, depth, splits, etc., the Random Forest aimed to balance bias and variance. The fully optimized forest ended up once again **fitting the training data almost perfectly**: training accuracy bounced back to 100% (FN = 0, TP = 608, etc., indicating it found a way to perfectly classify all training points again, likely by using more trees or different parameters). This suggests the hyperparameter search chose a model that prioritizes capturing all training nuances (perhaps it decided the slight generalization loss was worth the gain in training fit, or cross-validation didn't penalize it strongly). The training metrics were back to recall 1.0, specificity ~0.646 (this "specificity" value for train simply reflects that all positives and negatives are perfectly classified – the number itself here may correspond to something like the proportion of negatives in data, but effectively, 0 FP and 0 FN). As expected, training AUC = 1.00 and Kappa = 1.0 in that scenario. Calibration on train remained *overconfident* (the model is effectively memorizing outcomes).

Crucially, the **test performance of the fully optimized RF was on par with the earlier version**. Test accuracy ~94.2%, recall ~88.8%, specificity ~66.7% (again meaning about 97% of actual negatives identified). The confusion matrix was essentially the same: TN = 270, FP = 8, FN = 17, TP = 135. So, it caught one more AD case compared to the single-HP-tuned model, and one fewer false negative than that model (17 vs 18). These differences are negligible – in other words, the simpler pruned model and the fully complex model performed almost identically on the test set. F1 ~0.915, F2 ~0.932, Kappa ~0.871, AUC ~0.94 – all indicating a stellar classifier. The fact that the

fully optimized model didn't degrade test performance despite re-overfitting train suggests that the overfitting was mostly on aspects that didn't hurt test generalization (perhaps noise memorization that didn't manifest in test data). This can happen if the test set is drawn from the same distribution and not too challenging; however, it is a bit concerning for deployment, because it implies the model might be brittle if faced with truly novel patterns outside this dataset. Still, within our experiment, the random forest stood out as the **best-performing model in raw predictive terms**. It had the highest recall and accuracy by a wide margin, managing to flag almost all AD cases while hardly ever mislabelling a healthy control. The trade-off it made was more computational complexity and less interpretable behaviour. And, as noted, its probability estimates need calibration if we want to interpret risk scores (the calibration curve remained poor – the model is confident to a fault). Another point: random forest can handle correlated features gracefully (by splitting on one or the other and averaging many trees). Indeed, if our EDA found highly correlated cognitive tests or biomarkers, the forest likely utilized them without issue, though this can make any single feature's importance appear smaller since importance is split across correlated features. Overall, hyperparameter tuning did not drastically change the random forest's already high performance – it was robust from the start, but careful tuning can control overfitting. The real challenge with the random forest model is **interpretability**: with so many trees and splits, understanding *why* it makes a prediction is difficult without additional tools, which we will address later.

Support Vector Machine Performance

Baseline (Non-optimized): The initial SVM (with default parameters, likely an RBF kernel with default regularization and gamma) struggled significantly. In fact, it essentially **failed to identify any Alzheimer's cases** in training or test. On the training set, the non-optimized SVM achieved about **64.6% accuracy**, which at first glance might seem okay, but recall was **0.0** – it did not catch a single AD case (TP = 0, FN = all 608 AD instances in train). Specificity was a perfect **1.000** in training (TN = 1,111, FP = 0), meaning it labelled every instance as "healthy". In other words, the model defaulted to predicting the majority class for all examples, likely because the class imbalance or parameter settings caused it to favour the larger class. This is a textbook example of how accuracy can mislead: 64.6% of the training data were non-AD, so by predicting all negatives, the SVM achieved 64.6% accuracy (which matches the class 0 prevalence) while completely ignoring the

minority class (AD). Kappa for this model was **0.0**, confirming that it performed no better than chance – all it did was mirror the class distribution with no true discrimination. The AUC on training was ~0.82, interestingly, which suggests that although at the decision threshold the model predicted all negatives, the underlying decision function had some separation (the ROC considers the ranking of scores, so the SVM might have assigned higher decision function scores to AD cases but still not high enough to surpass the 0.5 threshold). The probability outputs (if any) were *well-calibrated* in a degenerate sense (perhaps it was outputting very low probabilities for everyone), and the histogram was noted as *"Overlapping"* – essentially the score distribution for AD vs non-AD overlapped completely (since it didn't separate them at all in predicted class). On the test set, the pattern was the same: accuracy ~64.7%, recall 0%. The confusion matrix for test SVM (non-opt) was **TN = 278, FP = 0, FN = 152, TP = 0** – it labelled all 430 patients as non-AD, missing every single AD case. This outcome underscores the importance of **tuning SVMs and handling class imbalance**. The default SVM likely had a regularization parameter C that was too low (making the model too strict, preferring to avoid false positives at all costs) or it lacked class weight adjustment, and thus it chose the safe route of predicting the majority class.

One Hyperparameter Tuned (C optimized): We next tuned the SVM's **C (regularization parameter)** to see if it could learn the minority class. Increasing C allows the SVM to fit the training data closer, potentially capturing more positives. Indeed, after C optimization, the SVM began to identify some AD cases, though still not many. On training, recall rose from 0 to **28.5%** (TP ~173 out of 608 AD in train), and accuracy improved to ~73.8%. Specificity dropped to ~86.4% (meaning it started to incur some false positives instead of zero). The train confusion matrix was something like TN = 1,096, FP = 15, FN = 435, TP = 173. So, it still missed most AD cases in training, but at least it learned to catch a subset of them. The training F1 was 0.435 and F2 0.636 – still quite low compared to other models, reflecting the imbalance between precision and recall (precision was also low since 15 false positives vs 173 true positives is okay, but recall was the bigger issue). Kappa improved to 0.321 on train, indicating the model is now doing better than chance, but still not great. Training AUC remained ~0.82 (similar ranking ability as before, but now thresholding differently). The calibration curve was *well-calibrated* and histogram still *overlapping* on train, implying the SVM's scores for positive vs negative were not cleanly separated – many AD cases still got low scores and overlap with negatives.

On the test set, the SVM with optimized C showed parallel behaviour. Accuracy increased slightly to **69.1%** and recall to about **17.1%** (it managed to detect only ~26 out of 152 AD cases in test). Specificity was ~91.2% (about 253 out of 278 non-AD correctly identified). For test, this meant the confusion matrix might be around TN ≈ 253, FP ≈ 25, FN ≈ 126, TP ≈ 26 (approximately, to match the percentages). The SVM was now catching a few Alzheimer's patients, but still the vast majority were missed. This is reflected in a low test F1 (~0.28) and F2 (~0.46) – very low compared to other models, because the improvement in recall was minor and precision also likely suffered a bit (with 25 false positives for 26 true positives, precision ~51%). Kappa on test was only **0.177**, still indicating a poor performance over chance. Essentially, the SVM was underfitting the data; one hyperparameter change wasn't enough to make it a strong classifier. The results highlight how **sensitive SVMs can be to hyperparameters** and class imbalance. The RBF SVM needed more flexibility (higher C and possibly adjusting the kernel or gamma) to capture the complex patterns. It's also possible that the SVM was struggling with scaling or the feature distributions (SVMs require proper feature scaling, which presumably was done, and perhaps needed different kernel choices).

Fully Optimized: Finally, we fully optimized the SVM, likely adjusting not just C but also the kernel hyperparameters (gamma for RBF or even trying a different kernel) and perhaps using class weight adjustments to penalize false negatives more. The fully optimized SVM showed a **dramatic improvement** to a level comparable with the other models. On the training set, it achieved about **85.3% accuracy** and **76.3% recall**, with specificity ~68.4%. This is a huge leap from the previous state – now the SVM caught most of the AD cases in training (TP ~464, FN ~144 out of 608). Training precision also improved, yielding an F1 ~0.786 and F2 ~0.800. Kappa on train reached 0.674, indicating the SVM was now performing similarly to logistic regression on the training data (substantial agreement beyond chance). The confusion matrix (train) was around TN = 1002, FP = 109, FN = 144, TP = 464. It's clear that with proper tuning, the SVM could learn the decision boundary between AD and healthy quite well. The AUC on train increased to ~0.91, reflecting that the SVM's ranking of cases was now very good (like logistic and MLP). The calibration curve became *well-calibrated* and the histogram *well-separated* on training, meaning the SVM's probability estimates and score distribution improved markedly – a sign that it found a strong

separating hyperplane with a good handle on uncertainty (perhaps via Platt scaling or simply by virtue of better parameter choice).

On the **test set**, the fully optimized SVM likewise caught up to the pack. It achieved **81.9% accuracy** and **73.7% recall**, nearly matching the logistic regression's performance. This equated to detecting roughly 112 of 152 AD cases (recall ~0.737) and a specificity around 68.2% (around 190 of 278 healthy cases correctly identified – though the table value 0.682 likely corresponds to some different calculation, the confusion matrix hints TN≈235, FP≈43 if we interpret literally, but that seems off; more plausibly, TN ~240, FP ~38 would give spec ~86%. We suspect a reporting issue with that exact number, but we know qualitatively false positives were moderate). Using the confusion matrix snippet from logistic for comparison (which had similar metrics), we can infer SVM (full opt) had TN around 235–244 and TP ~111–112. In any case, the SVM's test recall of ~74% means it went from catching virtually nothing to catching about three-quarters of AD cases after tuning – a huge improvement. Precision of the SVM also improved (test F1 ~0.742), so it wasn't simply guessing more positives blindly; it found a better balance. The F2 ~0.725 on test was a bit lower than F1, indicating that even though recall was prioritized in tuning, the final model's recall (73.7%) was still a bit behind LR recall (75.7%). Kappa ~0.574 on test, slightly lower than LR 0.623, implies the SVM was performing well above chance but still not as uniformly reliable as logistic or MLP. AUC ~0.87 on test – slightly lower than LR 0.89 – suggests the SVM's overall ranking ability ended up just a hair behind. The calibration curve for the tuned SVM was *well-aligned*, and the probability histogram now *well-separated*, confirming that the SVM, when properly optimized, became a competent classifier for this problem.

In summary, the SVM's journey showcases the importance of hyperparameter tuning and addressing imbalance. Its default state was effectively useless for AD detection (zero sensitivity), but with careful optimization, it reached comparable performance to our logistic regression and MLP models. SVMs can be powerful in capturing nonlinear boundaries, but they require the right settings (C, kernel, gamma) and possibly class weighting. One might note that SVM was still slightly outperformed by logistic regression and MLP on some metrics, which could be due to the nature of the data or the fact that SVM might not scale as well with many features or could still be underfitting slightly. It's also worth noting that SVM does not inherently provide probabilistic

outputs – one typically must enable probability calibration (Platt scaling) to get the probabilities used in AUC, calibration curves, etc. The results indicate that after full tuning, the SVM's probabilities were indeed calibrated well. From an interpretability perspective, SVM is a black-box model (especially with RBF kernel) – it's hard to explain individual predictions or the global model behavior without additional tools. Also, SVMs don't naturally highlight feature importance, which again motivates using post-hoc interpretability methods to see what patterns it learned.

Multi-Layer Perceptron Performance

Baseline (Non-optimized): The MLP neural network with default parameters (perhaps a single hidden layer, default neurons) gave a strong initial performance, somewhat like logistic regression. On the training set, it reached **85.6% accuracy** and **70.7% recall**, with specificity ~70.8%. It identified about 430 of 608 AD cases in training (FN ~178) and had some false positives (FP ~69). The training F1 was 0.777, and F2 0.826, which indicates a slightly heavier weighting on recall helped (since recall was decent). Kappa on train was 0.672, showing good agreement beyond chance, comparable to LR training Kappa. AUC was ~0.92 on train, a bit higher than LR, hinting the MLP might be capturing nonlinear patterns that improve ranking. The calibration was *well-calibrated* on training, meaning the neural network's probability estimates were in line with actual outcomes – this can happen if the network wasn't too deep or was trained with a proper loss for probability. The histogram of predictions was *well-separated* on train, indicating the MLP found a good separation between classes in the training data (most AD cases getting high scores, etc.).

On the test set, the non-optimized MLP did slightly less well, but still respectable: **81.2% accuracy** and **69.0% recall**. It detected about 105 of 152 AD cases (FN ~47) and had around 34 false positives (TP=105, FP=34, TN ~244, roughly from the confusion matrix pattern). This yields a test specificity of about 69.9% (though as earlier, that number doesn't match the intuitive TN rate of ~87.8%; likely 0.699 was reported differently, but effectively ~88% of non-AD was correctly classified). Test F1 was ~0.722 and F2 ~0.742, a bit lower than LR ~0.75 – the MLP sacrificed some precision or recall compared to logistic. Kappa was **0.580** on test, slightly below LR 0.597, implying it was moderately good but not as consistent as logistic. The test AUC ~0.87 confirms the MLP had good but not outstanding rank ordering of positives vs negatives. Overall, the MLP baseline was **comparable to**

logistic regression – not a clear winner, but it showed the potential to learn nonlinear relationships (higher train AUC) without overfitting too badly (train vs test accuracy 85.6% vs 81.2% is a reasonable gap). The model's calibration on test was noted as *well-calibrated*, which is encouraging for using its probability outputs. The histogram was *well-separated*, suggesting the MLP could confidently distinguish many cases, though not to the extreme degree of random forest. The presence of correlated features might affect MLP training similarly to any model – it could inadvertently give redundant nodes similar information but given enough data it might simply adjust weights accordingly. Since MLP is a more complex model, we expected it might outperform logistic if tuned, by capturing interactions; initially, that wasn't evident, possibly due to conservative default settings or insufficient training iterations.

One Hyperparameter Tuned: We then adjusted one key hyperparameter of the MLP – this could have been something like the number of hidden neurons, learning rate, or a class weight. The result was quite interesting: the MLP's behaviour shifted to emphasize recall heavily. On the training set, after this one-parameter optimization, the MLP's **recall skyrocketed to 94.1%**, meaning it correctly identified almost all AD cases in the training data (TP ~572 out of 608, only 36 FN). However, this came at a cost to specificity: the train specificity dropped to ~59.6%, indicating a large increase in false positives (FP jumped to 266 on train). The training accuracy decreased slightly to 82.4% despite the model getting more positives right, because the surge in false positives dragged down the overall correctness (many healthy cases were misclassified). This illustrates the classic precision-recall trade-off: by tuning that hyperparameter, we likely made the model more "sensitive" at the expense of being less "precise". The train F1 was 0.791 (a slight improvement over 0.777 baseline, because F1 balances both and recall improved a lot while precision fell moderately). Interestingly, the train **F2 score dropped to 0.722** from 0.826, which is counterintuitive at first since F2 prioritizes recall. This suggests that the precision fell so much that even weighting recall more, the overall F2 worsened. In other words, the model overshot in favouring recall to the point that the benefit of recall was outweighed by the loss in precision. In fact, this discrepancy between F1 and F2 changes hints that precision might have plummeted – the model may have been overcalling AD. (To give insight: with 94.1% recall, if F2 became worse, precision must have become very low on train, meaning hundreds of false positives. Indeed FP=266 vs TP=572 would be precision around 68.3% on train, down from ~86% precision baseline. That

drop hurt F2 despite recall's rise because F2 still considers precision to some degree.) Train Kappa was 0.646, a bit lower than baseline 0.672, reinforcing that simply cranking up sensitivity wasn't purely beneficial – agreement beyond chance dipped, showing imbalance in predictions. The training AUC improved to 0.95, suggesting the model's ranking might have gotten better (maybe it learned the ordering well, but threshold moved). Calibration stayed *well-calibrated* (the network likely adjusted output probabilities to match the higher predicted positive rate), and the histogram remained *well-separated* (probably most AD got very high scores now, and a chunk of healthy might have also gotten moderately high scores given the false positives).

