Predicción de Ocurrencia de Accidentes Cerebrovasculares

Andrés Julián Espinal Benjumea

Monografía presentada para optar al título de Especialista en Analítica y Ciencia de Datos

Asesor

Yony Fernando Ceballos, Doctor (PhD)

Universidad de Antioquia

Facultad de Ingeniería

Especialización en Analítica y Ciencia de Datos

Medellín, Antioquia, Colombia

2023

Cita	(Espinal Benjumea, 2023)
Referencia **Estilo APA 7 (2020)**	Espinal Benjumea, A. J. (2023). *Predicción de Ocurrencia de Accidentes Cerebrovasculares* Trabajo de grado especialización]. Universidad de Antioquia, Medellín, Colombia.

Especialización en Analítica y Ciencia de Datos, Cohorte V.

Centro de Investigación Ambientales y de Ingeniería (CIA).

UNIVERSIDAD DE ANTIOQUIA
Vicerrectoría de Docencia

Sistema de Bibliotecas

Centro de Documentación Ingeniería (CENDOI)

Repositorio Institucional: http://bibliotecadigital.udea.edu.co

Universidad de Antioquia - www.udea.edu.co

Rector: John Jairo Arboleda Céspedes.
Decano: Julio Cesar Saldarriaga Molina
Jefe departamento: Diego José Luis Botia Valderrama

Tabla de contenido

Lista de tablas

Lista de figuras

Siglas, acrónimos y abreviaturas

PhD	Philosophiae Doctor
AUC	Area Under the Curve
SMOTE	Synthetic Minority Over-sampling Technique
OMS	Organización Mundial de la Salud
ACV	Accidentes Cerebrovasculares
KB	Kilobytes
UCI	Unidad de Cuidados Intensivos
CSV	Comma-separated Values
ROC	Receiver Operating Characteristic
BMI	Body Mass Index
LOF	Local Outlier Factor
Q-Q	Quantil-Quantil
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative
KNN	K Nearest Neighbor

Resumen

La idea principal de este proyecto es construir un modelo capaz de predecir los accidentes cerebro vasculares, siendo éstos la segunda causa de muertes a nivel mundial, razón por la cual despierta el interés de esta investigación. Además, cuenta con su variable objetivo desbalanceada en sus clases en un porcentaje de 95.13 % para la clase mayoritaria y 4.87 % en la clase minoritaria. Los modelos usados fueron la regresión logística, random forest, máquinas de soporte vectoriales, k nearest neighbor y árboles de decisiones. Las métricas principales fueron el f1-score, recall y AUC porque clasifican mejor los casos positivos que son la clase minoritaria. La base de datos fue encontrada en kaggle y posee 5110 registros con 12 variables. Se realizaron cuatro iteraciones; la primera se usó el parámetro class_weight = balanced sin balancear la variable objetivo. La segunda iteración se balanceó dicha variable con la técnica SMOTE y se usaron modelos con parámetros por default. La tercera iteración se usó la técnica de GridSearchCV basado en la métrica f1-score y la última iteración se redujo la dimensionalidad en dos clases más. Los principales obstáculos en este proyecto consistían en lograr mantener la clase minoritaria con la menor pérdida de información posible al aplicar el preprocesamiento y medir la capacidad de generalizar el modelo sin que haya sobreajuste. Se trazó un objetivo de lograr un f1-score del 85 % pero al final el modelo de regresión logística logró llegar hasta 80 % siendo el mejor modelo de entre los evaluados.

Palabras Claves: Accidentes Cerebrovasculares, desbalanceo de clases, clasificación, matriz de confusión, f1-score, recall, AUC.

Github: https://github.com/AndresEspinal/EAYCD-UDEA-Monografia

Abstract

The main idea of this project is to build a model capable of predicting strokes, which are the second leading cause of death worldwide, making this research particularly interesting. Furthermore, the target variable is imbalanced in its classes, with a 95.13 % majority class and a 4.87 % minority class. The models used included logistic regression, random forest, support vector machines, k nearest neighbor, and decision trees. The primary metrics used were the f1-score, recall, and AUC, as they better classify positive cases, which are from the minority class. The dataset was found on Kaggle and consists of 5110 records with 12 variables. Four iterations were performed: in the first, the 'class_weight = balanced' parameter was used without balancing the target variable. In the second iteration, the variable was balanced using the SMOTE technique, and models with default parameters were used. The third iteration employed the GridSearchCV technique based on the f1-score metric, and the final iteration reduced the dimensionality to two additional classes. The main challenges in this project involved maintaining the minority class with minimal information loss during preprocessing and measuring the model's generalization ability without overfitting. The goal was to achieve an 85 % f1-score, but in the end, the logistic regression model achieved 80 %, making it the best-performing model among those evaluated.

Keywords: Stroke, class imbalance, classification, confusion matrix, AUC.

1. Descripción del problema

La base de datos llamada *stroke prediction dataset* busca clasificar entre *0* y *1* la probabilidad que un paciente sufra de un accidente cerebrovascular dado unas variables como tabaquismo, índice de masa corporal, sexo, edad, otras enfermedades, entre otras. Adicional, se busca mirar aquellas variables, de esta base de datos en particular, que tienen más importancia o aportan más información a los modelos al momento de clasificar. Para dar una idea al lector acerca de esta condición, se explicará brevemente de qué se trata, sus implicaciones y cómo se divide esta enfermedad o condición.

Según la OMS, los accidentes cerebrovasculares ocurren cuando se impide que la sangre fluya hacia el cerebro debido a una obstrucción, ya sea por depósitos de grasa, coágulos de sangre y hemorragias de los vasos cerebrales y sanguíneos que irrigan el cerebro (OMS, 2017). Es importante destacar que los accidentes cerebrovasculares se ubican en el segundo puesto de causas de defunciones a nivel mundial (OMS, 2020). Esto también lo confirma la National Health Services cuando advierte que, incluso si el paciente sobrevive a este accidente, está el riesgo de padecer problemas de salud a largo plazo o requerir de extensos periodos de rehabilitación, implicando reaprender ciertas habilidades para volver a vivir de forma independiente. (National Health Services, 2022). Por tal motivo, la necesidad de aprender más sobre esta condición e intentar descifrar tanto su comportamiento como las causas que lo provocan, se convierte en una necesidad de vital importancia para el sector de la salud.

En un estudio más local, realizado por Guerrero Agámez et al. (2021) señala que en los últimos 30 años se presentaron 374.713 defunciones por enfermedades cerebrovasculares en Colombia (Guerrero Agámez, Pestana Utria, Diaz Arrieta, Vargas Moranth, & Alvis Guzmán, 2021).

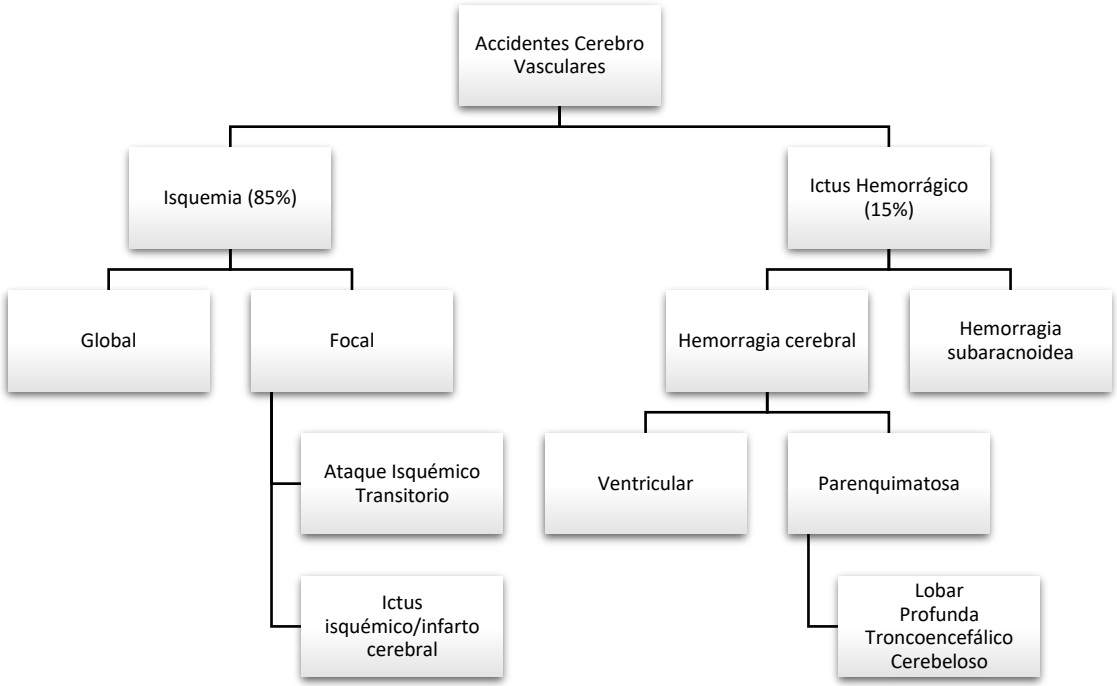
Los resultados anteriormente mencionados concuerdan con lo que señala el instituto Nacional de Salud junto con el Observatorio Nacional de Salud, que para el año 2011, la enfermedad cerebro vascular se convirtió en la tercera causa de muerte en Colombia, después de la Enfermedad Cardiaca Isquémica y las agresiones, con una tasa cruda del 29 % y entre los principales departamentos de Colombia siempre ocupó los cinco primeros puestos de muertes. (Instituto Nacional de Salud, Observatorio Nacional de Salud, 2013).

