Aplicación de algoritmos de machine learning para la predicción de factores de riesgo en la prevalencia del accidente cerebrovascular en la Subred de Servicios de Salud Sur Occidente

Ronal Alberto Roa Ayala

Asesor

Sebastián Vélez Jaramillo

Universidad Nacional Abierta y a Distancia UNAD

Escuela Ciencias Básicas Tecnología e Ingeniería

Especialización Ciencia de Datos y Analítica

2025

Sebastian Velez Jaramillo

Andres Felipe Hernandez Giraldo

Jurado

2025

Resumen

Este proyecto de Investigación Aplicada se realizara en la Subred de Servicios de Salud Sur Occidente Empresa Social del Estado que presta servicios de salud a la población de la ciudad de Bogotá conformada por los Hospitales Occidente de Kennedy, Hospital Fontibón, Hospital de Bosa, Hospital del Sur, Hospital Tintal y otras 30 unidades de servicios de salud de niveles de atención I, II y III, que cubre las localidades de Bosa, Kennedy, Fontibón, Puente Aranda en la ciudad de Bogotá, con el fin de analizar la población que ha presentado ACV (Accidente Cerebrovascular) en pacientes que ha sido atendidas en los centros de salud de la Subred.

En la actualidad el servicio de Neurología de la subred no cuenta con un mecanismo tecnológico para el análisis de la información de los pacientes que han sufrido esta enfermedad, la estadística que se lleva en el servicio es muy manual donde mediante archivos de Excel recopilan la información sustrayendo datos de las historias clínica de cada uno de los pacientes lo que se convierte en una tarea muy tediosa y compleja para hacer todo un proceso de análisis con los datos capturados de forma uno a uno.

Con la aplicación de técnicas de inteligencia artificial con modelos de aprendizaje en machine learning en el sector salud las cuales son cada vez más comunes para la prevención de enfermedades donde con el análisis de datos y patrones de la población se pueden aplicar diagnósticos y tratamientos de forma oportuna y así reducir la mortalidad por enfermedades, a nivel mundial el ACV Enfermedad Cerebrovascular es una de las enfermedades con altas tasas de mortalidad y que a su vez deja grandes secuelas a los pacientes que alcanzan a tener un tratamiento medianamente oportuno.

Con este proyecto se pretende aplicar Algoritmos de Machine Learning para la Predicción de Factores de Riesgo en la Prevalencia del Accidente Cerebrovascular en la Subred de Servicios de Salud Sur Occidente, donde mediante consultas SQL a la base de datos del sistema de información de historia clínica electrónica se obtenga los datos de los pacientes que fueron registrados con el diagnóstico ACV y hacer el análisis de los datos con el fin de poder determinar posibles factores de riesgo que prevalecen en la población, donde se pueda obtener indicadores de riesgo personalizado en base a la información recolectada de los datos de las historias clínicas con características que pueden incluir marcadores biológicos, datos médicos, datos relativos al estilo de vida o datos contextuales que describen el entorno de un paciente y aplicar algoritmos de Machine Learning para predecir el riesgo de un paciente para una patología concreta y su tratamiento a tiempo.

Dentro del estudio se tomara un filtro en la consulta a la base de datos de la subred donde la población que se hará el estudio se aplicara a pacientes atendidos desde el año 2019 a 31 de Diciembre 2024 y que se hayan registrado con diagnóstico de Accidente Cerebrovascular, se determinara las variables a estudiar demográficas, descriptivas y relacionadas con la atención medica que se haya registrado en las historias clínicas, esta Data será anonimizada para proteger los derechos del paciente de privacidad e integridad y no incurrir en una falta legal teniendo en cuenta el manejo de datos clínicos de la Historia Clínica.

Palabras clave: Accidente Cerebrovascular, Machine Learning, Patrones Demográficos, Random Forest, Regresión Logística, Python, SQL.

Abstract

This applied research project will be carried out in the Subred de Servicios de Salud Sur Occidente Empresa Social del Estado that provides health services to the population of the city of Bogota, formed by the Hospitals Occidente de Kennedy, Hospital Fontibón, Hospital de Bosa, Hospital del Sur, Hospital Tintal and other 30 units of health services of levels of care I, II and III, covering the localities of Bosa, Kennedy, Fontibón, Puente Aranda in the city of Bogota, Hospital Tintal and other 30 health service units of levels of care I, II and III, which covers the localities of Bosa, Kennedy, Fontibón, Puente Aranda in the city of Bogotá, in order to analyze the population that has presented CVA (Cerebrovascular Accident) in patients that have been attended in the health centers of the Sub-Network.

At present, the Neurology service of the sub-network does not have a technological mechanism for the analysis of the information of the patients who have suffered this disease, the statistics that are kept in the service is very manual where by means of Excel files they compile the information subtracting data from the clinical histories of each one of the patients which becomes a very tedious and complex task to make a whole process of analysis with the data captured one by one.

With the application of artificial intelligence techniques with machine learning models in the health sector which are increasingly common for the prevention of diseases where with the analysis of data and patterns of the population can be applied diagnoses and treatments in a timely manner and thus reduce mortality from diseases, worldwide stroke Cerebrovascular Disease is one of the diseases with high mortality rates and in turn leaves great sequelae to patients who manage to have a moderately timely treatment.

This project aims to apply Machine Learning Algorithms for the Prediction of Risk Factors in the Prevalence of Stroke in the South West Health Services Subnetwork, where through SQL queries to the database of the electronic medical record information system to obtain the data of patients who were registered with the diagnosis of stroke and analyze the data in order to determine possible risk factors prevalent in the population, where personalized risk indicators can be obtained based on the information collected from the medical records data with characteristics that may include biological markers, medical data, lifestyle data or contextual data describing a patient's environment and apply Machine Learning algorithms to predict a patient's risk for a particular pathology and its timely treatment.

Within the study a filter will be taken in the consultation to the database of the subnetwork where the population to be studied will be applied to patients treated from 2019 to December 31, 2024 and who have been registered with a diagnosis of stroke, the variables to be studied will be determined demographic, descriptive and related to the medical care that has been recorded in the medical records, this data will be anonymized to protect the patient's rights of privacy and integrity and not to incur in a legal offense taking into account the handling of clinical data of the medical records.

***Keywords*:** Stroke, Machine Learning, Demographic Patterns, Random Forest, Logistic Regression, Python, SQL.

Tabla de Contenido

Lista de Figuras

Introducción

El accidente cerebrovascular es una de las principales causas de discapacidad y mortalidad a nivel global, donde hay un desafío para los sistemas de salud como para las familias de los pacientes que presentan esta enfermedad.

En la Subred de Servicios de Salud Sur Occidente, el manejo de la información de los pacientes que presentan un accidente cerebrovascular es muy manual, lo que implica un proceso, tedioso y lento para la generación de datos para su análisis lo cual puede generar errores al ser todo muy manual, dificultando a los especialistas del servicio de Neurología poder identificar patrones de factores de riesgo para poder diseñar estrategias preventivas para mejorar la toma de decisiones a nivel clínico como administrativo. La implementación de algoritmos de aprendizaje automático machine learning, permitirá automatizar y optimizar el análisis de los datos de los pacientes contribuyendo a un diagnóstico más preciso y a la identificación temprana de factores de riesgo prevalentes en la población de la ciudad de Bogotá.

El uso de machine learning en el análisis de datos clínicos representa un avance tecnológico y da un valor muy alto aportando a la transformación digital de la salud en Colombia promoviendo la integración eficiente de los sistemas de información para beneficio de los pacientes y los profesionales de la salud.

Marco Conceptual y Teórico

Accidente Cerebrovascular

Los accidentes cerebrovasculares representan un grupo de patologías que impactan el sistema nervioso central, situado en el cráneo. En la actualidad, el término ACV se ha consolidado como una forma más completa de designar este grupo de enfermedades, que incluye la isquemia y la hemorragia intracerebral. Asimismo, los accidentes isquémicos transitorios, conocidos por su etiología cerebrovascular isquémica, están contemplados dentro de los grupos de enfermedades cardiovasculares. El ataque cerebral o evento cerebrovascular es un síndrome neurológico resultante de una anormalidad cerebrovascular que ocurre debido a la pérdida súbita de la función neurológica. Esto se origina por un trastorno vascular, con una duración mínima de 24 horas, o con la posibilidad de ser fatal. En cuanto a la implementación de terapias basadas en factores etiológicos verdaderamente específicos, el 80% de la etiología del ACV está directamente relacionada con causas cardioembólicas, siendo la fibrilación auricular su principal origen. (Puy & Jouvent, 2020)

Machine Learning en el Ámbito de la Salud

El machine learning es una técnica de inteligencia artificial que se basa en un conjunto de algoritmos que permiten hacer predicciones sobre conjuntos específicos de datos. Estas técnicas emergieron de enfoques cognitivos o inspirados en la neurofisiología de las máquinas. Durante el proceso de aprendizaje, los algoritmos buscan patrones que puedan prever nuevas observaciones a partir de conjuntos de datos, utilizando técnicas como el cálculo de un conjunto inicial de reglas para modelar el comportamiento general de un objeto o el aprendizaje a partir de ejemplos conocidos. También se utilizan algoritmos de comités que combinan varios algoritmos para aumentar el rendimiento de uno solo. El rápido acceso a dispositivos diseñados para captar

señales fisiológicas y al enorme volumen de datos generados por registros electrónicos, ha contribuido al incremento del machine learning en el ámbito de la salud. Actualmente, las tecnologías han facilitado la adquisición y almacenamiento de datos desde la prevención hasta el tratamiento y monitoreo de la población de alto o mediano riesgo, o que padece la enfermedad. Las técnicas supervisadas se utilizan ampliamente en diagnóstico y pronóstico, centrándose en las relaciones entre las observaciones y los resultados. (Carrillo, 2024)

Justificación

El accidente cerebrovascular es una de las principales causas de discapacidad y mortalidad a nivel global, donde hay un desafío para los sistemas de salud como para las familias de los pacientes que presentan esta enfermedad.

En la Subred de Servicios de Salud Sur Occidente, el manejo de la información de los pacientes que presentan un accidente cerebrovascular es muy manual, lo que implica un proceso, tedioso y lento para la generación de datos para su análisis lo cual puede generar errores al ser todo muy manual, dificultando a los especialistas del servicio de Neurología poder identificar patrones de factores de riesgo para poder diseñar estrategias preventivas para mejorar la toma de decisiones a nivel clínico como administrativo. La implementación de algoritmos de aprendizaje automático machine learning, permitirá automatizar y optimizar el análisis de los datos de los pacientes contribuyendo a un diagnóstico más preciso y a la identificación temprana de factores de riesgo prevalentes en la población de la ciudad de Bogotá.

El uso de machine learning en el análisis de datos clínicos representa un avance tecnológico y da un valor muy alto aportando a la transformación digital de la salud en Colombia promoviendo la integración eficiente de los sistemas de información para beneficio de los pacientes y los profesionales de la salud.

Objetivos

Objetivo General

Evaluar la prevalencia del accidente cerebrovascular (ACV) y los factores de riesgo asociados en la población atendida por la Subred de Servicios de Salud Sur Occidente.

Objetivos Específicos

Elaborar un Dataset con datos anonimizados con información extraída de la base de datos del sistema de información clínica de la Subred de Servicios de Salud Sur Occidente para el estudio de todos los pacientes atendidos dentro del período 2019 al 2024 y que hayan presentado como diagnostico principal un Accidente Cerebrovascular (ACV).

Identificar y analizar los patrones de asociación entre variables demográficas y la presencia de ACV en la población atendida por la Subred durante el período 2019-2024, mediante la aplicación de modelos Supervisados de Machine Learning estadísticos de regresión, con el fin de cuantificar la fuerza de estas asociaciones y determinar cuáles presentan una relación significativa con la ocurrencia de ACV.

Interpretar los resultados obtenidos del análisis de prevalencia y las asociaciones entre factores demográficos y ACV, para identificar aquellos con mayor impacto en la ocurrencia del ACV en la población estudiada.

Contenido del Trabajo

Estudio de Variables de la Historia Clínica

Se realizo un estudio de las Historias Clínicas de la Subred de Servicios de Salud Sur Occidente para determinar el conjunto de datos que contienen los formatos de Historia Clínica que cuenta el sistema de información de la Subred, dentro de este primer estudio de revisión de fuente de datos se destacó que los formatos a estudiar para la captura de información correspondían a dos servicios, el servicio de Urgencias con el formato de Triage y un formato de Evolución Medica, el resto de formatos no se tuvieron en cuenta ya que el uso directo del objetivo de este proyecto tenía una baja probabilidad de encontrar datos relevantes para el estudio, para dar soporte al trabajo de estudio de obtención de fuente de datos se hizo una consulta adicional a la base de datos de la Subred para identificar los formatos que tienen una mayor usabilidad dentro de la Institución.

Una vez detectado los formatos fuente para la extracción de datos se inició un proceso de estudio de las variables que contiene dichos formatos, con el fin de empezar a realizar el Dataset para nuestro estudio.

La creación del Dataset tuvo una comprensión de los datos, donde en esta fase se realizó la consulta a la base de datos mediante SQL SERVER con las variables relevantes donde se revisó la completitud, consistencia y exactitud de los datos a estudiar. En el proceso de extracción de datos se tuvo en cuenta variables demográficas del paciente y variables clínicas dentro del periodo del 01 de Enero del 2019 al 31 de Diciembre del 2024, es de tener en cuenta que en esta fase se realizó el proceso de Anonimización y conformidad en el uso ético de Datos Personales sensibles teniendo en cuenta la Ley Colombiana 1581 de 2012 y el las directrices del Reglamento General de Protección de Datos (RGPD) la Unión Europea, donde se aplicó técnicas

de Anonimización de datos como la Supresión de variables sensibles con mayor impacto de identificación.