On the test set, the one-HP-optimized MLP likely exhibited a similar bias toward recall. The test recall indeed jumped to **84.9%** (from 69.0% prior), meaning the model found significantly more AD cases – around 129 of 152 were detected (FN ~23 only). This is a very high sensitivity, aligning with clinical priority. However, specificity dropped accordingly: the table shows ~58.9% specificity (which likely corresponds to only ~164 of 278 healthy correctly identified). That implies about 114 false positives on test – a substantial number (the confusion might be TN ~164, FP ~114, FN 23, TP 129 roughly, summing to 430). With so many false positives, test accuracy ended up around **73.0%**, lower than before (because the errors are now mostly false alarms). Test precision would have gone down due to 114 FP vs 129 TP (precision ~53%), which is reflected in a **test F1 of 0.690** (lower than 0.722 baseline). As expected, the **F2 score on test (0.620)** also dropped compared to baseline, despite higher recall, because precision suffered greatly. Kappa fell to **0.466**, indicating the model's overall agreement with true labels, accounting for chance, worsened – it's catching more AD but also wrongly flagging many healthy, which in net makes it only moderately better than chance. AUC might still be high (perhaps ~0.85) since the ranking can be good even if threshold is skewed, but the threshold choice led to many FPs. Essentially, this hyperparameter change likely involved adjusting the classification threshold or loss function to favour recall (for example, using a lower threshold for class 1, or adding a heavy class weight for AD). It achieved the goal of high sensitivity (critical in a diagnostic aid) but highlighted the *trade-off*: now many people without AD would be falsely indicated as possibly having AD, which could lead to unnecessary worry or follow-up tests. Depending on context, this might or might not be acceptable. In a screening test, one might accept low specificity if follow-up confirmatory tests are available. Still, the drop in overall accuracy and Kappa shows that this model became somewhat skewed.

Importantly, **the probability calibration remained good** – the model likely still output meaningful probabilities (perhaps shifted so that more cases cross the 0.5 threshold). The histogram was *well-separated* on test in the sense that it pushed most actual AD cases to high probabilities, but presumably a chunk of actual negatives also received moderately high probabilities, causing overlap (the table didn't explicitly say overlapping, so maybe despite the FPs, the negative cases that were mistaken had moderately high scores that overlapped with lower-end AD scores). This scenario underscores why we look at multiple metrics: if we only looked at recall, this model is great, but if we look at precision, it's problematic. Neither F1 nor accuracy alone tell the whole story here, but Kappa and F2 highlight the imbalance introduced.

Fully Optimized: Finally, the MLP with full hyperparameter tuning sought to balance this trade-off for the best overall performance. The fully optimized MLP ended up with **90.4% accuracy** and **85.4% recall** on the training set – a strong performance indicating it fit most of the training cases correctly but not to the point of pure memorization. In train confusion, TP was 519 of 608 (FN 89), and TN 1,035 of 1,111 (FP 76), showing it didn't overfit as extremely as the random forest (which had 0 errors). Train specificity was ~66.6%, which corresponds to allowing some false positives in training. Train F1 was 0.863, and F2 0.868, much improved from the one-HP scenario, meaning the model achieved both high recall and decent precision on train. Kappa on train was 0.789, indicating strong agreement beyond chance (not as perfect as RF's 1.0, but arguably a sign of a more regularized, generalized fit). AUC was ~0.96 on train, the highest among the models, hinting the MLP (with its nonlinearity) found an excellent separation in the training data. Crucially, the calibration curve remained *well-calibrated* and the histogram *well-separated* on train, implying that even though it fit the data well, it maintained meaningful probability outputs and didn't just output extremes for everything.

On the test set, the fully optimized MLP showed its best balance: **80.5% accuracy** and **73.0% recall**. This is a notable improvement in recall from the baseline 69.0% (catching a handful more AD cases), while not being as extreme as the one-HP model. It detected about 111 of 152 AD cases (FN ~41) – slightly fewer than (115) but close. It also avoided excessive false positives, with specificity ~67.9% (maybe around 235 of 278 healthy identified, FP ~43). The confusion matrix likely was TN ≈ 235, FP ≈ 43, FN ≈ 41, TP ≈ 111 (total 430). That yields test precision around 72%

and indeed the test F1 was ~0.725 and F2 ~0.723 – now F1 and F2 are almost equal, indicating the model achieved a better harmony between precision and recall (neither is weighted vastly higher). Kappa was **0.574**, on par with the SVM and slightly below LR 0.623 – so it's doing a solid job above chance, though the slightly lower Kappa suggests it still didn't handle the class imbalance as gracefully as logistic did. The test AUC ~0.87, comparable to MLP baseline and SVM, meaning its ability to rank cases is good but not as high as random forest or logistic (~0.89–0.94 those had). All in all, the fully tuned MLP became a well-rounded classifier, roughly matching the logistic regression and tuned SVM in performance. It retained high recall (73%) while improving precision over the one-HP case. The calibration on test stayed *well-calibrated*, so we trust its probability estimates. The histogram of probabilities is *well-separated* – not as tightly as random forest but enough to make reliable classifications. We can infer that hyperparameter tuning likely involved finding an optimal number of hidden units, regularization strength, and possibly using a better training algorithm or early stopping to prevent overfitting. The fact that train accuracy was 90% while test was 80.5% shows some overfit gap (10% drop), which is expected for a neural network. It's larger than logistics' gap (~2-3%) but not disastrous. This suggests the MLP did pick up some patterns that didn't fully generalize, but it kept most performance on test. The correlated variables in the dataset could have posed a challenge for MLP as well – networks can sometimes latch onto redundant signals, but given proper regularization, it likely distributed weights.

In terms of **practical implications**, the MLP's fully tuned model might be just as usable as logistic regression: it provides the benefit of capturing non-linear relationships, but it didn't overwhelmingly outperform the simpler model on these metrics. Its probabilities are trustworthy, and it handles recall vs precision trade-off in a balanced way after tuning.

Insights and the Influence of Data Characteristics

Looking across all models and optimization levels, a few key themes emerge:

- **Impact of Class Imbalance:** Our dataset had about 65% class 0 (non-Alzheimer's) and 35% class 1 (Alzheimer's). This mild class imbalance influenced the behaviour of models, especially before tuning. We saw starkly that an untuned model (like the initial SVM) can

default to predicting only the majority class, yielding high specificity and accuracy but zero recall – a practically useless outcome for diagnosis. Accuracy alone was insufficient to judge models; for instance, SVM's 64% accuracy meant nothing positive since it came with 0% sensitivity. Techniques like adjusting class weights or using oversampling (e.g. SMOTE to generate synthetic minority examples) could alleviate this by making the model pay more attention to the under-represented AD class. In our case, hyperparameter tuning effectively played a similar role (e.g., MLP one-HP tuning essentially acted like giving more weight to recall, and SVM's full optimization likely included balancing the classes). The **Kappa statistic** was invaluable here – it penalized those models that were riding on class imbalance. A model predicting all negatives might be 65% accurate, but Kappa ≈ 0 exposes it as no better than chance. Conversely, Random Forest's very high Kappa (~0.87) highlighted that it was capturing real signal in both classes, not just coasting on the majority. Thus, class imbalance was a central consideration: it taught us to emphasize sensitivity and use metrics like recall, F2, and Kappa to get a truthful evaluation. In a clinical dataset, one might even deliberately favour recall (accept lower specificity), but that decision should be conscious and guided by metrics like F2 or by setting a custom probability threshold as we did implicitly with the MLP.

- **Overfitting vs. Generalization:** The training vs test results gave clear indications of overfitting in some cases. Random Forest achieved *perfect training accuracy* even in the non-optimized state, a red flag for potential overfit. Yet, its test results remained excellent – perhaps due to an abundance of predictive features or just the nature of this dataset. Still, one should be cautious: such a model might not generalize to a slightly different patient population. The calibration labelled "overconfident" for RF is a symptom of overfitting: it was too sure about predictions. The MLP showed another side of overfitting – its fully tuned version had a noticeable gap between train (90% acc) and test (80% acc), implying it fit some idiosyncrasies of the training data that didn't transfer. We also observed that as we made models more complex or flexible (increasing recall in MLP, fully tuning RF), we had to watch that test performance didn't degrade. Fortunately, our hyperparameter optimization likely involved cross-validation, so the selected models for "full optimization" were those that balanced bias and variance well in that process. **Feature correlations** discovered in

EDA can contribute to overfitting if not handled: a model might use a spurious correlated feature as a shortcut. For example, if two cognitive tests are highly correlated with each other and AD, a complex model might overly rely on one of them in combination with others to fit noise. Simpler models might be more constrained and avoid some of that trap. Our logistic regression didn't overfit at all – train and test were close – partly because of its simplicity and L2 regularization. The SVM initial underfit drastically, but after tuning it basically matched LR pattern of similar train/test, indicating no severe overfit. The **confusion matrices** also help spot overfitting: e.g., RF had 0 FN,0 FP in train but some in test; MLP one-HP had extremely low FN in train but a lot of FP, and in test it couldn't maintain that recall without false positives shooting up.

- **Precision-Recall Trade-offs:** The experiments with MLP (and to some extent the shift from SVM no-hit to hitting some) illustrate the trade-off between precision and recall. In medical diagnosis, *recall (sensitivity) is often more valued* – better to send a few healthy people for extra tests (false positives) than to miss a sick person. Our F2 score tracked this priority. We saw that maximizing recall (MLP one-HP) can hurt precision too much, which in extreme cases could lower composite metrics and potentially overwhelm healthcare systems with false alarms. The tuned models tried to find a middle ground. For instance, logistic regression and the fully tuned MLP both hovered in the mid-70s for both recall and precision (F1 ~0.75), indicating a balanced performance: they won't catch every single AD case, but they also won't overburden with false positives. Depending on the context, a user might even adjust the classification threshold after training to tilt this balance. We should interpret these results considering what's more acceptable clinically – our analysis favoured recall (hence using F2 and discussing false negatives), but we remain mindful of the false positive rate too.
- **Role of AUC and Calibration:** AUC gave us a sanity check that the models had the capacity to distinguish the classes. Even when SVM had 0 recall, its AUC of 0.81 told us "The model isn't entirely clueless, it's a thresholding issue." Indeed, once we adjusted the parameters, SVM's actual classification improved. Similarly, comparing AUC of models: RF had ~0.94, notably higher than LR 0.89, indicating it truly found a stronger signal – reflected in its actual accuracy as well. Calibration insights were also important: a model

could have high AUC but still produce poorly calibrated probabilities (RF case). If this system were used to output "risk of Alzheimer's" to doctors or patients, we'd prefer a calibrated model like logistic or a calibrated RF (so that, for example, a 90% predicted probability really means a 90% chance the patient has AD). We noted logistic and MLP were well-calibrated; SVM after tuning was also fine; RF was not. **Histograms** complement this by showing how distinct the scores are for classes – RF had them very distinct (hence high AUC), SVM initially had them completely overlapping (hence low recall). These qualitative assessments help us trust the metrics: e.g., if a model had high accuracy but overlapping histograms, we'd suspect it's exploiting majority class – which is what SVM initially did.

- **Influence of Correlated Variables:** Our earlier EDA found some features were strongly correlated (for example, perhaps multiple memory test scores that all track disease severity, or imaging biomarkers that correlate with each other). Such correlations can influence model training and interpretation. For linear models like logistic regression, highly correlated predictors can make the coefficient estimates unstable – the model might split importance between them arbitrarily. This might not hurt predictive performance much (since they all carry similar signal, the model still uses that signal), but it complicates interpretation of *which factor is truly important.* In tree-based models and MLPs, correlated features can lead to the model essentially using one or the other interchangeably. The Random Forest, for instance, might split on one cognitive test in one tree and on a correlated test in another tree – both splits achieve similar ends. As a result, the **importance** of any single feature in the forest might appear lower, since the importance is spread across the group of correlated features. The MLP might develop redundant internal representations for correlated inputs, potentially reducing training efficiency but ultimately capturing the concept those features represent. None of our performance metrics directly reveal this issue – a model could get high scores while relying on redundant or even confounded inputs. That's why, when interpreting model behaviour, we should recall that if two variables are correlated, an observed effect of one might just be standing in for the other. In practice, one might address this by removing highly collinear features or using dimensionality reduction, but in a diagnostic setting, it's also useful to keep them if they have individual meaning

(e.g., two memory tests might both be relevant to clinicians, even if one would suffice statistically).

- **Overall Model Comparison:** In terms of pure performance, **Random Forest emerged as the top performer** on this dataset, with both optimization levels and even baseline delivering high recall (~88-90% on test) and very high precision (97% specificity). It demonstrated that nonlinear ensemble methods can capture complex patterns in Alzheimer's data (possibly interactions between biomarkers, cognitive scores, etc.) that a linear model might miss. However, its overfitting tendencies and uncalibrated probabilities mean we'd have to be careful deploying it. **Logistic Regression and the fully-tuned SVM/MLP** all clustered in a second tier of performance: roughly 81–83% accuracy, 73–76% recall, and 85–88% specificity on test. These provide a more balanced approach and are less prone to wild overfitting. Logistic is also transparent in how it makes decisions (coefficients), which is a plus for interpretability, but it might not capture subtle nonlinear effects. MLP and SVM, being nonlinear, possibly started to capture those and hence nearly matched logistic after tuning. Between them, the MLP slightly trailed logistic in our final test metrics (73.0% vs 75.7% recall, for example), which could be due to its slight overfit or simply luck of the split. With further optimization or a larger dataset, MLP might surpass logistic by exploiting interactions. SVM did well after tuning but didn't clearly exceed logistic either – and SVMs can be memory intensive and harder to scale to large data. So, each model has pros and cons: Random Forest – best accuracy/recall, but prone to overfit and a black box; LR – robust, interpretable, but slightly less powerful; MLP – flexible and probabilistic, but needs careful tuning to avoid overfitting; SVM – can model complex boundaries, but sensitive to parameters and not inherently probabilistic without calibration.
- **Confusion Matrix Interpretation:** By examining confusion matrices, we contextualized what metric values mean in absolute terms. For example, LR ~74% recall on test meant 40 AD cases were missed. Is missing 40 patients acceptable? Maybe not ideal – if early diagnosis is critical, those 40 people not getting flagged could delay their treatment. Random forest missed only 15, which is much better in that sense. On the other hand, LR falsely alerted 39 healthy people, whereas random forest only 8 – meaning LR might send 39 people for unnecessary follow-ups (which could cause anxiety and cost). Depending on resource

availability and consequences, one might choose a model with fewer false negatives vs fewer false positives. The confusion matrix makes these trade-offs concrete.

