

Guía de estructura de carpetas y archivos del sistema Guardián de Fraudes

Este documento describe la organización del proyecto Guardián de Fraudes, con el objetivo de facilitar la navegación, revisión técnica y posible extensión del sistema por parte de terceros. A continuación, se presentan las carpetas principales, su propósito, y los archivos clave.

1. Estructura general del proyecto

- └─ main.py # Archivo principal para levantar el dashboard Streamlit
- └─ pagina_resumen.py # Página de vista general del sistema
- └─ pagina_eje1.py # Página de Detección de Anomalías (Eje 1)
- └─ pagina_cliente.py # Página de Predicción por Cliente (Eje 2)
- └─ evaluar_eje1_integrado.py # Backend para evaluar modelos de Eje 1
- └─ evaluar_eje2_integrado.py # Backend para evaluar modelos de Eje 2
- └─ df_modelo_procesado.pkl # Dataset procesado con variables físicas derivadas
- └─ modelos/
 - | └─ series/
 - | └─ prophet/ # Modelos Prophet entrenados (.pkl) por cliente y horizonte
 - | └─ neuralprophet/ # Modelos NeuralProphet entrenados (.pt)
- └─ resultados/
 - | └─ resultados_eje1_consolidado.pkl # Resultados finales con votación por modelo
 - | └─ resultados_prophet_consolidado.pkl # Predicciones por cliente - Prophet
 - | └─ resultados_neuralprophet_consolidado.pkl # Predicciones por cliente - NeuralProphet

└─ scaler_autoencoder_general.pkl # Scaler usado en Autoencoder general

└─ autoencoder_general.h5 # Modelo Autoencoder entrenado con todos los datos

└─ entrenar_prophet_todos.py # Script de entrenamiento masivo de Prophet

└─ entrenar_neuralprophet_todos.py # Script de entrenamiento masivo de NeuralProphet

└─ probar_autoencoder.py # Script de prueba del modelo Autoencoder

└─ consolidar_resultados_eje1.py # Junta los resultados de todos los modelos del Eje 1

└─ fondo1.png / fondo2.png # Imágenes de fondo utilizadas en Streamlit

└─ requirements.txt # Lista de dependencias del entorno

└─ data/

└─ archivo_cliente.csv # Archivo cargado manualmente para pruebas o evaluación

2. Carpetas principales

- **modelos/series/**: almacena los modelos entrenados por cliente para predicción. Organizado por algoritmo (Prophet / NeuralProphet).
- **resultados/**: contiene todos los archivos .pkl con resultados de evaluación de modelos, predicciones y etiquetas.
- **scripts**: archivos .py usados para entrenamiento, validación o pruebas, pueden ejecutarse en consola o en entorno Colab.
- **data/**: carpeta auxiliar para pruebas manuales o carga de archivos tipo .csv.

3. Consideraciones adicionales

- Todos los modelos se entrenan una sola vez y luego se cargan al iniciar el sistema para evitar retrasos.
- Los archivos .pkl y .pt se nombran según el cliente, variable objetivo y horizonte (por ejemplo, prophet_CLIENTE1_pv_t_h24.pkl).

- La base de datos principal (df_modelo_procesado.pkl) ya contiene las variables derivadas necesarias (pv_t, pv_zt, etc.).
- Las funciones que evalúan modelos (por ejemplo, evaluar_eje1_integrado.py) están diseñadas para recibir nuevos registros y retornar decisiones inmediatas.

Este esquema permite mantener claridad en el mantenimiento