A continuación, se relaciona la consulta realizada para la obtención del dataset a trabajar:

```
SELECT  DISTINCT  U_PACTIPDOC.TIPDOC AS 'TIPO DOC PACIENTE'
                ,DBO.EDAD_YEARS(GENPACIEN.GPAFECNAC, GETDATE())
AS 'EDAD PACIENTE'
                ,GENPACIEN.GPASEXPAC AS 'SEXO (1 = MASCULINO 2 =
FEMENINO)'
                ,GENTERCER.TERNOMCOM AS 'ENTIDAD'
                ,GENPACIEN.GENESTRATO AS 'CODIGO ESTRATO'
                ,GENESTRATO.GETNOMEST AS 'ESTRATO ECONOMICO'
                ,GEBNOMBRE AS 'BARRIO DONDE VIVE PAC'
                ,ADNCENATE.ACANOMBRE AS 'NOMBRE HOSPITAL ATENCION'
                ,CASE GDETIPREG
                        WHEN 0 THEN 'NINGUNO'
                        WHEN 1 THEN 'CONTRIBUTIVO'
                        WHEN 2 THEN 'SUBSIDIADO'
                        WHEN 3 THEN 'EXCEPCION'
                        WHEN 4 THEN 'ESPECIAL'
                        WHEN 5 THEN 'NO ASEGURADO - VINCULADO'
                        ELSE NULL
                        END AS REGIMEN
                ,CASE HCNINDMED.HCITIPIND  WHEN -1 THEN 'NINGUNA'
```

```
WHEN 0 THEN 'HOSPITALIZACION'

WHEN 1 THEN 'URGENCIAS_OBSERVACION'

WHEN 2 THEN 'CIRUGIA'

WHEN 3 THEN 'REMISION'

WHEN 4 THEN 'MORGUE'

WHEN 5 THEN 'SALIDA'

WHEN 6 THEN 'SALIDA_CONSULTA_EXTERNA'

WHEN 7 THEN 'EN_ESPERA'

WHEN 8 THEN 'CONTROL'

WHEN 9 THEN 'PROCEDIMIENTO'

WHEN 10 THEN 'MEDICAMENTO'

WHEN 11 THEN 'INTERCONSULTA'

WHEN 12 THEN 'APOYO_DIAGNOSTICO'

WHEN 13 THEN 'NO_RESPONDE_EL_LLAMADO'

WHEN 14 THEN 'ESPERA_DE_TERMINACION_DE_TRATAMIENTO'

WHEN 15 THEN 'ESPERA_DE_EVOLUCION_POR_ESPECIALISTA'

        END AS 'INDICACION MEDICA'

                , HCNINDMED.HCITIPIND AS 'TIPO INDI MEDICA'

                ,GENDIAGNO.DIACODIGO AS 'DIAGNOSTICO'

                ,GENDIAGNO.DIANOMBRE AS 'NOMBRE DIAGNOSTICO'

                ,HCTMOTCON AS 'MOTIVO CONSULTA TRIAGE'

                ,HCTTENART AS 'TENSION ARTERIAL TRIAGE'

                ,HCTFRECAR AS 'FRECUENCIA CARDIACA TRIAGE'
```

```
,HCTFRERES AS 'FRECUENCIA RESPIRATORIA TRIAGE'
,HCTTEMPER AS 'TEMPERATURA TRIAGE'
,HCTSO2 AS 'SATURACION OXIGENO TRIAGE'
,HCTOBSERV AS 'OBSERVACION TRIAGE'
,HCTDIABET AS 'DIABETES (0.NO 1.SI)'
,HCTCORONA AS 'ENFERMEDAD CORONARIA (0.NO 1.SI)'
,HCTACV AS 'ACV (0.NO 1.SI)'
,HCTHIPERT AS 'HIPERTENSO (0.NO 1.SI)'
,HCTDROGAS AS 'MEDICAMENTOS'
,HCPACANT AS 'ANTICUAGULADO (0.NO 1.SI)'
,HCTGLUCOM AS 'GLOCOMETRIA (0.NO 1.SI)'
,CAST(ADNINGRESO.AINFECING AS DATE) AS 'FECHA DE
INGRESO'
,HCNPACPESO AS 'PESO'
,HCNPACTALLA AS 'TALLA'
,HCCAMPO09 AS 'OBJETIVO'
,HCCAMPO10 AS 'RESULTADO E INTERPRETACION
PARACLINICOS'
,HCCAMPO11 AS 'ANALISIS Y JUSTIFICACION DE LA
ESTANCIA'
,HCCAMPO01 AS 'FRECUENCIA CARDIACA EVOLUCION'
,HCCAMPO04 AS 'TEMPERATURA EVOLUCION'
,HCCAMPO03 AS 'SATURACION OXIGENO EVOLUCION'
```

```
                    ,HCCAMPO06 AS 'PESO ACTUAL KILOGRAMOS'
                    ,HCCM03N13 AS 'DIAGNOSTICOS'
                    ,HCCAMPO08 AS 'SUBJETIVO'
                    ,HCCM02N16 AS 'GLASGOW'
                    ,CONCAT(HCCM00N24,'/',HCCM00N25) AS 'TENSION
ARTERIAL'
                    ,HCCM01N26 AS 'TENSION ARTERIAL MEDIA'
                    ,HCCM01N27 AS 'INDICE MASA CORPORAL'
                    ,HCPOBSERV AS 'OBSERVACIONES'
                    --,*
    FROM HCNFOLIO WITH(NOLOCK)
    INNER JOIN ADNINGRESO WITH(NOLOCK)ON HCNFOLIO.ADNINGRESO =
ADNINGRESO.OID
    INNER JOIN SLNFACTUR WITH(NOLOCK)ON ADNINGRESO.OID =
SLNFACTUR.ADNINGRESO
    INNER JOIN HCNDIAPAC WITH(NOLOCK)ON HCNFOLIO.OID =
HCNDIAPAC.HCNFOLIO
    INNER JOIN GENDIAGNO WITH(NOLOCK)ON HCNDIAPAC.GENDIAGNO =
GENDIAGNO.OID
    INNER JOIN HCNTIPHIS WITH(NOLOCK)ON HCNFOLIO.HCNTIPHIS =
HCNTIPHIS.OID
    INNER JOIN GENESPECI WITH(NOLOCK)ON HCNFOLIO.GENESPECI =
GENESPECI.OID
```

INNER JOIN GENPACIEN WITH(NOLOCK)ON HCNFOLIO.GENPACIEN = GENPACIEN.OID

INNER JOIN U_PACTIPDOC WITH(NOLOCK)ON GENPACIEN.PACTIPDOC = U_PACTIPDOC.PACTIPDOC

INNER JOIN GENBARRIO WITH(NOLOCK)ON GENBARRIO.OID = GENPACIEN.GENBARRIO

INNER JOIN GENDETCON WITH(NOLOCK)ON GENDETCON.OID = GENPACIEN.GENDETCON

LEFT JOIN ADNGRUETN WITH(NOLOCK)ON ADNGRUETN.OID = GENPACIEN.ADNGRUETN

INNER JOIN HCMHOSEVA WITH(NOLOCK)ON HCMHOSEVA.HCNFOLIO = HCNFOLIO.OID

INNER JOIN ADNCENATE WITH(NOLOCK)ON ADNCENATE.OID = ADNINGRESO.ADNCENATE

INNER JOIN GENPACIEND WITH(NOLOCK)ON GENPACIEND.OID = GENPACIEN.GENPACIEND

INNER JOIN GENPACIENT WITH(NOLOCK)ON GENPACIENT.OID = GENPACIEN.GENPACIENT

INNER JOIN GENCONTRA WITH(NOLOCK)ON GENCONTRA.OID = GENDETCON.GENCONTRA1

INNER JOIN GEENENTADM WITH(NOLOCK)ON GEENENTADM.OID = GENCONTRA.DGNENTADM1

INNER JOIN GENTERCER WITH(NOLOCK)ON GENTERCER.OID = GEENENTADM.GENTERCER1

INNER JOIN GENMEDICO WITH(NOLOCK)ON GENMEDICO.OID = HCNFOLIO.GENMEDICO

INNER JOIN HCNINDMED WITH (NOLOCK)ON HCNFOLIO.HCNINDMED = HCNINDMED.OID

INNER JOIN HCNTRIAGE WITH (NOLOCK)ON ADNINGRESO.HCENTRIAGE = HCNTRIAGE.OID

INNER JOIN GENESTRATO WITH (NOLOCK)ON GENPACIEN.GENESTRATO = GENESTRATO.OID

WHERE ADNINGRESO.AINFECING >= '20190101' AND ADNINGRESO.AINFECING <= '20241231' AND HCNINDMED.HCITIPIND IN (0,1,4,5) AND GENDIAGNO.DIACODIGO IN ('I678','I679')

Figura 1

Consulta Base Datos SQL

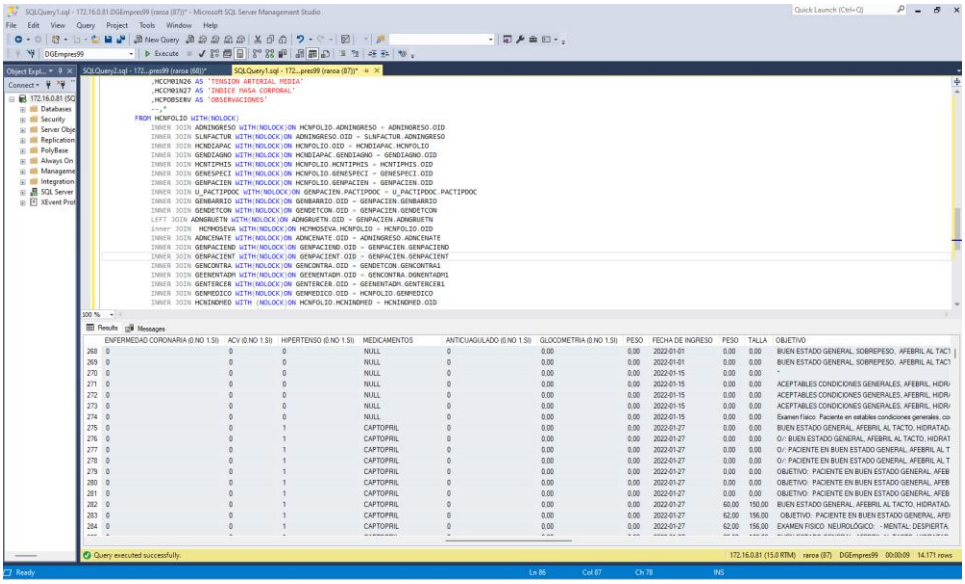
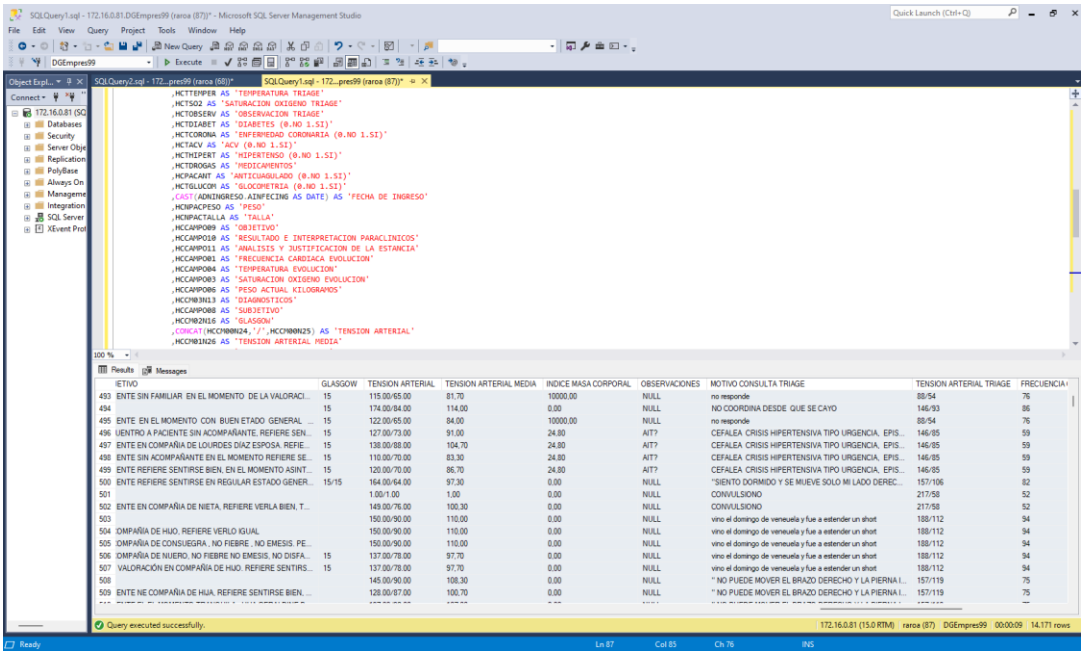


Figura 2

Consulta Base Datos - SQL



Preparación de los Datos

Dentro de la consulta se realizó tres filtros, ADNINGRESO.AINFECING >= '20190101' ADNINGRESO.AINFECING <= '20241231' correspondiente a las fechas de ingreso del paciente a la institución, HCNINDMED.HCITIPIND IN (0,1,4,5) correspondiente a la indicación médica (Hospitalización, Urgencias, Morgue, Salida) GENDIAGNO.DIACODIGO IN ('I678','I679'), correspondiente a diagnósticos CIE 10 “I678 - OTRAS ENFERMEDADES CEREBROVASCULARES ESPECIFICADAS” “I679 - ENFERMEDAD CEREBROVASCULAR, NO ESPECIFICADA”., dentro de la consulta se obtuvo 14.171 registros con el cual se da inicio a la fase de transformación de los datos dándoles un formato adecuado para el entrenamiento del modelo supervisado de ML, dentro de las actividades a realizar se encuentran las siguientes: Limpieza de datos, Codificación de variables categóricas,

Escalamiento o normalización de variables numéricas y Generación de conjuntos de datos de entrenamiento y prueba.

Para el proceso de modelado de ML se utilizó Python como lenguaje de programación para la Ciencia de Datos y Analítica.

En el trabajo con Python se importaron las siguientes librerías:

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns
from imblearn.over_sampling import SMOTE
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import plot_tree
from sklearn.metrics import roc_auc_score, roc_curve
```

Se realizó la lectura del dataset, este dataset se tuvo que trabajar en archivo xlsx por motivo que al exportarlo se tuvo inconvenientes por campos de tipo texto donde habían datos que separados por comas pero estos eran parte de una sola variable, es decir el campo de Observación tenía texto de las observaciones médicas, estos campos generaban conflicto al hacer la lectura del dataset y no se podía dar un manejo de normalización ya que esto implicaría eliminar datos que podían ser fundamentales en el entrenamiento del modelo.

```
df = pd.read_excel('/content/dataset acv 2019-2024.xlsx')
```

Se realizó un escaneo de los datos para validar los tipos de datos que componen el dataset y proceder hacer una limpieza para dar inicio al entrenamiento del modelo ML.