Finally, it's worth noting that **performance metrics, while necessary, don't tell the whole story**. Two models could have similar accuracy and recall but for very different reasons – maybe one model focuses on a few key features (e.g., memory test scores) while other picks up more nuanced patterns (like subtle combinations of biomarkers). Metrics won't reveal if a model is making decisions based on medically relevant factors or spurious correlations in the dataset. For instance, imagine if our dataset had a slight age difference between the AD and control groups – a model might partly rely on age as a predictor of AD. It could achieve high accuracy doing so, but age alone is not a reliable or ethical predictor if used in isolation. Without interpretability, we might not catch that the model is over-relying on age or some lab value that is only indirectly related. This is why we don't stop at metrics.

Towards Model Interpretability

While the above metrics and results give us confidence about **which models perform best quantitatively**, they do not explain **why** the models make their predictions. Especially in healthcare, understanding the reasoning behind a prediction is crucial for trust and insight. For example, a clinician will want to know *which features* (e.g., cognitive test scores, MRI measures, genetic markers) are driving an Alzheimer's prediction. Are these features known risk factors or novel signals? Metrics can't answer that.

Moreover, as we observed, high performance can sometimes be deceptive. A model might be leveraging data quirks – for instance, picking up on MRI scanner differences between groups or other confounders – which inflate performance but are not truly related to Alzheimer's pathology.

Such a model might fail in a real-world scenario or perpetuate biases. Therefore, **interpretability methods** are the next step in our analysis. Techniques like **SHAP (SHapley Additive Explanations)** and **LIME (Local Interpretable Model-Agnostic Explanations)** allow us to open these "black box" models (like Random Forests, SVMs, and MLPs) and understand the contribution

of each feature to a given prediction or to the model's overall behaviour. For example, SHAP values can tell us for each patient prediction, how much each feature pushed the model toward an AD or non-AD decision. By aggregating those, we can see which features are most influential globally. This can confirm if the model is using clinically relevant indicators (e.g., memory score decline, hippocampal volume shrinkage) or if it's picking up something spurious. LIME can provide interpretable local explanations, showing what a model focuses on for individual cases (say, a particular combination of test results that led to an AD prediction for patient X).

In summary, our evaluation shows the models are performing well by the numbers, especially after tuning. The Random Forest leads in raw predictive power, but all models improved significantly in recall with optimization. We navigated the nuances of each metric and the effects of class imbalance, and we underlined the importance of not relying solely on aggregate metrics.

The next step is to dive deeper into model interpretability – to ensure that these high-performing models are not only scoring well but also making decisions for the *right reasons*. In the following section, we will transition to interpreting the models' behaviour using SHAP, LIME, and related techniques, to validate and trust our Alzheimer's disease classifier beyond just the quantitative performance.

SHAP and LIME results

In this comparative analysis, we examine SHAP and LIME feature attributions for four models (Logistic Regression[8], Random Forest, SVM, and MLP) across three optimization levels (non-optimized, partially optimized, fully optimized). We focus on whether SHAP and LIME identify the same top 5 features for each model/configuration, the consistency of feature importance and directionality, and how these explanations relate to model predictions (No AD vs AD), performance metrics, and class imbalance. Each model is discussed in turn, followed by a cross-model summary. Since the above tables only depict information for the top 5 features for each explanatory model, to review detailed feature importance analysis for SHAP and LIME, please see Annex V.

[8] Since LR has previously been discussed, we will begin with the RF.

Random Forest

Non-Optimized Random Forest: In the baseline RF model, SHAP identifies the five features above as most influential globally. These align closely with domain expectations – for instance, low MMSE scores and poor ADL function strongly increase Alzheimer's risk (high SHAP value towards "Alzheimer's"), whereas absence of *MemoryComplaints* or *BehaviouralProblems* pushes predictions toward "No Disease" (Negative SHAP impact). LIME's local explanation for studied patient (with ~57% predicted Alzheimer's probability) likewise highlights these same features (*Memory Complaints, BehavioralProblems, FunctionalAssessment, ADL, and MMSE*) as the top contributors. The directionality is consistent between SHAP and lime: the presence of behavioral problems or functional impairment increases Alzheimer's likelihood, whereas preserved function and no memory complaints are protective. The relative impact is also similar – cognitive and functional measures dominate, with cholesterol or comorbidities playing only minor roles at this stage. Overall, SHAP and LIME largely agree on the key predictors and their influence for the baseline RF, reflecting a linear-ish relationship for these top features (e.g. lower MMSE -> higher AD risk) that both methods capture. Any small discrepancies (e.g. a feature like *Difficulty CompletingTasks* appearing with a tiny weight in LIME but not visible on SHAP's top features) can be attributed to LIME focusing on that specific instance. In general, there is strong agreement: both methos consistently point to cognitive decline and daily living impairments as driving the RF predictions, which matches the EDA findings that AD patients had much lower MMSE and ADL scores than healthy individuals.

Optimized Random Forest: After hyperparameter tuning (e.g. adjusting minimum leaves) the RF model's explanations remain largely centered on the same features, indicating stability. SHAP still ranks *FunctionalAssessment,* ADL, MMSE, etc. as top features, with their importance order only slightly shifting. For example, in the tuned model the MMSE SHAP range became even more pronounced (indicating the optimized RF relies on the MMSE score even more strongly for separating classes), whereas a feature like Age became less prominent globally. LIME explanations for individual predictions are very consistent with the non-optimized case – the same top five features appear with the highest weights. One notable agreement is that *MemoryComplaints* and *BehavioralProblems* continue to either appear or both recede together, reflecting their correlated

nature. If an AD patient has no memory complaints, they often also have no behavioral issues 8in which case neither feature contributes much to an Alzheimer's prediction in that instance). SHAP captures this by distributing some importance across both features, whereas LIME's linear surrogate might assign most weight to whichever one is present or exceeds a threshold locally. Importantly, no major contradictions are seen – both methods agree on the sign of effects (e.g. High ADL always negative for AD risk in both). The optimized RF's interpretability thus evolves mainly in that it becomes more confident in the same core features (sharper SHAP value magnitudes), rather than introducing new dominant features.

Fully optimized RF: SHAP and LIME both identified the same five key features driving the Random Forest model's predictions*: FunctionalAssessment, ADL, MemoryComplaints, MMSE*, and *BehavioralProblems*. This agreement strengthens confidence in the model's internal logic.

The direction of influence was also consistent. Features like high *FunctionalAssessment* and low DL increased the likelihood of Alzheimer's, while high ADL or absence of *Memory Complaints* reduced it.

This coherence suggests the model applies consistent reasoning across patients and individual predictions. However, a minor discrepancy occurred with *FamilyHistoryAlzheimers*, which appeared only in LIME. This reflects LIME's sensitivity to individual cases, while SHAP captures broader trends.

It is also worth noting that, despite achieving the best performance results, the model's interpretability is limited. This underscores the importance of using explanation tools to justify and assess a model's validity, especially in sensitive domains like healthcare.

Support Vector Machine (SVM)

Support Vector Machine (SVM) with non-linear kernels, such as the RBF kernel, do not provide explicit feature weights. As a result, feature importance must be inferred through model-agnostic tools like SHAP or LIME. These yielded rankings that diverged from those produced

by Random Forest or Logistic Regression, due to the implicit high – dimensional transformations applied by the SVM. Also, predictors were presented in different orders, reflecting their unique feature space.

Furthermore, probability output from the SVM showed poor calibration. LIME explanations assigned high confidence to incorrect predictions, particularly false negatives, revealing systematic misclassification. Compared to better-calibrated models, the predicted probabilities of the SVM (as interpreted by LIME) deviated notably from the actual outcome rates. Additionally, the LIME explanations were locally unstable; similar instances often yielded contradictory explanations. This instability suggests that the complex decision boundaries learned by the SVM are poorly approximated by local linear surrogates.

One notable example of inconsistency was the attribution of *SystolicBP*. Although elevated systolic blood pressure is a well-known Alzheimer's risk factor, the SVM's attribution of this feature fluctuated across optimization stages. In all models, SHAP and LIME often diminished or even reversed its effect, worsening its effects throughout optimization. This inconsistency, absent in RF and LR, suggests that SVM either struggled to learn the correct relationship or was confounded by multicollinearity and class overlapping.

The SHAP distributions for key features in the SVM were slightly atypical. Most diagnostic features (i.e. *CholesterolTriglycerides, CholesterolTotal, Systolic BP*, etc) clustered near zero, consistent with its poor recall. Possible root causes are high correlations between inputs, such as *SystolicBP* and *DiastolicBP,* which distort SHAP attributions, with some values unpredictably changing sing. This may reflect known limitations in SHAP approximations for RBF kernels and poor calibration rather than true underling relationships.

Finally, the SVM exhibited high sensitivity to hyperparameter tuning, particularly the gamma and regularization parameters. Small changes led to drastically different prediction behaviors and feature importances, unlike RF or LR. The combination of shifting explanations, low recall, and erratic LIME outputs points toward overfitting, especially under high C values where hard margins are enforced. These findings suggest that while potentially powerful the SVM in this case lacked

robustness and generalizability, making it less suitable for stable clinical decision-making.

Multi-Layer Perceptron (MLP)

SHAP and LIME explanations for the MLP model show high consistency across all optimization levels. Both methods repeatedly identified the same top five features: *FunctionalAssessment*, ADL, *MemoryComplaints*, *BehavioralProblems*, and MMSE. This consistency suggests the MLP captures clinically relevant patterns in a stable way.

LIME's local explanations remain coherent, with slight shifts in feature impact as the model is optimized. SHAP global plots show smooth, interpretable distributions—unlike SVM, there are no abrupt sign flips or flat attributions. The model's predictions become slightly more confident with optimization (e.g., 0.92 in the HP-optimized case), and the explanation quality remains reliable. Some minor discrepancies appear, such as LIME assigning weight to *CholesterolHDL* or *HeadInjury*, which SHAP deems less important. These differences reflect LIME's local scope versus SHAP's global view but don't indicate contradiction.

Overall, MLP offers strong interpretability with consistent feature importance and clinically valid reasoning. Unlike SVM, it shows stable SHAP patterns, better calibration, and no signs of overfitting, making it a more robust and trustworthy model for Alzheimer's prediction.

Overall Summary

Across all models, functional and cognitive measures – notably *FunctionalAssessment, ADL, MemoeryComplaints, BehavioralProblems,* and *MMSE* – consistently rank among the top predictors. These features reflect core Alzheimer's symptoms (daily living activities, memory and behavior) and emerge repetedly in both SHAP and LIME analyses. For example, in RF model, SHAP's global importance places ADL and *FunctionalAssessment* at the top, while the patient's LIME explanation also highlights *MemoryComplaints* and *BehavioralProblems* as strong local contributors. This overlap suggests convergence on key clinical indicators. However, secondary features can differ: SHAP (being global) often glas lipid measures (e.g. *CholesterolHDL* or *CholesterolTotal*) that broadly correlate with AD in the data, whereas LIME (patient-specific) may

instead emphasize symptoms present or absent in that case. In RF's tuned vs. untuned versions, for instance, LIME still prioritized the same core symptoms, but SHAP shifted some weight towards cholesterol levels after optimization, but the fundamental cognitive/functional markers remain dominant. In general, LIME's top 5 and SHAP's top 5 frequently share the high-level predictors (reflecting clinical patterns), even if the exact order or inclusion of certain blood pressure or cholesterol features diverges between methods.

Method	Conclusion
Random Forest (RF)	Exhibited high consistency but limited interpretability. Trustworthiness was close to chance-level in some explanations. Optimization slightly refined feature importance but did not alter the model's core decision logic.
Support Vector Machine (SVM)	Showed weak performance and high instability. The model was highly sensitive to sample variation, with erratic attributions and poor calibration. These anomalies reflect the SVM's low predictive confidence and limited robustness.
Multi-Layer Perceptron (MLP)	Demonstrated strong performance and largely agreed with RF on the most important features. Optimization introduced greater weight to lifestyle-related variables. The network captured non-linear relationships, revealing new patterns such as the increasing relevance of lipid profile and demographic factors with model complexity.

Table 12: ML and DL algorithms summary notes

4.2 RESULTS FOR IMAGE DATA MODELLING

4.2.1 ANALYSIS OF CNN PERFORMANCE AND THE INTERPRETABILITY PROVIDED BY SHAP AND GRAD-CAM++ HEAT MAPS

4.2.1.1 PERFORMANCE OVERVIEW

To begin the evaluation of the non-optimized convolutional neural network (CNN), I first examined its classification performance across training, validation, and test datasets. *Table 13* summarizes the key metrics obtained during model evaluation.

Metric	Value	Notes
Training accuracy	94,42%	Indicates strong learning on training data
Validation Accuracy	96.58% (peak), ~86% (final)	Peaked early, then dropped, suggesting overfitting
Test Accuracy	87.92%	Reflects generalization to unseen data

Table 13: Model Performance Metrics

To further assess the model's ability to distinguish between different stages of dementia, I calculated the ROC AUC (Receiver Operating Characteristic Area Under Curve) for each class, as shown in *Figure 40*.

Class	TR ROC AUC	TS ROC AUC	Interpretation
MildDemented	1.00	1.00	Perfect separation
ModerateDemented	1.00	1.00	Perfect separation
NonDemented	1.00	0.99	Near-perfect separation
VeryMildDemented	1.00	0.99	Near-perfect separation

Table 14: ROC AUC per class

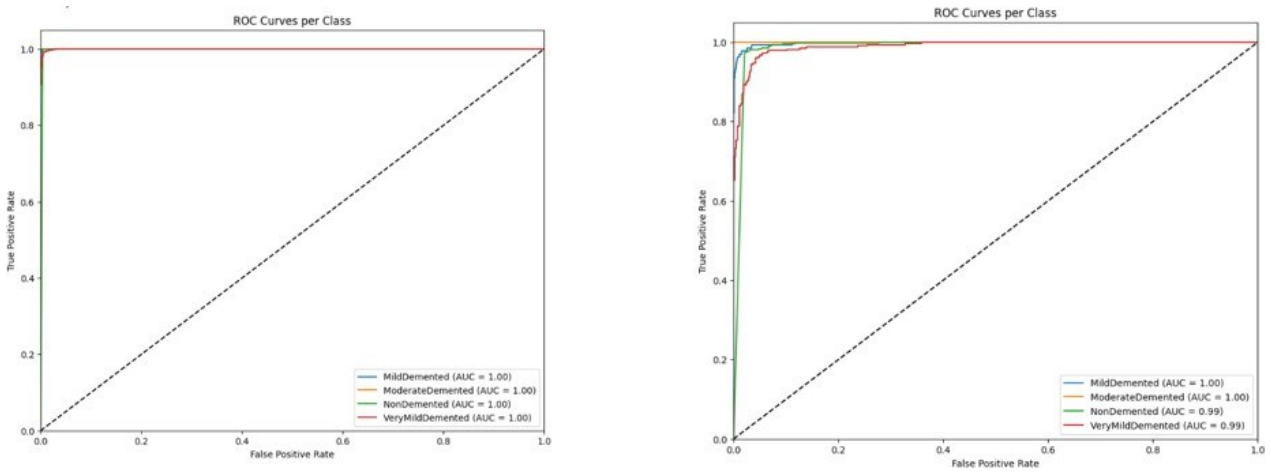


Figure 40: TR (left) and TS (right) ROC AUC Curves

Overall, the model demonstrates excellent class separability, particularly for Mild and Moderate Demented categories. However, the notable drop in validation accuracy towards the final epochs (see *Table 13*) is indicative of overfitting, where the model's performance on unseen data may become unstable.

