AlzheimInk: A Mobile Web Application for Detecting Alzheimer's Disease through Handwriting Analysis

Universitat Oberta de Catalunya

Jordi Soriano Reos

Statistical Bioinformatics and Machine Learning

Master's Degree in Bioinformatics and Biostatistics

Name of the tutor:
Romina Astrid Rebrij

Name of the SRP:
Carles Ventura Royo

June 18, 2024

UNIVERSITAT DE BARCELONA

Final Work Card

Title of the work:	AlzheimInk: A Mobile Web Application for Detecting Alzheimer's Disease through Handwriting Analysis
Name of the author:	Jordi Soriano Reos
Name of the tutor:	Romina Astrid Rebrij
Name of the SRP:	Carles Ventura Royo
Date of delivery:	June 18, 2024
Studies or Program:	Master's Degree in Bioinformatics and Biostatistics
Area or the Final Work:	Statistical Bioinformatics and Machine Learning
Language of the work:	English
Keywords:	Alzheimer's Disease, Early Diagnosis, Handwriting, Machine Learning, Neurodegenerative Diseases

Abstract

Early diagnosis of Alzheimer's disease, one of the major neurodegenerative diseases of this century, is crucial for delaying the onset of symptoms, despite the lack of a definitive cure. However, diagnosis remains a challenge due to the need for various tests (cognitive, biomarker and protein analysis, neuroimaging, etc.) and the reality that it is frequently diagnosed by excluding other potential diseases.

In this context, this study aims to develop an accessible alternative for most people: a mobile web application that adapts a diagnostic test based on handwriting, a process that involves cognitive, kinesthetic, and perceptual-motor skills. This test, which was originally performed with a graphic tablet in a prior study, can now be conducted over the internet using only a stylus.

To achieve this goal, several supervised machine learning classifiers were trained using the original study's database. The best model, achieving a theoretical accuracy of 88%, was implemented in the web application. This application, accessible online, performs a series of 19 tasks involving drawing or handwriting and can make diagnostic predictions based on the collected data. In addition, it offers the ability to track users and collect new data to generate an exclusive database for the application.

In conclusion, the tool developed in this study, or future versions of it, has demonstrated the potential to be used as a support in the diagnosis of Alzheimer's disease. It offers an easily accessible, quick, and non-invasive test, making it a valuable addition to existing diagnostic tools.

Resumen

El diagnóstico temprano de la enfermedad de Alzheimer es crucial para retrasar la aparición de síntomas, a pesar de la falta de una cura definitiva. Sin embargo, sigue siendo un desafío debido a la necesidad de múltiples pruebas (cognitivas, análisis de biomarcadores y proteínas, neuroimagen, etc.) y el hecho de que a menudo se diagnostica por exclusión de otras enfermedades.

En este contexto, el estudio busca desarrollar una alternativa accesible para la mayoría: una aplicación web móvil que adapta una prueba de diagnóstico basada en la escritura a mano. Este proceso involucra habilidades cognitivas, cinestésicas y perceptivo-motoras, y se realizó originalmente con una tableta gráfica. Ahora puede realizarse por internet usando un lápiz capacitativo.

Para lograrlo, se entrenaron varios clasificadores de aprendizaje automático supervisado con la base de datos del estudio original. El mejor modelo, con una precisión teórica del 88%, se implementó en la aplicación web. Esta aplicación, accesible en línea, incluye una serie de 19 tareas de dibujo o escritura a mano y puede hacer predicciones diagnósticas basadas en los datos recopilados. Además, permite rastrear a los usuarios y recopilar nuevos datos, generando una base de datos exclusiva para la aplicación.

En conclusión, la herramienta desarrollada en este estudio, o futuras versiones de la misma, tiene el potencial de apoyar el diagnóstico de la enfermedad de Alzheimer. Ofrece una prueba accesible, rápida y no invasiva, lo que la convierte en una valiosa adición a las herramientas de diagnóstico existentes, mejorando con la recopilación continua de datos.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and justification of the project

Alzheimer's disease (AD), a neurodegenerative disorder associated with the accumulation of tau and beta-amyloid proteins in the brain [6], is one of the greatest health challenges of this century [7]. Primarily affecting the elderly, it is characterized by the progressive loss of cognitive abilities. According to 2019 estimates, around 55.2 million patients worldwide suffer from some type of dementia [8], with AD and Parkinson's disease (PD) being the most common causes. Although there is no cure for AD [6], treatments exist that can alleviate associated symptoms [9]. Early diagnosis is known to improve treatment effectiveness [10]. Diagnostic methods include cognitive tests and physical diagnostics such as brain imaging through magnetic resonance imaging (MRI), blood tests, and lumbar puncture. However, due to the technical difficulty and high cost of these methods, diagnostic alternatives have been studied.

A study published in 2022 [5] highlights the possibility of diagnosing AD from handwriting, a skill that requires a complex network of cognitive, kinesthetic, and perceptual-motor abilities [11] and that has long been known to be affected by AD [12]. In the original study [5], various handwriting data were extracted from patients with AD and control patients, obtaining a dataset called DARWIN (*Diagnosis AlzheimeR WIth haNdwriting*). From this data, several models based on machine learning were made with the aim of predicting whether the subject studied has AD or not. The results obtained present accuracies of more than 80%. However, to collect the data, a certain type of graphic tablet connected to a computer with a specific program was used. This graphic tablet, designed for drawing, is a somewhat complex and not exactly cheap device, and its use requires prior knowledge. In order to facilitate access to this type of tests for an early diagnosis of the disease, this work proposes a simplified adaptation of the original study [5]. In this adaptation, interaction will take place by writing and drawing by hand, preferably with a stylus, in a web application adapted to mobile devices with a touch screen. In addition, features can be implemented to monitor the cognitive status of the patient by health professionals through private access. The purpose of this project is to determine if acceptable results can be obtained that yield a possible initial diagnosis of AD in a quick and simple way. This would speed up contact with health professionals to carry out the relevant tests and start treatment as soon as possible, thus improving the lives of patients.

1.2 Objectives of the Project

1.2.1 General Objective

Develop a mobile web app that can predict whether the user suffers from Alzheimer's disease based on parameters obtained from various handwriting tasks.

1.2.2 Specific Objectives

- Evaluate various classification techniques to obtain the most effective predictive model possible.
- Design the application to ensure that the data obtained from it closely resemble the variables of the original database.
- Optimize the selected model to achieve a minimum accuracy of 75%.
- Compare the results obtained against those of the original study.

1.3 Impact on Sustainability, Ethical-Social, and Diversity

This work is linked with the Sustainable Development Goals (SDGs) 3 and 10:

1.3.1 SDG 3 - Good Health and Well-being

The application to be developed aims to assist in the preventive diagnosis of a neurological disease. It does not distinguish between users and is based solely on the recording of their handwriting. This approach promotes well-being by providing access to preventive diagnostic tools, which can lead to early treatment and, therefore, better health outcomes. However, this project may face certain challenges. The main ethical problem is that it provides a diagnosis of a serious disease without the intervention of a health professional expert in the subject. This could result in incorrect diagnoses, which is contrary to SDG 3 that seeks to promote well-being for all at all ages. In addition, a false positive or a false negative can generate stress and anxiety, which can negatively affect the mental well-being of individuals. To better align with SDG 3, the following alternatives can be considered:

- Clarity in communication: Before starting and after finishing the test, it should be clearly indicated that the platform may have faults and that its use is merely informative. This can help establish realistic expectations for users.
- Reference to health professionals: Users should be advised that, to obtain a definitive diagnosis, they should contact a health professional if they deem it appropriate. This can help ensure that users receive appropriate medical care.

- Access through health professionals: To prevent the aforementioned problems that could arise from public access to the application, it could be considered to limit its use to health professionals. In this way, the application would serve as a support diagnostic tool for these professionals, eliminating the need to specifically use the tablets from the original study. In addition, another alternative would be for the doctor to provide a code associated with the patient. This code would allow the patient to access the application remotely, which could be used to track cognitive decline, which is essential for clinical management in early stages of dementia [13].

1.3.2 SDG 10 - Reduced Inequalities

This project has the potential to positively impact society by providing free preventive diagnosis for all. This could help reduce the gap between people with few resources and those with access to good healthcare. In addition, it could facilitate informed decision-making, especially in situations where access to more advanced tests is not available due to the country's socioeconomic situation or the limitations of the user's health service.

1.4 Approach and Methodology

In this work, an approach based on machine learning was adopted. This approach has proven to be effective in reducing the evaluation times of motor functions affected by neurodegenerative diseases [14]. A prominent example is the analysis of handwriting in PD patients, where machine learning models have been developed that have achieved good results in distinguishing between individuals with and without the disease [15]. However, this work focused on the original study by Cilia et al. [5], where a model was created that allowed distinguishing between healthy individuals and individuals with Alzheimer's based on handwriting records on a graphic tablet. In this case, a mobile web app was used.

For the development of this work, the DARWIN database, available in the UC Irvine Machine Learning Repository [16], was used. This database contains data derived from the handwriting records of 174 participants, 89 with AD and 85 healthy, who performed 25 tests, finally obtaining 451 variables in the dataset (25 tests * 18 variables + 1 label). The methodology for this work, which attempted to replicate the original study, was divided into three distinct phases:

- **Design of the data collection application:** In this initial phase, the tests and variables of each one were analyzed. It was decided which tests could not be performed on a small interface, such as writing paragraphs or those that, due to the limitations of the hardware used, could not be recreated, such as measuring the pressure with which one writes or those that could not be replicated as in the original dataset. These types of variables were discarded and, using Python, specifically the Tkinter package [17], a prototype of a graphical interface capable of collecting data when writing or drawing was made. At the end, it provided a Python list that the developed models could predict.

- **Creation and optimization of the predictive model:** With the DARWIN database modified according to the tests and variables required for this case, an initial study was

conducted on the dataset. In this step, functions from the Python scikit-learn package [18] were primarily used. Following this, the samples were partitioned into test data and training data for the development of machine learning models. Supervised machine learning models were generated since user labels with AD or healthy were available for each of the samples. Supervised learning was characterized by its attempt to optimize a model with the aim of obtaining the appropriate combination of values and features that result in a specific label [19]. Models that included those used in the original study [5] (such as Random forest, k-NN, SVM, Decision Tree, etc.) as well as boosting-based algorithms (like AdaBoost, Gradient Boosting, and XGBoost) or other bagging-based algorithms (such as Extra Trees Classifier) were used. During the process, the best hyperparameters were selected using grid search and cross-validation was used to ensure the robustness of the model and prevent overfitting. Finally, the accuracy of the models along with other performance metrics (such as sensitivity, specificity, precision, recall, F1-score and Cohen's Kappa Score) were evaluated, and work was done to improve them to achieve the highest possible performance.

- **Web implementation:** The final phase involved the creation of a functional web application, specifically designed for mobile devices. The development process required the use of the Flask package from Python, building upon a previously created Tkinter interface prototype. Additional technologies such as JavaScript, HTML, and CSS were also employed. The end product was a web application capable of displaying various tests and generating predictions based on the results. Furthermore, it included features for tracking registered users and collecting new data.

Following this methodology, in case of encountering difficulties in any aspect of application development, other options were explored, either by removing variables, improving and optimizing models, or implementing it in a simpler way.

1.5 Work Planning

1.5.1 Tasks

Throughout the entire project, a schedule divided into 5 PECs has been followed, which can be seen in Table 1.1. This schedule has been subject to changes during the course of the project's development.

1.5.2 Calendar

Building on the previously created schedule, a Gantt chart has also been developed to provide a more visual representation of the project's progression. The chart can be viewed in Figure 1.1.

Table 1.1: Schedule of the project.

Description	Start date	End date
PEC1 - Definition and work plan	**28/Feb/2024**	**24/Mar/2024**
Compilation and exploratory analysis of data	28/Feb/2024	01/Mar/2024
Choice of Python package for the development of the data collection graphical interface and initial tests	02/Mar/2024	07/Mar/2024
Selection of variables for the model	08/Mar/2024	08/Mar/2024
Development of the work plan (PEC1)	09/Mar/2024	18/Mar/2024
Delivery and feedback of the work plan	19/Mar/2024	24/Mar/2024
PEC2 - Work development - Phase 1	**20/Mar/2024**	**23/Apr/2024**
Start writing the report (Introduction, Objectives, State of the Art, Materials and Methods...)	20/Mar/2024	08/Apr/2024
Design of the different tests	21/May/2024	22/Mar/2024
Development of the data collection graphical interface	23/Mar/2024	05/Apr/2024
Training and validation of the predictive models	06/Apr/2024	09/Apr/2024
Optimization of the constructed models	10/Apr/2024	14/Apr/2024
Model testing and choice of variables	15/Apr/2024	16/Apr/2024
Integrate the data step from the interface to the model	17/Apr/2024	17/Apr/2024
First GUI calibration	18/Apr/2024	19/Apr/2024
Document everything done for the report	20/Apr/2024	23/Apr/2024
Perform follow-up report (PEC2)	20/Apr/2024	23/Apr/2024
PEC3 - Work development - Phase 2	**24/Apr/2024**	**03/06/2024**
Construction, Optimization and Testing of additional models	24/May/2024	11/05/2024
Results Analysis	29/Apr/2024	11/May/2024
Selection of the best model for the application	12/May/2024	12/May/2024
Research of the different options to develop the web application	13/May/2024	13/May/2024
Definition of the application scheme	14/May/2024	14/May/2024
Development of the application	15/May/2024	23/May/2024
Second GUI calibration	24/May/2024	24/May/2024
Documentation of the work done for the report	25/May/2024	03/Jun/2024
Perform follow-up report (PEC3)	25/May/2024	03/Jun/2024
PEC4 - Closure of the report and presentation	**04/Jun/2024**	**18/Jun/2024**
Exposure of the application on a web server	04/Jun/2024	08/Jun/2024
Finish the project report	09/Jun/2024	12/Jun/2024
Optimize scheme and presentation of the report and the source code	13/Jun/2024	18/Jun/2024
Preparation of the presentation and video	13/Jun/2024	18/Jun/2024
PEC5 - Public defense	**25/Jun/2024**	**05/Jul/2024**
Public defense	28/Jun/2024	28/Jun/2024

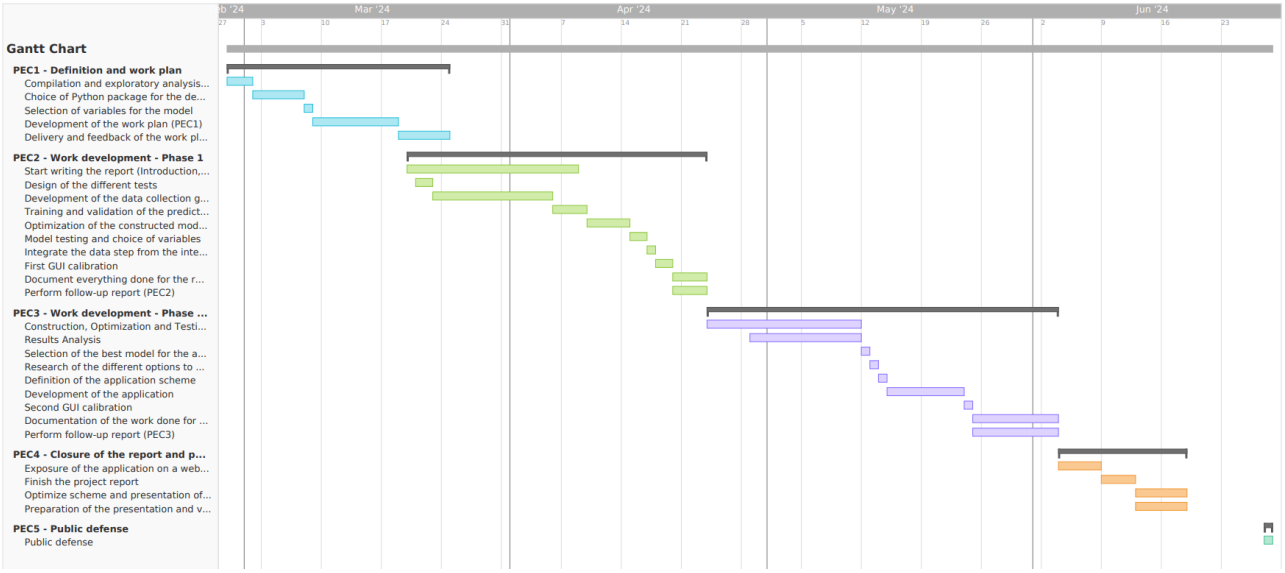


Figure 1.1: Gantt Chart of the Project. Made with TeamGantt [1]

1.5.3 Milestones

During the project's development, milestones were also established with their respective completion dates. This allows for the division of the project into its various important parts. These milestones are displayed in Table 1.2.

Table 1.2: Milestones of the Project.

Description	Date
Delivery of the work plan	19/Mar/2024
Full integration of the graphical interface with predictive models	11/Apr/2024
Delivery of work development - Phase 1	23/Apr/2024
Fully developed application	19/May/2024
Delivery of work development - Phase 2	28/May/2024
Delivery of the report and presentation	18/Jun/2024
Public defense	28/Jun/2024

1.5.4 Risk analysis

At the outset of the project's development, potential factors that could negatively impact the project were identified, obviously not taking into account the project's scope and the time available for its development. These factors can be consulted in Table 1.3.

Table 1.3: Project Risk Analysis

Risk Description	Severity	Probability	Mitigation
Problems replicating certain predictor variables obtained with the created interface.	High	Moderate	Contact the original study researchers or dispense with the variable.
Problems creating predictive models in Python as R has normally been used.	Moderate	Low	Delve into online resources about the models that can be performed in Python.
Failing to achieve the minimum target performance due to predictor variables reduction.	High	Moderate	Research all factors to optimize the models.
Unable to develop the web application due to implementation difficulties.	High	Moderate	Research how to implement a mobile web app in Python.

1.6 Brief Summary of the Products Obtained

- **Report**: Document in PDF format that will detail all the research carried out, the methodology used, the results obtained, discussion, economic evaluation, future work, and conclusions reached during the development of the project.
- **Product**: A mobile web application accessible through a QR code or link, which can be accessed from any device via the internet.
- **Virtual Presentation**: A video where an oral presentation of the completed project will be delivered using slides.

1.7 Brief Description of the Other Chapters in the Project Report

- **State of the Art:** This chapter provides an explanation of the topics that surround the project, such as Alzheimer's disease, Machine Learning, Handwriting analysis, and discusses their current state of development.
- **Materials and Methods:** This chapter provides a detailed explanation of the software and libraries used in the project. It also describes how they were utilized and what was accomplished during the project.
- **Results:** This chapter presents the results obtained from the project, including the performance of the different models developed and the outcomes of the application implementations.

- **Discussion:** This chapter delves into the functionality of the application, explores potential enhancements, and draws comparisons with the application from the original study. It also addresses the implications and limitations of the project, providing a comprehensive overview of the work done.
- **Economic Evaluation:** This chapter highlights the most relevant aspects of the costs associated with the application, from the development phase to the maintenance expenses. It also discusses the expected economic benefits.
- **Conclusion and Future Work**: This final chapter provides a brief summary of the key findings obtained during the project, discusses potential future improvements, and recaps the achievement of the initial project objectives.
- **Glossary**: The acronyms that appear in the report are defined here.
- **Appendix**: In the appendix, one will first encounter tables and figures that may not be as relevant to the report or that were too large to be included in the report. Following this, relevant links associated with the project are presented.

Chapter 2

State of the Art

2.1 Alzheimer's Disease

The increase in life expectancy of over 60 years worldwide has been rising over the years, and it seems that there is no limit in sight [20]. However, this fact does not always bring good news. With the increase in life expectancy, other types of problems also arise. One of them, concerning people's health, is that the risk of suffering from some type of dementia or neurodegenerative disorder increases exponentially with age [20]. According to estimates made in 2019, it is estimated that 55.2 million patients worldwide suffer from some type of dementia [6], with AD being the most common, followed by PD. Therefore, the fight against these diseases is one of the greatest health challenges of this century [7].

Alzheimer's disease, the first record of which dates back to 1907 by clinical psychiatrist Alois Alzheimer from the case of a 50-year-old woman [21], is a neurodegenerative disorder associated with the accumulation of tau and beta-amyloid proteins in the brain [6].

2.1.1 Sintomatology

This condition, which primarily affects the elderly, is characterized by the progressive loss of an individual's cognitive abilities, such as memory at various levels, thinking, judgment, learning ability, executive and visuospatial dysfunction [6, 5]. The main symptom of the disease, however, is memory deterioration, initially affecting short-term memory and, as the disease progresses, also affecting more distant memories. In addition to memory loss, Alzheimer's disease affects other cognitive deficits not related to memory, such as aphasia, executive dysfunction, apathy, or personality change. Other common symptoms include difficulty in finding words and a decline in visuospatial skills in the early stages. Executive dysfunction appears in predementia stages [6].

