SAPIENZA
UNIVERSITÀ DI ROMA

4 Hands-on Seminars:

Python Machine Learning Techniques for Classifying Patients with Neurodegenerative Disorders and Controls based on EEG, MRI, and Clinical Markers

Responsible

Verónica Henao Isaza,

MSc in engineering, BSc Bioengineering

Emphasis on Neuroengineering

Grupo Neuropsicología y Conducta GRUNECO

Universidad de Antioquia

Medellín, Colombia.

Related Project (Thesis):

Machine Learning model for the classification of individuals at risk of Alzheimer's dementia from multimodal databases of EEG and clinical information.

Content

Objectives

1. Introduction and Application of Basic Machine Learning Techniques for Neuropsychological Data Classification
2. Optimization and Validation of Machine Learning Models for Predicting Cognitive Markers from Telemonitoring Data.
3. Analysis and Visualization of Machine Learning Results for the Association Between Telemonitoring Markers and Clinical Data.

Methodology

The seminar methodology is designed to combine theoretical knowledge with practical experience in using machine learning techniques for neuropsychological data classification.

Github repository: https://github.com/vhenaoi/Python-Machine-Learning-Techniques

Session 1: Introduction (Artificial Intelligence in Health and Python)
Content: Definitions of Artificial Intelligence (AI), Data Science, and Deep Learning in health.
Objective: Familiarize with key concepts and their application in neuroscience.
Workshops:

- Introduction to Python
- **Workshop 1**
  - **Script1:** Mathematics, Statistics, and Graphing in Python

Session 2: Exploratory Data Analysis and Feature Selection
Content: Techniques for exploratory analysis, descriptive statistics, and feature selection.
Objective: Perform guided practical activities in data analysis and feature selection.
Workshops:

- **Workshop 2**
  - **Script2:** Exploratory Data Analysis and Feature Selection Techniques

Session 3: Implementing Machine Learning Techniques
Content: Machine learning models, selection, and tuning parameters.
Objective: Train and implement machine learning models.
Workshops:

- Metrics AI
- **Workshop 3**
  - **Script3:** Implementing and Training Machine Learning Models

Session 4: Model Interpretation and Summary
Content: Analysis of metrics and summary of covered topics.
Objective: Interpret results and evaluate model performance.
Workshops:

- **Workshop 4**
  - **Script4:** Model Interpretation and Summary Analysis.

Methodological Details

- **Theory:** The theoretical content is provided in a PDF covering essential definitions and concepts.
- **Practice:** Practical activities will be conducted using Google Colab, an accessible platform with any Gmail email. Concepts for running the code are documented in both the code and the PDF.

Concepts

- **Artificial Intelligence (AI):** Initially defined by Professor John McCarthy in 1956, AI refers to using computers and technology to simulate human-like intelligence and reflective thinking. In healthcare, AI broadly encompasses the use of machine learning algorithms and cognitive technologies to analyze and act on healthcare data, to predict outcomes and solve problems that typically require human intelligence, such as pattern recognition and decision-making.
- **Data Science**: Integrates Big Data and Artificial Intelligence (AI). It encompasses techniques and methodologies for extracting insights from large datasets using AI and machine learning technologies.
- **Applications in Health:** AI is utilized in various healthcare processes, including automated diagnosis, interpretation of medical images (e.g., MRI and EEG), disease prediction, and personalized treatments, leading to more accurate and efficient care.

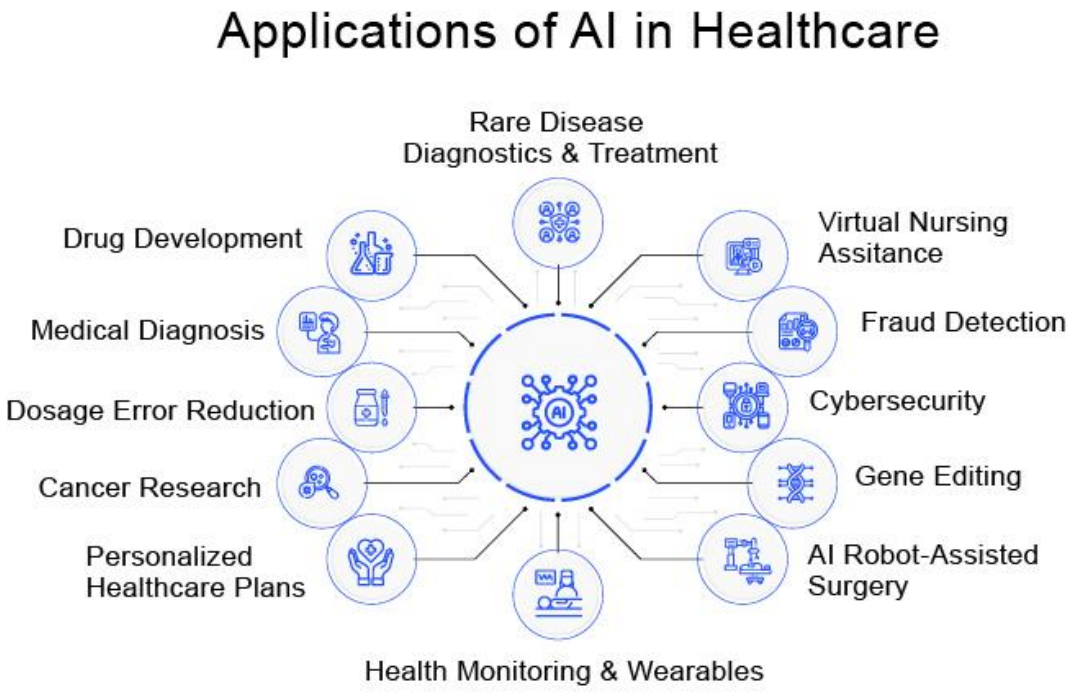


Figure 1. Applications of AI in Healthcare.

Relevance in Neurosciences

Patient Classification: Machine learning algorithms are employed to differentiate between patients with neurodegenerative diseases, like Alzheimer's, and healthy individuals.

Multimodal Data: Combining various types of data, such as EEG, MRI, and clinical information, enhances the precision and applicability of classification models.

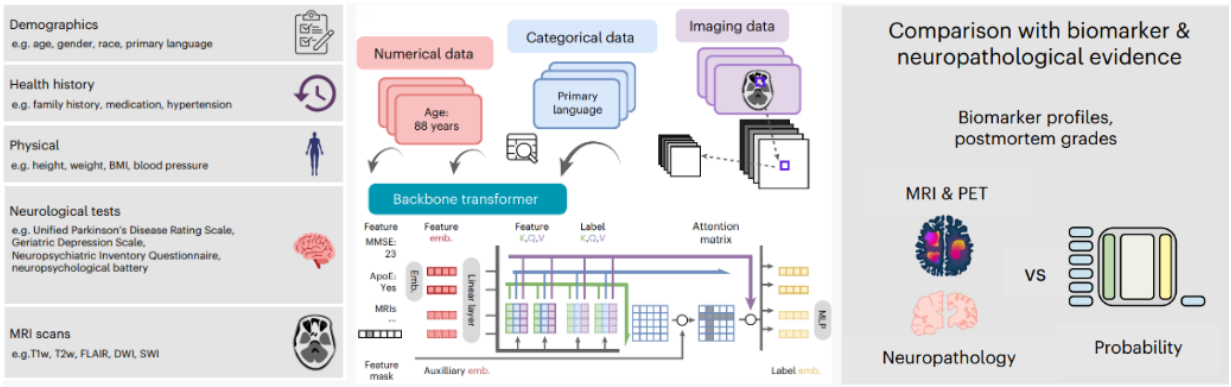


Figure 2 Relevance in Neurosciences.

Ref: https://www.nature.com/articles/s41591-024-03118-z

Types of AI in health care

AI, or Artificial Intelligence, is an umbrella term encompassing various distinct yet interrelated processes used in health care. Some of the most common types include:

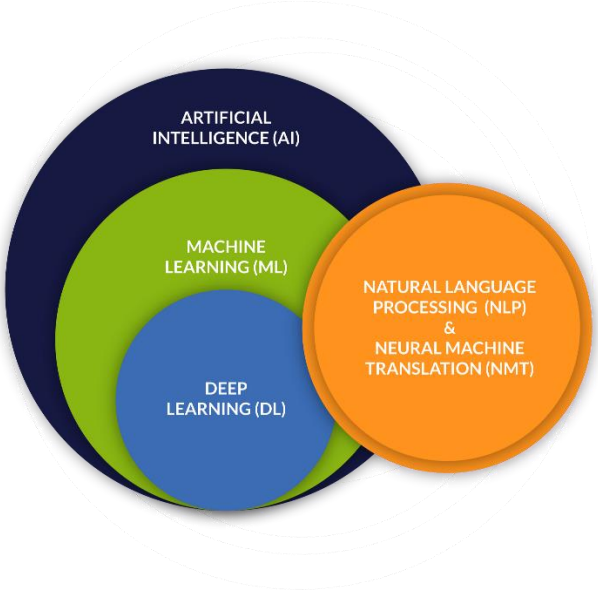


Figure 3. The Artificial Intelligence Ecosystem..

1. **Machine Learning (ML):**

A core branch of AI that involves training algorithms with data sets, such as health records, to create models capable of performing tasks like categorizing information or predicting outcomes.

- **Types of Machine Learning:**
    i) **Supervised Learning:** Predicts outcomes from a given set of features (e.g., classification).
    ii) **Unsupervised Learning:** Used when training data lacks a response variable, involving clustering and dimensionality reduction algorithms.
    iii) **Reinforcement Learning:** Focuses on balancing exploration and exploitation without the need for labeled input/output pairs, sometimes combining with supervised learning.
- **Machine Learning Workflow:**
    i) **Experimentation (Science):** Developing models.
    ii) **Product Development (Engineering):** Calibrating models with specific data.
- **Roles in Machine Learning:**
    i) **ML Algorithm Designer:** Focuses on model generation.
    ii) **ML Algorithm User:** Calibrates models with concrete data.
- **Classification Models:**
    i) **Linear Models:**
        (1) Logistic Regression
        (2) Support Vector Machines (SVM)
    ii) **Non-Linear Models:**
        (1) k-Nearest Neighbors (k-NN)
        (2) Kernel SVM
        (3) Bayesian Classification
        (4) Decision Trees
        (5) Random Forest Classification

2. **Deep Learning:**

A specialized subset of machine learning that involves large volumes of data, longer training times, and multiple layers of algorithms. It produces neural networks capable of handling more complex tasks. Representation learning through neural networks is a key aspect, focusing on finding better data representations through automatic processes.

3. **Natural Language Processing (NLP):**

Uses machine learning techniques to understand and interpret human language, whether verbal or written. In health care, NLP is commonly used to interpret documentation, notes, reports, and published research.

4. **Robotic Process Automation (RPA):**

Involves using AI in computer programs to automate administrative and clinical workflows. Health care organizations often utilize RPA to enhance patient experiences and optimize daily operations.

Examples of Artificial Intelligence Applications

To illustrate the rapidly advancing field of AI, we highlight some key applications in healthcare.

AI in Clinical Decision-Making

The massive increase in healthcare data, which doubles every three years, makes it challenging for medical professionals to stay current. AI, particularly through machine learning and natural language processing, helps manage this data overload. Tools like IBM Watson can quickly analyze electronic health records and provide treatment recommendations, significantly improving the efficiency of clinical decision-making.