```
print("\nPrimeras 5 filas del DataFrame:")
print(df.head())
print("\nInformación general del DataFrame:")
print(df.info())
print("\nEstadísticas descriptivas de las columnas numéricas:")
print(df.describe())
print("\nValores únicos y conteo de algunas columnas clave para el preprocesamiento:")
print("\n'EDAD PACIENTE':")
print(df['EDAD PACIENTE'].value_counts(dropna=False))
print("\n'SEXO (1 = MASCULINO 2 = FEMENINO)':")
print(df['SEXO (1 = MASCULINO 2 = FEMENINO)'].value_counts(dropna=False))
print("\n'ESTRATO ECONOMICO':")
print(df['ESTRATO ECONOMICO'].value_counts(dropna=False))
print("\n'DIABETES (0.NO 1.SI)':")
print(df['DIABETES (0.NO 1.SI)'].value_counts(dropna=False))
print("\n'ACV (0.NO 1.SI)':")
print(df['ACV (0.NO 1.SI)'].value_counts(dropna=False))
print("\n'TENSION ARTERIAL TRIAGE':")
print(df['TENSION ARTERIAL TRIAGE'].value_counts(dropna=False).head())
print("\n'FRECUENCIA CARDIACA TRIAGE':")
print(df['FRECUENCIA CARDIACA TRIAGE'].value_counts(dropna=False).head())
```

```
print("\n'TEMPERATURA TRIAGE':")
print(df['TEMPERATURA TRIAGE'].value_counts(dropna=False).head())
print("\n'SATURACION OXIGENO TRIAGE':")
print(df['SATURACION OXIGENO TRIAGE'].value_counts(dropna=False).head())
```

En este proceso se obtuvo los siguientes resultados:

Primeras 5 filas del DataFrame:

```
   TIPO DOC PACIENTE  EDAD PACIENTE  SEXO (1 = MASCULINO 2 =
FEMENINO)  \
0               AS             67                        1
1               AS             67                        1
2               AS             71                        2
3               AS             71                        2
4               AS             71                        2

                                ENTIDAD  CODIGO ESTRATO   ESTRATO ECONOMICO  \
0  FONDO FINANCIERO DISTRITAL DE SALUD              14  POBLACION
ESPECIAL
1  FONDO FINANCIERO DISTRITAL DE SALUD              14  POBLACION
ESPECIAL
2                ASMET SALUD EPS SAS               5  SUBSIDIADO NIVEL 1
3                ASMET SALUD EPS SAS               5  SUBSIDIADO NIVEL 1
4                ASMET SALUD EPS SAS               5  SUBSIDIADO NIVEL 1
```

```
  BARRIO DONDE VIVE PAC       NOMBRE HOSPITAL ATENCION  \
0   SCRI-SAN ISIDRO SUR  HOSPITAL OCCIDENTE DE KENNEDY
1   SCRI-SAN ISIDRO SUR  HOSPITAL OCCIDENTE DE KENNEDY
2   SCRI-SAN ISIDRO SUR  HOSPITAL OCCIDENTE DE KENNEDY
3   SCRI-SAN ISIDRO SUR  HOSPITAL OCCIDENTE DE KENNEDY
4   SCRI-SAN ISIDRO SUR  HOSPITAL OCCIDENTE DE KENNEDY

  REGIMEN INDICACION MEDICA  ...  TEMPERATURA EVOLUCION  \
0  NO ASEGURADO - VINCULADO   HOSPITALIZACION  ...                36
1  NO ASEGURADO - VINCULADO   HOSPITALIZACION  ...                 0
2             SUBSIDIADO   HOSPITALIZACION  ...                37
3             SUBSIDIADO   HOSPITALIZACION  ...                37
4             SUBSIDIADO   HOSPITALIZACION  ...                37

  SATURACION OXIGENO EVOLUCION PESO ACTUAL KILOGRAMOS  \
0                  95                0.0
1                   0                0.0
2                  99               85.0
3                  99               85.0
4                  99               85.0

                                DIAGNOSTICOS  \
0  EVOLUCION NEUROCIRUGIA     SEGÚN DECRETO LEGIS...
```

1	EVOLUCIÓN DIARIA NEUROLOGÍA	RESIDENTE DE NE...
2	***NOTA DE EVOLUCIÓN NEUROLOGÍA***	RESIDE...
3	EVOLUCION DE NEUROLOGIA:	RESIDENTE DE NEUR...
4	EVOLUCION DE NEUROLOGIA:	RESIDENTE DE NEUR...

	SUBJETIVO	GLASGOW \
0	PACIETE REIFER SETUIRSE BIEN , CEFALE ALEVE, N...	13
1	PACIENTE EN UNIDAD DE CUIDADOS INTENSIVOS, SIN...	NaN
2	PACIENTE EN EL MOMENTO DE LA VALORACIÓN CON HI...	15/15
3	PACIENTE EN EL MOMENTO DE LA VALORACION CON HI...	15/15
4	PACIENTE EN EL MOMENTO DE LA VALORACION CON HI...	15/15

	TENSION ARTERIAL	TENSION ARTERIAL MEDIA	INDICE MASA CORPORAL \
0	138.00/89.00	105.3	0.00
1	110.00/60.00	76.7	0.00
2	125.00/70.00	88.3	30.12
3	125.00/70.00	88.3	30.12
4	125.00/70.00	88.3	30.12

	OBSERVACIONES
0	NaN
1	NaN

```
2          NaN

3          NaN

4          NaN


[5 rows x 43 columns]


Información general del DataFrame:

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 14134 entries, 0 to 14133

Data columns (total 43 columns):

 #   Column                              Non-Null Count  Dtype

---  ------                              --------------  -----

 0   TIPO DOC PACIENTE                   14134 non-null  object

 1   EDAD PACIENTE                       14134 non-null  int64

 2   SEXO (1 = MASCULINO 2 = FEMENINO)   14134 non-null  int64

 3   ENTIDAD                             14134 non-null  object

 4   CODIGO ESTRATO                      14134 non-null  int64

 5   ESTRATO ECONOMICO                   14134 non-null  object

 6   BARRIO DONDE VIVE PAC               14134 non-null  object

 7   NOMBRE HOSPITAL ATENCION            14134 non-null  object

 8   REGIMEN                             14134 non-null  object

 9   INDICACION MEDICA                   14134 non-null  object

 10  CODIGO INDI MEDICA                  14134 non-null  int64
```

11 DIAGNOSTICO 14134 non-null object

12 NOMBRE DIAGNOSTICO 14134 non-null object

13 MOTIVO CONSULTA TRIAGE 13173 non-null object

14 TENSION ARTERIAL TRIAGE 14134 non-null object

15 FRECUENCIA CARDIACA TRIAGE 14134 non-null int64

16 FRECUENCIA RESPIRATORIA TRIAGE 14134 non-null int64

17 TEMPERATURA TRIAGE 14134 non-null float64

18 SATURACION OXIGENO TRIAGE 14134 non-null int64

19 OBSERVACION TRIAGE 11125 non-null object

20 DIABETES (0.NO 1.SI) 14034 non-null float64

21 ENFERMEDAD CORONARIA (0.NO 1.SI) 14034 non-null float64

22 ACV (0.NO 1.SI) 14034 non-null float64

23 HIPERTENSO (0.NO 1.SI) 14034 non-null float64

24 MEDICAMENTOS 3247 non-null object

25 ANTICUAGULADO (0.NO 1.SI) 14034 non-null float64

26 GLOCOMETRIA (0.NO 1.SI) 14034 non-null float64

27 FECHA DE INGRESO 14034 non-null datetime64[ns]

28 TALLA 11445 non-null float64

29 OBJETIVO 13957 non-null object

30 RESULTADO E INTERPRETACION PARACLINICOS 13890 non-null object

31 ANALISIS Y JUSTIFICACION DE LA ESTANCIA 13981 non-null object

32 FRECUENCIA CARDIACA EVOLUCION 14030 non-null float64

33 TEMPERATURA EVOLUCION 14031 non-null object

```
 34  SATURACION OXIGENO EVOLUCION          14031 non-null  object
 35  PESO ACTUAL KILOGRAMOS                13534 non-null  float64
 36  DIAGNOSTICOS                          14007 non-null  object
 37  SUBJETIVO                             13920 non-null  object
 38  GLASGOW                               6375 non-null   object
 39  TENSION ARTERIAL                      14027 non-null  object
 40  TENSION ARTERIAL MEDIA                14027 non-null  object
 41  INDICE MASA CORPORAL                  14027 non-null  float64
 42  OBSERVACIONES                         786 non-null    object
dtypes: datetime64[ns](1), float64(11), int64(7), object(24)
memory usage: 4.6+ MB
None


Estadísticas descriptivas de las columnas numéricas:
       EDAD PACIENTE  SEXO (1 = MASCULINO 2 = FEMENINO)  CODIGO
ESTRATO  \
count   14134.000000                 14134.000000    14134.000000
mean       70.688552                     1.498585        6.096788
min         6.000000                     1.000000        1.000000
25%        60.000000                     1.000000        5.000000
50%        73.000000                     1.000000        5.000000
75%        83.000000                     2.000000        6.000000
max       105.000000                     2.000000       19.000000
```

std	16.659333	0.500016	3.053772

	CODIGO INDI MEDICA	FRECUENCIA CARDIACA TRIAGE \
count	14134.000000	14134.000000
mean	0.343215	83.061907
min	0.000000	1.000000
25%	0.000000	73.000000
50%	0.000000	80.000000
75%	0.000000	93.000000
max	5.000000	190.000000
std	1.199013	19.539326

	FRECUENCIA RESPIRATORIA TRIAGE	TEMPERATURA TRIAGE \
count	14134.000000	14134.000000
mean	19.425570	35.959492
min	1.000000	0.100000
25%	18.000000	36.000000
50%	19.000000	36.000000
75%	20.000000	36.500000
max	94.000000	41.000000
std	4.700641	2.984943

	SATURACION OXIGENO TRIAGE	DIABETES (0.NO 1.SI) \

count	14134.000000	14034.000000
mean	91.982383	0.036982
min	0.000000	0.000000
25%	91.000000	0.000000
50%	93.000000	0.000000
75%	95.000000	0.000000
max	100.000000	1.000000
std	6.370185	0.188723

	ENFERMEDAD CORONARIA (0.NO 1.SI)	ACV (0.NO 1.SI) \
count	14034.000000	14034.000000
mean	0.007696	0.023016
min	0.000000	0.000000
25%	0.000000	0.000000
50%	0.000000	0.000000
75%	0.000000	0.000000
max	1.000000	1.000000
std	0.087389	0.149958

	HIPERTENSO (0.NO 1.SI)	ANTICUAGULADO (0.NO 1.SI) \
count	14034.000000	14034.000000
mean	0.097905	0.002779
min	0.000000	0.000000

25%	0.000000	0.000000
50%	0.000000	0.000000
75%	0.000000	0.000000
max	1.000000	1.000000
std	0.297197	0.052644

	GLOCOMETRIA (0.NO 1.SI)	FECHA DE INGRESO	TALLA \
count	14034.0	14034	11445.000000
mean	0.0	2022-06-18 08:42:34.938008064	9.071510
min	0.0	2019-01-04 00:00:00	0.000000
25%	0.0	2021-02-24 00:00:00	0.000000
50%	0.0	2022-10-28 00:00:00	0.000000
75%	0.0	2023-12-29 18:00:00	0.000000
max	0.0	2024-12-01 00:00:00	640.000000
std	0.0	NaN	37.608972

	FRECUENCIA CARDIACA EVOLUCION	PESO ACTUAL KILOGRAMOS \
count	14030.000000	13534.000000
mean	76.131803	14.463389
min	0.000000	0.000000
25%	69.000000	0.000000
50%	76.000000	0.000000
75%	83.000000	0.000000

max	7973.000000	168.000000
std	95.906276	28.757272

INDICE MASA CORPORAL

count	14027.000000
mean	3157.211550
min	0.000000
25%	0.000000
50%	0.000000
75%	0.000000
max	750000.000000
std	31696.395307

Valores únicos y conteo de algunas columnas clave para el preprocesamiento:

'EDAD PACIENTE':

EDAD PACIENTE

75	485
68	434
79	428
66	389
80	384

...

6 2

23 1

21 1

19 1

18 1

Name: count, Length: 94, dtype: int64

'SEXO (1 = MASCULINO 2 = FEMENINO)':

SEXO (1 = MASCULINO 2 = FEMENINO)

1 7087

2 7047

Name: count, dtype: int64

'ESTRATO ECONOMICO':

ESTRATO ECONOMICO

SUBSIDIADO NIVEL 1 7865

SUBSIDIADO NIVEL 2 1739

COTIZANTE CATEGORIA A 1221

SUBSIDIADO NIVEL 0 752

VINCULADO TRANSITORIO 584

BENEFICIARIO CATEGORIA A 490

SISBEN NIVEL 1 433

SISBEN NIVEL 2 305

POBLACION ESPECIAL 194

SUBSIDIADO NIVEL 3 137

PARTICULARES 115

SISBEN NIVEL 3 111

COTIZANTE CATEGORIA B 67

BENEFICIARIO CATEGORIA B 44

REGIMEN ESPECIAL 34

NO UTILIZAR 24

COTIZANTE CATEGORIA C 14

BENEFICIARIO CATEGORIA C 5

Name: count, dtype: int64

'DIABETES (0.NO 1.SI)':

DIABETES (0.NO 1.SI)

0.0 13515

1.0 519

NaN 100

Name: count, dtype: int64

'ACV (0.NO 1.SI)':

ACV (0.NO 1.SI)

0.0 13711

1.0 323

NaN 100

Name: count, dtype: int64

'TENSION ARTERIAL TRIAGE':

TENSION ARTERIAL TRIAGE

100/60 113

256/119 91

153/75 87

135/80 84

126/78 68

Name: count, dtype: int64

'FRECUENCIA CARDIACA TRIAGE':

FRECUENCIA CARDIACA TRIAGE

75 865

78 797

85 760

76 431

74 405

Name: count, dtype: int64

'TEMPERATURA TRIAGE':

TEMPERATURA TRIAGE

36.0 7648

36.5 2254

37.0 906

36.2 824

36.4 488

Name: count, dtype: int64

'SATURACION OXIGENO TRIAGE':

SATURACION OXIGENO TRIAGE

95 2152

94 1741

96 1623

92 1584

90 1324

Name: count, dtype: int64

Teniendo en cuenta los resultados se procedió a realizar un análisis de los mismos y efectuar un plan de procesamiento inicial, dentro de este análisis se revisó cada una de las variables determinando lo siguiente:

Variable Objetivo (ACV (0.NO 1.SI)): Tiene valores 0.0, 1.0 y NaN. Se requiere un manejo de los valores NaN (posiblemente eliminarlos ya que son relativamente pocos).

Variable EDAD PACIENTE: Es una variable numérica (int64), lo cual es bueno. No parece haber valores extraños en la salida de value_counts().

Variable SEXO (1 = MASCULINO 2 = FEMENINO): Es una variable numérica (int64) con los valores esperados (1 y 2). Esta variable se puede mantener así o recodificarla a 0 y 1 para algunos modelos si es necesario.

Variable ESTRATO ECONOMICO: Es una variable categórica (object) con varias categorías. Se hace necesario aplicar one-hot encoding.

Variable DIABETES (0.NO 1.SI): Es una variable float64 con valores 0.0, 1.0 y NaN, al igual que con la variable objetivo, manejaremos los NaN posiblemente eliminándolos.

Variables ENFERMEDAD CORONARIA (0.NO 1.SI), HIPERTENSO (0.NO 1.SI), ANTICUAGULADO (0.NO 1.SI), GLOCOMETRIA (0.NO 1.SI):
Son variables de tipo float64 con valores 0.0, 1.0 y NaN, también requerirán manejo de los NaN.

Variable TENSION ARTERIAL TRIAGE: Es object y contiene valores como "100/60", "256/119", etc., para usar esta información es necesario dividir la cadena por "/", convertir las partes a tipo numérico (presión sistólica y diastólica) en este ejercicio es posible crear dos nuevas columnas.

Variable FRECUENCIA CARDIACA TRIAGE: Es una variable numérica (int64). Aunque se evidencia valores extremos como 1 lo que podrían requerir investigación por ese tipo de valor en una frecuencia Cardiaca.

Variable TEMPERATURA TRIAGE: Es una variable numérica (float64), los valores parecen razonables.

Variable SATURACION OXIGENO TRIAGE: Es variable numérica (int64), los valores parecen estar dentro de un rango normal.

Otras Variables: Las demás variables (textuales, identificadores, fechas, etc.) no parecen ser directamente relevantes para el objetivo de identificar asociaciones demográficas y factores

de riesgo binarios con el ACV en este momento. No se descartan ya que quizás más adelante se podría reconsiderar para explorar otras hipótesis.

Para el manejo de NaN en las variables claves, se determina eliminar las filas donde la variable objetivo (ACV) o los factores de riesgo binarios tienen valores faltantes para asegurar que tengamos datos completos para el modelado. Por otra parte se determina recodificar la variable SEXO a 0 (masculino) y 1 (femenino) y eliminar la columna original.

Se hace uso de One-Hot Encoding a la variable de ESTRATO ECONOMICO: Utilizando pd.get_dummies() para convertir la variable categórica ESTRATO ECONOMICO en múltiples columnas binarias (una por cada categoría), el prefijo 'ESTRATO' ayuda a identificar estas nuevas columnas.

Para el manejo de la variable TENSION ARTERIAL TRIAGE: Definimos una función extract_blood_pressure para dividir la cadena por "/" y convertir las partes a números, al aplicar esta función a la columna se crean dos nuevas columnas: SISTOLICA_TRIAGE y DIASTOLICA_TRIAGE y se elimina la columna original TENSION ARTERIAL TRIAGE.

De acuerdo a la estandarización y manejo de las variables se procede a realizar los siguientes pasos para el procesamiento de los datos del dataframe:

Selección de Variables para el Modelo: Creamos una lista features con las columnas que vamos a utilizar como predictoras en nuestro modelo inicial, incluyen las variables demográficas (edad, sexo, estrato económico codificado), los factores de riesgo binarios, y las variables de triage numéricas (frecuencia cardíaca, temperatura, saturación de oxígeno, y la presión arterial extraída).

Creación del DataFrame para el Modelo: Creamos un nuevo DataFrame df_model que contiene solo las características seleccionadas y la variable objetivo, se elimina cualquier fila que aún tenga valores NaN en estas columnas seleccionadas.

