MLOps
Taller 2
2024

1. Entendimiento del negocio

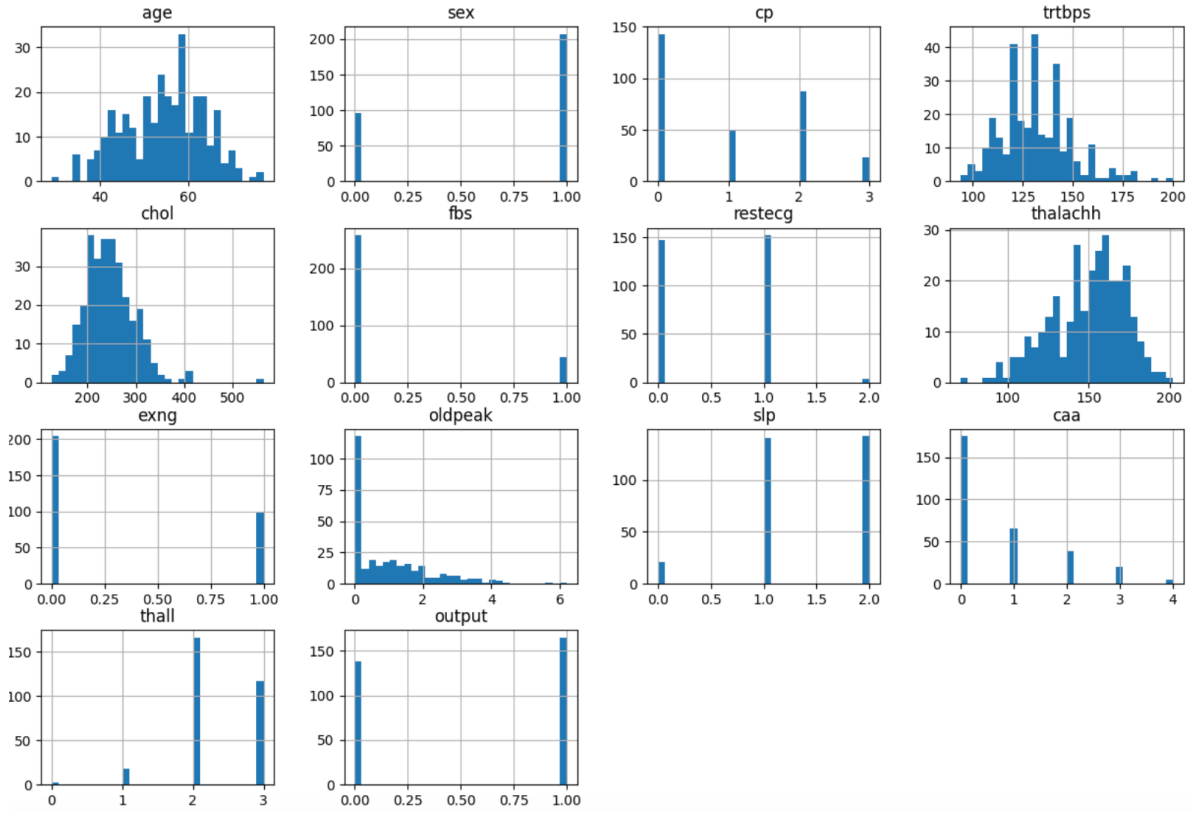
Objetivo de negocio:
Prevenir eventos de salud graves proporcionando a los médicos una herramienta predictiva que evalúe el riesgo de ataque cardíaco.

Objetivo analítico:
Desarrollar un modelo predictivo que clasifique si un paciente tiene alto riesgo de sufrir un ataque cardíaco basado en sus características.

