

Proyecto: Diagnóstico Temprano de Alzheimer mediante Machine Learning y Algoritmos Genéticos

Problema: Detección Temprana del Alzheimer

El Alzheimer es una enfermedad neurodegenerativa progresiva que afecta principalmente a la memoria y otras funciones cognitivas. La detección temprana es crucial para ralentizar su progresión mediante intervenciones terapéuticas y cambios de estilo de vida.

Objetivo General


Desarrollar un sistema de predicción temprana del Alzheimer combinando técnicas de Machine Learning (ML) y Deep Learning con optimización mediante algoritmos genéticos (AG).

Datos Usados

Datasets principales

- [OASIS-3](#): MRI longitudinal de adultos mayores sanos y con demencia (NIH + datos clínicos).
- [ADNI \(Alzheimer's Disease Neuroimaging Initiative\)](#): Incluye MRI, PET, pruebas cognitivas y genéticas.
- [Student Dropout and Academic Success](#): No relacionado directamente, pero puede servir para pruebas previas en modelos de clasificación.

Modelos actuales y efectividad

Modelo	AUC / Accuracy aprox.
Random Forest (clínico)	0.75 - 0.85
CNN sobre MRI (desde cero)	0.80 - 0.88
ResNet50 con fine-tuning	0.90 - 0.95 
Multimodal (MRI + tests)	> 0.95 (estado del arte)

Estrategias de mejora posibles

Fine-tuning sobre modelos preentrenados

- Aplicación de redes como **ResNet50**, **EfficientNet** sobre MRI/PET
- Ajuste fino: descongelar capas superiores + tasa de aprendizaje baja
- Regularización (dropout), data augmentation, normalización

🤖 Tuneo de hiperparámetros

- `learning_rate`, `dropout_rate`, `batch_size`, `n° capas ocultas`
- Herramientas: `KerasTuner`, `Optuna`, `RandomSearchCV`, `GridSearchCV`

😊 Aplicación de algoritmos genéticos

😞 Selección de características (Feature Selection)

- Individuo = vector binario de selección
- Fitness = AUC del modelo entrenado

😞 Optimización de hiperparámetros

- Representar combinaciones de parámetros como cromosomas
- Ej: `learning_rate`, `num_layers`, `dropout`

😞 Diseño de arquitecturas

- Uso de Neuroevolution (NEAT o similar) para buscar mejores redes neuronales

😞 Optimización del preprocesamiento

- Selección automática de cortes, escalado, aumentaciones

■ Ejemplo de aplicación práctica

Título

Optimización del diagnóstico temprano de Alzheimer mediante selección evolutiva de variables clínicas y estructurales

Flujo del proyecto

1. **Carga y limpieza de datos** (OASIS-3 / ADNI)
2. **Extracción de variables relevantes** (MRI + datos cognitivos)
3. **Aplicación de algoritmo genético para selección de features**
4. **Entrenamiento de modelos** (Random Forest, XGBoost, CNN)
5. **Evaluación con AUC, F1, sensibilidad, especificidad**
6. **Visualización de resultados e interpretabilidad** (SHAP, Grad-CAM)

■ Librerías Python recomendadas

- `scikit-learn`, `xgboost`, `keras`, `tensorflow`, `pytorch`
- `deap`, `tpot`, `pygad` → para algoritmos genéticos
- `nibabel`, `SimpleITK` → para procesamiento de MRI
- `matplotlib`, `seaborn`, `shap`, `lime` → para interpretabilidad

Planificación (si trabajas 5 h/día)

Semana	Objetivo
1	Revisión bibliográfica, descarga dataset, preprocesamiento inicial
2	EDA (análisis exploratorio), limpieza, preparación de variables
3	Implementación de AG para selección de features (DEAP o PyGAD)
4	Entrenamiento de modelos base + evaluación inicial
5	Fine-tuning con CNN preentrenada + comparación de resultados
6	Interpretabilidad (SHAP, Grad-CAM) + visualización de resultados
7	Documentación, conclusiones, entrega y presentación final

Duración total estimada: **7 semanas** (35 días de trabajo efectivo, 5h/día)

✅ Este enfoque te permite combinar la parte médica (problema real), técnica (ML + Deep Learning) y evolutiva (AG) en un proyecto completo, ambicioso y publicable.