2.1.2 Neuropathology

When it comes to the neuropathological changes caused by AD, there are generally two types: positive lesions, due to the accumulation of molecular deposits, and negative lesions, due to loss or atrophy of functions [2, 22]. The main causes are as follows:

- **Senile Plaques**: Senile plaques are deposits of beta-amyloid (Aβ) protein with different morphological forms formed by the action of β-secretase and γ-secretase enzymes that cleave the amyloid precursor protein (APP). These deposits can accumulate and form amyloid plaques, contributing to neurotoxicity. The build-up of these plaques in brain areas can cause neuronal damage and cognitive impairment [23].

- **Neurofibrillary Tangles (NFTs)**: Neurofibrillary tangles are abnormal filaments of hyperphosphorylated tau protein that can twist together to form paired helical filaments (PHF) and accumulate in nerve cells, causing loss of microtubules and associated proteins. These NFTs, present in the brains of AD patients, go through several stages, from the accumulation of phosphorylated tau to the formation of extracellular tangles due to neuronal loss [24, 25].

- **Synaptic Loss**: Synaptic loss in the neocortex and limbic system, observed in the early stages of AD, causes memory impairment. This synaptic damage involves defects in axonal transport, mitochondrial damage, oxidative stress, and accumulation of beta-amyloid and tau proteins at synaptic sites, which ultimately leads to the loss of dendritic spines, presynaptic terminals, and axonal dystrophy [26].

Figure 2.1 illustrates the areas affected by AD as previously mentioned.

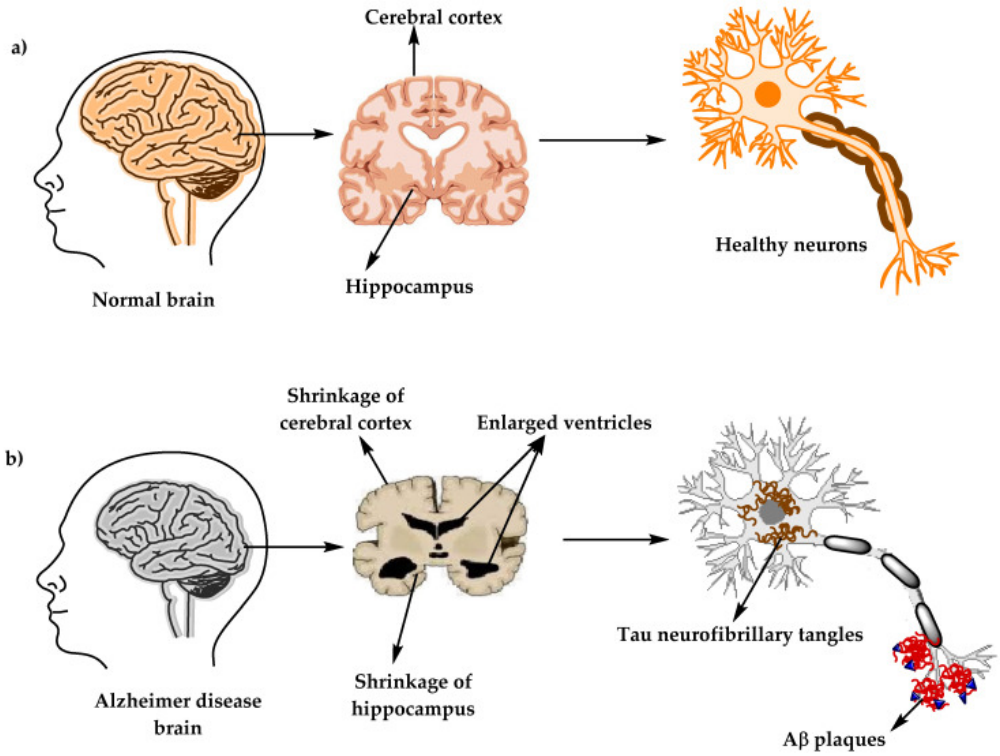


Figure 2.1: The physiological structure of the brain and neurons in **(a)** healthy brain and **(b)** Alzheimer's disease (AD) brain [2].

2.1.3 Causes and Risk Factors

Hypotheses

Alzheimer's disease is considered a multifactorial disease associated with a variety of risk factors, however, the underlying cause that causes the neuropathological changes mentioned in the previous section is still unknown. That is why there are currently mainly two hypotheses:

- **Cholinergic Hypothesis**: In 1970, a relationship was established between neocortical and presynaptic cholinergic deficits and the enzyme choline acetyltransferase (ChAT), precursor of acetylcholine (ACh), an essential neurotransmitter involved in memory, attention, sensory information, learning, and other critical functions. β-amyloid is believed to affect cholinergic neurotransmission and to cause a reduction in the choline uptake and a release of ACh. Studies demonstrated that cholinergic synaptic loss and amyloid fibril formation are related to Aβ oligomers' neurotoxicity and to interactions between Acetylcholinesterase (AChE) and Aβ peptide [27, 28, 2].

- **Amyloid Hypothesis**: This is the most accepted hypothesis of inherited Alzheimer's disease (IAD). This hypothesis focuses on the fact that the abnormal deposition of β-sheets in the central nervous system has a strong correlation with dementia. However, amyloid plaque deposits also appear in healthy brains with age, so it was studied whether the amyloid plaques were responsible for the AD or not. The amyloid hypothesis suggests that the degradation of Aβ, derived from APP by β- and γ-secretase, is decreased by age or pathological conditions, which leads to the accumulation of Aβ peptides (Aβ40 and Aβ42). Increasing the ratio of Aβ42/Aβ40 induces Aβ amyloid fibril formation, resulting in neurotoxicity and tau pathology induction, and consequently, leading to neuronal cell death and neurodegeneration [2, 29, 30, 31].

Risk Factors

As previously mentioned, the risk factors involved in Alzheimer's disease are varied, as can be seen in Figure 2.2:

- **Advanced Age**: Age is indisputably the most significant risk factor for Alzheimer's disease. The majority of cases are diagnosed in individuals over the age of 65 [32], and the likelihood of developing the disease doubles approximately every five years thereafter [2].

- **Genetics**: Mutations in genes such as APP, PSEN1, PSEN2, ApoE, ABCA1, CLU, BIN1, ECSIT, ESR, among others, are strongly related to AD. These genes somehow affect the production and accumulation of beta-amyloid protein deposits, thus potentially accelerating the progression of the disease [2].

- **Environmental Factors**: Among them, elements such as air pollution, an unhealthy diet, frequent exposure to metals, and even chronic infections can all contribute to the development of AD by inducing oxidative stress and inflammation [2].

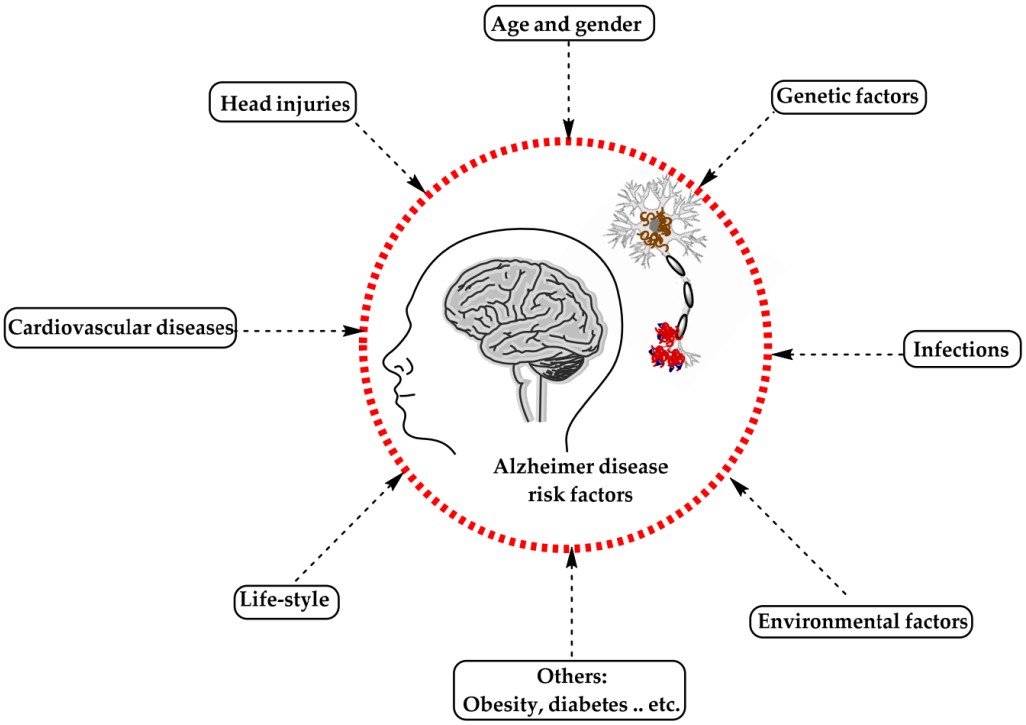


Figure 2.2: Summary diagram of the main risk factors associated with AD [2].

- **Medical Factors**: Among them, cardiovascular diseases such as obesity and diabetes are related to a higher risk of developing the disease, possibly due to the impact of these on blood circulation and brain function [2].

2.1.4 Treatment

As of today, there is no cure for Alzheimer's disease, however, there are a series of treatments available to try to alleviate its associated symptoms [33, 34]:

- **Cholinesterase Inhibitors:** These medications, which include donepezil, rivastigmine, and galantamine, increase the amount of acetylcholine in the brain by inhibiting the enzyme that breaks it down [35]. Acetylcholine, as it has been said before, is a crucial neurotransmitter for the processing of memory and learning. These drugs are primarily effective in treating the cognitive symptoms of AD, but they do not stop its progression. They are used to treat mild to moderate symptoms [2].

- **NMDA Antagonists**: Memantine is a low-affinity noncompetitive antagonist of the NMDAR, a subtype of glutamate receptor. This medication prevents over-activation of the glutaminergic system involved in neurotoxicity in AD cases. It adjusts the activity of glutamate, another essential neurotransmitter involved in memory and learning. It helps control the symptoms of the disease by preventing neuronal toxicity caused by excess glutamate. It is used to treat moderate to severe symptoms [36, 37, 38, 2].

- **Disease Modifying Therapies (DMTs)**: These therapies, which are under research and development, aim to alter the progression of AD disease by acting on various pathophysiological mechanisms. Several DMTs have been developed and have entered clinical trials, such as AN-1792, a synthetic peptide of Aβ and the first active immunotherapy treatment for Alzheimer's. Other DMTs targeting Aβ and tau pathologies, such as *aducanumab*, *gantenerumab*, and *crenezumab*, are still under investigation. These therapies aim to intervene in specific pathogenic mechanisms such as the abnormal accumulation of tau and beta-amyloid proteins, inflammation, oxidative damage, iron dysregulation, and cholesterol metabolism [2, 39, 40, 41, 2, 33].

- **Natural Compounds**: Various natural substances, such as nicotine, vitamins C, E, and D, *bryostatin* [42], and other traditional Chinese medicine compounds [43], are being explored for their potential to prevent or treat AD. These compounds could offer benefits due to their antioxidant, anti-inflammatory, and neuroprotective properties [2].

2.1.5 Diagnosis

The effectiveness of AD treatment largely hinges on early diagnosis [2], which is why a significant amount of effort is dedicated to developing early and efficient diagnostic methods.

The diagnostic process for AD is multifaceted and complex. For many years, due to the high risk-to-benefit ratio associated with biopsies, autopsies have been considered the *gold standard* for diagnosing AD [44]. Over time, new methods have emerged, such as the analysis of biomarkers in cerebrospinal fluid (CSF) through lumbar puncture, which generally achieve an accuracy between 80-90% [45, 46]. Additionally, blood biomarker analysis offers similar diagnostic accuracy and some recent studies claim to have achieved up to 95% accuracy [47].

Furthermore, patients must undergo a series of other tests, including cognitive examinations, routine laboratory analyses (such as vitamin B12 levels, thyroid levels, and complete blood count), and optional analyses (such as erythrocyte sedimentation rate, HIV test, Lyme disease serology, and electroencephalography). Neuroimaging, such as computed tomography (CT) or MRI for neurons, may also be conducted [5, 48].

In 1984, a collaborative group was established by the National Institute of Neurological and Communicative Disorders and Stroke (NINCDS) and the Alzheimer's Disease and Related Disorders Association (ADRDA), collectively known as NINCDS-ADRDA. Their mission was to set the standards for diagnosing AD. The criteria included the diagnosis of dementia through neuropsychological assessments, the progression of memory impairment, the interference with daily life activities, and other symptoms such as aphasia (a disorder affecting language), apraxia (a disorder affecting motor skills), and agnosia (a loss of perception). These symptoms could appear between the ages of 40 and 90 in the absence of systemic or brain diseases [49, 50].

In 2011, the criteria underwent revisions and enhancements to refine the diagnosis of the disease, thereby increasing its specificity and sensitivity. Separate criteria were formulated for those exhibiting probable and possible AD dementia in clinical environments and for those showing probable or possible AD with pathophysiological evidence for research-oriented purposes, along with the inclusion of clinical biomarkers. Moreover, the introduction of new biomarkers further improved the diagnostic process. These biomarkers were divided into two categories: the first being markers of brain amyloid, such as positron emission tomography (PET) and

cerebrospinal fluid (CSF), and the second being markers of neuronal damage, like cerebrospinal fluid tau, fluorodeoxyglucose (FDG) for assessing metabolic activity, and MRI for measuring atrophy [51, 52, 53].

2.2 Machine Learning

Machine Learning (ML), a branch of computer science, aims to learn from data to improve performance in various tasks [54]. This approach is especially useful in the field of health, where an automated, highly flexible, and computationally intensive approach is required to identify underlying patterns in complex data structures [55].

Unlike traditional parametric methods, which require prior knowledge of the relationships between variables and make numerous statistical assumptions, ML-based methods allow work without prior understanding of what is being sought. ML algorithms establish relationships between data by applying logical rules and mathematical tools [55, 56].

These algorithms are particularly useful for working with high-dimensional data, where humans would have difficulty finding meaning among the variables or would require a great effort to do so. However, it is important to note that these algorithms tend to overfit the data with which they have been trained, which can hinder their generalization to new data unless good practices and methods are applied to prevent it [56]. An example of a field where ML is gaining ground is in psychology and psychiatry. Here, the spectrum of collected data (such as self-report measures, psychological factors, imaging data) is being expanded to levels of thousands of measures per dataset. This is done in order to evaluate the phenotypes, risk factors, and prognoses associated with various mental illnesses. It can even help conceptualize mental disorders and predict the risk and trajectory of symptoms [56, 57].

It is important to note that there are many types of algorithms and there is not one that works best in all contexts. However, many algorithms can produce similar results [56]. In general terms, ML methods can be classified into three categories: descriptive models (discover patterns without a measured outcome), predictive models (forecast values based on other variables), and meta-learners (instead of learning a specific task these models focus on learning how to learn more effectively)[19]. Examples of descriptive models include clustering and PCA, while predictive models include probabilistic classifiers like Naive Bayes, and linear classifiers such as Logistic Regression and Support Vector Machine. Other important classifiers are K-Nearest Neighbors and boosting methods like AdaBoost and Gradient Boosting.

2.3 Handwriting and Neurodegenerative Diseases

Neurodegenerative diseases (NDs), such as Alzheimer's and Parkinson's, are known to cause cognitive and motor deterioration in those affected, leading to memory loss and confusion [58]. Among the daily activities significantly impacted by NDs is handwriting [59]. Handwriting arises from a complex network composed of cognitive, kinesthetic, and perceptual-motor skills [11], which can be compromised when developing an ND.

Literature indicating that AD affected handwriting has existed since 1907 [12]. More recent studies show how patients with AD exhibit alterations in spatial organization and poor control of movements [60]. Similar observations have been made in cases of PD, where the main

functions affected in handwriting are the presence of micrography, slower movements, and jerks [61].

2.3.1 Modeling Handwriting

Due to the scientific community's keen interest in discovering all aspects of handwriting affected by NDs, research has been conducted to try to model the various processes involved in writing tasks. These models can be divided into two types: computational and cognitive [62, 63].

- Computational models aim to reconstruct the final outcome of handwriting movements in terms of velocity and acceleration profiles, stroke shapes, all from the perspective of mathematics, physics, and computer science [3]. Among the investigated approaches are the Kinematic Theory [64, 65], which models neuromuscular systems in rapid movements, geometry-based models [66] that are based on the superposition of basic elliptical strokes, and oscillatory models [67], which represent strokes as coupled oscillations in orthogonal directions.

- Cognitive models attempt to model the generative processes that give rise to cognitive and/or motor acts [3]. These models typically deal with issues such as learning, movement memory, planning, and sequencing, among others. These studies are based on the vast amount of data related to the neuronal processes that take place in the brain areas related to motor learning. An example of this can be VITEWRITE [68], which is a motor program that interacts with a trajectory generator to create complex handwritten scripts. Another instance is the handwriting motor learning model proposed by R. Senatore and A. Marcelli [69]. This model suggests that the process of learning handwriting involves mastering the sequence of points to be reached and acquiring the corresponding sequence of motor commands.

2.3.2 Current State of Research

In 2019, C. De Stefano et al. [3] presented a review of the literature on handwriting analysis to support the diagnosis of AD and PD, as well as mild cognitive impairments. The tables 2.1, 2.2 and 2.3 group all the studies carried out based on the tasks and findings. Currently, advancements continue to be made in this line of research from a machine learning perspective. For instance, a dataset for AD prediction called DARWIN was created, with which predictive models were developed based on various handwriting tests [5]. Various machine learning approaches have also been implemented for AD using Bayesian Networks, Lognormal Features, and even Deep Learning through image analysis [70, 71, 72], as well as for PD using similar approaches [73, 74].

2.3.3 Project Rationale

After what has been exposed in the previous sections, the importance of early diagnostic methods and support for neurodegenerative diseases such as AD is evident. In addition, the great potential of predictive models based on machine learning has been observed and how these

Table 2.1: Summary of statistical studies of AD. Detailed references in C. De Stefano et al. [3].

#	Tasks	Findings
1	Straight lines, cursive-connected loops, single circle, continuous circles drawing. "llll" writing.	Slow movements, lower peak velocity, reduced (time) duration.
2	Written and oral spelling task; irregular words and non-words; auditive stimuli of concrete and abstract words.	Moderate AD subjects differ from mild subjects and controls for all written and verbal tasks; Grapho-motor impairments come in addition to the lexical and phonological impairments.
3	Signature and spontaneous writing.	Repetitions, omissions and substitutions of letters; correlations between spontaneous writing indexes and neuropsychological test results.
4	Name drawings of objects.	AD patients were more successful in retrieving names of objects presented in the dated compared to contemporary unique conditions.
5	Copy of a shopping list and of a letter; copy of a drawing.	Alterations in spatial organization accompanied by poor control of the movement; Time-in-air differs significantly among MCI, AD and HC patients.
6	Picture description, word fluency, spelling to dictation and confrontational naming; mnemonic task concerning semantic knowledge and spatial and temporal orientation.	Mild AD patients differ from controls only for verbal and written versions of the word fluency task; performance deterioration along the days.

Table 2.2: Summary of classification studies of AD. Detailed references in C. De Stefano et al. [3].

#	Tasks	Findings
1	Copying task (words, numbers, text with or without cues).	Kinematic measures within to MMSE; pressure and time in-air were the best performing features.
2	Signature.	Online signature analysis can be used as a tool for early diagnosis of AD.
3	Simple words writing.	Non-smooth movements (irregular velocity profile).
4	Copying task (words, drawings, etc.).	Qualitative combination of the parameters is crucial for group discrimination.

Table 2.3: Summary of statistical studies of PD. Detailed references in C. De Stefano et al. [3].

#	Tasks	Findings
1	Meanders, circle, star and spiral drawing. Sentence/name writing. Copying task.	Slower movements.
2	Loops drawing. Sentence/name writing. Copying task.	Reduced dimension.
3	Meanders, horizontal, straight forward and backward slanted lines, circles drawing. Sentence writing.	Tremor/jerk.
4	Figure drawing. “llll” writing.	Visual feedback can help PD patients to increase stroke dimension.
5	Figure drawing. Adjust the drawing size based on visual information. ”llll” writing under different size and time conditions.	PD patient less able than EC to adjust the size of their drawing to a specific target.
6	Circle drawing before and after medication. “llll” and “eeee” and sentence writing before and after the medication.	Medications reduces (on a limited timespan) main PD handwriting characteristics.
7	Writing under visual and auditory feedback.	Training can help PD patients to increase writing dimension.

are being used to detect AD through handwriting analysis. Therefore, the development of a web application capable of detecting Alzheimer's through this analysis is an economical and easy-to-use option that can contribute to the diagnosis and the start of Alzheimer's disease treatment.

Chapter 3

Materials and methods

The development of this project can be segmented into several distinct phases: the design of the various tasks for the test, the creation of the graphical interface where the test is conducted, the generation and selection of the most effective predictive models, and the development and deployment of the web application.

This study is based on the DARWIN (Diagnosis AlzheimeR WIth haNdwriting) database [10, 5], extracted directly from the UC Irvine Machine Learning Repository [16]. This is a collection of databases, domain theories, and data generators that are designed to be used in analysis through machine learning algorithms.

The chosen programming language is Python [75], due to its versatility. Specifically, version 3.8.10 is used in the integrated development environment (IDE) PyCharm, version 2023.3.3 [76], developed by the JetBrains company.