```
# Manejar los valores NaN en la variable objetivo y los factores de riesgo binarios
df_cleaned = df.dropna(subset=['ACV (0.NO 1.SI)', 'DIABETES (0.NO 1.SI)',
                               'ENFERMEDAD CORONARIA (0.NO 1.SI)',
                               'HIPERTENSO (0.NO 1.SI)', 'ANTICUAGULADO (0.NO 1.SI)',
                               'GLOCOMETRIA (0.NO 1.SI)'])

print(f"Tamaño del DataFrame después de eliminar NaN en variables clave: {df_cleaned.shape}")

# Codificar la variable 'SEXO' a 0 y 1
df_cleaned['SEXO'] = df_cleaned['SEXO (1 = MASCULINO 2 = FEMENINO)'].replace({1: 0, 2: 1})
df_cleaned = df_cleaned.drop(columns=['SEXO (1 = MASCULINO 2 = FEMENINO)'])

# Codificar la variable categórica 'ESTRATO ECONOMICO' usando one-hot encoding
df_encoded = pd.get_dummies(df_cleaned, columns=['ESTRATO ECONOMICO'], prefix='ESTRATO')
print(f"Tamaño del DataFrame después de one-hot encoding de 'ESTRATO ECONOMICO': {df_encoded.shape}")
```

```
        # Preprocesar 'TENSION ARTERIAL TRIAGE' para extraer presión sistólica y
diastólica
        def extract_blood_pressure(tension):
            try:
                systolic, diastolic = map(float, tension.split('/'))
                return systolic, diastolic
            except:
                return None, None

        df_encoded[['SISTOLICA_TRIAGE', 'DIASTOLICA_TRIAGE']] =
df_encoded['TENSION ARTERIAL TRIAGE'].apply(lambda x:
pd.Series(extract_blood_pressure(x)))
        df_encoded = df_encoded.drop(columns=['TENSION ARTERIAL TRIAGE'])
        print(f"Tamaño del DataFrame después de procesar 'TENSION ARTERIAL TRIAGE':
{df_encoded.shape}")

        # Seleccionar las variables que usaremos para el modelo inicial
        features = ['EDAD PACIENTE', 'SEXO', 'ESTRATO_BENEFICIARIO CATEGORIA
A',
                    'ESTRATO_BENEFICIARIO CATEGORIA B', 'ESTRATO_BENEFICIARIO
CATEGORIA C',
                    'ESTRATO_COTIZANTE CATEGORIA A', 'ESTRATO_COTIZANTE
CATEGORIA B',
```

```
        'ESTRATO_COTIZANTE CATEGORIA C', 'ESTRATO_NO UTILIZAR',
        'ESTRATO_PARTICULARES', 'ESTRATO_POBLACION ESPECIAL',
        'ESTRATO_REGIMEN ESPECIAL', 'ESTRATO_SISBEN NIVEL 1',
        'ESTRATO_SISBEN NIVEL 2', 'ESTRATO_SISBEN NIVEL 3',
        'ESTRATO_SUBSIDIADO NIVEL 0', 'ESTRATO_SUBSIDIADO NIVEL 1',
        'ESTRATO_SUBSIDIADO NIVEL 2', 'ESTRATO_SUBSIDIADO NIVEL 3',
        'ESTRATO_VINCULADO TRANSITORIO', 'DIABETES (0.NO 1.SI)',
        'ENFERMEDAD CORONARIA (0.NO 1.SI)', 'HIPERTENSO (0.NO 1.SI)',
        'ANTICUAGULADO (0.NO 1.SI)', 'GLOCOMETRIA (0.NO 1.SI)',
        'FRECUENCIA CARDIACA TRIAGE', 'TEMPERATURA TRIAGE',
        'SATURACION OXIGENO TRIAGE', 'SISTOLICA_TRIAGE',
'DIASTOLICA_TRIAGE']
    target = 'ACV (0.NO 1.SI)'

    df_model = df_encoded[features + [target]].copy()
    df_model = df_model.dropna()

    print(f"Tamaño del DataFrame listo para el modelo: {df_model.shape}")
    print("\nPrimeras filas del DataFrame listo para el modelo:")
    print(df_model.head())
```

Una vez ejecutado el plan de limpieza y tratamiento de los datos se obtuvieron los siguientes resultados:

Tamaño del DataFrame después de eliminar NaN en variables clave: (14034, 43)

Tamaño del DataFrame después de one-hot encoding de 'ESTRATO ECONOMICO': (14034, 60)

df_cleaned['SEXO'] = df_cleaned['SEXO (1 = MASCULINO 2 = FEMENINO)'].replace({1: 0, 2: 1})

Tamaño del DataFrame después de procesar 'TENSION ARTERIAL TRIAGE': (14034, 61)

Tamaño del DataFrame listo para el modelo: (14034, 31)

Primeras filas del DataFrame listo para el modelo:

	EDAD PACIENTE	SEXO	ESTRATO_BENEFICIARIO CATEGORIA A	ESTRATO_BENEFICIARIO CATEGORIA B	ESTRATO_BENEFICIARIO CATEGORIA C	ESTRATO_COTIZANTE CATEGORIA A	ESTRATO_COTIZANTE CATEGORIA B
0	67	0	False	False	False	False	False
1	67	0	False	False	False	False	False
2	71	1	False	False	False	False	False
3	71	1	False	False	False	False	False
4	71	1	False	False	False	False	False

```
   ESTRATO_COTIZANTE CATEGORIA C  ESTRATO_NO UTILIZAR
ESTRATO_PARTICULARES  \
0                 False          False          False
1                 False          False          False
2                 False          False          False
3                 False          False          False
4                 False          False          False
    ...  ENFERMEDAD CORONARIA (0.NO 1.SI)  HIPERTENSO (0.NO 1.SI)  \
0  ...                   0.0                 0.0
1  ...                   0.0                 0.0
2  ...                   0.0                 0.0
3  ...                   0.0                 0.0
4  ...                   0.0                 0.0
   ANTICUAGULADO (0.NO 1.SI)  GLOCOMETRIA (0.NO 1.SI)  \
0                        0.0                      0.0
1                        0.0                      0.0
2                        0.0                      0.0
3                        0.0                      0.0
4                        0.0                      0.0
   FRECUENCIA CARDIACA TRIAGE  TEMPERATURA TRIAGE  SATURACION OXIGENO
TRIAGE  \
0                         65                36.2                  98
1                         65                36.2                  98
2                        111                36.0                  97
3                        111                36.0                  97
4                        111                36.0                  97
```

	SISTOLICA_TRIAGE	DIASTOLICA_TRIAGE	ACV (0.NO 1.SI)
0	170.0	88.0	0.0
1	170.0	88.0	0.0
2	170.0	127.0	0.0
3	170.0	127.0	0.0
4	170.0	127.0	0.0

[5 rows x 31 columns]

Resumen del Procesamiento

Se eliminaron las filas con valores NaN en las variables clave (variable objetivo y factores de riesgo binarios), el tamaño del DataFrame quedo en 14034 filas, la variable SEXO se recodificó a 0 y 1, la variable categórica ESTRATO ECONOMICO se codificó utilizando one-hot encoding lo que aumentó el número de columnas a 60 en el DataFrame intermedio (df_encoded), se extrajeron las presiones sistólica y diastólica de la columna TENSION ARTERIAL TRIAGE creando dos nuevas columnas numéricas.

Finalmente se seleccionaron las variables relevantes para el modelo incluyendo las nuevas columnas de presión arterial y las columnas de estrato codificadas, resultando en un DataFrame df_model de 14034 filas y 31 columnas (30 características y la variable objetivo ACV (0.NO 1.SI)).

Modelamiento Supervisado de Machine Learning

Se realiza el entrenamiento del modelo Regresión Logística el cual es el más apropiado para la implementación, teniendo en cuenta el tipo de datos obtenidos de las Historias Clínicas, al tener varias variables objetivos con valores binarios nos descarta la posibilidad de realizar una regresión lineal o regresión múltiple, para el entrenamiento del modelo de Regresión Logística se

trabaja con un dataframe 80% de entrenamiento y un 20% de prueba, lo que nos permitirá entrenar nuestro modelo en una parte de los datos y luego evaluar su rendimiento en datos no vistos para tener una estimación de cómo generalizará el modelo a nueva información.

Algoritmo Regresión Logística.

```
X = df_model.drop(columns=[target])
y = df_model[target]

# Dividimos los datos en conjuntos de entrenamiento y prueba (80% entrenamiento, 20% prueba)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42, stratify=y)
print(f"Tamaño del conjunto de entrenamiento: {X_train.shape}")
print(f"Tamaño del conjunto de prueba: {X_test.shape}")

# Iniciar y entrenar el modelo de Regresión Logística
model_logistic = LogisticRegression(solver='liblinear', random_state=42)
model_logistic.fit(X_train, y_train)

# Realizar predicciones en el conjunto de prueba
y_pred_logistic = model_logistic.predict(X_test)

# Evaluamos el rendimiento del modelo
accuracy_logistic = accuracy_score(y_test, y_pred_logistic)
```

```
print(f"\nExactitud del modelo de Regresión Logística en el conjunto de prueba: {accuracy_logistic:.4f}")

print("\nReporte de Clasificación del modelo de Regresión Logística:")
print(classification_report(y_test, y_pred_logistic))

