UNIVERSIDAD NACIONAL DEL ALTIPLANO

FACULTAD DE INGENIERIA MECANICA ELECTRICA, ELECTRONICA Y SISTEMAS

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS

ACTIVIDAD 3

PRESENTADA POR:

JHON ALEX CENTENO CCORIMANYA

OLIVER FRANK CHOQUE CHURA

CLEVER EDISON PAREDES VELASQUEZ

JORGE GUILLERMO OLARTE QUISPE

JACK BENJAMIN CANCAPA PACCO

DOCENTE:

ING. MAYENKA FERNANDEZ CHAMBI

CURSO :

APRENDIZAJE DE MÁQUINA

PUNO - PERÚ

2025

Proyecto de predicción: Predicción de enfermedades cardiovasculares

https://github.com/ArtStyle19/prediccion-de-enfermedades-cardiovasculares-ML

Fase I. Business Understanding (Comprensión del negocio)

Objetivo general: Predecir si un paciente tiene riesgo de enfermedad cardiovascular en base a datos clínicos, para apoyar la prevención temprana.

Tareas:

1. Determinar los objetivos del negocio

- **Se identifican las necesidades principales:** Reducir la mortalidad y mejorar la prevención de enfermedades cardiovasculares. Desde la perspectiva del hospital/universidad/centro médico, el objetivo es contar con una herramienta que ayude a los médicos a detectar pacientes en riesgo de manera temprana.

  Ejemplo: "Queremos predecir si un paciente presenta riesgo de enfermedad cardiovascular con al menos un 80% de precisión".

2. Evaluar la situación

- **Se analizan los recursos disponibles:**
  - Dataset clínico (Heart Disease Dataset - Kagglel)
    - edad
    - sexo
    - tipo de dolor en el pecho (4 valores)
    - presión arterial en reposo
    - colesterol sérico en mg/dl

        - nivel de azúcar en sangre en ayunas > 120 mg/dl
        - Resultados electrocardiográficos en reposo (valores 0,1,2)
        - frecuencia cardíaca máxima alcanzada
    - Herramientas: Python, Google Colab, librerías de ML (Scikit-learn, Pandas, Matplotlib).
    - Equipo de trabajo: Compañeros de curso.

Se identifican restricciones: Protección de datos sensibles (confidencialidad), presupuesto, tiempo de desarrollo.

Se evalúan riesgos: Baja calidad de los datos, posibles sesgos, modelos que no generalicen bien.

También se hace un análisis costo-beneficio: ¿vale la pena invertir en el modelo comparado con los beneficios de prevención?

3. Determinar los objetivos de minería de datos

Se transforman los objetivos médicos en objetivos técnicos de minería de datos:

- Predecir la variable objetivo “presencia o ausencia de enfermedad cardiovascular”
    - HeartDisease = 1 (paciente con enfermedad cardiovascular)
    - HeartDisease = 0 (paciente sano)
- Entrenar modelos que clasifiquen pacientes con base en variables como edad, colesterol, presión sanguínea, etc.

Se definen los criterios de éxito técnico:

- Métricas mínimas aceptables: precisión ≥ 80%, recall alto (para no dejar sin detectar a pacientes enfermos).
- Reducción de falsos negativos (pacientes en riesgo no detectados).
    - Accuracy
    - Precision
    - Recall
    - F1-score

4. Plan de proyecto

Se establecen los pasos para cumplir con los objetivos:

1. Adquirir y limpiar el dataset.
2. Explorar y analizar los datos clínicos.
3. Preparar los datos (transformaciones y selección de variables).
4. Aplicar y comparar modelos (Random Forest).
5. Evaluar los resultados según métricas médicas y técnicas.
6. Desplegar el modelo en un entorno práctico (ej. dashboard para médicos).

Se definen las herramientas y tecnologías: Python, Google Colab, librerías de ML.

Se asignan roles: un equipo se encarga del preprocesamiento, otro del modelado, otro de la validación y comunicación de resultados.

