Predicción de riesgo de enfermedad cardíaca a partir de indicadores de salud y estilo de vida

INTEGRANTES:

Valentina Rendon y Miguel Moncada

1. ENTENDIMIENTO DEL NEGOCIO (Experto del Negocio)

1.1 DESCRIPCIÓN DEL NEGOCIO

Las enfermedades cardíacas representan una de las principales causas de muerte en el mundo. Se originan por múltiples factores de riesgo relacionados con el estilo de vida, la alimentación, el consumo de tabaco o alcohol, la actividad física, la edad y las condiciones médicas preexistentes.

El corazón es responsable de bombear sangre oxigenada a todo el cuerpo y su funcionamiento adecuado depende de una serie de hábitos saludables y del control de enfermedades como la hipertensión o la diabetes. Cuando el flujo sanguíneo hacia el corazón se ve comprometido pueden presentarse afecciones graves como infartos, insuficiencia cardíaca o arritmias.

Según los Centros para el Control y la Prevención de Enfermedades (CDC), las enfermedades cardíacas son la principal causa de muerte en Estados Unidos, siendo responsables de una de cada cinco muertes. Sin embargo muchos de los factores de riesgo son prevenibles mediante cambios en el estilo de vida y diagnósticos tempranos. Por ello el uso de modelos predictivos basados en datos de salud y hábitos personales permite identificar a las personas con mayor probabilidad de desarrollar enfermedades cardíacas facilitando estrategias preventivas y la toma de decisiones médicas oportunas.

1.2 DESCRIPCIÓN DEL PROBLEMA

Un número considerable de personas vive con factores de riesgo que aumentan su probabilidad de sufrir una enfermedad cardíaca sin ser conscientes de ello, el desafío consiste en detectar de forma temprana a los individuos con alto riesgo cardíaco utilizando información de salud general, condiciones médicas, indicadores físicos y hábitos de vida.

De esta forma se busca apoyar a los sistemas de salud pública en la priorización de intervenciones preventivas y en la mejora de la calidad de vida de la población.

1.3 OBJETIVOS DE LA MINERÍA

Objetivo general:

Desarrollar un modelo predictivo capaz de clasificar a las personas con base en su riesgo de padecer enfermedad cardíaca a partir de datos de salud, demografía y estilo de vida.

Objetivos específicos:

- Analizar el comportamiento de las variables relacionadas con la salud, hábitos y condiciones médicas para comprender su relación con la presencia de enfermedad cardíaca.
- Preparar y limpiar el conjunto de datos (heart_2020_cleaned.csv) aplicando procesos de detección y tratamiento de valores atípicos, manejo de nulos, codificación de variables categóricas y normalización de variables numéricas.
- Dividir los datos en conjuntos de entrenamiento (70%) y prueba (30%)**,** aplicando balanceo únicamente al conjunto de entrenamiento para corregir el desbalance de clases.
- Construir y evaluar múltiples modelos de clasificación aplicando al menos cuatro algoritmos de aprendizaje supervisado, por ejemplo: Regresión Logística**,** Random Forest, SVM y XGBoost**)** y tres métodos de ensamble (como Bagging, Boosting y Stacking**),** comparando su desempeño mediante métricas como Accuracy, Precision, Recall, F1 y ROC-AUC**.**
- Identificar las variables más influyentes en la predicción de enfermedad cardíaca mediante técnicas de interpretación de modelos como importancia de características o valores SHAP**.**
- Aplicar minería descriptiva (clustering**)** para segmentar a la población en grupos de riesgo y descubrir patrones comunes entre ellos.
- Implementar y desplegar el mejor modelo mediante un pipeline reproducible, acompañado de una interfaz interactiva desarrollada con Streamlit que permita realizar predicciones en tiempo real.

1.4 DISEÑO DE SOLUCIÓN

Problema	Tipo de Minería	Tipo de aprendizaje	Requerimiento datos	Métodos	Evaluación
Clustering (Segmentación de perfiles de riesgo)	Descriptiva	No supervisado	Identificación de patrones ocultos en los datos	K-Means, DBSCAN, SOM	Silhouette, Davies-Bouldin, Calinski-Harabasz
Clasificación (Predicción de diabetes)	Predictiva	Supervisado	* Histórico * Variable obj * Relación entre predictoras y la objetivo	RN Arboles RF Xgboost	Matriz de conf Precision Recall ROC

- Evaluación esperada

Línea base P=60%

1.5 RECURSOS PARA CREACIÓN DEL MODELO Y PARA **DESPLIEGUE**

Hardware (Hw)

- Computador portátil o de escritorio con mínimo 8 GB de RAM, procesador de **4 núcleos** y al menos 20 GB de almacenamiento disponible para manejo de datos.
- Conexión estable a Internet para descarga de librerías y datasets.

Software (Sw)

- Python 3.x con librerías: *pandas*, *numpy*, *scikit-learn*, *xgboost*, *matplotlib*, *seaborn*, *imbalanced-learn*.
- Google Colab para preparación y modelado.
- Streamlit para despliegue del modelo en interfaz web.
- GitHub para control de versiones y entrega final.

2. ENTENDIMIENTO DE LOS DATOS (Datos específicos del problema) (Experto TI)

2.1 CICLO DE LOS DATOS: Generación, Almacenamiento, Modificación (ruta), Periodicidad

- Generación: El conjunto de datos fue generado por el Centers for Disease Control and Prevention (CDC) como parte del programa Behavioral Risk Factor Surveillance System (BRFSS), una encuesta anual telefónica que recopila información sobre el estado de salud, hábitos y factores de riesgo de los residentes adultos en los Estados Unidos.
- Almacenamiento: base de datos pública alojada en los servidores del CDC y distribuida en Kaggle en formato CSV.
- Modificación (ruta): los datos fueron descargados desde Kaggle y se almacenarán en el entorno de trabajo local/Colab para su limpieza y modelado.
- Periodicidad: El conjunto de datos tiene periodicidad anual, ya que el BRFSS realiza sus encuestas y actualizaciones de datos cada año en los 50 estados de EE. UU., el Distrito de Columbia y tres territorios estadounidenses.
  La versión más reciente del dataset corresponde al año 2020.

2.2 DICCIONARIO DE DATOS

Variable	Descripción	Tipo

HeartDisease	Indica si la persona tiene enfermedad cardíaca (Variable objetivo)	categórica (binaria)
Smoking	Indica si la persona ha fumado alguna vez	categórica
AlcoholDrinking	Indica si la persona consume alcohol en exceso	categórica
PhysicalHealth	Días con mala salud física en los últimos 30 días	numérica (float)
BMI	Índice de masa corporal	numérica (float)
MentalHealth	Días con mala salud mental en los últimos 30 días	numérica (float)
Stroke	Alguna vez diagnosticado con accidente cerebrovascular	categórica
Race	Grupo étnico	categórica
Diabetic	Diagnóstico de diabetes	categórica
DiffWalking	Dificultad para caminar o subir escaleras	categórica
Sex	Sexo del encuestado	categórica
AgeCategory	Grupo de edad	categórica
PhysicalActivity	Actividad física en los últimos 30 días	categórica
GenHealth	Estado general de salud percibido	categórica
SleepTime	Horas promedio de sueño en 24 horas	numérica (float)
Asthma	Diagnóstico de asma	categórica
KidneyDisease	Diagnóstico de enfermedad renal	categórica
SkinCancer	Diagnóstico de cáncer de piel	categórica