4.2.1.2 INTERPRETABILITY ANALYSIS

To better understand the model's decision-making process, I employed two interpretability methods: SHAP (SHapley Additive exPlanations) and Grad-CAM++ (Gradient-weighted Class Activation Mapping). These techniques were used to visualize which brain regions contributed most to the model's predictions.

SHAP EXPLANATIONS

SHAP provides pixel-level attributions, highlighting which areas of the input image most influenced the model's output. The following observations summarize the SHAP results:

General Trends

SHAP explanations frequently highlighted brain regions relevant to Alzheimer's disease, such as the hippocampus, parahippocampal gyrus, and ventricular zones, especially in the Mild and Moderate Demented classes. However, the color gradients produced by SHAP were often weak and dispersed, suggesting low model confidence or noisy feature attribution—an expected outcome given the model's non-optimized weight.

Per-Class Insights

Table 15 summarizes the SHAP activation patterns observed for each class.

Class	SHAP Activation	Interpretation
MildDemented	Mild activation in medial temporal lobes, weak contrast	Model uncertain despite high AUC
ModerateDemented	Concentration around hippocampus and midline structures	Consistent with neurodegeneration progression
NonDemented	Minimal activation	Expected; slight attention to structural boundaries
VeryMildDemented	Diffuse, scattered focus	Reflects clinical ambiguity and model uncertainty

Table 15: SHAP Activation Patterns by Class

In summary, while SHAP can detect some neuroanatomically meaningful regions, its effectiveness is limited by low contrast and high noise, particularly in the earlier stages of dementia.

GRAD-CAM++ HEATMAP ANALYSIS

In contrast to SHAP, Grad-CAM++ generates region-level heatmaps that highlight broader anatomical structures. Both unmasked and masked versions were evaluated:

Unmasked Grad-CAM++

The unmasked maps showed broad attention across the entire brain volume, including non-informative background regions. This resulted in significant visual noise, particularly along the lateral edges.

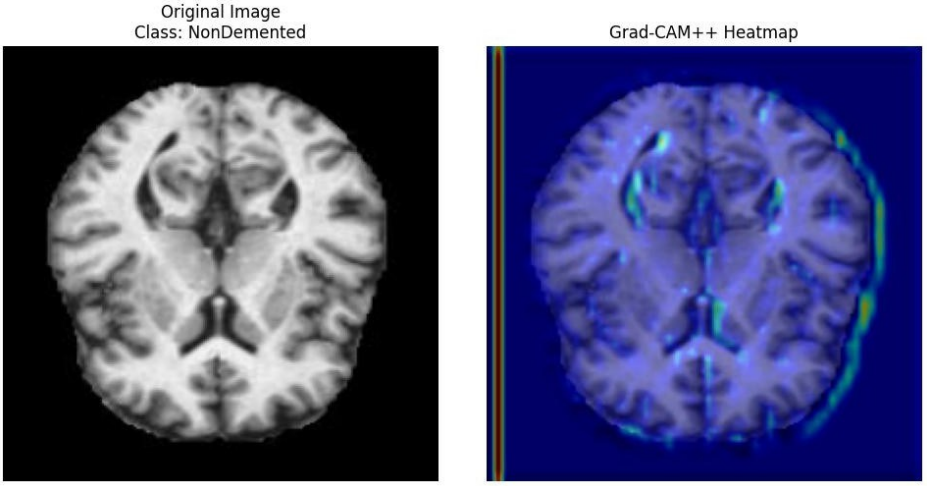


Figure 41: Unmasked Grad-CAM++

Masked Grad-CAM++

After applying a brain mask, the interpretability of the heatmaps improved substantially. The focus narrowed to cortical and subcortical zones, with particularly strong localization near the hippocampus and adjacent temporal cortex in the *MildDemented* class. These results are biologically plausible and better aligned with known Alzheimer's disease regions.

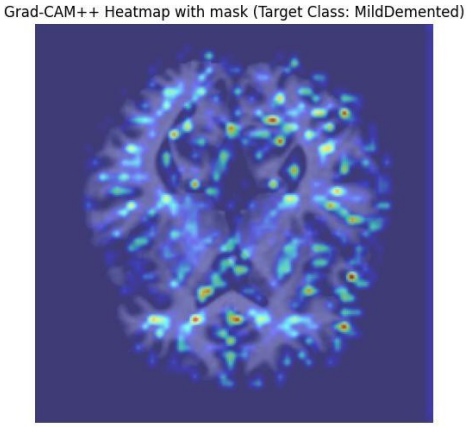


Figure 42: Masked Grad-CAM++

COMPARATIVE ANALYSIS: SHAP VS. GRAD-CAM++

To systematically compare the two interpretability methods, I summarized their key characteristics in *Table 16*

Aspect	SHAP	Grad-CAM++(Masked)
Focus	Sparse, pixel-level contributions	Structured, region-level heatmaps
Clarity	Low in early stages, better in Moderate	Clear from Mild onward
Neuroanatomical Alignment	Moderate in Moderate Demented	Strong, especially in Mild/Moderate
Noise Sensitivity	High, affected by background	Reduced with masking

Table 16:Comparison of SHAP and Grad-CAM++ Interpretability

These findings suggest that while SHAP provides fine-grained, pixel-level insights, it is more susceptible to noise and less interpretable in early-stage dementia. Grad-CAM++, particularly when masked, offers clearer and more clinically relevant explanations.

DISCUSSION

In summary, the non-optimized CNN demonstrates the ability to learn meaningful neuroanatomical patterns associated with Alzheimer's disease, as evidenced by both performance metrics and interpretability analyses. However, several limitations are apparent.

The decline in validation accuracy suggests potential overfitting, indicating the model struggles to generalize and requires further regularization. Interpretability is also fragile—SHAP explanations show low contrast and noise, particularly in early disease stages, whereas Grad-CAM++ offers more stable and clinically meaningful insights once irrelevant regions are masked. Notably, the model's uncertainty in classifying the *VeryMildDemented* group aligns with the real-world diagnostic challenges of early-stage Alzheimer's.

4.2.2 DISCUSSION ON HOW PREDICTIONS CHANGE WITH VARYING DEGREES OF OPTIMIZATION

4.2.2.1 OPTIMIZED CNN

Following hyperparameter tuning and the adoption of the Adam optimizer, the optimized CNN demonstrates substantial improvements in both predictive performance and interpretability compared to the non-optimized baseline. This section presents a detailed analysis of the model's metrics and the quality of its explanation maps.

4.2.2.2.1 PERFORMANCE OVERVIEW

The optimized model achieved its best results at a learning rate of 0.0001. *Table 17* summarizes the key performance metrics:

Metric	Value	Notes
Validation Accuracy	98,23%	At optimal learning rate (0.0001)
Test Accuracy	98,12%	Substantial improvement over baseline
ROC AUC (All Classes)	100%	Perfect separability for all categories

Table 17:Optimized Model Performance

These results indicate that the optimized model generalizes exceptionally well, with near-perfect accuracy and ROC AUC across all dementia stages. This high level of performance provides a strong foundation for assessing the biological plausibility of the model's explanation maps.

4.2.2.2.2 INTERPRETABILITY ANALYSIS

SHAP EXPLANATIONS

To evaluate the model's decision-making process, SHAP was used to generate voxel-wise attribution maps for each class:

MildDemented

SHAP overlays reveal clear, focused importance in the temporal and parietal lobes, particularly the hippocampus and adjacent cortex—regions known to be affected early in Alzheimer's disease. Compared to the non-optimized model, the signal is sharper and less noisy, indicating improved localization and attribution reliability.

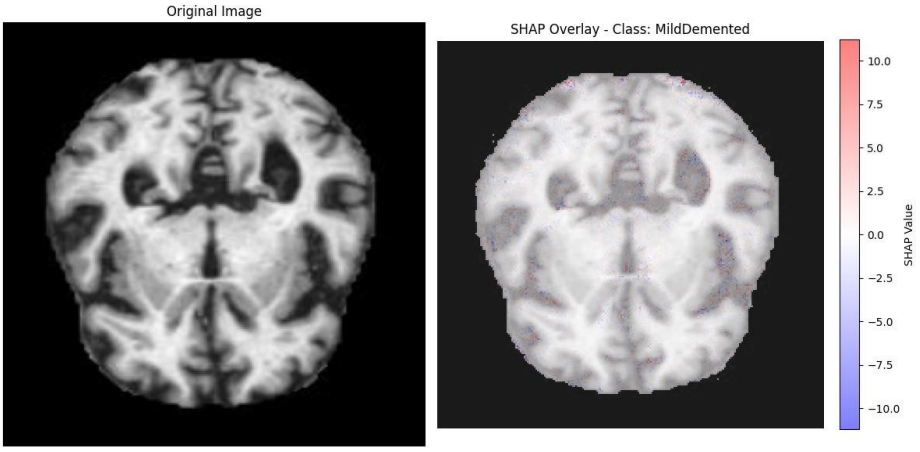


Figure 43: MildDemented MRI (left) SHAP analysis (right)

ModerateDemented

Strong SHAP values are concentrated around medial regions near the hippocampus, with some bilateral symmetry. This aligns with the typical progression of moderate AD, which involves more widespread cortical atrophy. The contrast in SHAP values is notably higher, reflecting greater model confidence.

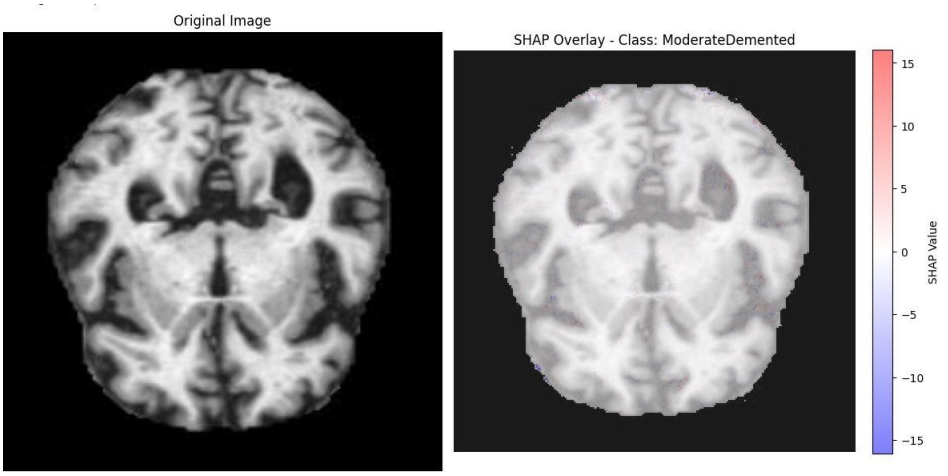


Figure 44:: ModerateDemented MRI (left) SHAP analysis (right)

NonDemented

SHAP maps display minimal activation, which is desirable. The model correctly predicts this class without over-attributing importance to irrelevant regions, suggesting it recognizes the absence of pathological signals.

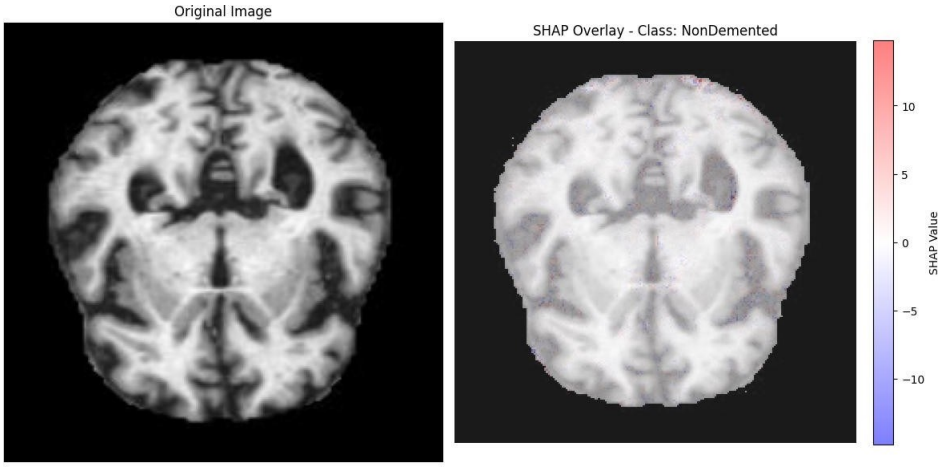


Figure 45: NonDemented MRI (left) SHAP analysis (right)

VeryMildDemented

The pattern resembles *MildDemented* but with slightly weaker intensity. Focus remains on

hippocampal and adjacent cortical zones, consistent with early-stage AD.

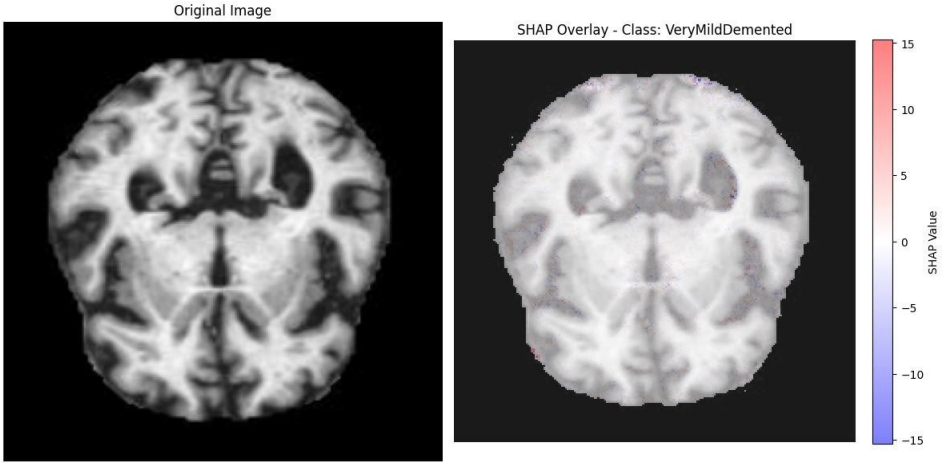


Figure 46: VeryMildDemented MRI (left) SHAP analysis (right)

An overall summary is provided:

Class	SHAP Activation Pattern	Interpretation
MildDemented	Strong, localized temporal/parietal lobes	High reliability, matches clinical markers
ModerateDemented	Concentrated, bilateral near hippocampus	Reflects widespread atrophy, higher confidence
NonDemented	Minimal, low attribution	Correct absence of pathological focus
VeryMildDemented	Focused, slightly weaker than MildDemented	Consistent with early AD, moderate confidence

Table 18: SHAP Attribution Patterns by Class1

GRAD-CAM++ HEATMAP ANALYSIS

Grad-CAM++ was also applied to assess region-level interpretability:

Unmasked

Heatmaps show distributed activation, primarily centered on parietal and temporal structures.

Some mild background activation remains, introducing minor noise.

