Heart Disease Predictor

## Enunciado del Problema

El problema que abordaremos es el desarrollo de un modelo predictivo capaz de determinar el riesgo de enfermedades cardiovasculares en individuos. Utilizaremos variables como la edad, género, presión arterial, y otras medidas clínicas para predecir si un individuo tiene un alto riesgo de desarrollar este tipo de enfermedades.

Objetivos

Realizar un análisis exploratorio completo de los datos para entender las distribuciones, correlaciones y posibles anomalías en las variables.

Preparar los datos para el modelado mediante la limpieza, normalización y codificación de las variables necesarias.

Seleccionar la métrica de evaluación adecuada para medir la efectividad de los modelos de machine learning, considerando la precisión, sensibilidad, especificidad y el área bajo la curva ROC como posibles candidatos.

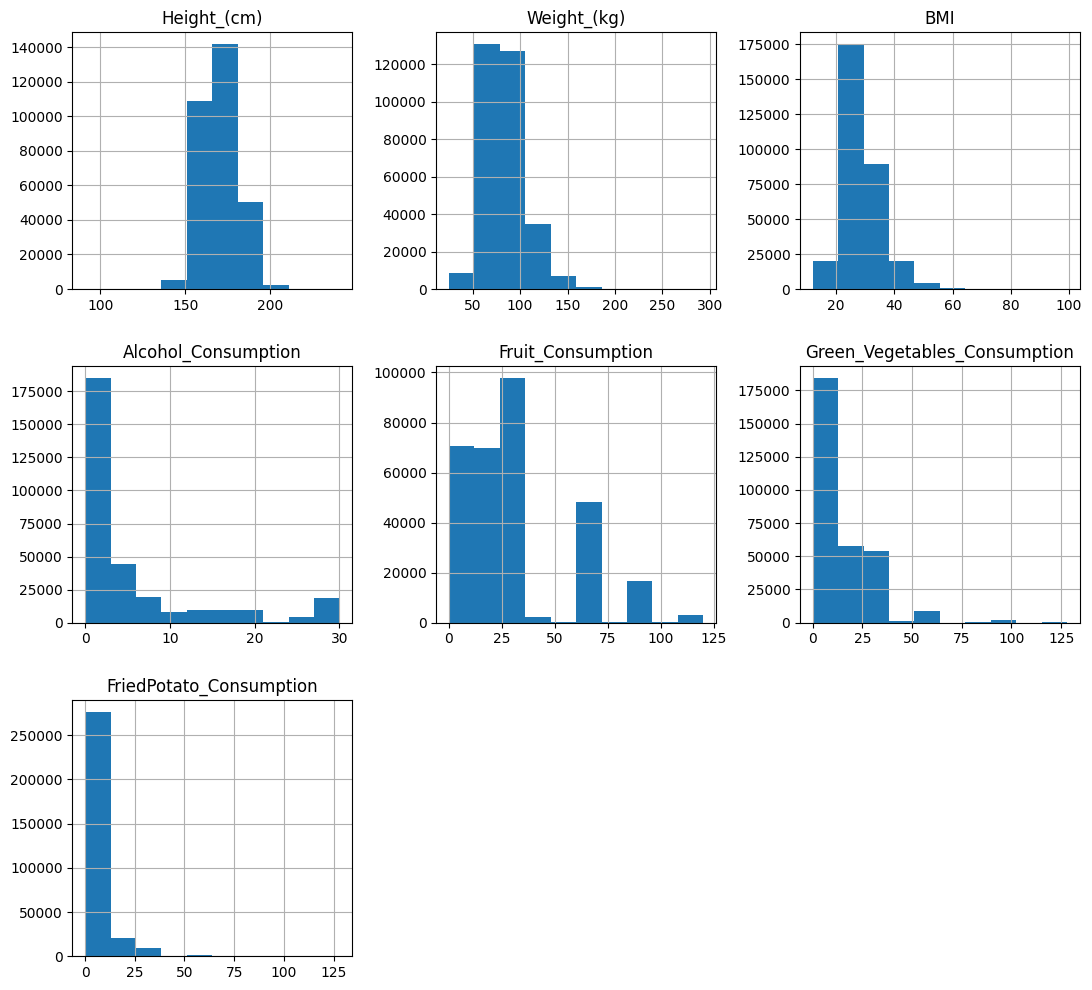
Comparar diferentes modelos de machine learning para identificar el más adecuado en términos de precisión y robustez. Experimentaremos con al menos tres modelos diferentes: regresión logística, random forest y XGBoost.

Comunicar efectivamente los resultados obtenidos a través de visualizaciones y un informe detallado que documente todo el proceso y las conclusiones.

## Análisis Exploratorio de Datos (EDA)

Vamos a cargar el dataset y empezar con el EDA, explorando las primeras filas del dataset, estadísticas descriptivas y visualizaciones de las distribuciones de las variables.

| **count** | **mean** | **std** | **min** | **25%** | **50%** | **75%** | **max** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Height\_(cm) | 308854.0 | 170.615249 | 10.658026 | 91.00 | 163.00 | 170.00 | 178.00 | 241.00 |
| Weight\_(kg) | 308854.0 | 83.588655 | 21.343210 | 24.95 | 68.04 | 81.65 | 95.25 | 293.02 |
| BMI | 308854.0 | 28.626211 | 6.522323 | 12.02 | 24.21 | 27.44 | 31.85 | 99.33 |
| Alcohol\_Consumption | 308854.0 | 5.096366 | 8.199763 | 0.00 | 0.00 | 1.00 | 6.00 | 30.00 |
| Fruit\_Consumption | 308854.0 | 29.835200 | 24.875735 | 0.00 | 12.00 | 30.00 | 30.00 | 120.00 |
| Green\_Vegetables\_Consumption | 308854.0 | 15.110441 | 14.926238 | 0.00 | 4.00 | 12.00 | 20.00 | 128.00 |
| FriedPotato\_Consumption | 308854.0 | 6.296616 | 8.582954 | 0.00 | 2.00 | 4.00 | 8.00 | 128.00 |



1. Height (cm): La distribución de la altura parece ser aproximadamente normal, con la mayoría de los individuos agrupados alrededor de una altura que podría estar en el rango de 150 a 175 cm. Hay pocos individuos con alturas extremadamente bajas o altas.

2. Weight (kg): La distribución del peso muestra una tendencia similar a la de la altura, con un pico que sugiere que la mayoría de los individuos tienen un peso que podría estar alrededor de 50 a 100 kg. La cola hacia la derecha indica que hay menos individuos con pesos más altos.

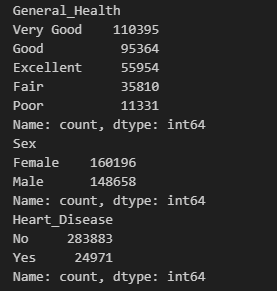
3. BMI (Body Mass Index): El Índice de Masa Corporal también muestra una distribución con un pico prominente, probablemente en el rango de 20 a 30, lo que indica un rango de peso normal a sobrepeso para la mayoría de los individuos. La distribución tiene una cola larga hacia la derecha, indicando la presencia de individuos con valores de BMI más altos, posiblemente obesos.

