Pontificia Universidad Católica de Chile
Magíster en Ciencia de Datos

MCD - Aprendizaje estadístico y computacional

Trabajo final: aprendizaje supervisado y no supervisado

Integrantes:

- Helen María Fuentes
- Ricardo García Ramírez
- Tomás Eduardo Hansen
- David Adolfo Mateluna

Septiembre 2024

Contenido

Introducción

En este trabajo se presenta el proceso y los resultados de la aplicación de técnicas de aprendizaje supervisado y no supervisado sobre un conjunto de datos médicos relacionados con el diagnóstico de enfermedades cardíacas. El objetivo principal de

este trabajo es generar un modelo predictivo para determinar si un paciente presenta una enfermedad cardíaca.

A lo largo del documento, se describirán los análisis exploratorios realizados, las estrategias de depuración y selección de variables, así como la implementación de diversas técnicas de modelado.

En particular, se expondrán los pasos llevados a cabo en el análisis de los datos, comenzando con la limpieza y preprocesamiento, seguido de la evaluación de diferentes modelos de aprendizaje supervisado, y finalizando con la aplicación de técnicas de aprendizaje no supervisado para la identificación de grupos dentro de los datos.

El informe también incluirá una evaluación de los modelos seleccionados mediante métricas relevantes a la clasificación, y se discutirán los resultados en relación con el contexto clínico del problema. Finalmente, se presentarán las conclusiones obtenidas a partir de este ejercicio, destacando las fortalezas y limitaciones de los modelos analizados.

Objetivos específicos

1. Aplicar técnicas de limpieza y preprocesamiento a un conjunto de datos médicos para garantizar su adecuación en el proceso de modelado.
2. Resolver un problema del mundo real relacionado con el diagnóstico de enfermedades cardíacas, utilizando enfoques de aprendizaje supervisado y no supervisado.
3. Identificar las herramientas y técnicas más apropiadas para maximizar el rendimiento de los modelos predictivos, priorizando las métricas de evaluación como la precisión, el *recall* y el *F1-score*.
4. Interpretar los resultados obtenidos a partir de los modelos desarrollados, validando su efectividad en la predicción de enfermedad cardiaca.

Información del *dataset*

El dataset utilizado en este trabajo incluye los resultados de pruebas realizadas a pacientes que se sometieron a una angiografía para detectar posibles enfermedades cardíacas. La base se compone de los datos registrados en diferentes instituciones médicas con la siguiente distribución: “303 pacientes en la Cleveland Clinic en Cleveland, Ohio (…); 425 pacientes del Instituto Húngaro de Cardiología en Budapest, Hungría (…); 200 pacientes en el Centro Médico de la Administración de Veteranos en Long Beach, California (…); y 143 pacientes de los Hospitales Universitarios en Zúrich y Basilea, Suiza” (Detrano et al., 1989).

Janosi, Steinbrunn, Pfisterer y Detrano (1989) explican que el dataset está compuesto por 14 variables y que el campo de la variable objetivo podría contener hasta 5 valores donde 0 indica la ausencia de enfermedades cardiacas y 1,2,3 y 4 indican la presencia de alguna. Para los fines de nuestro modelado, utilizaremos una transformación a una variable binaria en la que consideraremos únicamente si existe o no enfermedad cardiaca.

Adicionalmente, esta base de datos contiene información general de los pacientes y de los resultados de los exámenes que se le tomaron para conocer su estado de salud al presentarse a la consulta médica. Esta es la información que se usará para el análisis exploratorio y posterior entrenamiento y evaluación de dos modelos, uno de aprendizaje supervisado y otro de aprendizaje no supervisado.

La Tabla presenta el contenido y descripción de las variables del set de datos, estos son principalmente parámetros cardiovasculares utilizados para el análisis de enfermedades del corazón.

Categoría	Variable	Descripción
Demográficas y generales	age	Edad del paciente en años.
	sex	Sexo del paciente (1.0 para masculino y 0.0 para femenino).
Parámetros clínicos	cp	Tipo de dolor de pecho experimentado.
	trestbps	Presión arterial en reposo en mm Hg.
	chol	Colesterol sérico en mg/dl.
	fbs	Azúcar en ayunas (1.0 si es mayor a 120 mg/dl, 0.0 en caso contrario).
	restecg	Resultados del electrocardiograma en reposo.
	thalach	Frecuencia cardíaca máxima alcanzada.
	exang	Angina inducida por ejercicio (1.0 sí, 0.0 no).
	oldpeak	Depresión del segmento ST inducida por el ejercicio en relación con el reposo.
	slope	Pendiente del segmento ST en el ejercicio máximo.
	ca	Número de vasos principales coloreados por fluoroscopia.
	thal	Resultado de la prueba de talio (3.0 normal, 6.0 defecto fijo, 7.0 defecto reversible).
Variables objetivo	target_cat	Categoría simplificada de la variable objetivo (0 para ausencia y 1 para presencia de enfermedad).

Tabla : Tabla de Descripción de Variables

Resultados

Análisis exploratorio de datos

Preprocesamiento del dataset

A partir de la *Figura 1,* se observa que la información tiene estructura tabular, está compuesta por datos categóricos y numéricos correspondientes a posibles precedentes de la enfermedad que se busca diagnosticar, son 14 columnas de variables independientes, más una variable objetivo (target) de la clase a clasificar. Contiene 920 filas en total, aparentemente sin contener datos faltantes o *null*. Luego se tiene que verificar que las columnas tengan su Dtype asignado correctamente y no existan datos anómalos.

```
Data columns (total 15 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   age       920 non-null    float64
 1   sex       920 non-null    float64
 2   cp        920 non-null    float64
 3   trestbps  920 non-null    object
 4   chol      920 non-null    object
 5   fbs       920 non-null    object
 6   restecg   920 non-null    object
 7   thalach   920 non-null    object
 8   exang     920 non-null    object
 9   oldpeak   920 non-null    object
 10  slope     920 non-null    object
 11  ca        920 non-null    object
 12  thal      920 non-null    object
 13  target    920 non-null    int64
 14  location  920 non-null    int64
dtypes: float64(3), int64(2), object(10)
memory usage: 107.9+ KB
```

Figura : Información del dataset con el método pd.info()

Para corroborar que realmente todos los datos son valores válidos, se realizó la siguiente tabla con valores únicos de los elementos que tiene cada una de las variables.

Columna	Valores únicos
age	63., 67., 37., 41., 56., 62.
sex	1., 0.
cp	1., 4., 3., 2.
trestbps	136.0, 146.0, 106.0, 156.0, 154.0, 114.0, 164.0, '130', '120', '140', '170', '100', '105', '110', '125', '150', '98'
chol	131.0, '132', '243', '?', '237', '219', '198', '225', '254', '298', '161'

fbs	1.0, 0.0, '0', '?', '1'
restecg	2.0, 0.0, 1.0, '2', '0', '1', '?'
thalach	134.0, 90.0, '185', '160', '170'
exang	0.0, 1.0, '0', '1', '?'
oldpeak	3.4, 6.2, '?', '1', '2.8', '0', '-1.1', '1.6', '-1.5', '2', '.5', '-.1', '-2.6', '2.1', '-.7'
slope	3.0, 2.0, 1.0, '?', '2', '1', '3'
ca	'0.0', '3.0', '2.0', '1.0', '?', '0', '1', '2'
thal	'6.0', '3.0', '7.0', '?', '6', '3', '7'
target	0, 2, 1, 3, 4
location	0, 1, 2, 3

Tabla : Análisis de valores únicos por cada columna, se omiten algunos datos por la cantidad.