Con las cifras y datos mencionados se evidencia que los ACV deberían tener una mayor difusión debido al riesgo que representan en la población. Una mayor educación en estos temas puede

reducir los altos números de muertes e incluso ayudar a combatir los problemas médicos asociados que lo causan.

Para entender un poco mejor cómo se dividen los accidentes cerebro vasculares, (ver **Figura *1***), Bolaños Vaillant et al. (2009) comparte la siguiente información donde señala que: *"Según su naturaleza, las enfermedades cerebrovasculares (ECV) pueden presentarse como isquemia o hemorragia, con una proporción de 85 y 15 %, respectivamente. La isquemia se produce por la disminución del aporte sanguíneo cerebral, ya sea de forma total (isquemia global) o parcial (isquemia focal). Según la duración del proceso, se produce como un ataque [isquémico] transitorio (AIT) de esta o como infarto cerebral, según el déficit isquémico se revierta o no antes de las 24 horas. La hemorragia es la presencia de sangre en el cerebro, ya sea en el parénquima, el interior de los ventrículos cerebrales o el espacio subaracnoideo. La enfermedad cerebrovascular hemorrágica (ECVH), provocada por la rotura de un vaso, representa 15-20 % de todos los ictus y puede ser de diferentes tipos, de acuerdo con su localización, tales como: hemorragias intraventriculares (HIV), intraparenquimatosa (HIP), cerebromeníngea (HCM) y subaracnoidea (HSA), si bien esta última también puede ser causada por malformaciones arterovenosas. Según su topografía, la hemorragia cerebral se clasifica en lobar (frontal, temporal, parietal y occipital), profunda (afección talámica o capsular, o de los ganglios basales) y del tronco encefálico y cerebeloso."* (Bolaños Vaillant, Gómez García, Dosouto Infante, & Rodríguez Cheong, 2009)

Figura 1
División de los accidentes cerebro vasculares



Nota. Fuente https://www.semg.es/doc/documentos_SEMG/estrategias_ictus_SNS.pdf Adaptación del Ministerio de Sanidad y Consumo de España. (Ministerio de Sanidad y Consumo, 2008)

Una de las causas indirectamente asociadas a los accidentes cerebro vasculares es el estilo de vida que llevan las personas. Si se habla de tener una mala alimentación, tabaquismo y un estilo de vida sedentario, esto puede conducir a enfermedades como diabetes, hipertensión y altos niveles de colesterol que a su vez puede provocar los ACV. Por tal motivo uno de los controles apropiados para evitar esta condición es llevar una vida saludable, esto debido a que a la fecha no hay con certeza un medicamento que ataque de manera frontal la enfermedad, por el contrario, lo que sí se puede encontrar son tratamientos para prevenir estos factores de riesgo que pueden causarlo. Así lo señala la NHS al decir que los accidentes cerebrovasculares se tratan con medicamentos para prevenir otro tipo de enfermedades o condiciones médicas como coágulos de sangre, presión arterial alta o reducir niveles de colesterol. También hay situaciones más riesgosas en casos donde se deban eliminar coágulos de sangre con tratamientos quirúrgicos o para tratar las inflamaciones del cerebro y reducir la probabilidad de causar un accidente cerebrovascular (National Health Services, 2022).

1.1. Problema de negocio

En Colombia, los ACV deberían ocupar un espacio de discusión nacional en las entidades gubernamentales y de salud debido al impacto que generan en cifras de defunción y hospitalización sin contar con los costos asociados a los que estas entidades incurren en temas de tratamientos y rehabilitación. Además, esto sin incluir los factores de riesgo que lo generan como el consumo de tabaco, tensión arterial alta, hiperglucemia o diabetes, hiperlipidemia, obesidad, fibrilación auricular, entre otras. (Ministerio de Salud y Protección Social, 2020). Por tal motivo se ha ido generando todo un marco legal para su atención en los centros de salud. Las leyes, resoluciones y guías médicas que regulan los ACV son sugeridas por RecaVar, que es una ONG que está conformada por interesados en enfermedades cerebrovasculares para ayudar a prevenirlas y mejorar la atención en pacientes (RecaVar, s.f.). Estas leyes dispuestas en la **Tabla *I*** orientan y sugieren el paso a paso para exigir los derechos en atención.

Tabla I

Leyes, resoluciones y guías que están enfocadas a la atención de ACV.

Ley, resolución o guía	Descripción	Enlace
Ley Estatutaria 1751 de 2015	Por medio de la cual se regula el derecho fundamental a la salud.	**bit.ly/3KZmUF2**
Resolución 5596 del 2015	Por medio de la cual se definen los criterios técnicos para el sistema de selección y clasificación de pacientes en los servicios de urgencias “Triage”.	**bit.ly/3sup9d7**
Resolución del 2003-2014	Por la cual se definen los procedimientos y condiciones de inscripción de los prestadores de servicio de salud y de habilitación de servicios de salud.	**bit.ly/3OVrjdl**
Ley 1164 de 2007	Por la cual se dictan disposiciones en materia del talento humano en salud.	**bit.ly/45vlv1k**
Guía de Práctica Clínica	Para el diagnóstico, tratamiento y rehabilitación del episodio agudo del Ataque Cerebrovascular Isquémico en población mayor de 18 años.	**bit.ly/3qXgn6U**

Aunque la medicina tradicional y la preventiva aportan gran valor a la ciencia de datos para configurar modelos y conocer un poco mejor las enfermedades, a nivel de estado sería óptimo prevenir enfermedades en lugar de reaccionar a ellas. Por ello, implementar una buena recolección

y gestión de bases de datos suficientemente bien estructuradas, con el objetivo hacer uso de técnicas de machine learning en modelos supervisados y no supervisados de diferentes enfermedades lograría diagnósticos más rápidos, se optimizaría la atención de pacientes, se reducirían los costos, mejoraría la atención preventiva, entre otras.

1.2. Aproximación desde la analítica de datos

Una de las ventajas que puede aportar la ciencia de datos y los modelos de machine learning supervisados al proceso médico se puede derivar en la recopilación, almacenamiento, gestión y tratamiento de la información. Lo anteriormente mencionado va de la mano con la medicina preventiva que busca evitar las complicaciones médicas antes que estas sucedan y generen un gasto mayor al estado o las entidades encargadas de la prestación de la salud. Un ejemplo de ello son los accidentes cerebro vasculares que son accidentes que pueden ocurrir en cualquier momento, pero con el conocimiento de ciertas variables e información pueden ser previsibles estos sucesos. Si se toma esto como lección se evitarían muchos procesos muy costosos para este tipo de accidentes como procedimientos quirúrgicos, ocupación de UCI's, personal médico especializado, procedimientos de rehabilitación, entre otros.

Con departamentos bien estructurados de procesamiento de datos en técnicas de machine learning se podría aplicar mejor la medicina preventiva y se detectarían potenciales pacientes en lugar de esperar que ellos lleguen a los hospitales.

1.3. Origen de los datos

La base de datos usada en esta monografía fue obtenida de la página Kaggle que es un sitio web diseñado para compartir bases de datos con el propósito de estudiar, compartir conocimientos y desarrollar modelos de machine learning, entre otros

La base de datos se llama "*stroke prediction dataset*" y fue compilada por Federico Soriano, conocido en esta página como Fedesoriano. Esta base de datos contiene información relevante sobre los accidentes cerebro vasculares compuesta por 5110 registros y 12 variables, una de ellas se denomina variable de salida, en este proyecto llamada *stroke*, que está conformada por *0* y *1* en sus clases, lo que indica que se está tratando un modelo de aprendizaje supervisado de clasificación donde se busca que las predicciones digan *1* si es susceptible de padecer esta condición o *0* si no es propenso a padecerlo.