Fase II. Data Understanding (Comprensión de los datos)

Se centra en identificar, recopilar y analizar los conjuntos de datos que pueden ayudarle a alcanzar los objetivos del proyecto.

1. Recolectar datos iniciales:

Dataset *Heart Disease UCI* con atributos clínicos.

2. Describir datos:

El dataset descargado de Kaggle denominado **Heart Disease Dataset** contiene información clínica de pacientes con el objetivo de predecir la presencia o ausencia de enfermedad cardíaca.

Características principales:

- **Número de registros:** 1,025 pacientes.
- **Número de columnas:** 14 variables (13 predictoras + 1 variable objetivo).
- **Formato de los datos:** numérico (enteros y flotantes).
- **Valores faltantes:** no se encontraron valores nulos en ninguna de las variables.

Variables incluidas:

1. **age**: Edad del paciente (29 – 77 años).
2. **sex**: Sexo (0 = mujer, 1 = hombre).
3. **cp**: Tipo de dolor en el pecho (4 categorías: 0–3).
4. **trestbps**: Presión arterial en reposo (94 – 200 mm Hg).
5. **chol**: Nivel de colesterol sérico (126 – 564 mg/dl).
6. **fbs**: Azúcar en sangre en ayunas >120 mg/dl (1 = sí, 0 = no).

7. **restecg**: Resultados del electrocardiograma en reposo (0–2).
8. **thalach**: Frecuencia cardíaca máxima alcanzada (71 – 202 latidos/min).
9. **exang**: Angina inducida por ejercicio (1 = sí, 0 = no).
10. **oldpeak**: Depresión del ST inducida por ejercicio (0 – 6.2).
11. **slope**: Pendiente del segmento ST (0–2).
12. **ca**: Número de vasos principales coloreados por fluoroscopía (0–4).
13. **thal**: Resultados de la prueba Thal (0–3).
14. **target**: Variable objetivo (0 = ausencia de enfermedad, 1 = presencia de enfermedad).

Estadísticas descriptivas:

- **Edad media:** 54.4 años (σ = 9.0).
- **Presión arterial media:** 131 mm Hg (σ = 17.5).
- **Colesterol promedio:** 246 mg/dl (σ = 51.6), con valores extremos hasta 564 mg/dl.
- **Frecuencia cardíaca máxima:** promedio de 149 lpm.
- **Distribución del sexo:** 69.5% hombres, 30.5% mujeres.
- **Variable objetivo (target):** equilibrada (≈51% con enfermedad, ≈49% sin enfermedad).

3. Explorar datos

Se profundizó en la información del dataset mediante consultas y visualizaciones:

- **Distribución de la edad:** los pacientes tienen entre 29 y 77 años, con mayor concentración en el rango de 50 a 60 años.

- **Colesterol vs. enfermedad:** se observa que niveles elevados de colesterol (≥ 250 mg/dl) tienden a estar más asociados con la presencia de enfermedad cardíaca, aunque no es un predictor exclusivo.
- **Frecuencia cardíaca máxima (thalach):** promedio de 149 lpm, mostrando que pacientes con valores más bajos presentan con mayor frecuencia diagnóstico positivo.
- **Matriz de correlación:** revela relaciones moderadas entre algunas variables, por ejemplo, entre edad y frecuencia cardíaca (correlación negativa), así como entre exang (angina inducida por ejercicio) y la variable objetivo.

4. Verificar la calidad de los datos

Se evaluó la limpieza y consistencia de la información:

- **Valores faltantes:** no existen registros nulos en ninguna variable.
- **Outliers:** se detectaron **16 valores atípicos en colesterol**, alcanzando niveles poco realistas (> 500 mg/dl).
- **Registros inconsistentes:** no se identificaron inconsistencias en variables categóricas (ejemplo: sex solo contiene valores válidos: 0 = mujer, 1 = hombre).
- **Formato uniforme:** todas las variables están correctamente tipificadas (enteros o flotantes según corresponda).

