**🔍 Problema a Resolver**

Queremos construir un modelo de *Machine Learning* que pueda predecir si un paciente tiene una enfermedad cardíaca (variable objetivo target, donde 1 = enfermedad y 0 = saludable) basándose en sus características médicas.

**📊 Exploración de Datos**

El dataset contiene varias variables como:

* **Edad** (age)
* **Sexo** (sex)
* **Presión arterial en reposo** (trestbps)
* **Colesterol** (chol)
* **Frecuencia cardíaca máxima alcanzada** (thalach)
* **Angina inducida por ejercicio** (exang)
* **Depresión del ST inducida por el ejercicio** (oldpeak)
* **Tipo de dolor en el pecho** (cp - diferentes tipos de dolor)
* **Resultados de electrocardiograma en reposo** (restecg)

**📌 Pasos para Resolverlo**

**1️⃣ Carga y Preprocesamiento de Datos**

✔️ Cargar los datos con pandas.  
✔️ Manejar valores faltantes y convertir variables categóricas en numéricas.  
✔️ Normalizar variables numéricas para mejorar el rendimiento de algunos modelos.

**2️⃣ Análisis Exploratorio (EDA)**

✔️ Visualizar la distribución de pacientes con y sin enfermedad.  
✔️ Analizar correlaciones entre variables (ej. la frecuencia cardíaca thalach suele estar relacionada con la enfermedad).  
✔️ Identificar la importancia de cada característica con Seaborn o Matplotlib.

**3️⃣ Entrenamiento del Modelo**

Probamos varios modelos para ver cuál da mejor resultado:  
✔️ **Regresión Logística** (modelo base).  
✔️ **Random Forest** (buen equilibrio entre interpretabilidad y rendimiento).  
✔️ **XGBoost** (potente para clasificación).  
✔️ **Redes Neuronales** (si quieres probar Deep Learning).

**4️⃣ Evaluación del Modelo**

✔️ Usar métricas como **AUC-ROC**, **F1-score** y **matriz de confusión**.  
✔️ Comparar distintos modelos para ver cuál predice mejor.

**5️⃣ Interpretación y Explicabilidad**

✔️ Usar **SHAP** o **LIME** para entender qué características tienen más impacto en la predicción.

**🛠 Herramientas Recomendadas**

📌 **Librerías**: pandas, numpy, matplotlib, seaborn, scikit-learn, XGBoost, SHAP.  
📌 **Notebook en Kaggle o Google Colab** para pruebas rápidas sin necesidad de configurar un entorno local.

**🚀 Retos Adicionales**

Si quieres llevar este proyecto más lejos, puedes:

* Comparar los resultados con técnicas de *Feature Engineering* avanzadas.
* Probar con datos sintéticos generados con SMOTE si el dataset está desbalanceado.
* Implementar una API con Flask o FastAPI para desplegar el modelo.

¿Quieres que te ayude con algún código de ejemplo para empezar? 😃