The equipment used for the development of the work consists of a PC with an Intel Core i5-7400 processor and 8GB of RAM. However, for the training phase of the predictive models, a laptop with an AMD Ryzen 7 4800H processor and 16 GB of RAM was used, due to its better performance in that phase.

3.1 Task Design

The first phase of this project is based on the design of the different tests that each user will perform through the graphical interface. For this, each of the tasks was first selected and modified, if necessary, based on the tasks of the original dataset, which consists of 25 tasks that include memory and dictation tasks, graphic tasks, and copy tasks, which can be seen in Table A.1.

In that study, the "Wacom Bamboo Folio Large" graphic tablet was used, on which users wrote directly on a sheet of paper that was placed right on top of the graphic tablet. The pencil used was special, allowing to write with ink and register the movement and other variables such as pressure, which allowed a later analysis through a computer program. However, in this project, an adaptation of the mentioned one has been made. Unlike graphic tablets, touchscreen smartphones have been used in this case to perform various tests. A capacitive stylus is required for these tests, an example of which can be seen in Figure 3.1. Given that we are limited by the technical specifications of smartphones and the stylus, tasks and variables that cannot be reproduced have been eliminated.

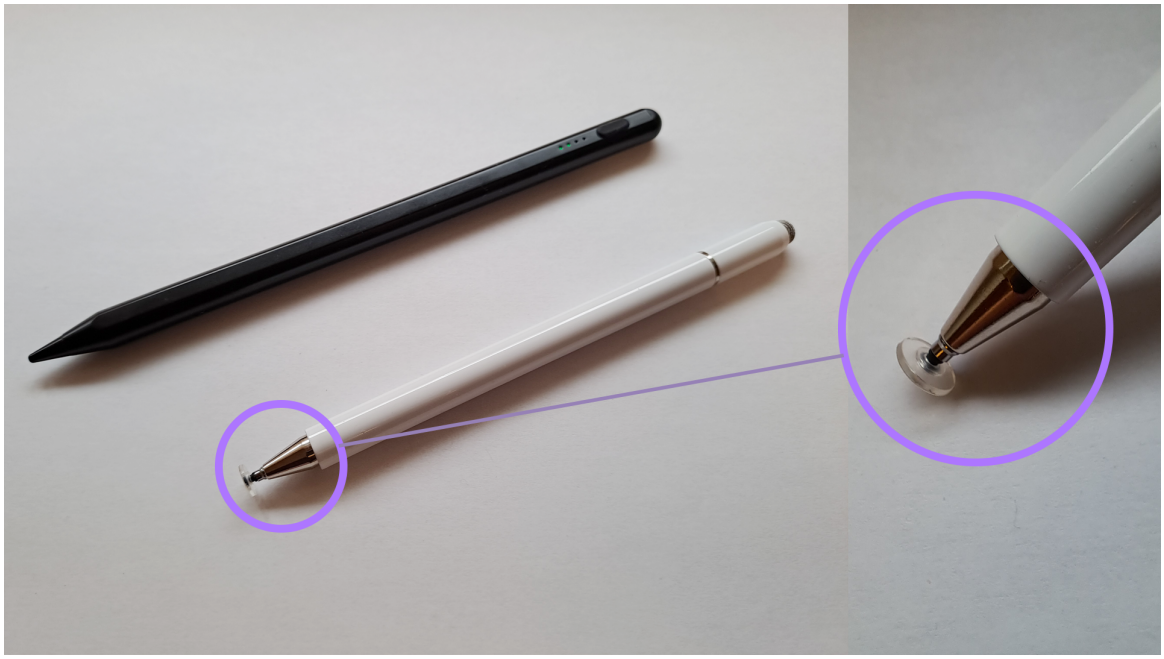


Figure 3.1: Examples of capacitive styluses with which the test can be performed. The black one is active and the white one is passive.

Regarding the tasks, numbers 7, 17, 19, and 25 will be discarded because, due to screen size issues, something that requires a large writing area cannot be written. Also, tasks 20 and 23 will be discarded because a dictation of the required phrases/text cannot be reliably performed without having a physical person to dictate the sentence. Another aspect to consider is that the original study was conducted in Italian, as it was developed in Italy. In this case, an adaptation to Spanish has been wanted due to the written similarity of some words between both languages and the greater extension of Spanish worldwide. For this, the words have been modified for tasks 10 and 11, going from "foglio" to "folios", in tasks 12 and 13 from "mamma" to "mamá", in test 14 from "telefono", "cane" and "negozio" to "teléfono", "cena" and "negocio", in test 15 from "bottiglia to "botiquín" and in test 18 from "sedia" to "silla", simply by translating. With all the changes made, the protocol will include a total of 19 different tests, which can be seen in the table 3.1.

Each of these tests has a reason why it is interesting to predict Alzheimer's:

- Task 1: Performing a signature is a very popular task in the literature [77].
- Tasks 2 and 3: These tasks of joining two points horizontally and vertically continuously 4 times are required for left-right hand movements that include wrist joint movements, while in task 3 to perform up-down movements finger joint movements are required [78].
- Tasks 4 and 5: In this task, a circle is traced continuously 4 times, first 6 cm in task 4 and then 3 cm in task 5. These tasks together allow to test the automaticity of movements and the regularity and coordination of the sequence of movements [79].
- Task 6: This task consists of copying three letters ('l', 'm', 'p') chosen due to their different graphic composition and ascending and descending strokes [80].

Table 3.1: List of tasks performed. Task categories are: memory and dictation (M), Graphic (G), and Copy (C). This table is based on the one from the original study [4].

#	Description	Category
1	Signature drawing	M
2	Join two points with a horizontal line, continuously for four times	G
3	Join two points with a vertical line, continuously for four times	G
4	Retrace a circle (6 cm of diameter) continuously for four times	G
5	Retrace a circle (3 cm of diameter) continuously for four times	G
6	Copy the letters 'l', 'm' and 'p'	C
7	Write cursively a sequence of four lowercase letter 'l', in a single smooth movement	C
8	Write cursively a sequence of four lowercase cursive bigram 'le', in a single smooth movement	C
9	Copy the word "folios"	C
10	Copy the word "folios" above a line	C
11	Copy the word "mamá"	C
12	Copy the word "mamá" above a line	C
13	Memorize the words "teléfono", "cena", and "negocio" and rewrite them	M
14	Copy in reverse the word "botiquín"	C
15	Copy in reverse the word "casa"	C
16	Write the name of the object shown in a picture (a chair)	M
17	Retrace a complex form	G
18	Copy a telephone number	C
19	Draw a clock, with all hours and put hands at 11:05 (Clock Drawing Test)	G

- Tasks 7 and 8: They require to write continuously and in cursive four times the letter "l" and the bigram "le" respectively. In this way, the control of movement alternation is checked [15].

- Tasks 9, 10, 11 and 12: They involve copying words (folios and mamá), being this one of the most explored activities when analyzing people with some cognitive impairment. To see if there are differences in spatial disposition, they have been tested with and without a guide line where to write [15, 81, 80].

- Task 13: This task is based on checking the user's short-term memory. First showing three words that must be memorized and then written [82].

- Tasks 14 and 15: These tasks consist of writing backwards two words "botiquín" and "casa", inspired by a test of the MMSE test that consists of spelling a word backwards [83].

- Task 16: Here it is required that the user write the name of the object that will be shown on the screen, being this a "silla", thus producing a semantic articulation of meaning attribution [84].

- Task 17: This task requires retracing a complex shape with different curves and sizes, starting from one point and ending at another. In this way, both the user's fine and long motor control abilities can be evaluated [85, 86].

- Task 18: A phone number has to be copied. This task was related to another task eliminated in which the phone number was dictated, since there is the hypothesis that

the motor planning between the two forms is different. However, the task of copying the phone number only has been left [80].

- Task 19: In this task, an analog clock must be drawn with all the numbers marking 11:05. This test is known as the Clock Drawing Test (CDT), which has been shown to have high sensitivity for mild AD [87].

The background images used in each task can be seen in Figure A.1. These were created from the same images that were used in the original study, which were provided by the authors of the study. They were later edited with Photoshop to scale them to the original size and integrate the statement of the tasks to be performed in each one.

3.2 Prediction Variables

The first step was to establish which variables should be collected. In the original study, 18 variables were started for each task (air_time, disp_index, gmrt_in_air, gmrt_on_paper, max_x_extension, max_y_extension, mean_acc_in_air, mean_acc_on_paper, mean_gmrt, mean_jerk_in_air, mean_jerk_on_paper, mean_speed_in_air, mean_speed_on_paper, num_of_pendown, paper_time, pressure_mean, pressure_var, total_time), an identifier and the class to which it belongs (healthy or AD). However, since neither the pressure nor the records of when the pen is in the air (in_air) can be replicated, these variables have been eliminated.

An analysis of the multicollinearity of the variables that were suspected to present multicollinearity was carried out. The first case was for the total_time variables, since these variables are the direct sum of paper_time and air_time. The second case was for the mean_speed_on_paper, mean_acc_on_paper and mean_jerk_on_paper variables, since acceleration and jerk are respectively the first and second derivative of speed with respect to time. A Variance Inflation Factor (VIF) analysis was performed for these two cases for each of the tasks. In the first case, a VIF of infinite value was obtained as expected, while for the second case, for acceleration and jerk, there were cases in which the VIF was above or close to 5, the threshold that indicates a multicollinearity problem. In addition, these variables were not correctly extracted later with the graphical interface developed, so it was decided to eliminate acceleration and jerk.

Finally, a total of 8 variables were obtained for each of the 19 tasks, adding up to a total of 152 predictor variables. Below, each of the 8 variables is explained:

- **paper_time{i}**: Time spent performing on-paper movements in milliseconds.
- **air_time{i}**: Time spent performing in-air movements in milliseconds.
- **disp_index{i}**: The dispersion index measures how the handwritten trace is dispersed. For this, the canvas is divided into boxes of a certain pixel size. Then, the number of boxes that contain some fragment of writing or drawing is divided by those blank squares where nothing has been drawn/written, thus obtaining the dispersion index.
- **gmrt_on_paper{i}**: The generalization of the Mean Relative Tremor (MRT) [88] while writing on paper. The MRT measures the amount of tremor when drawing spirals and meanders. This feature is equal to the average distance between the ith sample of the

written trace and another taken d samples before. For the GMRT, the distance between the ith point of the trace and the origin of the reference system, which in this case is the top right corner of what would be the tablet, has been used. The formula is as follows:

$$\frac{1}{n-d}\sum_{i=d}^{n}\left|r_{HT}^{i}-r_{HT}^{(i-d+1)}\right|$$

- **max_x_extension{i}**: Maximum extension recorded on the X axis. It is calculated as the distance between the closest and farthest point on the X axis.
- **max_y_extension{i}**: Maximum extension recorded on the Y axis. It is calculated in the same way as in the previous case, but on the Y axis.
- **mean_speed_on_paper{i}**: Average speed in movements while writing on paper. It is the result of the distance of the movements divided by the writing time on paper.
- **num_of_pendown{i}**: It is the count of the number of times the pen touches the screen.

Given that a graphical tablet was used in the original study, to which the data have been adapted, and now with the adaptation to smartphones, the writing area has been considerably reduced. However, the useful area in which the tests can be performed is still sufficient to be able to perform the tests on a conventional smartphone.

It has been established that the canvas screen is in landscape format, with a height of 68 mm and a width of 120.9 mm, thus obtaining the commonly used 16:9 format. Since the original tablet has an A4 format (210 x 297 mm), the distribution of the smartphone within what is the area of the tablet is located in the location shown in Figure 3.2.

Taking into account the previous distribution, some changes have been made in the calculation of the dispersion_index and gmrt_on_paper variables. For the dispersion index, those boxes that are occupied have been calculated and have been divided between the total boxes of the A4 sheet, not between those of the canvas. In addition, a size of the boxes of 4 px has been established. For the GMRT, the distance to the reference point is not the top right corner of the canvas, but the top right corner of the A4 sheet. In addition, the constant d used is 4.

3.3 Predictive Models

This section describes the process of creating predictive models. Mainly, the Python package called Scikit-learn [89] has been used, which contains a multitude of tools for data mining and machine learning. All the code corresponding to the creation, training, testing, and selection of the best predictive models can be viewed in *models.py*, which can be accessed at Appendix B.2.

Initially, the data from the DARWIN dataset [5] were loaded from a CSV file, downloaded directly from the UCI Machine Learning Repository. Variables already mentioned in previous sections, as well as identifiers, were removed. Subsequently, in order to reduce bias due to randomness, 20 random partitions of the data were made into training and test data sets, along with their respective diagnostic labels. A 25% of the data was used for testing, that is, of a total of 174 instances, 130 were used for training and 44 for testing. In addition, to

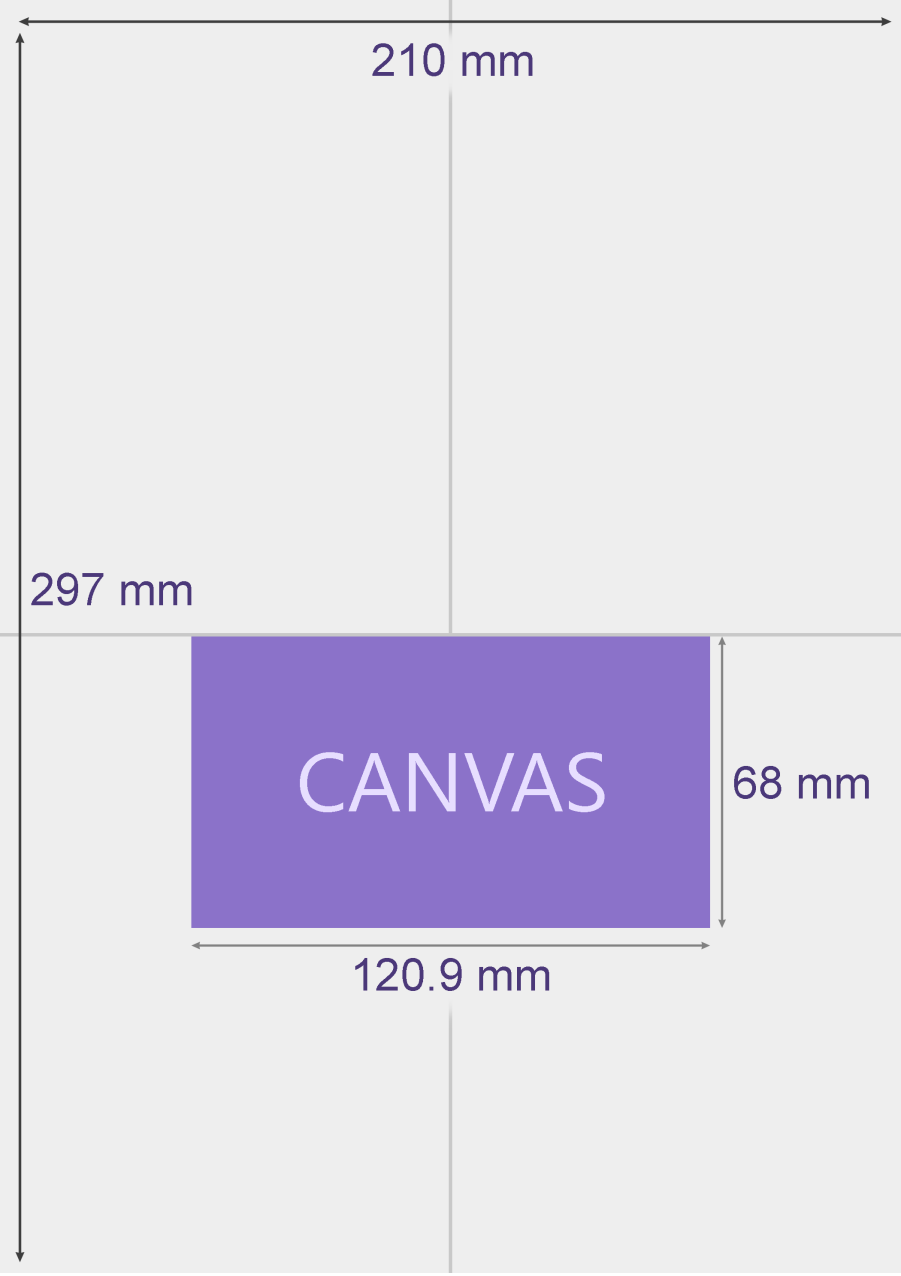


Figure 3.2: Distribution of the canvas within the virtual space it would occupy on the original studio's tablet.

obtain a homogeneous distribution of the samples of both healthy users and those with AD, the partitions were stratified to obtain the same original class proportions.

Within the Scikit-learn package, there are several machine learning models available. In this study, some of them that were used in the original study will be used, such as Random Forest, Logistic Regression, K-Nearest Neighbor, Linear Discriminant Analysis, Gaussian Naive Bayes, Support Vector Machine, Decision Tree, and Multilayer Perceptron. In addition, it has been decided to add Extremely Randomized Trees and other classifiers based on boosting such as Adaptive Boosting, Gradient Boosting, and Extreme Gradient Boosting. Next, each of the models used is described:

- **Naive Bayes Classifier (NB)**: This is a specific type of Bayesian Network, which is a probabilistic graphical model. The Naive Bayes Classifier consists of directed acyclic graphs that encode a joint probability distribution over a set of random variables. Each graph has a single parent node (which represents the unobserved node) and several child nodes (which correspond to the observed nodes). This model makes a strong assumption of independence between the child nodes in the context of their parent node. Therefore, this model is based on estimation [90, 91]. Gaussian Naive Bayes (GNB) models are the same, only that the features in the dataset are normally distributed.

- **Support Vector Machine (SVM)**: This model uses decision planes to separate objects of different classes in the feature space. Its goal is to find the Maximum Margin Hyperplane (MMH), which is the class dividing line with the widest margin, thus reducing the classifier error [19, 92].

- **Logistic Regression (LR)**: This algorithm employs the logit transformation and the logistic curve to model class probabilities. It is used to obtain predictions in binary categorical results [93, 94, 19].

- **Linear Discriminant Analysis (LDA)**: This is a generalization of Fisher's linear discriminant method that allows obtaining the linear combination of features that best separates the instances of a dataset into two or more classes [95, 96, 97].

- **Decision Tree (DT) Classifier**: This model uses a tree structure for decision-making. The internal nodes represent attribute tests, the branches represent the test results, and the leaf nodes represent class labels. The split criterion is based on normalized information entropy [98, 19, 99].

- **Random Forest (RF)**: This ensemble method focuses on ensembles of decision trees. After generating the ensemble of trees (the forest), the model uses a vote to combine the tree predictions. There is a version of this method called ExtraTrees Classifier (ET) that introduces more randomness by randomly selecting the cut points and attributes [100, 19, 101].

- **Multilayer Perceptron (MLP)**: This type of artificial neural network (ANN) is a feedforward type, meaning that the flow of information is unidirectional. It is often considered the basic form of neural networks [102]. ANNs were intentionally designed to resemble the functioning of neurons in the brain [19]. Their structure consists of a series

of simple processing elements, called "neurons", which are highly interconnected and organized in layers. These layers can vary in both the number of layers and the number of neurons per layer [5]. They also present activation functions that can be sigmoid, hyperbolic tangent, or Rectified Linear Unit (ReLU). Neural networks start from what is known as the *input layer* and end with the *output layer*. Modern feedforward networks are trained using the backpropagation method [103].

- **K-Nearest Neighbors (KNN)**: This algorithm is a widely used and well-documented non-parametric method, recognized for its simplicity in the field of ML. This algorithm can be applied in both classification and regression tasks [104]. The fundamental premise of KNN is that similar observations are close to each other in the feature space. Therefore, to classify an instance into one category or another, KNN assigns the category based on the majority vote of the k nearest neighbors in terms of distance. The value of k must be determined by the user. It is considered a "lazy learner" because it does not learn anything, as there is no abstraction and it merely stores the training data verbatim without building a model [19].

- **Boosting Classifiers**: These types of classifiers fall within the category of ensemble learning methods. Within this group are methods based on Bagging where the weak learners are trained in parallel, an example of this is the Random Forest, while in Boosting classifiers they are trained sequentially. In this way, Boosting adjusts the weights with each iteration to improve future performance [19, 105]. Within this category are models such as:

  - **Adaptive Boosting (AB)**: One of the first to be introduced and stands out for being one of the simplest. In this method, an approach is applied to generate weak learners that iteratively identify the data points that have not been correctly classified and adjust to minimize the training error. The model continues the optimization sequentially until the strongest predictor is achieved [106, 19].
  - **Gradient Boosting (GB)**: Models based on gradient boosting work by sequentially adding predictors to an ensemble, each correcting the errors of its predecessor in the same way as AdaBoost. However, it differs in that instead of adjusting the weights of the data points, gradient boosting is based on the residual errors of the previous predictor. Hence its name, due to the combination of gradient descent and the boosting method [107].
  - **Extreme Gradient Boosting (XGB)**: This is a more efficient and scalable implementation of gradient boosting. XGBoost provides parallel tree boosting, also known as Gradient Boosted Decision Trees (GBDT). Due to its ability to take advantage of multiple CPU cores, it allows for parallel training, resulting in estimates that are 10 times faster than Gradient Boosting. It can be used for regression, classification, and ranking tasks [108].