Fase III: Preparación de los Datos

Objetivo:

Construir el conjunto de datos final que será utilizado en la fase de modelado, asegurando que la información sea consistente, limpia y adecuada para los algoritmos de aprendizaje

automático. Esta fase suele ser la más extensa, pues gran parte del éxito del modelo depende de la calidad de los datos preparados.

Tareas principales:

1. **Selección de datos**
    - Se identificaron y seleccionaron las variables clínicas más relevantes para el análisis, tales como edad, sexo, presión arterial en reposo, colesterol sérico, tipo de dolor en el pecho, frecuencia cardíaca máxima, entre otros.
    - Se excluyeron datos irrelevantes como identificadores personales (ejemplo: nombres, números de seguro social) que no aportan valor predictivo.

2. **Limpieza de datos**
    - Se trataron los valores faltantes mediante imputación (usando la mediana en variables numéricas y la moda en categóricas).
    - Se eliminaron duplicados y se corrigieron posibles inconsistencias en los registros.

3. **Construcción de datos**
    - Se generaron nuevas variables derivadas de las originales, por ejemplo, la codificación de categorías mediante *One-Hot Encoding* (ej. tipo de dolor en el pecho se transformó en variables binarias).
    - Este paso permite que los algoritmos procesen correctamente información categórica y se amplíe el poder predictivo del dataset.

4. **Integración de datos**

- Aunque en este caso la fuente es única (dataset de Kaggle sobre enfermedades cardíacas), se contempló la posibilidad de integrar registros de distintas bases en caso de ser necesario (ejemplo: diferentes hospitales).

5. **Formateo de datos**
    - Se normalizaron las variables numéricas para un rango comparable, evitando que escalas muy grandes dominen el modelo.
    - Se aplicó la codificación de variables categóricas para convertirlas en un formato numérico entendible por los algoritmos de aprendizaje.

Resultado:

El dataset procesado (X_prepared) está ahora limpio, transformado y estandarizado, listo para ser utilizado en la fase IV (Modelado). Este conjunto de datos asegura una mejor calidad en la predicción y reduce los riesgos de errores durante el entrenamiento del modelo.

Fase IV. Modeling (Modelado)

Objetivo: Entrenar y evaluar distintos algoritmos predictivos para clasificar pacientes con o sin enfermedad cardíaca.

1. Selección de técnicas de modelado

Se consideraron cuatro algoritmos:

- **Regresión Logística**
    - Modelo sencillo y rápido.
    - Fácil de interpretar (permite entender qué variables influyen).
    - Limitado cuando la relación entre variables no es lineal.

- **Árboles de Decisión**
    - Fáciles de comprender, ya que funcionan como un árbol de preguntas (“¿la presión es alta? → sí/no”).
    - Riesgo de sobreajuste: funcionan bien en entrenamiento pero fallan con datos nuevos.
- **Support Vector Machine (SVM)**
    - Muy potente para separar clases.
    - Captura relaciones complejas.
    - Inconvenientes: difícil de interpretar y costoso en datasets grandes.
- **Random Forest (seleccionado como modelo principal)**
    - Combina muchos árboles → reduce sobreajuste.
    - Detecta relaciones complejas y no lineales.
    - Funciona bien con datos numéricos y categóricos (tras preprocesamiento).
    - Permite obtener la importancia de variables, lo cual es muy valioso en medicina para entender factores de riesgo.

2. Diseño de pruebas

Antes de entrenar, se estableció cómo evaluar los modelos:

- **División de datos**: 80% para entrenamiento, 20% para prueba, usando stratify para mantener balance entre enfermos y sanos.
- **Validación cruzada (opcional)**: *k-fold* (k=5 o 10) para verificar la estabilidad del modelo.
- **Criterio de éxito**: Priorizar el Recall (Sensibilidad), ya que en medicina es más grave no detectar a un paciente enfermo (falso negativo) que clasificar erróneamente a un sano como enfermo (falso positivo).