# Matriz de confusión
cm_logistic = confusion_matrix(y_test, y_pred_logistic)
plt.figure(figsize=(8, 6))
sns.heatmap(cm_logistic, annot=True, fmt='d', cmap='Blues',
            xticklabels=['No ACV', 'ACV'], yticklabels=['No ACV', 'ACV'])
plt.xlabel('Predicción')
plt.ylabel('Valor Real')
plt.title('Matriz de Confusión (Regresión Logística)')
plt.show()
```

Una vez realizada la implementación del modelo se obtuvo los siguientes resultados:

Tamaño del conjunto de entrenamiento: (11227, 30)

Tamaño del conjunto de prueba: (2807, 30)

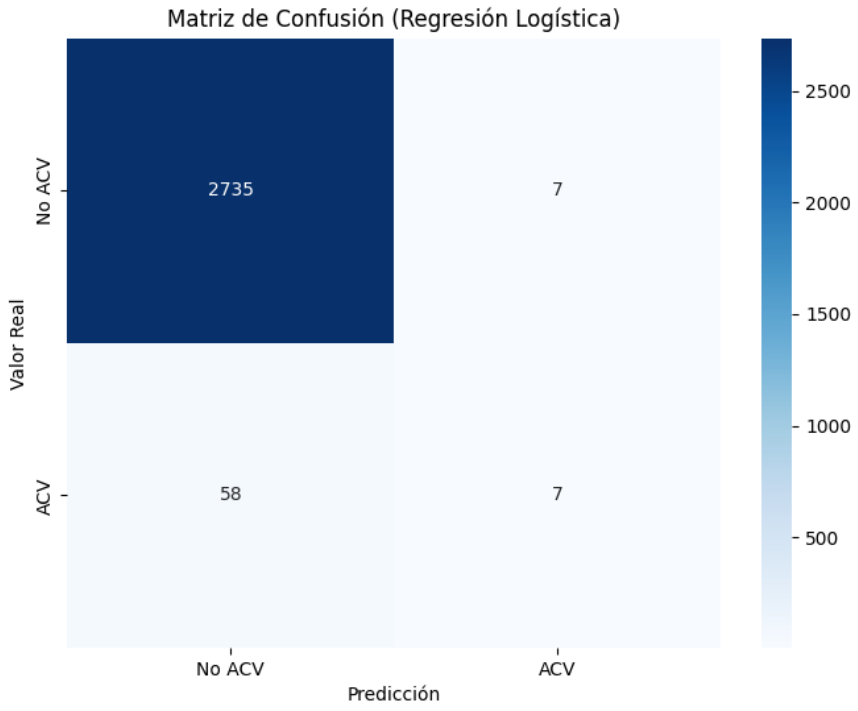
Exactitud del modelo de Regresión Logística en el conjunto de prueba: 0.9768

Reporte de Clasificación del modelo de Regresión Logística:

	precision	recall	f1-score	support
0.0	0.98	1.00	0.99	2742
1.0	0.50	0.11	0.18	65
accuracy			0.98	2807
macro avg		0.74	0.55	0.58	2807
weighted avg	0.97	0.98	0.97	2807

Figura 3

Matriz de Confusión



Análisis de los Resultados

Exactitud (Accuracy): 0.9768: La exactitud es muy alta, lo que indica que el modelo predice correctamente la clase mayoritaria (No ACV) en la mayoría de los casos, sin embargo se observa que esta métrica puede ser engañosa cuando la variable objetivo está desbalanceada.

Reporte de Clasificación:

Clase 0 (No ACV) - Precisión (Precisión): 0.98, de todos los pacientes que el modelo predijo que no tuvieron un ACV el 98% realmente no lo tuvo, Recall (Sensibilidad): 1.00, de todos los pacientes que realmente no tuvieron un ACV, el modelo identificó correctamente al 100%. F1-score: 0.99, es un puntaje alto, esto nos indica un buen equilibrio entre precisión y recall para la clase No ACV.

Clase 1 (ACV) - Precisión (Precisión): 0.50, de todos los pacientes que el modelo predijo que tuvieron un ACV, solo el 50% realmente lo tuvo. Esto significa que el modelo tuvo una cantidad considerable de falsos positivos donde predijo ACV cuando no ocurrió, Recall (Sensibilidad): 0.11, de todos los pacientes que realmente tuvieron un ACV (65 casos en el conjunto de prueba), el modelo solo identificó correctamente al 11% equivalente aproximado a 7 casos, esto nos indica que el modelo tiene una baja sensibilidad para detectar los casos de ACV, F1-score: 0.18, es un puntaje muy bajo, lo que refleja el desequilibrio entre la precisión y el recall para la clase ACV.

Matriz de Confusión:

Verdaderos Negativos (TN) = 2735: El modelo predijo correctamente que 2735 pacientes no tuvieron un ACV, Falsos Positivos (FP) = 7: El modelo predijo incorrectamente que 7 pacientes tuvieron un ACV cuando en realidad no lo tuvieron, Falsos Negativos (FN) = 58: El modelo predijo incorrectamente que 58 pacientes no tuvieron un ACV cuando en realidad sí lo

tuvieron,. Verdaderos Positivos (TP) = 7: El modelo predijo correctamente que 7 pacientes tuvieron un ACV.

Aunque el modelo tiene una alta exactitud general, su bajo recall para la clase ACV sugiere que puede no estar capturando bien los patrones asociados con la ocurrencia de ACV, en este caso el modelo tiende a predecir la clase mayoritaria.

Para cuantificar la fuerza de las asociaciones, es necesario analizar los coeficientes del modelo de regresión logística, estos nos indicarán la dirección y la magnitud del impacto de cada variable predictora en la probabilidad de tener un ACV.

Análisis de coeficientes del modelo regresión logística

```
# Obtener los coeficientes del modelo
coefficients = model_logistic.coef_[0]
# Obtener los nombres de las características
feature_names = X_train.columns
# Crear un DataFrame para mostrar los coeficientes y sus nombres
df_coefficients = pd.DataFrame({'Feature': feature_names, 'Coefficient': coefficients})
# Calcular los odds ratios
df_coefficients['Odds Ratio'] = np.exp(df_coefficients['Coefficient'])
# Ordenar el DataFrame por la magnitud del odds ratio
df_coefficients = df_coefficients.sort_values(by='Odds Ratio', ascending=False)
print("\nCoeficientes y Odds Ratios del Modelo de Regresión Logística:")
print(df_coefficients)
```

Coeficientes y Odds Ratios del Modelo de Regresión Logística:

	Feature	Coefficient	Odds Ratio
22	HIPERTENSO (0.NO 1.SI)	2.281408	9.790451
21	ENFERMEDAD CORONARIA (0.NO 1.SI)	2.185341	8.893680
18	ESTRATO_SUBSIDIADO NIVEL 3	1.260066	3.525654
5	ESTRATO_COTIZANTE CATEGORIA A	1.037786	2.822961
10	ESTRATO_POBLACION ESPECIAL	0.734520	2.084482
2	ESTRATO_BENEFICIARIO CATEGORIA A	0.599110	1.820498
19	ESTRATO_VINCULADO TRANSITORIO	0.539984	1.715979
9	ESTRATO_PARTICULARES	0.219985	1.246059
29	DIASTOLICA_TRIAGE	0.007799	1.007830
26	TEMPERATURA TRIAGE	0.007652	1.007681
25	FRECUENCIA CARDIACA TRIAGE	0.007124	1.007150
24	GLOCOMETRIA (0.NO 1.SI)	0.000000	1.000000
0	EDAD PACIENTE	-0.004904	0.995108
27	SATURACION OXIGENO TRIAGE	-0.020115	0.980086
28	SISTOLICA_TRIAGE	-0.020367	0.979839
4	ESTRATO_BENEFICIARIO CATEGORIA C	-0.109367	0.896401
11	ESTRATO_REGIMEN ESPECIAL	-0.131905	0.876424
7	ESTRATO_COTIZANTE CATEGORIA C	-0.136812	0.872134
16	ESTRATO_SUBSIDIADO NIVEL 1	-0.200352	0.818443
23	ANTICUAGULADO (0.NO 1.SI)	-0.213710	0.807582
17	ESTRATO_SUBSIDIADO NIVEL 2	-0.262877	0.768836

6	ESTRATO_COTIZANTE CATEGORIA B	-0.282840	0.753640
3	ESTRATO_BENEFICIARIO CATEGORIA B	-0.319112	0.726794
8	ESTRATO_NO UTILIZAR	-0.347141	0.706706
14	ESTRATO_SISBEN NIVEL 3	-0.362412	0.695996
1	SEXO	-0.468833	0.625732
20	DIABETES (0.NO 1.SI)	-0.660162	0.516768
12	ESTRATO_SISBEN NIVEL 1	-0.678227	0.507516
13	ESTRATO_SISBEN NIVEL 2	-0.975741	0.376913
15	ESTRATO_SUBSIDIADO NIVEL 0	-1.172911	0.309465

Interpretación de los Resultados:

Las variables con odds ratios mayores que 1 se asocian con una mayor probabilidad de ACV, mientras que las variables con odds ratios menores que 1 se asocian con una menor probabilidad de ACV.

Factores de riesgo con mayor asociación (Odds Ratio > 1):

HIPERTENSO (0.NO 1.SI) (Odds Ratio = 9.79): Ser hipertenso se asocia con un aumento de aproximadamente 9.79 veces en las chances (odds) de haber tenido un ACV, manteniendo otras variables constantes, esta es una asociación muy fuerte y consistente con el conocimiento médico.

ENFERMEDAD CORONARIA (0.NO 1.SI) (Odds Ratio = 8.89): Tener enfermedad coronaria se asocia con un aumento de aproximadamente 8.89 veces en las chances de haber tenido un ACV, esta también es una asociación fuerte y esperada.

ESTRATO_SUBSIDIADO NIVEL 3 (Odds Ratio = 3.53): Pertenecer al estrato subsidiado nivel 3 se asocia con un aumento de aproximadamente 3.53 veces en las chances de

haber tenido un ACV en comparación con la categoría de referencia (que es implícita en la codificación one-hot, y sería la categoría omitida para evitar multicolinealidad).

ESTRATO_COTIZANTE CATEGORIA A (Odds Ratio = 2.82): Ser cotizante de categoría A se asocia con un aumento de aproximadamente 2.82 veces en las chances de haber tenido un ACV en comparación con la categoría de referencia, esta variable no se considera como predominante en las causas teniendo en cuenta que es una variable de asociación a un régimen de categoría de afiliación.

ESTRATO_POBLACION ESPECIAL (Odds Ratio = 2.08): Pertenecer a la población especial se asocia con un aumento de aproximadamente 2.08 veces en las chances de haber tenido un ACV en comparación con la categoría de referencia.

ESTRATO_BENEFICIARIO CATEGORIA A (Odds Ratio = 1.82): Ser beneficiario de categoría A se asocia con un aumento de aproximadamente 1.82 veces en las chances de haber tenido un ACV en comparación con la categoría de referencia.

ESTRATO_VINCULADO TRANSITORIO (Odds Ratio = 1.72): Estar en el régimen vinculado transitorio se asocia con un aumento de aproximadamente 1.72 veces en las chances de haber tenido un ACV en comparación con la categoría de referencia.

ESTRATO_PARTICULARES (Odds Ratio = 1.25): Ser paciente particular se asocia con un aumento de aproximadamente 1.25 veces en las chances de haber tenido un ACV en comparación con la categoría de referencia.

DIASTOLICA_TRIAGE (Odds Ratio = 1.01): Un aumento de una unidad en la presión diastólica en el triage se asocia con un ligero aumento (1.01 veces) en las chances de haber tenido un ACV.

TEMPERATURA TRIAGE (Odds Ratio = 1.01): Un aumento de un grado en la temperatura en el triage se asocia con un ligero aumento (1.01 veces) en las chances de haber tenido un ACV.

FRECUENCIA CARDIACA TRIAGE (Odds Ratio = 1.01): Un aumento de un latido por minuto en la frecuencia cardíaca en el triage se asocia con un ligero aumento (1.01 veces) en las chances de haber tenido un ACV.

GLOCOMETRIA (0.NO 1.SI) (Odds Ratio = 1.00): Esta variable no muestra asociación según el modelo (el odds no cambian).

Factores con Menor Asociación (Odds Ratio < 1):

ESTRATO_SUBSIDIADO NIVEL 0 (Odds Ratio = 0.31): Pertenecer al estrato subsidiado nivel 0 se asocia con una disminución significativa aproximadamente 0.31 veces las chances de haber tenido un ACV en comparación con la categoría de referencia.

ESTRATO_SISBEN NIVEL 2 (Odds Ratio = 0.38): Pertenecer al SISBEN nivel 2 se asocia con una disminución significativa aproximadamente 0.38 veces las chances de haber tenido un ACV en comparación con la categoría de referencia.

ESTRATO_SISBEN NIVEL 1 (Odds Ratio = 0.51): Pertenecer al SISBEN nivel 1 se asocia con una disminución aproximadamente 0.51 veces las chances de haber tenido un ACV en comparación con la categoría de referencia.

DIABETES (0.NO 1.SI) (Odds Ratio = 0.52): Tener diabetes se asocia con una disminución aproximadamente 0.52 veces las chances de haber tenido un ACV según este modelo. Esta es una asociación inesperada y podría ser resultado del desbalance de clases o la influencia de otras variables.

SEXO (Odds Ratio = 0.63): Ser mujer (SEXO = 1) se asocia con una disminución aproximadamente 0.63 veces las chances de haber tenido un ACV en comparación con ser hombre (SEXO = 0).

ESTRATO_SUBSIDIADO NIVEL 2 (Odds Ratio = 0.77): Pertenecer al estrato subsidiado nivel 2 se asocia con una disminución aproximadamente 0.77 veces las chances de haber tenido un ACV en comparación con la categoría de referencia.

ESTRATO_COTIZANTE CATEGORIA B (Odds Ratio = 0.75): Ser cotizante de categoría B se asocia con una ligera disminución aproximadamente 0.75 veces las chances de haber tenido un ACV en comparación con la categoría de referencia.

ESTRATO_BENEFICIARIO CATEGORIA B (Odds Ratio = 0.73): Ser beneficiario de categoría B se asocia con una ligera disminución aproximadamente 0.73 veces las chances de haber tenido un ACV en comparación con la categoría de referencia.

ESTRATO_NO UTILIZAR (Odds Ratio = 0.71): La categoría "No Utilizar" en el estrato económico se asocia con una disminución aproximadamente 0.71 veces las chances de haber tenido un ACV en comparación con la categoría de referencia.

ESTRATO_SISBEN NIVEL 3 (Odds Ratio = 0.70): Pertenecer al SISBEN nivel 3 se asocia con una ligera disminución aproximadamente 0.70 veces las chances de haber tenido un ACV en comparación con la categoría de referencia.

ESTRATO_REGIMEN ESPECIAL (Odds Ratio = 0.88): Pertenecer al régimen especial se asocia con una ligera disminución aproximadamente 0.88 veces las chances de haber tenido un ACV en comparación con la categoría de referencia.

ESTRATO_COTIZANTE CATEGORIA C (Odds Ratio = 0.87): Ser cotizante de categoría C se asocia con una ligera disminución aproximadamente 0.87 veces las chance de haber tenido un ACV en comparación con la categoría de referencia.

ESTRATO_BENEFICIARIO CATEGORIA C (Odds Ratio = 0.90): Ser beneficiario de categoría C se asocia con una ligera disminución aproximadamente 0.90 veces las chances de haber tenido un ACV en comparación con la categoría de referencia.

SATURACION OXIGENO TRIAGE (Odds Ratio = 0.98): Un aumento de un punto porcentual en la saturación de oxígeno en el triage se asocia con una ligera disminución aproximadamente 0.98 veces las chances de haber tenido un ACV.

SISTOLICA_TRIAGE (Odds Ratio = 0.98): Un aumento de una unidad en la presión sistólica en el triage se asocia con una ligera disminución aproximadamente 0.98 veces las chances de haber tenido un ACV.

EDAD PACIENTE (Odds Ratio = 0.995): Un aumento de un año en la edad se asocia con una ligera disminución aproximadamente 0.995 veces las chances de haber tenido un ACV, esta es otra asociación inesperada, ya que la edad es un factor de riesgo conocido para el ACV.

De acuerdo a estos resultados se debe hacer un ajuste al modelo por el desbalance de las clases, se aborda el método SMOTE (Synthetic Minority Over-sampling Technique), generando instancias sintéticas minoritarias en lugar de duplicar las existentes, lo que puede ayudar a prevenir el sobreajuste.