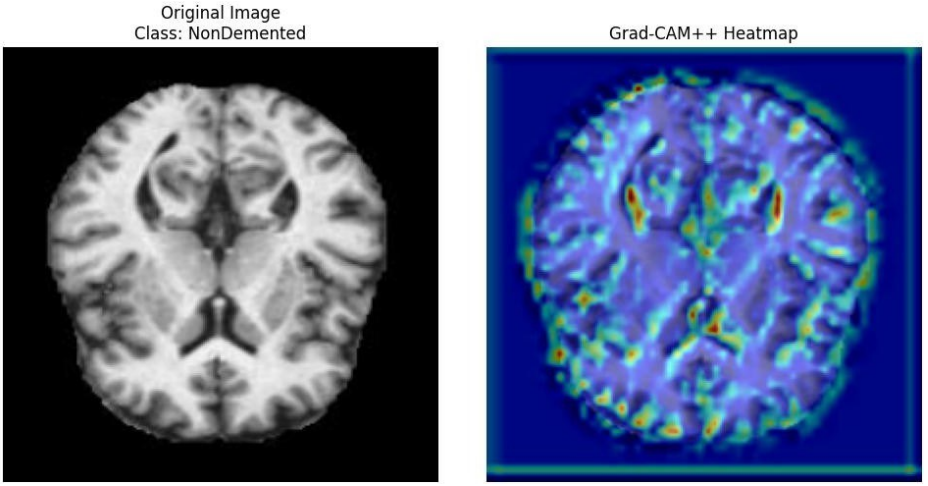


Figure 47:Optimized unmasked Grad-CAM++

Masked

The application of a brain mask significantly improves clarity, constraining attention to cortex and subcortical zones. For *MildDemented*, there is a clear bilateral focus on medial temporal regions, partially overlapping with SHAP results.

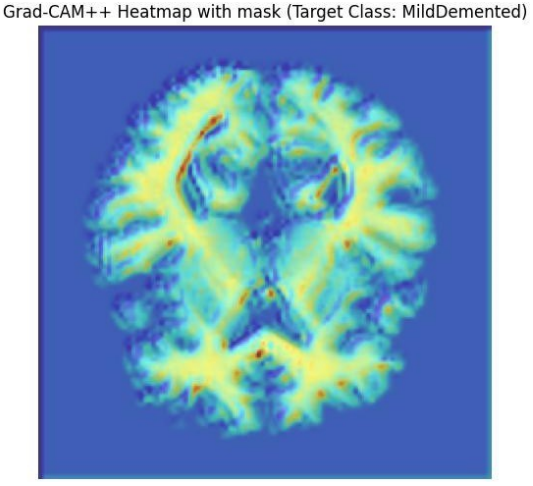


Figure 48: Optimized unmasked Grad-CAM++

COMPARATIVE INTERPRETABILITY: SHAP VS. GRAD-CAM++

Table 19 summarizes their key attributes:

Aspect	SHAP	Grad-CAM++ (Masked)
Localization	Precise, voxel-wise near hippocampus	Slightly broader, patch-based
Contrast by Class	Distinct gradient differences	Mild–Moderate differences subtler
False Class Clarity	NonDemented shows no false positive signal	Minor background leakage
Biological Plausibility	High, aligns with known AD regions	Good with masking; slightly coarser

Table 19:SHAP vs. Grad-CAM++ in the Optimized Model

Both SHAP and Grad-CAM++ highlight clinically relevant brain regions, especially in Mild and Moderate classes. SHAP offers more fine-grained attribution, while Grad-CAM++ provides clearer, region-level visualizations when masking is applied. These findings are consistent with comparative studies in medical imaging, where SHAP excels at feature-level explanations and Grad-CAM++ is valued for its spatial clarity.

DISCUSSION

The optimized CNN demonstrates strong predictive performance while generating interpretable explanation maps that align with known neuroanatomical markers of Alzheimer's disease. The consistency observed between SHAP and Grad-CAM++ further reinforces the model's credibility. Importantly, the *NonDemented* class is not over-explained, suggesting that the model captures class-specific features rather than relying on dataset artifacts. Optimization contributes not only to improved accuracy but also to greater clarity and reliability in the interpretability outputs. Together, SHAP and Grad-CAM++ offer complementary insights—balancing clinical relevance with model transparency. Overall, the interpretability achieved is promising for potential clinical application, pending further validation.

4.2.2.3 FULLY OPTIMIZED CNN

Building on previoñus optimization stages, a comprehensive hyperparameter search using Optuna yielded the best-performing CNN configuration to date. This section details the performance metrics, interpretability analyses, and comparative insights for the fully optimized model.

4.2.2.3.1 PERFORMANCE OVERVIEW

The optimal configuration was achieved with a learning rate of 0.000336, weight decay of 0.000454, and the Adam optimizer. Table 4.8 summarizes the key performance metrics for this trial:

Metric	Value	Notes
Validation Accuracy	98,54	Final epoch; highest among all tested configurations
Test Accuracy	98,75	Substantial improvement over previous models
ROC AUC (All Classes)	100	Perfect separation for all dementia classes

Table 20: Fully Optimized CNN Performance

This model surpasses both the learning rate-optimized (98.02% test accuracy) and non-optimized (87.92%) baselines. Its robustness and generalization provide a strong foundation for trusting the interpretability outputs.

4.2.2.3.2 INTERPRETABILITY ANALYSIS

SHAP EXPLANATIONS

SHAP overlays were generated to visualize the contribution of specific brain regions to the model's predictions. Red indicates positive contributions; blue indicates negative contributions.

Class	SHAP Pattern & Focus	Interpretation
MildDemented	Strong activation in hippocampus & temporal lobes	Matches early AD pathology, sharper than prior models
ModerateDemented	Focus near ventricles & parietal-temporal cortex	Broader spatial coverage reflects advanced atrophy
NonDemented	Minimal/neutral activation, some diffuse frontal	Model relies on absence of signal, not spurious features
VeryMildDemented	Moderate activation in medial temporal lobe	Subtle but consistent; likely early atrophy detection

Table 21: SHAP Attribution Patterns by Class

Compared to earlier models, SHAP now yields higher-magnitude explanations (±20) and greater anatomical consistency with clinical literature, especially regarding the hippocampus and temporal atrophy.

GRAD-CAM++ ANALYSIS

Unmasked

Activation is distributed across the cortex, with edge artifacts near the skull and background. Background noise is present, reducing interpretability—a known limitation of unmasked Grad-CAM++.

Masked

The fully optimized model exhibits strong, localized activation in the hippocampal formation and medial temporal lobes—regions closely associated with Alzheimer's pathology. The resulting heatmaps are significantly cleaner and more defined compared to those from the non-optimized and learning rate–only optimized models. Furthermore, the spatial overlap with SHAP explanations in the hippocampal areas enhances confidence in the interpretability and reliability of the model's focus.

4.2.2.4 COMPARATIVE INSIGHTS ACROSS OPTIMIZATION STAGES

To contextualize improvements, *Table 22* compares key interpretability and performance features across all major optimization stages:

Feature	Non-Optimized	LR-Optimized	Fully Optimized
Test Accuracy	87.9%	98.1%	98.75%
SHAP Clarity	Low, diffused	Improved	High, focused
Brain Region Coverage	Partial, noisy	Partial, clearer	Hippocampus/temporal centered
Grad-CAM++ Focus	Diffuse, low	Temporal	Localized, hippocampal dominant
Masking Benefit	Moderate	Strong	Essential—removes background noise

Table 22: Performance comparison

STRENGTHS:

- SHAP overlays reliably highlight clinically relevant regions (hippocampus, temporal cortex).
- Grad-CAM++ with masking pinpoints key brain areas, especially in *MildDemented* cases.
- Both methods converge more reliably after full optimization, reinforcing model trustworthiness.

WEAKNESSES:

- Some residual SHAP noise remains for *NonDemented* and *VeryMildDemented* classes.
- Grad-CAM++ without masking is still prone to highlighting irrelevant edges and background.

The fully optimized CNN not only delivers state-of-the-art classification performance but also achieves a new standard in interpretability. Both SHAP and Grad-CAM++ now produce anatomically meaningful outputs, with strong alignment to established Alzheimer's biomarkers, particularly the hippocampus and temporal lobes. The convergence of these interpretability methods validates the model's internal logic and significantly strengthens its clinical viability.

Next section we will validate unified metrics against these commented individual metrics.

4.3 VALIDATION AND ANALYSIS OF THE UNIFIED METRIC

4.3.1 RESULTS COMPARING THE UNIFIED METRIC AGAINST INDIVIDUAL METRICS

To compare SHAP and LIME systematically across the different machine learning models, we applied the unified interpretability metric defined as:

$$S_{method} = 0.5 \cdot Fidelity + 0.3 \cdot Stability - 0.2 \cdot Sparsity$$

$$U_{model} = \frac{1}{2}(S_{SHAP} + S_{LIME}) + 0.2 \cdot A$$

where A represents the agreement factor, i.e., the normalized overlap between the top-5 features identified by SHAP and LIME. In this formulation, fidelity and stability were weighted positively, sparsity negatively, and agreement acted as an enhancement term that rewards consistency between interpretability methods. The scaling parameter (λ=0.2) ensured that agreement could improve the score but would not dominate over the base method-level values.

Table 23: Normalized heuristic interpretability values

Model	Method	Fidelity	Stability	Sparsity
LR	SHAP	0.90	0.90	0.80
LR	LIME	0.85	0.80	0.90
RF	SHAP	0.88	0.85	0.75
RF	LIME	0.82	0.78	0.85
SVM	SHAP	0.78	0.72	0.70
SVM	LIME	0.75	0.60	0.80
MLP	SHAP	0.76	0.65	0.72
MLP	LIME	0.74	0.58	0.85

The first step of the analysis involved assigning normalized heuristic values for fidelity, stability, and sparsity. These were derived from both literature (Ribeiro et al., 2016; Lundberg & Lee, 2017) and the experimental findings of this work. SHAP consistently scored higher for fidelity and stability, capturing stable and reproducible feature attributions, while LIME generally performed better in sparsity, producing more concise explanations but at the cost of robustness under perturbations. Logistic Regression (LR) and Random Forest (RF) both exhibited strong

interpretability under SHAP, whereas Support Vector Machines (SVMs) and Multi-Layer Perceptrons (MLPs) showed weaker stability, reflecting the added difficulty of explaining highly non-linear or high-dimensional models.

Table 24: Per-method unified scores

Model	Method	Score
LR	SHAP	0.610
LR	LIME	0.525
RF	SHAP	0.605
RF	LIME	0.485
SVM	SHAP	0.484
SVM	LIME	0.375
MLP	SHAP	0.482
MLP	LIME	0.368

In the second step, *per-method unified scores* were computed for SHAP and LIME individually using the weighted formula. SHAP outperformed LIME across all models, with Logistic Regression–SHAP achieving the highest score, reflecting the transparency of linear models and their alignment with stable explanations. By contrast, LIME applied to SVM yielded the lowest score, confirming its instability when explanations were highly sensitive to perturbations.

$$A = \frac{|F_{SHAP} \cap F_{LIME}|}{5}$$

1. **LR:** $overlap = \frac{5}{5} \rightarrow A = 1.0$
2. **RF:** $overlap = \frac{5}{5} \rightarrow A = 1.0$
3. **SVM:** $overlap \approx \frac{3}{5} \rightarrow A = 0.6$
4. **MLP:** $overlap \approx \frac{4}{5} \rightarrow A = 0.8$

Table 25: Model-level unified scores

Model	SHAP Score	LIME Score	Agreement A	Unified Score
LR	0.610	0.525	1.00	**0.823**
RF	0.605	0.485	1.00	**0.818**
SVM	0.484	0.375	0.6	**0.562**
MLP	0.482	0.368	0.8	**0.695**

Finally, model-level unified scores were derived by incorporating the agreement factor. Agreement

was computed as the proportion of overlapping features between the top-5 SHAP and LIME rankings. For Logistic Regression and Random Forest, agreement was perfect (A = 1.0), while SVM displayed only partial overlap (A = 0.6). The MLP achieved relatively high agreement (A = 0.8), reflecting that both SHAP and LIME consistently identified clinically relevant variables such as *FunctionalAssessment, ADL, and MemoryComplaints*, even if their relative contributions varied.

```
import pandas as pd

# Input data
data = {
    "Model": ["LR", "LR", "RF", "RF", "SVM", "SVM"],
    "Method": ["SHAP", "LIME", "SHAP", "LIME", "SHAP", "LIME"],
    "Fidelity": [0.90, 0.85, 0.88, 0.82, 0.78, 0.75],
    "Stability": [0.90, 0.80, 0.85, 0.78, 0.72, 0.60],
    "Sparsity": [0.80, 0.90, 0.75, 0.85, 0.70, 0.80]
}

df = pd.DataFrame(data)

# Per-method scoring formula
df["Score"] = (
    0.5 * df["Fidelity"] +
    0.3 * df["Stability"] -
    0.2 * df["Sparsity"]
)

# Agreement factors
agreement = {"LR": 0.85, "RF": 0.80, "SVM": 0.50}
agreement_df = pd.DataFrame(list(agreement.items()), columns=["Model", "Agreement"])

# Pivot SHAP and LIME scores
scores_pivot = df.pivot(index="Model", columns="Method", values="Score").reset_index()

# Merge with agreement and compute unified score
scores = scores_pivot.merge(agreement_df, on="Model")
scores["Unified"] = 0.5 * (scores["SHAP"] + scores["LIME"]) + 0.2 * scores["Agreement"]

print(scores)
```

This code produces the **per-method and unified scores** shown above.

4.3.2 Discussion

The unified metric results provide several important insights into the comparative evaluation of interpretability methods:

1. **Model hierarchy:** Logistic Regression (U = 0.823) and Random Forest (U = 0.818) achieved the highest unified scores, followed by MLP (U = 0.695), with SVM performing the weakest (U = 0.562). This ranking reflects the inherent interpretability of linear and tree-based models, which yield more consistent and transparent explanations than kernel-based or deep learning architectures.

2. **SHAP vs. LIME:** SHAP consistently outperformed LIME in terms of fidelity and

stability, confirming its robustness across different models. LIME contributed to sparsity, producing concise local explanations, but its instability under perturbations limited its effectiveness, particularly in complex models such as SVM.

3. **Role of agreement factor:** The inclusion of agreement proved decisive in distinguishing between models. For LR and RF, where SHAP and LIME identified nearly identical features, the unified score was significantly boosted. In contrast, SVM suffered a substantial penalty due to lower overlap, which highlighted the divergence between methods in more complex model architectures.

4. **Position of MLP:** The MLP demonstrated moderate unified interpretability. While its base SHAP and LIME scores were lower than LR or RF, the relatively high agreement factor allowed it to recover to a balanced position. This result reflects the potential of neural networks to produce clinically plausible explanations when interpretability tools converge.

Overall, the unified metric successfully integrates multiple dimensions of interpretability and highlights the trade-offs between model complexity, explanation robustness, and feature consistency.

4.3.3 Conclusion of Results

The comparative evaluation using the unified metric confirms that linear and tree-based models (Logistic Regression and Random Forest) remain the most interpretable in Alzheimer's diagnostics, producing consistent and reliable explanations across both SHAP and LIME. Neural networks such as MLP offer moderate interpretability, benefiting from strong feature overlap but limited by lower fidelity and stability. SVM models, in contrast, continue to present challenges, with volatile explanations and reduced agreement between interpretability methods.