2.3 REGLAS DE CALIDAD DESDE EL NEGOCIO (No salen de los datos)

Variable	Regla calidad (valores válidos)
`HeartDisease`	Solo puede ser `"Yes"` o `"No"`
`Sex`	Solo puede ser `"Male"` o `"Female"`

Smoking, AlcoholDrinking, Stroke, DiffWalking, Asthma, KidneyDisease, SkinCancer	Solo "Yes" o "No"
Diabetic	Solo puede ser "Yes", "No", "No, borderline diabetes", "Yes (during pregnancy)"
GenHealth	Solo "Excellent", "Very good", "Good", "Fair", "Poor"
AgeCategory	Solo los grupos definidos: "18-24", "25-29", "30-34", …, "80 or older"
Race	Solo "White", "Black", "Asian", "Hispanic", "Other"
BMI	Debe estar entre **10 y 80**
PhysicalHealth	Debe estar entre **0 y 30**
MentalHealth	Debe estar entre **0 y 30**
SleepTime	Debe estar entre **0 y 24**

3. PREPARACIÓN DE DATOS

3.1 INTEGRACIÓN

```
#cargamos los datos
df = pd.read_csv("/content/heart_2020_cleaned.csv")

df.head(10)
```

	HeartDisease	BMI	Smoking	AlcoholDrinking	Stroke	PhysicalHealth	MentalHealth	DiffWalking	Sex	AgeCategory	Race	Diabetic	PhysicalActivit
0	No	16.60	Yes	No	No	3.0	30.0	No	Female	55-59	White	Yes	Ye
1	No	20.34	No	No	Yes	0.0	0.0	No	Female	80 or older	White	No	Ye
2	No	26.58	Yes	No	No	20.0	30.0	No	Male	65-69	White	Yes	Ye
3	No	24.21	No	No	No	0.0	0.0	No	Female	75-79	White	No	N
4	No	23.71	No	No	No	28.0	0.0	Yes	Female	40-44	White	No	Ye
5	Yes	28.87	Yes	No	No	6.0	0.0	Yes	Female	75-79	Black	No	N
6	No	21.63	No	No	No	15.0	0.0	No	Female	70-74	White	No	Ye
7	No	31.64	Yes	No	No	5.0	0.0	Yes	Female	80 or older	White	Yes	N
8	No	26.45	No	No	No	0.0	0.0	No	Female	80 or older	White	No, borderline diabetes	N
9	No	40.69	No	No	No	0.0	0.0	Yes	Male	65-69	White	No	Ye

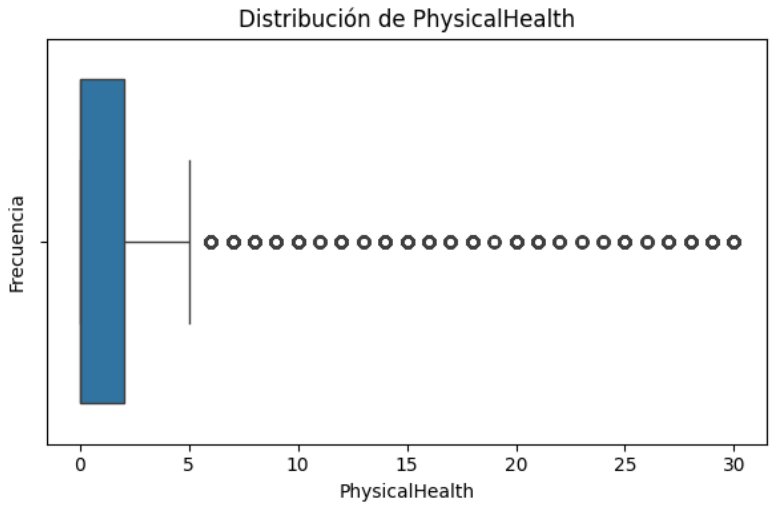
3.2 SELECCIÓN DE VARIABLES

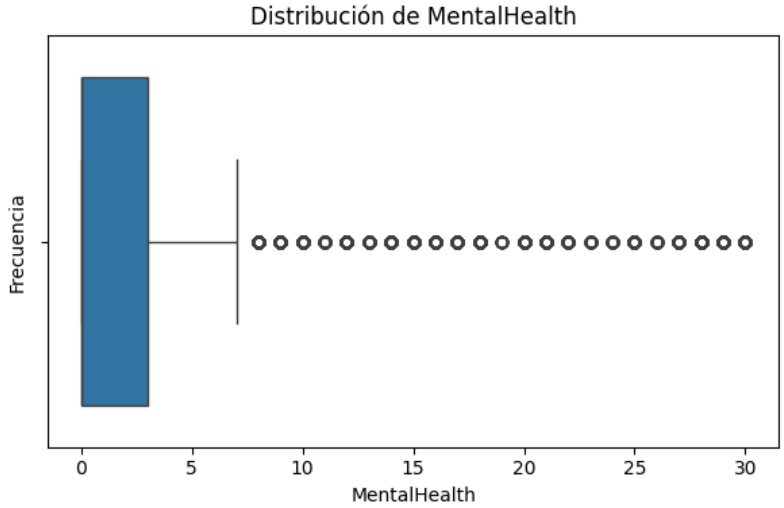
```
X = df.drop('HeartDisease', axis=1)
y = df['HeartDisease']
```

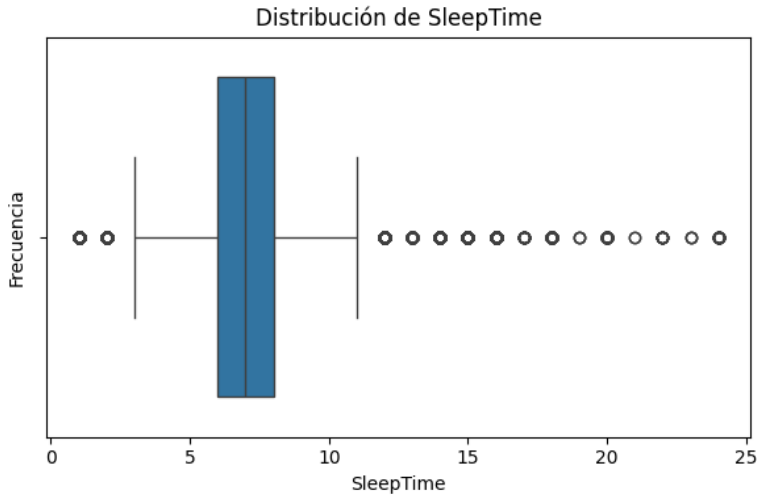
3.3 ESTADÍSTICA DESCRIPTIVA

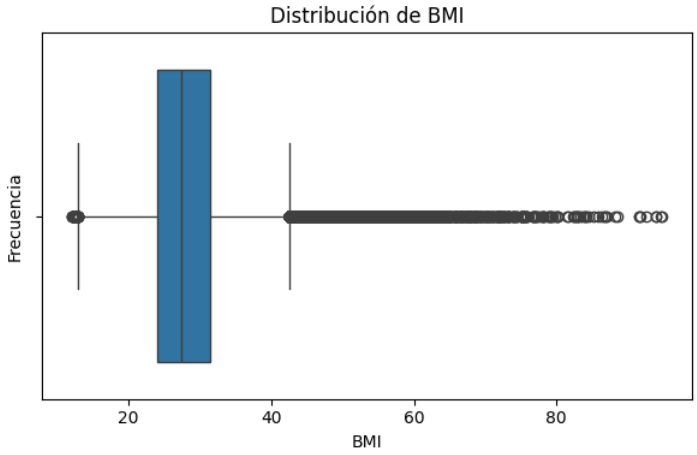
```
#Describimos variables numericas
df.describe()
```