```
!pip install imbalanced-learn
X = df_model.drop(columns=[target])
y = df_model[target]
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42, stratify=y)

# Aplicar SMOTE al conjunto de entrenamiento
smote = SMOTE(random_state=42)
X_train_smote, y_train_smote = smote.fit_resample(X_train, y_train)

print(f"Tamaño del conjunto de entrenamiento original: {X_train.shape}")
print(f"Tamaño del conjunto de entrenamiento después de SMOTE: {X_train_smote.shape}")
print(f"Proporción de la clase ACV en el entrenamiento original: {y_train.value_counts(normalize=True)[1]:.4f}")
print(f"Proporción de la clase ACV en el entrenamiento después de SMOTE: {y_train_smote.value_counts(normalize=True)[1]:.4f}")

# Iniciar y entrenar el modelo de Regresión Logística con los datos sobremuestreados
model_logistic_smote = LogisticRegression(solver='liblinear', random_state=42)
model_logistic_smote.fit(X_train_smote, y_train_smote)

# Realizar predicciones en el conjunto de prueba usando el modelo entrenado con SMOTE
y_pred_logistic_smote = model_logistic_smote.predict(X_test)
```

```
# Evaluar el rendimiento del modelo con SMOTE
accuracy_logistic_smote = accuracy_score(y_test, y_pred_logistic_smote)
print(f"\nExactitud del modelo de Regresión Logística con SMOTE en el conjunto de prueba: {accuracy_logistic_smote:.4f}")

print("\nReporte de Clasificación del modelo de Regresión Logística con SMOTE:")
print(classification_report(y_test, y_pred_logistic_smote))

# Visualizar la matriz de confusión del modelo con SMOTE
cm_logistic_smote = confusion_matrix(y_test, y_pred_logistic_smote)
plt.figure(figsize=(8, 6))
sns.heatmap(cm_logistic_smote, annot=True, fmt='d', cmap='Blues',
            xticklabels=['No ACV', 'ACV'], yticklabels=['No ACV', 'ACV'])
plt.xlabel('Predicción')
plt.ylabel('Valor Real')
plt.title('Matriz de Confusión (Regresión Logística con SMOTE)')
plt.show()
```

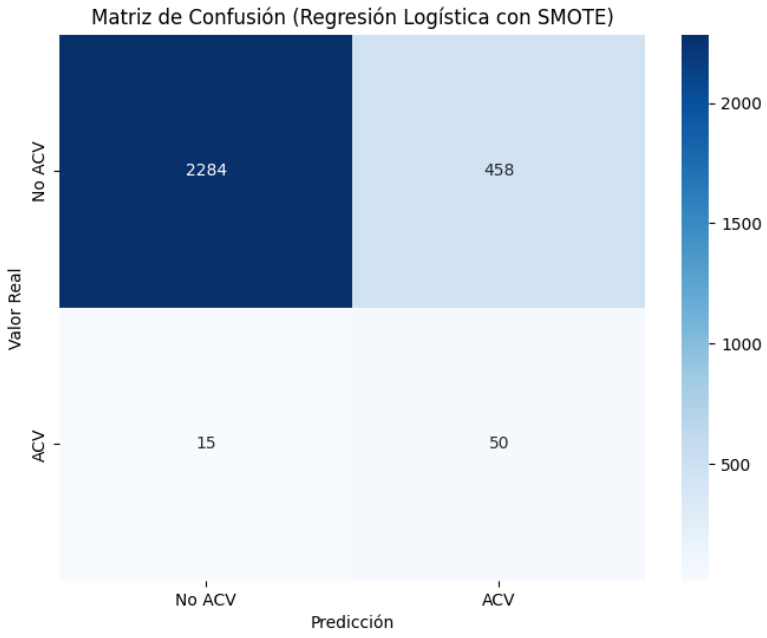
```
Tamaño del conjunto de entrenamiento original: (11227, 30)
Tamaño del conjunto de entrenamiento después de SMOTE: (21938, 30)
Proporción de la clase ACV en el entrenamiento original: 0.0230
Proporción de la clase ACV en el entrenamiento después de SMOTE: 0.5000
Exactitud del modelo de Regresión Logística con SMOTE en el conjunto de prueba: 0.8315
```

Reporte de Clasificación del modelo de Regresión Logística con SMOTE:

	precision	recall	f1-score	support
0.0	0.99	0.83	0.91	2742
1.0	0.10	0.77	0.17	65
accuracy			0.83	2807
macro avg	0.55	0.80	0.54	2807
weighted avg	0.97	0.83	0.89	2807

Figura 4

Matriz de Confusión con SMOTE



Análisis de los Resultados con SMOTE:

El tamaño del conjunto de entrenamiento aumentó significativamente después de aplicar SMOTE, pasando de 11227 a 21938 muestras. La proporción de la clase ACV en el conjunto de entrenamiento ahora es del 50%, lo que indica que SMOTE logró equilibrar las clases en el entrenamiento.

Exactitud (Accuracy): 0.8315: La exactitud disminuyó significativamente del 97.68% al 83.15%, este comportamiento es común cuando se intenta mejorar la detección de la clase minoritaria ya que el modelo ahora está haciendo más predicciones de la clase positiva, lo que puede llevar a más errores en la predicción de la clase negativa.

Reporte de Clasificación:

Clase 0 (No ACV): Precisión (Precisión) = 0.99, Se mantuvo alta. Recall (Sensibilidad): 0.83 disminuyó del 1.00, lo que significa que ahora el modelo está identificando correctamente el 83% de los casos reales de No ACV donde antes era el 100%, pero también está clasificando incorrectamente algunos casos de No ACV como ACV falsos positivos, F1-score: 0.91, disminuyó ligeramente.

Clase 1 (ACV): Precisión (Precisión): 0.10 disminuyó drásticamente del 0.50 al 0.10, esto nos indica que de todos los pacientes que el modelo predijo que tuvieron un ACV solo el 10% realmente lo tuvo, el número de falsos positivos aumentó considerablemente, Recall (Sensibilidad): 0.77 aumentó significativamente del 0.11 al 0.77, esto es el resultado directo del sobre muestreo donde el modelo ahora está identificando correctamente el 77% de los casos reales de ACV antes solo el 11%, F1-score: 0.17 se mantuvo bajo similar a 0.18 antes de SMOTE, aunque el recall mejoró mucho, la gran caída en la precisión limitó la mejora en el F1-score.

Matriz de Confusión:

Verdaderos Negativos (TN): 2284, disminuyó antes 2735, Falsos Positivos (FP): 458, aumentó significativamente antes 7, Falsos Negativos (FN): 15, disminuyó drásticamente antes 58, Verdaderos Positivos (TP): 50, aumentó significativamente antes 7.

Con SMOTE se logró aumentar significativamente la sensibilidad (recall) del modelo para la clase ACV, aunque se detecta una proporción mucho mayor de los casos reales de ACV, sin embargo la disminución en la precisión para la clase ACV indica que el modelo está generando muchos más falsos positivos, el F1-score para la clase ACV no mejoró sustancialmente indicando que el equilibrio entre precisión y recall sigue siendo un desafío por el desajuste.

Por otra parte el aumento en el recall (sensibilidad) que se obtuvo con SMOTE fue positivo ya que significa que se está identificando una proporción mucho mayor de los pacientes que realmente tuvieron un ACV, Sin embargo la baja precisión sigue siendo de preocupación ya que implica que se está generando muchos falsos positivos lo que podría llevar a evaluaciones innecesarias.

Teniendo en cuenta que el modelo aún no tiene una buena predicción es necesario realizar una implementación de ajuste de Pesos de Clase para tener un ajuste automático a cada clase inversamente proporcionales a su frecuencia en el conjunto de entrenamiento.

```
X = df_model.drop(columns=[target])
y = df_model[target]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42, stratify=y)


# Iniciar y entrenar el modelo de Regresión Logística con pesos de clase balanceados
model_logistic_balanced = LogisticRegression(solver='liblinear', random_state=42, class_weight='balanced')
model_logistic_balanced.fit(X_train, y_train)
```

```
# Realizar predicciones en el conjunto de prueba
y_pred_logistic_balanced = model_logistic_balanced.predict(X_test)

# Evaluar el rendimiento del modelo con pesos de clase balanceados
accuracy_logistic_balanced = accuracy_score(y_test, y_pred_logistic_balanced)
print(f"\nExactitud del modelo de Regresión Logística con pesos balanceados en el conjunto de prueba: {accuracy_logistic_balanced:.4f}")

print("\nReporte de Clasificación del modelo de Regresión Logística con pesos balanceados:")
print(classification_report(y_test, y_pred_logistic_balanced))

# Visualizar la matriz de confusión del modelo con pesos de clase balanceados
cm_logistic_balanced = confusion_matrix(y_test, y_pred_logistic_balanced)
plt.figure(figsize=(8, 6))
sns.heatmap(cm_logistic_balanced, annot=True, fmt='d', cmap='Blues',
            xticklabels=['No ACV', 'ACV'], yticklabels=['No ACV', 'ACV'])
plt.xlabel('Predicción')
plt.ylabel('Valor Real')
plt.title('Matriz de Confusión (Regresión Logística con Pesos Balanceados)')
plt.show()
```

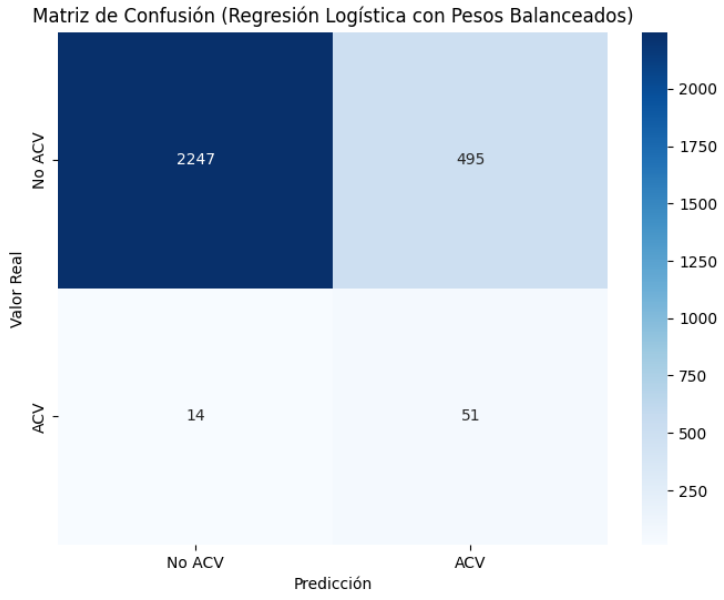
Exactitud del modelo de Regresión Logística con pesos balanceados en el conjunto de prueba: 0.8187

Reporte de Clasificación del modelo de Regresión Logística con pesos balanceados:

	precision	recall	f1-score	support
0.0	0.99	0.82	0.90	2742
1.0	0.09	0.78	0.17	65
accuracy			0.82	2807
macro avg	0.54	0.80	0.53	2807
weighted avg	0.97	0.82	0.88	2807

Figura 5

Matriz de Confusión con Pesos Balanceados



Análisis de los Resultados con Pesos de Clase Balanceados:

Exactitud (Accuracy): 0.8187, la exactitud es ligeramente menor que con SMOTE (0.8315) y significativamente menor que el modelo original (0.9768).

Reporte de Clasificación:

Clase 0 (No ACV): Precisión (Precisión): 0.99, Se mantuvo alta, Recall (Sensibilidad): 0.82 ligeramente menor que con SMOTE (0.83) y menor que el modelo original (1.00), esto significa que el modelo está identificando correctamente el 82% de los casos reales de No ACV, F1-score: 0.90, ligeramente menor que con SMOTE (0.91).

Clase 1 (ACV): Precisión (Precisión): 0.09, ligeramente menor que con SMOTE (0.10) y mucho menor que el modelo original (0.50) de todos los pacientes que el modelo predijo que tuvieron un ACV donde solo el 9% realmente lo tuvo, el número de falsos positivos sigue siendo alto. Recall (Sensibilidad): 0.78, ligeramente mejor que con SMOTE (0.77) y significativamente mejor que el modelo original (0.11), el modelo ahora está identificando correctamente el 78% de los casos reales de ACV. F1-score: 0.17 es similar al obtenido con SMOTE (0.17) pero aún bajo.

Matriz de Confusión:

Verdaderos Negativos (TN): 2247, ligeramente menor que con SMOTE (2284) y menor que el modelo original (2735), Falsos Positivos (FP): 495, Ligeramente mayor que con SMOTE (458) y mucho mayor que el modelo original (7), Falsos Negativos (FN): 14, ligeramente menor que con SMOTE (15) y mucho menor que el modelo original (58), Verdaderos Positivos (TP): 51, ligeramente mayor que con SMOTE (50) y mucho mayor que el modelo original (7).

El ajuste de los pesos de clase logró un recall para la clase ACV similar al obtenido con SMOTE (alrededor del 77-78%), lo cual es una mejora sustancial con respecto al modelo original, sin embargo la precisión para la clase ACV se mantuvo muy baja (alrededor del 9-10%) lo que indica un alto número de falsos positivos, el F1-score para la clase ACV tampoco mostró una mejora significativa.

Conclusión del modelo Regresión Logística.

Las técnicas para abordar el desbalance de clases SMOTE y Ajuste de Pesos si bien lograron aumentar significativamente la sensibilidad (recall) para la detección de ACV ambas también resultaron en una disminución considerable de la precisión, generando muchos más falsos positivos, el F1-score para la clase ACV se mantuvo bajo lo que indica que el equilibrio entre precisión y recall sigue siendo un desafío.

Teniendo en cuenta que la precisión del modelo es muy baja, se considera en trabajar un nuevo modelo como el Algoritmo Random Forest que podría manejar una mejor precisión y recall.

Algoritmo Random Forest

```
X = df_model.drop(columns=[target])
y = df_model[target]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42, stratify=y)

# Inicializar y entrenar el modelo de Random Forest
model_rf = RandomForestClassifier(random_state=42, class_weight='balanced')
model_rf.fit(X_train, y_train)

# Realizar predicciones en el conjunto de prueba
y_pred_rf = model_rf.predict(X_test)

# Evaluar el rendimiento del modelo Random Forest
accuracy_rf = accuracy_score(y_test, y_pred_rf)
```

```
print(f"\nExactitud del modelo Random Forest en el conjunto de prueba con un accuracy de: {accuracy_rf:.4f}")

print("\nReporte de Clasificación del modelo Random Forest:")
print(classification_report(y_test, y_pred_rf))