In addition, to allow each classifier to work best in its configuration, a 5-fold cross-validated grid search was performed in order to obtain the best hyperparameters for the classifier. In addition, each of the folds was scaled using MinMax Scaler, as it is not advisable to use the

StandardScaler because the data does not follow the normal distribution it assumes. The grid used for each of the models can be seen in Table 3.2.

It is worth mentioning that, in order not to repeat in the future the steps of data partitioning, training, and testing, which take a long time, the Joblib library [109] has been used. This library allows to store any internal variable of the code, which in this case has served to save the partitions and the models generated on the computer's disk to be able to recover them later.

Before going into the details of how the models were obtained, it would be helpful to first present a scheme that summarizes the different types of models created:

- **Single Classifier Models**
  - One type of classifier
  - Trained with the total predictive variables (152)
- **Task-Specific Classifier Models**
  - One type of classifier
  - Trained with task-specific predictive variables (8)
- **Task-Specific Classifier Combination Models**
  - Formed by the combination of 19 Task-Specific Classifier Models
  - Final prediction based on the majority vote of models
  - Two types:
    * Single: Formed by small models of the same type of classifier
    * Mixed: Formed by the best models for each task, regardless of the type of classifier
- **Top N Classifiers Combination Models**
  - Formed by the combination of the N best Single Classifier Models
  - Performed for N=3 and N=5
  - Final prediction based on the majority vote of models
  - Two types:
    * Single: Formed by small models of the same type of classifier
    * Mixed: Formed by the best models for each task, regardless of the type of classifier

The training was carried out in two phases. Firstly, models were trained with the entire set of predictor variables, called *single classifier models*, and secondly, models were trained with task-specific variables for each of the tasks, called *task-specific classifier models*. With this, a total of 240 models (12 classifiers x 20 partitions) were obtained in the first case and 4560 models (12 classifiers x 20 partitions x 19 tasks) in the second. As each model finished training,

Table 3.2: Hyperparameters ranges explored during the grid search for each classifier.

Classifier	Parameter	Values
RF	n_estimators	50, 100, 150
	criterion	gini, entropy
	min_samples_leaf	1, 3
	bootstrap	True, False
LR	C	0.001-5.00, step:0.01
	max_iter	800
	solver	liblinear, lbfgs
KNN	n_neighbors	3-48, step: 5
	weights	uniform, distance
	algorithm	ball_tree, kd_tree, brute
LDA	solver	svd, lsqr, eigen
	shrinkage	None, auto
GNB	priors	None, [0.0, 1.0], [0.1, 0.9], [...], [0.9, 0.1], [1.0, 0.0]
SVM	kernel	rbf, linear, poly, sigmoid
	C	0.1-1.55, step:0.05
	gamma	scale, auto, 0.5
DT	criterion	gini, entropy
	splitter	best, random
	max_depth	None, 2, 5, 10
	min_samples_split	2, 3, 5
	min_samples_leaf	1, 2, 5, 10
	max_leaf_nodes	None, 2, 5, 10
MLP	activation	relu, logistic
	hidden_layer_sizes	5, 15, 30, 100
	max_iter	1000
	solver	sgd, adam
ET	n_estimators	50, 100
	criterion	gini, entropy
	min_samples_split	2, 4
	min_samples_leaf	1, 3
	bootstrap	True, False
AB	n_estimators	50, 150, 300
	learning_rate	0.01, 0.1, 0.5, 1.0
GB	n_estimators	50, 150
	subsample	0.7, 1.0
	max_depth	None, 3
XGB	max_depth	4, 6, 10
	eta	0.1, 0.3
	subsample	0.7, 1.0
	colsample_bytree	0.7, 1.0
	colsample_bynode	0.7, 1.0

the testing phase was also carried out immediately afterwards. In this way, the corresponding test data set was used to obtain predictions that were compared with the actual labels. Thus, performance metrics were obtained, such as accuracy, sensitivity, specificity, precision, recall, F1 score, and Cohen's kappa coefficient. These metrics were stored in a data frame along with model information so that, when the training was finished, they could be saved in a CSV file. In this way, the metrics obtained from all the models could be analyzed, both for the single classifier for the entire data set and for those specific to the task. The combined duration of training and testing lasted 1 hour and 42 minutes and 10 hours and 55 minutes respectively, although these can vary depending on the specifications of the equipment used.

The next step consisted of creating additional models from the combination of the previous models. Firstly, two different types of models were generated, formed by the combination of the task-specific classifier models. The so-called *single classifier task-specific models* are formed by 19 small models of the same classifier that to generate the final prediction are subjected to a vote and the class that has been predicted the most times is the final diagnosis. The *mixed classifiers task-specific model* has also been created, which unlike the previous one uses the best task-specific classifier for each of the tasks, resulting in a mix of 19 different types of classifiers. Secondly, models were generated based on the combination of the best single classifier models. To do this, all of these were first sorted from highest to lowest F1-score and the top 3 and top 5 models were selected, obtaining the *top3_mixed_classifiers* and *top5_mixed_classifiers models*. Like the initial models, these combination models have been tested through the test phase for the 20 partitions, thus obtaining the performance metrics of these. In the results section later on, the metrics of all the models obtained are specified and the reason for the choice of the model that has been used to predict the diagnosis of the user who takes the test in the application is explained, but for now here it is only worth mentioning that the model finally used is the *top3_mixed_classifiers*.

Finally, once the predictive model of the app was established, a function was created responsible for receiving the input of the list with the user's data when taking the test and making the predictions. These would be made in the same way as explained above, the models are adjusted and the user's class is predicted for each of the classifiers as well as for each of the N splits with the highest accuracy. It is recommended that the number of votes be odd to avoid ties between the two classes, and that as many splits as possible be chosen to increase the robustness of the model. Therefore, 19 splits have been chosen. Each model then generates a prediction from the total of its 19 predictions and the final prediction is the one in which 2 or 3 models coincide. An example of how the predictions would be made in a case of a user with AD can be seen in Table 3.3. In this case, the GaussianNB model is not able to correctly predict AD (1) while the other two models are, so the final prediction is AD.

Table 3.3: Example of how the final prediction for the user data would be performed using the *top3_mixed_classifiers*.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	prediction
GaussianNB_5	0	0	0	0	0	0	1	1	0	0	1	0	0	1	0	0	1	0	0	0
RandomForestClassifier_13	1	1	1	1	1	1	1	0	0	1	0	1	1	1	1	1	1	1	0	1
RandomForestClassifier_15	1	1	1	1	1	1	1	0	0	1	0	1	1	1	1	1	1	1	0	1

3.4 Web Application Development

This section explains the development of the web application, which started from a prototype of the canvas made with the Tkinter module. However, this option had to be discarded for the web development. The Flask package [110] from Python has been primarily used to develop the Back-end of the application, that is, its internal operation. For the Front-end, which is the part of the application that the user sees and interacts with, HTML, CSS, and JavaScript have been used. HTML provides the basic structure of the web page, CSS gives style and appearance to the HTML content displayed on the screen, and JavaScript allows the web page to be interactive and dynamic, enabling real-time interaction.

3.4.1 Application Scheme

The first step in creating the application was to design a scheme, which served as a basis for code development. The structure used is as follows:

- Perform Test
- Log In
  - Log in as User
    - Perform Test as User
  - Log in as Administrator
    - Register User
    - Consult User
    - Collect new data

Upon entering the application, the user has two options: perform the test directly, obtaining a prediction of the diagnosis at the end, or log in. To log in, two options are offered: log in as User or as Administrator.

The User would be a patient or a person who wishes to undergo study and monitoring over a period of time by performing the test. They only have to enter their ID and their provided password to start the test and get a prediction at the end. Unlike the previous case, at the end of the test, the data related to the ID, the date it was performed, the prediction obtained, the duration of the test, the data obtained with which the prediction has been made with the chosen predictive model and the percentage of predictors that have given an AD diagnosis for each case are stored in a database.

On the other hand, the Administrator interface is intended for doctors, health professionals, or authorized persons. This profile is responsible for tasks such as registering users, including name and surname, assigning a password, ID, and if the patient has a confirmed diagnosis, which can be Without Diagnosis, Healthy, or with AD. The Administrator can also consult a user's data through their ID, accessing their personal information as well as a record in table format of the data from all the tests they have previously performed, facilitating medical follow-up. The last functionality of the Administrator is the collection of new data to generate a new

exclusive database of the application. For this, the Administrator must be physically present with the user who is going to take the test, the ID is entered and if the user is healthy or presents AD and the test is started. At the end of the test, the data is automatically stored in a database along with an ID and its corresponding class label.

3.4.2 Back-end

For the Back-end of the application, the development began with the format established for Flask applications. This consists of an instance to create the application at the beginning, another at the end to run the application, and in between these are the Flask route decorators that assign a URL to a certain function that is found within them. An example of this is shown below:

```
# Creates the app
app = Flask(__name__)
# Defines a route and its corresponding controlling function
@app.route('/')
def home():
    return "Hello, World!"
# Runs the app
if __name__ == '__main__':
    app.run()
```

The development of the application began with the implementation of the canvas functions. The Canvas, which is the area where tasks are performed, can have its background altered depending on the specific task in progress. User interaction with the Canvas is managed through mouse or tactile events, but from now on, it will only be mentioned in the context of stylus use. These events include touching the screen, moving the stylus while maintaining contact, and releasing the stylus from the screen.

The user begins with Task 1. Upon touching the screen, the analysis begins. Drawing in the application is done by tracing lines between two points continuously while holding down the stylus on the screen. Thanks to the high refresh rate, curves can be drawn smoothly. Each time the screen is touched, a *pendown* counter is incremented. It also records the time elapsed from when a *pendown* is performed until the stylus is released stops touching the screen. The sum of all these times at the end of the task is recorded as *paper_time*. In addition, the Euclidean distances between the points of the same drawn segment are recorded and the sum of these distances is the total distance traveled. If this total distance is divided by the *paper_time*, the *mean_speed_on_paper* is obtained. The *air_time* is the result of subtracting the *paper_time* from the *total_time* (time elapsed from the first *pendown* until the next task button is pressed). Since the X and Y positions of each point are also being recorded, it is easy to determine which is the closest and furthest point from the X and Y axes, which allows obtaining the *max_x_extension* and *max_y_extension* by the difference between these two points. To calculate the dispersion index, two zero matrices are created, one for the Canvas and another for the original tablet extension, whose dimensions are the height and width in pixels divided by the established box_size constant. Subsequently, in the Canvas, those positions where it has been

drawn are replaced by a 1. The dispersion index (*disp_index*) is then obtained by adding all the values of the Canvas matrix and dividing the result by the total size of the tablet matrix. In addition, at each point, the distance from the top right reference point is calculated and stored in a list. Subsequently, the above formula is applied to calculate the absolute value of the difference between two radii taken in d measurements and then obtain the *gmrt_on_paper*.

Apart from the Canvas, a large button is included to switch tasks. This button performs all the necessary calculations for the previously mentioned variables, and adds these variables to a final list as the test progresses. This list will later be used to obtain a diagnosis using predictive models. When this button is pressed, some internal variables are reset, the drawings on the screen are erased, and the background of the task changes to the next statement.

In addition, additional functions have been added such as functions in charge of checking IDs, checking if the fields entered when logging in are correct, functions to store data whether they are from registration or obtained from the test and also to obtain result tables.

It's important to note that the canvas used both at the start of the session and without logging in are the same. However, when not logged in, since the user will not be registered in the app, there will be warning messages at the beginning and end of the test. These messages indicate that the prediction may be erroneous and that, if deemed necessary, it is advisable to contact a health professional. On the other hand, the canvas used to collect data does not have adjustments to resemble the data from the original study, which is ideal for creating a new database. That's why it has specific functions to differentiate itself.

All the Back-end code can be seen in detail in the GitHub repository of this project, specifically in the code file *app.py*, which can be accessed at Appendix B.2.

3.4.3 Front-end

In this section, the code used to give the structure and design to the application that is visible to the user is discussed.

Through HTML code, the different parts of the application have been established based on the scheme provided above. The image of the application logo, a container for the different canvases, several buttons to start tests, sessions or select administrator functions have been established. Forms have also been included to log in or register a user, in addition to a table element to display the results of the tests.

This HTML code is linked to a CSS style sheet that gives the page a simple and clean design with blue tones. The elements are centered and the buttons and texts are large to facilitate reading. There is the canvas to draw, several buttons and forms to interact with the application, and a table to display information about the users and the tests they have performed.

On the other hand, the same HTML code is linked to a JavaScript script, which is responsible for handling the interactivity of the application. In general, the script is mainly responsible for event handling, that is, events are triggered by user actions, such as tapping a button, and the script responds to these events by executing specific functions. Another script has also been included separately that is exclusively responsible for the three available drawing canvases, since each one is initialized with a specific button and handles different functions. It is here where it was included that when drawing on the canvas it could be used both with the click of the computer and touching the screen on touch devices.

All codes related to the Front-end of the application are also available in the project's

GitHub repository, which can be accessed through Appendix B.2. Specifically, they can be found in the *index.html*, *styles.css*, *script.js* and *canvasFunctions.js*.

3.4.4 Application Name and Logo Design

The name of the application, *AlzheimInk*, was selected with the aim of combining two concepts related to the application. A part of the word *Alzheimer* was used, in reference to the disease that is predicted, and the word *Ink*, in relation to the use of writing or drawing. The resulting combination, *AlzheimInk*, stands out because the capital letters form AI (Artificial Intelligence), a reference to the machine learning mechanism that underlies the application. As for the logo, a design of a pencil drawing a brain that is fading was devised, following a minimalist style. This design seeks to relate the concepts of the application. It was made using the Adobe Photoshop graphic editing tool. The logo can be seen in Figure 3.3.

Figure 3.3: Logo for the AlzheimInk application.

3.5 Exposure of the application on a web server

Once an application was developed and verified on a computer's localhost, acting as a virtual server, various options were explored to make the application accessible via the internet from any device. Initially, the deployment of the application on the internet was considered through a web hosting service. However, due to the limitations of free hosting services in terms of server space and the maximum number of requests, and the fact that it did not function correctly in some instances, the decision was made to use ngrok, a platform offering various services for web developers.

Ngrok enables the exposure of a computer's localhost to the public network using a protocol known as tunneling. This protocol establishes an HTTPS tunnel between a computer's localhost and ngrok's servers, which publish the localhost on a specific web domain and redirect all traffic arriving at the server from the domain to the computer.

The first step in this process involves installing the ngrok executable on the computer. Upon logging in, ngrok provides a command to add an authentication token to the ngrok.yml configuration file, thereby linking the account and the computer.

After purchasing a user plan, ngrok provides a static domain. The execution of a single command from the terminal, specifying the provided domain and the port where it is located (5000 for Flask applications), initiates the tunneling.

Thus, when the tunnel is active and the application is running from the computer, access to the web application is possible through the link. To facilitate access to the web page from a mobile device, a QR code has been generated that users can scan with their camera, which will redirect them directly to the web.

Chapter 4

Results

4.1 Models performance metrics

4.1.1 Single Classifier Models

Firstly, for the models that were trained with all the predictor variables with a single type of classifier, that is, the *single_classifier_models*, the Table A.5 has been created where a summary of the average precision (along with its standard deviation), specificity, and sensitivity obtained from the 20 models of each of the classifiers is shown. To display it graphically, a boxplot has been created in Figure 4.1 where the accuracies of the models can be summarized.

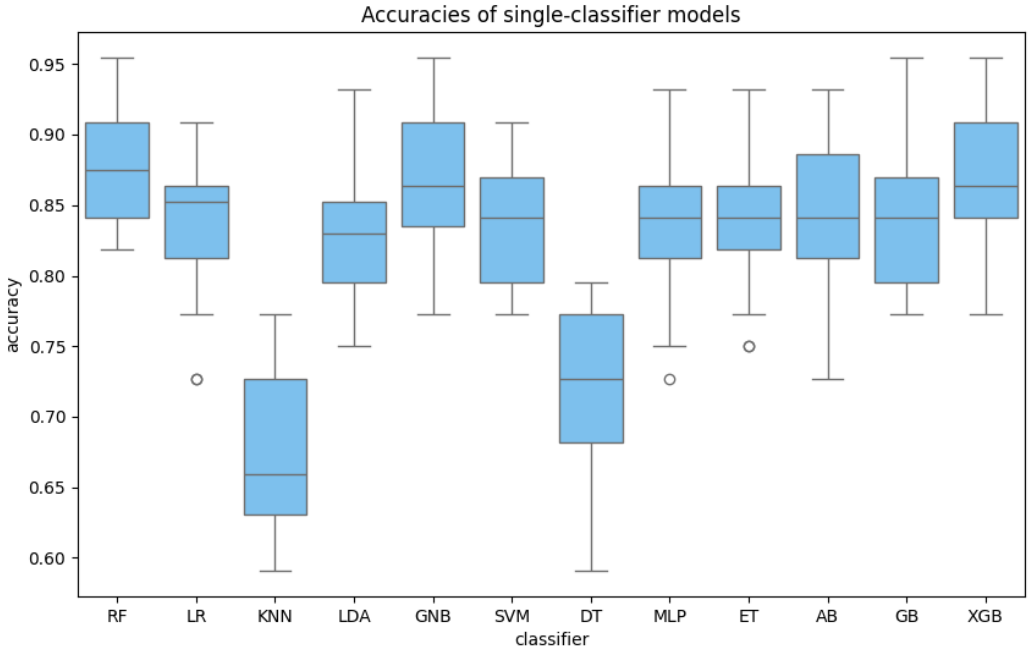


Figure 4.1: Boxplot representing the distribution of accuracies for the *single_classifier_models* based on the classifier.

As can be observed, in general, accuracies greater than 80% have been obtained in almost all of the classifiers except for the KNN and DT models. On the other hand, the three classifiers that seem to have shown the highest performance have been the RF, GNB, and XGB. Upon performing a Friedman test for the accuracy values among the different classifiers and a subsequent post-hoc Nemenyi test, the Table A.2 was obtained. Here it was verified that there are mainly significant differences in the KNN and DT models compared to all the other models.

4.1.2 Task-Specific Classifier Models

Secondly, it is necessary to discuss the performance metrics obtained for those task-specific models. As in the previous case, in the corresponding table, the accuracy values and typical deviation of the different classifiers are also represented and broken down in the corresponding table, the values of specificity and sensitivity, in both cases divided by tasks. As can be observed in Tables A.6 and A.7, their values have been considerably reduced as expected since these models are only trained with 8 variables. For example, in the accuracies, most average values are around 65-70%. To check if there were significant differences between the different classifiers at the task level, various Friedman test were performed, finding that there are significant differences between the models of different types of classifiers within the same task except for the models of task 12.

A breakdown of the previous table has also been made into two boxplots. In the first one (Figure 4.2), the accuracy for the different models can be observed according to the classifiers without taking into account the task. It can be observed that the distributions are similar and there are no apparent differences between the classifiers. Upon performing a Friedman test and the subsequent Nemenyi test (Table A.3), it was verified that there were mainly differences against other models in the RF, DT, ET, and GNB models. In the second boxplot (Figure 4.3), the distribution of the accuracy for the models based on the tasks is observed. It can be observed how here the models based on different tasks show that there are some with higher accuracy and others less. To confirm that there are significant differences between the different classifiers, a Friedman test and a subsequent Nemenyi test (Table A.4) were performed. In this, it was observed that in general, the models of tasks 1, 8, 15, 17, and 18 presented significant differences compared to many of the other models, with the latter four being the ones that present the highest accuracy.

4.1.3 Task-Specific Classifier Combination Models

The performance metrics of the single_classifier_task_specific_models and the mixed_classifiers_-task_specific_model can be observed in detail in Table A.8, although a summary of the precisions of the models has been made, which is shown in Figure 4.4. On the X-axis, the classifiers are shown, with the last one called MIX, for mixed classifiers. Specifically, the MIX models have been formed from the combination of the best task specific models per task shown in Table 4.1.

In general, average accuracies of mostly between 75-85% have been obtained. The KNN and DT models have improved compared to the single_classifiers of the previous section, while GNB has decreased. The best model in this case is the mixed_classifier_task_specific_models with an average accuracy of 85%.