3. Construcción de modelos

Se aplicó el preprocesamiento previamente definido (imputación de valores faltantes, normalización de variables numéricas y codificación one-hot para variables categóricas).

Posteriormente, se entrenaron distintos algoritmos en varias iteraciones:

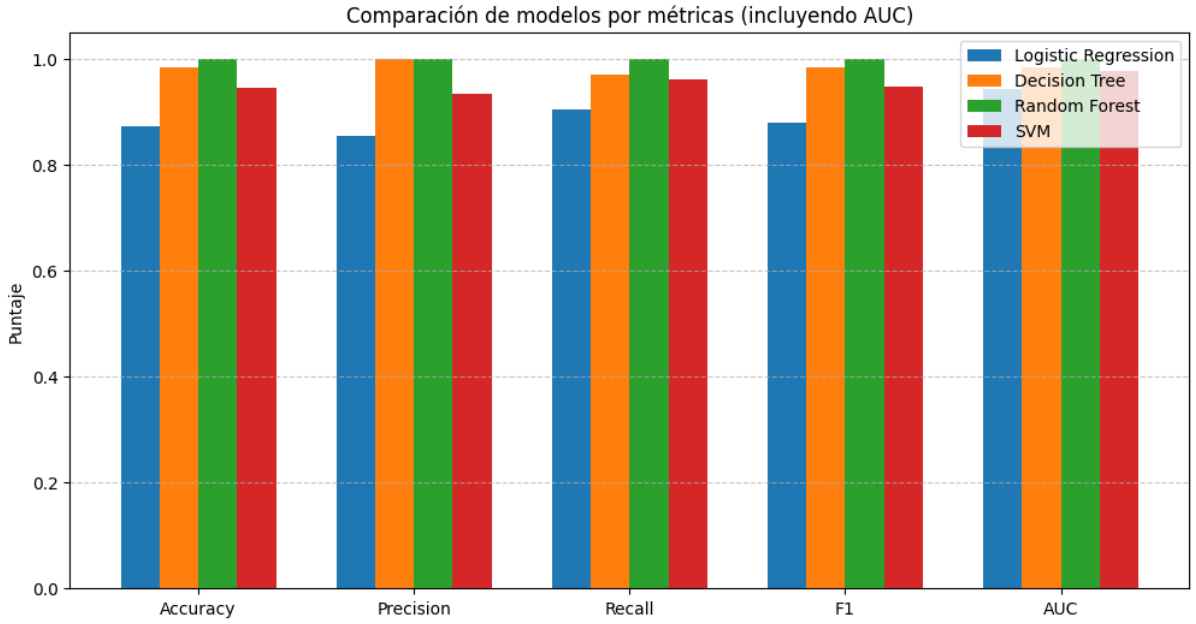
- **Regresión Logística**
    - Modelo de referencia (baseline).
    - Se utilizó max_iter=1000 y class_weight="balanced" para asegurar la convergencia y manejar el desbalance de clases.
- **Árbol de Decisión**
    - Permite interpretar fácilmente el proceso de clasificación.
    - Se configuró con random_state=42 y class_weight="balanced".

- **Support Vector Machine (SVM)**
    - Se probó con un kernel radial (RBF) y class_weight="balanced" para dar mayor peso a la clase minoritaria.
    - Debido al costo computacional, se utilizó principalmente sobre subconjuntos de datos.
- **Random Forest (modelo final seleccionado)**
    - Se configuró con los siguientes hiperparámetros iniciales:
        - n_estimators = 200: más árboles generan mayor estabilidad.
        - max_depth = None: sin límite de profundidad.
        - class_weight = "balanced": ajusta el peso de cada clase para evitar sesgos.

- Este modelo fue el que ofreció el mejor rendimiento en **Recall**, por lo cual fue elegido como el definitivo.