	BMI	PhysicalHealth	MentalHealth	SleepTime
count	319795.000000	319795.00000	319795.000000	319795.000000
mean	28.325399	3.37171	3.898366	7.097075
std	6.356100	7.95085	7.955235	1.436007
min	12.020000	0.00000	0.000000	1.000000
25%	24.030000	0.00000	0.000000	6.000000
50%	27.340000	0.00000	0.000000	7.000000
75%	31.420000	2.00000	3.000000	8.000000
max	94.850000	30.00000	30.000000	24.000000

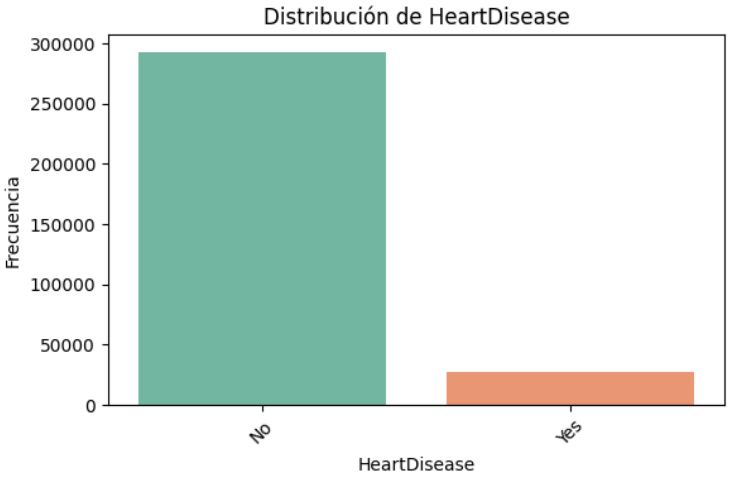


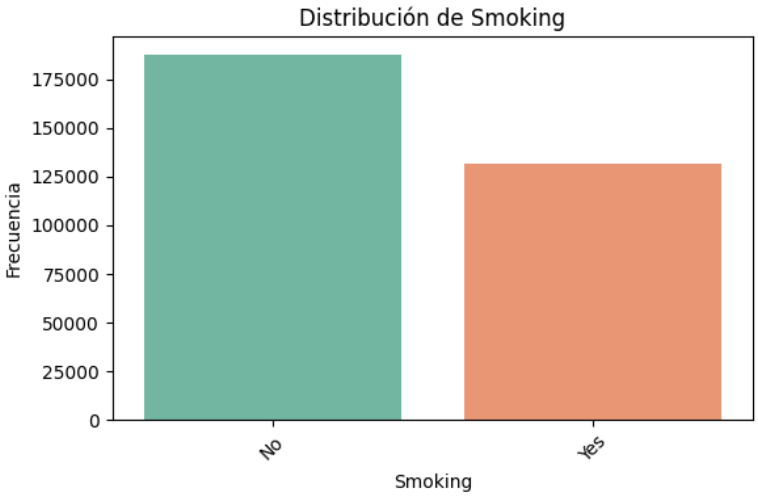


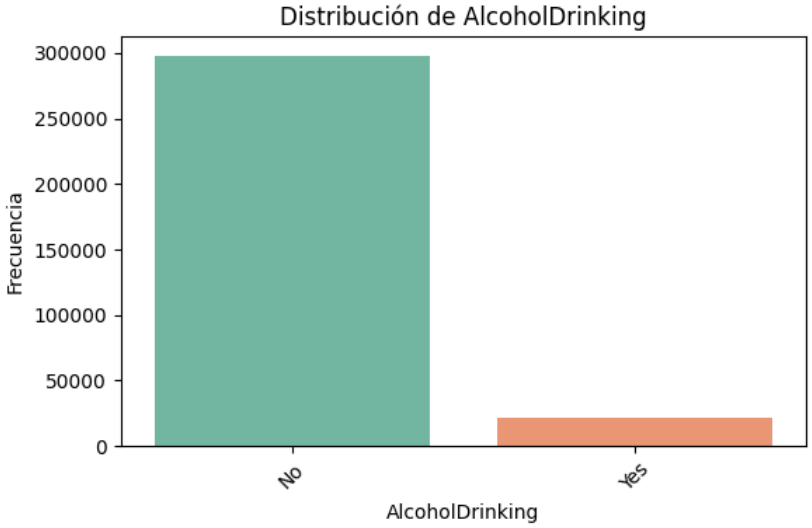


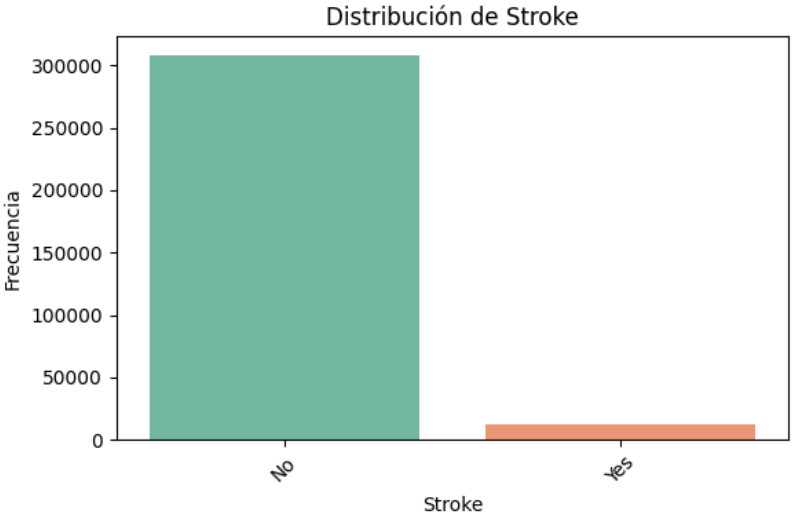


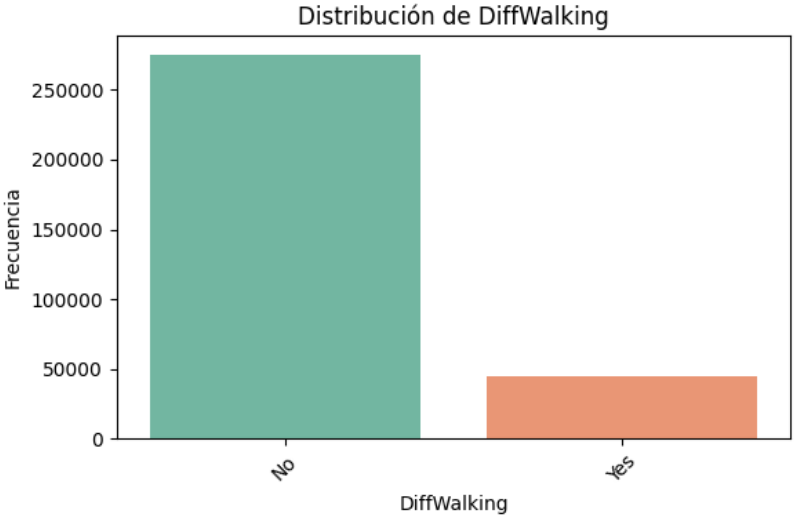
	count	unique	top	freq
HeartDisease	319795	2	No	292422
Smoking	319795	2	No	187887
AlcoholDrinking	319795	2	No	298018
Stroke	319795	2	No	307726
DiffWalking	319795	2	No	275385
Sex	319795	2	Female	167805