1.4. Métricas de desempeño

Cuando se aborda un problema de machine learning supervisado, como el de clasificación, una de las métricas más relevantes a la hora de medir la efectividad de los modelos es por la matriz de confusión. Esta funciona separando los datos en entrenamiento y validación y estos últimos serán clasificados por el modelo para luego ser comparados con la etiqueta real. Las medidas que se obtienen son la precision, recall, f1 score y accuracy. Para este trabajo se decide que la medida f1-score será la principal y el recall como medida secundaria para evaluar el modelo debido a que la base de datos cuenta con muy pocos registros positivos y lo que interesa en mayor medida es que haga una buena medición de los nuevos casos positivos, son estos los que alertarán a los pacientes que sean propensos a sufrir los ACV.

Además, se usará la curva ROC y la medida AUC como apoyo para distinguir los casos positivos y negativos.

2. Objetivos

2.1.Objetivo general

Predecir los casos positivos de los accidentes cerebro vasculares por encima del 85 % en sus métricas f1-score como métrica principal y en recall como métrica secundaria.

2.2.Objetivos específicos

- Comparar los resultados de las métricas usando la base de datos desbalanceada contra las métricas usando las técnicas de sobremuestreo para balancear variables.
- Identificar de manera adecuada si los modelos planteados generan un sobreajuste con los datos de validación.
- Definir una técnica de sobremuestreo adecuada que permita crear datos sintéticos que mejoren la variedad de los registros debido a un desbalance cercano al 95 % de sus clases.
- Establecer un preprocesamiento de datos que ayude a comparar las clases de las variables para encontrar relaciones o independencias de las variables.
- Identificar cuáles son las variables que más contribuyen a los resultados del modelo en esta base de datos en particular.

3. Datos

3.1. Datos originales

La base de datos tiene un tamaño de 309KB en formato CSV y está compuesta por 12 variables y 5110 registros. Posee 26 clases diferentes entre todas sus variables. Los datos están relacionados con condiciones de salud, datos personales y estilo de vida organizados para predecir accidentes cerebro vasculares.

La **Tabla *II*** explicará a fondo cada una de sus variables y el tamaño de cada una de sus clases en conteos nominales.

Tabla II

Conformación y explicación de las variables de la base de datos.

Variable	Detalle	Distribución
id	Se refiere a un código único que tiene cada paciente.	Su distribución es única en cada registro
gender	Distribuida en 3 géneros los cuales son *Male*, *Female* y *Other*.	*Male* = 2115, *Female* = 2994 y *Other* = 1
age	Hace referencia a la edad del paciente.	Escala numérica de 0.08 a 82 años.
hypertension	Clasifica con *0* si el paciente no tiene hipertensión y con *1* si el paciente sufre hipertensión.	*0* = 4612 y *1* = 498
heart_disease	Clasifica con *0* si el paciente no tiene ninguna enfermedad cardíaca y con *1* si el paciente padece una enfermedad cardíaca.	*0* = 4834 y *1* = 276
ever_married	Esta variable explica si el paciente está casado con *Yes* y con *No* si no lo está.	*Yes* = 3353 y *No* = 1757
work_type	Se divide en si trabajó con niños como *children*, si obtuvo un trabajo en el gobierno *Govt_jov*, si nunca trabajó *Never_worked*, si trabajó en el sector privado *Private* o por el contrario trabajó como independiente *Self-employed*.	*Private* = 2925, *Self-employed* = 819, *children* = 687, *Govt_job* = 657 y *Never_worked* = 22
Residence_type	Se divide en si la zona de residencia es rural *Rural* o urbana *Urban*.	*Urban* = 2596 y *Rural* = 2514

avg_glucose_level	Dato numérico que mide el nivel promedio de glucosa en sangre.	Escala numérica de 55.12 a 271.74
bmi	Dato numérico que muestra el índice de masa corporal.	Escala numérica de 10.3 a 97.6
smoking_status	Columna que especifica si el paciente ya había fumado anteriormente *formerly smoked*, si nunca fumó *never smoked*, si en la actualidad fumaba *smokes* o si la información no está disponible para el paciente como *Unknown*.	*never smoked = 1892, Unknown = 1544, formerly smoked = 884 y smokes = 789*
stroke	Es la variable respuesta e indica con *1* si el paciente tuvo un accidente cerebrovascular o con *0* si no lo tuvo.	*1* = 249 y *0* = 4861

El tipo de variable y el conteo de datos no nulos es mostrado en la **Tabla *III***.

Tabla III
Formato y cantidad de registros de cada variable

Column	Non-Null Count	Dtype
id	5110 non-null	int64
gender	5110 non-null	object
age	5110 non-null	float64
hypertension	5110 non-null	int64
heart_disease	5110 non-null	int64
ever_married	5110 non-null	object
work_type	5110 non-null	object
Residence_type	5110 non-null	object
avg_glucose_level	5110 non-null	float64
bmi	5110 non-null	float64
smoking_status	4909 non-null	object
stroke	5110 non-null	int64

Los tipos de datos de las 12 variables están conformados por cinco (5) con formato object, cuatro (4) en formato int64 y tres (3) en float64. También se puede observar que solo en la variable *bmi* tiene datos faltantes o nulos.

3.2. Datsets

En primera medida se verifica que el tipo de datos de cada variable corresponda con su verdadera naturaleza con la cuál fue registrada y se logra evidenciar que todas cumplen con sus características apropiadas.

Se analiza cada variable buscando que el aporte de información pueda ser relevante para la creación de los modelos y como primer hallazgo se encuentra que la variable llamada *id* viene siendo un identificador único o un código en cada registro, por lo cual se toma la decisión de eliminarlo antes de realizar algún procedimiento.

También se identifica que la base de datos posee una variable de salida llamada *stroke* que es la variable objetivo. Esta variable es de naturaleza binaria ya que está representada por *0* y *1* y en ella se evidencia que su composición presenta un desbalanceo de las dos clases evidenciada en la **Tabla *IV***.

Tabla IV
Variable de salida. Stroke

Stroke	***Registros***	***Porcentaje %***
0	4860	95.13%
1	249	4.87 %

Por medio de una búsqueda se encontró que ningún registro se encuentra duplicado por lo cual no hubo necesidad de eliminar alguno.

Se procede a verificar todas las clases de cada variable para saber internamente que es lo que representan en la base de datos y analizar posibles tendencias. En esta búsqueda se relaciona ahora la **Tabla *V*** y la **Tabla *VI***.

Tabla V
Variable gender. Clase Other

gender	***Registros***	***Porcentaje%***
Male	2115	41.3 %
Female	2994	58.5 %
Other	1	0.01 %

Tabla VI

Variable Work_type. Clase Never_worked

work_type	***Registros***	***Porcentaje %***
Private	2925	57.2 %
Self-employed	819	16.02 %
Govt_job	657	12.8 %
children	687	13.4 %
Never_worked	22	0.4 %

Al momento de evidenciar el aporte de cada clase en su variable, se encuentran dos de ellas que aportan muy poco a la base de datos, algo menor al 0.41 % de los registros totales. Por un lado, se encuentra *Other* de la variable *gender* y por el otro *Never_worked* de la variable *work_type*. Para el tratamiento de datos de estas dos clases se elimina la clase *Other* ya que puede ser poco o nulo lo que puede aportar al modelo y más adelante cuando sea aplicado dummies se estaría reduciendo la dimensionalidad en una variable. En el segundo caso se desea dejar la clase *Never_worked* para comprobar cómo funciona en primera instancia los modelos con esta clase y luego será eliminada en la cuarta iteración para observar el cambio de las métricas de los modelos.

Como se mencionó previamente, la variable *bmi*, que corresponde al índice de masa corporal por sus siglas en inglés, posee datos faltantes en un porcentaje del 3.93 %, equivalentes a 201 registros (ver **Figura *2***). Debido a que no se desea perder información importante de la base de datos se opta por imputar y no eliminar datos faltantes. Adicional se realiza un boxplot de la variable *bmi*, (ver **Figura *3***), para determinar cuál es el mejor método de imputación. Se observa que la variable tiene datos atípicos muy alejados de la caja y bigotes y por consiguiente se realiza la imputación usando la mediana como medida central para evitar que esta se vea influenciada por los datos atípicos.