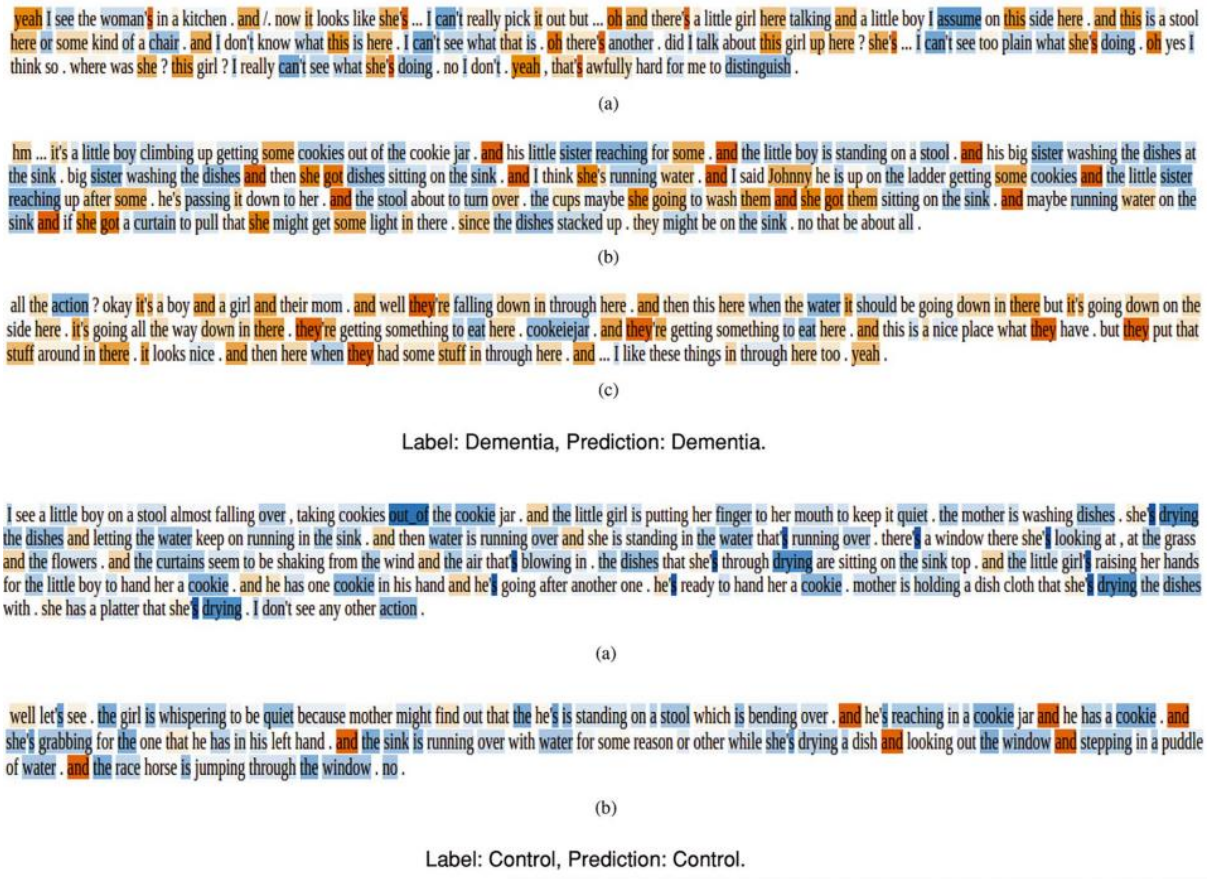


Figure 4. LIME Explanation – Textual Explanation. Reproduced.

Ref: https://link.springer.com/article/10.1007/s12559-023-10192-x

AI in Personalized Medicine

One of the most promising applications of AI in healthcare is personalized medicine, which tailors treatments to individual genetic profiles and lifestyles. This approach aims to improve treatment outcomes and reduce side effects. As AI technology advances, it will play a critical role in drug development, diagnostics, and treatment, although ethical considerations must be addressed.

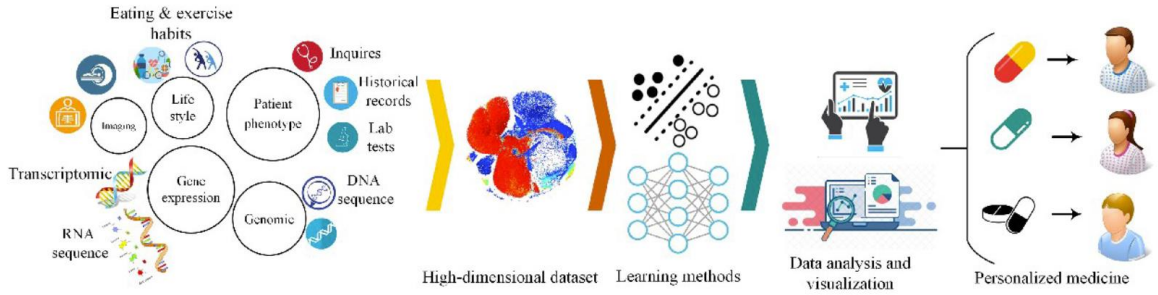


Figure 5. The integration of deep learning into clinical workflows is enhancing precision health by enabling healthcare providers to deliver more personalized and effective care to their patients.

Ref: https://thesciencebrigade.com/jst/article/view/124

AI is accelerating drug development by designing new drugs, predicting side effects, and identifying ideal candidates for clinical trials. This innovation reduces the time and cost associated with bringing new drugs to market.

AI in Medical Diagnosis

Each year, preventable medical errors in hospitals harm 400,000 patients, resulting in 100,000 deaths. AI offers a promising solution to improve diagnosis by reducing human error, often caused by incomplete medical histories and overwhelming caseloads. AI systems can predict and diagnose diseases faster than many healthcare professionals.

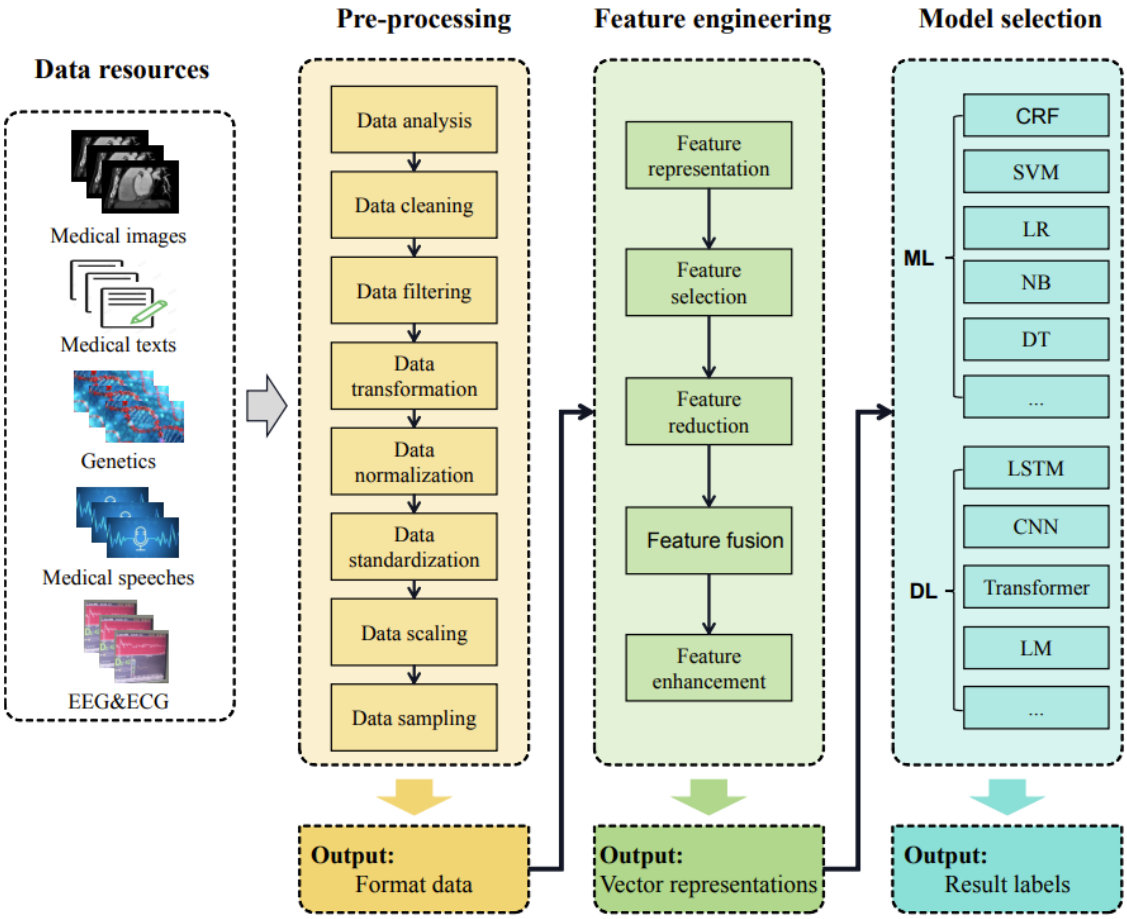


Figure 6. The framework for AI in disease diagnosis modeling (ML and DL denote machine learning and deep learning, respectively).

Ref: https://www.mdpi.com/2306-5354/11/3/219

AI in Imaging Diagnosis

In the U.S., diagnostic errors in imaging affect about 12 million patients annually, with half of these errors posing significant harm. AI has become essential in improving the accuracy of diagnostic imaging, particularly in fields like oncology. For example, AI tools can detect early signs of cancer in imaging studies, aiding in more effective treatment.

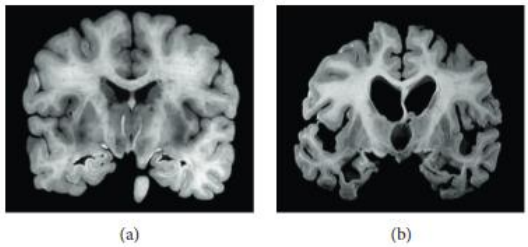


Figure 7. MRI images of normal people and AD patients.

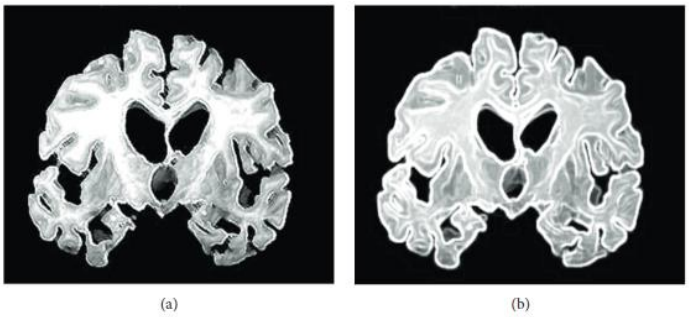


Figure 8. MRI images of AD patients processed by CNN and DML algorithms. Note. Figure 6A was an MRI image of AD patients based on CNN algorithm, and Figure 6B was a MRI image of AD patients based on DML algorithm.

Ref: https://onlinelibrary.wiley.com/doi/10.1155/2021/8198552

Example of a Personal Project
Project 1

Automatic Classification of Subjects of the PSEN1-E280A Family at Risk of Developing Alzheimer's Disease Using Machine Learning and Resting State Electroencephalography

Background: The study of genetic variant carriers provides an opportunity to identify neurophysiological changes in preclinical stages. Electroencephalography (EEG) is a low-cost and minimally invasive technique which, together with machine learning, provide the possibility to construct systems that classify subjects that might develop Alzheimer's disease (AD).

Objective: The aim of this paper is to evaluate the capacity of the machine learning techniques to classify healthy Non-Carriers (NonCr) from Asymptomatic Carriers (ACr) of PSEN1-E280A variant for autosomal dominant Alzheimer's disease (ADAD), using spectral features from EEG channels and brain-related independent components (ICs) obtained using independent component analysis (ICA).

Methods: EEG was recorded in 27 ACr and 33 NonCr. Statistical significance analysis was applied to spectral information from channels and group ICA (gICA), standardized low-resolution tomography (sLORETA) analysis was applied over the IC as well. Strategies for feature selection and classification like Chi-square, mutual informationm and support vector machines (SVM) were evaluated over the dataset.

Results: A test accuracy up to 83% was obtained by implementing a SVM with spectral features derived from gICA. The main findings are related to theta and beta rhythms, generated in the parietal and occipital regions, like the precuneus and superior parietal lobule.