AgeCategory	319795	13	65-69	34151
Race	319795	6	White	245212
Diabetic	319795	4	No	269653
PhysicalActivity	319795	2	Yes	247957
GenHealth	319795	5	Very good	113858
Asthma	319795	2	No	276923
KidneyDisease	319795	2	No	308016
SkinCancer	319795	2	No	289976

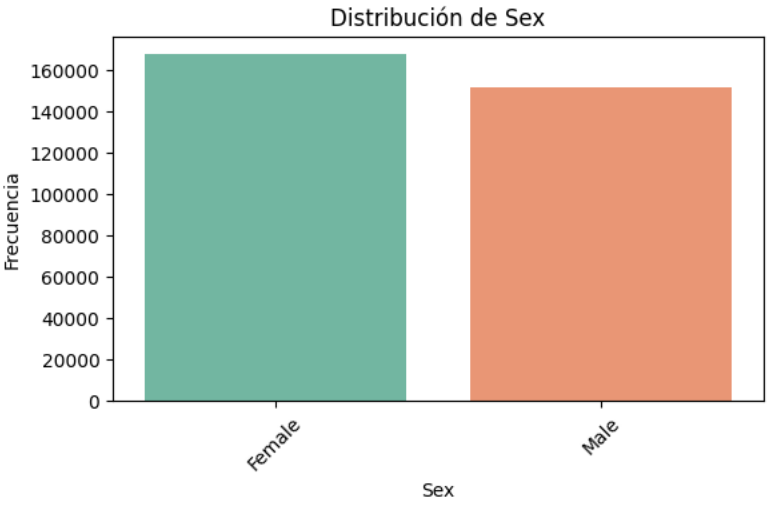

Distribución de HeartDisease
300000
250000
200000
150000
100000
50000
0
Frecuencia
No
Yes
HeartDisease

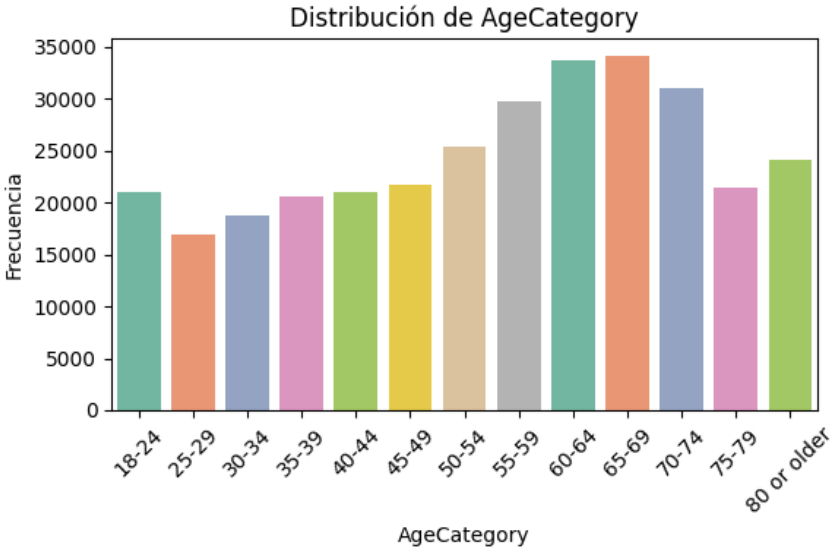

Distribución de Smoking
175000
150000
125000
100000
75000
50000
25000
0
Frecuencia
No
Yes
Smoking

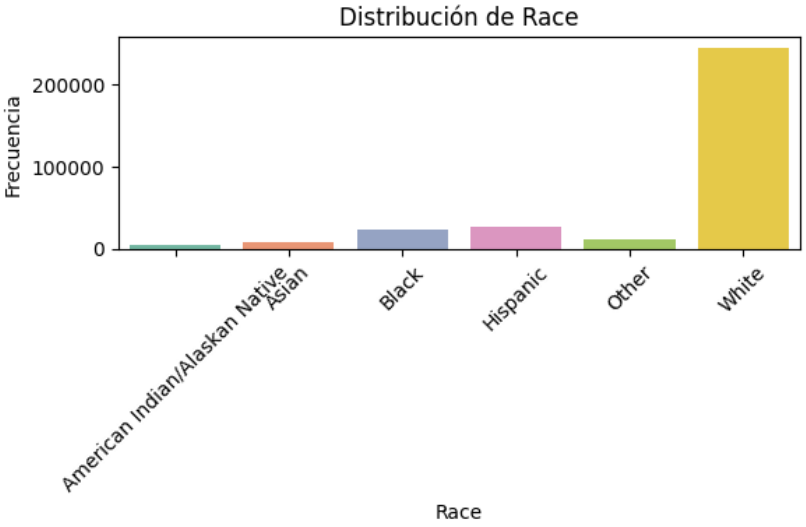

Distribución de AlcoholDrinking
300000
250000
200000
150000
100000
50000
0
Frecuencia
No
Yes
AlcoholDrinking

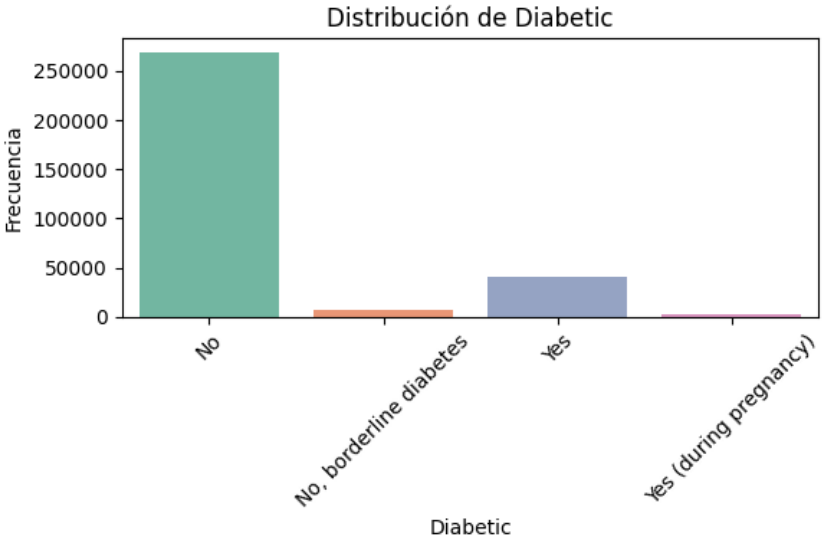

Distribución de Stroke
Frecuencia
300000
250000
200000
150000
100000
50000
0
No
Yes
Stroke