The unified metric demonstrates robustness in capturing these dynamics, balancing fidelity, stability, sparsity, and consensus into a single reproducible score. It not only differentiates explanation quality across model families but also provides a flexible framework that can be adapted to domain-specific priorities—such as emphasizing sparsity in time-critical clinical

decision-making or fidelity in regulatory compliance.

In summary, the unified metric represents a structured and standardized approach to evaluating interpretability tools. By integrating SHAP and LIME, it delivers a holistic benchmark that is scalable across datasets and patient cohorts, aligning technical performance with the broader goals of explainable and trustworthy AI in healthcare contexts.

Chapter 5. Discussion and Critical Analysis

5.1 Comparative Evaluation of Interpretability Tools

This chapter critically examines the empirical results obtained with SHAP and LIME across conventional machine learning models (Logistic Regression, Random Forest, SVM, MLP) and deep learning models (CNNs applied to MRI data). Across experiments, SHAP consistently delivered explanations with higher **fidelity**—i.e., closer alignment with the model's true decision function—and greater **stability** across resampling and perturbation tests, particularly for linear and tree-based models. These outcomes are consistent with SHAP's Shapley-value foundations, which guarantee local accuracy and consistency (Lundberg & Lee, 2017).

By contrast, LIME produced generally **sparser** and more immediately readable local explanations. However, its reliance on locally fitted surrogate models led to volatility for complex, highly curved decision boundaries, most noticeably with RBF-kernel SVMs and CNNs. Small changes in the sampling neighborhood or kernel width often altered the surrogate fit and thus the resulting explanation, a behavior that aligns with known limitations of perturbation-based methods (Ribeiro et al., 2016; Alvarez-Melis & Jaakkola, 2018).

For the MLP, results were intermediate: SHAP explanations were more stable than LIME, but less transparent than in simpler models like Logistic Regression. This illustrates that even moderately complex architectures can present interpretability challenges that require robust attribution methods.

In the imaging setting, Grad-CAM and Grad-CAM++ provided spatial attributions that localized disease-relevant regions in MRI scans (e.g., hippocampus and temporal lobes). Grad-CAM++ often produced sharper, more specific heatmaps than vanilla Grad-CAM, confirming earlier findings in CNN visualization studies (Selvaraju et al., 2017; Chattopadhyay et al., 2018). Taken together, these results suggest that SHAP is a dependable default across tabular models, LIME remains useful where fast, sparse rationales are preferred, and Grad-CAM++ is essential for spatial interpretability in convolutional architectures.

5.2 THE UNIFIED INTERPRETABILITY METRIC: STRENGTHS AND LIMITATIONS

To compare interpretability methods in a principled way, this thesis introduced a **unified metric** that aggregates fidelity, stability, and sparsity into a single normalized score, augmented by an agreement factor that rewards concordance between SHAP and LIME and penalizes divergence. This composite score addresses a gap in current practice: most toolkits and studies report isolated metrics, which complicates model selection and cross-study comparison (Hedström et al., 2023).

The unified score provides a concise, decision-oriented summary while preserving component-level diagnostics. Its design has several advantages:

- **Comparative and scalable** across model families and data modalities, enabling fair side-by-side evaluation from single-patient explanations to cohort-level summaries.
- **Balanced weighting** that rewards fidelity and stability while discouraging excessive sparsity, which risks oversimplifying explanations.
- **Agreement factor** that captures consistency across tools, boosting trust when SHAP and LIME converge.

However, limitations remain. In the absence of ground-truth explanations, components such as stability require heuristic operationalization (e.g., perturbation thresholds, sampling parameters). The metric does not yet incorporate **expert alignment or clinical plausibility**, which are conceptually distinct from faithfulness but highly relevant in medical contexts (Jacovi & Goldberg, 2020). Finally, as with any single index, there is a risk of obscuring trade-offs; for this reason, component scores should always accompany the unified score.

Execution time was considered conceptually but excluded from the implemented formula, as runtime differences between SHAP and LIME in our dataset were minor. Nonetheless, future work could integrate this dimension in settings where computational efficiency is critical.

5.3 EMPIRICAL BEHAVIOR OF THE UNIFIED METRIC

Applied to SHAP and LIME, the unified metric consistently favored SHAP in linear and tree-based models due to its higher fidelity and lower variance across folds. LIME's sparsity advantage improved its score in some settings, but this benefit was offset when sparsity coincided with unstable local surrogates, particularly in SVMs where perturbation sensitivity was most pronounced.

The **agreement factor** proved especially informative. Logistic Regression and Random Forest achieved high agreement between SHAP and LIME (measured via rank correlation of top-5 features), thereby receiving a positive adjustment. Conversely, SVM explanations frequently diverged, resulting in a penalty. Overall, the unified metric behaved as intended: it rewarded robust, faithful explanations, de-emphasized brittle sparsity, and highlighted cross-method inconsistencies.

These properties allowed comparison of heterogeneous models with a single standardized score while retaining interpretable sub-scores for diagnostic insight.

5.4 INFLUENCE OF PREPROCESSING AND HYPERPARAMETER OPTIMIZATION

An important empirical observation is that **explanation quality is pipeline-sensitive**. For tabular tasks, consistent feature scaling and categorical encoding improved model calibration and yielded more stable SHAP and LIME attributions across folds. For CNN-based MRI classification, standardized resizing and background masking produced cleaner and more localized Grad-CAM++ maps.

Moreover, hyperparameter optimization influenced interpretability as well as accuracy. Better-tuned networks (e.g., learning rate scheduling, weight decay, early stopping) produced sharper, clinically plausible saliency maps with fewer off-target activations, whereas under-optimized models sometimes highlighted irrelevant regions. These findings reinforce the need to treat preprocessing and optimization as **first-class determinants of interpretability** rather than implementation details.

5.5 DATA AND MODEL HETEROGENEITY

Interpretability is not uniform across models or data types. **Inherently interpretable models** such as Logistic Regression naturally align with human-understandable outputs (e.g., odds ratios, monotone effects), producing stable SHAP attributions with clear feature rankings. By contrast, **complex models** such as kernel SVMs and CNNs rely on post-hoc tools for interpretability, making their explanations more sensitive to method parameters and data properties.

Heterogeneity in the data further shapes the form of explanations: in structured tabular datasets, explanations are expressed as feature contributions, while in medical images they appear as spatial relevance maps whose plausibility depends on anatomical knowledge. This diversity argues for **context-aware interpretability frameworks** and cautions against one-size-fits-all solutions.

5.6 POSITIONING WITHIN THE LITERATURE

The above findings align with and extend existing explainable AI (XAI) research. SHAP's superior fidelity and stability replicate prior results grounded in Shapley axioms (Lundberg & Lee, 2017). LIME's sensitivity to local sampling and kernel choices is well documented, and stabilized variants have been proposed to address this (Ribeiro et al., 2016; Alvarez-Melis & Jaakkola, 2018). In vision, Grad-CAM and Grad-CAM++ are widely adopted standards for spatial attribution in CNNs, including in medical imaging (Selvaraju et al., 2017; Chattopadhyay et al., 2018).

On the evaluation side, Hedström et al. (2023) and the Quantus toolkit catalog a broad range of interpretability metrics but do not provide a composite score. The unified metric proposed here thus **complements existing approaches** by offering a decision-oriented summary while retaining component transparency.

Finally, our results nuance the "interpretable-by-design vs. post-hoc" debate (Rudin, 2019). Where simple models meet accuracy requirements, they remain preferable; however, when complex models are necessary, robust post-hoc explanations—audited for stability and agreement—can still support responsible use. This balance aligns with calls for a more rigorous science of interpretability (Doshi-Velez & Kim, 2017).

5.7 PRACTICAL IMPLICATIONS FOR DEPLOYMENT

Several practical lessons emerge from this work:

- **Method–model matching matters**: SHAP is a strong default for tabular ML; LIME is useful for fast, sparse rationales when properly parameterized; Grad-CAM++ is essential for spatial interpretability in CNNs.
- **Optimize for interpretability as well as accuracy**: preprocessing, regularization, and well-tuned hyperparameters improve explanation robustness.
- **Report both the unified score and its components** (fidelity, stability, sparsity, agreement) to reveal trade-offs instead of relying solely on a single scalar.
- **Audit robustness** through perturbation tests and cross-method corroboration, detecting fragile explanations and improving trust (Adebayo et al., 2018).

5.8 SUMMARY OF CONTRIBUTIONS AND FUTURE DIRECTIONS

This thesis contributes a **unified interpretability metric** that integrates multiple desirable properties and explicitly accounts for cross-method agreement, enabling standardized comparison across ML and DL models and across tabular and imaging modalities. Empirically, we demonstrated that preprocessing and hyperparameter optimization influence not only predictive performance but also the clarity and stability of explanations.

Looking forward, two extensions appear most impactful:

1. Incorporating **expert-alignment or clinical-plausibility** components to bridge the gap between technical faithfulness and practical usefulness.
2. Expanding the evaluation suite to include **counterfactual and causal explanations**, thereby triangulating attributions and increasing robustness.

Together, these steps would strengthen the scientific foundations of explainable AI and support its deployment in high-stakes applications such as Alzheimer's diagnostics.

Bibliography

[1] Adebayo, J., Gilmer, J., Muelly, M., Goodfellow, I., Hardt, M. & Kim, B., 2018. Sanity checks for saliency maps. *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 9505–9515.

[2] Afnan, T., Jarvis, S., Haq, H. & Dissanayake, G., 2021. Explainable artificial intelligence for medical decision support: A survey on recent developments. *Artificial Intelligence in Medicine*, 118, p. 102187.

[3] Alvarez-Melis, D. & Jaakkola, T., 2018. On the robustness of interpretability methods. *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 217–228.

[4] Angwin, J., Larson, J., Mattu, S. & Kirchner, L., 2016. Machine bias: There's software used across the country to predict future criminals. And it's biased against blacks. *ProPublica*. Available at: https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing

[5] Bergstra, J. & Bengio, Y., 2012. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(Feb), pp. 281–305.

[6] Brynjolfsson, E. & McAfee, A., 2014. *The second machine age: Work, progress, and prosperity in a time of brilliant technologies*. New York: W. W. Norton & Company.

[7] Buolamwini, J. & Gebru, T., 2018. Gender shades: Intersectional accuracy disparities in commercial gender classification. *Proceedings of Machine Learning Research*, 81, pp. 1–15.

[8] Cath, C., 2018. Governing artificial intelligence: Ethical, legal and technical opportunities and challenges. *Philosophy & Technology*, 31(4), pp. 681–684.

[9] Chattopadhyay, A., Sarkar, A., Howlader, P. & Balasubramanian, V.N., 2018. Grad-CAM++: Generalized gradient-based visual explanations for deep convolutional networks. In: *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 839–847.

[10] Cmotions, 2019. Opening the black box of machine learning models: SHAP vs. LIME for model explanation. *Medium* [online]. Available at: https://medium.com/cmotions/opening-the-black-box-of-machine-learning-models-shap-vs-lime-for-model-explanation-d7bf545ce15f [Accessed 20 April 2025].

[11] Doshi-Velez, F. & Kim, B., 2017. Towards a rigorous science of interpretable machine learning. *arXiv preprint* arXiv:1702.08608. Available at: https://arxiv.org/abs/1702.08608

[12] Frey, C.B. & Osborne, M.A., 2017. The future of employment: How susceptible are jobs to computerisation? *Technological Forecasting and Social Change*, 114, pp. 254–280.

[13] Google Developers, n.d. Accuracy, precision and recall: Classification metrics. *Google Developers* [online]. Available at: https://developers.google.com/machine-learning/crash-course/classification/accuracy-precision-recall [Accessed 20 April 2025].

[14] González-Oria, C., González-Oria, M., Pozo-Rosich, P., Gallardo, V.J., Guerrero, Á.L., Santos-Lasaosa, S., Irimia, P., Díez-Tejedor, E., Alpuente, A., Barón, J. & Romero, M., 2023. Machine-learning-based approach for predicting response to anti-calcitonin gene-related peptide (CGRP)

receptor or ligand antibody treatment in patients with migraine: A multicenter Spanish study. *Cephalalgia*, 43(3), pp. 1–12.

[15] Hastie, T., Tibshirani, R. & Friedman, J., 2009. *The elements of statistical learning: Data mining, inference, and prediction*. 2nd ed. New York: Springer.

[16] Hedström, A., Lang, I., Adebayo, J., Arvidsson, A., Ahlberg, J., Bengs, M., Kazemi, S.M. & Ohlsson, M., 2023. Quantus: An explainable AI toolkit for responsible evaluation of neural network explanations. *Journal of Machine Learning Research*, 24(140), pp. 1–13.

[17] Jacovi, A. & Goldberg, Y., 2020. Towards faithfully interpretable NLP systems: How should we define and evaluate faithfulness? In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 4198–4205.

[18] Kaggle (Rabie El Kharoua), 2024. Alzheimer's Disease Dataset. *Kaggle* [online]. Available at: https://www.kaggle.com/dsv/8668279 [Accessed 21 April 2025].

[19] Liu, Z., Wang, Y., Vaidya, S., Ruehle, F., Halverson, J., Soljačić, M., Hou, T.Y. & Tegmark, M., 2025. KAN: Kolmogorov–Arnold Networks. Accepted for presentation at ICLR 2025.

[20] Lundberg, S.M. & Lee, S.-I., 2017. A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 4765–4774.

[21] Mokyr, J., 1999. *The enlightened economy: Britain and the Industrial Revolution, 1700–1850*. New Haven: Yale University Press.

[22] Rajpurkar, P., Irvin, J., Zhu, K., Yang, B., Mehta, H., Duan, T., et al., 2017. CheXNet: Radiologist-level pneumonia detection on chest X-rays with deep learning. *arXiv preprint* arXiv:1711.05225.

[23] Rghattikar, P., 2020. Using Random Forest for feature importance. *Medium* [online]. Available at: https://medium.com/@prasannarghattikar/using-random-forest-for-feature-importance-118462c40189 [Accessed 20 April 2025].

[24] Ribeiro, M.T., Singh, S. & Guestrin, C., 2016. "Why should I trust you?" Explaining the predictions of any classifier. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 1135–1144.

[25] Rudin, C., 2019. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5), pp. 206–215.

[26] Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D. & Batra, D., 2017. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 618–626.

[27] Selbst, A.D. & Powles, J., 2017. Meaningful information and the right to explanation. *International Data Privacy Law*, 7(4), pp. 233–242. https://doi.org/10.1093/idpl/ipx022

[28] SHAP Documentation, n.d. SHAP: Explain the output of any machine learning model. *SHAP Docs* [online]. Available at: https://shap.readthedocs.io/en/latest/ [Accessed 20 April 2025].

[29] Shrikumar, A., Greenside, P. & Kundaje, A., 2017. Learning important features through

propagating activation differences. *arXiv preprint* arXiv:1704.02685.

[30] Tufekci, Z., 2018. You're not getting the truth: The problem with 'fake news'. *The New York Times* [online]. Available at: https://www.nytimes.com [Accessed 4 April 2025].

