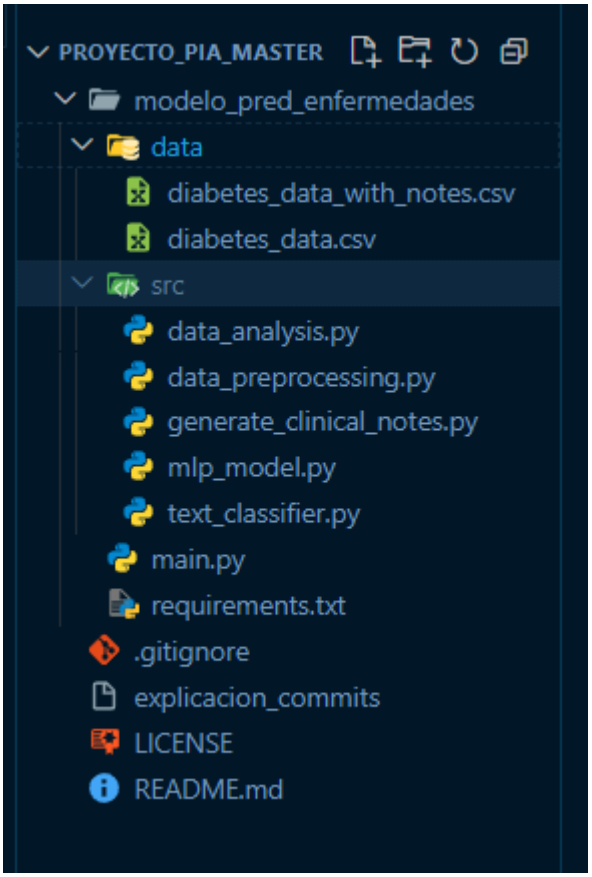
Guía usuario Proyecto PIA Nicolás Gómez García

En este proyecto tendremos la siguiente estructura:



Tendremos los archivos python para el análisis de predicción de enfermedades junto el ML y el NLP.
data_preprocessing.py se encargará del preprocesamiento del dataset junto a la limpieza de este.
data_analysis.py se encargará de un análisis breve de los datos del dataset y sus valores.
generate_clinical_notes.py se encargará de añadir las consultas médicas en texto para poder practicar el NLP ya que solo tenía datos numéricos, por consecuencia la parte de NLP predicción es simulada.
mlp_model.py se encargará del análisis y predicción de datos del dataset diabetes.csv con los datos reales antes del análisis simulado del NLP para ver la diferencia de ambos.
text_classifier.py se encargará del análisis y predicción de datos simulados junto a las consultas clínicas del dataset diabetes_data_with_notes.csv.

Ahora procederemos a ver la instalación del entorno virtual para poder usar el proyecto ya que no se puede subir a git y tenerlo instantáneamente por su peso.

Paso 1: Abrir la terminal (PowerShell, CMD o terminal de tu sistema operativo).

Paso 2: Navegar a la carpeta src del proyecto:
cd ruta/a/tu/proyecto/src

Paso 3: Crear el entorno virtual:
python -m venv venv

Esto creará una carpeta llamada venv/ dentro de src que contendrá el entorno virtual.

2. Activar el entorno virtual

En Windows (PowerShell):
.\venv\Scripts\Activate

Una vez activado, verás que el prompt de la terminal cambia a algo como:

(venv) PS C:\Users\usuario\ruta\src>

3. Instalar las dependencias necesarias

Ejecuta los siguientes comandos estando dentro del entorno virtual:

pip install pandas scikit-learn matplotlib seaborn tensorflow keras nltk scipy

Estas librerías cubren todos los módulos utilizados en los scripts del proyecto:

Bibliotecas utilizadas:

- os: manejo de rutas y archivos
- pandas: manipulación de datos
- matplotlib: visualización de datos
- seaborn: gráficos estadísticos
- scikit-learn:
    - train_test_split: división de datos
    - classification_report, confusion_matrix, roc_auc_score, roc_curve, auc
    - RandomForestClassifier: modelo de ML tradicional
    - TfidfVectorizer, StandardScaler, LabelEncoder
- tensorflow.keras:
    - Sequential, Dense, Dropout, EarlyStopping: red neuronal MLP
- nltk: procesamiento de lenguaje natural
- scipy.sparse.hstack: combinación de matrices de texto y numéricas

4. Crear un archivo requirements.txt (opcional pero recomendado)

Para guardar todas las librerías instaladas:

pip freeze > requirements.txt

Para que otros puedan instalar todas las dependencias en un nuevo entorno:

pip install -r requirements.txt

5. Desactivar el entorno virtual

Cuando hayas terminado de trabajar:

deactivate