Figura 2
Valores nulos o faltantes

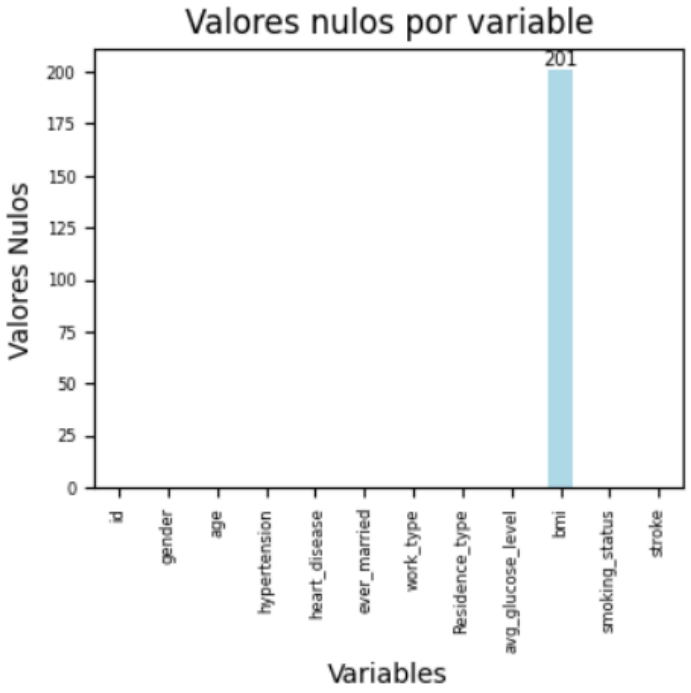
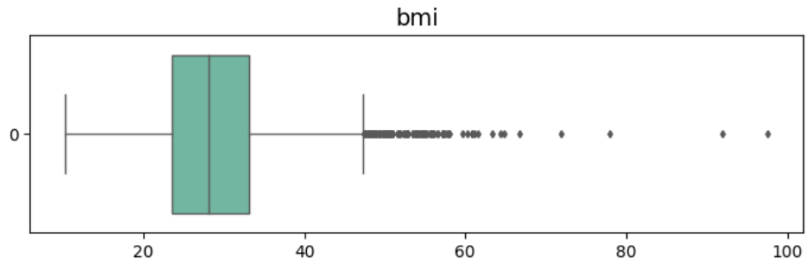


Figura 3
BoxPlot para la variable bmi



El escalamiento se realiza por medio de la técnica MinMaxScaler ya que entre sus ventajas mantiene la relación entre las variables y que no genera valores negativos.

Para la eliminación de datos atípicos se usó LOF (Local Outlier Factor) que se basa en vecinos cercanos y en densidades logrando detectar elementos que se encuentran por fuera de los ratios generados. (Breunig, Kriegel, Ng, & Sander, 2000) La aplicación de esta técnica arrojó 55 datos atípicos que fueron removidos de la base de datos. Para complementar la eficacia del procedimiento anterior, se usó el valor de la entropía, (ver **Tabla *VII***), para determinar si hay cambios considerables o no en la información de los datos.

Tabla VII
Valor de la Entropía

	Entropía Original	***Entropía LOF***
age	*0.251818*	*0.252542*
avg_glucose_level	*0.261906*	*0.262434*
bmi	*0.311535*	*0.312245*

Se puede decir que, al tener entropías relativamente cercanas, no han introducido gran cantidad de cambios con respecto en las variables originales. Es decir que LOF logró mantener gran parte de la información original de la base de datos.

Anteriormente se mencionó que la variable de salida posee un desbalance en un porcentaje de 4.87 % para registros positivos *1* contra un 95.13 % para registros negativos *0*. Para resolver este problema, se utiliza la técnica SMOTE para aumentar la cantidad de registros sintéticos positivos. Se usa esta técnica porque crea datos sintéticos muy cercanos a los existentes lo que conlleva a regiones de decisión más amplias con mayor variabilidad. (Chawla, Bowyer, Hall, & Kegelmeyer, 2002).

Finalmente se usa un proceso de transformación de las variables categóricas a numéricas por medio de dummies debido a que no serán muchas variables las que se crearán, adicional se usa el parámetro de drop_first que reduce una variable por cada variable original, logrando que no se genere una dimensionalidad alta. Además, las clases de cada columna no siguen un orden o una jerarquía por lo cual no otorga pesos a las clases al momento de aplicar modelos de machine learning.

3.3. Analítica descriptiva

Después de explicado el proceso de análisis exploratorio, de cómo los datos fueron abordados punto por punto ahora se explicará como las variables están conformadas y la información que brindan a la hora de abordar la etapa de entrenamiento y validación.

Como primer paso se desea conocer que tan correlacionadas están las variables entre sí y por tal motivo se realiza mapa de correlación, (ver **Figura *4***), en las variables numéricas y binarias.

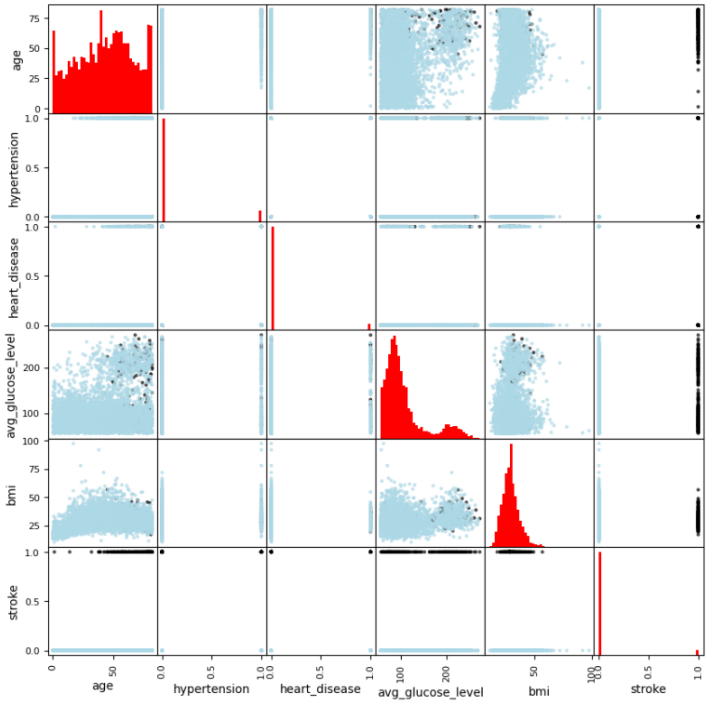
Figura 4
Mapa de correlación

Se puede observar que ninguna de estas variables posee una correlación fuerte, es decir que estén por encima de 0.7 en valor absoluto. Ahora entre todas las variables se tiene que *age* y *bmi* poseen una correlación baja positiva ya que se encuentran en el umbral de 0.3 a 0.5 en valor absoluto.

De igual manera si se comprar la matriz de correlación con la matriz de dispersión (ver **Figura *5***) es posible observar que entre todas las relaciones solo *age* y *bmi* son las que visualmente se ven más lineales quitando las variables binarias que son de otra naturaleza. Los puntos negros que se visualizan en el gráfico corresponden a la clase *1* de la variable de la salida, lo que hace notar nuevamente el nivel de desbalanceo de la variable.

Figura 5

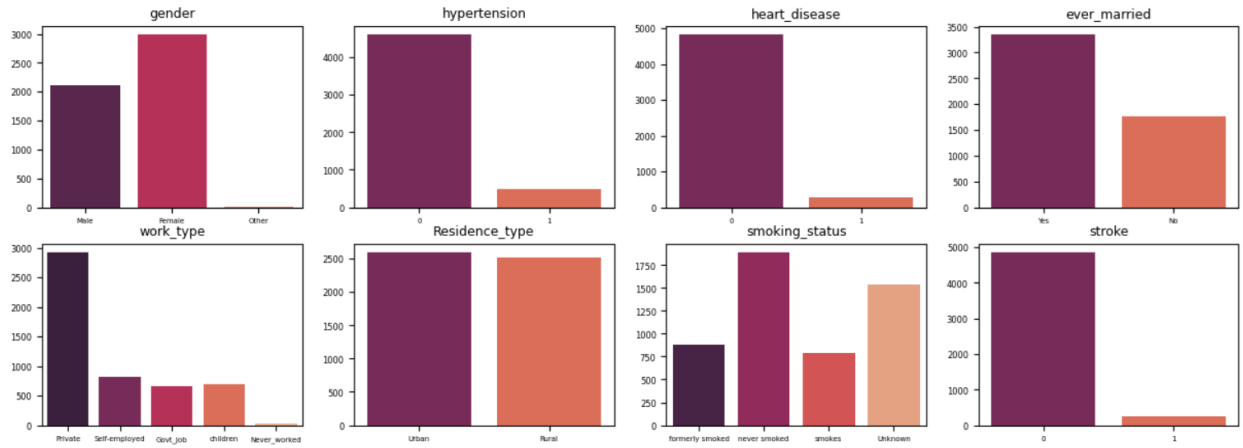
Gráfico de dispersión



Sobre las variables discretas (ver **Figura *6***), se tiene que aquellas que son binarias como *hypertension*, *heart_disease* y la variable de salida *stroke* tienen un desbalance muy parecido de sus clases. Acerca de las variables categóricas se ve a *work_type* muy desbalanceada en su clase *Private*, de igual manera la variable *smoking_status* tiene algunas clases desbalanceadas, también se puede decir que entre las variables más balanceadas se encuentran *gender*, *ever_married* y *Residence_type*.