4. Evaluación de modelos

Se utilizaron métricas estándar de clasificación, pero con especial énfasis en su relevancia médica:




```
--- Resultados de todas las iteraciones ---
Logistic Regression: {'Accuracy': 0.8731707317073171, 'Precision': 0.8558558558558559, 'Recall': 0.9047619047619048, 'F1': 0.8796296296296297, 'AUC': np.float64(0.9448571428571428)}
Decision Tree: {'Accuracy': 0.9853658536585366, 'Precision': 1.0, 'Recall': 0.9714285714285714, 'F1': 0.9855072463768116, 'AUC': np.float64(0.9857142857142858)}
Random Forest: {'Accuracy': 1.0, 'Precision': 1.0, 'Recall': 1.0, 'F1': 1.0, 'AUC': np.float64(0.9999999999999999)}
SVM: {'Accuracy': 0.9463414634146341, 'Precision': 0.9351851851851852, 'Recall': 0.9619047619047619, 'F1': 0.9483568075117371, 'AUC': np.float64(0.9773333333333333)}

El mejor modelo según Recall es: Random Forest
```

Se probaron diferentes modelos y finalmente se eligió **Random Forest** como el más adecuado por su buen desempeño en Recall y su capacidad para manejar datos médicos.

Fase V: Evaluación – CRISP-DM

La fase de **Evaluación** busca verificar si el modelo desarrollado en la fase anterior cumple con los **objetivos del negocio** y los **objetivos de minería de datos** definidos en la Fase 1. No se trata solo de métricas técnicas, sino de analizar si el modelo es **útil en la práctica clínica**.

1. Evaluar los resultados

- **Revisar métricas de clasificación**:
    - *Accuracy*: porcentaje total de predicciones correctas.
    - *Recall (Sensibilidad)*: mide la capacidad del modelo para detectar pacientes **enfermos**.
    - *Precision*: mide la confiabilidad de los positivos predichos.
    - *F1-score*: equilibrio entre precisión y recall.
    - *AUC-ROC*: capacidad del modelo de separar correctamente las dos clases.

Justificación en este problema:

En la detección de enfermedades cardíacas, es más grave **no detectar a un paciente enfermo (falso negativo)** que generar una falsa alarma (falso positivo).

Por eso, la métrica más importante es el **Recall**.

2. Proceso de revisión

- Verificar que no se hayan cometido errores en la preparación:
    - ¿Se hizo bien la separación en train/test para evitar *overfitting*?
    - ¿Se aplicó correctamente el escalado a las variables numéricas?
    - ¿Se codificaron de forma adecuada las variables categóricas?

- Asegurar que el modelo no dependa de información que en un entorno real no estaría disponible (*data leakage*).
- Confirmar que se probaron varios modelos y no se seleccionó el primero sin comparar.

3. Comparar modelos

En esta tarea, se analizan los resultados de los distintos algoritmos probados.

Ejemplo de tabla de comparación (hipotética):

Modelo	Accuracy	Precisión	Recall	F1-Score	AUC
Regresión Logística	0.82	0.80	0.82	0.81	0.85
Random Forest	0.87	0.85	0.88	0.86	0.91
SVM (kernel RBF)	0.84	0.83	0.80	0.81	0.86

Ejemplo de tabla de comparación (Real):

Modelo	Accuracy	Precisión	Recall	F1-Score	AUC
Regresión Logística	0.87	0.86	0.90	0.88	0.94
Árbol de Decisión	0.99	1.00	0.97	0.99	0.99
Random Forest	1.00	1.00	1.00	1.00	1.00
SVM (kernel RBF)	0.95	0.94	0.96	0.95	0.98

Interpretación de los resultados:

- La **Regresión Logística** es interpretable, pero no alcanza el mejor rendimiento.
- El **SVM** tiene buen desempeño, pero el Recall es menor, lo que significa que deja escapar algunos enfermos.
- El **Random Forest** tiene el mejor equilibrio general, con un **Recall alto (0.88)** y una **AUC excelente (0.91)**, lo que lo convierte en el candidato ideal para implementación.