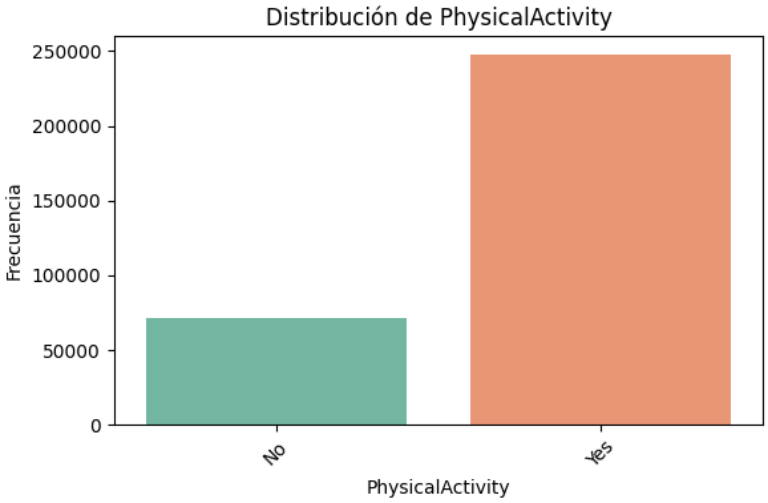

Distribución de DiffWalking
Frecuencia
250000
200000
150000
100000
50000
0
No
Yes
DiffWalking

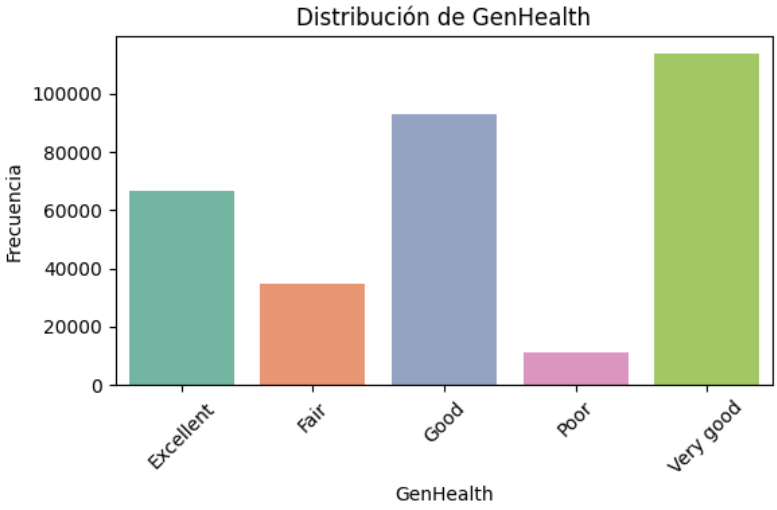

Distribución de Sex
Frecuencia
160000
140000
120000
100000
80000
60000
40000
20000
0
Female
Male
Sex

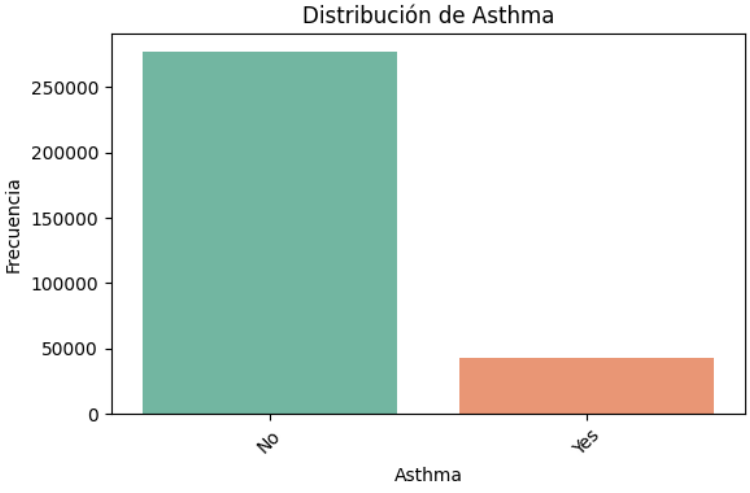

Distribución de AgeCategory
Frecuencia
0
5000
10000
15000
20000
25000
30000
35000
18-24
25-29
30-34
35-39
40-44
45-49
50-54
55-59
60-64
65-69
70-74
75-79
80 or older
AgeCategory

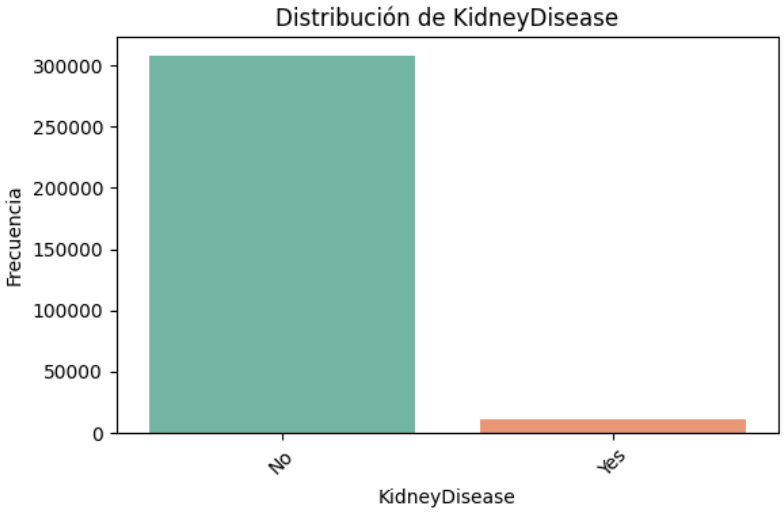

Distribución de Race
Frecuencia
0
100000
200000
American Indian/Alaskan Native
Asian
Black
Hispanic
Other
White
Race

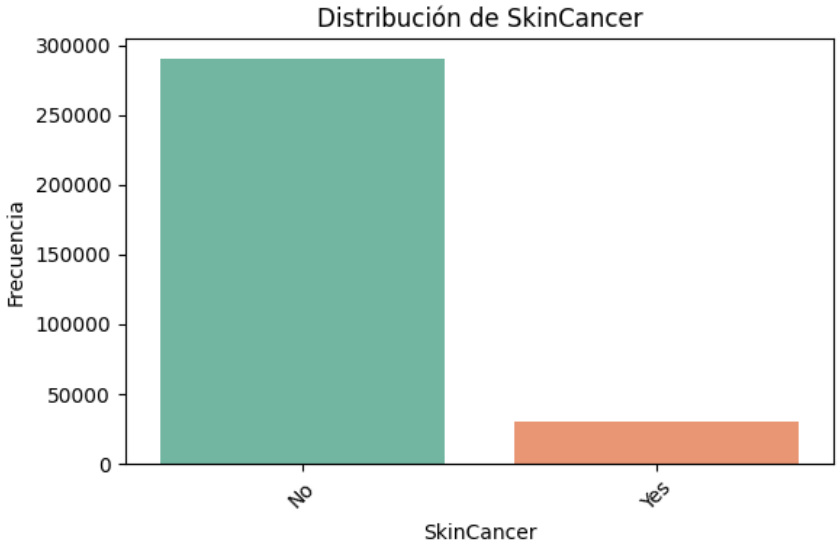

Distribución de Diabetic
Frecuencia
0
50000
100000
150000
200000
250000
No
No, borderline diabetes
Yes
Yes (during pregnancy)
Diabetic