Figura 6

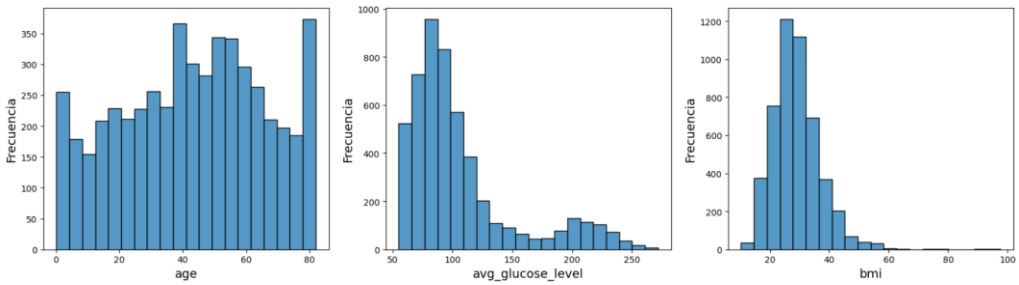
Gráfico de barras de las variables binarias y categóricas



Ahora bien, en la **Figura *7***, las variables numéricas *bmi*, *age* y *avg_glucose_level*, sus histogramas pueden evidenciar que sus formas intentan parecer una distribución normal. Pero para comprobarlo, es necesario aplicar una gráfica Q-Q que dice si una variable numérica sigue una distribución normal.

Figura 7

Histogramas de variables numéricas



Observando las gráficas de Q-Q por cada variable numérica, (ver **Figura *8***), se puede apreciar que las variables numéricas no siguen una distribución normal ya que los datos no se encuentran por encima de la diagonal en todos sus puntos. Para comprobarlo, se realiza la prueba de Shapiro-Wilk para probar las hipótesis, donde:

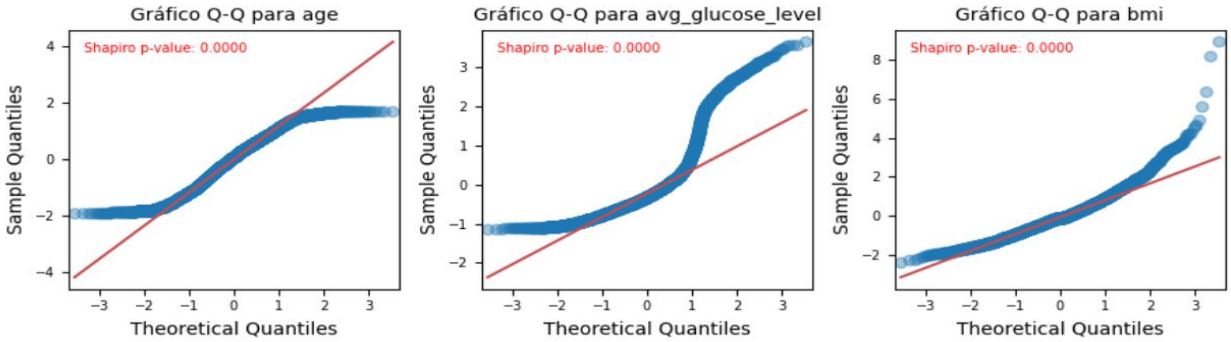
H_0: Los datos siguen una distribución normal.

H_A: Los datos no siguen una distribución normal.

En este caso las tres pruebas dieron un p-valor por debajo de 0.05, resultados que se encuentran dentro de la **Figura *8***, rechazando la hipótesis nula y aprobando la alternativa.

Figura 8

Q-Q de las variables numéricas

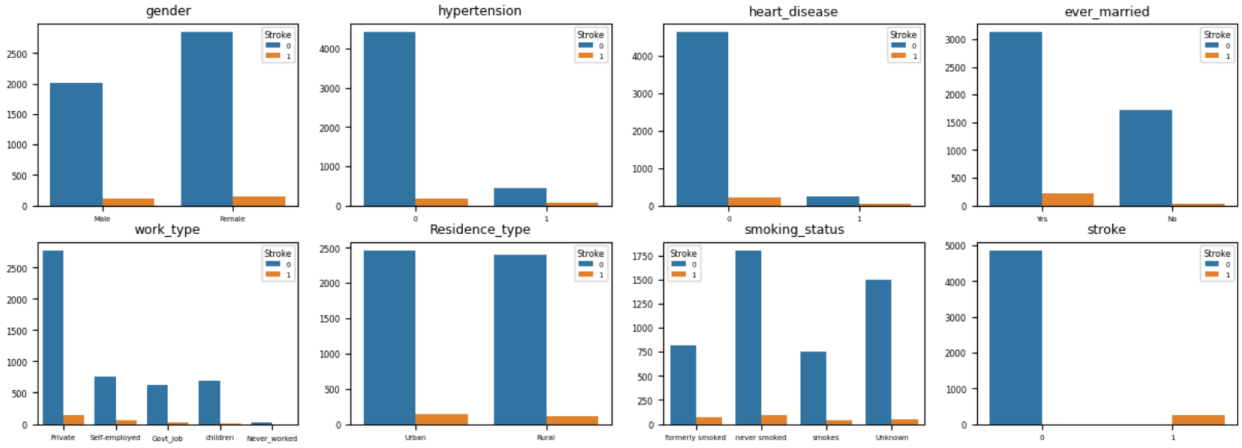


Para conocer como cada clase de cada variable estaba asociada a la variable de salida, se calculó con una medida porcentual de cada una de las clases. A continuación, se muestran la **Figura *9*** y **Figura *10*** donde se tienen las variables categóricas, numéricas y binarias.

En las variables categóricas y binarias se encuentra que las clases que más casos positivos porcentualmente se observan de la variable *stroke* son *heart_disase* en su clase *1* con 17.03 %, *hypertension* en su clase *1* con 13.25 %, *work_type* en su clase *Self-employed* con un 7.94 %, *smoking_status* en su clase *formerly smoked* con 7.92 % y para completar el top 5 *ever_married* en su clase "*Yes*" con 6.56 %.

Figura 9

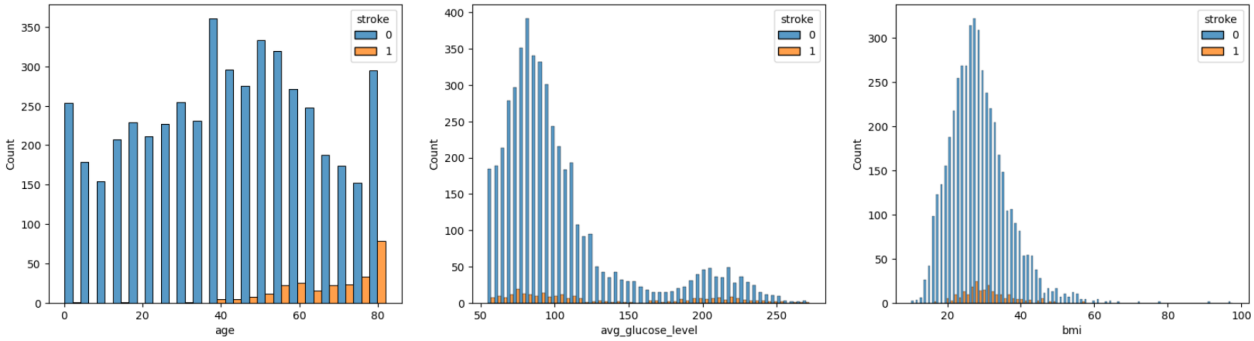
Composición de las variables contra la variable objetivo



A nivel de variables cuantitativas se puede observar que en la variable *age* a partir de los 40 años los casos positivos comienzan una tendencia alcista hacia la enfermedad y esta representa la situación más destacada en comparación con *bmi* y *avg_glucose_level*.