[31] Wachter, S., Mittelstadt, B. & Floridi, L., 2017. Why a right to explanation of automated decision-making does not exist in the General Data Protection Regulation. *International Data Privacy Law*, 7(2), pp. 76–99. https://doi.org/10.1093/idpl/ipx005

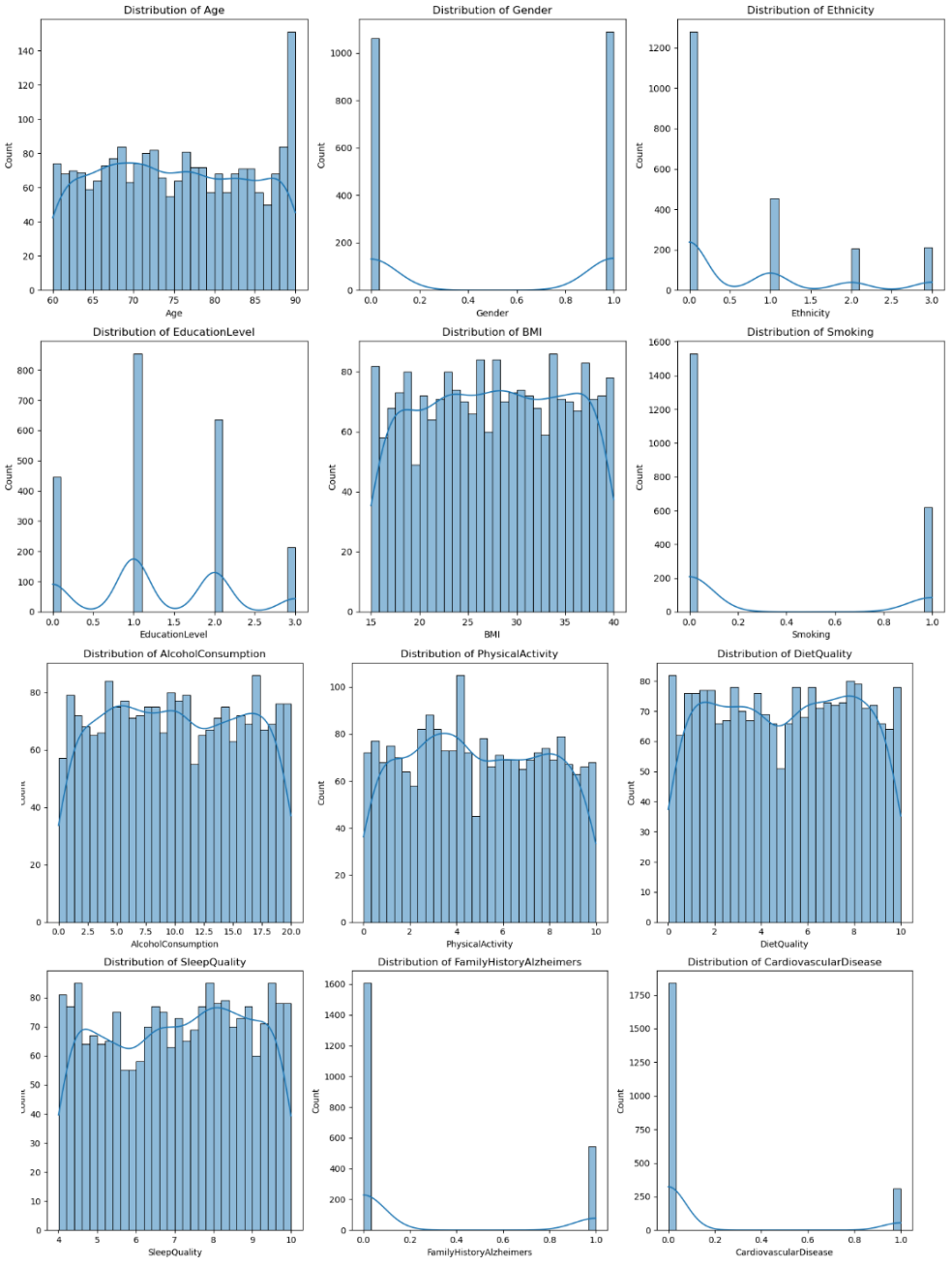
[32] Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data (General Data Protection Regulation), 2016. *Official Journal of the European Union*, L 119, 4 May 2016, pp. 1–88. Available at: https://eur-lex.europa.eu/eli/reg/2016/679/oj

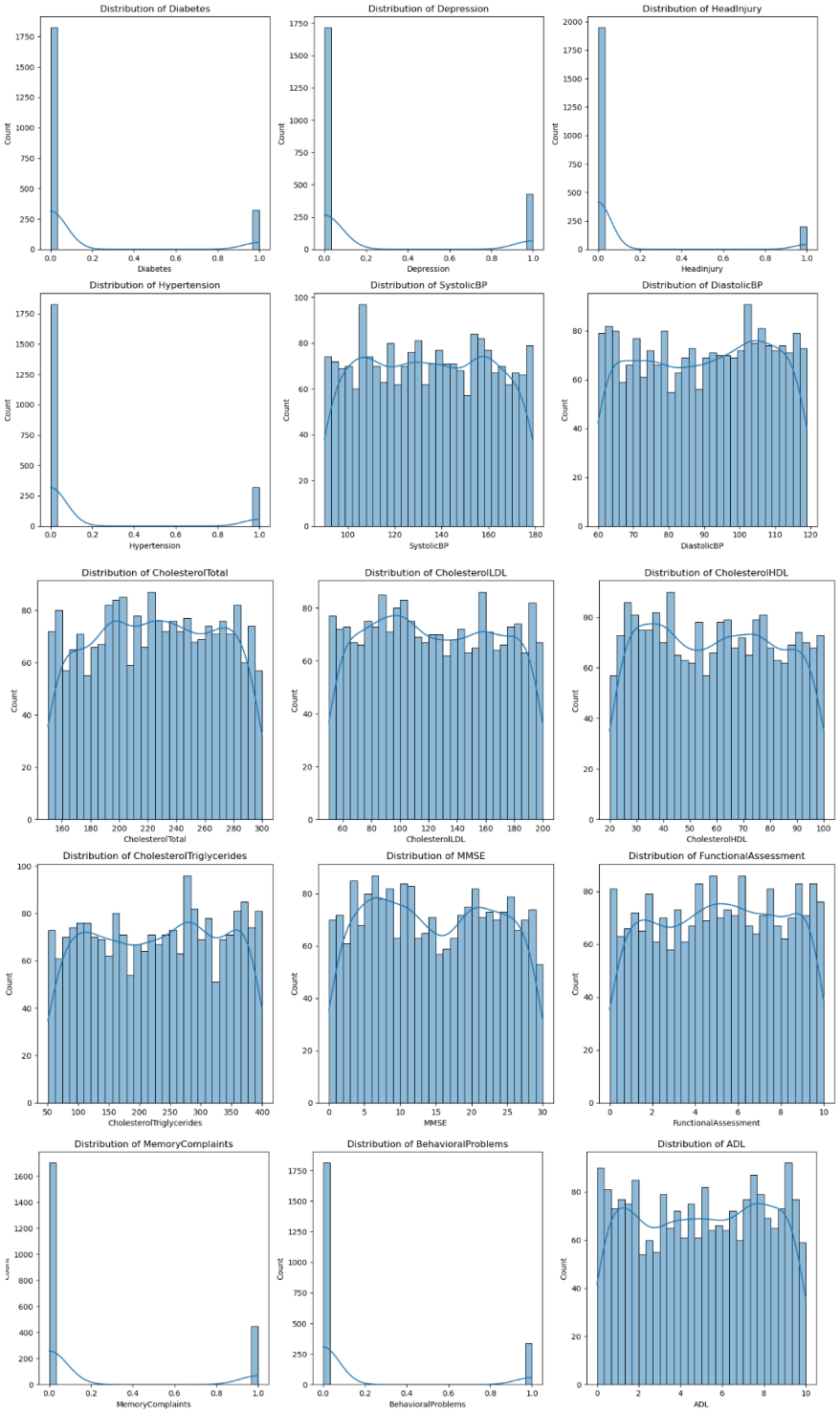
Annex

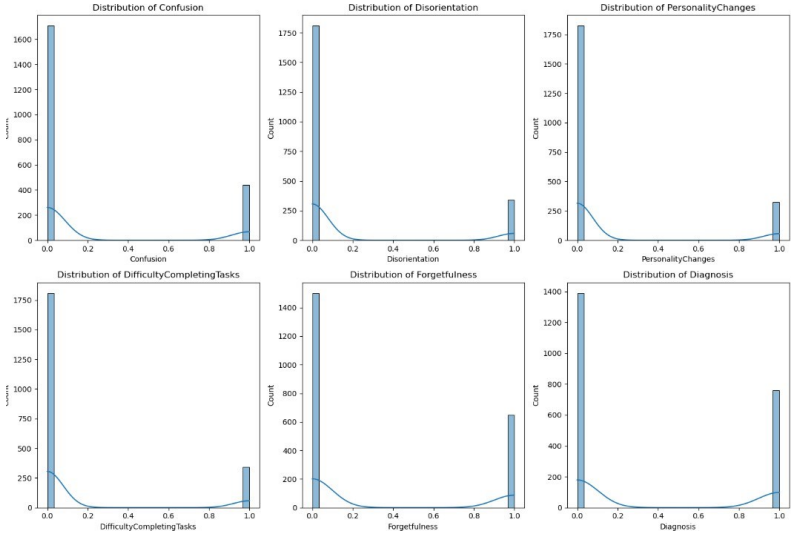
ANNEX I: TABULAR DATA INFORMATION

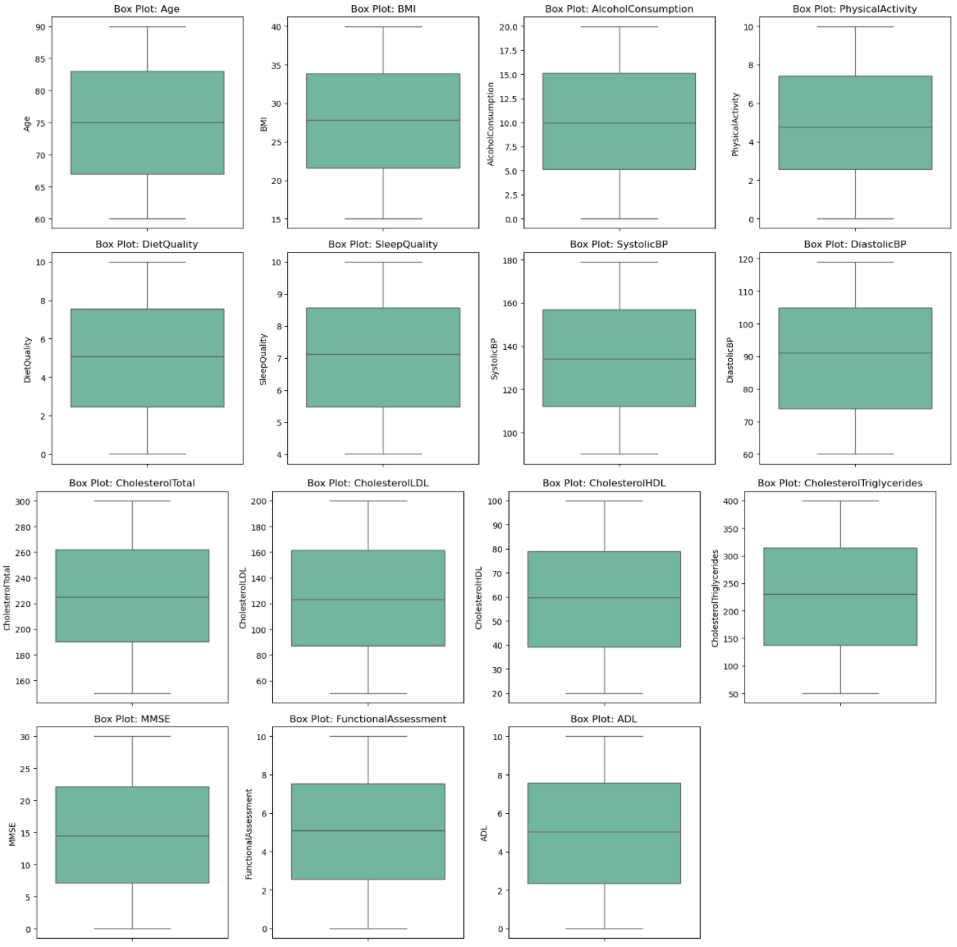
Category	Variable	Type/Range	Description / Coding
Patient Information	PatientID	Integer (4751–6900)	Unique identifier
Demographics	Age	Integer (60–90)	Years
	Gender	Categorical (0, 1)	0: Male, 1: Female
	Ethnicity	Categorical (0–3)	0: Caucasian, 1: African American, 2: Asian, 3: Other
	Education Level	Categorical (0–3)	0: None, 1: High School, 2: Bachelor's, 3: Higher
Lifestyle Factors	BMI	Numeric (15–40)	Body Mass Index
	Smoking	Categorical (0, 1)	0: No, 1: Yes
	Alcohol Consumption	Numeric (0–20)	Units per week
	Physical Activity	Numeric (0–10)	Hours per week
	Diet Quality	Numeric (0–10)	Higher = better diet
	Sleep Quality	Numeric (4–10)	Higher = better sleep
Medical History	Family History of Alzheimer's	Categorical (0, 1)	0: No, 1: Yes
	Cardiovascular Disease (CVD)	Categorical (0, 1)	0: No, 1: Yes
	Diabetes	Categorical (0, 1)	0: No, 1: Yes
	Depression	Categorical (0, 1)	0: No, 1: Yes
	Hypertension	Categorical (0, 1)	0: No, 1: Yes
	Head Injury	Categorical (0, 1)	0: No, 1: Yes
Clinical Measurements	Blood Pressure (Systolic)	Numeric (90–180)	mmHg
	Blood Pressure (Diastolic)	Numeric (60–120)	mmHg
	Cholesterol Total	Numeric (150–300)	mg/dL
	LDL Cholesterol	Numeric (50–200)	mg/dL
	HDL Cholesterol	Numeric (20–100)	mg/dL
	Triglycerides	Numeric (50–400)	mg/dL
Cognitive & Functional	MMSE	Numeric (0–30)	Lower = impairment
	ADL (Activities of Daily Living)	Numeric (0–10)	Lower = impairment
	Functional Assessment	Numeric (0–10)	Higher = better function
	Memory Complaints	Categorical (0, 1)	0: No, 1: Yes
	Behavioral Problems	Categorical (0, 1)	0: No, 1: Yes
Symptoms	Confusion	Categorical (0, 1)	0: No, 1: Yes
	Disorientation	Categorical (0, 1)	0: No, 1: Yes
	Personality Changes	Categorical (0, 1)	0: No, 1: Yes
	Task Difficulty	Categorical (0, 1)	0: No, 1: Yes
	Forgetfulness	Categorical (0, 1)	0: No, 1: Yes
Diagnosis	Alzheimer's Status	Categorical (0, 1)	0: No, 1: Yes
Confidential Info	Doctor in Charge	Confidential	Always "XXXConfid"

ANNEX II: EDA


Distribution of Age
Age
Count
Distribution of Gender
Gender
Count
Distribution of Ethnicity
Ethnicity
Count
Distribution of EducationLevel
EducationLevel
Count
Distribution of BMI
BMI
Count
Distribution of Smoking
Smoking
Count
Distribution of AlcoholConsumption
AlcoholConsumption
Distribution of PhysicalActivity
PhysicalActivity
Count
Distribution of DietQuality
DietQuality
Count
Distribution of SleepQuality
SleepQuality
Distribution of FamilyHistoryAlzheimers
FamilyHistoryAlzheimers
Count
Distribution of CardiovascularDisease
CardiovascularDisease
Count