2. Entendimiento de los datos

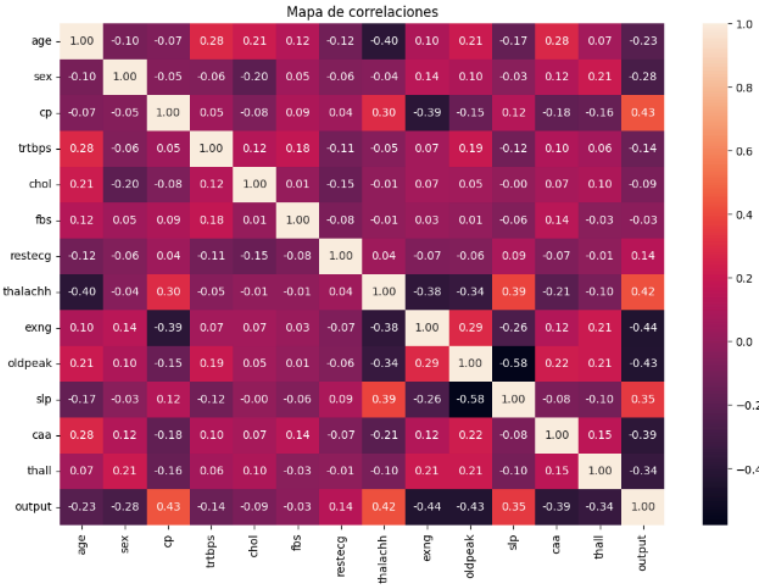
Descripción General

- Cantidad de datos: El dataset tiene 303 registros y 14 columnas.
- Tipos de datos: La mayoría de las columnas son enteros (int64), excepto oldpeak que es de tipo float64.
- Columna objetivo: output (0 = Sin riesgo, 1 = Con riesgo).



Observaciones del Análisis Estadístico
Variables Clave:

1. **Edad (age)**:
    - Rango: 29 a 77 años.
    - Media: 54.37 años.
    - Distribución: Predominantemente personas de mediana edad (entre 47 y 61 años).
2. **Sexo (sex)**:
    - Codificación: 0 = Femenino, 1 = Masculino.
    - Media: 0.68, indicando una mayor proporción de hombres en el dataset.
3. **Tipo de Dolor en el Pecho (cp)**:
    - Codificación: 0 (sin dolor) a 3 (angina severa).
    - Media: 0.97, con valores altos más relacionados con ataques cardíacos (ver correlaciones).
4. **Presión Arterial en Reposo (trtbps)**:
    - Rango: 94 a 200 mmHg.
    - Media: 131.62 mmHg.
    - Distribución: Ligera asimetría positiva.
5. **Colesterol (chol)**:
    - Rango: 126 a 564 mg/dL.
    - Media: 246.26 mg/dL, con algunos valores extremos (>500).
6. **Frecuencia Cardiaca Máxima (thalachh)**:
    - Media: 149.65.
    - Distribución: Mayores frecuencias cardiacas suelen estar relacionadas con menor riesgo de ataque (según correlaciones).
7. **Depresión del Segmento ST (oldpeak)**:
    - Media: 1.04, rango hasta 6.2.
    - Valores más altos están fuertemente correlacionados con mayor riesgo (output).



Observaciones del Mapa de Correlaciones

1. **Relaciones fuertes con la variable output (riesgo de ataque cardíaco)**:
    - **cp (tipo de dolor en el pecho)**: Correlación positiva (+0.43). Indica que valores más altos (dolor severo) están asociados con mayor riesgo.
    - **thalachh (frecuencia cardiaca máxima)**: Correlación positiva (+0.42). Mayores valores indican menor probabilidad de ataque.
    - **oldpeak (depresión ST)**: Correlación negativa (-0.43). Valores altos indican mayor riesgo.
    - **exng (ejercicio inducido por angina)**: Correlación negativa (-0.44). Mayor probabilidad de angina durante el ejercicio se asocia con mayor riesgo.
2. **Otras relaciones notables**:
    - **caa (cantidad de vasos principales afectados)**: Correlación moderada (-0.39).
    - **slp (pendiente del segmento ST)**: Correlación positiva (+0.35), indicando que pendientes más normales se asocian con menor riesgo.