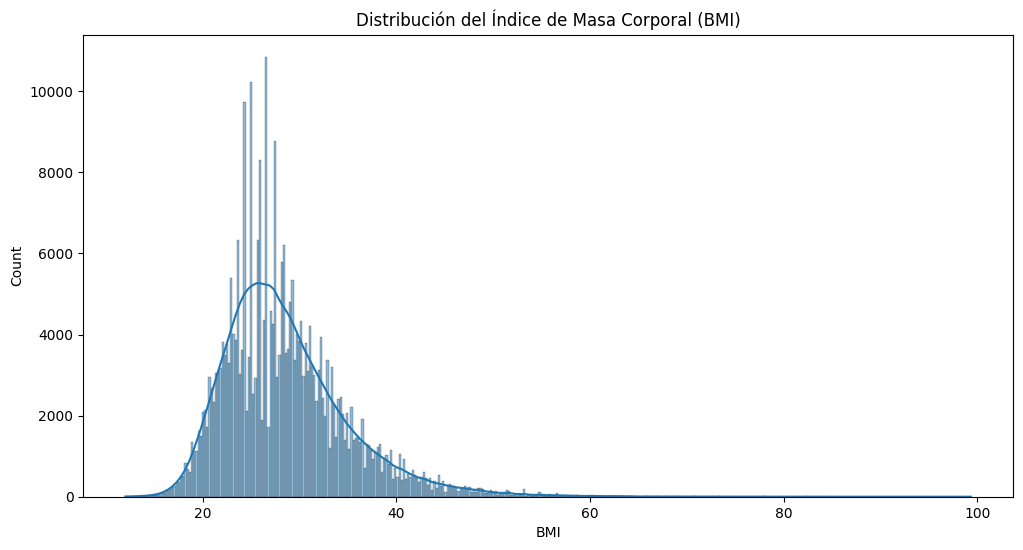
4. Alcohol\_Consumption: Esta variable parece estar sesgada a la derecha, con la mayoría de los individuos consumiendo poco o ningún alcohol, y muy pocos individuos consumiendo grandes cantidades.

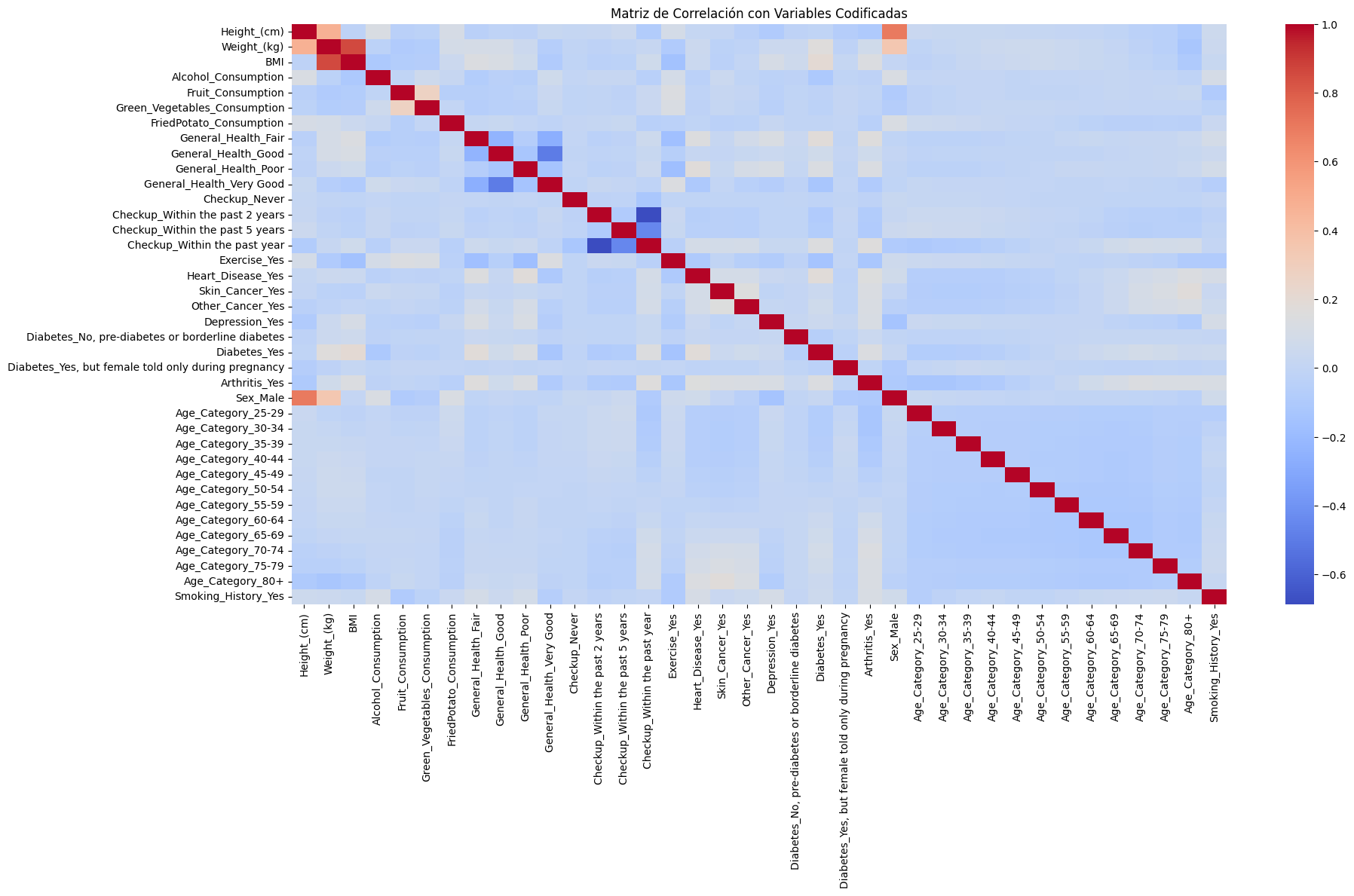
5. Fruit\_Consumption: El consumo de frutas varía más que el consumo de alcohol, con picos en valores bajos y luego en un rango que podría ser alrededor de 100 unidades (la unidad no está definida en el eje x). Esto podría indicar que algunos individuos consumen frutas con mucha frecuencia o en grandes cantidades.

6. Green\_Vegetables\_Consumption: El consumo de vegetales verdes tiene un patrón similar al consumo de frutas, con un pico en un valor bajo y otro pico alrededor de 100 unidades. Esto sugiere dos grupos de individuos: aquellos que consumen pocas verduras verdes y aquellos que consumen una cantidad considerablemente mayor.

7. FriedPotato\_Consumption: La mayoría de los individuos consume una cantidad muy baja de papas fritas, con un pico muy alto cerca del valor más bajo. Es probable que la frecuencia del consumo de papas fritas sea generalmente baja entre la población del dataset.







En la gráfica, vemos una matriz de correlación que presenta las relaciones entre variables que han sido codificadas, probablemente para prepararlas para un análisis de aprendizaje automático. Las correlaciones varían entre -1 y 1, donde 1 indica una correlación positiva perfecta, -1 indica una correlación negativa perfecta, y valores cercanos a 0 indican poca o ninguna correlación lineal entre las variables.

En el contexto de un análisis predictivo para enfermedades cardiovasculares, aquí están algunos puntos destacados:

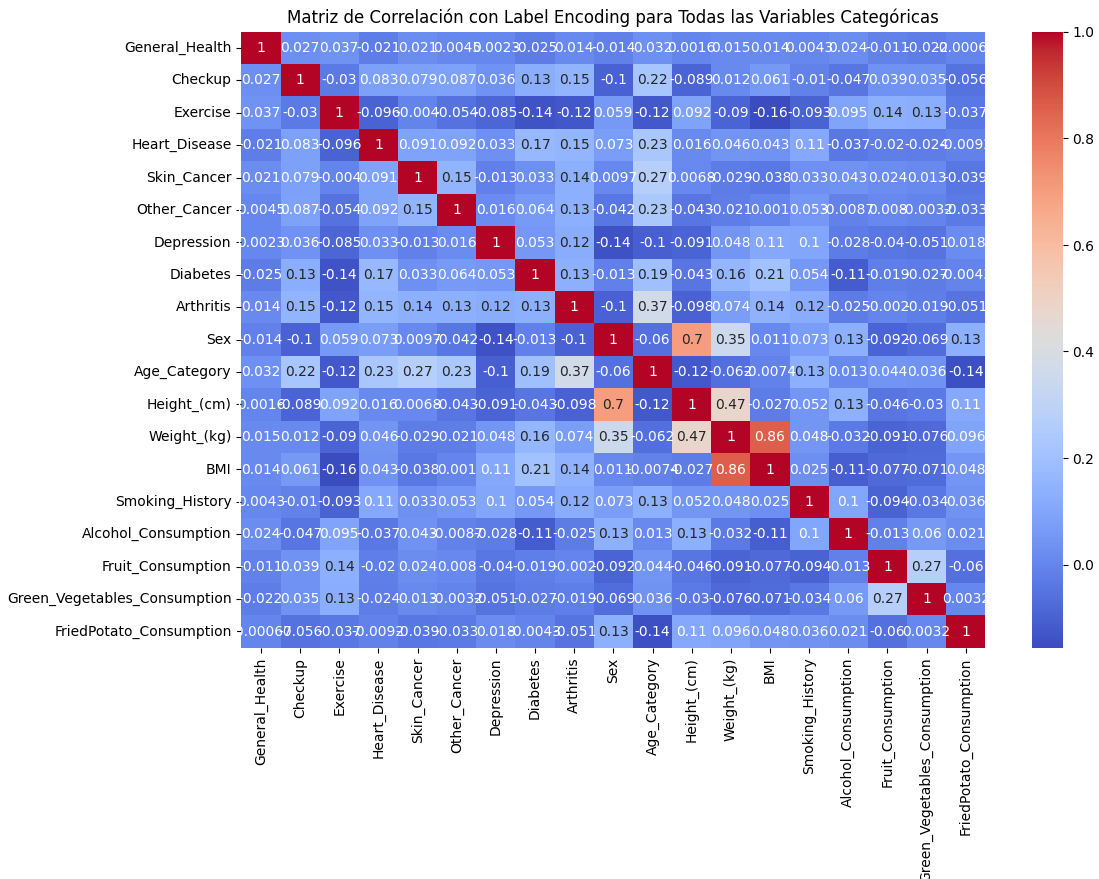
\* Edad y enfermedades: Hay bloques de correlación positiva más oscuros que indican una relación entre categorías de edad avanzada y la presencia de varias condiciones, como enfermedades cardiacas y artritis. Esto es esperado, ya que el riesgo de muchas enfermedades aumenta con la edad.

\* Estado de salud general: Las variables que representan el auto-reportaje del estado de salud general (por ejemplo, General\_Health\_Fair, General\_Health\_Good, etc.) parecen tener una correlación negativa con las condiciones de salud específicas. Esto podría significar que aquellos que reportan un estado de salud mejor tienden a tener menos diagnósticos de enfermedades.

\* IMC y peso: El Índice de Masa Corporal (BMI) y el peso (Weight\_(kg)) están altamente correlacionados, lo que tiene sentido dado que el BMI se calcula a partir del peso y la altura. El BMI podría tener una relación significativa con las condiciones de salud y es probable que sea una variable importante para el modelado predictivo de enfermedades cardiovasculares.

Comportamientos de salud: Hay algunas correlaciones leves entre variables como la historia de fumar y la salud cardíaca, lo que sugiere posibles factores de riesgo o indicadores de enfermedades cardiovasculares.

\* Correlaciones entre enfermedades: Existe una correlación entre diferentes condiciones de salud (por ejemplo, enfermedades del corazón y otros tipos de problemas de salud como diabetes y cáncer de piel)



Análisis bivariado

Analizando el análisis bivariado para variables numéricas podemos sacar ciertas conclusiones.

\* Relación entre Heart Disease y Height (cm)

\* Distribución de la altura: Se observa cómo varía la altura de las personas con y sin enfermedad cardíaca.

\* En cuanto a las medianas (la línea dentro del boxplot) presentan diferencias ligeras estando la mediana del grupo que no presenta ninguna complicación en torno a los 170 cm y en los pacientes que sí presentan enfermedades cardíacas ligeramnete superior. No se cree que esto explique ninguna causalidad de la variable objetivo si no que se puede tratar de un sesgo en los propios datos.

\* Outliers en la altura: Los puntos fuera de las "bigotes" (whiskers) del boxplot indican valores atípicos, lo cual podría sugerir que hay casos extremos en la altura que podrían estar asociados con la enfermedad. En este caso, el mayor número de outliers presentes se encuentran en el grupo que no padecen problemas de corazón.

\* Relación entre Heart Disease y Weight (kg)

\* Distribución del peso: se compara el peso de individuos con y sin enfermedad cardíaca. Las cajas o boxes presentes son de un tamaño similar, al igual que la mediana dentro de esta que está en torno a unos ochenta o noventa kilogramos. En cuanto al número de outliers, el grupo que presenta una enfermedad de corazón identifica menos outliers que el que no. Se puede justificar esta diferencia por un sesgo menor en datos.

\* Relación entre Heart Disease y BMI

\* Distribución del BMI: El índice de masa corporal (BMI) es un factor importante relacionado con las enfermedades cardíacas. De manera muy similar al caso anterior, tanto el tamaño de los boxes como la posición de la mediana y de los "bigotes" están en rangos muy similares.

\* Relación entre Heart Disease y Alcohol\_Consumption

\* Consumo de alcohol: Este gráfico podría mostrar si las personas con enfermedad cardíaca tienden a consumir más o menos alcohol que aquellas sin la enfermedad. Si no hay mucha superposición entre los grupos, el consumo de alcohol podría estar asociado con la enfermedad.

\* Relación entre Heart Disease y Fruit\_Consumption

\* Consumo de frutas: Este análisis podría revelar si existe una relación entre la cantidad de frutas consumidas y la presencia de enfermedades cardíacas, algo que se espera sea un factor de estilo de vida saludable.

\* Relación entre Heart Disease y Green\_Vegetables\_Consumption

\* Consumo de vegetales verdes: Similar al consumo de frutas, este boxplot puede sugerir si el consumo de verduras tiene alguna relación con la salud cardíaca.

\* Relación entre Heart Disease y FriedPotato\_Consumption

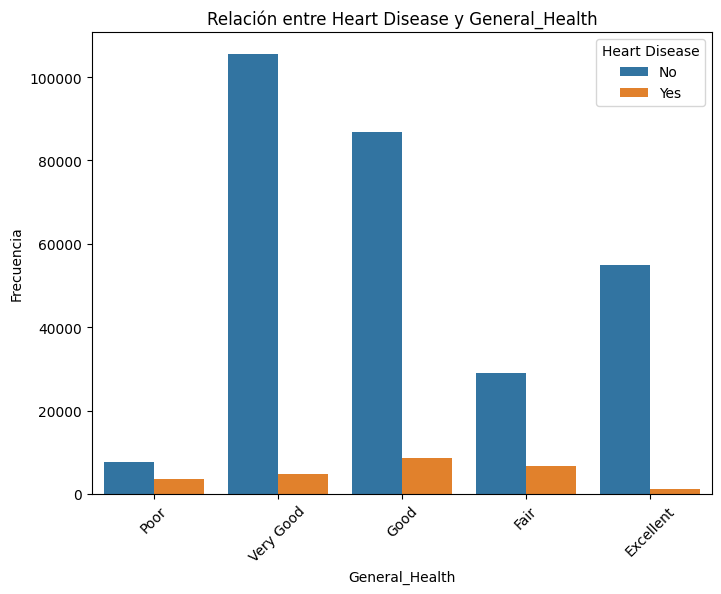
\* Consumo de patatas fritas: Este gráfico podría mostrar la relación entre el consumo de alimentos potencialmente no saludables como las patatas fritas y la enfermedad cardíaca.