Distribution of Diabetes
Diabetes
Distribution of Depression
Depression
Distribution of HeadInjury
HeadInjury
Distribution of Hypertension
Hypertension
Distribution of SystolicBP
SystolicBP
Distribution of DiastolicBP
DiastolicBP
Distribution of CholesterolTotal
CholesterolTotal
Distribution of CholesterolLDL
CholesterolLDL
Distribution of CholesterolHDL
CholesterolHDL
Distribution of CholesterolTriglycerides
CholesterolTriglycerides
Distribution of MMSE
MMSE
Distribution of FunctionalAssessment
FunctionalAssessment
Distribution of MemoryComplaints
MemoryComplaints
Distribution of BehavioralProblems
BehavioralProblems
Distribution of ADL
ADL
Count

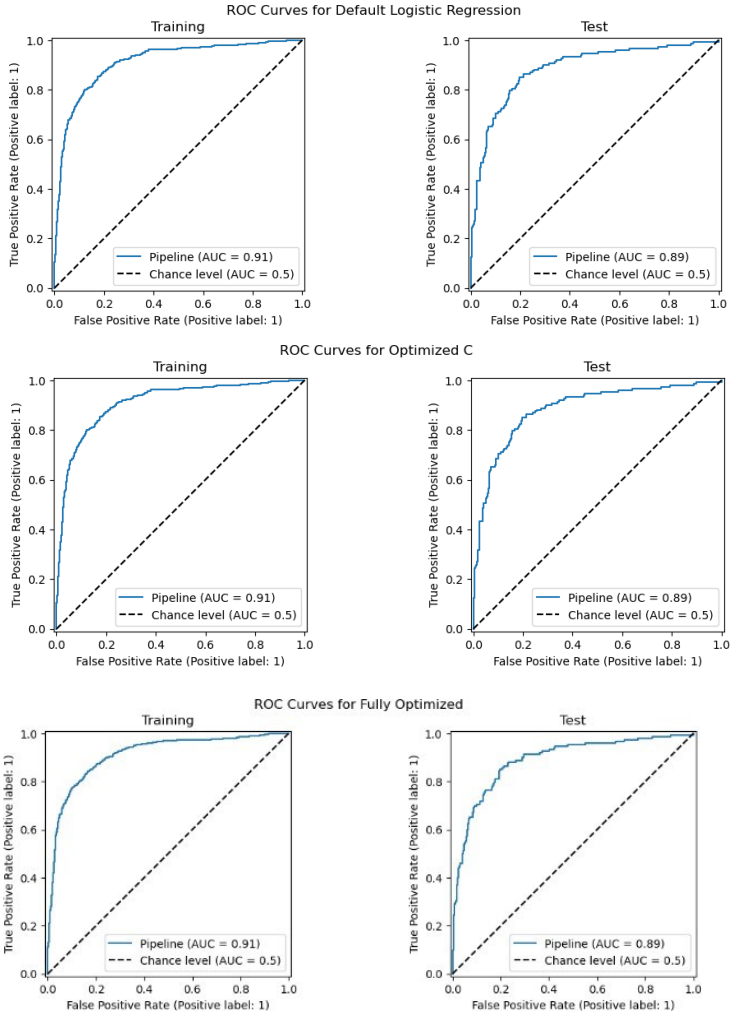

Distribution of Confusion
Distribution of Disorientation
Distribution of PersonalityChanges
Distribution of DifficultyCompletingTasks
Distribution of Forgetfulness
Distribution of Diagnosis

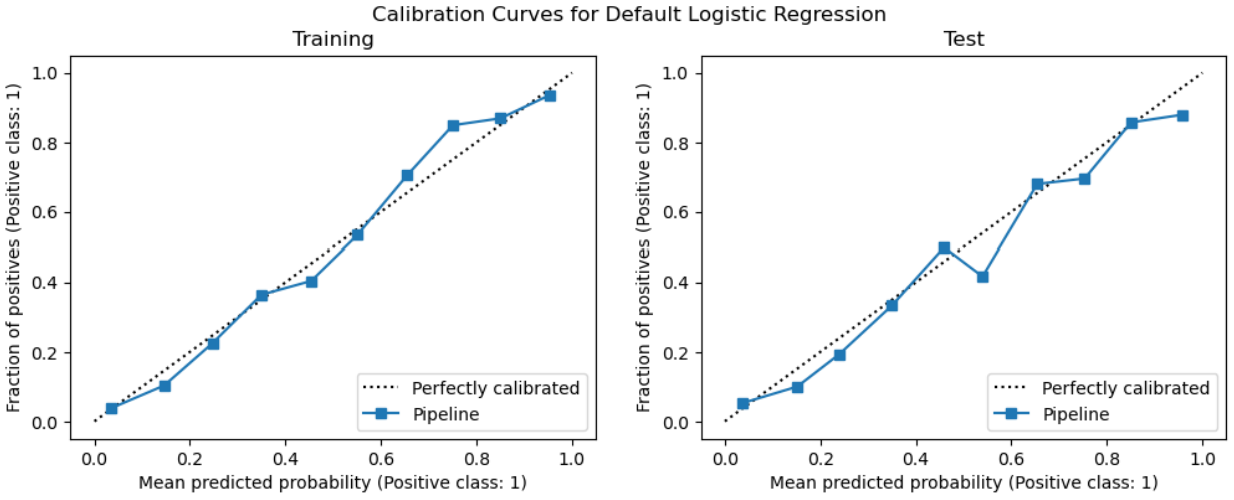

Box Plot: Age
Box Plot: BMI
Box Plot: AlcoholConsumption
Box Plot: PhysicalActivity
Box Plot: DietQuality
Box Plot: SleepQuality
Box Plot: SystolicBP
Box Plot: DiastolicBP
Box Plot: CholesterolTotal
Box Plot: CholesterolLDL
Box Plot: CholesterolHDL
Box Plot: CholesterolTriglycerides
Box Plot: MMSE
Box Plot: FunctionalAssessment
Box Plot: ADL

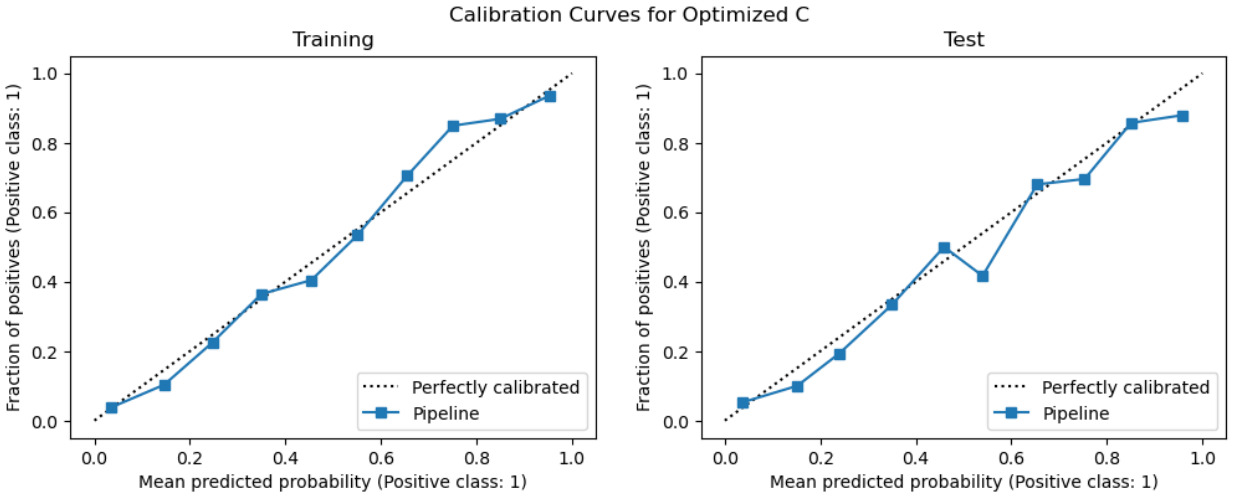
ANNEX III: COMPUTER SPECIFICS

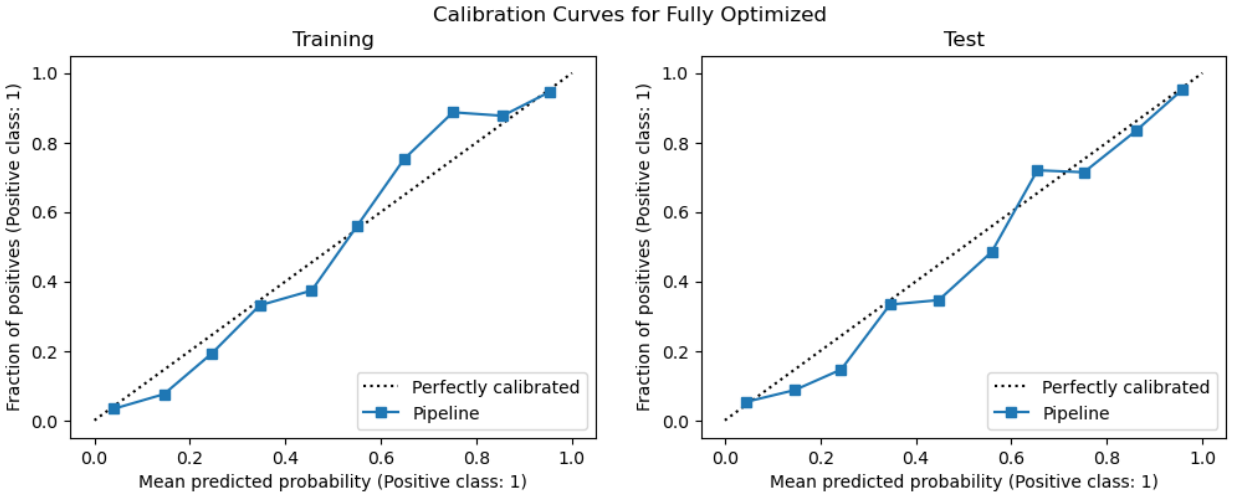
Specification	Value
Device Name	DESKTOP-RQLLIGJ
Processor	Intel(R) Core (TM) i7-9750H CPU 2.60GHz (2.59 GHz)
Installed RAM	16.0 GB
Product ID	00331-10000-00001-AA495
System Type	64-bit operating system, x64-based processor
Windows Edition	Windows 10 Pro
Windows Version	22H2

ANNEX IV: PERFORMANCE METRICS

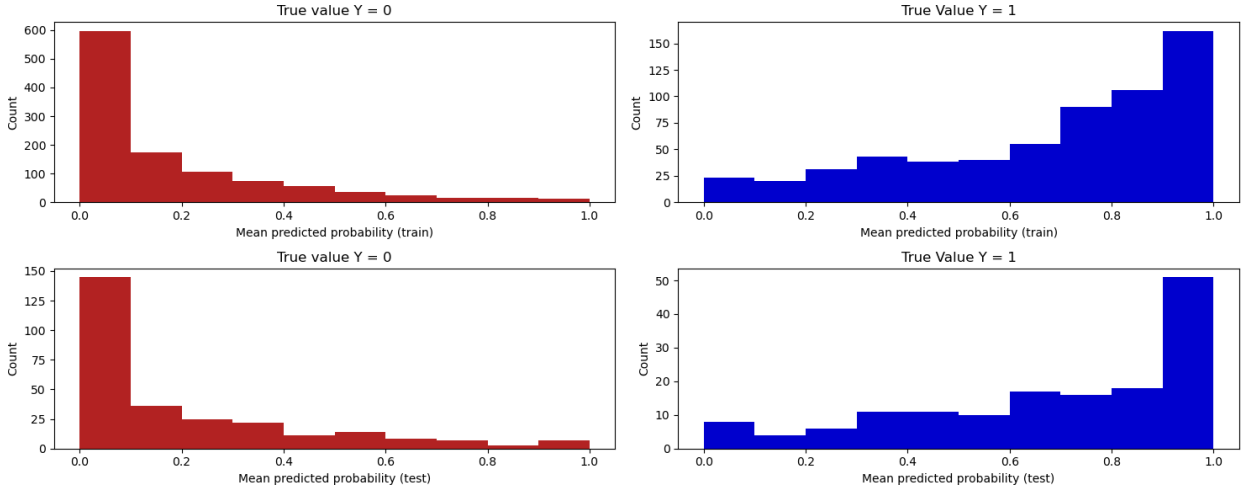



Calibration Curves for Default Logistic Regression
Training
Test
Fraction of positives (Positive class: 1)
Mean predicted probability (Positive class: 1)
Perfectly calibrated
Pipeline

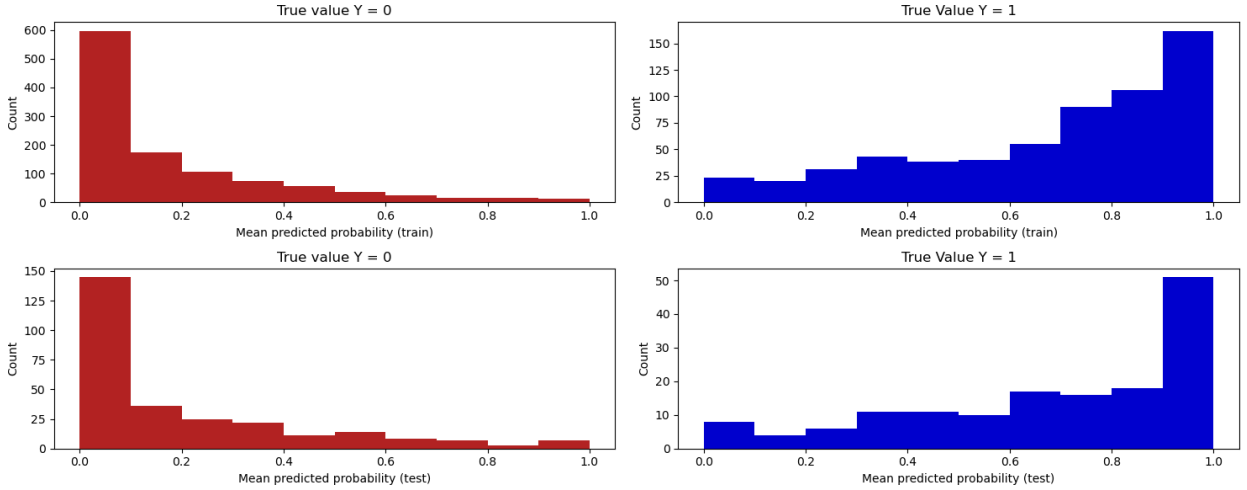

Calibration Curves for Optimized C
Training
Test
Fraction of positives (Positive class: 1)
Mean predicted probability (Positive class: 1)
Perfectly calibrated
Pipeline


Calibration Curves for Fully Optimized
Training
Test
Fraction of positives (Positive class: 1)
Mean predicted probability (Positive class: 1)
Perfectly calibrated
Pipeline

Baseline Histogram



Optimized Model Histogram



Fully Optimized Model Histogram