Figura 10

Composición de las variables numéricas contra la variable objetivo en histograma

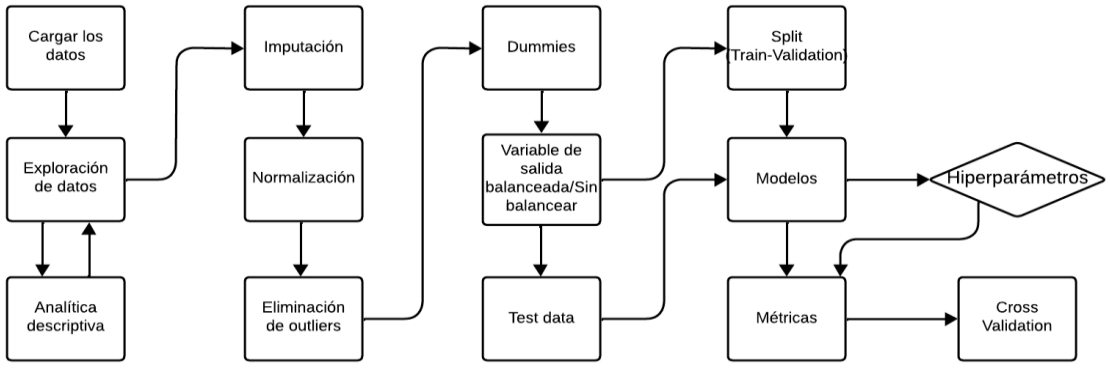


4. Proceso de analítica

4.1. Pipeline principal

El pipeline principal (ver **Figura *11***) comienza con el cargue de datos obtenido de kaggle. La exploración de datos se buscó conocer cómo está distribuida la base de datos, número y naturaleza de las variables con eliminación de registros que no aportan al modelo. En la sección de analítica descriptiva se observa una doble configuración de entrada y salida con el fin de entender la base de datos y buscar los métodos de preprocesamiento óptimos, allí se observó como cada variable se comportaba con las otras variables buscando dependencia o independencia y así mismo contra la variable objetivo. Se observó además su grado de desbalance. Luego sigue la imputación de registros faltantes. Se procede a normalizar las variables en este caso con el método MinMaxScaler. Se eliminan datos atípicos con el método LOF y luego se revisa su entropía. Antes de realizar el balanceo de clases de la variable objetivo se transforman las variables categóricas a numéricas por el método dummies. Se realiza el balanceo de clases de la variable de salida por el método SMOTE. Se separan los datos en tres grupos que son datos de entrenamiento y validación y adicional en datos de prueba donde se usó alrededor del 5 % de la base de datos. Se recrean los modelos primero con la variable de salida desbalanceada con hiperparámetros y luego se balancea con una configuración de parámetros por default y se miden sus métricas. Para mejores resultados se crea una tercera iteración donde se modelan unos hiperparámetros mediante validación cruzada con unos nuevos resultados. Se hace una cuarta iteración retirando más clases y regresa el ciclo nuevamente a las métricas que darían el resultado final.

Figura 11
Pipeline principal



4.2. Preprocesamiento

En este paso se elimina la variable *id* porque no aporta información de valor al momento de usar las técnicas de machine learning. Se imputa por la mediana y se eliminan datos atípicos por LOF. Se usa MinMaxScaler para normalizar las variables numéricas y se transforman los datos categóricos a numéricos por el método de dummies por la razón que las clases de las variables no tenían ningún peso u orden jerárquico. También se usó el balanceo por el método SMOTE y no por la técnica de random oversampler debido a que esta última duplica los registros. Por último, se sacan alrededor del 5 % de datos de prueba y el 95 % restante se separa en 80 % para datos de entrenamiento y 20 % para validación.

4.3.Modelos

Para la parte predictiva y de aprendizaje de modelos, se usarán aquellos modelos referentes a problemas de clasificación de modelos de aprendizaje supervisado con el fin de obtener aquel que muestre una mejor métrica tanto en f1-score como en recall. De esta manera los modelos que serán utilizados son:

Regresión logística: la regresión logística se puede resumir como una técnica estadística que toma una combinación ponderada de las características de entrada y la pasa a través de una función sigmoidea, que asigna suavemente cualquier número real a un número entre 0 y 1. Cuando se varían sus parámetros se puede controlar dónde ocurre dicho cambio (Zheng & Casari, 2018) y su fórmula matemática es la siguiente:

$$f(x) = \frac{1}{1 + e^{-x}}$$

K Nearest Neighbours: El modelo KNN es un clasificador de aprendizaje no paramétrico donde su característica principal es usar la proximidad de los datos para hacer predicciones o clasificaciones y parte de la suposición encontrando puntos similares uno cercano del otro.

Para determinar las distancias entre los puntos se usa un punto central y se mide con los demás, es decir, se forma un radio que luego forma espacios o regiones, señalando cuantos puntos k deben estar en esa región. (Zaki & Meira, 2014).

Los modelos más usados son la distancia euclidiana que mide con una línea recta entre el punto central y los próximos.

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(y_i - x_i)^2}$$

La distancia de manhattan mide el valor absoluto de los puntos y su adecuación es muy parecida a la de la cuadrícula de las calles de una ciudad.

$$d(x,y) = \left(\sum_{i=1}^{m}|x_i - y_i|\right)$$

Support Vector Machines: Las máquinas de soporte vectoriales son un clasificador binario donde separa las clases por medio de los puntos usando un hiperplano óptimo. En este punto se crean muchos hiperplanos, pero el modelo seleccionará el del margen máximo. Donde el margen viene siendo la distancia entre el clasificador y los puntos de entrenamiento. (Saifuk Bahari, Ahmad, & Aboobaider, 2014). Estas son algunas de sus variantes en el caso de las fórmulas matemáticas[1] en la que se basan los siguientes modelos y usados en este proyecto:

Función de base radial (RBF) o gaussiana:

$$K(x_1, x_2) = exp\left(-\frac{\|x_1 - x_2\|^2}{2\sigma^2}\right)$$

[1] Los modelos matemáticos fueron obtenidos en: https://iopscience.iop.org/article/10.1088/1755-1315/20/1/012038/pdf

Lineal:

$$K(x_1, x_2) = x_1^t x_2$$

Polinómica:

$$K(x_1, x_2) = (x_1^t x_2 + 1)^\rho$$

Decision Tree Classifier: Es un modelo que hace predicciones simples planteando una serie de pruebas sobre un punto dado. La manera más fácil de verlo es como un árbol donde los nodos representan las pruebas y en las hojas se encuentran los valores a predecir y se empieza a descender por cada uno de los nodos. (Nasiriany, Thomas, Wang, & Yang, 2019)

Entropía y ganancia de información:

$$Entropy(S) = -\sum_{c \in C} p(c)\, log_2\, p(c)$$

Impureza de Gini:

$$Gini\ Impurity = 1 - \sum_i (p_i)^2$$

Random Forest Classifier: El random forest es un modelo no paramétrico y es un método de conjunto que evalúa múltiples árboles de decisiones y se basa en la agregación de resultados de un conjunto de estimadores más simples. Su resultado final se da por votación donde se escoge la mayor votación. A medida que los estimadores generan aportes, se pueden generar mejores resultados que usando los estimadores individualmente. (VanderPlas, 2017)

4.4.Métricas

Para abordar el problema de clasificación de los accidentes cerebrovasculares se tienen unas métricas para conocer qué tan acertada son las predicciones. Una de las medidas más importantes es la matriz de confusión[2] donde clasifica los datos de prueba en un recuadro de dos por dos, en los casos donde la variable de salida es binaria, y se clasifican los registros si fueron bien o mal

[2] La información aquí contenida está disponible en el sitio web de scikit-learn en el enlace: https://scikit-learn.org/stable/modules/model_evaluation.html#confusion-matrix

acertados. Este método abarca la siguiente información: Los verdaderos positivos (TP) donde los valores positivos son clasificados como positivos, los verdaderos negativos (TN) cuando los registros negativos son clasificados como negativos, los falsos positivos (FP) que sucede cuando el valor real del registro es negativo y el modelo lo clasifica como positivo y por último los falsos negativos (FN) que ocurren cuando el valor real del registro es positivo y el modelo lo clasifica como negativo. Importante tener en cuenta que lo ideal es que tanto los falsos negativos como los falsos positivos sean mínimos en esta matriz de confusión.

Después de obtenidos los resultados de la matriz de confusión, se busca complementar los resultados con unos scores o métricas que van a dar unos porcentajes de la efectividad del modelo:

***Accuracy*:** Corresponde el porcentaje total de elementos clasificados correctamente es decir tanto los verdaderos positivos y los verdaderos negativos dividido la muestra total de los registros de prueba.

$$\frac{TP + TN}{TP + FP + FN + TN}$$

No es recomendable usar esta medida si los datos se encuentran desbalanceados.

***Recall o sensibilidad*:** esta medida permite conocer la tasa de registros que fueron clasificados como positivos del total de positivos.

$$\frac{TP}{TP + FN}$$

***Precisión*:** es el número de elementos clasificados como positivos del total de elementos encontrados como positivos.

$$\frac{TP}{TP + FP}$$

***Especificidad*:** Se refiere a los registros identificados correctamente como negativos de todos los registros negativos.

$$\frac{TN}{TN + FP}$$

***F1-score*:** esta métrica es usada en bases de datos donde la variable de salida está desbalanceada y combina las métricas de precisión y recall.

$$2 * \left(\frac{recall * precisión}{recall + precisión}\right)$$

Curva ROC

La curva ROC es otra de las medidas a usar especialmente en problemas de clasificación, en ella se busca generar una curva mediante los puntos de la proporción de los verdaderos positivos con la proporción de los falsos positivos. Mientras la curva esté más desplazada hacia la esquina superior izquierda quiere decir que posee una mejor predicción.