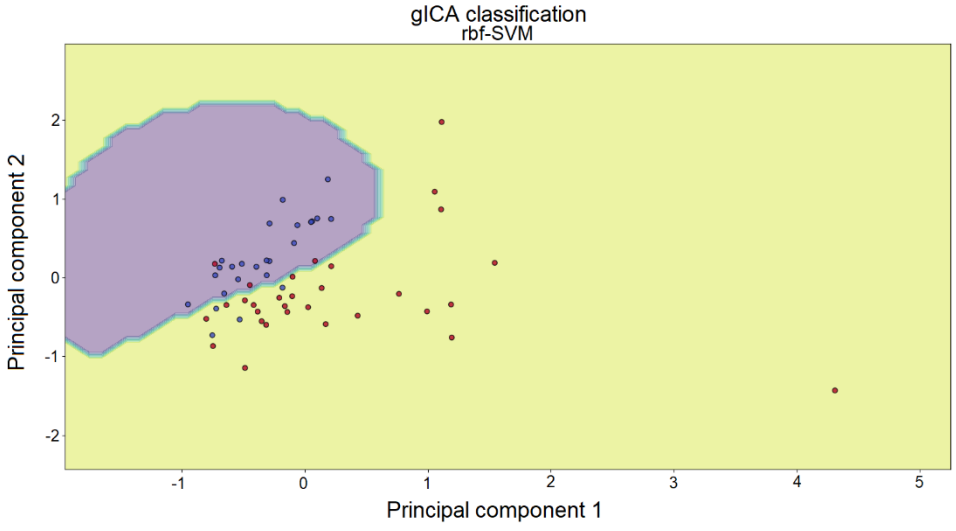
Conclusion: Promising models for classification of preclinical AD due to PSEN-1-E280A variant can be trained using spectral features, and the importance of the beta band and precuneus region is highlighted in asymptomatic stages, opening up the possibility of its use as a screening methodology.

Table I. Subjects demographics.

	ACr	*NonCr*	*ACr vs NonCr (p-value)*	*Effect size (hedges)*
N	27	33	NA	NA
Age	32.44 ± 5.800	32.70 ± 5.828	0.868	-0.043
Gender (M/F)	11/16	13/20	1.000	0.020 (Cramer)
Education level (years)	10.81 ± 2.936	13.21 ± 2.759	0.001	-0.833
Verbal Fluency	21.74 ± 3.829	22.67 ± 3.688	0.346	-0.244
Boston Naming Test	12.33 ± 3.101	13.61 ± 1.116	0.032	-0.562
MMSE	29.30 ± 0.953	29.64 ± 0.822	0.044	-0.38
Word List Recall	7.59 ± 1.551	8.39 ± 1.171	0.048	-0.584
Word List Recognition	9.78 ± 0.506	10.00 ± 0.000	0.011	-0.647
Constructional Praxis	9.96 ± 1.160	10.39 ± 0.827	0.171	-0.43
Delayed Constructional Praxis	9.04 ± 1.720	9.82 ± 1.776	0.018	-0.44

Table 2
Best models founded for classification between ACr and NonCr using relative power over gICA components

Model	Train		Test					
	Accuracy	Accuracy	ACr-Precision	NonCr-Precision	ACr-Recall	NonCr-Recall	ACr-F1	NonCr-F1
RBF-SVM	0.89	0.83	0.8	0.86	0.8	0.86	0.8	0.86
K-Neighbors	0.86	0.83	0.8	0.86	0.8	0.86	0.8	0.86
L1 normalization + Gradient Boosting	0.84	0.75	0.75	0.75	0.6	0.86	0.67	0.8



Project 2

Comprehensive Methodology for Sample Augmentation in EEG Biomarker Studies for Alzheimer's Risk Classification

Background: Dementia, characterized by progressive cognitive decline, is a major global health challenge. Alzheimer's disease (AD) is the predominant type, accounting for approximately 70% of dementia cases worldwide. Electroencephalography (EEG)-derived measures have shown potential in identifying AD risk, but obtaining sufficiently large samples for reliable comparisons remains a challenge.

Objective: This study implements a comprehensive methodology that integrates signal processing, data harmonization, and statistical techniques to increase sample size and improve the reliability of Alzheimer's disease risk classification models.

Methods: We used a multi-step approach combining advanced EEG preprocessing, feature extraction, harmonization techniques, and propensity score matching (PSM) to optimize the balance between healthy non-carriers (HC) and asymptomatic E280A mutation Alzheimer's disease carriers (ACr). Data were harmonized across four databases, adjusting for site effects while preserving important covariate effects such as age and sex. PSM was applied at different ratios (2:1, 5:1, and 10:1) to explore the impact of sample size differences on model performance. The final dataset was subjected to machine learning analysis using decision trees, with cross-validation to ensure robust model performance.

Results: Our analysis revealed significant improvements in classification accuracy when balancing sample sizes using PSM, with accuracy metrics ranging from 0.92 to 0.96 across different class distribution ratios. The refined data-driven approach enabled more accurate identification of individuals at risk for AD, even in populations with limited sample sizes.

Conclusion: This comprehensive methodology highlights the potential of integrating data processing, harmonization, and statistical balancing techniques to improve the accuracy of Alzheimer's disease risk classification models, paving the way for personalized diagnostic and therapeutic interventions.

Table 1. Summary characteristics of the databases

	Database	Group	Count	Age (Mean ± SD)	Sex (F/M)
2:1	CHBMP	HC	38	27.63 ± 6.67	13/25
	SRM	HC	31	30.77 ± 5.21	19/12
	UdeA1	ACr	68	35.81 ± 4.36	49/19
	UdeA1	HC	77	30.45 ± 4.81	47/30
	UdeA2	ACr	11	33.45 ± 3.64	9/2
	UdeA2	HC	12	31.42 ± 7.15	10/2
	Total		**237**		**147/90**
5:1	CHBMP	HC	38	27.63 ± 6.67	13/25
	SRM	HC	31	30.77 ± 5.21	19/12
	UdeA1	ACr	30	39.78 ± 2.85	21/9
	UdeA1	HC	77	30.45 ± 4.81	47/30
	UdeA2	ACr	1	43.0 ± nan	1/0
	UdeA2	HC	12	31.42 ± 7.15	10/2
	Total		**189**		**111/78**
10:1	CHBMP	HC	38	27.63 ± 6.67	13/25
	SRM	HC	31	30.77 ± 5.21	19/12
	UdeA1	ACr	14	41.86 ± 2.35	12/2
	UdeA1	HC	77	30.45 ± 4.81	47/30
	UdeA2	ACr	1	43.0 ± nan	1/0
	UdeA2	HC	12	31.42 ± 7.15	10/2
	Total		**173**		**102/71**

HC: Healthy non-carrier subjects. ACr: Asymptomatic E280A mutation Alzheimer's disease carriers.

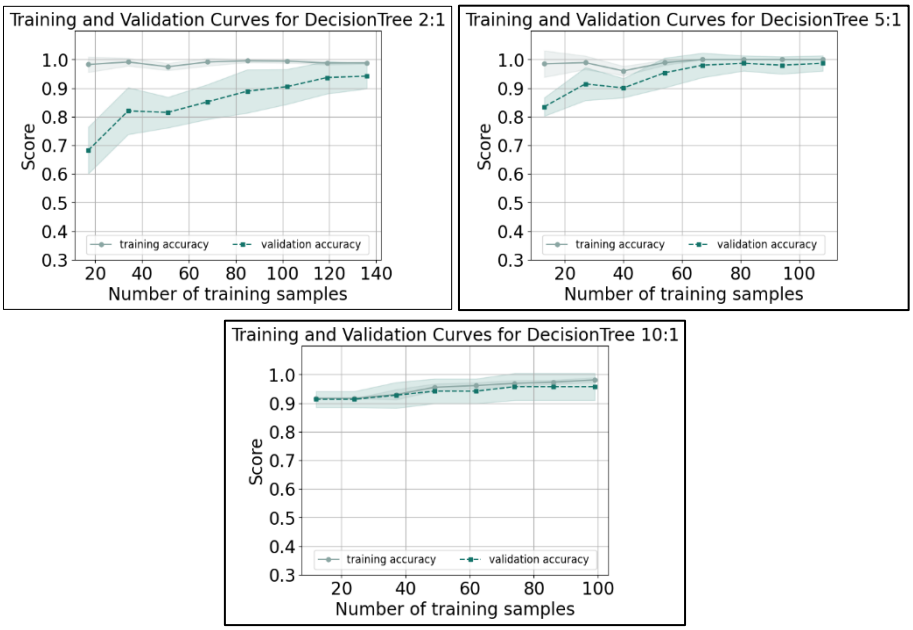


Fig 1 Comparison of accuracy between learning curves on 2:1, 5:1, and 10:1 with different ratio configurations.

Table 2. Evaluation of Model Performance Using Computer Precision

2:1	5:1	10:1
Accuracy: 91%	**Accuracy: 98%**	**Accuracy: 96%**
Precision: 91%	**Precision: 97%**	**Precision: 97%**
Recall: 97%	**Recall: 100%**	**Recall: 100%**
F1-score: 94%	**F1-score: 98%**	**F1-score: 98%**
AUC: 92%	**AUC: 99%**	**AUC: 93%**

How Much Data Do We Need?

- The amount of data required depends on various factors, including the complexity of the problem and the performance of the model. It is crucial to start by testing with the available data to assess initial model performance. From there, we can iteratively expand the dataset to improve accuracy and robustness.
- It is important to emphasize that the quality and design of the dataset are as critical as the choice of algorithm. A well-designed dataset, with representative samples and relevant features, is essential for developing effective and reliable models. Ensuring that the data is diverse and comprehensive helps the model generalize better and perform well in real-world scenarios.

How is a Model Created?

Observations: Symptoms, metrics

Predictions: Diagnosis

The creation of a model involves transforming observations into predictions. Initially, you start with example data where X represents the input observations and y represents the expected outputs or predictions. The process becomes mechanical as it matches inputs with outputs, effectively calibrating the model based on this template.

Model Creation and Calibration

1. **Model Template Selection:** Choosing the right model template is crucial. This involves understanding the types of models available and selecting a template that aligns with the structure of the data and desired outcomes. The model will then undergo a calibration process based on the template and input data. This means the model is not merely learning but is being calibrated to fit the given template.
2. **Design Considerations:** Designing the process involves several key aspects:

   Input and Output Data: Determine the quantity and types of data needed for both inputs and outputs.

   Model Template Definition: This is guided by experience, knowledge, and iterative experimentation. It involves selecting and defining the model architecture.

   Calibration Control: Manage how the model is trained and adjusted. This includes overseeing the calibration process to ensure it aligns with the desired outcomes.

   Evidence-Based Decisions: Make decisions based on data-driven evidence, ensuring that the model's predictions are supported by the data.

Data Splitting: Initially, divide your data into two groups: one for training and one for testing. It is crucial to test the model on data it has not seen during training to avoid overfitting and ensure generalization.

Recommended Data Repositories

Here is the list of recommended pages with available datasets, all in English:

1. https://www.kaggle.com/datasets
2. https://archive.ics.uci.edu/ml/index.php
3. https://datasetsearch.research.google.com/
4. https://www.data.gov/
5. https://registry.opendata.aws/
6. https://azure.microsoft.com/en-us/services/open-datasets/
7. https://data.world/
8. https://www.quandl.com/
9. https://zenodo.org/
10. https://www.europeandataportal.eu/en
11. https://openneuro.org/
12. https://www.synapse.org/
13. https://fcon_1000.projects.nitrc.org/indi/retro/MPI_LEMON.html
14. https://chbmp-open.loris.ca/
15. https://cocodataset.org

Seminar 1

Introduction (Artificial Intelligence in Health and Python)

Introduction to Python for Machine Learning

- **Advantages:** Python is popular in the AI field due to its clear and straightforward syntax, extensive range of libraries for machine learning and data analysis, and active community.
- **Key Libraries:**
    - **NumPy and pandas:** For data manipulation and analysis.
    - **Scikit-Learn (sklearn):**
        - A popular Python library for supervised machine learning.
        - Integrates well with the SciPy stack, making it robust and powerful.
        - Used for both classification and regression problems.
    - **TensorFlow and Keras:** For building and training deep neural networks.