De la Tabla se observa principalmente la existencia de valores numéricos expresados como texto y signos '?', '-.1' ambiguos. Por lo cuál será necesario hacer la limpieza de estas columnas. Se describe el siguiente procedimiento:

Se manejan los valores faltantes representados por el símbolo '?':

- Primero, se reemplazan estos valores por NaN (not a number) para que pandas pueda identificarlos como datos faltantes.
- Luego, se convierten las columnas afectadas en tipo float, ya que originalmente estaban mal interpretadas como object.
- Posteriormente, se eliminan las filas que contienen valores nulos.
- Finalmente, se revisan estadísticas descriptivas del dataset limpio.

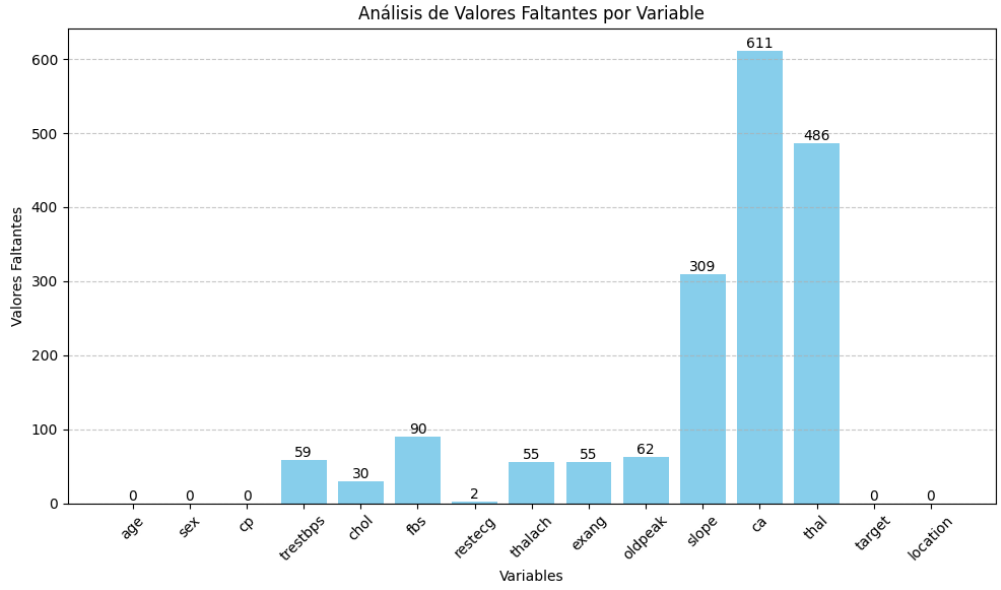


Figura : Gráfica de cantidad de variables con valores faltantes.

De la Figura , se observa que existe un gran número de datos faltantes para las columnas ‘slope’, ‘ca’ y ‘thal’. Sería necesario eliminar estas características para tener una mayor cantidad de datos, pero podría hacer el modelado menos exacto. También es posible eliminar los datos faltantes para tener la mayor cantidad de características para el modelo dejándonos con un número menor de datos a evaluar lo que podría ocasionar un modelo sobre ajustado a un dataset pequeño.

Vamos por el momento a eliminar los datos incompletos para evaluar un modelo con todas las características, posteriormente se tomará el modelo sin estas características y compararemos cuál nos da mejores métricas de desempeño.

Tras eliminar todos los datos faltantes, tenemos un total de 299 valores para nuestras 15 características y nuestro ‘target’. Luego de esto se revisará el balance del nuevo set de datos reducido.

location	Cantidad
0	297
1	1
3	1

Tabla : Cantidad de ocurrencias por variable, para location.

target	Cantidad
0	160
1	56
2	35
3	35
4	13

Tabla : Cantidad de ocurrencias por variable, para target.

De acuerdo con la Tabla , tenemos que *location* se encuentra muy desbalanceada, por lo que probablemente no aporte ninguna información relevante en este modelo actual, se decide por eliminarla del dataset.

De acuerdo con la Tabla , para *target* la clasificación es algo desbalanceada, vemos que los valores del 1 al 4 incluyen múltiples niveles de la enfermedad, pero para nuestros fines, únicamente nos interesa saber si el paciente tiene o no la enfermedad dadas las características mencionadas anteriormente. Debido a esto se decide utilizar una variable binaria *target_cat*, donde 0 representa que no existe enfermedad cardiaca y 1 si presenta enfermedad cardiaca. Véase *Tabla 5.*

categoría target_cat	Descripción
0	No existe enfermedad cardiovascular
1	Existe enfermedad cardiovascular

Tabla : Redefinición de variable categórica target u objetivo.

La nueva distribución de clases se muestra en la Figura . De ser necesario, se realizará un rebalance de clases para el modelado.

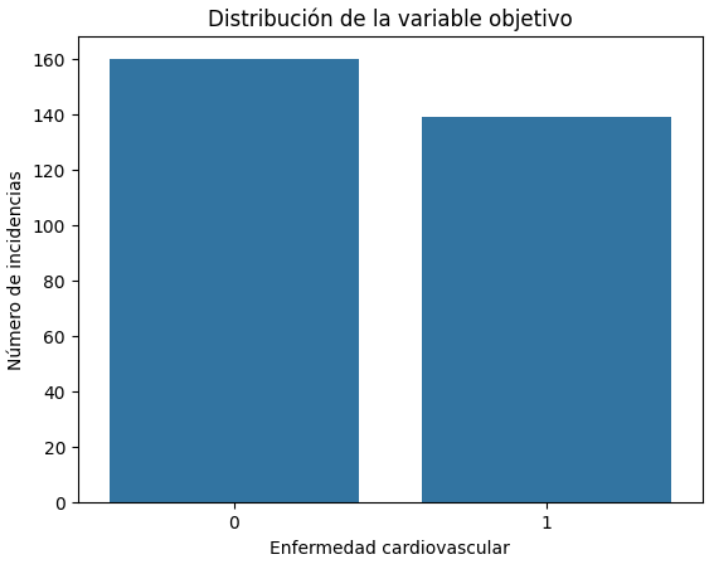


Figura : Gráfica de histograma para la variable objetivo.

Análisis exploratorio de datos (EDA) post-limpieza

Mediante EDA, examinaremos distribuciones, relaciones y patrones entre las variables, lo que nos permitirá identificar tendencias significativas, detectar anomalías o valores atípicos y formular hipótesis informadas para el desarrollo de nuestro modelo predictivo.

Correlación entre variables

Se realiza para visualizar las correlaciones entre las variables independientes del conjunto de datos. Se elimina la columna *target*, así da enfoque exclusivamente en las características o predictores, lo cual es esencial para entender cómo se relacionan entre sí, sin la influencia de la variable objetivo (Chan et al., 2022).

Este análisis es útil por tres razones principales:

1. Identificar multicolinealidad, ya que variables altamente correlacionadas pueden afectar negativamente el rendimiento de algunos modelos.
2. Entender las relaciones internas entre las variables, lo que puede ofrecer información sobre el comportamiento del sistema.
3. Mejorar el modelo predictivo al seleccionar características más relevantes y evitar incluir variables redundantes que no aportan nueva información.