Distribución de PhysicalActivity
Frecuencia
250000
200000
150000
100000
50000
0
No
Yes
PhysicalActivity


Distribución de GenHealth
Frecuencia
100000
80000
60000
40000
20000
0
Excellent
Fair
Good
Poor
Very good
GenHealth


Distribución de Asthma
Frecuencia
250000
200000
150000
100000
50000
0
No
Yes
Asthma






```
#Distribucion de la variable objetivo
df['HeartDisease'].value_counts(normalize=True) * 100
```

HeartDisease	proportion
No	91.440454
Yes	8.559546

dtype: float64

3.4 LIMPIEZA DE ATÍPICOS

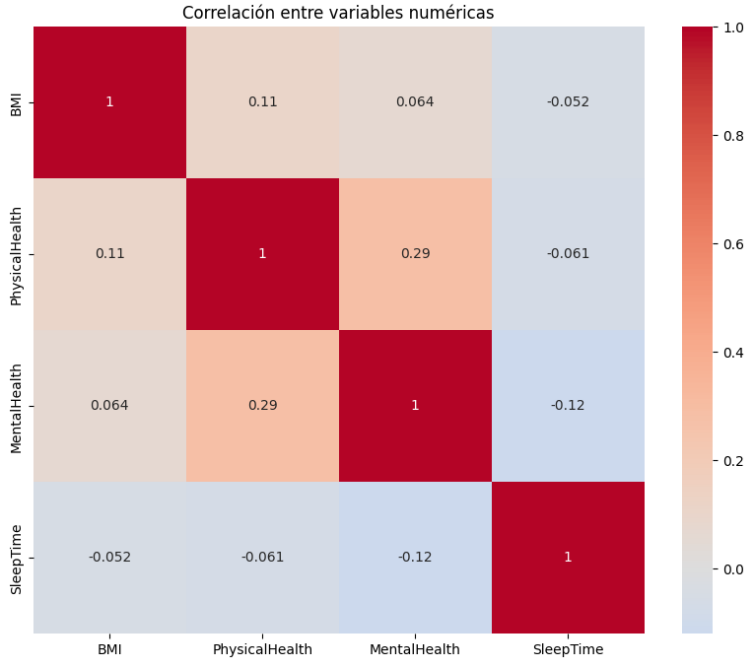
No presentamos valores atípicos por lo que podemos ver en los BOXPLOT

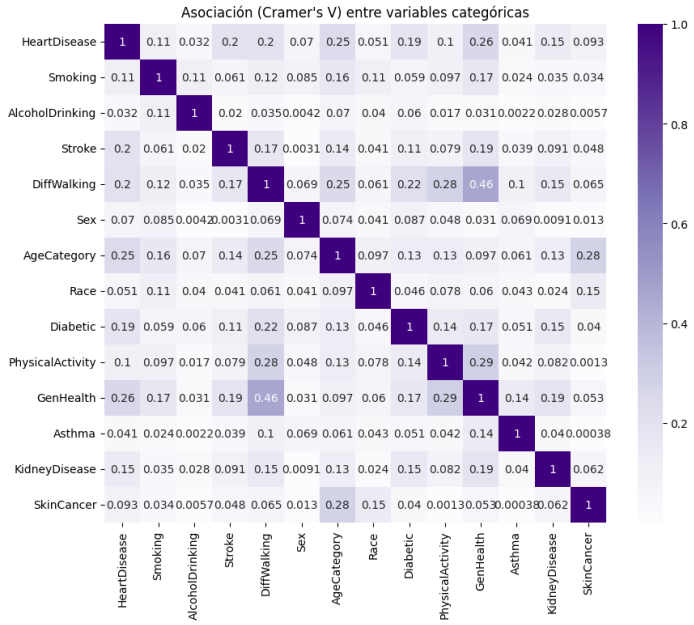
3.5 LIMPIEZA DE NULOS

```
print("\nCantidad de valores nulos por columna:")
print(df.isnull().sum())
```

```
Cantidad de valores nulos por columna:
HeartDisease        0
BMI                 0
Smoking             0
AlcoholDrinking     0
Stroke              0
PhysicalHealth      0
MentalHealth        0
DiffWalking         0
Sex                 0
AgeCategory         0
Race                0
Diabetic            0
PhysicalActivity    0
GenHealth           0
SleepTime           0
Asthma              0
KidneyDisease       0
SkinCancer          0
dtype: int64
```

3.6 ANÁLISIS DE CORRELACIONES PARA REDUNDANCIA





3.7 ANÁLISIS DE CORRELACIONES PARA IRRELEVANCIA (PREDICCIONES)

3.8 BALANCEO (CLASIFICACIÓN)

```
Distribución original:
 HeartDisease
No      204695
Yes      19161
Name: count, dtype: int64

Distribución después del balanceo (100%):
 HeartDisease
No      204695
Yes     204695
Name: count, dtype: int64
```

3.9 INGENIERÍA DE CARACTERÍSTICAS

3.9.1 CREACIÓN DE NUEVAS VARIABLES

No hubo necesidad de crear nuevas variables, después de hacer PCA nos quedaban aún más variables que las originales.

3.9.2 TRANSFORMACIONES

```
#Pasamos las columnas tipo object a category
df['HeartDisease'] = df['HeartDisease'].astype('category')
df['Smoking'] = df['Smoking'].astype('category')
df['AlcoholDrinking'] = df['AlcoholDrinking'].astype('category')
df['Stroke'] = df['Stroke'].astype('category')
df['DiffWalking'] = df['DiffWalking'].astype('category')
df['Sex'] = df['Sex'].astype('category')
df['AgeCategory'] = df['AgeCategory'].astype('category')
df['Race'] = df['Race'].astype('category')
df['Diabetic'] = df['Diabetic'].astype('category')
df['PhysicalActivity'] = df['PhysicalActivity'].astype('category')
df['GenHealth'] = df['GenHealth'].astype('category')
df['Asthma'] = df['Asthma'].astype('category')
df['KidneyDisease'] = df['KidneyDisease'].astype('category')
df['SkinCancer'] = df['SkinCancer'].astype('category')

df.info()
```

4. MODELAMIENTO, EVALUACIÓN E INTERPRETACIÓN

4.1 CONFIGURACIÓN MÉTODOS DE MACHINE LEARNING

```
'Decision Tree': DecisionTreeClassifier(
    criterion='gini',
    max_depth=5,
    min_samples_split=10,
    min_samples_leaf=5,
    random_state=42
),
"Regresión Logística": LogisticRegression(
    max_iter=1000,
    random_state=42
    ),

'Naive Bayes': GaussianNB(),

'Random Forest': RandomForestClassifier(
    n_estimators=100,
    max_depth=10,
    min_samples_split=10,
    min_samples_leaf=5,
    random_state=42,
    n_jobs=-1
),
```

```
'Gradient Boosting': GradientBoostingClassifier(
    n_estimators=100,
    learning_rate=0.1,
    max_depth=5,
    min_samples_split=10,
    min_samples_leaf=5,
    random_state=42
),

'AdaBoost': AdaBoostClassifier(
    estimator=DecisionTreeClassifier(max_depth=3),
    n_estimators=50,
    learning_rate=1.0,
    random_state=42
),

'XGBoost': XGBClassifier(
    n_estimators=100,
    max_depth=6,
    learning_rate=0.1,
    subsample=0.8,
    colsample_bytree=0.8,
    random_state=42,
    n_jobs=-1,
    eval_metric='logloss'
)
```