Installation and Setup

- **Development Environments:** Python can be installed directly on your computer, allowing you to create more customized workflows and use user-friendly interfaces like Visual Studio, Anaconda, or the system console (cmd).

  In this seminar, we will exclusively use Google Colab, an interactive, cloud-based environment that allows you to run Python code and visualize results in real-time. Google Colab is accessible from any web browser and requires no prior installation, making it a convenient option for quickly getting started without the need for extensive setup.

- **Package Installation:** Essential libraries will be installed using pip, a Python package manager. In Google Colab, this process is automated and straightforward, further simplifying the initial setup.

- **Additional Recommendation:** If you're interested in continuing to use Python regularly after this seminar, we recommend installing Python directly on your computer. This will allow you to create more customized workflows and take advantage of more robust development environments suited to your specific needs.

Workshop 1

Introduction to Python

1. What is Python?

Python is an interpreted, dynamically typed programming language known for its simplicity and readability. It allows for the development of everything from simple scripts to complex applications.

2. Basic Data Types:

Numbers: Includes integers (int) and floating-point numbers (float).

Strings: Sequences of characters defined within single (') or double (") quotes.

3. Operators:

Division:

/ performs floating-point division (results in a decimal number).

// performs integer division (results in the nearest lower integer).

4. Comments:

Single-line Comments: Created using #.

Multi-line Comments: Can be created using ''' or """.

5. Type Conversion:

Functions like int(), float(), and str() are used to convert between different data types.

Lists

1. What is a List?

A list is an ordered, mutable collection of items that can be of different types. It is defined within square brackets [].

2. Basic Operations:

Accessing Elements: Using indices, for example, my_list[0] accesses the first element.

Modification: Lists are mutable; you can change their elements after creation.

Slicing: Allows extracting a subsequence from the list. Example: my_list[1:3] returns elements from index 1 to 2.

Length: Use len(my_list) to get the total number of elements in the list.

3. Common Methods:

Adding Elements: append(), extend().

Removing Elements: remove(), pop().

Strings

1. What is a String?

A string is a sequence of characters defined within single or double quotes.

2. Basic Operations:

Indexing and Slicing: You can access parts of the string using indices. Example: string[1:4] extracts a substring.

Common Methods: upper(), lower(), split() for manipulating and analyzing strings.

Functions

1. What is a Function?

A function is a block of code designed to perform a specific task. It is defined using def function_name(parameters):.

2. Calling Functions:

You invoke the function by its name and pass the necessary arguments.

3. Returning Values:

Functions can return values using return.

Dictionaries

1. What is a Dictionary?

A dictionary is a collection of key-value pairs, defined within curly braces {}.

2. Basic Operations:

Accessing Values: Using keys, for example, my_dict['key'].

Adding or Modifying: You can add or modify values using my_dict['key'] = value.

Iterations

1. For Loops:

Allow iterating over a sequence of elements. Example: for element in list:.

2. While Loops:

Execute a block of code as long as a condition is true.

Introduction to Libraries

1. What is a Library?

A library is a collection of pre-defined modules and functions that can be used to simplify common tasks.

2. Common Libraries:

NumPy: Provides support for arrays and advanced mathematical operations.

Pandas: Offers data structures and data analysis tools.

Matplotlib: Used for creating plots and visualizations.

Data Manipulation with Pandas

1. Data Structures:

DataFrames: Two-dimensional tables that can contain different types of data.

Series: Columns of data in a DataFrame.

2. Basic Operations:

Reading Data: You can read data from CSV files, Excel, etc., using functions like pd.read_csv().

Manipulation: Includes selecting, filtering, grouping, and modifying data in a DataFrame.

Graphs and Visualization

1. Using Matplotlib:

Basic Setup: import matplotlib.pyplot as plt.

Creating Plots: Use plt.plot() for line plots, plt.bar() for bar charts, among others.

Advanced Concepts

1. Object-Oriented Programming (OOP):

Python supports OOP, allowing the creation of classes and objects to organize code.

2. Exception Handling:

Use try, except to handle errors during code execution.

3. File Handling:

You can open, read, and write files using functions like open(), read(), write().

Script 1

Libraries in Python

Common libraries like NumPy, pandas, and Matplotlib are essential for numerical operations, data manipulation, and creating visualizations.

These libraries extend Python's capabilities, making it easier to perform complex tasks with simple function calls.

DataFrames

A DataFrame is a two-dimensional data structure in pandas that stores data in a tabular format (rows and columns).

Data Exploration Techniques

Initial DataFrame Exploration:

Methods like head(), columns, index, shape, describe(), and info() provide an initial understanding of a dataset's content, size, structure, and characteristics.

Details:

- *head():* Displays the first few rows of the DataFrame.
- columns: Lists all column names, providing an overview of the data fields.
- *index:* Shows the row labels or indices, indicating how the rows are identified.
- shape: Returns a tuple representing the number of rows and columns, indicating the dataset's dimensions.
- *describe():* Provides statistical summaries of numerical columns, offering insights into the data distribution.
- *info():* Gives a concise summary of the DataFrame, including data types and the presence of missing values.

File Path Specification

Specifying the file path correctly is crucial for accessing data stored in different locations within the file system.

Loading Data from Excel Files

The pd.read_excel() function reads data from Excel files into a DataFrame.

Understanding .describe()

The .describe() function provides a statistical summary of the numerical columns in a DataFrame.

Metrics:

- *Count:* Number of non-null values in each column, indicating data availability.
- *Mean:* Average value, showing the central tendency of the data.
- Standard Deviation (Std): Measures data spread around the mean; higher values indicate greater variability.
- *Min:* The smallest value in the column.
- *1st Quartile (25%):* The value below which 25% of the data falls.
- *Median (50%):* The middle value, dividing the data into two equal parts.
- *3rd Quartile (75%):* The value below which 75% of the data falls.
- *Max:* The largest value in the column.
- *Additional Information:* When applied to non-numerical columns, .describe() provides metrics like count, unique values, most frequent value, and its frequency.

Grouping in Pandas (groupby)

The groupby function splits data into groups based on criteria and applies operations on these subsets independently.

Process:

- *Splitting:* The DataFrame is divided into groups based on unique values in one or more columns.
- *Applying:* Functions like counting, finding max/min values, or summing are applied to each group.
- *Combining:* Results from each group are combined into a new DataFrame.

Enables focused analysis on subgroups within the dataset, offering insights that may be obscured in the overall data.

Aggregating Data

Aggregation operations like finding maximum and minimum values within groups provide insights into the range of data.

Helps to identify extreme values and understand the variability within each group.

Seminar 2

Exploratory Data Analysis and Feature Selection (Exploratory Data Analysis, Feature Selection Techniques, Descriptive Statistics)

MLHOps: Machine Learning for Healthcare Operations

Faiza Khan Khattak[a], Vallijah Subasri[a,b,c], Amrit Krishnan[a], Elham Dolatabadi[a], Deval Pandya[a], Laleh Seyyed-Kalantari[d], Frank Rudzicz[a,c,e]

[a] *Vector Institute for Artificial Intelligence, Toronto, Ontario, Canada*
[b] *Hospital for Sick Children, Toronto, Ontario, Canada*
[c] *University of Toronto, Toronto, Ontario, Canada*
[d] *York University, Toronto, Ontario, Canada*
[e] *Dalhousie University, Halifax, Nova Scotia, Canada*

Machine Learning Health Operations (MLHOps) integrates processes to ensure the reliable, efficient, usable, and ethical deployment and maintenance of machine learning models in healthcare. This paper surveys current practices and offers guidelines for developers and clinicians on how to deploy and maintain models in clinical settings. It discusses foundational concepts of machine learning operations, including the setup of MLHOps pipelines, which encompasses data sources, preparation, and tools. Additionally, it covers long-term aspects such as monitoring, updating models in response to data shifts, and ethical considerations like bias, fairness, interpretability, and privacy. The work provides comprehensive guidance from model conception through ongoing deployment.

Model Lifecycle:

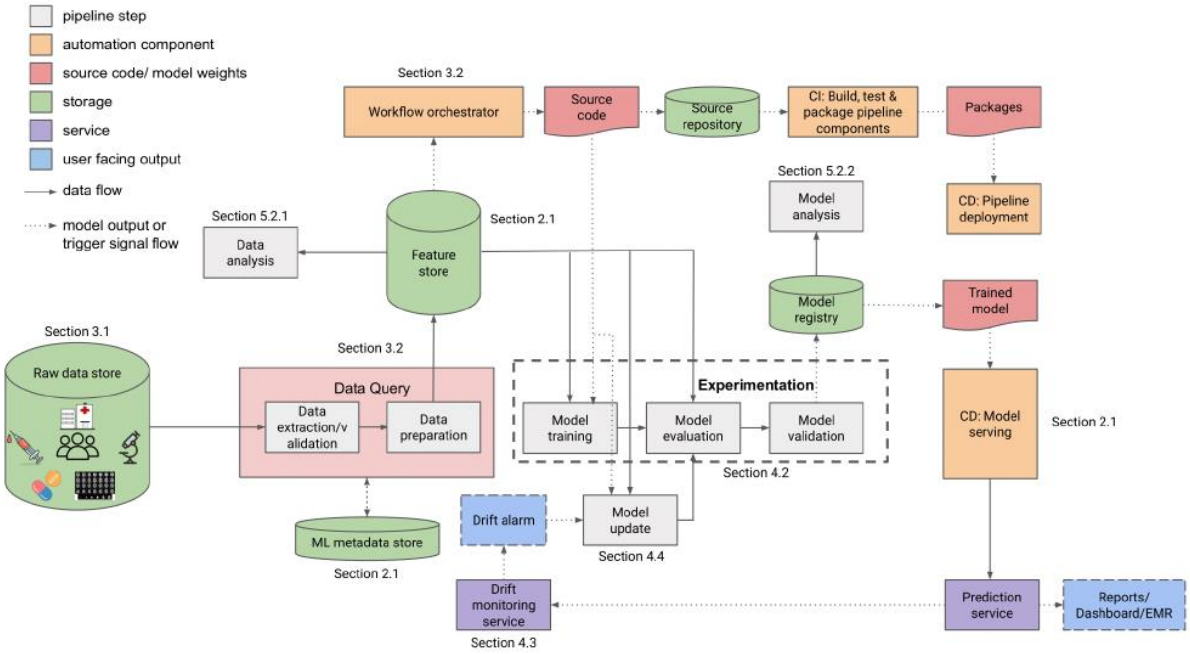


Figure 9. MLOps pipeline.

Workshop 2

Recommended pages for graphics

Data-to-Viz

- **Choose the Right Chart**: Helps you select the best chart type based on your data and objectives. Whether you need to visualize comparisons, distributions, or relationships, this resource guides you to the most appropriate visualization tool.
- **Learn from Examples**: Offers a variety of examples and tutorials to illustrate how different types of visualizations work. You can explore practical applications and see which types of charts perform best for different data scenarios.
- **Interactive Explorer**: An interactive tool that provides recommendations tailored to your specific data needs. This feature helps you navigate through various visualization options to find the best fit for your dataset.

Adobe Color

- **Create Stunning Color Schemes**: Allows you to generate harmonious color palettes using various color rules like complementary, analogous, and triadic. Ideal for designers and creatives who need balanced and appealing color combinations.
- **Experiment with Color Variations**: Lets you explore and adjust hues, shades, and tints to suit your project's requirements. The tool provides real-time visualization of how different colors work together.
- **Save and Share Palettes**: You can save your favorite color schemes to your Adobe Creative Cloud library and share them with your team or community. This feature ensures consistent branding and design across projects.

Script 2

DataFrame

Definition: A DataFrame is a two-dimensional data structure with labeled axes (rows and columns). Each column can hold different types of data. In Python, pandas is a popular library for working with DataFrames.

Columns and Rows:

Columns: Represent variables or features in the data. For example, in an EEG DataFrame, you might have columns for power in different frequency bands.