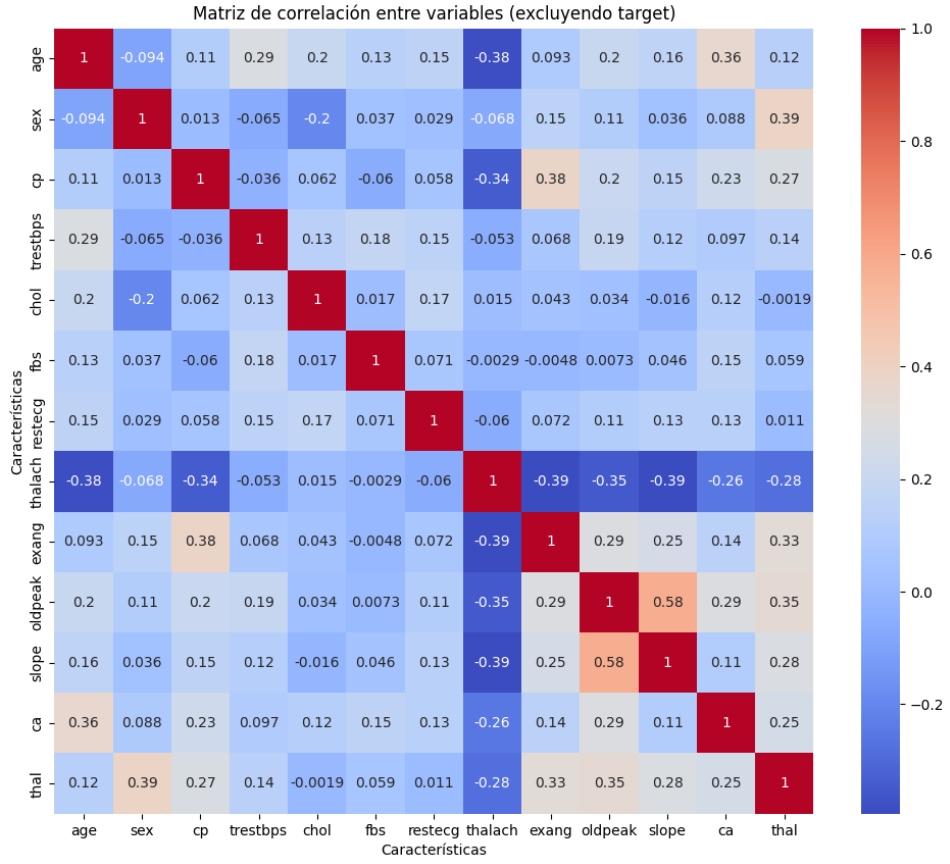


Figura : Gráfica de mapa de calor o 'heatmap' para la correlación de variables.

A partir de la Figura se distinguen dos casos respecto al valor absoluto de correlación y posibles casos con multicolinealidad de variables, lo que puede llevar a afectar la estabilidad, interpretabilidad y rendimiento de los modelos.

Existen algunas correlaciones moderadas que sugieren potencial multicolinealidad, como la relación entre 'cp' (tipo de dolor en el pecho) y 'exang' (angina inducida por ejercicio) con 0.38, y entre 'slope' y 'oldpeak' con 0.58. Estas correlaciones podrían indicar información redundante. Sin embargo, existe un umbral comúnmente utilizado para identificar multicolinealidada problemática, este es el valor absoluto > (0.7, 0.8). En este caso de estudio, ninguna de las relaciones supera el umbral de 0.7, por lo que, según este criterio, no es necesario eliminarlas (Allison, 2021).

Ahora, vamos a revisar la correlación de la variable objetivo con el resto de variables a evaluar:

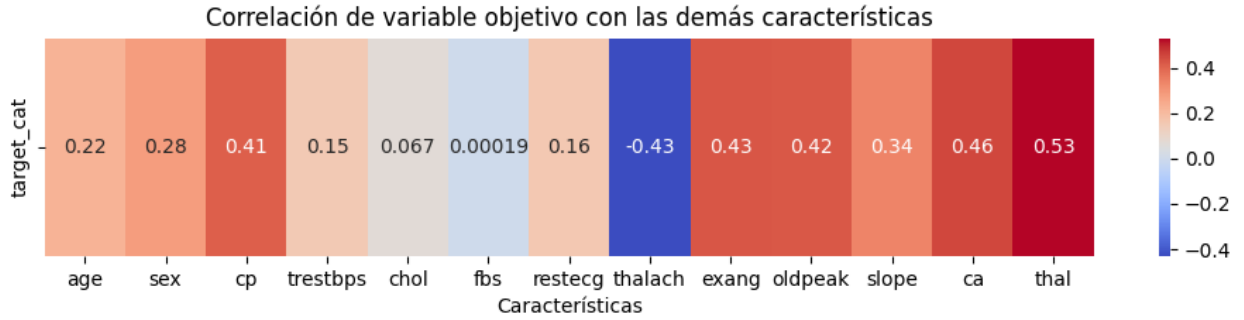


Figura : Mapa de calor de variable objetivo y variables independientes.

La Figura muestra esta correlación entre la variable objetivo 'target_cat' con las demás características del conjunto de datos, proporcionando una visión clara de cuáles variables tienen mayor influencia en la predicción del resultado.

Se destacan 'thal' y ca como las variables con la correlación más alta (0.53 y 0.46, respectivamente), indicando una fuerte asociación con la variable objetivo y sugiriendo que son características clave para el modelo predictivo.

Por otro lado, 'thalach' muestra una correlación negativa significativa (-0.43), lo que indica que valores altos de esta variable están asociados con un menor valor de 'target_cat'.

Variables como 'fbs' y 'chol' presentan correlaciones muy bajas (0.00019 y 0.067), sugiriendo que su influencia sobre la variable objetivo es mínima y podrían ser consideradas menos relevantes en el análisis predictivo.

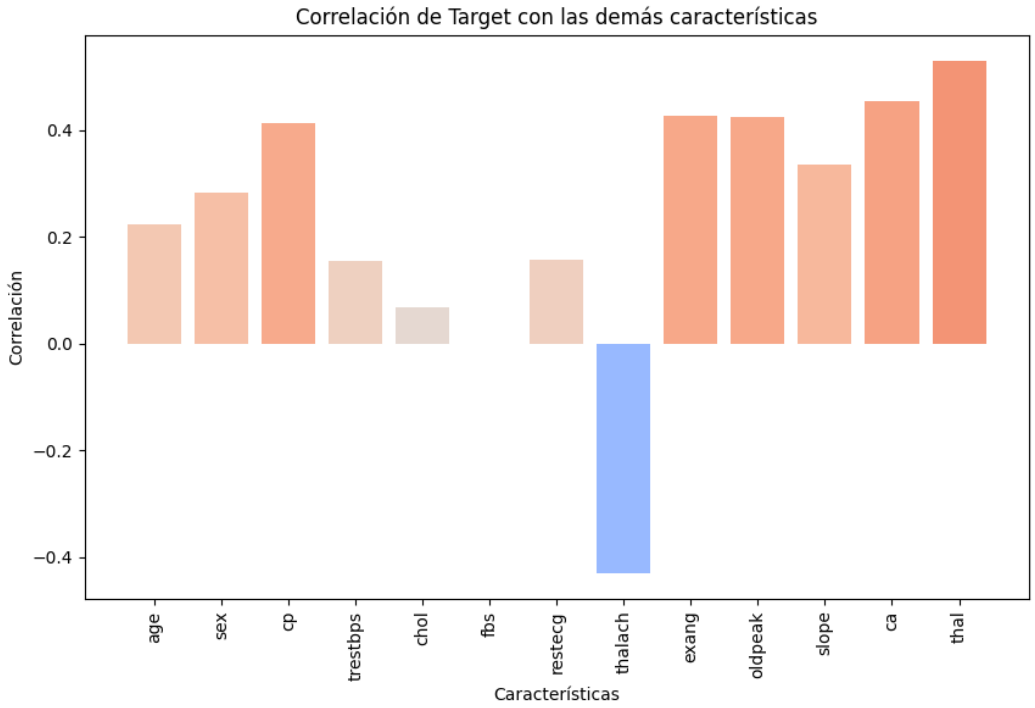


Figura : Gráfico de barras para representar valores de correlación y su influencia.

