|-- nombre\_proyecto\_final\_ML

|-- data

| |-- raw

| |-- dataset.csv

| |-- ...

| |-- processed

| |-- train

| |-- test

|

|-- notebooks

| |-- 01\_Fuentes.ipynb

| |-- 02\_LimpiezaEDA.ipynb

| |-- 03\_Entrenamiento\_Evaluacion.ipynb

| |-- ...

|

|-- src

| |-- data\_processing.py

| |-- training.py

| |-- evaluation.py

| |-- ...

|

|-- models

| |-- trained\_model.pkl

| |-- model\_config.yaml

| |-- ...

|

|-- app\_streamlit

| |-- app.py

| |-- requirements.txt

| |-- ...

|

|-- docs

| |-- negocio.ppt

| |-- ds.ppt

| |-- memoria.md

| |-- ...

|

|

|-- README.md

A continuación se detallan las carpetas y los requisitos de cada una:

1. **data**: se almacenarán los datos utilizados en el proyecto. Se deben crear las siguientes subcarpetas:
   * raw: Contiene los datos en su formato original, sin procesar.
   * processed: Almacena los datos procesados después de realizar las transformaciones necesarias antes de utilizarlos para el modelo.
   * train: Contiene los datos de entrenamiento utilizados para entrenar el modelo a partir de los datos procesados
   * test: Almacena los datos de prueba utilizados para evaluar el modelo a partir de los datos procesados
2. **notebooks**: se encuentran los archivos Jupyter Notebook que contienen el desarrollo del proyecto. Se deben nombrar y numerar adecuadamente según el orden de ejecución.
   * 01\_Fuentes.ipynb: adquisición de datos y unión de las diferentes fuentes.
   * 02\_LimpiezaEDA.ipynb: transformaciones y limpiezas, incluyendo el feature engineering, así como visualizaciones dentro de un análisis exploratiorio.
   * 03\_Entrenamiento\_Evaluacion.ipynb: entrenamiento de modelos (mínimo 5 modelos supervisados diferentes y al menos 1 no supervisado) junto con su hiperparametrización, así como evaluación de los modelos (métricas de evaluación, interpretación de variables,...).
3. **src**: contiene los archivos fuente de Python que implementan las funcionalidades clave del proyecto. Los requisitos de los archivos son los siguientes:
   * data\_processing.py: código para procesar los datos de la carpeta data/raw y guardar los datos procesados en la carpeta data/processed.
   * training.py: código para entrenar y guardar el modelo entrenado con el input de los datos de la carpeta data/processed y guardar los datasets de data/train y data/test utilizados en el entrenamiento.
   * evaluation.py: código para evaluar el modelo utilizando los datos de prueba de la carpeta data/test y generar métricas de evaluación.
4. **models**: se almacenarán los archivos relacionados con el modelo entrenado. Los requisitos son los siguientes:
   * trained\_model\_n.pkl: modelos entrenados y guardados en formato pickle, siendo n un identificador para cada modelo.
   * final\_model.pkl: modelo final guardado en formato pickle.
   * model\_config.yaml: archivo con la configuración del modelo final (parámetros)
5. **app\_streamlit**: contendrá los archivos necesarios para el despliegue del modelo en Streamlit u otra plataforma similar. Los requisitos son los siguientes:
   * app.py: código para la aplicación web que utiliza el modelo entrenado final(Streamlit).
   * requirements.txt: especifica las dependencias del proyecto para poder ejecutar la aplicación.
6. **docs**: contendrá la documentación adicional relacionada con el proyecto, como las dos presentaciones u otros documentos relevantes como la memoria.
7. **README**: portada de tu proyecto.