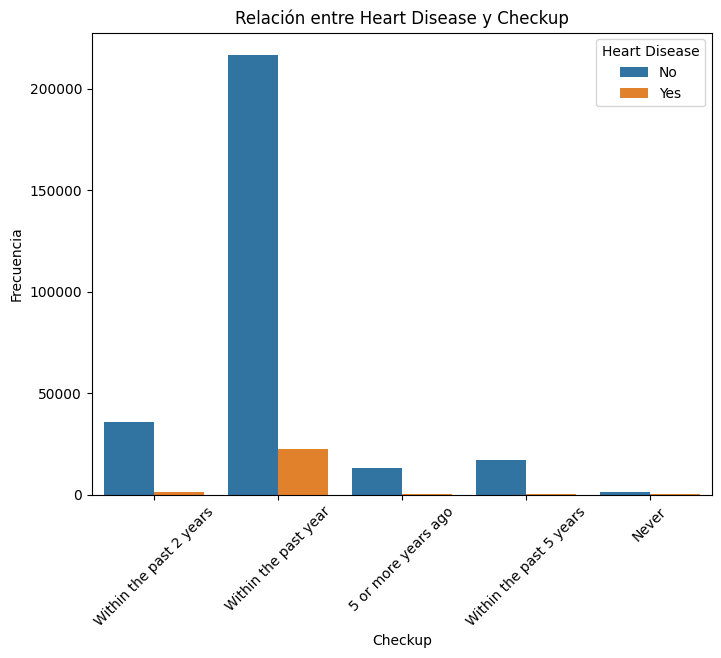
Para todas estas gráficas, algunas consideraciones comunes incluyen:

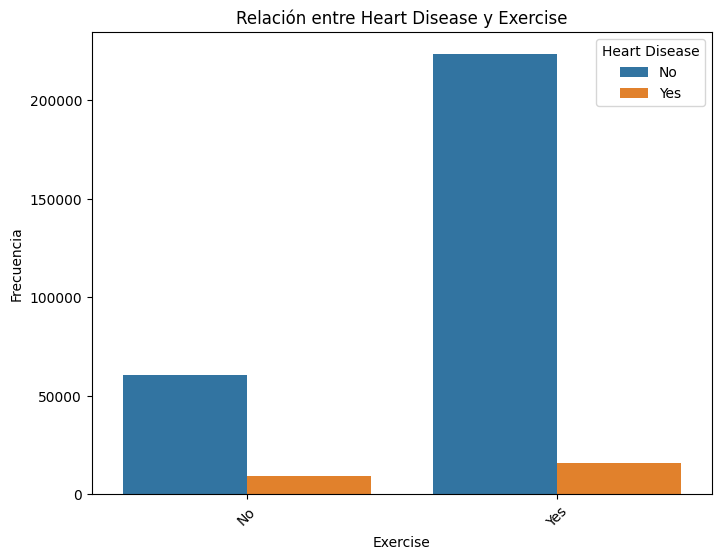
Medianas y cuartiles: La mediana es la línea central en cada caja, y las alturas de las cajas representan los cuartiles (25% y 75% percentiles). Si las medianas están en niveles muy diferentes, eso indica una diferencia central entre los grupos.

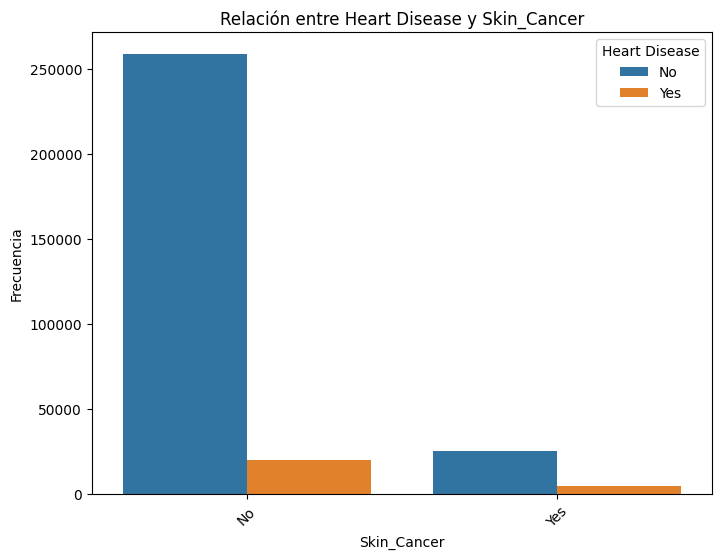
Bigotes: Los extremos de los bigotes representan el rango típico de los datos, y los puntos fuera de estos son considerados outliers.

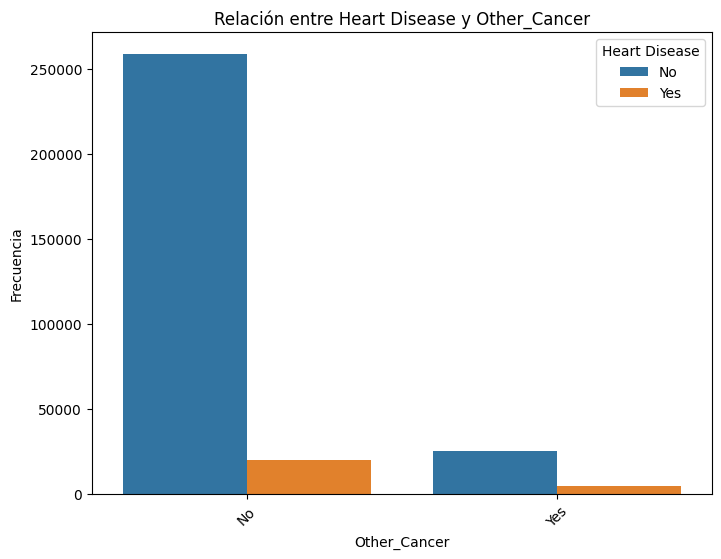
Ancho de las cajas: El ancho de la caja indica la dispersión entre los cuartiles medianos, lo cual puede hablar de la variabilidad de los datos en ese grupo.