En la Figura , se observa la correlación de la variable objetivo con las demás características del conjunto de datos de una forma más visual.

Las características con mayor correlación positiva son: ‘cp’ (tipo de dolor en el pecho), ‘thal’, ‘ca’, ‘slope’ y ‘exang’, todas con valores cercanos o mayores a 0.4. Estos valores indican una relación directa moderada, es decir, si sus valores aumentan, también lo hace la probabilidad de presentar la enfermedad cardiaca; son candidatas para ser incluidas en el modelo predictivo.

Por otro lado, ‘thalach’ (frecuencia cardíaca máxima) muestra una correlación negativa moderada, aprox. -0.4, lo que sugiere que a medida que esta variable aumenta, la probabilidad del presentar la enfermedad disminuye; también puede ser una característica relevante en el análisis.

Las demás variables: ‘chol’, ‘fbs’ y ‘restecg’ presentan correlaciones cercanas a cero, indicando que tienen un impacto mínimo en la variable objetivo y podrían ser consideradas para una revisión adicional o posible eliminación del modelo si su inclusión no agrega valor predictivo.

Selección de modelo usando técnicas de aprendizaje supervisado

Selección de características

Este procedimiento mejora el rendimiento del modelo al eliminar variables irrelevantes, reduciendo el ruido y el riesgo de sobreajuste. Esto facilita que el modelo generalice mejor, reduce tiempos de entrenamiento y simplifica la interpretación, destacando solo las variables más influyentes en las predicciones (Göcs & Johanyák, 2023).

En esta sección vamos a usar dos tipos de selección de características: selección hacia adelante y hacia atrás. Como métrica de comparación y selección se utiliza F1_score pues nos importan evitar falsos negativos (es decir, no dejar de diagnosticar a personas que sí tienen la enfermedad) además de limitar el número de falsos positivos.

Se define un modelo de clasificación binario basado en Regresión Logística con ajuste de pesos para clases desbalanceadas, sin aplicar técnicas de muestreo y estandarizando los datos con **StandardScaler**. Se utiliza una función de puntuación personalizada enfocada en ***F1_score*** para priorizar el manejo de clases desbalanceadas. La selección de características se realiza mediante métodos hacia adelante y hacia atrás usando **SequentialFeatureSelector**. Los pipelines generados se evalúan en los datos de entrenamiento y prueba, comparando su rendimiento en términos de precisión y ajuste, obteniendo los siguientes resultados:

Method	Accuracy	Precision	Recall	F1	AUC
Logistic Regression - StandardScaler	0.824113	0.806212	0.828458	0.813962	0.900433
Logistic Regression - StandardScaler - Forward Selection	0.773670	0.736960	0.819763	0.772605	0.861380
Logistic Regression - StandardScaler - Backward Selection	0.845035	0.845493	0.819763	0.828061	0.900355

Tabla : Cross-Validation Metrics

Method	Accuracy	Precision	Recall	F1	AUC
Logistic Regression - StandardScaler	0.80	0.833333	0.714286	0.769231	0.907366
Logistic Regression - StandardScaler - Forward Selection	0.75	0.782609	0.642857	0.705882	0.866071
Logistic Regression - StandardScaler - Backward Selection	0.75	0.782609	0.642857	0.705882	0.895089

Tabla : Test Set Metrics

Los resultados anteriores muestran las métricas de rendimiento tanto en validación cruzada como en el conjunto de prueba para diferentes configuraciones de la Regresión Logística, ya que se quiere determinar el mejor conjunto de características para trabajar con el modelo.

En la validación cruzada, el pipeline con la Regresión Logística utilizando StandardScaler sin selección de características obtuvo el mejor rendimiento en términos de F1-score (0.817633) y AUC (0.901120), lo cual indica un buen balance entre precisión y recall, además de una capacidad robusta de discriminación entre clases.

En el conjunto de prueba, el modelo con StandardScaler sin selección de características logró el mejor F1-score de 0.769231. Además, el modelo con StandardScaler sin selección de características también mostró un AUC superior (0.907366), sugiriendo un mejor desempeño en la clasificación de clases positivas frente a negativas.

Por lo tanto, basado en el F1-score y un desempeño superior en AUC, el mejor modelo es el que utiliza Regresión Logística con StandardScaler sin selección de características.

Verificación de modelo eliminando las características con el menor número de datos

A continuación, se debe verificar los resultados del modelo si es que se quitaran las variables categóricas que tenían el mayor número de datos faltantes (‘slope’, ‘ca’ y ‘thal’), con el fin de comprobar como cambian las métricas con los mismos pipelines anteriormente generados. Se hace el mismo proceso para limpiar los datos que en el preprocedamiento anterior. El nuevo dataset y métricas quedan de la siguiente manera:

```
Data columns (total 13 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   age         740 non-null    float64
 1   sex         740 non-null    float64
 2   cp          740 non-null    float64
 3   trestbps    740 non-null    float64
 4   chol        740 non-null    float64
 5   fbs         740 non-null    float64
 6   restecg     740 non-null    float64
 7   thalach     740 non-null    float64
 8   exang       740 non-null    float64
 9   oldpeak     740 non-null    float64
 10  target      740 non-null    int64
 11  location    740 non-null    int64
 12  target_cat  740 non-null    int64
dtypes: float64(10), int64(3)
memory usage: 80.9 KB
```

Figura : Información del dataset con el método pd.info() sin variables con mayor número de datos faltantes

Method	Accuracy	Precision	Recall	F1	AUC
Logistic Regression - StandardScaler	0.797265	0.819335	0.784030	0.799142	0.887397
Logistic Regression - StandardScaler - Forward Selection	0.756758	0.757447	0.781121	0.768867	0.842317
Logistic Regression - StandardScaler - Backward Selection	0.785501	0.797892	0.787361	0.790754	0.882197

Tabla : Cross-Validation Metrics (sin variables categorícas con muchos datos faltantes)

Method	Accuracy	Precision	Recall	F1	AUC
Logistic Regression - StandardScaler	0.844595	0.864865	0.831169	0.847682	0.902140
Logistic Regression - StandardScaler - Forward Selection	0.837838	0.863014	0.818182	0.840000	0.912018
Logistic Regression - StandardScaler - Backward Selection	0.831081	0.833333	0.844156	0.838710	0.908542

Tabla : Test Set Metrics (sin variables categorícas con muchos datos faltantes)

Con esto, surgen 2 preguntas sobre el modelo y el dataset a escoger para avanzar con las siguientes etapas:

- Sobre el modelo, cuál se debe escoger dentro de las siguientes elecciones: sin feature selection, backward, forward.
- Sobre el dataset, cuál se debe escoger dentro de aquel que tiene más características, pero menor muestra o aquel que tiene menos características, pero mayor muestra.

La respuesta al primer punto lo entregan la Tabla y la Tabla , en donde se aprecian mejores valores en términos de Recall, F1 y AUC para el modelo sin selección de características. Con respecto a la segunda pregunta, esta será revisada en los siguientes puntos de este informe.