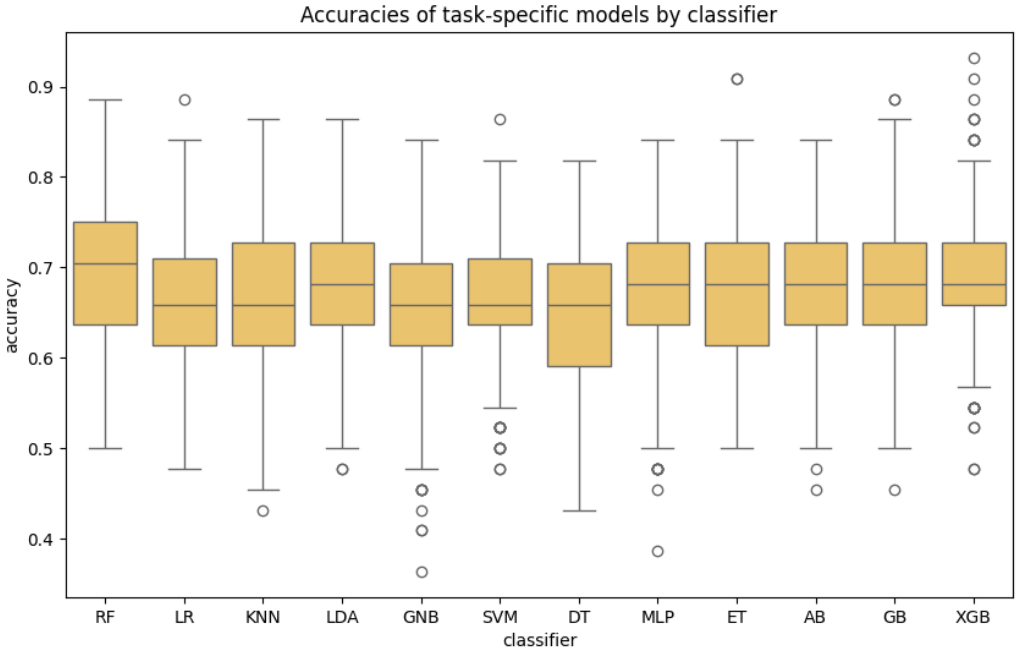


Figure 4.2: Boxplot representing the distribution of accuracies for the *task-specific* models based only on the classifier.

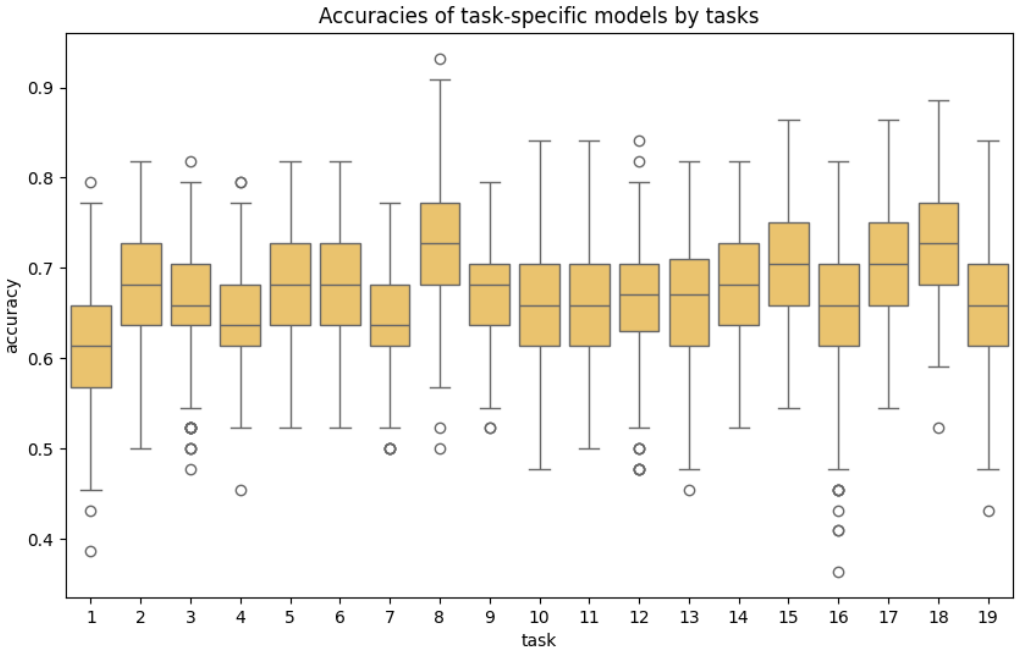


Figure 4.3: Boxplot representing the distribution of accuracies for the *task-specific* models based only in the task.

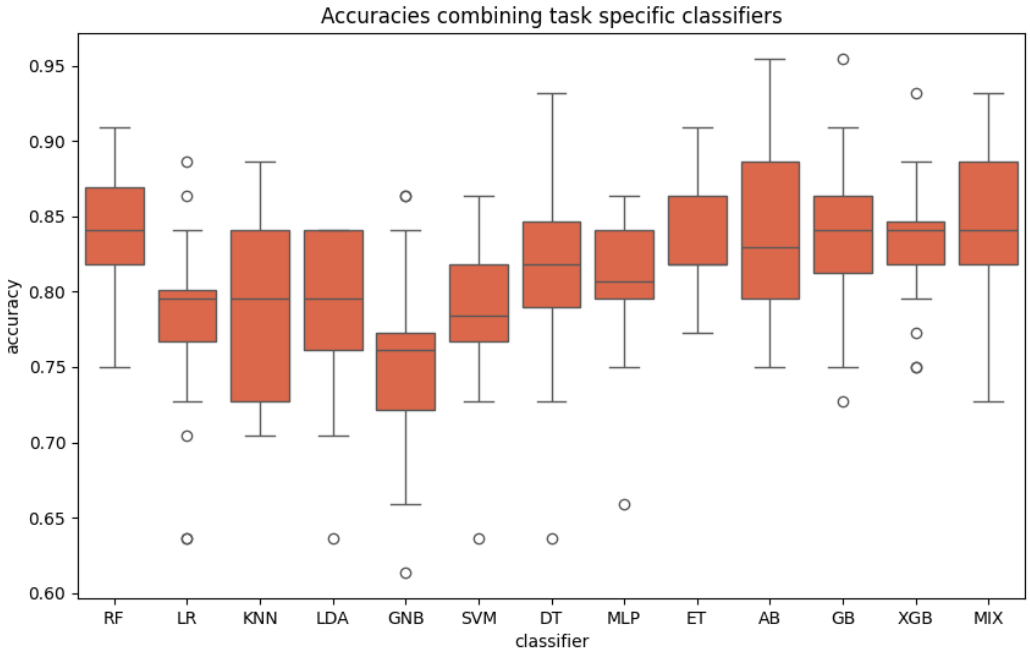


Figure 4.4: Boxplot representing the distribution of accuracies for models based on combining *task-specific* models.

Table 4.1: Best *task-specific* models and its accuracy.

# Task	Classifier	Accuracy
1	ExtraTreesClassifier_5_1	77.27
2	GradientBoostingClassifier_5_2	81.82
3	LinearDiscriminantAnalysis_9_3	81.82
4	MLPClassifier_4_4	79.55
5	RandomForestClassifier_15_5	81.82
6	DecisionTreeClassifier_2_6	81.82
7	LogisticRegression_12_7	75.0
8	ExtraTreesClassifier_13_8	93.18
9	GradientBoostingClassifier_5_9	79.55
10	LogisticRegression_20_10	84.09
11	GaussianNB_15_11	84.09
12	MLPClassifier_5_12	84.09
13	DecisionTreeClassifier_12_13	81.82
14	MLPClassifier_6_14	81.82
15	SVC_15_15	86.36
16	SVC_15_16	81.82
17	ExtraTreesClassifier_5_17	86.36
18	LogisticRegression_1_18	88.64
19	LogisticRegression_8_19	84.09

4.1.4 Top N Classifiers Combination Models

For both the single-classifier models and the MIX models, the first step was to sort the models based on the F1-score and select the top N models in both cases. As an example, the top 5 single-classifier models for the MIX models are shown below, all with an accuracy of 95.45%:

1. GaussianNB()
2. RandomForestClassifier()
3. RandomForestClassifier(bootstrap=False, criterion='entropy', n_estimators=50)
4. GradientBoostingClassifier(max_depth=None, n_estimators=50, subsample=0.7)
5. ExtraTreesClassifier(min_samples_leaf=3,min_samples_split=4)

It should be noted that in the first two models, their best hyperparameters are the ones that come by default while the others do have other hyperparameters obtained from the grid search performed. Table A.9 shows the performance metrics for the single and mixed classifiers of the top 3 models, and Table A.10 shows the performance metrics for the top 5 models. Two boxplots have also been made to get a better view of the results (Figures 4.5 and 4.6).

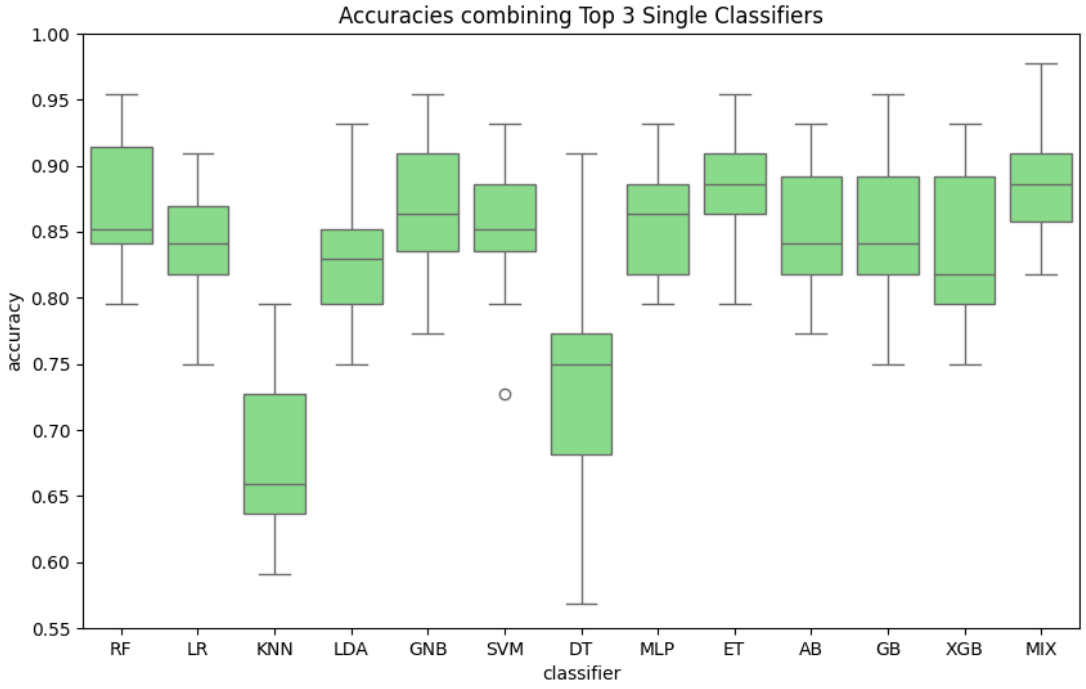


Figure 4.5: Boxplot representing the distribution of accuracies for the *top3_single_classifiers*.

It can be observed how the results are very similar except for the DT models that have improved with the 5 top classifiers compared to the 3. In general, the average accuracies seem to be distributed around 85% and the best models seem to be those formed by ET, RF, and MIX models, all around 88%.

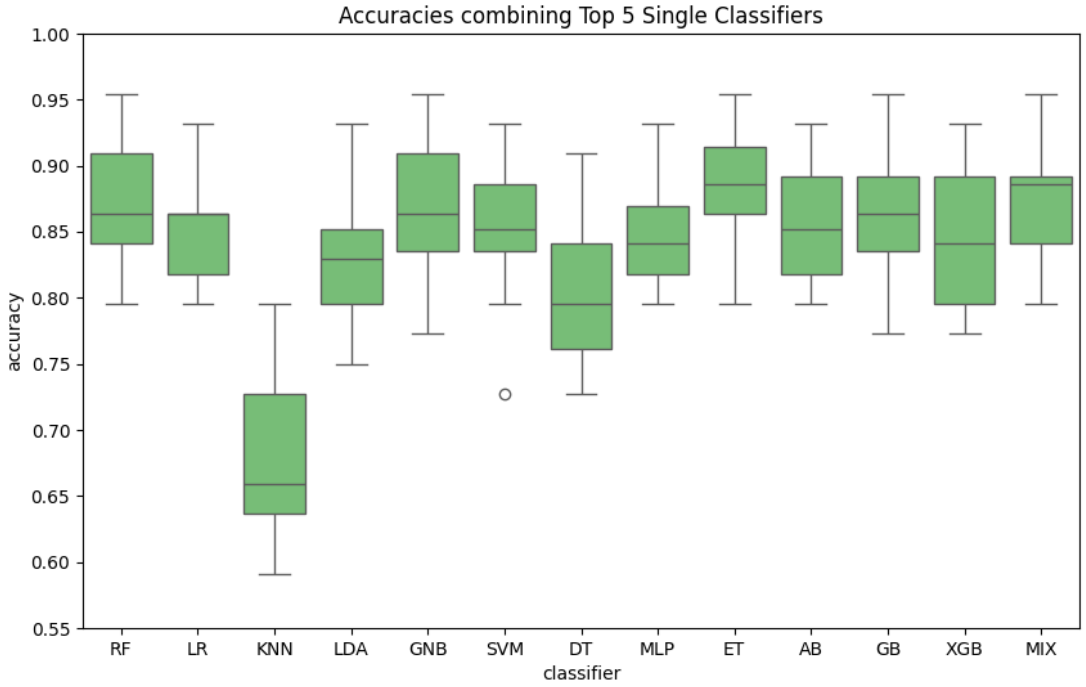


Figure 4.6: Boxplot representing the distribution of accuracies for the $top_s ingle\_classifiers$.

4.1.5 Comparison of the Models Obtained

Finally, Table 4.2 has been created to provide a summary of the average accuracies for each type of model, including those from the original study for comparison.

Table 4.2: Mean accuracy (in %) of all models comparison against the models from the original study.

	RF	LR	KNN	LDA	GNB	SVM	DT	MLP	ET	AB	GB	XGB	MIX
single_classifiers_orig	88.29	81.86	71.43	72.14	85.00	79.00	78.57	83.14	-	-	-	-	-
comb_task_specific_orig	88.57	85.71	85.71	77.14	82.86	88.57	94.29	88.57	-	-	-	-	91.43
single_classifiers	87.61	83.75	67.16	83.07	86.93	83.98	71.48	83.41	87.05	84.09	84.89	83.98	-
comb_task_specific	84.20	78.07	78.98	78.41	75.23	78.52	81.36	80.68	83.30	84.55	83.86	83.18	85.00
top3_single_classifiers	87.27	84.43	67.73	83.07	86.93	85.45	72.95	85.23	88.75	85.23	85.91	84.20	88.52
top5_single_classifiers	87.39	84.89	67.73	83.07	86.93	85.57	80.11	84.55	88.64	85.68	86.36	84.20	87.27

It can be observed that, when comparing the original *single_classifiers* and those that have been created in this study, the values are very similar to each other, surpassing in some cases the accuracy of the original. A Wilcoxon test was performed and it was corroborated that there were no significant differences between the averages of the models from the original study and those developed in this one. On the other hand, for the models formed by the combination of task-specific models, there seem to be more differences, with the models developed having lower accuracy than the original models. To corroborate this, another Wilcoxon test was performed, confirming that there were significant differences between these. The best model of this type

was the MIX, formed by the best predictors for each task, with an accuracy of 85%, compared to the 91.43% of the original. For the models through the voting of the three and the five best *single_classifiers*, in general, in both cases, precisions have been obtained that are somewhat better than simply with a single *single_classifier model*. Of the *top3*, the ET, RF, and MIX models stand out with accuracies of 87.27, 88.75, and 88.52 respectively. Of the *top5*, the same ones stand out as in the previous case but with accuracies of 87.39, 88.64, and 87.27.

Therefore, for the choice of the predictive model that will be used in the web application, the dispute was between the ET of *top5* and *top3*, and the MIX of *top3*. Since the three present very similar distributions and averages of accuracies, the MIX model of *top3_single_classifiers* has been chosen. This model is simpler than those of *top5*, presents minimum and maximum accuracies higher. In addition, being formed by two RF models and a GNB, it allows focusing the predictions in a more robust way and not so influenced by a single type of classifier, being able to capture different patterns and avoid errors. It is also worth mentioning that the models that were wanted to be introduced as the ET, AB, GB, and XGB have not had a performance especially different from the other models in general, perhaps a little more the ET. In general, their accuracies vary between 84-88%, so they are good models, but not better in terms of accuracy.

4.2 Web Application

The final outcome of the project is a fully functional web application. It can be accessed either via a link provided in Appendix B.1 or by scanning the QR code displayed in Figure 4.7.

Figure 4.7: QR code to access AlzheimInk application.

Next, several screenshots of the web application taken from a smartphone are presented. As can be seen in Figure 4.8, the main page of the created website displays the option to either take the test directly or log in.

If the test is taken logging in, it would be displayed on the screen as shown in Figure 4.9. The top part of the screen displays an example of a writing task and a drawing task. The bottom part of the screen displays what would be obtained at the end of the test while predictions are being made.

On the other hand, when logged in as an administrator, access is provided to functionalities such as Register User (Figure 4.10), Consult User (Figure 4.11), and Collect New Data (Figure 4.12).

Figure 4.8: Application home page.

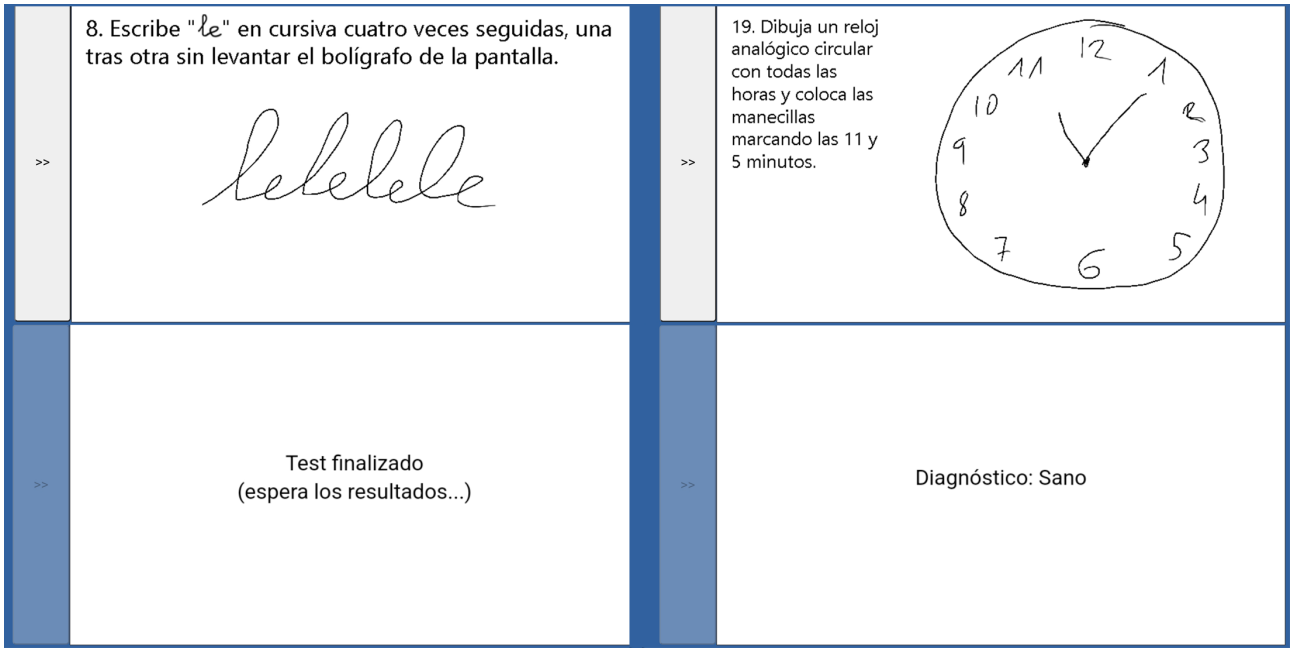


Figure 4.9: Example of how the tests would look in the canvas, specifically tasks 8 and 19 in the upper part. At the bottom is the prediction at the end of the test.

Figure 4.10: Administrator tool to register users in the application.

The application does not have a back button to navigate between different steps, so it is recommended to reload the page if one wishes to go back. Additionally, to test the functionalities of the application, a user (12345678A) and an administrator (12345678B) have already been registered, both with the password 123.

Figure 4.11: Administrator tool to consult the personal data and previous tests of registered users.

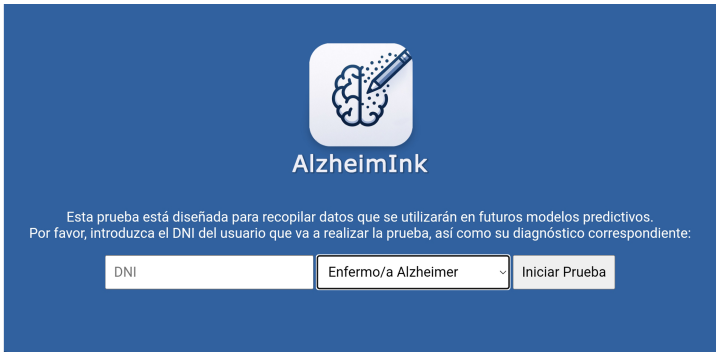


Figure 4.12: Administrator tool for collecting user data through a new test with the purpose of generating a new database exclusive to the application.

Chapter 5

Discussion

The purpose of the development of this project was to adapt a study that examined the possibility of diagnosing Alzheimer's disease through the analysis of handwriting using machine learning-based predictive models. Instead of using a specific graphic tablet as in the original, this project aimed to verify the feasibility of transferring this tool to a globally extended interface such as smartphones.