# Visualizar la matriz de confusión del modelo Random Forest
cm_rf = confusion_matrix(y_test, y_pred_rf)
plt.figure(figsize=(8, 6))
sns.heatmap(cm_rf, annot=True, fmt='d', cmap='Blues',
            xticklabels=['No ACV', 'ACV'], yticklabels=['No ACV', 'ACV'])
plt.xlabel('Predicción')
plt.ylabel('Valor Real')
plt.title('Matriz de Confusión (Random Forest)')
plt.show()

# Importancia de las características
feature_importances = model_rf.feature_importances_
df_importance = pd.DataFrame({'Feature': X_train.columns, 'Importance': feature_importances})
df_importance = df_importance.sort_values(by='Importance', ascending=False)
print("\nImportancia de las Características (Random Forest):")
print(df_importance.head(10))
```

Una vez modelado el algoritmo de Ramdon Forest se obtuvieron los siguientes resultados:

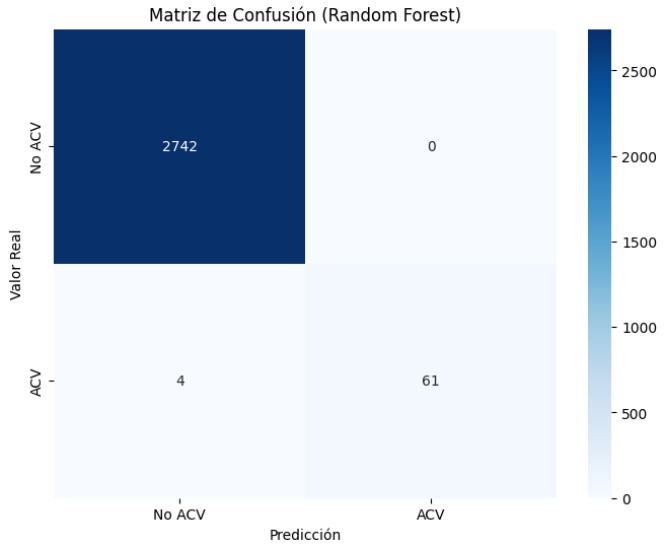
Exactitud del modelo Random Forest en el conjunto de prueba con un accuracy de: 0.9986

Reporte de Clasificación del modelo Random Forest:

	precision	recall	f1-score	support
0.0	1.00	1.00	1.00	2742
1.0	1.00	0.94	0.97	65
accuracy			1.00	2807
macro avg	1.00	0.97	0.98	2807
weighted avg	1.00	1.00	1.00	2807

Figura 6

Matriz de Confusión Random Forest



Importancia de las Características (Random Forest):

Feature	Importance

28	SISTOLICA_TRIAGE	0.165322
0	EDAD PACIENTE	0.132186
22	HIPERTENSO (0.NO 1.SI)	0.129614
29	DIASTOLICA_TRIAGE	0.128345
25	FRECUENCIA CARDIACA TRIAGE	0.115256
27	SATURACION OXIGENO TRIAGE	0.102558
26	TEMPERATURA TRIAGE	0.056193
21	ENFERMEDAD CORONARIA (0.NO 1.SI)	0.027130
5	ESTRATO_COTIZANTE CATEGORIA A	0.026193
1	SEXO	0.025168

Análisis de los Resultados de Random Forest:

Exactitud (Accuracy): 0.9986, tiene una exactitud extremadamente alta, cercana al 100%, indica que el modelo está clasificando correctamente la gran mayoría de las instancias en el conjunto de prueba.

Reporte de Clasificación:

Clase 0 (No ACV): Precisión (Precisión): 1.00, cuando el modelo predice que no hay ACV es correcto el 100% de las veces, Recall (Sensibilidad): 1.00: El modelo identifica correctamente el 100% de los casos reales de No ACV, F1-score: 1.00, es un puntaje perfecto, indicando un equilibrio ideal entre precisión y recall para esta clase.

Clase 1 (ACV): Precisión (Precisión): 1.00, cuando el modelo predice que hay ACV, es correcto el 100% de las veces, esto nos muestra que es una mejora drástica en comparación con la Regresión Logística, Recall (Sensibilidad): 0.94, el modelo identifica correctamente el 94% de los casos reales de ACV 61 de 65 casos, esta es también una mejora significativa, F1-score: 0.97

es un puntaje muy alto, indicando un excelente equilibrio entre precisión y recall para la clase ACV.

Matriz de Confusión:

Verdaderos Negativos (TN): 2742 - El modelo predijo correctamente que 2742 pacientes no tuvieron ACV.

Falsos Positivos (FP): 0 - El modelo no predijo incorrectamente ningún caso de No ACV como ACV.

Falsos Negativos (FN): 4 - El modelo predijo incorrectamente que 4 pacientes no tuvieron ACV cuando sí lo tuvieron.

Verdaderos Positivos (TP): 61 - El modelo predijo correctamente 61 casos de ACV.

El modelo Random Forest ha demostrado una capacidad excepcional para clasificar los casos de ACV en el conjunto de prueba. Logró una precisión y un recall muy altos para ambas clases lo que se traduce en un F1-score excelente para la clase minoritaria (ACV). Esto nos indica que Random Forest pudo aprender patrones complejos en los datos que permitieron una discriminación muy efectiva entre los pacientes que tuvieron y no tuvieron un ACV.

Conclusión.

El modelo Random Forest ha demostrado ser muy efectivo para la clasificación de ACV en este dataset, además de su alto rendimiento predictivo, la importancia de las características proporcionada por el modelo ofrece información valiosa sobre los factores que parecen ser más relevantes para la predicción de ACV en esta población. Estos hallazgos son en gran medida consistentes con el conocimiento médico sobre los factores de riesgo de ACV.

Importancia de las características según el modelo Random Forest

Obtener la importancia de las características

```
feature_importances = model_rf.feature_importances_
# Crear un DataFrame para mostrar la importancia de las características
df_importance = pd.DataFrame({'Feature': X_train.columns, 'Importance':
feature_importances})
# Ordenar el DataFrame por la importancia de mayor a menor
df_importance = df_importance.sort_values(by='Importance', ascending=False)

print("\nImportancia de las Características (Random Forest):")
print(df_importance)
```

Importancia de las Características (Random Forest):

	Feature	Importance
28	SISTOLICA_TRIAGE	0.165322
0	EDAD PACIENTE	0.132186
22	HIPERTENSO (0.NO 1.SI)	0.129614
29	DIASTOLICA_TRIAGE	0.128345
25	FRECUENCIA CARDIACA TRIAGE	0.115256
27	SATURACION OXIGENO TRIAGE	0.102558
26	TEMPERATURA TRIAGE	0.056193
21	ENFERMEDAD CORONARIA (0.NO 1.SI)	0.027130
5	ESTRATO_COTIZANTE CATEGORIA A	0.026193
1	SEXO	0.025168
16	ESTRATO_SUBSIDIADO NIVEL 1	0.019281
20	DIABETES (0.NO 1.SI)	0.015951

2	ESTRATO_BENEFICIARIO CATEGORIA A	0.009675
18	ESTRATO_SUBSIDIADO NIVEL 3	0.009200
17	ESTRATO_SUBSIDIADO NIVEL 2	0.008895
19	ESTRATO_VINCULADO TRANSITORIO	0.007997
15	ESTRATO_SUBSIDIADO NIVEL 0	0.006874
10	ESTRATO_POBLACION ESPECIAL	0.005737
12	ESTRATO_SISBEN NIVEL 1	0.002786
13	ESTRATO_SISBEN NIVEL 2	0.002575
6	ESTRATO_COTIZANTE CATEGORIA B	0.001227
9	ESTRATO_PARTICULARES	0.000534
3	ESTRATO_BENEFICIARIO CATEGORIA B	0.000461
14	ESTRATO_SISBEN NIVEL 3	0.000348
23	ANTICUAGULADO (0.NO 1.SI)	0.000199
11	ESTRATO_REGIMEN ESPECIAL	0.000130
8	ESTRATO_NO UTILIZAR	0.000114
4	ESTRATO_BENEFICIARIO CATEGORIA C	0.000038
7	ESTRATO_COTIZANTE CATEGORIA C	0.000014
24	GLOCOMETRIA (0.NO 1.SI)	0.000000

Características más importantes identificadas por el modelo Random Forest:

SISTOLICA_TRIAGE (0.165): La presión sistólica medida en el triage es, con diferencia, la característica más importante para la predicción. Esto subraya la importancia de la presión arterial en la evaluación inicial del paciente.

EDAD PACIENTE (0.132): La edad del paciente es el segundo factor más importante, lo cual es consistente con el conocimiento médico de que la edad es un factor de riesgo significativo para el ACV.

HIPERTENSO (0.NO 1.SI) (0.130): La presencia de hipertensión es el tercer factor más importante, lo que también concuerda con su rol conocido como un importante factor de riesgo para el ACV.

DIASTOLICA_TRIAGE (0.128): La presión diastólica en el triage también es altamente relevante, complementando la importancia de la presión sistólica.

FRECUENCIA CARDIACA TRIAGE (0.115): La frecuencia cardíaca medida en el triage contribuye significativamente a la predicción.

SATURACION OXIGENO TRIAGE (0.103): La saturación de oxígeno en el triage también es un factor importante.

TEMPERATURA TRIAGE (0.056): La temperatura en el triage tiene una importancia moderada en comparación con las anteriores.

ENFERMEDAD CORONARIA (0.NO 1.SI) (0.027): La presencia de enfermedad coronaria es un factor de riesgo relevante, aunque su importancia es menor que las variables de presión arterial y la edad.

ESTRATO_COTIZANTE CATEGORIA A (0.026): El estrato económico de cotizante de categoría A muestra cierta importancia, sugiriendo que el nivel socioeconómico podría estar indirectamente relacionado con otros factores de riesgo o el acceso a la salud.

SEXO (0.025): El sexo del paciente también contribuye a la predicción, aunque su importancia es relativamente baja en comparación con las variables fisiológicas y la hipertensión.

Características Menos Importantes:

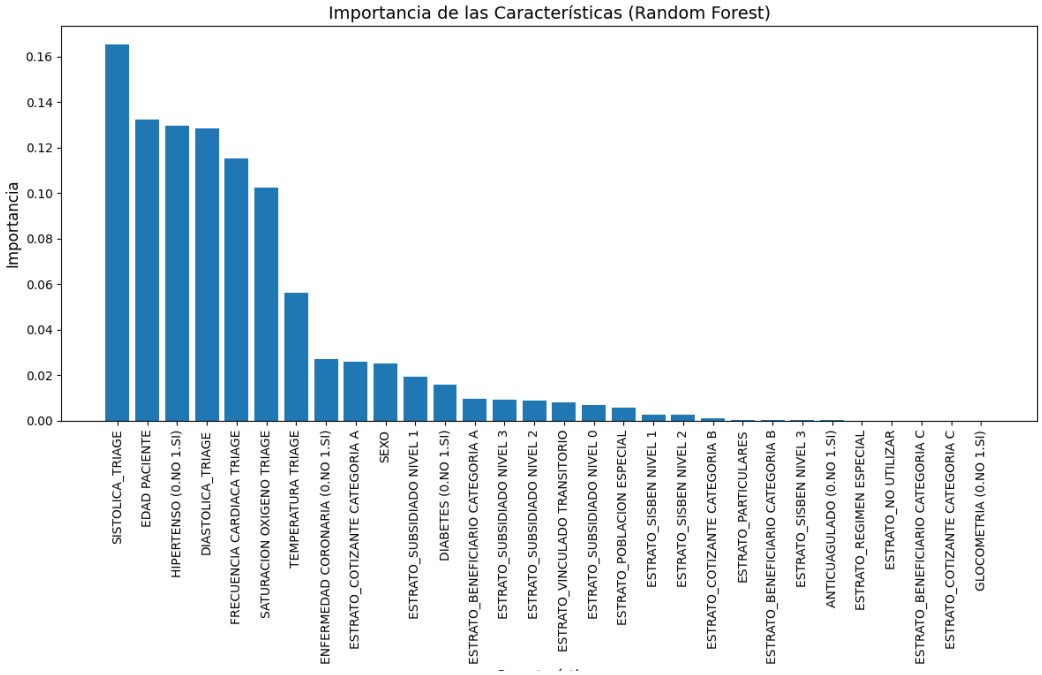
Varias de las variables relacionadas con el estrato económico, la diabetes, el uso de anticoagulantes y la glucometría tienen una importancia relativamente baja según el modelo, esto no necesariamente significa que estas variables no estén relacionadas con el ACV, sino que en el contexto de este modelo y las otras variables incluidas, su capacidad para discriminar entre los casos de ACV y no ACV es menor.

Conclusiones de la Importancia de las Características:

Las variables fisiológicas medidas en el triage (presión arterial sistólica y diastólica, frecuencia cardíaca, saturación de oxígeno, temperatura) y los factores de riesgo clínicos establecidos (edad e hipertensión) son los predictores más importantes de ACV según este modelo Random Forest, el sexo también contribuyen, aunque en menor medida, la variable Diabetes y el uso de anticoagulantes, aunque son factores de riesgo conocidos para el ACV, tienen una importancia relativamente baja en este modelo, posiblemente debido a la presencia de otras variables más fuertemente correlacionadas o porque su relación con el ACV en este dataset está mediada por otros factores.

Figura 7

Importancia de Características Random Forest



Análisis mediante AUC-ROC (Área Bajo la Curva Característica Operativa del Receptor) comparativo entre los dos modelos trabajados.