4.2 ANALISIS DE MEDIDAS DE CALIDAD

```
Evaluando Decision Tree...
  F1: 0.7912 | Recall: 0.8204 | Precision: 0.7639 | Accuracy: 0.7835
Evaluando Regresión Logística...
  F1: 0.7819 | Recall: 0.8043 | Precision: 0.7608 | Accuracy: 0.7757
Evaluando Naive Bayes...
  F1: 0.7751 | Recall: 0.8658 | Precision: 0.7017 | Accuracy: 0.7488
Evaluando Random Forest...
  F1: 0.8740 | Recall: 0.8599 | Precision: 0.8887 | Accuracy: 0.8761
Evaluando Gradient Boosting...
  F1: 0.9233 | Recall: 0.8986 | Precision: 0.9494 | Accuracy: 0.9254
Evaluando AdaBoost...
  F1: 0.9088 | Recall: 0.8940 | Precision: 0.9242 | Accuracy: 0.9103
Evaluando XGBoost...
  F1: 0.9234 | Recall: 0.8991 | Precision: 0.9491 | Accuracy: 0.9255
```

Decision Tree:

- **Precision (0.7639):** De todas las predicciones positivas que hizo el modelo, el 76% fueron correctas. → Hay un número moderado de **falsos positivos**.
- **Recall (0.8204):** Detectó el 82% de los casos que realmente eran positivos. → Tiende a **capturar bastantes positivos**, aunque se le escapan algunos.
- **Accuracy (0.7835):** Globalmente acierta el 78% de las veces. Correcto, pero no excelente.
- **F1 (0.7912):** Buen equilibrio entre precisión y recall, pero sin destacar.

Regresion Logistica:

- **Precision (0.7608):** De cada 10 positivos predichos, unos 7.6 son correctos.
- **Recall (0.8043):** Detecta el 80% de los positivos verdaderos.
- **Accuracy (0.7757):** Globalmente clasifica bien el 77% de los casos.
- **F1 (0.7819):** Representa un balance entre ambos.

Naive Bayes:

- **Precision (0.7017):** Solo el 70% de los positivos predichos son realmente positivos, **muchos falsos positivos**.
- **Recall (0.8658):** Detecta casi el 87% de los positivos reales, **pocos falsos negativos**.
- **Accuracy (0.7488):** 75% de aciertos globales.
- **F1 (0.7751):** Compensa la baja precisión con un alto recall.

Random Forest:

- **Precision (0.8887):** 89% de las predicciones positivas fueron correctas.
- **Recall (0.8599):** Captura el 86% de los positivos reales.
- **Accuracy (0.8761):** Muy buen desempeño global.
- **F1 (0.8740):** Equilibrio sólido entre precisión y recall.

Gradient Boosting:

- **Precision (0.9494):** El 95% de los positivos predichos fueron correctos, **casi no hay falsos positivos**.
- **Recall (0.8986):** Detecta el 90% de los casos positivos, **pocos falsos negativos**.
- **Accuracy (0.9254):** Acierta el 92.5% de las veces.
- **F1 (0.9233):** Excelente balance entre precisión y recall.

AdaBoost:

- **Precision (0.9242):** 92% de aciertos entre las predicciones positivas.
- **Recall (0.8940):** Detecta casi el 90% de los positivos.
- **Accuracy (0.9103):** Muy buena tasa global de aciertos.
- **F1 (0.9088):** Excelente equilibrio.

XGboost:

- **Precision (0.9491):** 95% de las predicciones positivas son correctas, **muy pocos falsos positivos**.
- **Recall (0.8991):** Captura casi el 90% de los positivos reales, **excelente sensibilidad**.
- **Accuracy (0.9255):** Clasifica correctamente el 92.5% de los casos.
- **F1 (0.9234): El mejor equilibrio total**.

4.3 SELECCIÓN DEL MEJOR MODELO

Comparación de calidad mediante pruebas estadística ANOVA, Tukey

```
Estadístico F: 17227.1036
P-value: 0.000000

 Resultado: Hay diferencias SIGNIFICATIVAS entre los modelos (p < 0.05)
  → Procederemos con el Test de Tukey para comparaciones múltiples
TEST DE TUKEY (Honestly Significant Difference)
```

```
TEST DE TUKEY (Honestly Significant Difference)

 Advertencia: scikit-posthocs no está instalado
Instalando: pip install scikit-posthocs

Realizando comparaciones pareadas manualmente con t-test...

Matriz de p-values (t-test pareado):
(Valores < 0.05 indican diferencias significativas)
--------------------------------------------------------------------------------------------------
                    Decision Tree  Regresión Logística  Naive Bayes  Random Forest  Gradient Boosting  AdaBoost  XGBoost
Decision Tree              1.0000               0.0015       0.0002         0.0000             0.0000    0.0000   0.0000
Regresión Logística        0.0015               1.0000       0.0005         0.0000             0.0000    0.0000   0.0000
Naive Bayes                0.0002               0.0005       1.0000         0.0000             0.0000    0.0000   0.0000
Random Forest              0.0000               0.0000       0.0000         1.0000             0.0000    0.0000   0.0000
Gradient Boosting          0.0000               0.0000       0.0000         0.0000             1.0000    0.0000   0.6471
AdaBoost                   0.0000               0.0000       0.0000         0.0000             0.0000    1.0000   0.0000
XGBoost                    0.0000               0.0000       0.0000         0.0000             0.6471    0.0000   1.0000
SELECCIÓN DE LOS 3 MEJORES MODELOS
```

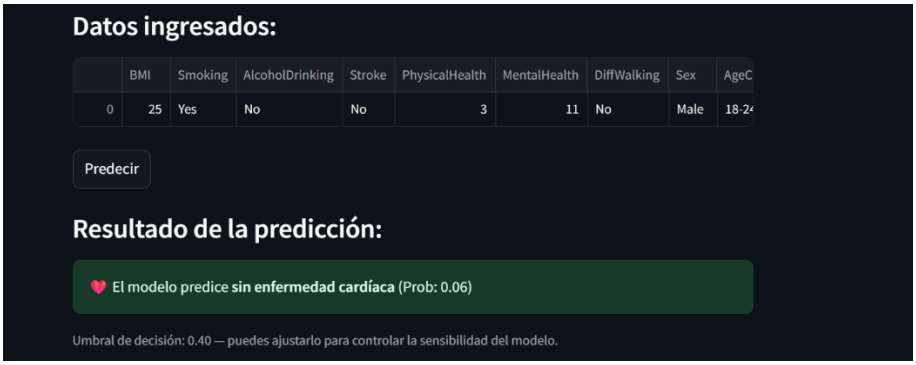
```
   TOP 3 MODELOS SELECCIONADOS (basado en F1-Score medio):
  1. XGBoost                   F1-Score: 0.8334 (±0.0004)
  2. Gradient Boosting         F1-Score: 0.8333 (±0.0011)
  3. AdaBoost                  F1-Score: 0.8159 (±0.0025)

   ANÁLISIS DE DIFERENCIAS ENTRE TOP 3:
    • XGBoost vs Gradient Boosting: p=0.9241 → Sin diferencia significativa
    • XGBoost vs AdaBoost: p=0.0006 → Diferencia SIGNIFICATIVA
    • Gradient Boosting vs AdaBoost: p=0.0008 → Diferencia SIGNIFICATIVA
  ESTOS 3 MODELOS PASARÁN AL PROCESO DE OPTIMIZACIÓN (GRID SEARCH)
```