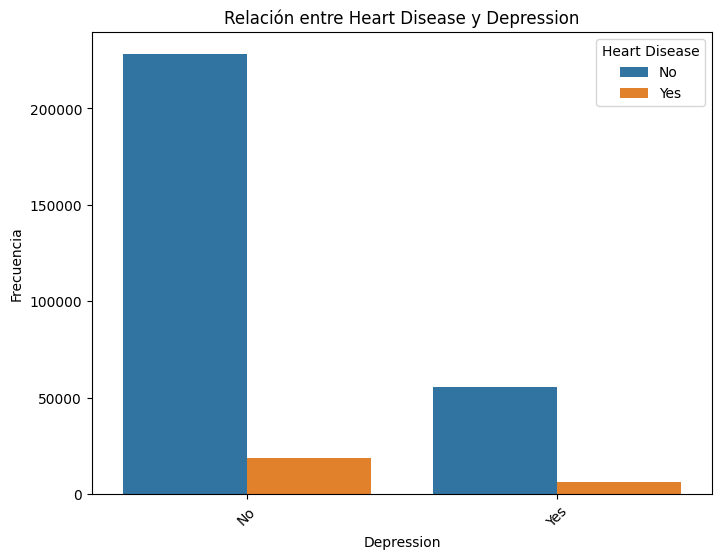


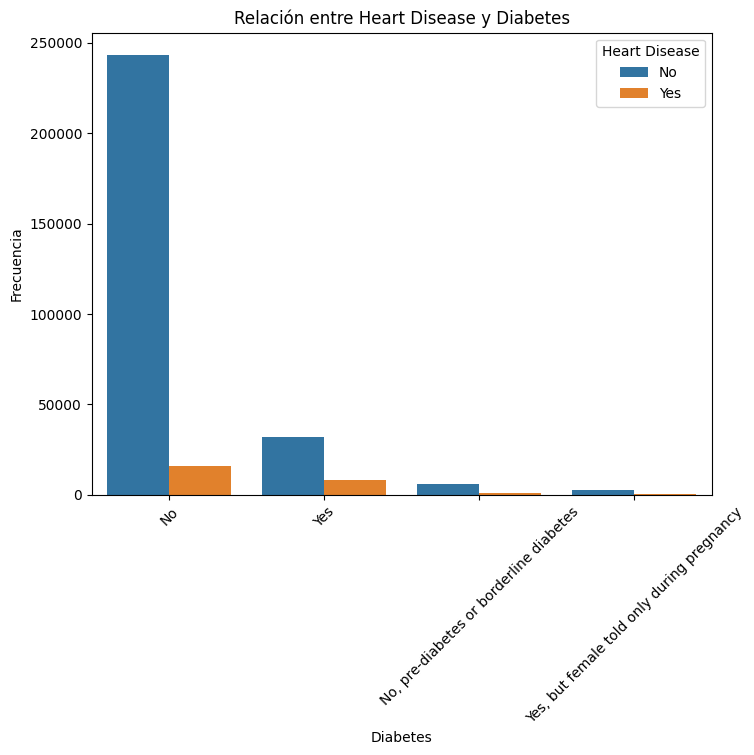


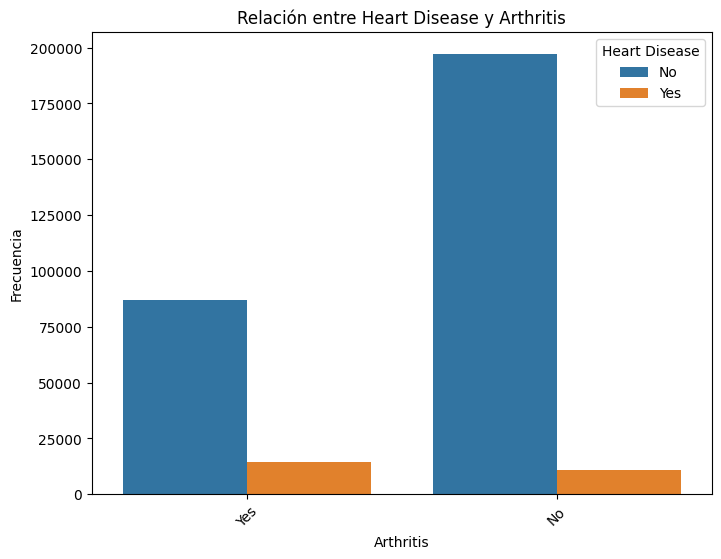


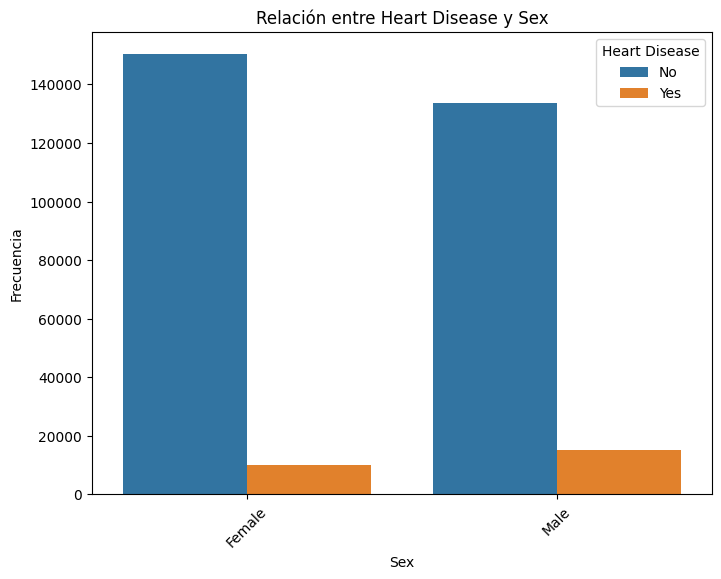


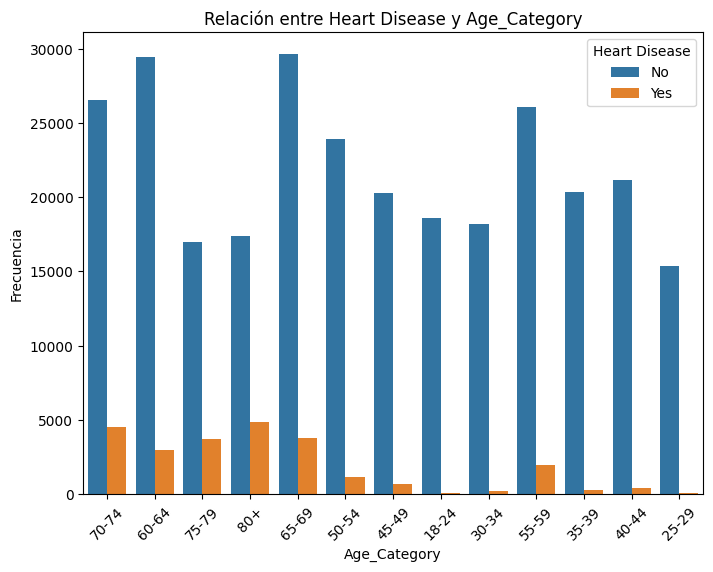


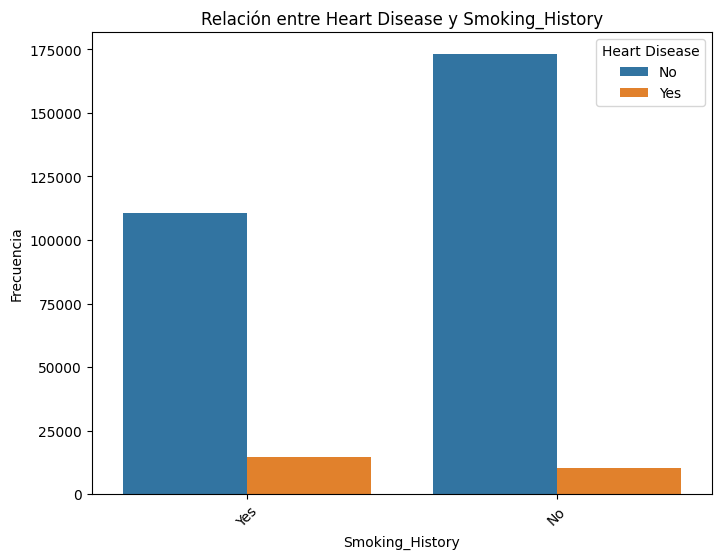




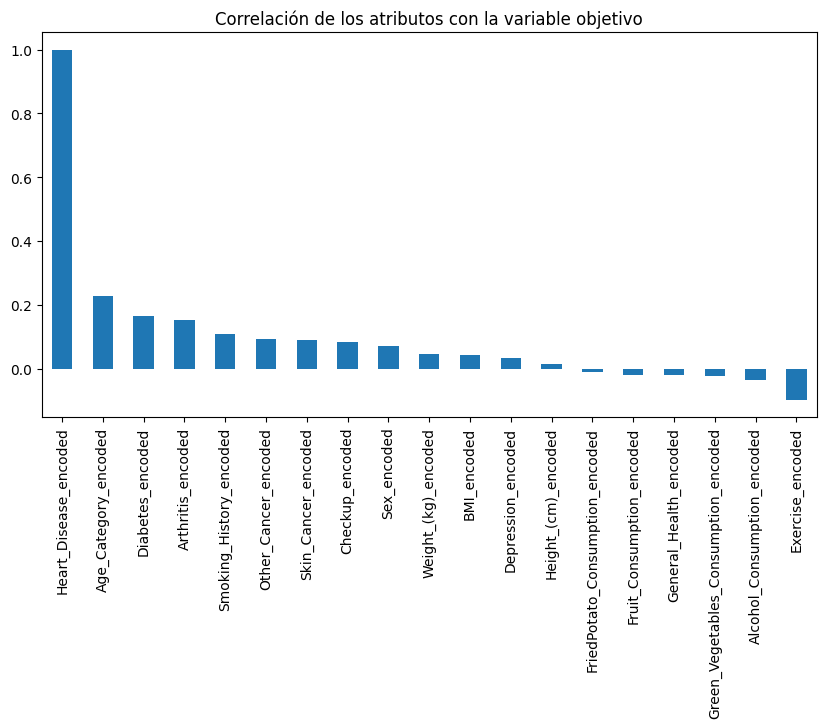








El coeficiente de correlación de Pearson varía entre -1 y 1, donde 1 significa una correlación positiva perfecta, -1 significa una correlación negativa perfecta y 0 significa que no hay correlación.



Ingeniería de Características

Para la preparación del dataset, abordaremos los siguientes pasos:

Gestión de valores faltantes: Imputaremos o eliminaremos filas/columnas según sea necesario.