AUC

Otras de las medidas que acompaña la curva ROC es el Área bajo la curva o por sus siglas en inglés Area Under the Curve y tiene que ver con el área que representa la curva ROC. La manera de entender los resultados es que mientras más cercano a 1 es mucho mejor y 0.5 es una predicción del 50 % que es la línea diagonal.

Conociendo estas medidas, se plantea abordar el ejercicio de clasificación con las métricas que aportan especialmente la matriz de confusión que en este caso serían el f1-score, el recall, y la curva ROC – AUC debido a que son las mejores métricas en términos de base de datos desbalanceados. El f1-score como se mencionó anteriormente posee la armonía de las variables recall y precisión, de gran utilidad en base de datos desbalanceadas, El recall importante en este estudio aborda los casos positivos ya que son los de la clase minoritaria y son de por sí los que en cuestiones médicas deben ser analizados con más acierto por ser una complicación médica con más alcance, por ello también será una medida de interés en este estudio y para finalizar también la curva ROC – AUC para verificar la capacidad de predicción de los modelos.

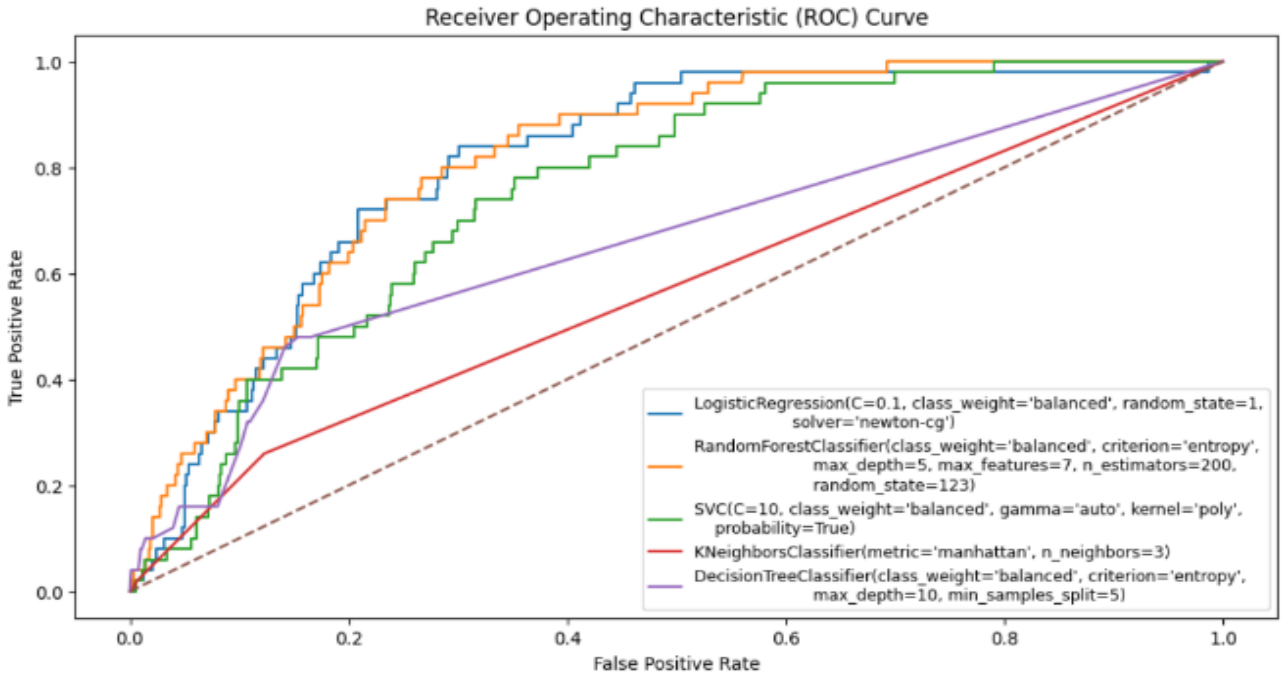
5. Metodología

5.1. Baseline

La **primera iteración** se usó la base de datos con su variable objetivo con clases desbalanceadas, pero agregando el parámetro class_weight = 'balanced', que es un parámetro que se encarga, en lugar de crear nuevas muestras o disminuir las existentes, de otorgar pesos inversamente proporcionales a las dos clases y castigar la clase mayoritaria sesgada. Cabe resaltar que el modelo de K Nearest Neighbor no tiene este parámetro, por lo cual se evidencia que obtuvo los peores resultados, (ver **Tabla *VIII***), además requiere que la variable de salida esté balanceada. Adicional, se usó GridSearchCV para buscar los mejores hiperparámetros basado en la métrica del f1-score usando el parámetro scoring. También se analizó la curva ROC-AUC (ver **Figura 12**), tanto en su métrica AUC como en la distancia que toma la curva con respecto a la línea diagonal.

La medida AUC de la curva ROC obtuvo sus mejores resultados en el modelo random forest con 0.82. Este buen rendimiento se puede deber que al clasificar bien la clase mayoritaria que es la que mayor peso posee entonces mejora la métrica en este sentido a nivel global.

Figura 12
Curva ROC para todos los modelos con variable de salida desbalanceada



Ahora en la evaluación de los modelos de la métrica f1-score, se evidencia unos resultados muy bajos y esto se debe al contar con la variable de salida con clases desbalanceadas. El modelo castiga demasiado la clase de salida *1* al tener menos datos.

Tabla VIII

Métrica y parámetros de la primera iteración

Modelos	f1-score	Recall	AUC	Parámetros
Regresión Logística	0.23	0.74	0.81	C = 0.1, penalty = 'l2', solver = 'newton-cg', class_weight = 'balanced'
Random Forest	0.24	0.66	0.82	max_features = 7, n_estimators = 200, random_state = 123, class_weight = 'balanced', criterion = 'entropy', max_depth = 5
Support Vector Machine	0.18	0.52	0.76	C = 10, class_weight = 'balanced', gamma = 'auto', kernel = 'poly', probability = True
K Nearest Neighbor*	0.04	0.02	0.57	n_neighbors = 3, metric = 'manhattan', weights = 'uniform'
Decision Tree	0.21	0.48	0.66	criterion = 'entropy', max_depth = 10, min_samples_split = 2, class_weight = 'balanced'

* K Nearest Neighbor fue el único modelo que no cuenta con el parámetro *class_weight* dentro de su configuración.

Adicional se midieron los registros de prueba que fueron sustraídos de la base de datos con el fin de evaluar los modelos y posibles sobreajustes y los resultados se pueden ver en la **Tabla *IX***.

Tabla IX

Resultado de los datos test de la primera iteración

Modelos	Clase	
	0	1
Regresión Logística	0.758	0.8
Random Forest	0.8375	0.8
Support Vector Machine	0.8	0.8
K Nearest Neighbor	0.9958	0
Decision Tree	0.85	0.4

Nota: Los datos test corresponden a 240 muestras de la clase *0* y 10 muestras de la clase *1* que corresponden a menos del 5 %

Los resultados muestran que el modelo Random Forest tiene los mejores resultados prediciendo las dos clases.

5.2. Validación

Para las particiones entrenamiento, validación y test se dividieron de la siguiente manera: El 5 % de los datos se sacaron de manera aleatoria, pero conservando la misma proporción tanto para las muestras positivas y negativas de la variable de salida. Cabe resaltar que este procedimiento donde la variable de salida está desbalanceada, no es posible obtener muestras lo suficientemente grande como lo dice la teoría para realizar este tipo de pruebas, pero según la naturaleza desbalanceada de la misma se considera que la muestra puede dar indicios sobre el correcto funcionamiento de los modelos al probar con muestras reales. Ahora explicado lo anterior, de ese 95 % de datos restantes el 80 % se separó como datos de entrenamiento y el 20 % restante como datos de validación, esto con el fin de entrenar muy bien el modelo debido a que la variable de salida en su clase positiva cuenta con muy pocos registros y el objetivo es tener un modelo mejor entrenado.