Rows: Represent individual observations or records in the DataFrame. Each row could represent a subject or a measurement instance.

Frequency Bands

Definition: Ranges of frequencies in an EEG signal, used to classify brain activity into different frequency ranges.

Groups

Definition: Classifications in data analysis, such as:

PDD (Alzheimer's Cognitive Risk): A group of patients at risk of Alzheimer's.

HC (Healthy Controls): A group of healthy individuals used for comparison.

Preprocessing Strategies

Normalization: Adjusting data to a standard scale, such as the range [0, 1], to ensure comparability.

Filtering: Removing unwanted noise from the EEG signal to improve data quality.

Statistical Analysis

ANOVA (Analysis of Variance): A statistical test that compares the means of three or more groups to determine if at least one group differs significantly from the others. It is used to evaluate variability within and between groups.

Regression: A statistical method for modeling the relationship between a dependent variable and one or more independent variables. It is used to predict the value of the dependent variable based on the independent variables.

T-test: Compares the means of two groups to determine if there is a significant difference between them.

Fisher Test: Similar to the T-test, but primarily used for comparing proportions in contingency tables.

Mann-Whitney U Test: A non-parametric test used to compare two independent groups to assess if they come from the same distribution.

Effect Size:

Cohen's d: Measures the difference between two means in terms of standard deviation units. Effect sizes are: Small, Medium or Large.

r: Correlation that measures the strength of the relationship between two variables.

Data Visualization

Histogram: Shows the distribution of a continuous variable by dividing the data into bins and counting the frequency of data within each bin.

Boxplot: Visualizes the distribution of data through quartiles, showing the median, upper and lower quartiles, and potential outliers.

Boxplot with Swarmplot: Adds individual data points on top of the boxplot to show detailed distribution.

Violin Plot: Combines features of the boxplot and KDE (Kernel Density Estimate) to show the distribution and density of the data, including a horizontal line for the median.

Preprocessing and Data Analysis

Recommendations:

- BIDS (Brain Imaging Data Structure): A standard for organizing and storing brain imaging data to facilitate sharing and reuse.
- Quality Control: The process of ensuring that data is accurate and reliable. It can include checking data integrity and identifying artifacts.
- Harmonization: Techniques for adjusting data from different sources to make them comparable, minimizing differences that are not due to the variables of interest.
- Matching Between Subjects: Technique for pairing subjects from different groups based on similar characteristics to control for confounding variables.
- Using PSM (Propensity Score Matching): Technique for pairing subjects based on the estimated probability of receiving a treatment, considering covariates like age and sex, to reduce bias in observational studies.

Feature Extraction

Definition: The process of identifying and extracting relevant features from the data, such as power in specific bands or coherence measures.

Power: The amount of electrical activity in a specific frequency band.

Entropy: A measure of the complexity or uncertainty in the EEG signal.

Coherence: Measures the phase relationship between two EEG signals from different brain regions.

Correlation Matrix

Definition: Displays pairwise correlations between features, helping to identify multicollinearity.

Code Summary and Operations

Min-Max Scaling: Uses MinMaxScaler to scale values between 0 and 1. This is crucial for techniques that require features to be on the same scale.

Column Name Adjustment: Replaces '-' characters in column names to maintain consistency.

Column Selection: Filters the DataFrame to include only the desired columns for analysis.

Label Encoding: Uses LabelEncoder to convert categorical variables into numerical values.

Boruta

Definition: Boruta is a feature selection algorithm designed to identify all relevant features in a dataset. It is particularly useful for dealing with high-dimensional data and works well with random forests.

- **How It Works:**

  Random Forests: Boruta uses a random forest classifier to assess feature importance.

  Shadow Feature Method: It creates shadow features by randomly shuffling the values of the original features and compares the importance of the original features with these shadow features.

  Importance Comparison: Features are classified into three categories:

  Important: Features that have a significantly higher importance score than the shadow features.

  Not Important: Features that have lower importance scores compared to the shadow features.

  Tentative: Features that are neither significantly more important nor less important than shadow features.

- **Steps:**

  Generate Shadow Features: Randomly permute the values of each feature to create shadow features.

  Train Random Forest: Train a random forest model on the dataset including the shadow features.

  Compare Importances: Evaluate the importance scores of the original features and shadow features.

  Feature Classification: Determine which features are relevant based on their importance compared to shadow features.

- **Advantages:**

  Robustness: Handles noisy data and can identify all relevant features.

  Non-Linearity: Works well with non-linear relationships between features and the target variable.

  Usage: Commonly used in machine learning pipelines to reduce dimensionality and improve model performance by selecting the most informative features.

Seminar 3

Analysis Implementing Machine Learning Techniques (Types of Models, Model Selection, Model Parameters, Model Training and Execution)

Workshop 3:

Metrics AI

Explanation of Metrics

Accuracy

Proportion of correct predictions out of the total number of predictions.

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

Where:

TP = True Positives

TN = True Negatives

FP = False Positives

FN = False Negatives

Sensitivity (or Recall)

Proportion of true positives among the actual positives. It measures how well the model identifies positive instances.

$$Sensitivity = \frac{TP}{TP + FN}$$

Specificity

Proportion of true negatives among the actual negatives. It measures how well the model identifies negative instances.

$$Specificity = \frac{TN}{TN + FP}$$

Precision

Proportion of true positives among all positive predictions made by the model. It reflects the accuracy of the positive predictions.

$$Precision = \frac{TP}{TP + FP}$$

F1 Score

Harmonic mean of precision and sensitivity (recall). It provides a single metric that balances precision and recall.

$$F1\ Score = \frac{2\ .\ (Precision\ .\ Recall)}{Precision + Recall}$$

RMSLE (Root Mean Squared Logarithmic Error)

Measures the mean squared error on the logarithm of the predicted values, typically used when predictions are in the form of counts or probabilities.

$$RMSLE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\log(\hat{y}_i + 1) - \log(y_i + 1))^2}$$

Where:

$\hat{y}_i = Predicted\ values$

$y_i = Actual\ values$

Log Loss

Measure of the performance of a classifier based on the probabilities it assigns to each class. It penalizes false classifications with a cost proportional to the confidence of the incorrect prediction.

$$Log\ Loss = -\frac{1}{n}\sum_{i=1}^{n}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)]$$

Where:

$\hat{y}_i = Predicted\ probabilities$

$y_i = Actual\ binary\ outcomes$

MSE (Mean Squared Error)

Average of the squared errors between predicted values and actual values.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2$$

RMSE (Root Mean Squared Error)

Square root of the MSE. It provides the standard deviation of the residuals (prediction errors) and is in the same units as the target variable.

$$RMSE = \sqrt{MSE}$$

MAE (Mean Absolute Error)

Average of the absolute errors between predicted values and actual values. It measures the average magnitude of the errors without considering their direction.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i|$$

AUC (Area Under the Curve)

Refers to the area under the Receiver Operating Characteristic (ROC) curve. It measures the model's ability to discriminate between positive and negative classes.

Interpretation: A value of 1 indicates perfect discrimination, while a value of 0.5 indicates no discrimination.

Script 3

Concepts and Their Functionality

1. **Logistic Regression**
    - **Theory:** Logistic regression is a type of regression analysis used for binary classification models. Instead of predicting continuous values, it predicts probabilities that an instance belongs to one of two classes. The logistic function, or sigmoid function, transforms a linear combination of independent variables into a probability that ranges between 0 and 1.
    - **Application:** It is used to predict the probability of belonging to a category based on observed features, such as classifying patients into different groups based on EEG features.
2. **Standardization and Normalization**
    - **Theory:** Standardization (or z-score normalization) involves transforming data to have a mean of 0 and a standard deviation of 1. Normalization (or Min-Max Scaling) scales data to a specific range, typically between 0 and 1. Both techniques help improve the performance of many machine learning algorithms by ensuring features are on a comparable scale.
    - **Application:** It is applied to ensure that all features contribute equally to the model, preventing variables with wider ranges from dominating the training process.
3. **Label Encoding**
    - **Theory:** Machine learning algorithms often require categorical variables to be converted into numeric values. Label encoding converts each category value

into an integer. This is essential for the algorithms to interpret and process categorical labels**.**

- **Application:** In classification tasks, categories such as 'HC' (Healthy Controls) and 'PDD' (Parkinson's Disease Dementia) are converted into numbers so the model can learn to differentiate between them.

4. **Data Manipulation**
    - **Theory:** Data manipulation includes operations like renaming columns, creating new columns based on existing ones, and selecting subsets of data. These techniques are crucial for preparing data for analysis or modeling, ensuring variables are consistent and in the right format.
    - **Application:** Manipulating column names and selecting relevant features help organize and structure the data for easier analysis and modeling, facilitating interpretation and processing.
5. **EEG Features**
    - **Theory:** EEG (electroencephalography) features are categorized into frequency bands such as Delta, Theta, Alpha, Beta, and Gamma, each associated with different brain states and cognitive activities. These bands reflect various types of electrical activity in the brain.
    - **Application:** They are used to study brain activity patterns associated with different clinical conditions or cognitive states. In analysis, these bands are segmented and combined with demographic information to build predictive models regarding cognitive health or disease risk.
6. **DataFrames and Data Structures**
    - **Theory:** A DataFrame is a two-dimensional data structure that organizes data into rows and columns, allowing efficient handling of tabular data. It is fundamental for data manipulation, analysis, and visualization in machine learning.
    - **Application:** DataFrames facilitate operations such as filtering, aggregating, and transforming data efficiently, enabling the preprocessing necessary to prepare data for predictive modeling.

Integration in Data Analysis Context

In a data analysis project, these techniques and concepts are integrated to prepare and transform data so that it is suitable for building and evaluating predictive models. For instance, when analyzing EEG data to predict neurodegenerative disease risk, extensive preprocessing is performed, including feature standardization, label encoding, and column name manipulation to ensure the data is consistent and appropriate for modeling.

These steps are crucial for ensuring that machine learning models can learn meaningful patterns and make accurate predictions based on available data. Proper data preparation and

manipulation allow models to train effectively and generate useful results for research or clinical practice.

1. **ANOVA (Analysis of Variance)**

   ANOVA is a statistical method used to determine if there are significant differences between the means of three or more groups. It helps in comparing the means of different groups and assessing whether any observed differences are statistically significant.

   In the code, ANOVA is used to assess whether the metric values differ significantly across different groups (e.g., healthy controls vs. patients). This is achieved through the ols function to fit an Ordinary Least Squares (OLS) model and then performing ANOVA using sm.stats.anova_lm.

2. **Kolmogorov-Smirnov Test**

   The Kolmogorov-Smirnov test is a non-parametric test used to determine if a sample comes from a specific distribution. In this case, it checks if the distribution of the metric values follows a normal distribution.

   This test is applied to the metrics to check for normality. If the p-value is less than 0.05, indicating that the data significantly deviates from a normal distribution, a logarithmic transformation is applied to the metric to stabilize variance and make the data more normally distributed.

3. **Tukey's Post-Hoc Test**

   Tukey's Honestly Significant Difference (HSD) test is used after an ANOVA to determine which specific group means are different. It helps in identifying which pairs of groups have statistically significant differences.

   Tukey's test is performed on the metrics to further investigate the differences between group means identified by ANOVA.

4. **Feature Selection with ANOVA**

   Feature selection is a process used to identify and select the most important features for model building. ANOVA-based feature selection uses statistical tests to score features based on their ability to discriminate between classes.

   SelectKBest with f_classif is used to select the top 30 features that have the highest ability to discriminate between classes. This helps in reducing the dimensionality of the dataset and improving model performance.

5. **Correlation Analysis**

   Correlation analysis measures the strength and direction of the linear relationship between features. High correlation between features can lead to redundancy and multicollinearity in the model.

A correlation matrix is computed for the selected features, and features with a correlation higher than 0.9 are identified and removed to reduce redundancy and improve model interpretability.

6. **Data Splitting and Standardization**

   Splitting the dataset into training and testing sets is crucial for evaluating model performance and avoiding overfitting. Standardization scales features to have a mean of 0 and a standard deviation of 1, which helps many machine learning algorithms perform better.