```
# Modelo de Regresión Logística
model_lr = LogisticRegression(solver='liblinear', random_state=42)
model_lr.fit(X_train, y_train)
y_pred_proba_lr = model_lr.predict_proba(X_test)[:, 1]
auc_lr = roc_auc_score(y_test, y_pred_proba_lr)
fpr_lr, tpr_lr, thresholds_lr = roc_curve(y_test, y_pred_proba_lr)

print(f"AUC-ROC Regresión Logística: {auc_lr:.4f}")
```

```
        # Modelo de Regresión Logística con SMOTE
        smote = SMOTE(random_state=42)
        X_train_smote, y_train_smote = smote.fit_resample(X_train, y_train)
        model_lr_smote = LogisticRegression(solver='liblinear', random_state=42)
        model_lr_smote.fit(X_train_smote, y_train_smote)
        y_pred_proba_lr_smote = model_lr_smote.predict_proba(X_test)[:, 1]
        auc_lr_smote = roc_auc_score(y_test, y_pred_proba_lr_smote)
        fpr_lr_smote, tpr_lr_smote, thresholds_lr_smote = roc_curve(y_test,
y_pred_proba_lr_smote)

        print(f"AUC-ROC Regresión Logística con SMOTE: {auc_lr_smote:.4f}")

        # Modelo de Regresión Logística con Pesos Balanceados
        model_lr_balanced = LogisticRegression(solver='liblinear', random_state=42,
class_weight='balanced')
        model_lr_balanced.fit(X_train, y_train)
        y_pred_proba_lr_balanced = model_lr_balanced.predict_proba(X_test)[:, 1]
        auc_lr_balanced = roc_auc_score(y_test, y_pred_proba_lr_balanced)
        fpr_lr_balanced, tpr_lr_balanced, thresholds_lr_balanced = roc_curve(y_test,
y_pred_proba_lr_balanced)

        print(f"AUC-ROC Regresión Logística con Pesos Balanceados: {auc_lr_balanced:.4f}")
```

```
        # Modelo Random Forest
        model_rf = RandomForestClassifier(random_state=42, class_weight='balanced')
        model_rf.fit(X_train, y_train)
        y_pred_proba_rf = model_rf.predict_proba(X_test)[:, 1]
        auc_rf = roc_auc_score(y_test, y_pred_proba_rf)
        fpr_rf, tpr_rf, thresholds_rf = roc_curve(y_test, y_pred_proba_rf)

        print(f"AUC-ROC Random Forest: {auc_rf:.4f}")

        # Graficar las curvas ROC comparativas
        plt.figure(figsize=(10, 8))
        plt.plot(fpr_lr, tpr_lr, label=f'Regresión Logística (AUC = {auc_lr:.4f})')
        plt.plot(fpr_lr_smote, tpr_lr_smote, label=f'Regresión Logística (SMOTE) (AUC =
{auc_lr_smote:.4f})')
        plt.plot(fpr_lr_balanced, tpr_lr_balanced, label=f'Regresión Logística (Balanceado)
(AUC = {auc_lr_balanced:.4f})')
        plt.plot(fpr_rf, tpr_rf, label=f'Random Forest (AUC = {auc_rf:.4f})')
        plt.plot([0, 1], [0, 1], 'k--')
        plt.xlabel('Tasa de Falsos Positivos (FPR)')
        plt.ylabel('Tasa de Verdaderos Positivos (TPR) / Recall')
        plt.title('Comparativa de Curvas ROC')
        plt.legend()
        plt.grid(True)
```

plt.show()

Resultados:

AUC-ROC Regresión Logística: 0.8373

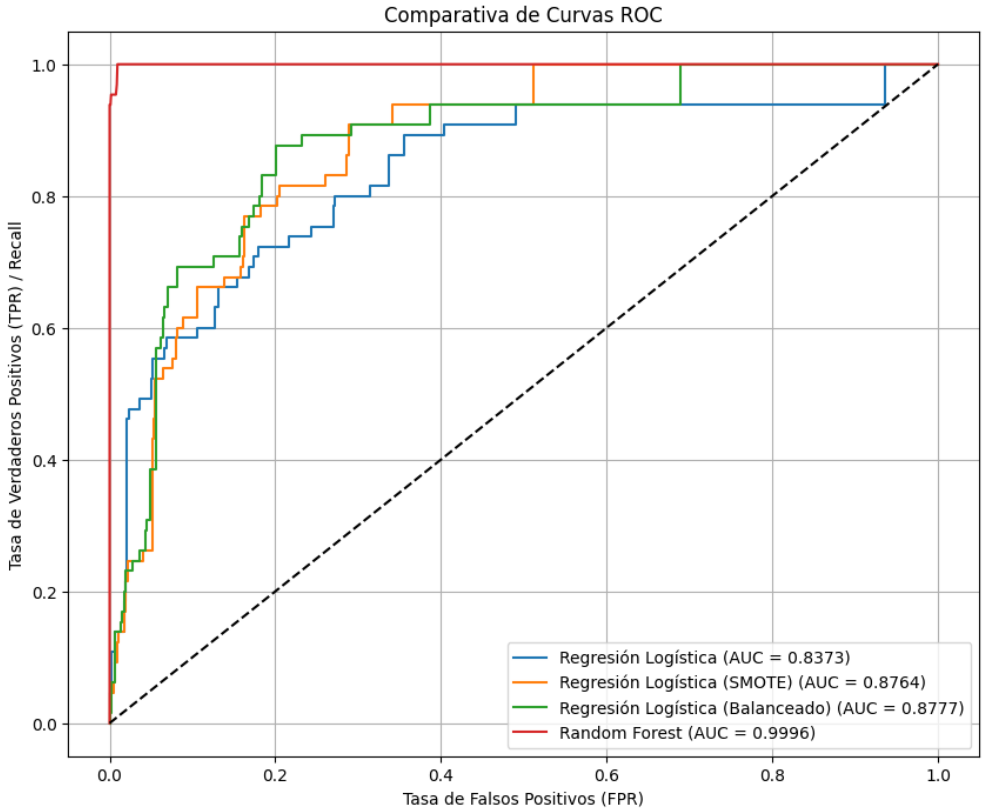
AUC-ROC Regresión Logística con SMOTE: 0.8764

AUC-ROC Regresión Logística con Pesos Balanceados: 0.8777

AUC-ROC Random Forest: 0.9996

Figura 8

AUC-ROC Regresión Logística vs Ramdon Forest



Análisis Comparativo de las Curvas ROC y AUC-ROC:

Regresión Logística (Base): Con un AUC de 0.8373, el modelo de regresión logística estándar muestra una capacidad razonable para discriminar entre los pacientes con y sin ACV, la

curva ROC se sitúa por encima de la línea diagonal que representa un clasificador aleatorio, indicando un rendimiento mejor que el azar.

Regresión Logística (SMOTE): La aplicación de la técnica de sobre muestreo SMOTE mejoró ligeramente el rendimiento del modelo de regresión logística, elevando el AUC a 0.8764, esto indica que abordar el desbalance de clases mediante la creación de muestras sintéticas ayudó al modelo a aprender mejor los patrones de la clase minoritaria (ACV), la curva ROC se desplaza un poco más hacia la esquina superior izquierda en comparación con la regresión logística base.

Regresión Logística (Pesos Balanceados): El uso de pesos balanceados para la regresión logística también resultó en una mejora similar a comparación del uso de SMOTE, con un AUC de 0.8777, donde obtuvo un AUC ligeramente superior al de la regresión logística con SMOTE, esto indica que asignar mayor importancia a la clase minoritaria durante el entrenamiento fue una estrategia efectiva para mejorar la capacidad de discriminación del modelo lineal, su curva ROC es muy similar a la de la regresión logística con SMOTE, ambas superando a la regresión logística base.

Random Forest: Con un AUC de 0.9996, el modelo Random Forest se destaca significativamente por encima de los modelos de regresión logística, su curva ROC se acerca muy rápidamente a la esquina superior izquierda, indicando una capacidad de discriminación casi perfecta, esto indica que el Random Forest puede distinguir entre los pacientes con y sin ACV con una precisión extremadamente alta en una amplia gama de umbrales de clasificación.

Conclusión Comparativa:

Los modelos de regresión logística tanto con SMOTE como con pesos balanceados mostraron una mejora en la capacidad de discriminación en comparación con la regresión logística base, lo que demuestra la importancia de abordar el desbalance de clases, sin embargo,

el modelo Random Forest demostró un rendimiento muy superior a todas las variantes de la regresión logística en términos de AUC-ROC, su capacidad para modelar relaciones no lineales y su robustez frente al desbalance de clases incluso con el parámetro class_weight='balanced' le permitieron lograr una discriminación casi perfecta entre las dos clases.

En resumen y basándonos en el AUC-ROC, el Random Forest es el modelo más eficaz para la tarea de clasificación de ACV en este dataset, seguido de cerca por las versiones de la regresión logística que abordaron el desbalance de clases SMOTE y pesos balanceados, las cuales superaron al modelo de regresión logística estándar.

Conclusiones

El modelo de regresión logística inicial aunque alcanzó una alta exactitud general aproximadamente 97.7% en la predicción de ACV, demostró ser limitado en su capacidad para identificar correctamente los casos positivos de ACV por su baja sensibilidad o recall con un 11% esto se evidenció en la cantidad tan alta de falsos negativos, si bien identificó correctamente a la mayoría de los pacientes sin ACV su rendimiento para el objetivo crucial de detectar la presencia de ACV fue deficiente, lo que se reflejó en un puntaje muy bajo F1-score para la clase positiva con un 0.18 reflejando un desequilibrio entre la precisión y el recall para la clase ACV en la matriz de confusión.

El análisis de los coeficientes reveló asociaciones significativas, como la fuerte correlación positiva entre la hipertensión y la enfermedad coronaria con la probabilidad de ACV Pero la baja sensibilidad general del modelo sugirió hacer ajustes en la configuración inicial aplicando SMOTE (Synthetic Minority Over-sampling Technique) para equilibrar la distribución, aunque el ajuste de pesos mejoraron significativamente el recall para la clase ACV se observó un aumento de la sensibilidad a aproximadamente 77-78% , esto produjo una disminución en la precisión, generando un número considerable de falsos positivos y sin una mejora sustancial en el F1-score general para la clase positiva.

El modelo Random Forest demostró una capacidad predictiva significativamente superior a la regresión logística para la identificación de ACV en el dataset que se obtuvo mediante consulta SQL a la base de datos de la Subred Sur Occidente de la ciudad de Bogotá, este modelo alcanzó una exactitud cercana al 100% con una alta precisión y recall tanto para la clase negativa (No ACV) como para la clase positiva (ACV) donde el modelo Random Forest pudo capturar

relaciones no lineales y complejas entre los factores de riesgo y la presencia de ACV de manera más efectiva.

Las variables fisiológicas medidas en el triage como la presión arterial sistólica y diastólica, frecuencia cardíaca, saturación de oxígeno, temperatura y los factores de riesgo clínicos establecidos de edad e hipertensión surgieron como los predictores más importantes de ACV según el modelo Random Forest.

El análisis bidimensional e inferencial confirmó asociaciones significativas entre las variables más relevantes y la presencia de ACV como por ejemplo, las medias de la presión arterial y la edad fueron significativamente diferentes entre los pacientes con y sin ACV, la hipertensión mostró una asociación estadísticamente significativa con una mayor probabilidad de ACV estas visualizaciones y pruebas estadísticas respaldan los hallazgos del modelo de machine learning.

Recomendaciones

Sugerencias para trabajos futuros:

Validar el modelo Random Forest en un conjunto de datos independiente de la Subred Sur Occidente o mediante técnicas de validación cruzada más robustas para asegurar su generalización a nuevas poblaciones de pacientes.

Profundizar en el análisis de los 4 casos falsos negativos de ACV que el modelo Random Forest clasificó incorrectamente.

Identificar patrones o características comunes en estos casos podría revelar factores de riesgo no capturados completamente por el modelo actual o la necesidad de datos adicionales.

Realizar incorporación de variables adicionales al dataset para explorar la inclusión de otras variables que puedan ser relevantes como el historial de tabaquismo, el índice de masa corporal (IMC), antecedentes médicos como diabetes mal controlada o resultados de pruebas de laboratorio clínico.

Investigar la posibilidad de desarrollar modelos de riesgo dinámicos que puedan actualizar la probabilidad de ACV de un paciente en función de la evolución de sus signos vitales y otros parámetros durante su estancia en el servicio de urgencias.

Mejoras para la Subred Integrada de Servicios de Salud Sur Occidente:

Implementación de alertas tempranas considerando la integración del modelo Random Forest en el sistemas de información de Historia Clínica para generar alertas tempranas de pacientes con alto riesgo de ACV en el servicio de urgencias como de hospitalización, lo que podría facilitar una priorización más rápida para la evaluación neurológica y el inicio de protocolos de tratamiento de ACV.

Recopilación de datos estandarizada y ampliada donde se debe fortalecer la recopilación de datos en el triage para asegurar la inclusión consistente y la calidad de cada variable identificadas como importantes por el modelo ya que en la recolección de datos se evidencio que varios campos son omitidos o no se registra un dato correcto estandarizado.

Considerar la expansión de la recopilación de datos a otras variables potencialmente relevantes mencionadas en las sugerencias para trabajos futuros ya que en el formato de historia clínica donde se capturo información carece de datos que pueden dar un mayor peso para trabajar modelos ML.

Capacitación del personal médico de triage y médicos de hospitalización sobre la importancia de las variables claves identificadas y cómo la identificación temprana del riesgo que puede mejorar los resultados de los pacientes con ACV.

Propuestas de ampliación o profundización del tema:

Realizar análisis específicos en subgrupos de pacientes por edad, sexo, comorbilidades para identificar posibles diferencias en los factores de riesgo y la precisión del modelo.

Comparación con guías clínicas sobre los factores de riesgo identificados por el modelo, establecidas para la clasificación del riesgo de ACV para evaluar la consistencia y posibles áreas de mejora en la práctica clínica.

Realizar protocolos en aspectos éticos relacionados con el uso de modelos de machine learning en la toma de decisiones clínicas y asegurar el cumplimiento de las regulaciones pertinentes.

Referencias Bibliográficas

Puy, L., & Jouvent, E. (2020). Accidente cerebro vascular isquémico en adultos mayores: un enfoque desde Terapia Ocupacional. *Revista Universidad Mariana*. https://revistas.umariana.edu.co/index.php/BoletinInformativoCEI/article/download/3931/4144/11015

Carrillo, E. (2024). Desarrollo de modelos de machine learning para la predicción del riesgo de desarrollar eventos cardiacos en pacientes de género femenino que son diagnosticadas con cáncer de mama Escuela Colombiana de Ingeniería Julio Garavito. https://repositorio.escuelaing.edu.co/server/api/core/bitstreams/83248bce-718c-4a95-8448-5ebd8276124e/content

Massaron, L. & Boschetti, A. (2016). *Regression Analysis with Python* (pp.15-28). https://bibliotecavirtual.unad.edu.co/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=1193721&lang=es&site=ehost live&ebv=EB&ppid=pp_23

Quiñones, A. & Oliveira M,. (2019*). Curso básico de Econometría clásica* (pp.27-35) Sello Editorial UNAD. https://doi.org/10.22490/9789586517171

Dangeti, P. (2017). *Statistics for Machine Learning: Build Supervised, Unsupervised, and Reinforcement Learning Models Using Both Python and R* (pp.15-28). https://bibliotecavirtual.unad.edu.co/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=e000xww&AN=1560931&lang=es&site=ehost live&ebv=EB&ppid=pp_83

Barreto, S. (2024). Aprendizaje Supervisado . [Objeto_virtual_de_Informacion_OVI]. Repositorio Institucional UNAD. https://repository.unad.edu.co/handle/10596/62836

Gaitan, R. (2022). Metrics . [Objeto_virtual_de_Informacion_OVI]. Repositorio Institucional UNAD. https://repository.unad.edu.co/handle/10596/50418