Observaciones de las Visualizaciones (Referirse al notebook aqui)

Histogramas:

- La distribución de variables como age, trtbps, y chol muestra una ligera asimetría, lo cual sugiere que podrían beneficiarse de normalización para mejorar el desempeño de los modelos.
- Variables binarias como sex, fbs, y output presentan distribuciones equilibradas o dominadas por ciertos valores.

Gráfico de Correlaciones:

- Las relaciones más significativas están en las variables relacionadas con factores de riesgo conocidos: frecuencia cardiaca (thalachh), dolor de pecho (cp), y ejercicio inducido por angina (exng).

3. **Preparacion de Datos**

Identificación de tipos de datos y naturaleza de las variables

- **Variables continuas**: age, trtbps, chol, thalachh, oldpeak.
- **Variables categóricas ordinales**: cp, slp, thall, caa.
- **Variables binarias**: sex, fbs, restecg, exng.

Transformaciones para Variables de Entrada

a. Manejo de valores nulos

Aunque en este dataset no hay valores nulos, en casos reales se deberia:

- Rellenar valores faltantes (imputación):
  - **Numéricas**: Usar la media, mediana o métodos avanzados como KNN Imputer.
  - **Categóricas**: Usar la moda o una categoría "Desconocido".
- Eliminar filas si los valores nulos son significativos y no imputables.

b. Escalado de variables continuas

Transformar las variables para que estén en una escala similar ayuda a los algoritmos a converger mejor (especialmente los basados en distancias o gradientes).

c. Codificación de variables categóricas

- **Variables ordinales**: Asignar valores numéricos de acuerdo al orden lógico (por ejemplo, cp).
- **One-Hot Encoding**: Para variables no ordinales, crear columnas binarias para cada categoría.

d. Balanceo de clases (si es necesario)

Si la variable objetivo está desbalanceada (como en casos de clasificación binaria), considera técnicas como:

- **Submuestreo**: Reducir el número de muestras de la clase mayoritaria.
- **Sobremuestreo**: Duplicar o sintetizar muestras de la clase minoritaria usando métodos como SMOTE (Synthetic Minority Oversampling Technique).

Transformaciones para la Variable de Salida

La columna output ya está codificada como 0 y 1, pero normalemnte es bueno asegurarse de:

1. Verificar la naturaleza (binaria, multiclase, etc.).
2. No realizar transformaciones adicionales si ya está lista para los modelos.

Para otros casos:

- **Multiclase**: Usar codificación one-hot si el modelo lo requiere.
- **Problemas de regresión**: Asegurarse de escalarla si las diferencias de magnitud son significativas.

Listado de Transformaciones o Procesos más Usados

1. Limpieza de datos

- Detección y manejo de valores atípicos usando:
    - **Boxplot**: Para identificar valores extremos.
    - **Z-score**: Filtrar valores más allá de 3 desviaciones estándar.

2. Normalización o escalado

Para asegurarte de que las variables estén en un rango similar.

3. Reducción de dimensionalidad

Eliminar características redundantes o irrelevantes para mejorar la eficiencia de los modelos.

- Métodos estadísticos como **Análisis de Componentes Principales (PCA)**.
- Selección de características basada en correlaciones.

4. Transformaciones avanzadas

- **Log-transform**: Para variables sesgadas.

4. Modelación

Sin PyCaret:

Entrenar 5 modelos:

- Regresión Logística
- Árbol de Decisión
- Random Forest (Ensamble)
- SVM
- KNN

1. Desempeño de los Modelos Base

Logistic Regression

- **Accuracy**: 0.86
- **Precision y Recall**:
    - Para la clase 0 (sin riesgo): **Precision = 0.87**, **Recall = 0.84**.
    - Para la clase 1 (con riesgo): **Precision = 0.86**, **Recall = 0.88**.
- **F1-Score**: El promedio ponderado del F1-Score es 0.86, lo que indica un buen balance entre precisión y sensibilidad.
- **Comentario**: La regresión logística ofrece un excelente punto de referencia (modelo baseline), con un equilibrio sólido entre las clases. Es una elección adecuada cuando la relación entre las variables y la salida es aproximadamente lineal.