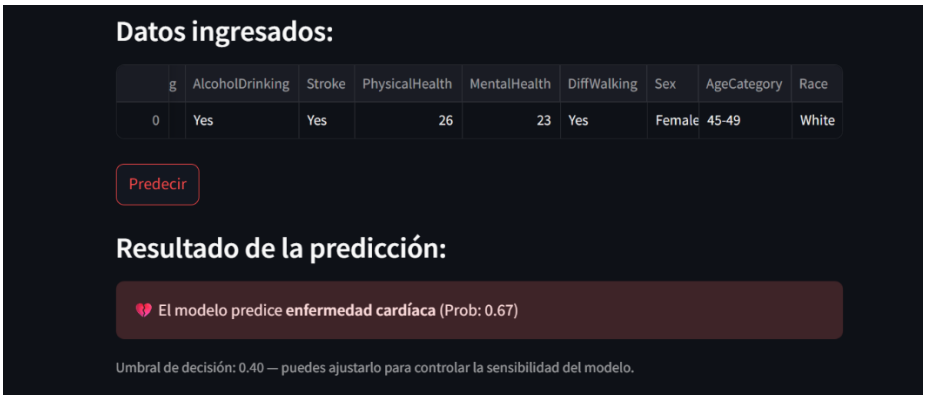
Tiempo computacional de creación y despliegue

```
Probando modelo: XGBoost
  Tiempo total (3 folds): 0.6s | aprox por fold: 0.2s

Probando modelo: Gradient Boosting
  Tiempo total (3 folds): 3.2s | aprox por fold: 1.1s

Probando modelo: AdaBoost
  Tiempo total (3 folds): 1.0s | aprox por fold: 0.3s
```

5. DESPLIEGUE

5.1 PREDICCIÓN DE DATOS FUTUROS: almacenar modelo, pipes para el despliegue, servicio web de desliegue





La etapa de Despliegue es la culminación del proyecto, cuyo objetivo es transformar el modelo optimizado en un servicio web funcional accesible a los usuarios. Para garantizar la integridad de los datos en tiempo real, se implementó una estrategia de persistencia basada en el Pipeline de scikit-learn.

5.2 MONITOREO

El Monitoreo es una fase crítica posterior al despliegue para asegurar que el modelo mantenga su rendimiento en el entorno de producción. Su objetivo es detectar a tiempo si el modelo está sufriendo un Decaimiento (Model Decay), generalmente causado por un cambio en las características de los datos.

Se requiere un sistema de monitoreo continuo enfocado en dos indicadores principales:

Indicador	Descripción	Implicación y Acción Requerida
Decaimiento del Modelo (Model Decay)	Medición constante de las métricas de rendimiento (especialmente F1-Score y ROC-AUC) del modelo en datos nuevos y etiquetados.	Si el rendimiento cae por debajo de un umbral aceptable (ej., 3% de caída en F1-Score), el modelo ya no es confiable, y se debe iniciar un proceso de reentrenamiento y Re optimización con datos recientes.
Deriva de Datos (Data Drift)	Detección de cambios significativos en la distribución de las variables de entrada del servicio web (ej., la edad promedio de los nuevos pacientes cambia drásticamente, o los hábitos de tabaquismo varían).	La Deriva de Datos es la causa principal del decaimiento. Si se detecta, indica que las reglas de preprocesamiento (el ColumnTransformer del Pipeline) y el modelo están entrenados con una realidad obsoleta, lo que también exige reentrenamiento.

5.3 CRONOGRAMA DE MANTENIMIENTO/RE-ENTRENAMIENTO

Re-entrenamiento Programado

Frecuencia Base: Trimestral (cada 3 meses)

- **Justificación**: Los datos médicos no cambian drásticamente a corto plazo
- **Actividades**:
    - Recolectar nuevos datos de los últimos 3 meses
    - Validar calidad de datos nuevos
    - Re-entrenar modelo con datos históricos + nuevos
    - Evaluar en conjunto de validación temporal
    - Comparar métricas: nuevo modelo vs modelo actual
    - Desplegar solo si mejora F1-Score > 2% o mantiene rendimiento

Fechas sugeridas:

- Enero (inicio de año)
- Abril (después del Q1)
- Julio (mitad de año)
- Octubre (preparación para fin de año)

Re-entrenamiento Extraordinario (Triggered)

Activar re-entrenamiento inmediato si:

- **F1-Score cae < 0.85** (umbral crítico)
- **Recall cae < 0.55** (no detecta enfermos adecuadamente)
- **Data drift significativo detectado** (KS test p-value < 0.05)
- **Cambios en protocolos médicos** o definiciones de variables
- **Nueva fuente de datos** disponible

Proceso de Re-entrenamiento

Paso 1: Preparación (Día 1)

- Extraer datos nuevos desde última fecha
- Validar calidad: valores faltantes, outliers, consistencia
- Unir con datos históricos (mantener últimos 2-3 años)

Paso 2: Entrenamiento (Día 2-3)

- Balancear datos (70% de casos positivos si es necesario)
- Validación cruzada estratificada (3-5 folds)
- Selección de top 3 modelos con ANOVA/Tukey
- Optimización bayesiana de hiperparámetros
- Evaluar en conjunto de validación temporal (datos más recientes)

Paso 3: Validación (Día 4)

- Comparar métricas: modelo nuevo vs modelo actual
- Análisis de errores: revisar falsos positivos/negativos
- Test A/B (si es posible): 10% tráfico al nuevo modelo
- Validación con experto médico (opcional pero recomendado)

Paso 4: Despliegue (Día 5)

- Guardar modelo antiguo como backup
- Desplegar nuevo modelo en producción
- Actualizar baseline de monitoreo
- Documentar cambios en versión del modelo

Versionado de Modelos

Versión	Fecha	F1-Score	Recall	Notas
v1.0	2025-01	0.9233	0.6389	Modelo inicial (XGBoost optimizado)
v1.1	2025-04	TBD	TBD	Re-entrenamiento Q1
v1.2	2025-07	TBD	TBD	Re-entrenamiento Q2

Mantenimiento de Infraestructura

Mensual:

- Backup de modelos y datos
- Revisión de logs de errores
- Actualización de dependencias (si hay parches de seguridad)

Trimestral:

- Optimización de performance (si latencia aumenta)
- Revisión de almacenamiento y escalabilidad

Anual:

- Evaluación completa del sistema
- Considerar migración a nuevos algoritmos (si hay avances)
- Revisión de costos de infraestructura