The result obtained from this project is a mobile web application available through the internet. With a capacitive stylus, a series of writing and drawing tests can be performed on the screen and, upon completion, a positive or negative diagnosis for AD is obtained with an estimated theoretical accuracy of 88%.

Regarding the predictive models obtained, it is worth noting that, despite the drastic reduction in the number of variables per task and the elimination of some tasks from the original study, models with similar performance have been achieved. An example of this are the single_-classifier models, which do not present significant differences with those of the original, despite these last ones having 450 predictor variables and those of this project 152. On the other hand, the predictive models based on the combination of task-specific models were significantly different from the originals, with worse performance. This may be due to the fact that it is much more difficult to extract accurate predictions from 8 predictor variables compared to the 18 of the original. That is why if the smaller models are worse, added to the fact that more important tasks for the final prediction may have been eliminated, it may have originated that when combined in the final model, this model is not as accurate. Regarding the models created with the top single classifiers, add that the best performance of these may be due to the fact that, through the combination of predictions by voting, a series of improvements are translated. Since the models are different from each other, either by hyperparameters or by classifiers, different ways of interpreting the data to obtain the predictions are obtained, helping to have a more robust model with a reduction of errors, variability, and overfitting.

Furthermore, it's been observed that the new classifiers, which were not part of the original study, have proven to be effective in generating competent models. However, they do not significantly outperform the others.

This project has significant implications as a potential support tool for the diagnosis of AD. Despite hardware limitations, it has been demonstrated that the project can generate theoretical predictions with precision comparable to other conventional methods. Furthermore, this is a relatively simple, fast, inexpensive, non-invasive test that is available online, making it

a beneficial resource for health services once an improved version of the application is developed.

The discussion now turns to the current limitations of the project. The primary issue lies in the discrepancy between the application and the methods used to collect the original data for model training, compared to those used for this project. Despite efforts to closely imitate the original, there are inevitable differences, some of which may not be immediately apparent. These differences could potentially bias the collected results. To address this, a function has been implemented in the application to allow independent testing and data collection, separate from the original dataset. The goal is to create a new dataset from scratch, exclusive to the language in which it is performed. This task, however, can only be performed by individuals with administrator access to the app.

Another limitation relates to the perceptual differences between writing on paper with a pencil or pen and writing on a screen with a stylus. Even though the recommended capacitive stylus type offers some resistance to the stroke, the sensation is not identical. Additionally, small strokes like dots are not always well detected when writing on the screen, and care must be taken not to rest the hand on the screen, which can affect some people's writing. However, on smaller interfaces such as smartphones, the hand can rest on the table just below the smartphone without needing to rest on the screen. The size of the canvas used for testing also presents a limitation due to the varying screen sizes of different devices. For example, the relative dimensions of a drawing will appear larger on a smaller screen, or smaller on a larger screen, like a tablet. This can impact tests involving freehand writing or drawing, where there are no specific strokes or pre-established shapes to follow. It's also possible that this could affect other variables, such as speed. There is no exact way to ensure that the canvas size is consistent across all screens, as it can vary for numerous reasons. However, it is expected that this effect will be mitigated by users' tendency to try and fill the entire screen with their drawing.

Lastly, it's important to consider that the elderly population may face difficulties writing on screens due to the existing digital divide. However, as time progresses, it is expected that the target population of the application will become more accustomed to these types of technologies, thus facilitating the performance of the test.

Chapter 6

Economic Evaluation

Regarding the economic evaluation of the project, it's important to note that the application currently operates through ngrok, a tunneling service, not a hosting service. This means that a personal computer is being used to constantly run the application. Additionally, a personal plan costing $10 per month has been required. However, this is not the ideal situation for expanding the application's usage, hence economic evaluations for future versions are necessary. The next version of the application will have a relatively low annual maintenance cost compared to other AD detection techniques. To estimate this, we have consulted Hostinger's website [111] for approximate prices to develop and maintain a website.

Firstly, within the project's fixed costs are the expenses of developing the web application. For this, a professional should be hired to develop the web application from scratch, even though it's based on this project. The expenses of hiring a senior web developer for this task would be around €5000. Additionally, additional improvements should be included such as the initial data collection exclusive to the application of healthy and sick patients, to generate new models, which would need to be economically compensated.

On the other hand, there are the annual costs of maintaining the application. Depending on the web hosting services contracted and the provider, costs can vary considerably. It is estimated that for a VPS Hosting service, costs range between €100-250 annually, between €120-350 annually for Cloud Hosting services, or even reaching €3000-3500 annually for dedicated hosting. Everything depends on the web data traffic, performance, and the number of users who will use the application.

Also, there are the maintenance costs of the application in terms of updates to improve the user experience, design, performance, security, and compatibility. These can vary depending on the purpose and the degree of optimization that is desired to give to the application over time. It is estimated that hiring a designer and a web developer can jointly vary between €1000-2000 per year. In addition, it's important to highlight the acquisition of a domain name, which will be around €50-100 per year. Given that the application contains sensitive personal health data, it would be important to contract an SSL certificate to ensure secure and encrypted connections, which could cost between €200-1000 annually.

One aspect to consider is technical support, in case help is needed or problems arise from customers, which could cost around €150 annually. Another aspect to consider is the purchase of passive capacitive pencils, recommended for the test, which can be found from less than €1 per unit and offered to users who contract the service.

Regarding how the funds to maintain this infrastructure annually would be obtained, it is proposed that this application be subsidized by state public health organizations. It wouldn't make sense to profit from a diagnostic method as serious as AD. In addition, allowing this application to be used without restrictions by health professionals in the public sector facilitates its use. This, in turn, can lead to a greater collection of data and, consequently, to obtaining more accurate predictive models in the future, compared to if it were done through the use of licenses or similar.

Furthermore, it's important to highlight that maintaining an application for an organization as large as public health in some countries represents an insignificant percentage of expenditure compared to other diagnostic techniques. This reinforces the feasibility of the proposal to subsidize the application through public health.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

Throughout this project, the creation of an additional tool for the preventive diagnosis of AD based on machine learning has been addressed. Once again, the enormous potential of these approaches for the diagnosis of diseases, which can be applied to any field of medicine, has been demonstrated. As long as there is a large amount of data for a problem, predictive models seem to offer a generally effective solution. Experience has been gained working with several machine learning classification models in Python, constantly seeking the best models, increasing their robustness and reducing the overfitting of data through various techniques and methods. Emphasis has also been placed on web development to give utility to these predictive models and generate a simple but effective web application in various aspects, such as the prediction of the diagnosis, the monitoring of a patient or the ability to collect data for future improvements of the models. As an added value, it has been verified that, when comparing the results with those of the original study on which this project was based, the results are very similar in some cases, despite the fact that this project is simpler than the original.

In addition, accessibility has been gained, as touch devices are within everyone's reach. This result has exceeded initial expectations, as it was thought that by significantly reducing the number of predictor variables and some tasks, the performance of the models would be compromised. However, as has been seen, the impact has not been serious. As for the objectives set from the beginning, although some details have been slightly modified as the project progressed, they have always remained consistent. It can be concluded that all the proposed objectives have been met, from the general objective of obtaining the web predictor application for mobile devices to the four specific objectives, which included evaluating several classification models and obtaining the best model, which had an accuracy higher than 75% (it has a theoretical 88%), that the developed application captured the data in a way similar to those of the original database, not only have they resembled but the option to collect new data has been included, and finally being able to compare the results with those of the original, which have proven to be similar.

7.2 Future Lines

The main line of future work would be to generate exclusive databases of the application for various languages, leaving aside the DARWIN database, which would allow improving the prediction of the diagnosis by generating more robust models. For this, it would be necessary to carry out many tests on sick and healthy patients, ensuring that the tests are carried out correctly. For obvious reasons, doing this during the development of the MTP would have been impossible to manage due to logistics and lack of time. Another pending aspect would be to carry out a more exhaustive study of the models, both for the models trained with all the predictor variables and for those specific to the task. As in the original study, it could be analyzed which tasks of the test have more meaning for the final prediction and if models can be made with these. Something similar could also be done with respect to the number of classifiers that can participate in each of the prediction votes, since in this project only three or five main classifiers have been tested for all types of classifiers, and it would have to be tested which number of classifiers is more optimal for each classifier. It would also be optimal to work on a second version of the application, this time developed by a professional in web development who worked on optimizing the app, for example, translating the application to other libraries or programming languages, since Flask is rather a framework for simple applications. In addition, the implementation of the collection of other predictor variables that can be obtained on smartphones, such as acceleration or well-implemented jerk, or the design of other additional tasks could be studied.

7.3 Planning Follow-up

In relation to the follow-up of the initially proposed planning, it has been followed in general as it was supposed at first, with the exception of some delays and additional tasks not foreseen. In the first phase of work development (PEC2), for example, the design of the data collection interface with Tkinter was delayed a few days, which caused a general delay in some later tasks. However, time was recovered with these last ones. However, in the second phase of development (PEC3) serious problems began to arise. It was observed that the models made up to that point were only the single-classifier and the task-specific classifier models were still to be made. This phase of development of the new models, where the Top N Classifiers were also included, had to be carried out simultaneously with the analysis of all the models. In addition, the models had to be trained three times in total due to unforeseen colinearity, and variables and tasks of the test that could not be replicated with the interface. All this delayed the two mentioned tasks from April 28, which was thought, until May 11. A few tasks later came the development of the web application, which also took a few days. However, the number of problems that the deployment of the application entailed was not counted. It was tested on several hosting servers, but none of them worked properly. Because several days were lost and the delivery dates were approaching, it was decided to do tunneling of the localhost through ngrok, which although it is not ideal for a web page, provides a link from which you can access the app via internet as long as the application is running on the computer. This is why this phase has been extended to the last phase of the project (PEC4), further adjusting the time of these. In summary, the planning initially carried out was good, but perhaps too optimistic in terms of

the dates that had to be followed, as they did not leave much margin for problems that could arise during the process.

On the other hand, it is also worth mentioning the impact of the project in relation to the SDGs that were expected to be affected, which have not increased, there were two: In SDG 3 - Good Health and Well-Being, with the aim of minimizing the negative impact that a wrong prediction can generate, it was tried to inform in the tests for unregistered users at all times that the test carried out is only informative and the predictions can be wrong. In addition, at the end of the test, they are recommended to contact health professionals, which together with a positive prediction for AD can help to take the next step contributing to an early diagnosis. On the other hand, for registered users, who will have to have been registered by a health professional, it is understood that these messages are not necessary and it will be the health professional who takes care of the advice, treatment and follow-up of each of the cases. Finally, regarding the impact of SDG 10 – Reduced Inequalities, it is expected that the developed application or future versions of this can serve as a real support tool in the diagnosis of AD, since with a smartphone, internet connection and a stylus, this test can be performed without problems. This service can undoubtedly help to reduce inequalities between economically disadvantaged countries that cannot afford certain more effective and complex tests.

Chapter 8

Glossary

- AB: Adaptive Boosting
- ACh: Acetylcholine
- AChE: Acetylcholinesterase
- AD: Alzheimer's Disease
- ADRDA: Alzheimer's Disease and Related Disorders Association
- AI: Artificial Intelligence
- APP: Amyloid Precursor Protein
- Aβ: Beta-Amyloid
- CDT: Clock Drawing Test
- ChAT: Choline Acetyltransferase
- CSF: Cerebrospinal Fluid
- CSS: Cascading Style Sheets
- CT: Computed Tomography
- CSV: Comma-Separated Values
- DARWIN: Diagnosis AlzheimeR WIth haNdwriting
- DMTs: Disease Modifying Therapies
- DT: Decision Tree
- ET: Extra Trees
- FDG: Fluorodeoxyglucose
- GB: Gradient Boosting

- GMRT: Generalization of the Mean Relative Tremor
- GNB: Gaussian Naive Bayes
- HC: Healthy Controls
- HIV: Human Immunodeficiency Virus
- HTML: Hyper Text Markup Language
- HTTPS: Hypertext Transfer Protocol Secure
- IAD: Inherited Alzheimer's Disease
- IDE: Integrated Development Environment
- KNN: K-Nearest Neighbors
- LDA: Linear Discriminant Analysis
- LR: Logistic Regression
- MCI: Mild Cognitive Impairment
- ML: Machine Learning
- MLP: Multilayer Perceptron
- MMH: Maximum Margin Hyperplane
- MMSE: Mini Mental State Examination
- MRT: Mean Relative Tremor
- MTP: Master Thesis Project
- NB: Naive Bayes
- NDs: Neurodegenerative Diseases
- NFTs: Neurofibrillary Tangles
- NINCDS: National Institute of Neurological and Communicative Disorders and Stroke
- PCA: Principal Component Analysis
- PD: Parkinson's Disease
- PEC: From Spanish (Prueba de Evaluación Continua), translated as Continuous Evaluation Test
- PET: Positron Emission Tomography

- PHF: Paired Helical Filaments
- QR: Quick Response
- RAM: Random Access Memory
- RF: Random Forest
- SDGs: Sustainable Development Goals
- SSL: Secure Sockets Layer
- SVM: Support Vector Machine
- VIF: Variance Inflation Factor
- VPS: Virtual Private Server
- XGB: Extreme Gradient Boosting

Bibliography

[1] Teamgantt. https://www.teamgantt.com/, 2024. [Accessed: 2024-06-18].

[2] Zeinab Breijyeh and Rafik Karaman. Comprehensive review on alzheimer's disease: causes and treatment. *Molecules*, 25(24):5789, 2020.

[3] Claudio De Stefano, Francesco Fontanella, Donato Impedovo, Giuseppe Pirlo, and Alessandra Scotto di Freca. Handwriting analysis to support neurodegenerative diseases diagnosis: A review. *Pattern Recognition Letters*, 121:37–45, 2019.

[4] Nicole Dalia Cilia, Claudio De Stefano, Francesco Fontanella, Mario Molinara, and Alessandra Scotto Di Freca. Handwriting analysis to support alzheimer's disease diagnosis: a preliminary study. In *Computer Analysis of Images and Patterns: 18th International Conference, CAIP 2019, Salerno, Italy, September 3–5, 2019, Proceedings, Part II 18*, pages 143–151. Springer, 2019.

[5] Nicole D Cilia, Giuseppe De Gregorio, Claudio De Stefano, Francesco Fontanella, Angelo Marcelli, and Antonio Parziale. Diagnosing alzheimer's disease from on-line handwriting: A novel dataset and performance benchmarking. *Engineering Applications of Artificial Intelligence*, 111:104822, 2022.

[6] Liana G Apostolova. Alzheimer disease. *Continuum: Lifelong Learning in Neurology*, 22(2):419–434, 2016.

[7] Wiesje M van der Flier, Marjolein E de Vugt, Ellen MA Smets, Marco Blom, and Charlotte E Teunissen. Towards a future where alzheimer's disease pathology is stopped before the onset of dementia. *Nature aging*, 3(5):494–505, 2023.

[8] World Health Organization et al. Global status report on the public health response to dementia. 2021.

[9] LW Chu. Alzheimer's disease: early diagnosis and treatment. *Hong Kong Medical Journal*, 18(3):228, 2012.

[10] Nicole Dalia Cilia, Claudio De Stefano, Francesco Fontanella, and Alessandra Scotto Di Freca. An experimental protocol to support cognitive impairment diagnosis by using handwriting analysis. *Procedia Computer Science*, 141:466–471, 2018.

[11] Mei Hui Tseng and Sharon A Cermak. The influence of ergonomic factors and perceptual–motor abilities on handwriting performance. *The American journal of occupational Therapy*, 47(10):919–926, 1993.

[12] Jany Lambert, Bénédicte Giffard, Florence Nore, Vincent de La Sayette, Florence Pasquier, and Francis Eustache. Central and peripheral agraphia in alzheimer's disease: from the case of auguste d. to a cognitive neuropsychology approach. *Cortex*, 43(7):935–951, 2007.

[13] Rosa López Mongil. Deterioro cognitivo/demencia: valoración y seguimiento. aspectos funcionales. *Revista Española de Geriatría y Gerontología*, 52:28–33, 2017.

[14] Monika A Myszczynska, Poojitha N Ojamies, Alix MB Lacoste, Daniel Neil, Amir Saffari, Richard Mead, Guillaume M Hautbergue, Joanna D Holbrook, and Laura Ferraiuolo. Applications of machine learning to diagnosis and treatment of neurodegenerative diseases. *Nature Reviews Neurology*, 16(8):440–456, 2020.

[15] Donato Impedovo, Giuseppe Pirlo, and Gennaro Vessio. Dynamic handwriting analysis for supporting earlier parkinson's disease diagnosis. *Information*, 9(10):247, 2018.

[16] Andrew Frank. Uci machine learning repository. http://archive.ics.uci.edu/ml, 2010. [Accessed: 2024-02-28].

[17] Fredrik Lundh. An introduction to tkinter. http://www.pythonware.com/library/tkinter/introduction/index.htm, 1999.

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

[19] B. Lantz. *Machine Learning with R: Expert techniques for predictive modeling.* Packt Publishing, 2019.

[20] Martin Prince, Anders Wimo, Maëlenn Guerchet, Gemma-Claire Ali, Yu-Tzu Wu, and Matthew Prina. *World Alzheimer Report 2015. The Global Impact of Dementia: An analysis of prevalence, incidence, cost and trends.* PhD thesis, Alzheimer's Disease International, 2015.

[21] Hanns Hippius and Gabriele Neundörfer. The discovery of alzheimer's disease. *Dialogues in clinical neuroscience*, 5(1):101–108, 2003.

[22] Alberto Serrano-Pozo, Matthew P Frosch, Eliezer Masliah, and Bradley T Hyman. Neuropathological alterations in alzheimer disease. *Cold Spring Harbor perspectives in medicine*, 1(1):a006189, 2011.

[23] Richard A Armstrong. The molecular biology of senile plaques and neurofibrillary tangles in alzheimer's disease. *Folia Neuropathologica*, 47(4):289–299, 2009.

[24] J-P Brion. Neurofibrillary tangles and alzheimer's disease. *European neurology*, 40(3):130–140, 1998.

[25] Athanasios Metaxas and Stefan J Kempf. Neurofibrillary tangles in alzheimer's disease: elucidation of the molecular mechanism by immunohistochemistry and tau protein phospho-proteomics. *Neural regeneration research*, 11(10):1579–1581, 2016.

[26] Cassia R Overk and Eliezer Masliah. Pathogenesis of synaptic degeneration in alzheimer's disease and lewy body disease. *Biochemical pharmacology*, 88(4):508–516, 2014.

[27] T Babic. The cholinergic hypothesis of alzheimer's disease: a review of progress. *Journal of Neurology, Neurosurgery & Psychiatry*, 67(4):558–558, 1999.

[28] Talita H Ferreira-Vieira, Isabella M Guimaraes, Flavia R Silva, and Fabiola M Ribeiro. Alzheimer's disease: targeting the cholinergic system. *Current neuropharmacology*, 14(1):101–115, 2016.

[29] Giulia Paroni, Paola Bisceglia, and Davide Seripa. Understanding the amyloid hypothesis in alzheimer's disease. *Journal of Alzheimer's Disease*, 68(2):493–510, 2019.

[30] Fuyuki Kametani and Masato Hasegawa. Reconsideration of amyloid hypothesis and tau hypothesis in alzheimer's disease. *Frontiers in neuroscience*, 12:328460, 2018.

[31] Roberta Ricciarelli and Ernesto Fedele. The amyloid cascade hypothesis in alzheimer's disease: it's time to change our mind. *Current neuropharmacology*, 15(6):926–935, 2017.

[32] Rita Guerreiro and Jose Bras. The age factor in alzheimer's disease. *Genome medicine*, 7:1–3, 2015.

[33] Konstantina G Yiannopoulou and Sokratis G Papageorgiou. Current and future treatments in alzheimer disease: an update. *Journal of central nervous system disease*, 12:1179573520907397, 2020.

[34] Gill Livingston, Jonathan Huntley, Andrew Sommerlad, David Ames, Clive Ballard, Sube Banerjee, Carol Brayne, Alistair Burns, Jiska Cohen-Mansfield, Claudia Cooper, et al. Dementia prevention, intervention, and care: 2020 report of the lancet commission. *The Lancet*, 396(10248):413–446, 2020.

[35] Preet Anand and Baldev Singh. A review on cholinesterase inhibitors for alzheimer's disease. *Archives of pharmacal research*, 36:375–399, 2013.

[36] Jinping Liu, Lirong Chang, Yizhi Song, Hui Li, and Yan Wu. The role of nmda receptors in alzheimer's disease. *Frontiers in neuroscience*, 13:43, 2019.

[37] Yu-Jhen Huang, Chieh-Hsin Lin, Hsien-Yuan Lane, and Guochuan E Tsai. Nmda neurotransmission dysfunction in behavioral and psychological symptoms of alzheimer's disease. *Current neuropharmacology*, 10(3):272–285, 2012.

[38] Júlia Companys-Alemany, Andreea L Turcu, Aina Bellver-Sanchis, Maria I Loza, José M Brea, Anna M Canudas, Rosana Leiva, Santiago Vázquez, Mercè Pallàs, and Christian Griñán-Ferré. A novel nmda receptor antagonist protects against cognitive decline presented by senescent mice. *Pharmaceutics*, 12(3):284, 2020.