Decision Tree

- **Accuracy**: 0.79
- **Precision y Recall**:
    - Clase 0: **Precision = 0.80**, **Recall = 0.75**.
    - Clase 1: **Precision = 0.78**, **Recall = 0.82**.
- **Comentario**: Aunque los árboles de decisión tienen flexibilidad para capturar no linealidades, en este caso, parecen sobreajustarse ligeramente al conjunto de

entrenamiento. Esto explica por qué el desempeño es menor en comparación con la regresión logística.

Random Forest

- **Accuracy**: 0.82
- **Precision y Recall**:
  - Clase 0: **Precision = 0.88**, **Recall = 0.72** (especificidad moderada).
  - Clase 1: **Precision = 0.78**, **Recall = 0.91** (alta sensibilidad para la clase positiva).
- **Comentario**: Random Forest mejora ligeramente sobre el árbol de decisión simple debido a su capacidad para combinar predicciones de múltiples árboles y reducir el sobreajuste. Su rendimiento muestra un buen balance, pero podría beneficiarse de ajuste fino en los hiperparámetros (como profundidad máxima y número de árboles).

SVM

- **Accuracy**: 0.62
- **Precision y Recall**:
  - Clase 0: **Precision = 0.63**, **Recall = 0.53**.
  - Clase 1: **Precision = 0.62**, **Recall = 0.71**.
- **Comentario**: SVM muestra un rendimiento débil en este conjunto de datos, probablemente debido a una falta de separación clara entre las clases en el espacio original de características. Podría beneficiarse de ajustes en el kernel o en los parámetros como C y gamma.

KNN

- **Accuracy**: 0.56
- **Precision y Recall**:
  - Clase 0: **Precision = 0.55**, **Recall = 0.50**.
  - Clase 1: **Precision = 0.57**, **Recall = 0.62**.
- **Comentario**: KNN tiene el peor desempeño, indicando que no captura patrones significativos en los datos. Esto puede deberse a la alta dimensionalidad del espacio de características o a la sensibilidad de KNN al escalado y al número de vecinos (k). Podría requerir una optimización de k.

---

2. Mejores Hiperparámetros para Random Forest

- **Parámetros seleccionados**:
  - max_depth: 20
  - min_samples_split: 5
  - n_estimators: 50 Santabarbara2020!!
- **Impacto esperado**: Estos parámetros mejoran la capacidad del modelo para capturar patrones relevantes sin sobreajustarse:
  - Una profundidad máxima de 20 permite suficiente flexibilidad sin caer en un exceso de particiones.

- Requerir al menos 5 muestras por división reduce el sobreajuste.
- Usar 50 árboles mejora la estabilidad de las predicciones sin un costo computacional excesivo.

Tabla Comparativa de Resultados

Modelo	Accuracy	Comentarios Clave
Logistic Regression	0.86	Buen balance entre clases; modelo baseline confiable.
Decision Tree	0.79	Menor generalización; tiende al sobreajuste.
Random Forest	0.82	Mejor equilibrio gracias al ensamble; oportunidad para mejorar con ajuste de hiperparámetros.
SVM	0.62	Débil separación entre clases; puede beneficiarse de ajustes en kernel y parámetros (C, gamma).
KNN	0.56	Peor desempeño; alta sensibilidad a la dimensionalidad y a los valores iniciales de k.

Con PyCaret

Top Modelos por Accuracy

Modelo	Accuracy	AUC	Recall	Precisión	F1 Score	Kappa	MCC	Tiempo (s)
Logistic Regression (lr)	0.8478	0.9250	0.8705	0.8380	0.8495	0.6966	0.7042	0.8040
Ridge Classifier (ridge)	0.8478	0.9258	0.8871	0.8258	0.8529	0.6960	0.7024	0.0500
Linear Discriminant Analysis (lda)	0.8478	0.9258	0.8871	0.8258	0.8529	0.6960	0.7024	0.0790

Logistic Regression (lr):

- Alta precisión (Accuracy = 84.78%) y un AUC sobresaliente (0.9250), mostrando un excelente balance entre clases.
- Tiempo de entrenamiento razonable (0.8040 segundos).