Validación de balanceo de clases

Aunque en este caso específico las clases están relativamente balanceadas, hacer el remuestreo es crucial en escenarios de desbalanceo significativo para asegurar que los modelos no solo alcancen altos valores de exactitud, sino que también funcionen de manera efectiva en el contexto específico de la predicción de interés, manejando correctamente ambas clases y reduciendo el riesgo de errores críticos en la clasificación.

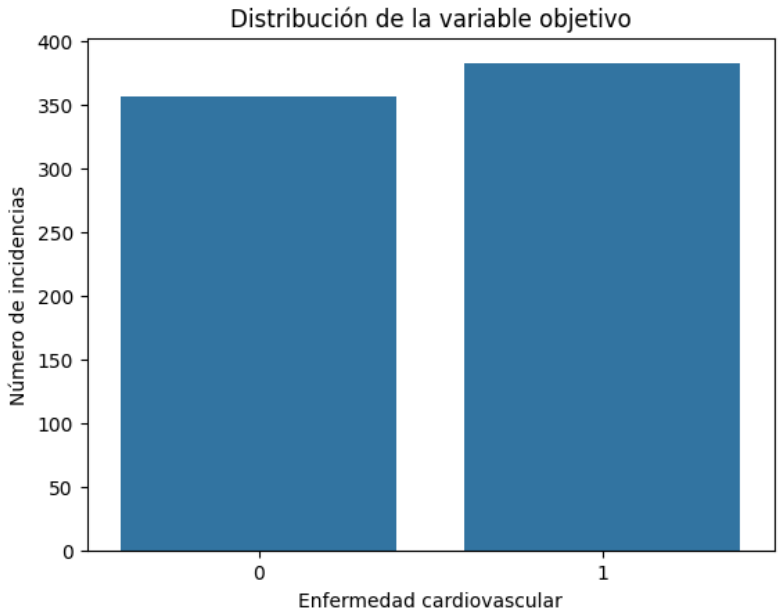


Figura : Distribución de variable objetivo

De acuerdo con lo que se observa de la Figura , posiblemente no sea necesario aplicar remuestreo, ya que las clases están razonablemente equilibradas y el rendimiento del modelo es bueno. Sin embargo, potencialmente podría mejorar el rendimiento del modelo al remuestrear de forma equilibrada y asegurando que el modelo continúe funcionando adecuadamente en ambas clases.

Para lo anterior, a continuación, se ilustran los resultados de un cross validation, para revisar un modelo de regresión logística utilizando distintas técnicas de remuestreo, como:

- Smote
- Random Oversampling
- Random Undersampling
- NearMiss
- Sin Remuestreo

Primero se analiza el dataset con menos variables, pero con más muestra, lo cual arrojó los siguientes resultados:

Method	Accuracy	Precision	Recall	F1	AUC
Logistic Regression - StandardScaler	0.797265	0.819335	0.784030	0.799142	0.887397
Logistic Regression - SMOTE - StandardScaler	0.795570	0.816263	0.784030	0.797368	0.886253
Logistic Regression - Random Oversampling - StandardScaler	0.798932	0.818949	0.787308	0.801001	0.887004
Logistic Regression - Random Undersampling - StandardScaler	0.798946	0.824093	0.780751	0.799552	0.886311
Logistic Regression - NearMiss - StandardScaler	0.793890	0.804665	0.797091	0.799137	0.886315

Tabla : Cross-Validation Metrics (dataset con menos variables, más muestra)

Method	Accuracy	Precision	Recall	F1	AUC
Logistic Regression - StandardScaler	0.844595	0.864865	0.831169	0.847682	0.902140
Logistic Regression - SMOTE - StandardScaler	0.837838	0.863014	0.818182	0.840000	0.905067
Logistic Regression - Random Oversampling - StandardScaler	0.851351	0.866667	0.844156	0.855263	0.899762
Logistic Regression - Random Undersampling - StandardScaler	0.837838	0.853333	0.831169	0.842105	0.905067
Logistic Regression - NearMiss - StandardScaler	0.844595	0.855263	0.844156	0.849673	0.904884

Tabla : Test Set Metrics (dataset con menos variables, más muestra)

Esto mismo se realiza para el dataset con más variables pero con menos muestra, obteniendo los siguientes resultados:

Method	Accuracy	Precision	Recall	F1	AUC
Logistic Regression - StandardScaler	0.824113	0.806212	0.828458	0.813962	0.900433
Logistic Regression - SMOTE - StandardScaler	0.828369	0.810550	0.837549	0.820177	0.900360
Logistic Regression - Random Oversampling - StandardScaler	0.836702	0.826070	0.828458	0.824999	0.904977
Logistic Regression - Random Undersampling - StandardScaler	0.824113	0.827722	0.801976	0.807449	0.905312
Logistic Regression - NearMiss - StandardScaler	0.836613	0.819790	0.837154	0.826630	0.905292

Tabla : Cross-Validation Metrics (dataset con más variables, menos muestra)

Method	Accuracy	Precision	Recall	F1	AUC
Logistic Regression - StandardScaler	0.800000	0.833333	0.714286	0.769231	0.907366
Logistic Regression - SMOTE - StandardScaler	0.800000	0.833333	0.714286	0.769231	0.908482
Logistic Regression - Random Oversampling - StandardScaler	0.783333	0.800000	0.714286	0.754717	0.891741
Logistic Regression - Random Undersampling - StandardScaler	0.833333	0.875000	0.750000	0.807692	0.921875
Logistic Regression - NearMiss - StandardScaler	0.750000	0.760000	0.678571	0.716981	0.879464

Tabla : Test Set Metrics (dataset con más variables, menos muestra)

Con lo anterior, se observa que para que caso del dataset con mayor número de variables un mejor desempeño del modelo en términos de Recall (0.844156) y F1 (0.855263) cuando la técnica de resampleo utilizada es Random Oversampling.

Por otro lado, en el dataset con menor número de variables, se observa un mejor desempeño del modelo en términos de Recall (0.750000), AUC (0.921875) y F1 (0.807692) cuando la técnica de resampleo utilizada es Random Undersampling.

Dado esto, para el dataset de menos variables se decide utilizar la técnica de resampleo de Random Oversampling y para el dataset de menos variables la técnica de Random Undersampling.

Selección de modelo supervisado

Derivado del punto anterior, para escoger el modelo con la técnica de ML que mejor predice la variable objetivo, se analizarán los siguientes modelos de clasificación:

- Logistic Regression
- Decision Tree
- Random Forest
- SVC
- K-Nearest Neighbors

A continuación, se ilustran los resultados obtenidos para cada dataset con las definiciones anteriores:

Method	Accuracy	Precision	Recall	F1	AUC
Logistic Regression - Random Oversampling - StandardScaler	0.798932	0.818949	0.787308	0.801001	0.887004
Decision Tree - Random Oversampling - StandardScaler	0.719598	0.735101	0.715389	0.724116	0.719582
Random Forest - Random Oversampling - StandardScaler	0.787139	0.803113	0.784135	0.791724	0.867577
SVC - Random Oversampling - StandardScaler	0.814101	0.831481	0.810312	0.818550	0.880463
K-Nearest Neighbors - Random Oversampling - StandardScaler	0.792152	0.803750	0.793919	0.797249	0.851133

Tabla : Cross-Validation Metrics (Dataset menos variables) (Selección Modelos)

