# Group Assignment – Machine Learning Foundations

## 5 stages of Alzheimer disease

**Luis Gomez-Acebo, Leonardo Camilleri, Noah Valderrama, Jose Miguel Serrano, Peter Norbert Karacs**

### Introduction

Our objective of this project has been to develop a machine learning model to predict Alzheimer's disease based features extracted from the given dataset of Alzheimer's-Disease-5-Class-Dataset-ADNI in Kaggle. This dataset allows the creation of prediction models to categorize the phases of the disease by providing a variety of variables from MRI imaging and demographic data.

The 6400 examples in the collection contain elements including cognitive scores, MRI imaging measures, and demographic data divided into five phases of Alzheimer's disease. This makes it possible to conduct a thorough examination and create prediction models.

### Data Exploration and Preprocessing

To determine how the characteristics were distributed and related to one another, we started with an exploratory data analysis (EDA). The distribution of numerical data was examined, and any outliers were found, using visualizations like scatter plots and picture sampling. Plotting a correlation matrix allowed researchers to investigate the connections between various characteristics, identifying important predictors including age, cognitive test scores, and certain MRI measures.

After processing the photos, we resized them to a consistent 128x128 pixel size and converted them to grayscale. To ensure that the input data for the neural network was standardized, this preprocessing step was crucial. Next, the pixel values were normalized to fall between 0 and 1. An 80-20 split was then used to divide the normalized pictures into training and testing sets. By doing this, it is made possible for the model to be trained on a sizable amount of the data and then tested for generalizability on a different subset.

For feature engineering, we used the function process_images to efficiently process only image files within a specified directory, ensuring that it does not mistakenly try to open directories as files. It checks each item in the directory to confirm whether it is a file, then opens it, converts it to grayscale, resizes, and appends its array to a list. This function is robust against errors, catching exceptions for files that cannot be processed, thus ensuring smooth execution without

interruption. This approach is crucial for preparing a clean and consistent dataset for machine learning models.

## Model Development

In the model development phase, we focused on creating a convolutional neural network (CNN) model using TensorFlow and Keras.

**Convolutional Neural Network (CNN) Model**
The CNN model was designed with multiple convolutional and pooling layers to effectively extract features from the input images, followed by dense layers for classification.

**Model Architecture**

- Convolutional Layers: Three convolutional layers with ReLU activation to learn spatial hierarchies of features.
- Pooling Layers: MaxPooling layers to reduce the spatial dimensions and computational load while retaining important features.
- Dense Layers: Two dense layers, with the first dense layer using ReLU activation and dropout regularization to prevent overfitting, and the output layer using softmax activation for multi-class classification.

## Model Evaluation and Selection

Our code implements a 5-fold cross-validation for a convolutional neural network (CNN) on an image dataset, training and evaluating the model on different subsets to assess its generalizability and robustness. It uses model checkpointing to save training states, allowing recovery and analysis at different stages. This approach helps optimize model performance and ensures stability across varied data samples, crucial for reliable medical image analysis.(See appendix)

We utilized Keras Tuner to optimize hyperparameters for a convolutional neural network. It defines a model-building function build_model that sets up a neural network with adjustable hyperparameters for convolutional filters, dense units, and learning rate. RandomSearch tuner explores different configurations over 10 trials to maximize validation accuracy. After finding the best hyperparameters, it prints them and re-trains the model with these optimized settings to ensure the best performance on the validation data. The process aims to improve model accuracy by tuning the architecture and training configuration based on the dataset provided.

## Group Reflections

Through effective division of labour and regular communication, we tackled challenges in data preprocessing and model optimization. Even though we had some difficulties in aligning schedules and tuning hyperparameters, we successfully implemented a robust CNN model. We think that the key takeaways include the importance of detailed planning, proactive problem-solving, and a good organisation between all team members.

## **Appendix**

Dataset used:
https://www.kaggle.com/datasets/madhucharan/alzheimersdisease5classdatasetadni/data#

Evaluation Plots