[39] Jeffrey Cummings and Nick Fox. Defining disease modifying therapy for alzheimer's disease. *The journal of prevention of Alzheimer's disease*, 4(2):109, 2017.

[40] Jeffrey Cummings, Garam Lee, Aaron Ritter, Marwan Sabbagh, and Kate Zhong. Alzheimer's disease drug development pipeline: 2020. *Alzheimer's & dementia: translational research & clinical interventions*, 6(1):e12050, 2020.

[41] Martin Tolar, Susan Abushakra, John A Hey, Anton Porsteinsson, and Marwan Sabbagh. Aducanumab, gantenerumab, ban2401, and alz-801—the first wave of amyloid-targeting drugs for alzheimer's disease with potential for near term approval. *Alzheimer's research & therapy*, 12(1):95, 2020.

[42] Stephanie Andrade, Maria João Ramalho, Joana Angélica Loureiro, and Maria do Carmo Pereira. Natural compounds for alzheimer's disease therapy: a systematic review of preclinical and clinical studies. *International journal of molecular sciences*, 20(9):2313, 2019.

[43] Yuan Ma, Man-wen Yang, Xin-wei Li, Jian-wei Yue, Jun-zong Chen, Mei-wen Yang, Xuan Huang, Lian-lian Zhu, Fen-fang Hong, and Shu-long Yang. Therapeutic effects of natural drugs on alzheimer's disease. *Frontiers in pharmacology*, 10:1355, 2019.

[44] Thomas G Beach, Sarah E Monsell, Leslie E Phillips, and Walter Kukull. Accuracy of the clinical diagnosis of alzheimer disease at national institute on aging alzheimer disease centers, 2005–2010. *Journal of neuropathology and experimental neurology*, 71(4):266–273, 2012.

[45] Orestes V Forlenza, Marcia Radanovic, Leda L Talib, Ivan Aprahamian, Breno S Diniz, Henrik Zetterberg, and Wagner F Gattaz. Cerebrospinal fluid biomarkers in alzheimer's disease: Diagnostic accuracy and prediction of dementia. *Alzheimer's & Dementia: Diagnosis, Assessment & Disease Monitoring*, 1(4):455–463, 2015.

[46] Femke H Bouwman, Giovanni B Frisoni, Sterling C Johnson, Xiaochun Chen, Sebastiaan Engelborghs, Takeshi Ikeuchi, Claire Paquet, Craig Ritchie, Sasha Bozeat, Frances-Catherine Quevenco, et al. Clinical application of csf biomarkers for alzheimer's disease: From rationale to ratios. *Alzheimer's & Dementia: Diagnosis, Assessment & Disease Monitoring*, 14(1):e12314, 2022.

[47] Nicolas R Barthélemy, Gemma Salvadó, Suzanne E Schindler, Yingxin He, Shorena Janelidze, Lyduine E Collij, Benjamin Saef, Rachel L Henson, Charles D Chen, Brian A Gordon, et al. Highly accurate blood test for alzheimer's disease is similar or superior to clinical cerebrospinal fluid tests. *Nature medicine*, 30(4):1085–1095, 2024.

[48] Armand S Schachter and Kenneth L Davis. Alzheimer's disease. *Dialogues in clinical neuroscience*, 2(2):91–100, 2000.

[49] Guy McKhann, David Drachman, Marshall Folstein, Robert Katzman, Donald Price, and Emanuel M Stadlan. Clinical diagnosis of alzheimer's disease: Report of the nincds-adrda work group* under the auspices of department of health and human services task force on alzheimer's disease. *Neurology*, 34(7):939–939, 1984.

[50] Judith Neugroschl and Sophia Wang. Alzheimer's disease: diagnosis and treatment across the spectrum of disease severity. *Mount Sinai Journal of Medicine: A Journal of Translational and Personalized Medicine*, 78(4):596–612, 2011.

[51] Guy M McKhann, David S Knopman, Howard Chertkow, Bradley T Hyman, Clifford R Jack Jr, Claudia H Kawas, William E Klunk, Walter J Koroshetz, Jennifer J Manly, Richard Mayeux, et al. The diagnosis of dementia due to alzheimer's disease: Recommendations from the national institute on aging-alzheimer's association workgroups on diagnostic guidelines for alzheimer's disease. *Alzheimer's & dementia*, 7(3):263–269, 2011.

[52] Richard Mayeux and Yaakov Stern. Epidemiology of alzheimer disease. *Cold Spring Harbor perspectives in medicine*, 2(8):a006239, 2012.

[53] Roy Yaari, Adam S Fleisher, and Pierre N Tariot. Updates to diagnostic guidelines for alzheimer's disease. *The primary care companion for CNS disorders*, 13(5):26971, 2011.

[54] Tom Mitchell. Introduction to machine learning. *Machine learning*, 7:2–5, 1997.

[55] Tammy Jiang, Jaimie L Gradus, and Anthony J Rosellini. Supervised machine learning: a brief primer. *Behavior therapy*, 51(5):675–687, 2020.

[56] Isaac R Galatzer-Levy, Kelly V Ruggles, and Zhe Chen. Data science in the research domain criteria era: relevance of machine learning to the study of stress pathology, recovery, and resilience. *Chronic Stress*, 2:2470547017747553, 2018.

[57] Adrian BR Shatte, Delyse M Hutchinson, and Samantha J Teague. Machine learning in mental health: a scoping review of methods and applications. *Psychological medicine*, 49(9):1426–1448, 2019.

[58] Rajib Saha, Anirban Mukherjee, Aniruddha Sadhukhan, Anisha Roy, and Manashi De. Handwriting analysis for early detection of a lzheimer's disease. *Intelligent data analysis: From data gathering to data comprehension*, pages 369–385, 2020.

[59] Jonas J de Paula, Maicon R Albuquerque, Guilherme M Lage, Maria A Bicalho, Marco A Romano-Silva, and Leandro F Malloy-Diniz. Impairment of fine motor dexterity in mild cognitive impairment and alzheimer's disease dementia: association with activities of daily living. *Brazilian Journal of Psychiatry*, 38:235–238, 2016.

[60] Jean Neils-Strunjas, Kathy Groves-Wright, Pauline Mashima, and Stacy Harnish. Dysgraphia in alzheimer's disease: a review for clinical and research purposes. 49:1313–1330, 2006.

[61] Esther J Smits, Antti J Tolonen, Luc Cluitmans, Mark Van Gils, Bernard A Conway, Rutger C Zietsma, Klaus L Leenders, and Natasha M Maurits. Standardized handwriting to assess bradykinesia, micrographia and tremor in parkinson's disease. *PloS one*, 9(5):e97614, 2014.

[62] Garipelli Gangadhar, Denny Joseph, and V Srinivasa Chakravarthy. An oscillatory neuromotor model of handwriting generation. *International journal of document analysis and recognition (ijdar)*, 10:69–84, 2007.

[63] Nicole Dalia Cilia. Understand me or duplicate me? levels of explanation in artificial methodology. *e Machine-Organism Analogy*, page 61, 2017.

[64] Réjean Plamondon. A kinematic theory of rapid human movements: Part i. movement representation and generation. *Biological cybernetics*, 72:295–307, 1995.

[65] Réjean Plamondon. A kinematic theory of rapid human movements. part ii. movement time and control. *Biological cybernetics*, 72:309–20, 02 1995.

[66] H. Bezine, A.M. Alimi, and N. Sherkat. Generation and analysis of handwriting script with the beta-elliptic model. In *Ninth International Workshop on Frontiers in Handwriting Recognition*, pages 515–520. IEEE, 2004.

[67] John M Hollerbach. An oscillation theory of handwriting. *Biological cybernetics*, 39(2):139–156, 1981.

[68] Daniel Bullock, Stephen Grossberg, and Christian Mannes. A neural network model for cursive script production. *Biological Cybernetics*, 70(1):15–28, 1993.

[69] Rosa Senatore and Angelo Marcelli. A neural scheme for procedural motor learning of handwriting. In *2012 international conference on frontiers in handwriting recognition*, pages 659–664. IEEE, 2012.

[70] Tiziana D'Alessandro, Claudio De Stefano, Francesco Fontanella, Emanuele Nardone, and Alessandra Scotto di Freca. Feature evaluation in handwriting analysis for alzheimer's disease using bayesian network. In *International Graphonomics Conference*, pages 122–135. Springer, 2023.

[71] Tiziana D'Alessandro, Cristina Carmona-Duarte, Claudio De Stefano, Moises Diaz, Miguel Angel Ferrer, and Francesco Fontanella. A machine learning approach to analyze the effects of alzheimer's disease on handwriting through lognormal features. In *International Graphonomics Conference*, pages 103–121. Springer, 2023.

[72] Nicole Dalia Cilia, Tiziana D'Alessandro, Claudio De Stefano, and Francesco Fontanella. Offline handwriting image analysis to predict alzheimer's disease via deep learning. In *2022 26th International Conference on Pattern Recognition (ICPR)*, pages 2807–2813. IEEE, 2022.

[73] Iqra Kamran, Saeeda Naz, Imran Razzak, and Muhammad Imran. Handwriting dynamics assessment using deep neural network for early identification of parkinson's disease. *Future Generation Computer Systems*, 117:234–244, 2021.

[74] Nihar M Ranjan, Gitanjali Mate, and Maya Bembde. Detection of parkinson's disease using machine learning algorithms and handwriting analysis. *Journal of Data Mining and Management (e-ISSN: 2456–9437)*, 8(1):21–29, 2023.

[75] Guido Van Rossum and Fred L. Drake. *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA, 2009.

[76] Pycharm 2023.3 documentation. https://www.jetbrains.com/help/pycharm/2023.3/getting-started.html, 2023. [Accessed 2024-05-14].

[77] Donato Impedovo, Giuseppe Pirlo, Donato Barbuzzi, Alessandro Balestrucci, Sebastiano Impedovo, et al. Handwritten processing for pre diagnosis of alzheimer disease. In *BIODEVICES 2014*, pages 193–199. 2014.

[78] Jin H Yan, Susan Rountree, Paul Massman, Rachelle Smith Doody, and Hong Li. Alzheimer's disease and mild cognitive impairment deteriorate fine movement control. *Journal of Psychiatric Research*, 42(14):1203–1212, 2008.

[79] Andreas Schröter, Roland Mergl, Katharina Bürger, Harald Hampel, H-J Möller, and Ulrich Hegerl. Kinematic analysis of handwriting movements in patients with alzheimer's disease, mild cognitive impairment, depression and healthy subjects. *Dementia and geriatric cognitive disorders*, 15(3):132–142, 2003.

[80] Perla Werner, Sara Rosenblum, Gady Bar-On, Jeremia Heinik, and Amos Korczyn. Handwriting process variables discriminating mild alzheimer's disease and mild cognitive impairment. *The Journals of Gerontology Series B: Psychological Sciences and Social Sciences*, 61(4):P228–P236, 2006.

[81] Emanuela Onofri, Marco Mercuri, M Salesi, M Ricciardi, and T Archer. Dysgraphia in relation to cognitive performance in patients with alzheimer's disease. *Journal of Intellectual Disability-Diagnosis and Treatment*, 1(2):113–124, 2013.

[82] Ziad S Nasreddine, Natalie A Phillips, Valérie Bédirian, Simon Charbonneau, Victor Whitehead, Isabelle Collin, Jeffrey L Cummings, and Howard Chertkow. The montreal cognitive assessment, moca: a brief screening tool for mild cognitive impairment. *Journal of the American Geriatrics Society*, 53(4):695–699, 2005.

[83] Mary Ganguli, Graham Ratcliff, F Jacob Huff, Steven Belle, Mary Jean Kancel, Lynn Fischer, and Lewis H Kuller. Serial sevens versus world backwards: a comparison of the two measures of attention from the mmse. *Journal of geriatric psychiatry and neurology*, 3(4):203–207, 1990.

[84] M Renier, F Gnoato, A Tessari, M Formilan, F Busonera, P Albanese, G Sartori, and A Cester. A correlational study between signature, writing abilities and decision-making capacity among people with initial cognitive impairment. *Aging clinical and experimental research*, 28:505–511, 2016.

[85] Angelo Marcelli, Antonio Parziale, and Adolfo Santoro. Modelling visual appearance of handwriting. pages 673–682, 2013.

[86] Angelo Marcelli, Antonio Parziale, Rosa Senatore, et al. Some observations on handwriting from a motor learning perspective. 1022:6–10, 2013.

[87] Martin Vyhnálek, Eva Rubínová, Hana Marková, Tomáš Nikolai, Jan Laczó, Ross Andel, and Jakub Hort. Clock drawing test in screening for alzheimer's dementia and mild

cognitive impairment in clinical practice. *International journal of geriatric psychiatry*, 32(9):933–939, 2017.

[88] Clayton R Pereira, Danillo R Pereira, Francisco A Da Silva, Christian Hook, Silke AT Weber, Luis AM Pereira, and Joao P Papa. A step towards the automated diagnosis of parkinson's disease: Analyzing handwriting movements. In *2015 IEEE 28th international symposium on computer-based medical systems*, pages 171–176. IEEE, 2015.

[89] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.

[90] GT Kneebone. Probability and the weighing of evidence. by ij good.(london: Charles griffin and company. 1950. pp. viii+ 119. price 16s.). *Philosophy*, 26(97):163–164, 1951.

[91] G Sebestyen. Review of'learning machines'(nilsson, nils j.; 1965). *IEEE Transactions on Information Theory*, 12(3):407–407, 1966.

[92] Chih-Chung Chang and Chih-Jen Lin. Libsvm: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):1–27, 2011.

[93] Hsiang-Fu Yu, Fang-Lan Huang, and Chih-Jen Lin. Dual coordinate descent methods for logistic regression and maximum entropy models. *Machine Learning*, 85:41–75, 2011.

[94] Ernest Yeboah Boateng, Daniel A Abaye, et al. A review of the logistic regression model with emphasis on medical research. *Journal of data analysis and information processing*, 7(04):190, 2019.

[95] Richard O Duda, Peter E Hart, et al. *Pattern classification*. John Wiley & Sons, 2006.

[96] Petros Xanthopoulos, Panos M Pardalos, Theodore B Trafalis, Petros Xanthopoulos, Panos M Pardalos, and Theodore B Trafalis. Linear discriminant analysis. *Robust data mining*, pages 27–33, 2013.

[97] C Radhakrishna Rao. The utilization of multiple measurements in problems of biological classification. *Journal of the Royal Statistical Society. Series B (Methodological)*, 10(2):159–203, 1948.

[98] Leo Breiman. *Classification and regression trees*. Routledge, 2017.

[99] J. Ross Quinlan. Induction of decision trees. *Machine learning*, 1:81–106, 1986.

[100] Leo Breiman. Random forests. *Machine learning*, 45:5–32, 2001.

[101] Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. *Machine learning*, 63:3–42, 2006.

[102] Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009.

[103] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[104] Christopher M Bishop. Pattern recognition and machine learning. *Springer google schola*, 2:645–678, 2006.

[105] Robert E Schapire and Yoav Freund. Boosting: Foundations and algorithms. *Kybernetes*, 42(1):164–166, 2013.

[106] Yoav Freund, Robert E Schapire, et al. Experiments with a new boosting algorithm. In *icml*, volume 96, pages 148–156. Citeseer, 1996.

[107] Jerome H Friedman. Stochastic gradient boosting. *Computational statistics & data analysis*, 38(4):367–378, 2002.

[108] Tianqi Chen, Tong He, Michael Benesty, Vadim Khotilovich, Yuan Tang, Hyunsu Cho, Kailong Chen, Rory Mitchell, Ignacio Cano, Tianyi Zhou, et al. Xgboost: extreme gradient boosting. *R package version 0.4-2*, 1(4):1–4, 2015.

[109] Joblib Development Team. Joblib: running python functions as pipeline jobs. https://joblib.readthedocs.io/en/latest/. [Accessed: 2024-04-23].

[110] Miguel Grinberg. *Flask web development: developing web applications with python*. " O'Reilly Media, Inc.", 2018.

[111] ¿cuál es el precio de crear una página web en 2024? https://www.hostinger.es/tutoriales/precio-pagina-web, 2024. [Accessed 2024-06-07].

Appendix A

Figures and Tables

A.1 Task Design

Table A.1: List of tasks performed in the original study. Task categories are: memory and dictation (M), Graphic (G), and Copy (C) [5].

#	Description	Category
1	Signature drawing	M
2	Join two points with a horizontal line, continuously for four times	G
3	Join two points with a vertical line, continuously for four times	G
4	Retrace a circle (6 cm of diameter) continuously for four times	G
5	Retrace a circle (3 cm of diameter) continuously for four times	G
6	Copy the letters 'l', 'm' and 'p'	C
7	Copy the letters on the adjacent rows	C
8	Write cursively a sequence of four lowercase letter 'l', in a single smooth movement	C
9	Write cursively a sequence of four lowercase cursive bigram 'le', in a single smooth movement	C
10	Copy the word "foglio"	C
11	Copy the word "foglio" above a line	C
12	Copy the word "mamma"	C
13	Copy the word "mamma" above a line	C
14	Memorize the words "telefono", "cane", and "negozio" and rewrite them	M
15	Copy in reverse the word "bottiglia"	C
16	Copy in reverse the word "casa"	C
17	Copy six words (regular, non regular, non words) in the appropriate boxes	C
18	Write the name of the object shown in a picture (a chair)	M
19	Copy the fields of a postal order	C
20	Write a simple sentence under dictation	M
21	Retrace a complex form	G
22	Copy a telephone number	C
23	Write a telephone number under dictation	M
24	Draw a clock, with all hours and put hands at 11:05 (Clock Drawing Test)	G
25	Copy a paragraph	C

1. Haz tu firma.

2. Une los puntos con una línea horizontal, yendo y viniendo cuatro veces.

3. Une los puntos con una línea vertical, subiendo y bajando cuatro veces.

4. Traza el círculo de forma continua cuatro veces.

5. Traza el círculo de forma continua cuatro veces.

6. Copia en cursiva las letras *"l"*, *"m"*, *"p"*.

7. Escribe en cursiva cuatro letras "ℓ" (ele) seguidas, una tras otra sin levantar el bolígrafo de la pantalla.

8. Escribe "le" en cursiva cuatro veces seguidas, una tras otra sin levantar el bolígrafo de la pantalla.

9. Escribe en cursiva la palabra "folios".

10. Escribe en cursiva la palabra "folios" sobre la línea.

11. Escribe en cursiva la palabra "mamá".

12. Escribe en cursiva la palabra "mamá" sobre la línea.

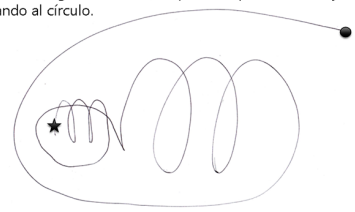
13. Escribe en cursiva las tres palabras que acabas de memorizar.

14. Copia la palabra "botiquín" empezando por la última letra, es decir, al revés.

15. Copia la palabra "casa" empezando por la última letra, es decir, al revés.

16. Escribe el nombre del objeto que se muestra en la figura.

17. Traza la siguiente forma, empezando por la estrella y llegando al círculo.



18. Copia el siguiente número telefónico:

0817425963

19. Dibuja un reloj analógico circular con todas las horas y coloca las manecillas marcando las 11 y 5 minutos.

Figure A.1: Set of background images with the statements that will appear in each of the test tasks.

A.2 Models Performance Metrics

Table A.2: Comparison of Accuracy of Single Classifier Models by Nemenyi's Post-hoc Test

	RF	LR	KNN	LDA	GNB	SVM	DT	MLP	ET	AB	GB	XGB
RF	1.000	0.135	0.001	0.047	0.900	0.206	0.001	0.095	0.900	0.262	0.533	0.159
LR	0.135	1.000	0.001	0.900	0.740	0.900	0.002	0.900	0.547	0.900	0.900	0.900
KNN	0.001	0.001	1.000	0.002	0.001	0.001	0.900	0.001	0.001	0.001	0.001	0.001
LDA	0.047	0.900	0.002	1.000	0.506	0.900	0.008	0.900	0.300	0.900	0.900	0.900
GNB	0.900	0.740	0.001	0.506	1.000	0.850	0.001	0.657	0.900	0.900	0.900	0.781
SVM	0.206	0.900	0.001	0.900	0.850	1.000	0.001	0.900	0.657	0.900	0.900	0.900
DT	0.001	0.002	0.900	0.008	0.001	0.001	1.000	0.003	0.001	0.001	0.001	0.001
MLP	0.095	0.900	0.001	0.900	0.657	0.900	0.003	1.000	0.462	0.900	0.900	0.900
ET	0.900	0.547	0.001	0.300	0.900	0.657	0.001	0.462	1.000	0.726	0.900	0.588
AB	0.262	0.900	0.001	0.900	0.900	0.900	0.001	0.900	0.726	1.000	0.900	0.900
GB	0.533	0.900	0.001	0.900	0.900	0.900	0.001	0.900	0.900	0.900	1.000	0.900
XGB	0.159	0.900	0.001	0.900	0.781	0.900	0.001	0.900	0.588	0.900	0.900	1.000

Table A.3: Comparison of Accuracy of Single Classifier Models by Nemenyi's Post-hoc (grouped by classifier).