Method	Accuracy	Precision	Recall	F1	AUC
Logistic Regression - Random Oversampling - StandardScaler	0.851351	0.866667	0.844156	0.855263	0.899762
Decision Tree - Random Oversampling – StandardScaler	0.736486	0.771429	0.701299	0.734694	0.737973
Random Forest - Random Oversampling – StandardScaler	0.783784	0.792208	0.792208	0.792208	0.880830
SVC - Random Oversampling – StandardScaler	0.817568	0.820513	0.831169	0.825806	0.895372
K-Nearest Neighbors - Random Oversampling - StandardScaler	0.790541	0.810811	0.779221	0.794702	0.854308

Tabla : Test Set Metrics (Dataset menos variables) (Selección Modelos)

Method	Accuracy	Precision	Recall	F1	AUC
Logistic Regression - Random Undersampling - StandardScaler	0.824113	0.827722	0.801976	0.807449	0.905312
Decision Tree - Random Undersampling – StandardScaler	0.740603	0.702446	0.783794	0.736899	0.744051
Random Forest - Random Undersampling – StandardScaler	0.799202	0.783957	0.774704	0.775233	0.893638
SVC - Random Undersampling – StandardScaler	0.815780	0.800101	0.820553	0.802700	0.914003
K-Nearest Neighbors - Random Undersampling - StandardScaler	0.782270	0.773491	0.784190	0.768544	0.856050

Tabla : Cross-Validation Metrics (Dataset más variables) (Selección Modelos)

Method	Accuracy	Precision	Recall	F1	AUC
Logistic Regression - Random Undersampling - StandardScaler	0.833333	0.875000	0.750000	0.807692	0.921875
Decision Tree - Random Undersampling - StandardScaler	0.716667	0.739130	0.607143	0.666667	0.709821
Random Forest - Random Undersampling - StandardScaler	0.816667	0.814815	0.785714	0.800000	0.881138
SVC - Random Undersampling - StandardScaler	0.833333	0.846154	0.785714	0.814815	0.924107
K-Nearest Neighbors - Random Undersampling - StandardScaler	0.800000	0.863636	0.678571	0.760000	0.906808

Tabla : Test Set Metrics (Dataset más variables) (Selección Modelos)

De los resultados anteriores, se observa que el modelo con mejores desempeños en términos de Recall (0.844156) es el “Logistic Regression - Random Oversampling - StandardScaler” del dataset con menos variables. También presenta valores cercanos a lo mejores modelos en términos de AUC (0.899762) y F1 (0.855263). Dado esto, éste será el modelo escogido en el presente análisis.

Ajuste de hiperparámetros del modelo

Dentro del modelo escogido, en esta sección se revisarán las combinaciones de los siguientes hiperparámetros:

- **'classifier__C':**
  Este hiperparámetro controla la fuerza de la regularización de forma inversa (un valor más pequeño de C implica mayor regularización). Los valores probados son [0.01, 0.1, 1, 10, 100], que van de una regularización más fuerte a más débil.
- **'classifier__penalty':**
  Este hiperparámetro especifica el tipo de regularización aplicada:
  - **'l1'**: Regularización Lasso que intenta forzar algunos coeficientes a cero, haciendo una selección de características.
  - **'l2'**: Regularización Ridge que penaliza los coeficientes altos, pero no los fuerza a cero.
  - **'elasticnet'**: Combinación de las penalizaciones L1 y L2, controlada por el hiperparámetro l1_ratio (definido más abajo).
  - **'None'**: Sin regularización (solo se prueba con 'lbfgs' y 'saga').
- '**classifier__solver**':
  Este hiperparámetro define el algoritmo que usará la Regresión Logística para encontrar los coeficientes óptimos:
  - **'lbfgs'**: Un algoritmo de optimización adecuado para problemas de tamaño medio a grande. Funciona bien con la regularización **L2**.
  - **'liblinear'**: Un solver que usa el método de descenso en coordenadas, adecuado para problemas más pequeños o con penalización **L1** o **L2**.
  - **'saga'**: Un algoritmo más avanzado y eficiente, adecuado para penalizaciones **L1**, **L2**, y **ElasticNet**.
- '**classifier__l1_ratio**':
  Este hiperparámetro solo aplica cuando se usa 'elasticnet' como penalización. Controla la mezcla de las regularizaciones L1 y L2. Los valores a utilizar serán los siguientes:
  - l1_ratio=0.0: Equivale a L2 (solo Ridge).
  - l1_ratio=1.0: Equivale a L1 (solo Lasso).
  - l1_ratio=0.5: Una mezcla equilibrada entre L1 y L2.

Junto con lo anterior, se utilizó la función **StratifiedKFold** para realizar una validación cruzada sobre 5 splits. Además, se utilizó la función **GridSearchCV**, para ejecutar las combinaciones definidas en la , de manera de seleccionar la mejor combinación de hiperparámetros que maximicen el F1 score.

Con esto se obtuvieron 55 candidatos de combinaciones a analizar con lo que se generaron 275 ajustes de modelos a realizar con los datos de entrenamiento, obteniéndose los siguientes resultados:

```
Fitting 5 folds for each of 55 candidates, totalling 275 fits
Mejores hiperparámetros: {'classifier__C': 0.1, 'classifier__l1_ratio': 0.5, 'classifier__penalty': 'elasticn
et', 'classifier__solver': 'saga'}
```

Figura : Resultados Grilla de hiperparámetros

Evaluación del modelo seleccionado

Con los hiperparámetros obtenidos en el ejercicio anterior, finalmente se procede a evaluar el modelo seleccionado. Estos son las principales métricas obtenidos sobre los datos de prueba, la Matriz de Confusión y el Threshold respectivamente:

	Accuracy	Precision	Recall	F1-score	AUC-ROC
Mejor Modelo	0.858108	0.868421	0.857143	0.862745	0.909091

Tabla : Resultados Test modelo seleccionado

La matriz de confusión presentada en la Figura 11 de la que se obtienen los resultados de la Tabla 18 es clave para evaluar el rendimiento del modelo seleccionado. En este caso, se observa que el modelo predice correctamente 61 casos negativos (sin enfermedad) y 66 casos positivos (con enfermedad). Sin embargo, comete 10 errores al predecir falsos positivos (clasificando erróneamente a pacientes sanos como enfermos) y 11 errores de falsos negativos (clasificando erróneamente a pacientes enfermos como sanos).

Estos resultados son importantes ya que el balance entre precisión (evitar falsos positivos) y recall (evitar falsos negativos) es crítico en el contexto clínico, donde no detectar a un paciente con enfermedad puede tener graves consecuencias. Por ello, el modelo muestra un buen rendimiento general, aunque es importante seguir monitoreando estos errores para mejorar la capacidad de detección del modelo de utilizarse en un ambiente clínico.

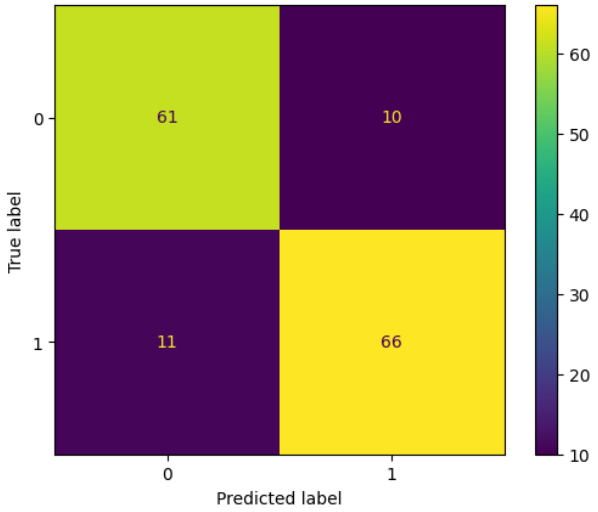


Figura : Matriz Confusión (Test) modelo seleccionado

La gráfica de precisión y recall en función del umbral que se muestra en la Figura 12 se complementa con las métricas de rendimiento del modelo seleccionado en la Tabla 18. En nuestro caso, el mejor modelo presenta una precisión de 0.868, un recall de 0.857, un F1-score de 0.863 y un AUC-ROC de 0.909. Estas métricas sugieren que el modelo tiene un buen balance entre precisión y recall, lo que significa que logra identificar correctamente una alta proporción de pacientes con enfermedad cardíaca, al mismo tiempo que minimiza los falsos positivos.