Codificación de variables categóricas: Usaremos técnicas como One-Hot Encoding o Label Encoding.

Normalización/Estandarización de variables: Estandarizaremos las variables numéricas para tener un rango común.

Alternativas como el Ordinal Encoding o el uso de técnicas de embedding pueden ser más adecuadas en ciertas circunstancias. Ordinal Encoding asigna un valor entero único a cada categoría de manera ordenada. Aunque es más compacto y evita el problema de la alta dimensionalidad inherente al One-Hot Encoding, puede introducir una relación de orden artificial entre las categorías que no existe naturalmente, lo cual podría ser inapropiado para algunos modelos.

Por otro lado, los métodos de embedding, comúnmente empleados en el aprendizaje profundo, pueden aprender una representación numérica densa de las categorías, lo que permite capturar relaciones potencialmente más complejas entre ellas.

Hay que volver a clarificar que el Ordinal Encoding ordena las categorías numéricamente, lo cual puede ser un problema si el modelo interpreta esta ordenación como una relación de magnitud.

Los resultados muestran que aproximadamente el 91.9% de las muestras pertenecen a la clase 0 y solo el 8.1% a la clase 1. Este tipo de desbalance es significativo y puede tener un impacto considerable en el rendimiento de muchos modelos de machine learning, especialmente aquellos que asumen (explícita o implícitamente) que las clases están balanceadas.

Submuestreo combinando SMOTE y Tomek Links

pipeline\_resampling = ImbPipeline([

('scaler', MinMaxScaler()),

('smote', SMOTE(sampling\_strategy='minority')),

('tomek', TomekLinks(sampling\_strategy='majority'))

])

# Aplicar el pipeline al conjunto de entrenamiento

X\_resampled, y\_resampled = pipeline\_resampling.fit\_resample(X\_train, y\_train)

pipeline\_resampling = ImbPipeline([

('scaler', MinMaxScaler()),

('smote', SMOTE(sampling\_strategy='minority')),

('tomek', TomekLinks(sampling\_strategy='majority'))

])

# Aplicar el pipeline al conjunto de entrenamiento

X\_resampled, y\_resampled = pipeline\_resampling.fit\_resample(X\_train, y\_train)

Aplicación de Modelos

A continuación, definiremos y entrenaremos diferentes modelos, compararemos su rendimiento y seleccionaremos el más adecuado. Después de completar estos pasos iniciales, podemos proceder a la evaluación y mejora de los modelos seleccionados.

\* Gestión de valores faltantes: Imputaremos o eliminaremos filas/columnas según sea necesario. Se utiliza para ello 'SimpleImputer' para ambas características numéricas y categóricas, para las numéricas se rellena con la mediana mientras que para las categóricas se utilizan los valores más frecuentes.

\* Codificación de variables categóricas: Usaremos técnicas como One-Hot Encoding o Label Encoding. OneHotEncoder se centra en crear nuevas columnas indicando presencia de un valor con un 1 y la ausencia con un 0. La opción de handle\_unknown='ignore' permite manejar categorías no vistas durante el entrenamiento.

\* Normalización/Estandarización de variables: Estandarizaremos las variables numéricas para tener un rango común. Se utiliza para esto el StandardScaler, es crucial para utilizar modelos que asumen que todas las características tienen aproximadamente la misma escala, como la regresión lineal o métodos en los que sean importante las distancias.

\* Resampling con SMOTE:

\* SMOTE (Synthetic Minority Over-sampling Technique) es una técnica avanzada de sobremuestreo que ayuda a abordar el desequilibrio de clases creando ejemplos sintéticos de la clase minoritaria en lugar de simplemente hacer un sobremuestreo con reemplazo. Esto se hace seleccionando ejemplos que están cerca en el espacio de características, trazando una línea entre los ejemplos en el espacio de características y generando ejemplos nuevos a lo largo de esa línea.

\* sampling\_strategy='minority' indica que solo la clase minoritaria será sobremuestreada para equiparar su número al de la clase mayoritaria.

\* Submuestreo con TomekLinks:

\* TomekLinks es una técnica de submuestreo que identifica pares de ejemplos cercanos pero de clases opuestas, conocidos como enlaces Tomek, y elimina los ejemplos de la clase mayoritaria de cada par. Esto puede ayudar a hacer que el límite de decisión entre las clases sea más distinto.

\* sampling\_strategy='majority' indica que se submuestreará la clase mayoritaria, eliminando los ejemplos de la clase mayoritaria que son Tomek Links.

\* Escalado de Características:

\* MinMaxScaler es utilizado para normalizar las características escalándolas a un rango específico, generalmente 0 a 1. Esto es especialmente útil para algoritmos que dependen de la distancia entre los datos, como K-Nearest Neighbors, y puede mejorar el rendimiento de muchos otros modelos.

\* Calculando scale\_pos\_weight:

Esta es una técnica específica para algoritmos que soportan ponderación de clases, como XGBoost. El scale\_pos\_weight se utiliza para contrarrestar el desequilibrio entre las clases ajustando la importancia que se da a las clases minoritarias durante el entrenamiento. Aquí se calcula como la proporción del número de ejemplos en la clase mayoritaria dividido por el número de ejemplos en la clase minoritaria, usando los datos de entrenamiento (y\_train).

Naive Bayes

Mejores parámetros:

{}

Mejor puntuación:

0.2920384443035796

precision recall f1-score support

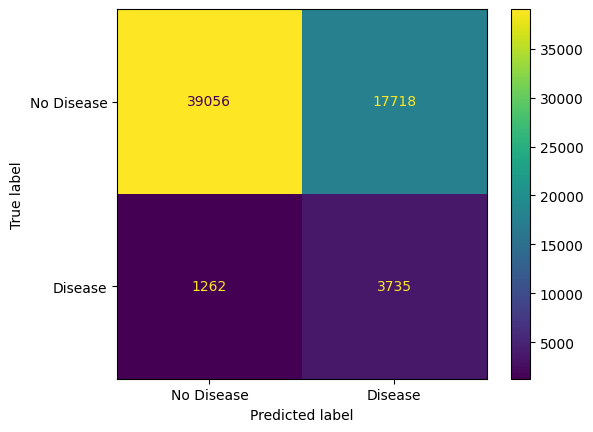
0.0 0.97 0.69 0.80 56774

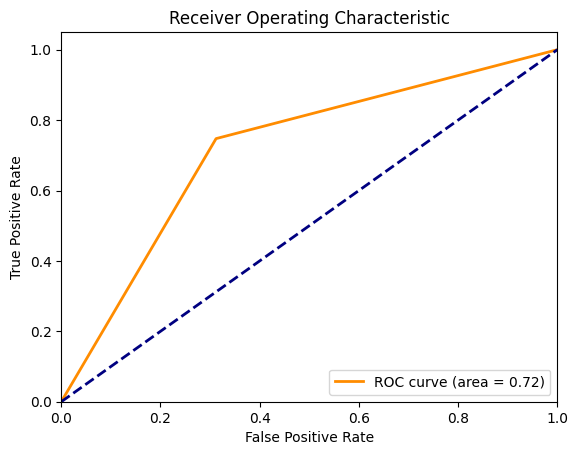
1.0 0.17 0.75 0.28 4997

accuracy 0.69 61771

macro avg 0.57 0.72 0.54 61771

weighted avg 0.90 0.69 0.76 61771





Fitting 5 folds for each of 20 candidates, totalling 100 fits

Mejores parámetros:

{'lr\_\_C': 1, 'lr\_\_solver': 'lbfgs'}

Mejor puntuación:

0.3022358421118973

precision recall f1-score support

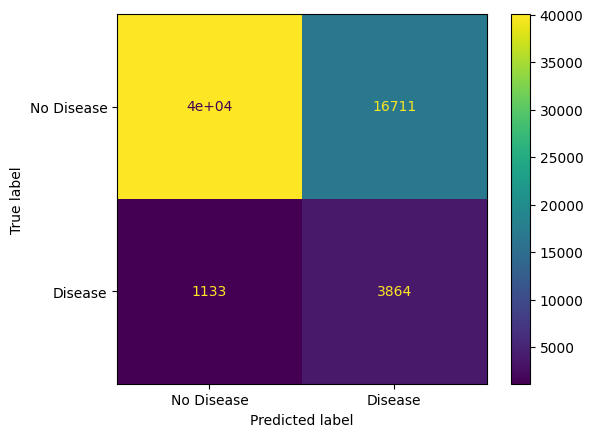
0.0 0.97 0.71 0.82 56774

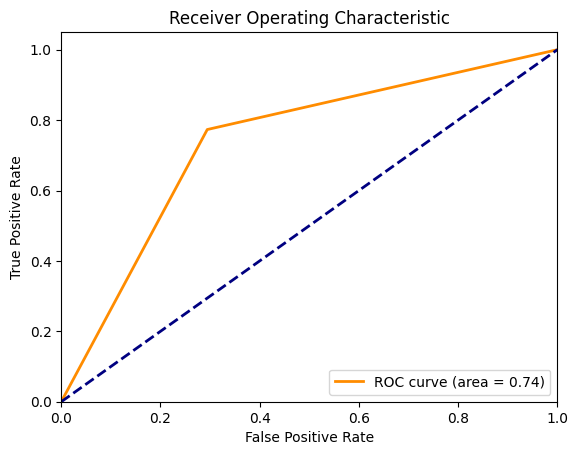
1.0 0.19 0.77 0.30 4997

accuracy 0.71 61771

macro avg 0.58 0.74 0.56 61771

weighted avg 0.91 0.71 0.78 61771





RandomForest

Fitting 5 folds for each of 27 candidates, totalling 135 fits

Mejores parámetros:

{'rf\_\_max\_depth': 20, 'rf\_\_min\_samples\_split': 10, 'rf\_\_n\_estimators': 200}

Mejor puntuación:

0.33162239291539236

precision recall f1-score support

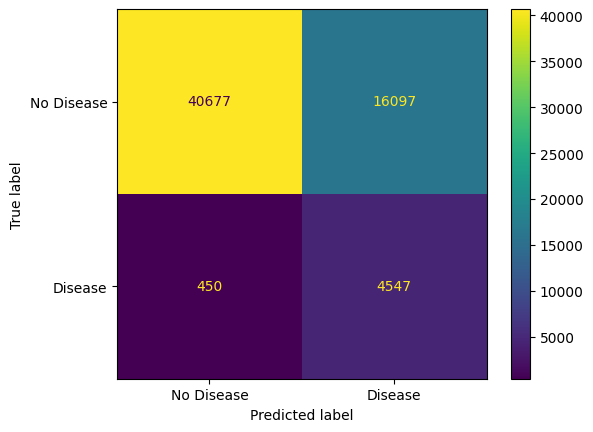
0.0 0.99 0.72 0.83 56774

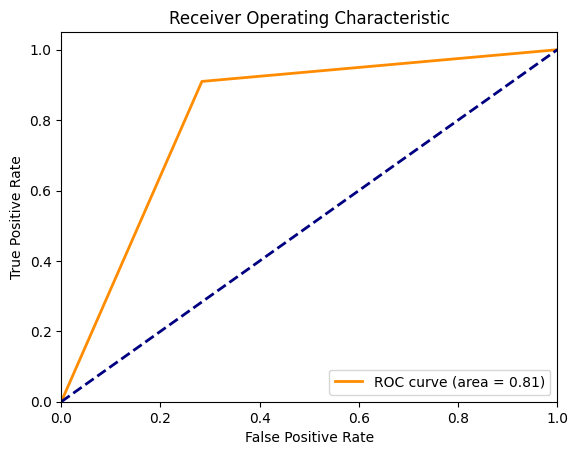
1.0 0.22 0.91 0.35 4997

accuracy 0.73 61771

macro avg 0.60 0.81 0.59 61771

weighted avg 0.93 0.73 0.79 61771





KNN

Fitting 5 folds for each of 16 candidates, totalling 80 fits

Mejores parámetros:

{'knn\_\_metric': 'manhattan', 'knn\_\_n\_neighbors': 3, 'knn\_\_weights': 'distance'}

Mejor puntuación:

0.16084231992235223

precision recall f1-score support

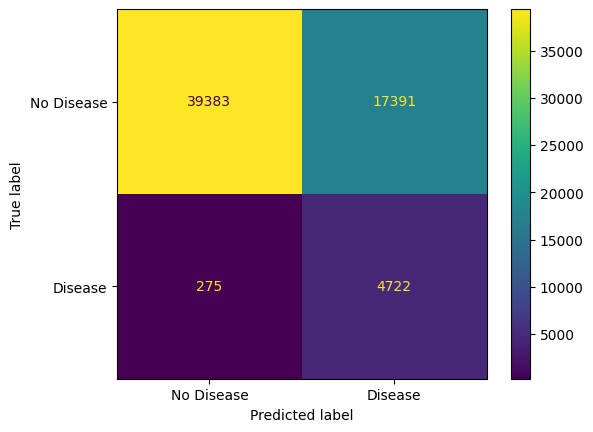
0.0 0.99 0.69 0.82 56774

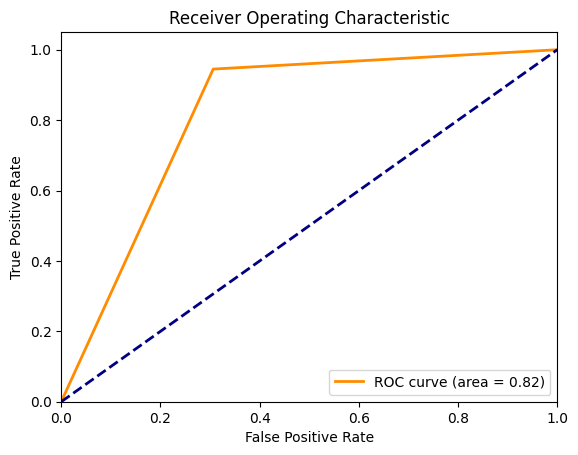
1.0 0.21 0.94 0.35 4997

accuracy 0.71 61771

macro avg 0.60 0.82 0.58 61771

weighted avg 0.93 0.71 0.78 61771





\* Modelo XGBoost Classifier:

\* XGBClassifier es un modelo basado en árboles de gradient boosting que es muy eficaz para una amplia gama de problemas de clasificación.

\* scale\_pos\_weight ajustado ayuda a tratar con clases desequilibradas.

\* eval\_metric='logloss' es la métrica de evaluación para optimizar durante el entrenamiento; logloss es útil para clasificación porque penaliza las clasificaciones incorrectas.

\* use\_label\_encoder=False se especifica para evitar advertencias ya que XGBoost puede manejar etiquetas codificadas sin necesidad de un codificador de etiquetas adicional.

\* ImbPipeline de imblearn:

A diferencia de Pipeline de scikit-learn, ImbPipeline está diseñado para trabajar con técnicas de remuestreo integradas como SMOTE y TomekLinks. Asegura que el remuestreo solo ocurra durante el entrenamiento del modelo y no durante el proceso de validación o prueba, evitando así el riesgo de filtración de información.