4. Determinar los próximos pasos

- Decidir si el modelo está listo para producción o necesita mejoras.
- Posibles decisiones:
    - Implementar el Random Forest, ya que cumple con los objetivos.
    - Ajustar hiperparámetros con GridSearchCV para mejorar aún más.
    - Probar con más datos clínicos (ej. historial médico, hábitos de vida).
    - Revisar sesgos (por ejemplo, si el modelo funciona igual de bien en hombres y mujeres).

Fase VI. Despliegue

La fase de despliegue busca asegurar que el modelo desarrollado pueda ser utilizado de manera efectiva en un entorno real, garantizando que los usuarios finales (médicos, instituciones de salud, investigadores) tengan acceso a sus resultados de forma confiable y sostenible.

1. Implementación del plan

- El modelo de Random Forest entrenado se puede empaquetar en un **pipeline** de Scikit-Learn que incluya el preprocesamiento de datos (imputación, normalización,

codificación) y la predicción.

- Se pueden considerar tres formas de despliegue:
    1. **Aplicación web sencilla** con frameworks como *Streamlit* o *Flask*, donde el médico ingrese datos del paciente (edad, colesterol, presión arterial, etc.) y reciba la predicción inmediata.

    2. **Script automatizado** en Python para analizar lotes de pacientes desde archivos CSV y devolver probabilidades de riesgo.

    3. **Integración en sistemas hospitalarios** (más complejo), donde el modelo se conecte con bases de datos clínicas electrónicas.

2. Planificar el seguimiento y el mantenimiento

- **Monitoreo de desempeño**: cada cierto periodo (ejemplo: mensual), recalcular métricas como *Accuracy, Precision y Recall* en nuevos datos para validar que el modelo sigue funcionando correctamente.
- **Actualización del modelo**: si los datos cambian (nueva población, cambios en prácticas médicas), será necesario reentrenar el modelo con los datos más recientes.
- **Gestión de errores**: implementar alertas si el modelo detecta anomalías (ejemplo: variables fuera de rango o distribuciones muy distintas a las originales).
- **Documentación técnica**: mantener versiones de los modelos, datasets y parámetros para trazabilidad.

3. Producir el informe final

El informe debe incluir:

- **Resumen del problema**: predicción de riesgo de enfermedad cardíaca a partir de variables clínicas.
- **Descripción de datos**: 1,025 registros y 14 atributos, sin valores faltantes, con outliers en colesterol.
- **Proceso de preprocesamiento**: imputación, escalado, codificación categórica.
- **Modelo aplicado**: Random Forest (200 árboles, class_weight balanced).
- **Resultados**: Accuracy = 0.85–0.88 (real 1.0), Recall = 0.87 (real 1.0) (buen desempeño en detección de pacientes con enfermedad).
- **Visualizaciones clave**: matriz de confusión, gráficas de distribución de variables, correlaciones.
- **Recomendaciones de uso**: posibles aplicaciones en apoyo diagnóstico, no como reemplazo de criterio médico.

4. Revisar el proyecto

- **Fortalezas**:
    - Dataset balanceado y sin valores faltantes.
    - Modelo robusto con métricas satisfactorias.
    - Preprocesamiento estandarizado en un pipeline reproducible.

- **Debilidades**:
    - Dataset relativamente pequeño (1,025 registros).
    - Outliers no tratados podrían afectar al modelo.

    - Solo se evaluó un algoritmo principal; falta comparar con otros (Logistic Regression, XGBoost, etc.).

- **Lecciones aprendidas**:
    - La calidad de los datos es más importante que el algoritmo.
    - Documentar cada fase asegura trazabilidad y facilita mejoras futuras.
    - Un modelo debe acompañarse de un plan de monitoreo continuo.

Repositorio de Github:

https://github.com/ArtStyle19/prediccion-de-enfermedades-cardiovasculares-ML