En relación con la gráfica, podemos ver que el umbral de decisión elegido permite mantener un buen compromiso entre precisión y recall, ya que ambos valores son altos. Esto indica que el umbral actual permite al modelo no solo predecir correctamente los casos de enfermedad (alto recall), sino también hacerlo con un bajo número de falsos positivos (alta precisión). El F1-score de 0.863, que es una medida armonizada de precisión y recall, confirma que el equilibrio alcanzado es óptimo, favoreciendo un rendimiento robusto del modelo.

Además, el AUC-ROC de 0.909 indica que el modelo distingue entre clases positivas y negativas en diferentes umbrales, lo que refuerza la idea de que es efectivo en términos de discriminación entre clases.

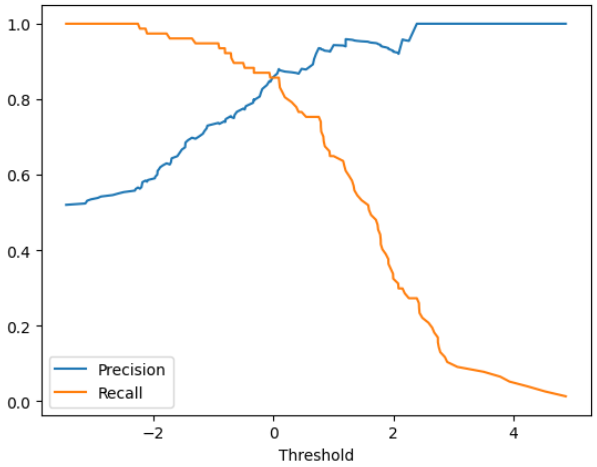


Figura : Threshold (Test) modelo seleccionado

En resumen, tanto la gráfica como las métricas generales del modelo sugieren que se ha encontrado un umbral que proporciona un equilibrio adecuado entre precisión y recall. Este umbral maximiza la detección de la enfermedad cardíaca sin generar una cantidad excesiva de falsos positivos. A partir de esta información, podemos concluir que el modelo es sólido y está bien ajustado para el propósito clínico de predicción de enfermedades cardíacas, sin embargo, sería necesario continuar evaluando el modelo con nuevos sets de datos para continuar mejorando el mismo con más datos.

Análisis de clúster con aprendizaje no supervisado

Análisis de Componentes Principales

La técnica de Análisis de Componentes Principales (PCA, por sus siglas en inglés) es un método de reducción de dimensionalidad que transforma un conjunto de variables posiblemente correlacionadas en un nuevo conjunto de variables no correlacionadas, denominadas componentes principales (Kurita, 2020). Cada componente principal captura la mayor cantidad posible de varianza en los datos, lo que permite simplificar la complejidad del conjunto de datos mientras se mantiene la mayor parte de la información relevante.

En este análisis, PCA nos permite reducir las variables originales del dataset a un número más manejable de componentes principales. Esto es especialmente útil en datasets donde las variables están correlacionadas entre sí, lo que puede afectar el rendimiento de los modelos de aprendizaje automático. Al utilizar PCA, eliminamos las correlaciones redundantes entre variables y nos centramos en las nuevas componentes que capturan la mayor variabilidad en los datos.

```
Varianza explicada por las primeras dos componentes: [0.20821463 0.11391912]
```

En este caso, hemos calculado la varianza explicada por cada componente principal para determinar cuántas de estas componentes necesitamos para conservar la mayor parte de la información del dataset. Las primeras dos componentes principales suelen ser las que explican una mayor proporción de la varianza total. Esto significa que gran

parte de la información del dataset original se puede representar adecuadamente con solo estas dos componentes.

Para este dataset, la primera componente principal explica una gran parte de la varianza de los datos, mientras que la segunda componente también captura una proporción significativa. Al sumar la varianza explicada por ambas componentes, logramos capturar una cantidad considerable de la información presente en el conjunto de datos original, lo que sugiere que podemos reducir la dimensionalidad sin perder demasiada información. Esto también facilita la visualización y el análisis de los datos, ya que podemos representar gráficamente las observaciones en un espacio de dos dimensiones, permitiendo una mejor comprensión de los patrones presentes en el dataset.

Este proceso de reducción de dimensionalidad a través de PCA ayuda a simplificar los datos y a mejorar la eficiencia de los modelos de aprendizaje automático al eliminar el ruido y las redundancias en los datos, lo que permite que el modelo se concentre en las características relevantes para la predicción.

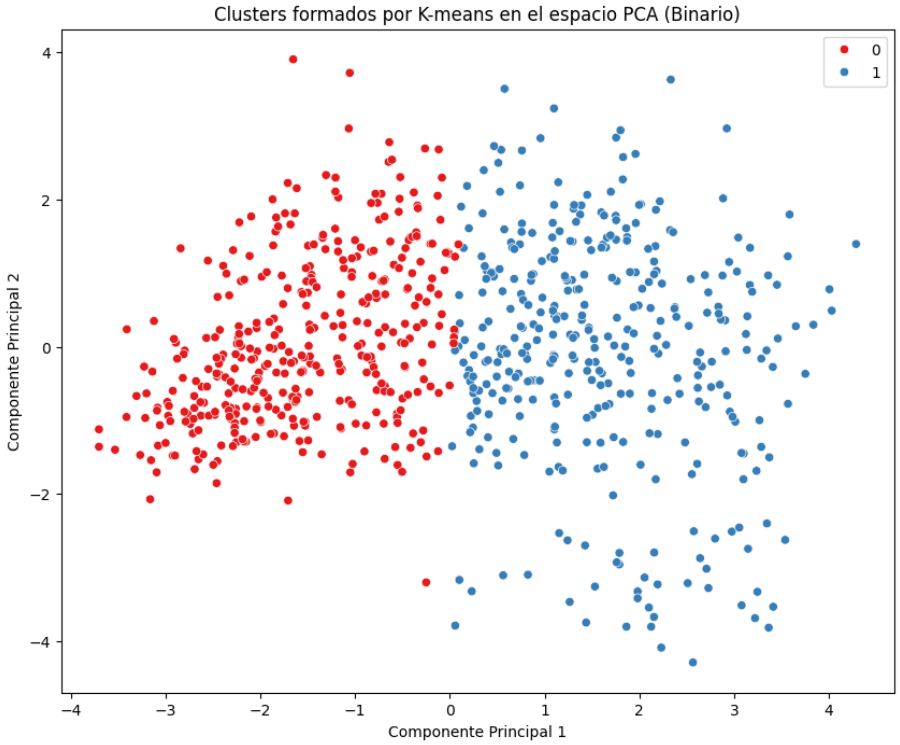
Clustering

La técnica de clúster es una herramienta de aprendizaje no supervisado que agrupa observaciones o puntos de datos en subconjuntos llamados clústeres o grupos. El objetivo principal es que las observaciones dentro de un mismo grupo sean más similares entre sí que a las observaciones en otros grupos. En el contexto de nuestro análisis de enfermedades cardíacas, aplicar un algoritmo de clustering nos permite identificar patrones ocultos en los datos sin utilizar la variable objetivo (es decir, sin saber de antemano si un paciente tiene o no la enfermedad).