	RF	LR	KNN	LDA	GNB	SVM	DT	MLP	ET	AB	GB	XGB
RF	1.000	0.001	0.001	0.001	0.001	0.001	0.001	0.002	0.900	0.001	0.096	0.001
LR	0.001	1.000	0.900	0.900	0.626	0.900	0.015	0.312	0.001	0.463	0.011	0.572
KNN	0.001	0.900	1.000	0.900	0.626	0.900	0.015	0.312	0.001	0.463	0.011	0.572
LDA	0.001	0.900	0.900	1.000	0.032	0.900	0.001	0.900	0.001	0.900	0.424	0.900
GNB	0.001	0.626	0.626	0.032	1.000	0.648	0.900	0.001	0.001	0.001	0.001	0.002
SVM	0.001	0.900	0.900	0.900	0.648	1.000	0.017	0.291	0.001	0.438	0.009	0.550
DT	0.001	0.015	0.015	0.001	0.900	0.017	1.000	0.001	0.001	0.001	0.001	0.001
MLP	0.002	0.312	0.312	0.900	0.001	0.291	0.001	1.000	0.011	0.900	0.900	0.900
ET	0.900	0.001	0.001	0.001	0.001	0.001	0.001	0.011	1.000	0.005	0.321	0.002
AB	0.001	0.463	0.463	0.900	0.001	0.438	0.001	0.900	0.005	1.000	0.900	0.900
GB	0.096	0.011	0.011	0.424	0.001	0.009	0.001	0.900	0.321	0.900	1.000	0.900
XGB	0.001	0.572	0.572	0.900	0.002	0.550	0.001	0.900	0.002	0.900	0.900	1.000

Table A.4: Comparison of Accuracy of Single Classifier Models by Nemenyi's Post-hoc Test (grouped by task).

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	1.000	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001
2	0.001	1.000	0.900	0.001	0.900	0.900	0.001	0.001	0.900	0.900	0.900	0.900	0.900	0.900	0.001	0.052	0.193	0.001	0.809
3	0.001	0.900	1.000	0.080	0.777	0.900	0.114	0.001	0.900	0.900	0.900	0.900	0.900	0.900	0.001	0.721	0.002	0.001	0.900
4	0.001	0.001	0.080	1.000	0.001	0.001	0.900	0.001	0.011	0.178	0.019	0.142	0.035	0.001	0.001	0.900	0.001	0.001	0.529
5	0.001	0.900	0.777	0.001	1.000	0.900	0.001	0.001	0.900	0.585	0.900	0.641	0.900	0.900	0.001	0.002	0.774	0.001	0.220
6	0.001	0.900	0.900	0.001	0.900	1.000	0.001	0.001	0.900	0.900	0.900	0.900	0.900	0.900	0.001	0.038	0.243	0.001	0.745
7	0.001	0.001	0.114	0.900	0.001	0.001	1.000	0.001	0.018	0.239	0.030	0.195	0.052	0.001	0.001	0.900	0.001	0.001	0.609
8	0.001	0.001	0.001	0.001	0.001	0.001	0.001	1.000	0.001	0.001	0.001	0.001	0.001	0.001	0.312	0.001	0.001	0.900	0.001
9	0.001	0.900	0.900	0.011	0.900	0.900	0.018	0.001	1.000	0.900	0.900	0.900	0.900	0.900	0.001	0.330	0.023	0.001	0.900
10	0.001	0.900	0.900	0.178	0.585	0.900	0.239	0.001	0.900	1.000	0.900	0.900	0.900	0.900	0.001	0.900	0.001	0.001	0.900
11	0.001	0.900	0.900	0.019	0.900	0.900	0.030	0.001	0.900	0.900	1.000	0.900	0.900	0.900	0.001	0.436	0.014	0.001	0.900
12	0.001	0.900	0.900	0.142	0.641	0.900	0.195	0.001	0.900	0.900	0.900	1.000	0.900	0.900	0.001	0.857	0.001	0.001	0.900
13	0.001	0.900	0.900	0.035	0.900	0.900	0.052	0.001	0.900	0.900	0.900	0.900	1.000	0.900	0.001	0.550	0.007	0.001	0.900
14	0.001	0.900	0.900	0.001	0.900	0.900	0.001	0.001	0.900	0.900	0.900	0.900	0.900	1.000	0.001	0.065	0.164	0.001	0.851
15	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.312	0.001	0.001	0.001	0.001	0.001	0.001	1.000	0.001	0.507	0.900	0.001
16	0.001	0.052	0.721	0.900	0.002	0.038	0.900	0.001	0.330	0.900	0.436	0.857	0.550	0.065	0.001	1.000	0.001	0.001	0.900
17	0.001	0.193	0.002	0.001	0.774	0.243	0.001	0.001	0.023	0.001	0.014	0.001	0.007	0.164	0.507	0.001	1.000	0.002	0.001
18	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.900	0.001	0.001	0.001	0.001	0.001	0.001	0.900	0.001	0.002	1.000	0.001
19	0.001	0.809	0.900	0.529	0.220	0.745	0.609	0.001	0.900	0.900	0.900	0.900	0.900	0.851	0.001	0.900	0.001	0.001	1.000

Table A.5: Mean accuracy (and standard deviation), specificity, and sensitivity (expressed in percentage) of the classifiers.

	RF	LR	KNN	LDA	GNB	SVM	DT	MLP	ET	AB	GB	XGB
Accuracy	87.61 (±4.28)	83.75 (±5.25)	67.16 (±5.68)	83.07 (±4.74)	86.93 (±4.81)	83.98 (±4.57)	71.48 (±5.77)	83.41 (±4.83)	87.05 (±4.44)	84.09 (±5.57)	84.89 (±5.15)	83.98 (±4.95)
Specificity	86.43	85.24	94.52	89.52	83.1	84.29	69.76	83.33	90.0	86.67	85.48	84.52
Sensitivity	88.7	82.39	42.17	77.17	90.43	83.7	73.04	83.48	84.35	81.74	84.35	83.48

Table A.6: Mean accuracy (expressed in percentage) achieved by the classifiers on each task.

Task #	RF	LR	KNN	LDA	GNB	SVM	DT	MLP	ET	AB	GB	XGB
1	63.41 (±5.83)	56.93 (±5.21)	61.93 (±6.38)	58.64 (±6.08)	56.93 (±5.45)	59.43 (±5.39)	59.55 (±6.76)	59.32 (±8.13)	63.75 (±7.79)	60.68 (±8.54)	64.66 (±5.11)	65.0 (±7.27)
2	69.77 (±5.38)	66.59 (±4.98)	66.59 (±5.24)	67.5 (±5.33)	66.48 (±5.28)	67.05 (±5.68)	64.66 (±4.68)	68.64 (±4.52)	66.82 (±4.68)	70.11 (±5.58)	67.16 (±8.05)	67.39 (±6.07)
3	65.91 (±5.93)	66.36 (±5.21)	66.25 (±5.85)	70.8 (±6.11)	66.48 (±4.06)	67.39 (±4.73)	61.93 (±8.35)	69.89 (±5.37)	65.0 (±6.24)	65.34 (±5.42)	66.14 (±7.04)	66.82 (±6.56)
4	67.16 (±4.46)	62.27 (±5.49)	63.3 (±5.29)	64.89 (±3.97)	68.18 (±5.61)	64.43 (±5.25)	61.59 (±5.92)	66.59 (±5.19)	66.82 (±5.59)	65.23 (±5.33)	63.64 (±6.97)	61.93 (±4.97)
5	70.91 (±5.41)	66.93 (±4.4)	68.86 (±5.43)	67.95 (±5.61)	69.32 (±5.16)	69.2 (±5.21)	67.05 (±4.9)	67.39 (±5.85)	70.0 (±5.5)	67.73 (±6.41)	66.25 (±6.83)	64.55 (±6.4)
6	70.45 (±6.1)	67.5 (±6.1)	63.64 (±3.66)	67.16 (±5.64)	68.07 (±4.63)	66.36 (±4.35)	68.52 (±8.87)	68.07 (±6.33)	66.82 (±4.34)	71.02 (±6.78)	68.64 (±6.41)	67.73 (±5.21)
7	65.91 (±6.66)	65.68 (±5.18)	62.5 (±5.54)	64.09 (±4.91)	67.5 (±5.09)	63.52 (±6.03)	61.59 (±6.62)	65.45 (±6.08)	67.5 (±4.2)	63.64 (±5.93)	63.64 (±5.88)	62.27 (±5.3)
8	77.73 (±5.59)	71.25 (±6.19)	71.82 (±5.54)	71.02 (±4.87)	73.3 (±4.18)	70.11 (±4.73)	67.27 (±6.72)	70.0 (±6.57)	78.75 (±6.94)	71.02 (±5.37)	78.18 (±5.99)	76.7 (±6.93)
9	68.75 (±5.7)	68.3 (±5.3)	65.11 (±5.89)	65.68 (±5.23)	64.77 (±5.4)	63.98 (±3.82)	65.11 (±5.2)	64.77 (±5.45)	69.09 (±6.12)	66.93 (±5.0)	71.14 (±4.38)	69.89 (±6.66)
10	68.86 (±6.19)	68.64 (±8.82)	66.93 (±9.71)	65.0 (±6.6)	65.0 (±6.52)	64.55 (±6.16)	62.61 (±7.79)	65.57 (±8.04)	67.84 (±5.53)	66.25 (±6.32)	69.2 (±6.08)	68.41 (±7.18)
11	67.39 (±5.89)	71.36 (±4.51)	68.86 (±7.26)	69.66 (±6.07)	63.07 (±7.07)	66.25 (±7.02)	62.95 (±6.67)	67.39 (±6.02)	67.73 (±6.29)	66.14 (±6.01)	65.8 (±6.72)	65.23 (±6.55)
12	68.41 (±5.52)	68.3 (±6.84)	64.89 (±5.82)	66.7 (±7.16)	68.18 (±6.01)	66.36 (±7.21)	65.57 (±7.72)	63.41 (±9.34)	65.91 (±5.79)	67.61 (±6.01)	66.36 (±5.64)	66.59 (±5.04)
13	70.91 (±5.41)	64.43 (±5.49)	67.16 (±6.53)	62.73 (±5.54)	61.93 (±5.88)	65.11 (±6.07)	64.89 (±7.89)	64.89 (±6.12)	71.36 (±6.64)	68.41 (±5.83)	69.89 (±5.13)	70.8 (±5.44)
14	67.05 (±6.84)	68.86 (±6.59)	65.45 (±4.75)	68.86 (±6.27)	70.34 (±4.74)	66.14 (±5.03)	67.5 (±6.51)	68.86 (±8.13)	69.89 (±6.09)	66.7 (±5.85)	66.36 (±6.21)	64.43 (±5.81)
15	71.02 (±4.24)	67.61 (±6.96)	70.34 (±5.95)	70.45 (±5.57)	66.82 (±7.2)	74.77 (±5.56)	69.89 (±6.18)	70.68 (±7.07)	71.59 (±5.5)	72.73 (±4.37)	73.52 (±6.02)	72.16 (±4.97)
16	66.36 (±5.31)	61.7 (±6.64)	66.14 (±5.79)	70.23 (±5.56)	48.07 (±5.2)	68.52 (±7.68)	63.75 (±7.99)	68.3 (±4.16)	66.7 (±5.94)	68.07 (±4.74)	65.11 (±4.33)	63.98 (±4.94)
17	71.25 (±5.85)	64.89 (±5.35)	74.32 (±5.89)	68.07 (±5.99)	62.73 (±4.51)	70.57 (±6.29)	66.82 (±6.72)	72.16 (±6.38)	73.3 (±5.32)	70.23 (±4.71)	68.52 (±5.76)	70.0 (±4.41)
18	76.59 (±5.8)	73.41 (±7.3)	71.25 (±6.19)	72.95 (±6.58)	68.86 (±3.94)	71.36 (±5.94)	66.82 (±6.32)	72.27 (±6.37)	75.34 (±6.64)	72.27 (±5.36)	74.43 (±5.7)	73.07 (±5.25)
19	67.5 (±8.6)	65.45 (±7.83)	65.45 (±9.87)	65.34 (±4.92)	63.75 (±5.3)	67.5 (±6.97)	64.77 (±8.73)	65.45 (±8.99)	68.98 (±7.41)	67.61 (±8.77)	67.73 (±7.42)	64.77 (±8.08)

Table A.7: Specificity (Sp) and sensitivity (Se) (expressed in percentage) achieved by the classifiers on each task.

Task #	RF$_{Sp}$	RF$_{Se}$	LR$_{Sp}$	LR$_{Se}$	KNN$_{Sp}$	KNN$_{Se}$	LDA$_{Sp}$	LDA$_{Se}$	GNB$_{Sp}$	GNB$_{Se}$	SVM$_{Sp}$	SVM$_{Se}$	DT$_{Sp}$	DT$_{Se}$	MLP$_{Sp}$	MLP$_{Se}$	ET$_{Sp}$	ET$_{Se}$	AB$_{Sp}$	AB$_{Se}$	GB$_{Sp}$	GB$_{Se}$	XGB$_{Sp}$	XGB$_{Se}$
1	70.71	56.74	48.57	64.57	75.0	50.0	59.76	57.61	72.62	42.61	68.57	51.09	70.48	49.57	64.76	54.35	69.29	58.7	70.0	52.17	69.76	60.0	70.95	59.57
2	70.0	69.57	65.71	67.39	81.9	52.61	70.48	64.78	86.43	48.26	76.67	58.26	72.14	57.83	73.33	64.35	70.48	63.48	68.1	71.96	65.48	68.7	65.0	69.57
3	69.05	63.04	60.95	71.3	78.33	55.22	78.81	63.48	80.48	53.7	80.71	55.22	75.71	49.35	74.29	65.87	65.48	64.57	74.05	57.39	67.86	64.57	68.33	65.43
4	73.33	61.52	74.29	51.3	85.48	43.04	81.67	49.57	88.33	49.78	84.52	46.09	76.19	48.26	79.76	54.57	78.33	56.3	73.1	58.04	66.43	61.09	64.05	60.0
5	72.38	69.57	75.95	58.7	80.95	57.83	83.1	54.13	92.62	48.04	90.0	50.22	75.0	59.78	83.33	52.83	77.38	63.26	71.9	63.91	66.19	66.3	62.86	66.09
6	75.95	65.43	81.67	54.57	91.19	38.48	88.33	47.83	92.86	45.43	92.62	42.39	74.76	62.83	80.95	56.3	78.81	55.87	75.48	66.96	71.43	66.09	73.81	62.17
7	70.95	61.3	60.95	70.0	73.81	52.17	69.29	59.35	89.52	47.39	68.81	58.7	65.95	57.61	68.57	62.61	73.1	62.39	65.48	61.96	66.67	60.87	63.81	60.87
8	81.19	74.57	78.1	65.0	91.19	54.13	83.57	59.57	95.71	52.83	88.1	53.7	73.81	61.3	74.29	66.09	83.57	74.35	75.95	66.52	79.76	76.74	78.33	75.22
9	68.33	69.13	62.62	73.48	69.76	60.87	73.81	58.26	87.14	44.35	73.57	55.22	65.95	64.35	69.76	60.22	70.71	67.61	66.9	66.96	67.62	74.35	69.76	70.0
10	67.62	70.0	62.38	74.35	69.05	65.0	68.33	61.96	69.29	61.09	68.81	60.65	66.19	59.35	66.9	64.35	65.71	69.78	55.71	75.87	68.33	70.0	67.38	69.35
11	68.81	66.09	74.52	68.48	82.38	56.52	74.29	65.43	77.14	50.22	76.67	56.74	70.95	55.65	65.95	68.7	70.71	65.0	67.14	65.22	67.14	64.57	65.71	64.78
12	67.62	69.13	73.57	63.48	80.0	51.09	75.95	58.26	84.05	53.7	80.48	53.48	73.81	58.04	76.9	51.09	68.33	63.7	74.52	61.3	66.19	66.52	66.19	66.96
13	64.52	76.74	62.14	66.52	65.71	68.48	62.86	62.61	50.95	71.96	57.14	72.39	59.52	69.78	62.38	67.17	63.81	78.26	67.14	69.57	64.76	74.57	66.19	75.0
14	66.19	67.83	73.33	64.78	80.48	51.74	77.62	60.87	86.67	55.43	78.1	55.22	72.62	62.83	72.14	65.87	73.33	66.74	69.05	64.57	65.24	67.39	60.71	67.83
15	71.43	70.65	65.95	69.13	79.29	62.17	74.76	66.52	64.05	69.35	84.52	65.87	74.52	65.65	73.57	68.04	71.67	71.52	79.52	66.52	71.9	75.0	71.43	72.83
16	63.57	68.91	58.33	64.78	72.62	60.22	66.67	73.48	6.9	85.65	65.95	70.87	63.57	63.91	67.38	69.13	63.57	69.57	64.76	71.09	61.19	68.7	64.76	63.26
17	75.95	66.96	80.71	50.43	84.52	65.0	82.38	55.0	90.48	37.39	84.76	57.61	71.9	62.17	82.14	63.04	82.62	64.78	83.57	58.04	68.57	68.48	70.24	69.78
18	77.38	75.87	70.48	76.09	79.29	63.91	75.95	70.22	91.19	48.48	77.14	66.09	67.62	66.09	73.57	71.09	76.67	74.13	75.95	68.91	75.0	73.91	70.95	75.0
19	69.29	65.87	65.71	65.22	73.33	58.26	75.0	56.52	86.19	43.26	77.14	58.7	65.71	63.91	68.33	62.83	72.86	65.43	68.1	67.17	68.33	67.17	64.29	65.22

Table A.8: Mean accuracy (and standard deviation), specificity, and sensitivity (in %) combining the task specific classifiers.

	RF	LR	KNN	LDA	GNB	SVM	DT	MLP	ET	AB	GB	XGB	MIX
Accuracy	84.20 (±4.34)	78.07 (±6.32)	78.98 (±5.70)	78.41 (±5.64)	75.23 (±6.18)	78.52 (±5.06)	81.36 (±6.08)	80.68 (±4.69)	83.30 (±3.82)	84.55 (±6.04)	83.86 (±5.13)	83.18 (±4.34)	85.00 (±4.90)
Specificity	87.38	79.29	97.38	90.71	98.10	96.19	92.86	87.86	89.05	90.71	86.19	84.52	90.95
Sensitivity	81.30	76.96	62.17	67.17	54.35	62.39	70.87	74.13	78.04	78.91	81.74	81.96	79.57

Table A.9: Mean accuracy (and standard deviation), specificity, and sensitivity (in %) combining the Top 3 Single Classifier models.

	RF	LR	KNN	LDA	GNB	SVM	DT	MLP	ET	AB	GB	XGB	MIX
Accuracy	87.27 (±5.35)	84.43 (±4.08)	67.73 (±6.00)	83.07 (±4.74)	86.93 (±4.81)	85.45 (±5.00)	72.95 (±7.77)	85.23 (±4.16)	88.75 (±3.90)	85.23 (±4.79)	85.91 (±5.21)	84.20 (±5.99)	88.52 (±3.97)
Specificity	86.43	86.67	93.81	89.52	83.10	88.81	72.62	86.19	89.05	88.57	86.90	85.00	87.86
Sensitivity	88.04	82.39	43.91	77.17	90.43	82.39	73.26	84.35	88.48	82.17	85.00	83.48	89.13

Table A.10: Mean accuracy (and standard deviation), specificity, and sensitivity (in %) combining the Top 5 Single Classifier models.

	RF	LR	KNN	LDA	GNB	SVM	DT	MLP	ET	AB	GB	XGB	MIX
Accuracy	87.39 (±4.46)	84.89 (±3.61)	67.73 (±6.00)	83.07 (±4.74)	86.93 (±4.81)	85.57 (±5.04)	80.11 (±5.56)	84.55 (±4.04)	88.64 (±4.13)	85.68 (±4.01)	86.36 (±5.08)	84.20 (±5.40)	87.27 (±4.90)
Specificity	86.67	86.90	93.81	89.52	83.10	88.57	78.81	85.24	90.48	88.81	87.62	84.52	86.90
Sensitivity	88.04	83.04	43.91	77.17	90.43	82.83	81.30	83.91	86.96	82.83	85.22	83.91	87.61

Appendix B

Featured Links

B.1 AlzheimInk App

In addition to the QR code provided above, the web application can also be accessed directly from the following link: AlzheimInk-App

B.2 GitHub Repository

The following link provides access to the app's GitHub repository: AlzheimInk-App GitHub
Within this, different code files can be found:

- *app.py* : This file contains the application developed in the Flask framework. To run the application locally, this file should be executed.
- *models.py* : This file contains everything related to the creation, training, validation, testing, and other functions of the different machine learning models developed.
- *app_predictor_model.py*: This file contains only the function chosen to make predictions in the application and everything it needs to work
- *results_analysis.py* : This file contains everything related to the analysis of results, graphs, and tables elaborated.
- *index.html*, *script.js*, *canvasFunctions.js*, and *styles.css*: These files are located within the templates and static folders, and they are the codes that handle the Front-end of the web application.