Fitting 5 folds for each of 27 candidates, totalling 135 fits

Mejores parámetros:

{'xgb\_\_learning\_rate': 0.2, 'xgb\_\_max\_depth': 5, 'xgb\_\_n\_estimators': 300}

Mejor puntuación:

0.09917669258151858

precision recall f1-score support

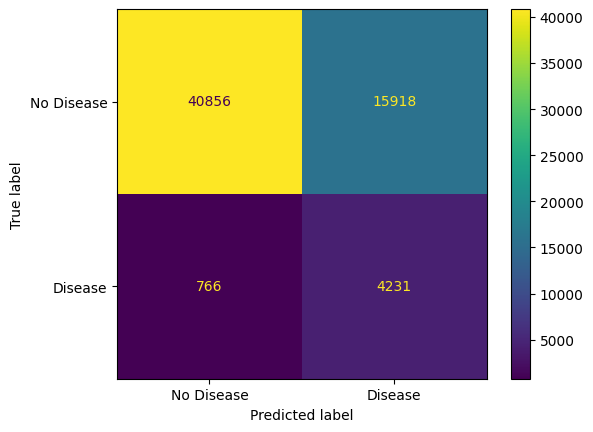
0.0 0.98 0.72 0.83 56774

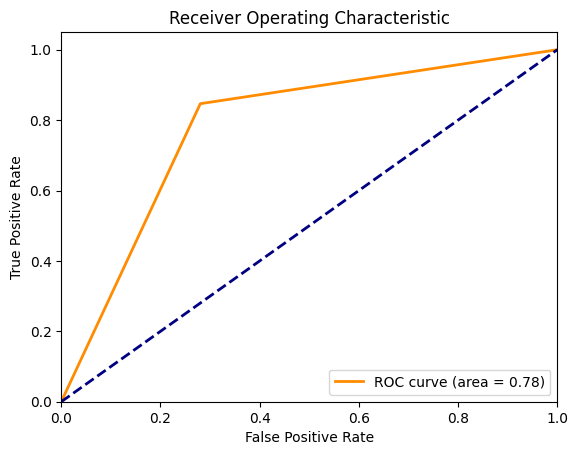
1.0 0.21 0.85 0.34 4997

accuracy 0.73 61771

macro avg 0.60 0.78 0.58 61771

weighted avg 0.92 0.73 0.79 61771





ADABOOST

Mejores parámetros:

{'clf\_\_learning\_rate': 1.0, 'clf\_\_n\_estimators': 200}

Mejor puntuación:

0.12074447385176508

precision recall f1-score support

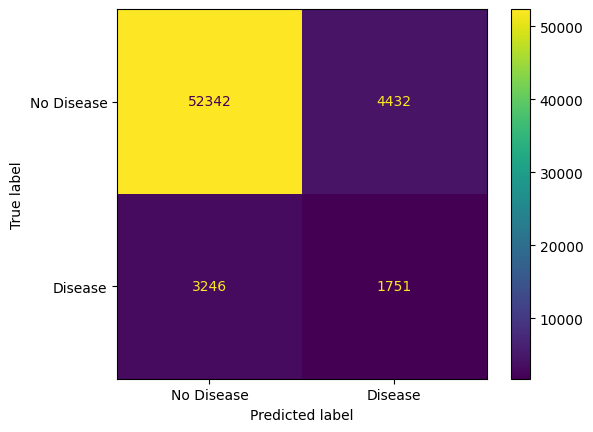
0.0 0.94 0.92 0.93 56774

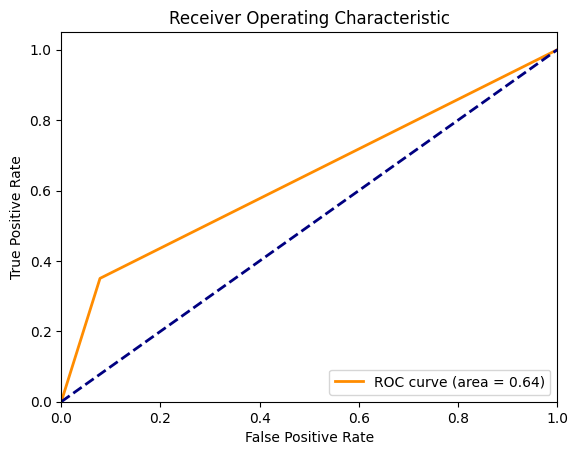
1.0 0.28 0.35 0.31 4997

accuracy 0.88 61771

macro avg 0.61 0.64 0.62 61771

weighted avg 0.89 0.88 0.88 61771





Redes Neuronales

Fitting 5 folds for each of 24 candidates, totalling 120 fits

Mejores parámetros:

{'mlp\_\_activation': 'tanh', 'mlp\_\_alpha': 0.01, 'mlp\_\_hidden\_layer\_sizes': (50, 50), 'mlp\_\_solver': 'adam'}

Mejor puntuación:

0.9192133820905424

precision recall f1-score support

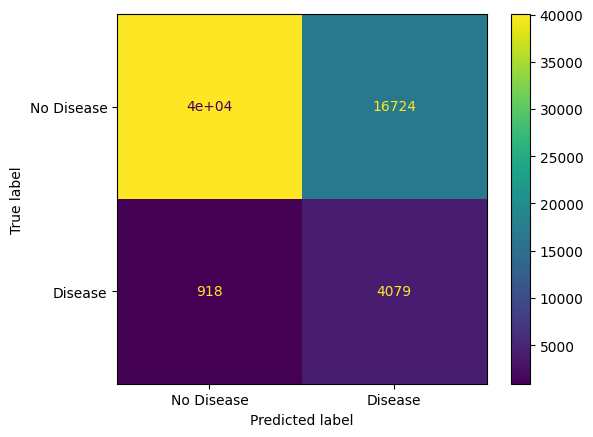
0.0 0.98 0.71 0.82 56774

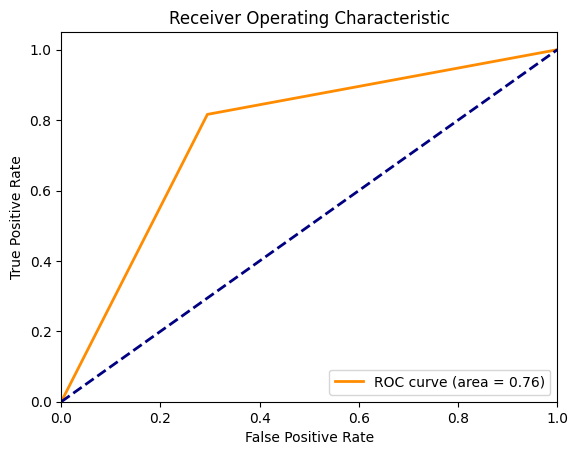
1.0 0.20 0.82 0.32 4997

accuracy 0.71 61771

macro avg 0.59 0.76 0.57 61771

weighted avg 0.91 0.71 0.78 61771





Plan de EDA para Nuestro Proyecto

Podemos combinar estos enfoques para desarrollar un plan de EDA exhaustivo:

Análisis univariado y bivariado: Examinar cada variable individualmente y en relación con la variable objetivo para entender las distribuciones y posibles correlaciones.

Visualización de datos: Usar gráficos como histogramas para variables continuas, gráficos de barras para categóricas, y mapas de calor para correlaciones.

Análisis de valores faltantes y outliers: Determinar cómo manejar los datos faltantes y los valores atípicos basados en su impacto en el análisis.

Ingeniería de características: Considerar la creación de nuevas variables basadas en el conocimiento del dominio y la correlación observada entre las variables.

Ejemplos Útiles:

K-Nearest Neighbors (KNN) - Útil para clasificación basada en la similitud de las características del paciente con casos históricos.

Support Vector Machine (SVM) - Efectivo para encontrar el hiperplano óptimo que separa las clases (enfermedad/no enfermedad).

Decision Trees - Proporciona una estructura clara de decisión basada en los atributos más influyentes.

Random Forests - Mejora la predicción de árboles de decisión a través de un enfoque de ensamble, ideal para manejar datasets con muchas características.

Enfoques Sugeridos:

XGBoost: Este algoritmo es excelente para tareas de clasificación y es conocido por su rendimiento superior, especialmente en datasets estructurados como el tuyo.

Logistic Regression: Ideal para problemas de clasificación binaria, como la predicción de la presencia o ausencia de enfermedades cardíacas.

Stacking: Combina múltiples modelos de predicción para mejorar la precisión. Sería útil si decides experimentar con un enfoque de ensamble.

Naive Bayes: Un enfoque rápido y eficaz para la clasificación, especialmente útil cuando las dimensiones del dataset son altas en relación con el tamaño de la muestra.