   The dataset is split into training and testing sets, and features are standardized using StandardScaler. This ensures that the model is trained and evaluated on data with comparable scales.

Classifiers and Their Evaluation

1. **Linear Discriminant Analysis (LDA)**

   LDA is a dimensionality reduction technique and classifier that seeks to project features in such a way that the classes are as separable as possible. It works well when the data is normally distributed and the classes have similar covariance matrices.

   The LDA model is trained on the training set and evaluated on the test set. Accuracy and detailed classification metrics (precision, recall, F1-score) are printed to assess performance.

2. **Logistic Regression**

   Logistic Regression is a statistical model used for binary classification that estimates probabilities using the logistic function. It is well-suited for problems where the relationship between features and the outcome is linear.

   A Logistic Regression model is trained and evaluated similarly, with the performance metrics printed out.

3. **k-Nearest Neighbors (k-NN)**

   k-NN is a non-parametric classification algorithm that assigns the class of a sample based on the majority class among its k-nearest neighbors. It is simple and effective for many problems but can be computationally expensive with large datasets.

   The k-NN model with 3 neighbors is trained and tested. The accuracy and classification report are provided.

4. **Random Forest Classifier**

   Random Forest is an ensemble method that combines multiple decision trees to improve classification accuracy and control overfitting. It aggregates the predictions from several trees to make a final decision.

   The Random Forest model with 100 trees is trained and evaluated. Performance metrics are displayed.

5. **Support Vector Machine (SVM)**

   SVM is a powerful classification algorithm that finds the optimal hyperplane that separates classes in high-dimensional space. It can use different kernel functions to handle non-linearly separable data.

   The SVM model with a linear kernel is trained and tested. The performance is assessed through accuracy and classification report.

6. **Decision Tree Classifier**

   Decision Trees are a model that splits the data into subsets based on feature values, leading to a tree-like structure of decisions. They are interpretable but can overfit if not pruned.

   The Decision Tree model is trained and evaluated. The accuracy and classification metrics are printed.

Considerations

- Model Selection: The choice of model may depend on the specific characteristics of the data, such as its dimensionality, distribution, and the problem's nature.
- Feature Engineering: Feature selection and transformation steps are crucial for improving model performance.
- Hyperparameters: Some models, like k-NN and Random Forest, have hyperparameters (e.g., number of neighbors, number of trees) that can be tuned to optimize performance.

The results from these models will provide insights into which classifier performs best on the given dataset, helping to choose the most appropriate model for the task at hand.

Hyperparameter Tuning

1. What is Hyperparameter Tuning?

Hyperparameter tuning is the process of finding the optimal set of parameters for a machine learning model. Hyperparameters are configurations that determine how the model is trained, such as the number of trees in a random forest or the learning rate in a neural network. Adjusting these parameters can improve the model’s performance.

2. Tuning Methods

- **Random Search**: Instead of testing all possible combinations of parameters, random search selects and evaluates a random subset of combinations. This method is useful when there are many hyperparameters and possible values, as it can find good configurations in less time compared to exhaustive search.
- **Grid Search**: Tests all possible combinations of parameters in a defined grid. Although it is more comprehensive, it can be very time-consuming and resource-intensive when there are many parameters or many possible values for each parameter.

3. Model Evaluation

After finding the best hyperparameters, the model is evaluated to measure its performance. Common evaluation metrics include:

- **Accuracy**: The proportion of correct predictions out of the total number of predictions made. It gives a general idea of the model's performance.
- **Classification Report**: Provides additional metrics such as precision, recall, and F1-score for each class. These metrics help understand the model's performance across different classes, which is particularly useful for imbalanced classification problems.
- **Confusion Matrix**: Displays the count of true and false predictions organized in a table. It helps visualize the model's performance in terms of true positives, false positives, true negatives, and false negatives.
- **AUC-ROC (Area Under the Receiver Operating Characteristic Curve)**: For binary classification problems, it measures the model's ability to distinguish between positive and negative classes. A higher AUC indicates better performance.

4. Results Analysis

Once the model evaluation results are obtained, analyzing them helps compare the performance of different models. This can be done using graphs that show the distribution of cross-validation scores or tables summarizing the best parameters found and their corresponding performance.

Seminar 4

Model Interpretation and Summary
(Metrics Analysis, Plots, Summary of Topics Covered)
Training and Validation Curves
Computer precision (Accuracy, Presion, Recall, F1-score, AUC)
Confusion matrix)

Workshop 4:

Script 4

Organization and Analysis of EEG Data

1. Data Extraction and Processing

EEG data analysis begins with the extraction and processing of brain feature data.

2. Data Transformation and Cleaning

Once the data is loaded, it is essential to perform cleaning and transformation to ensure that the information is in the right format for analysis. This includes standardizing column names for clarity. For example, the names of columns indicating different metrics and frequency bands are adjusted to maintain a consistent and understandable format.

3. Descriptive Analysis

Descriptive analysis involves evaluating the distribution of metrics across different groups. Visualizations such as histograms and boxplots are used to display how metrics (like power or entropy) are distributed within each group of interest. This analysis helps identify patterns and significant differences between groups.

4. Comparative Analysis

Comparative analysis is crucial for understanding differences between groups. For instance, differences between the means of metrics for experimental and control groups are calculated, as well as between different components of the EEG signal. This analysis allows for the evaluation of how specific features of the EEG signal may vary under certain conditions, such as disease risk.

5. Predictive Modeling

Predictive modeling involves using statistical and machine learning techniques to classify or predict based on EEG data. Data is divided into training and test sets to develop and evaluate models that can identify relevant patterns for classifying states or groups. Techniques such as Random Forest and cross-validation are employed to train and assess these models, adjusting parameters to improve prediction accuracy.

6. Interpretation and Visualization

Finally, the results of the analysis are interpreted and presented visually to facilitate understanding. This includes generating bar charts, confusion matrices, and ROC curves that illustrate model performance and group differences. Clear visualization of these results is crucial for communicating important findings and making informed decisions based on the data.

Interpretation of results

1. Confusion Matrix

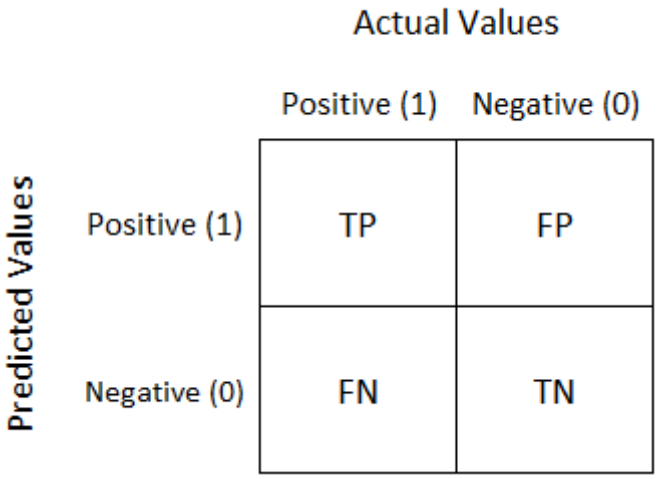


Figure 10. Interpretation of confusion matrix.

Interpretation:

A confusion matrix shows the performance of a classification model by comparing the predicted labels to the true labels. It provides insights into true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN).

Components:

- **True Positives (TP):** Correctly predicted positive cases.
- **True Negatives (TN):** Correctly predicted negative cases.
- **False Positives (FP):** Incorrectly predicted positive cases (Type I error).
- **False Negatives (FN):** Incorrectly predicted negative cases (Type II error).

Example Analysis:

- If a model has high TP and TN, and low FP and FN, it indicates good performance.

Considerations:

- Using only 20% of the data for testing is common and often sufficient if the data is representative. However, you should ensure that the split is random and representative of the overall data distribution.
- If the confusion matrix shows high FP or FN, consider adjusting model parameters or improving data quality.

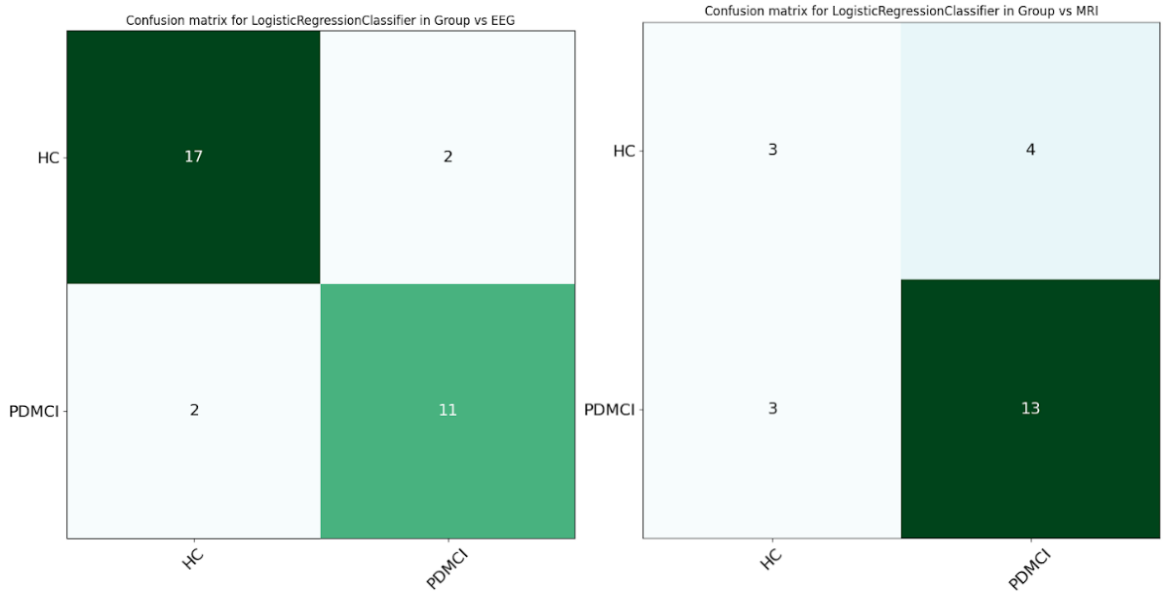


Figure 11 Examples confusion matrices.

In Figure 11, The confusion matrices reveal that the model using EEG data outperforms the MRI-based model in distinguishing between Healthy Controls (HC) and Parkinson's Disease Mild Cognitive Impairment (PD-MCI). The EEG model correctly classified 17 out of 19 HC samples and 11 out of 13 PD-MCI samples, with only two misclassifications in each class. In contrast, the MRI model struggled more with HC classification, correctly identifying only 3

out of 7 HC samples, while performing better with PD-MCI (13 correct out of 16). Overall, EEG data provided more accurate classifications than MRI data in this scenario.

2. Residual Plot

Interpretation:

Residual plots help diagnose the fit of a regression model by plotting the residuals (the differences between observed and predicted values) against the predicted values or other variables.

Components:

- **Red Line:** Represents the zero residual line (where the model perfectly predicts the target variable).
- **Points:** Represent residuals for individual observations.

Example Analysis:

- **Random Distribution:** If residuals are randomly scattered around the red line, it indicates that the model's assumptions are likely valid and the model fits the data well.
- **Patterns:** If there are patterns or trends in the residuals (e.g., a funnel shape or curve), it may suggest issues such as non-linearity, heteroscedasticity, or that a more complex model is needed.
- **High Dispersion of Points:** Indicates Poor Fit; Widespread dispersion around the zero line suggests the model is not capturing the relationship well, indicating potential issues with model fit or complexity.
- **Magnitude of Residuals:** Large residuals signal significant prediction errors, showing that the model's predictions are not close to the actual values.
- **Evaluation Metrics:** Use metrics like Mean Squared Error (MSE) or Mean Absolute Error (MAE). High values indicate poor model performance.
- **Compare Models:** Compare with other models to see if performance improves. More complex or different models might be needed.
- **Check Model Assumptions:** Ensure that model assumptions (e.g., linearity, homoscedasticity) are met. Deviations from these assumptions can impact residuals.