Análisis de clúster

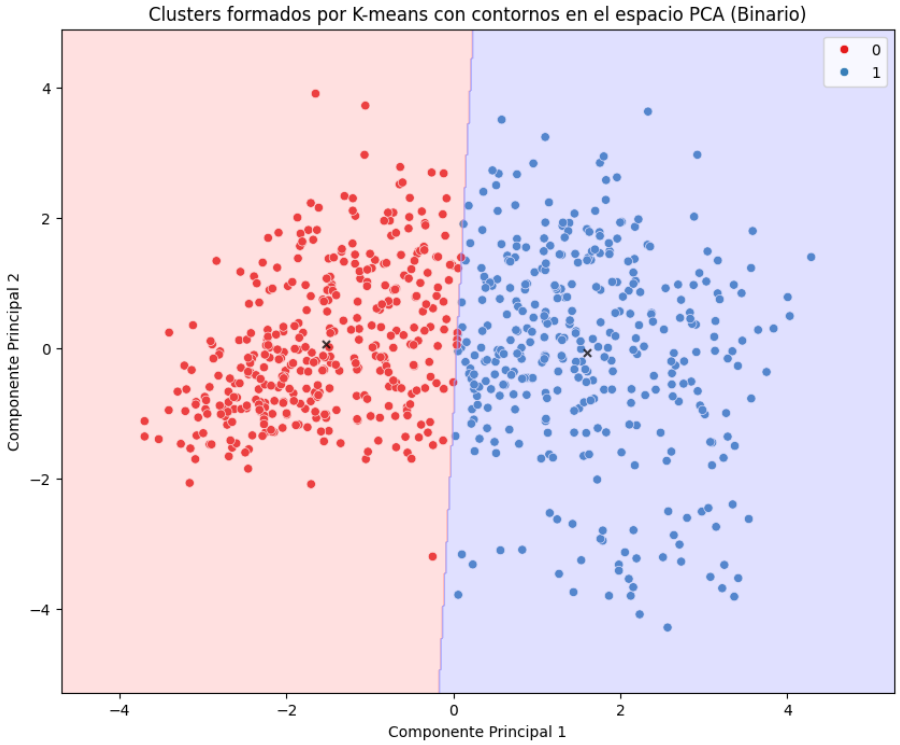
Existen varias técnicas de clustering, siendo una de las más comunes el algoritmo de K-means, que agrupa los datos en un número predefinido de clústeres, minimizando la variación interna dentro de cada grupo.

En este caso, el clustering es útil para explorar si existen patrones o subgrupos en el dataset que puedan corresponder, por ejemplo, a diferentes perfiles de pacientes en función de los factores de riesgo presentes. Esto puede ofrecer información adicional que puede no ser evidente en un análisis supervisado tradicional.



En este gráfico, los puntos están coloreados en función de los dos clústeres formados por K-means. Cada punto representa una observación del dataset proyectada en el espacio de dos dimensiones definido por las primeras dos componentes principales que tomamos del PCA en la sección anterior. Los puntos rojos pertenecen al clúster 0, y los puntos azules pertenecen al clúster 1.

Se puede observar que los puntos tienden a agruparse en dos regiones distintas. El clúster rojo (clúster 0) está en la parte izquierda del gráfico, mientras que el azul (clúster 1) se agrupa más hacia la derecha. Esto indica que el algoritmo K-means ha sido capaz de segmentar los datos en dos grupos distintos en función de las características representadas en las dos primeras componentes principales.



Este gráfico es similar al anterior, pero incluye contornos que marcan la frontera entre los dos clústeres. La región en rojo muestra el espacio asignado al clúster 0, mientras que la región azul corresponde al clúster 1. Los contornos ayudan a visualizar claramente la separación entre los dos grupos, mostrando cómo K-means ha trazado una línea divisoria en el espacio de las componentes principales para separar ambos clústeres.

La frontera entre los clústeres es lineal, lo que es característico del algoritmo K-means, ya que siempre busca minimizar la distancia euclidiana entre los puntos y los centroides de cada clúster. En este caso, se observa una buena separación entre los grupos, aunque con algunos puntos en la frontera que podrían considerarse casos ambiguos o difíciles de clasificar correctamente.

Conclusión

En este trabajo se implementaron técnicas de aprendizaje supervisado y no supervisado para abordar el diagnóstico de enfermedades cardíacas a partir de datos clínicos. El análisis exploratorio de los datos permitió identificar patrones importantes y limpiar el dataset para garantizar la robustez del modelado posterior. En la parte supervisada, la Regresión Logística fue seleccionada como el mejor modelo, obteniendo un buen equilibrio entre precisión y recall, con un AUC-ROC de 0.909, lo que evidencia su capacidad para discriminar de manera efectiva entre pacientes con y sin la enfermedad.

En cuanto al análisis no supervisado, el uso de PCA y K-means permitió reducir la dimensionalidad del dataset y agrupar las observaciones en clústeres claramente diferenciados, ofreciendo una interpretación adicional sobre los patrones presentes en los datos. Este análisis sugiere que, más allá de la clasificación supervisada, existen subgrupos naturales en el conjunto de datos que podrían ser utilizados para futuras investigaciones o personalización de tratamientos.

Finalmente, el modelo supervisado logrado presenta un rendimiento adecuado para la tarea de diagnóstico, pero hay margen para futuras mejoras, como la exploración de técnicas más avanzadas de rebalanceo de clases o el uso de otros modelos no lineales. Los resultados del clustering también sugieren que hay espacio para una interpretación más profunda de los grupos formados en el espacio PCA, lo cual podría conducir a nuevas hipótesis respecto a las características usadas en este trabajo.

En general, este trabajo muestra la efectividad del uso de aprendizaje estadístico en problemas del mundo real como el diagnóstico de enfermedades cardiacas y resalta la importancia de combinar técnicas de modelado para obtener una visión integral de los datos.

Bibliografía

Allison, P. (2021, July 22). When can you safely ignore multicollinearity? *Statistical Horizons*. https://statisticalhorizons.com/multicollinearity

Chan, J. Y.-L., Leow, S. M. H., Bea, K. T., Cheng, W. K., Phoong, S. W., Hong, Z.-W., & Chen, Y.-L. (2022). Mitigating the multicollinearity problem and its machine learning approach: A review. *Mathematics, 10*(1283). https://doi.org/10.3390/math10081283

Detrano, R., Janosi, A., Steinbrunn, W., Pfisterer, M., Schmid, J. J., Sandhu, S., Guppy, K. H., Lee, S., & Froelicher, V. (1989). International application of a new probability algorithm for the diagnosis of coronary artery disease. *The American Journal of Cardiology, 64*(5), 304–310. https://doi.org/10.1016/0002-9149(89)90524-9

Göcs, L., & Johanyák, Z. C. (2023). Feature selection with weighted ensemble ranking for improved classification performance on the CSE-CIC-IDS2018 dataset. *Computers, 12*(147). https://doi.org/10.3390/computers12080147

Janosi, A., Steinbrunn, W., Pfisterer, M., & Detrano, R. (1989). Heart Disease [Dataset]. *UCI Machine Learning Repository*. https://doi.org/10.24432/C52P4X

Kurita, T. (2020). Principal component analysis (PCA). In *Computer Vision*. Springer, Cham. https://doi.org/10.1007/978-3-030-03243-2_649-1