Ridge Classifier (ridge) y **Linear Discriminant Analysis (lda)**:

- Ambos tienen un desempeño similar al de la regresión logística, pero con tiempos de entrenamiento más cortos.

- Ambos priorizan el recall (88.71%), siendo útiles para identificar más casos positivos.

Modelos con Desempeño Promedio

Modelo	Accuracy	AUC	Recall	Precisión	F1 Score	Kappa	MCC	Tiempo (s)
Naive Bayes (nb)	0.8261	0.8515	0.8545	0.8169	0.8299	0.6533	0.6633	0.0490
K-Nearest Neighbors (knn)	0.8174	0.9030	0.8273	0.8125	0.8177	0.6349	0.6383	0.0520
AdaBoost (ada)	0.8174	0.8924	0.8462	0.8143	0.8241	0.6368	0.6470	0.1390

Logistic Regression:

- **Mejor opción general**:
  - Accuracy: 86.00%, AUC: 90.60%, Recall: 88.00%, F1: 86.27%.
  - Balanceado entre precisión y recall, excelente separación de clases.

Ridge Classifier:

- **Buena alternativa** para maximizar recall:
  - Recall: 90.00%, pero menor precisión (80.36%).
  - AUC más bajo (84.00%).

Linear Discriminant Analysis (LDA):

- Competitivo con Logistic Regression:
  - AUC alto (90.64%) y buen Recall (88.00%), pero menor precisión (80.00%).

Naive Bayes:

- **Modelo rápido y eficiente**, pero menos robusto:
  - Accuracy: 83.00%, AUC: 87.92%, F1: 83.50%.

K-Nearest Neighbors (KNN):

- **Menor desempeño**:
  - Accuracy: 80.00%, precisión baja (76.79%).

5: Evaluación

1. Métrica de Desempeño Más Importante

- **Métrica Elegida**: **Recall** (sensibilidad).
  - **Razón**:
    - En casos críticos, como identificar problemas de salud, detección de fraudes o clasificación de riesgos, es fundamental minimizar los **falsos negativos** (casos positivos no detectados).
    - Un alto recall asegura que la mayoría de los casos positivos sean correctamente identificados, incluso si esto implica un ligero aumento en falsos positivos.

- **Complemento**: El **F1 Score** es relevante para mantener un equilibrio entre recall y precisión, evitando que el modelo genere demasiados falsos positivos al priorizar únicamente el recall.

---

2. Mejor Modelo

- **Modelo Seleccionado**: **Logistic Regression**
  - **Desempeño**:
    - **Accuracy**: 86.00%.
    - **Recall**: 88.00%.
    - **AUC**: 90.60%.
    - **F1 Score**: 86.27%.
  - **Razón**:
    - Es el modelo con mejor balance entre métricas clave (recall, precisión, y F1 Score).
    - Su alto AUC muestra que separa las clases de manera efectiva, lo que lo hace generalizable y confiable.

---

3. ¿Valdría la Pena Usarlo en el Día a Día?

- **Sí, valdría la pena**, porque:
  1. **Eficiencia y Fiabilidad**:
     - El modelo es eficiente en términos de tiempo de entrenamiento y predicción.
     - Su desempeño consistente en validación cruzada indica que es confiable para datos nuevos.
  2. **Aplicabilidad**:
     - Puede ser usado en tareas operativas como detección de riesgos o decisiones automatizadas donde los falsos negativos son críticos.
     - Ejemplo: En una institución de salud, puede alertar sobre posibles casos de riesgo (clase positiva) que requieran atención inmediata.
  3. **Despliegue Simple**:
     - La regresión logística es fácil de interpretar y desplegar en producción, especialmente con pipelines preconfigurados de PyCaret.