Actions:

- **Adjust or Change Model:** Consider modifying the model or trying alternative approaches if residual dispersion is high and cannot be justified.
- **Review Data:** Check data quality and inclusion of relevant variables.
- **Explore Transformations:** Apply variable transformations or feature engineering to improve model fit.

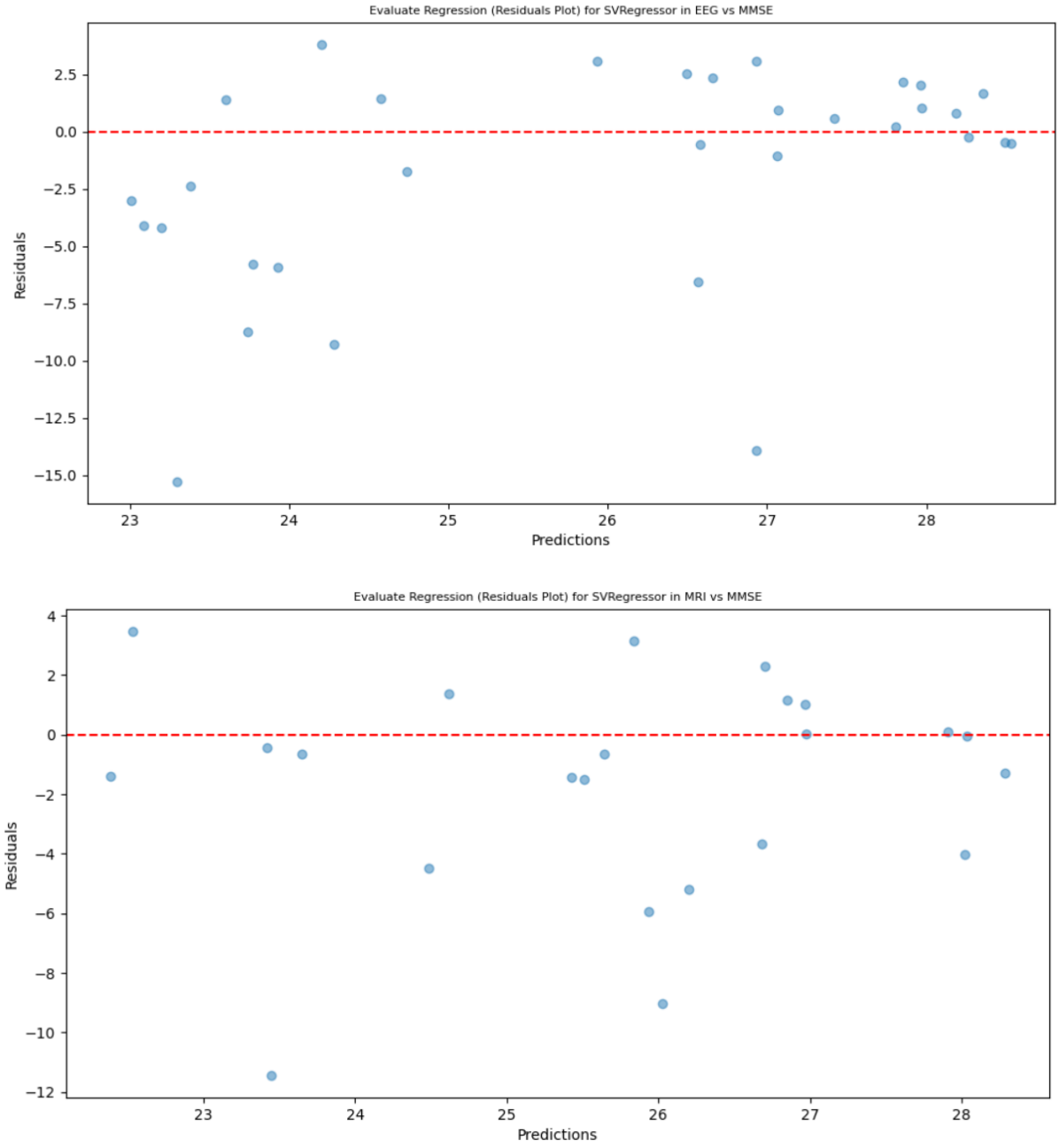
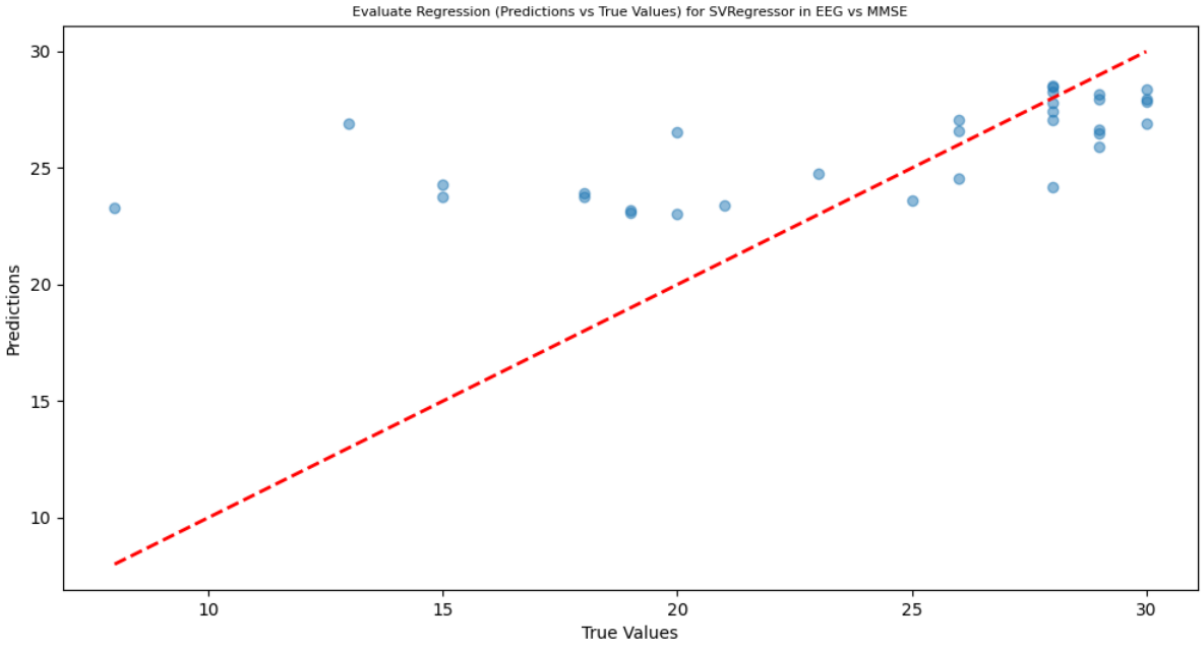


Figure 12 Residual Analysis of SVR Models Predicting MMSE Scores from EEG and MRI Data.

In Figure 12, the residual plots compare the performance of SVR models predicting MMSE scores using EEG and MRI data. The EEG-based model shows significant variability, especially for lower predictions, with residuals ranging from +2.5 to -15, indicating a tendency to overestimate in this range. The MRI-based model demonstrates slightly better performance, with residuals more concentrated around zero and less extreme variability. Overall, the MRI model appears to fit the data better, though both models exhibit some outliers, suggesting room for improvement in prediction accuracy.



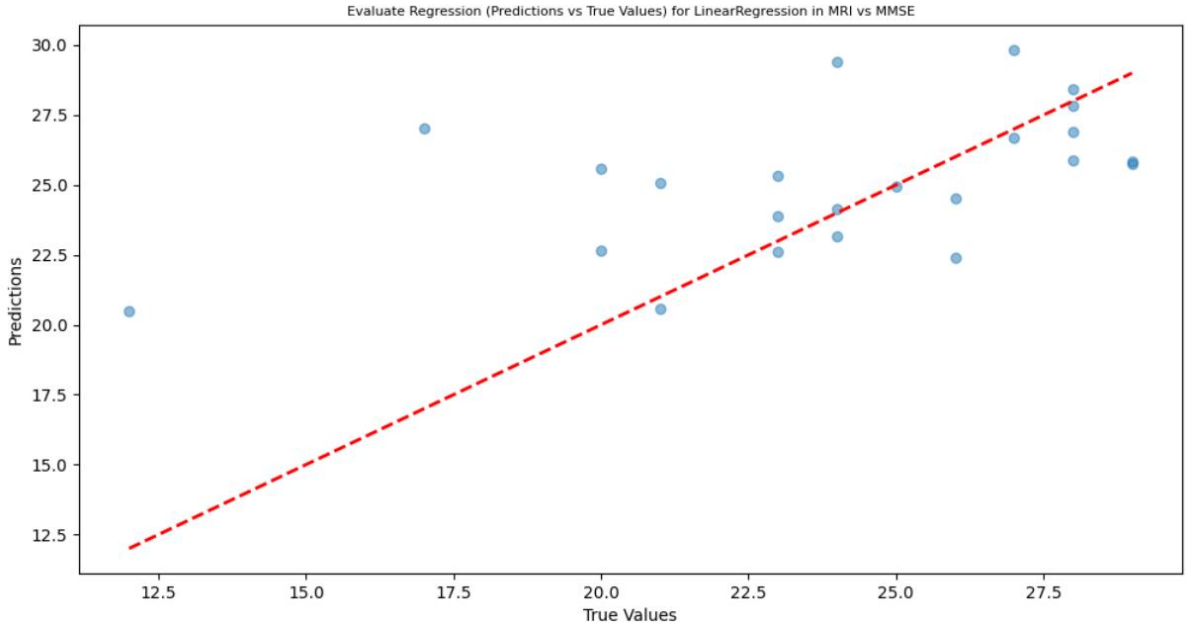


Figure 13 Evaluate Regression (Predictions vs True Values) MMSE Scores from EEG and MRI Data.

In Figure 13, the graphs compare predicted versus true values for two regression models: SVR using EEG data and Linear Regression using MRI data, both predicting MMSE scores. The SVR model shows significant scatter, especially for lower MMSE values, indicating inconsistent performance. In contrast, the Linear Regression model has predictions closer to the true values, suggesting a better fit, though some variability remains. Overall, Linear Regression appears to perform slightly better than SVR in these datasets.

3. Learning Curve

Interpretation:

Learning curves show the performance of a model on training and validation datasets as the number of training examples increases. They help diagnose issues related to overfitting and underfitting.

Components:

- **Training Score Curve:** Shows performance on the training data.
- **Cross-Validation Score Curve:** Shows performance on the validation data.

Example Analysis:

- **Constant Training Score:** If the training score curve is constant and high while the cross-validation score varies, it may indicate that the model is overfitting. The model performs well on the training set but does not generalize well to unseen data.
- **Decline in Training Score:** If the training score decreases dramatically towards the end of the training process, it may indicate that the model is underfitting or that the learning rate is too high, causing instability in learning.

Evaluating Model Performance:

- **Training Score:** A high and stable training score indicates the model is capturing the training data well.
- **Validation Score:** The validation score helps assess generalization. A large gap between training and validation scores suggests overfitting.
- **Learning Curve Trends:** To improve model performance:
    - If the training and validation scores converge to high values, the model is well-tuned.
    - If there is a large gap between training and validation scores, consider:
        - Increasing training data.
        - Adjusting model complexity (e.g., tuning hyperparameters).
        - Using regularization techniques to reduce overfitting.

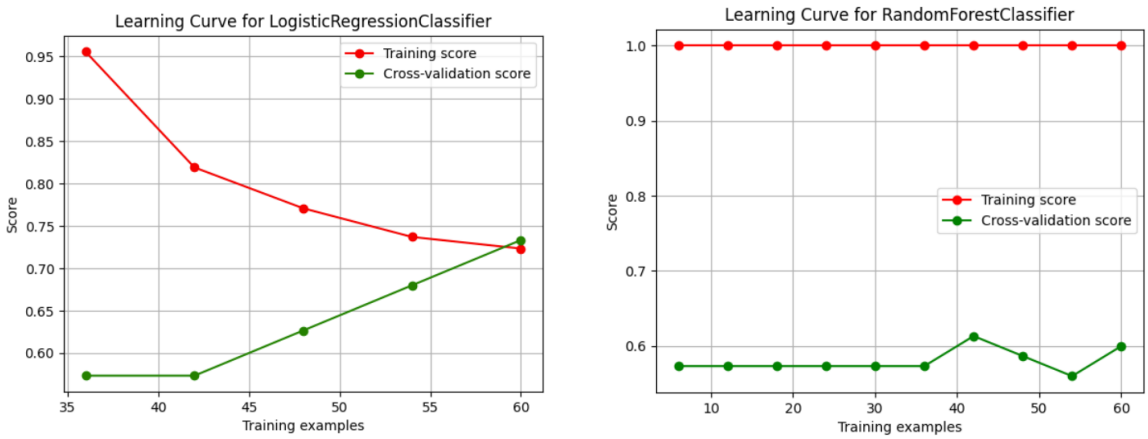


Figure 14 Learning Curve for Linear Regression and RandomForest Classifier. Axis x: Training Examples and Axis y: Accuracy Score.

Figure 14, the learning curves for the Logistic Regression Classifier and Random Forest Classifier show different behaviors. The Logistic Regression Classifier exhibits a positive trend, with training and cross-validation scores gradually converging as more training examples are added. This indicates that the model is learning effectively, with no significant overfitting and a good balance between bias and variance. In contrast, the Random Forest Classifier displays a consistently high training score but a lower and more variable cross-validation score, suggesting that the model is overfitting and struggling to generalize well to unseen data. Overall, the Logistic Regression shows better generalization compared to the Random Forest.

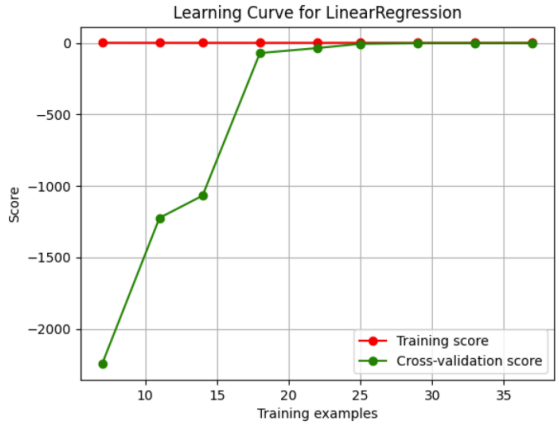


Figure 15 Learning Curve for LinearRegression. Axis x: Training Examples and Axis y: Negative Mean Squared Error (MSE).

Figure 15, Linear Regression shows high training scores close to zero, indicating low error during training, but the cross-validation scores start very low (high negative values) and improve slowly as more data is added. This suggests significant overfitting: the model performs well on training data but poorly on unseen data, indicating it does not generalize well. The metric on the Y-axis is not between 0 and 1 because it represents an error measure (Mean Squared Error), which can have a wide range of values, including negatives, indicating worse performance.

Summary

- **Confusion Matrix:** Helps understand classification performance, including accuracy, precision, and recall.
- **Residual Plot:** Assesses if the regression model fits the data well or if there are patterns suggesting model improvements.
- **Learning Curve:** Provides insights into model performance with different amounts of training data and helps diagnose overfitting or underfitting.

By analyzing these graphs and matrices, you can determine if your model needs adjustments or improvements and understand how well it performs with your given data.

More examples:

Table 1. Result of Classifier Model

Classifier Model	Accuracy (%)	Sensitivity (%)	Specificity (%)	Precision (%)	F1 Score (%)	AUC
LDA	78,12	78,12	68,42	82,22	78,23	0,8
LogisticRegression	87,5	87,5	89,47	87,5	87,5	0,87
RandomForest	87,5	87,5	89,47	87,5	87,5	0,87
KNN	84,38	84,38	84,21	84,7	84,45	0,84
SVC	87,5	87,5	89,47	87,5	87,5	0,87

Interpretation of Results 1:

1. LDA has lower accuracy and specificity compared to other models. While its precision is the highest, its ability to correctly identify negatives (specificity) is lower. This might suggest that LDA is more focused on correctly identifying positive cases but struggles with negative cases.
2. All these models show identical performance across all metrics. They have high accuracy, sensitivity, specificity, precision, and F1 Score, indicating a strong overall performance. These models balance well between detecting positive cases and correctly identifying negative cases.

Conclusion: For a robust performance, you might prefer using Logistic Regression, RandomForest, KNN, or SVC, given their higher overall metrics compared to LDA.

Table 2. Result Regression Model

Regression Model	Accuracy (%)	Sensitivity (%)	Specificity (%)	Precision (%)	F1 Score (%)	AUC
LDA	69,57	69,57	42,86	68,41	68,86	0,62
LogisticRegression	69,57	69,57	42,86	68,41	68,86	0,62
RandomForest	56,52	56,52	57,14	63,24	58,25	0,57

KNN	78,26	78,26	85,71	82,47	79,05	0,8
SVC	69,57	69,57	57,14	70,87	70,09	0,66

Interpretation of Results 2:

1. LDA has moderate accuracy and sensitivity but low specificity. This suggests that while LDA is decent at identifying positive cases, it struggles with correctly identifying negative cases. Precision is relatively high, indicating that when LDA predicts a positive result, it is fairly accurate.
2. Logistic Regression has similar accuracy and sensitivity to LDA but significantly lower specificity. This means it also has trouble with identifying negative cases. The precision is slightly lower than LDA, suggesting it may be less reliable when predicting positive cases compared to LDA.
3. RandomForest has the lowest accuracy and sensitivity among all models. Its specificity is moderate, but it still has poor overall performance, as indicated by the low accuracy and F1 Score. The model seems to struggle overall with both identifying positives and negatives correctly.
4. KNN performs similarly in terms of accuracy and sensitivity as LDA and Logistic Regression but with the highest specificity and precision. This suggests KNN is better at correctly identifying negative cases and when it predicts positive results, it is more accurate compared to the other models.
5. SVC has a lower accuracy and sensitivity compared to KNN but performs better in specificity and precision. This indicates that while SVC is better at correctly identifying negative cases and is relatively precise when predicting positives, it is not as good overall in detecting positives.

Conclusion: Based on these results, **KNN** appears to be the best model among those tested, especially if both precision and specificity are important.

Metrics Explained:

- **MSE (Mean Squared Error)**: Measures the average squared difference between predicted and actual values. Lower values indicate better model performance.
- **MAE (Mean Absolute Error)**: Measures the average absolute difference between predicted and actual values. Lower values indicate better model performance.
- **R^2 Score**: Indicates the proportion of variance in the dependent variable that is predictable from the independent variables. Higher values (close to 1) indicate a better fit of the model.

Table 3. Result of EEG vs MMSE

Model	MSE	MAE	R2 Score
LinearRegression	5,12	4,05	0,23
RandomForestRegressor	4,12	3,01	0,5

SVRegressor	5,06	3,46	0,25

Model Performance Interpretation:

1. Linear Regression has the highest MSE and MAE, indicating that its predictions are relatively far from the actual values. The R^2 score is low (0.23), meaning the model explains only 23% of the variance in the data. This suggests that Linear Regression may not be the best fit for this data and has a poor predictive performance.
2. RandomForestRegressor has lower MSE and MAE compared to Linear Regression, indicating better predictive accuracy. The R^2 score of 0.5 shows that the model explains 50% of the variance in the data, which is a significant improvement over Linear Regression. This model performs better overall and provides a reasonable fit to the data.
3. SVR has the lowest MSE and MAE among the models, suggesting the most accurate predictions. The R^2 score is slightly higher than that of the RandomForestRegressor (0.51), indicating that SVR explains 51% of the variance in the data. This model performs well and provides the best fit to the data among the three models.

Conclusion: **SVR** would be the preferred model based on these metrics, as it offers the best combination of low error and high explanatory power.

Table 4. Result of MRI vs MMSE

Model	MSE	MAE	R2 Score
LinearRegression	3,69	2,6	0,19
RandomForestRegressor	3,82	3,1	0,13
SVRegressor	3,98	2,77	0,06

Model Performance Interpretation:

1. Linear Regression has a moderate MSE and MAE, with an R^2 score of 0.19. This R^2 score indicates that only 19% of the variance in the data is explained by the model, suggesting a poor fit. The model's performance is not optimal, and predictions may be quite inaccurate.
2. RandomForestRegressor shows a slightly higher MSE and MAE compared to Linear Regression, and an even lower R^2 score of 0.13. This indicates that the model performs worse than Linear Regression in terms of explaining the variance in the data. The predictions are less accurate and the model seems to have a poor fit.
3. SVR has the highest MSE and MAE among the three models, with the lowest R^2 score of 0.06. This indicates that SVR performs the worst in terms of predictive accuracy and variance explanation. The model's predictions are the least accurate, and it explains only 6% of the variance in the data.

Conclusion: **Linear Regression** would be the preferred model based on these metrics, despite its low R^2 score. It has the lowest MSE and MAE compared to the other models. However, all

models show relatively poor performance, indicating that the current approach may not be well-suited for this dataset, or that additional feature engineering and tuning may be necessary to improve predictive accuracy.

Table 5. Result of Tele vs MMSE

Model	MSE	MAE	R2 Score
LinearRegression	2,36	1,57	0,53
RandomForestRegressor	2,92	1,72	0,28
SVRegressor	3,33	1,88	0,07

Model Performance Interpretation:

1. Linear Regression performs relatively well compared to the other models. It has the lowest MSE and MAE, indicating that on average, its predictions are closer to the actual values. The R^2 score of 0.53 suggests that the model explains 53% of the variance in the data, which is a moderate fit.
2. RandomForestRegressor has higher MSE and MAE than Linear Regression, indicating that its predictions are less accurate. The R^2 score of 0.28 suggests that it explains 28% of the variance in the data. This model performs worse than Linear Regression but better than SVR.
3. SVR shows the highest MSE and MAE, and the lowest R^2 score of -0.09. A negative R^2 score indicates that the model performs worse than a simple mean-based model, suggesting that SVR is not suitable for this dataset. Its predictions are the least accurate among the three models.

Conclusion: **Linear Regression** is the preferred model based on these results due to its better performance in terms of error metrics and variance explanation. The other models show poorer performance and may not be suitable for this particular problem without further tuning or feature engineering.

Activity for Integrating Concepts and Knowledge

Utilize the code from Workshop 4 to create a comparative analysis of results with the document **"TELEMAIA - Deliverable 3.1 SW AI-based for Patient Diagnosis and Assessment Support."** Present the information in organized tables to observe the data in the same format as it is presented in the document. Additionally, generate conclusions about the results obtained and suggest possible improvements in methodologies based on the concepts learned for future projects of this type.

Reference pages

https://machinelearningmastery.com/feature-selection-machine-learning-python/

https://www.asimovinstitute.org/neural-network-zoo/

https://rramosp.github.io/ai4eng.v1/intro.html

https://rramosp.github.io/2021.deeplearning/intro.html

https://www.pluralsight.com/guides/scikit-machine-learning

https://drive.google.com/drive/u/0/folders/1Y4MToxIocLYbLts1tFuvKVoHlin5WW5N

https://rramosp.github.io/2021.deeplearning/intro.html

https://developers.google.com/machine-learning/crash-course?hl=es-419

https://www.coursera.org/specializations/machine-learning-introduction?utm_medium=sem&utm_source=gg&utm_campaign=B2C_LATAM_machine-learning-introduction_deeplearning-ai_FTCOF_specializations_countrygroup-1&campaignid=20849949060&adgroupid=162342411808&device=c&keyword=ai%20programs&matchtype=b&network=g&devicemodel=&adposition=&creativeid=684376971992&hide_mobile_promo&gad_source=1&gclid=CjwKCAjwlbu2BhA3EiwA3yXyu06mPEJJxX1nkGguM_Sst3JudiIaSffCv5V3ctUdyYuhQGzcEpo9URoCMNwQAvD_BwE