5.3. Iteraciones y evolución

Las iteraciones dos (2), tres (3) y cuatro (4) están enfocadas en tener la variable de salida balanceada como primera medida y aquí se usó la técnica SMOTE para dicho propósito, (ver **Figura *13***). Esta técnica es de gran utilidad porque a diferencia del random oversampling, SMOTE no duplica registros, al contrario, crea registros sintéticos muy parecidos a los originales logrando que se aumente la variedad de las muestras. Al principio del proyecto también se evidenció que la base de datos no contaba con registros repetidos logrando así mantener la identidad de la base de datos.

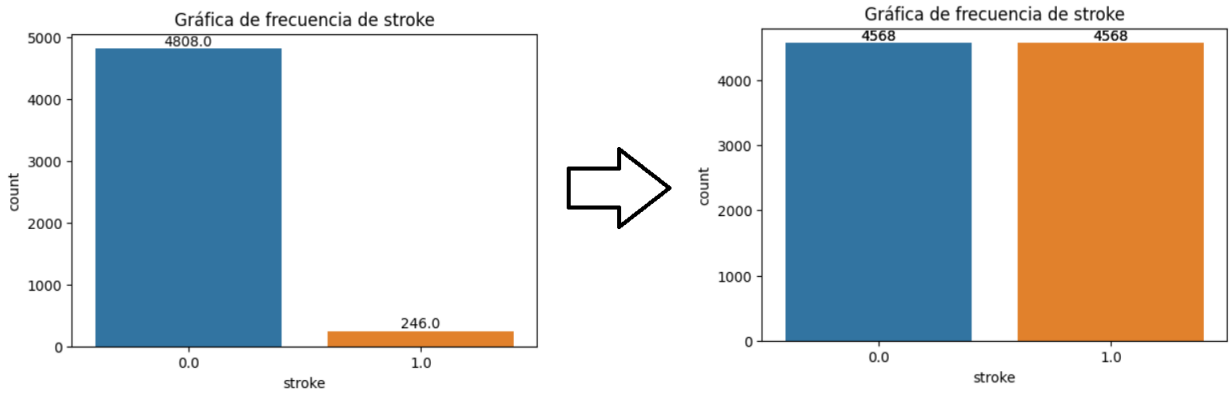
.

En la **iteración dos** los modelos fueron usados con sus parámetros por default o con pocos de ellos para comprobar que tanto mejoraban las métricas con el solo hecho de balancear las clases de salida.

La **tercera iteración** se usó GridSearchCV basada en cross validation con 5 pliegues buscando los mejores hiperparámetros para cada modelo y aquí se encontró que los modelos mejoraban en sus métricas f1-score, recall y AUC. También se realizó una nueva prueba con otro modelo de máquina de soporte de vectores con el parámetro kernel = ‘poly’, esto debido al gran consumo en el costo computacional y se llamó SVM1.

La **cuarta iteración** se centró un buscar posibles clases que generaban más ruido en los modelos y como se explicó al principio de este proyecto, la variable *work_type* de la clase *Never_worked* solo aporta 22 registros al modelo y todos pertenecen a la clase mayoritaria de la variable de salida, en este caso se decide eliminarla. La otra clase que se eliminó fue *children* de la variable *work_type* debido a que al crear una nueva matriz de correlación con las variables dummies está contaba con un valor de -0.63 lo que la hace tener una correlación media-alta, además solo cuenta con dos registros positivos de la clase minoritaria lo que es bueno porque no se pierde mucha información de esta clase.

Figura 13
Balanceo de la variable de salida por SMOTE



La variable de salida balanceada termina con 9136 registros distribuidos en partes iguales para ambas clases. Tener en cuenta que de la base de datos original se extrajo la clase *Other* de la variable *gender*, también se removieron 55 datos atípicos y además 10 muestras de la clase *1* y 240 muestras de la clase *0* de la variable de salida *stroke* para usarlos como prueba.

Continuando con el análisis después de aplicado SMOTE, se encuentra que las variables numéricas conservaron el peso y distribución de los registros positivos de *stroke*, esto se evidencia en la **Figura *14*** y **Figura *15*** donde se ven que las medidas de los boxplot de las variables numéricas balanceadas y desbalanceadas conservan su misma estructura. Se evidencia que, aunque el cambio fue fuerte, esto se debe a que la base de datos tenía un desbalance fuerte con alrededor del 5 % de la clase *1* de la variable *stroke*.

Figura 14

Variables numéricas balanceadas

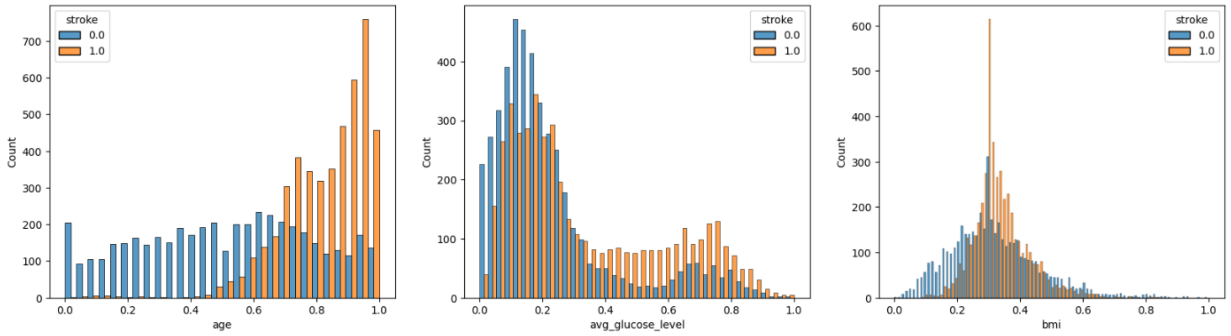
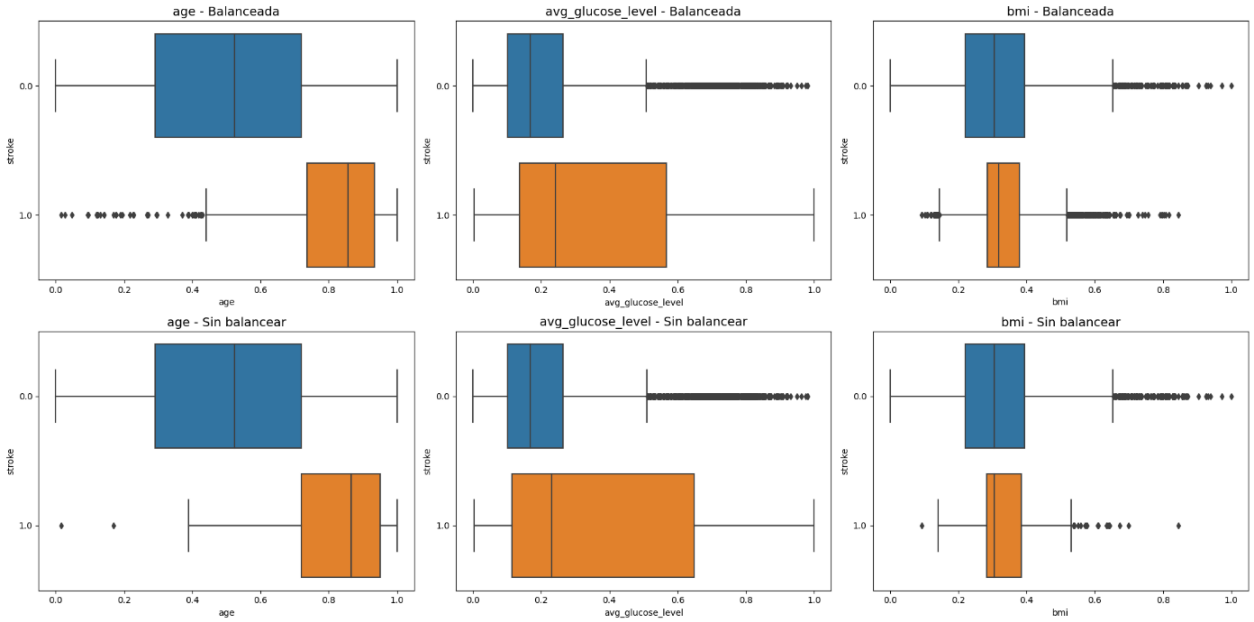


Figura 15

Comparación de los boxplot de la variable de salida en las variables numéricas. Sin balacear y balanceadas



5.4 Herramientas

Las herramientas usadas en este proyecto como programa de escritura y procesamiento de código fue Google Colaboratory y una laptop ASUS.

6. Resultados y discusión

En la **segunda iteración**, después de aplicada la técnica SMOTE, se inicia sin usar hiperparámetros, solo se usó el modelo con parámetros simples y con los que ofrecen los modelos por default con el fin de ver que tanto mejoraban los resultados. A continuación, en la **Tabla *X*** se puede visualizar el cambio en comparación con la primera iteración.

Tabla X

Métricas y parámetros de la segunda iteración

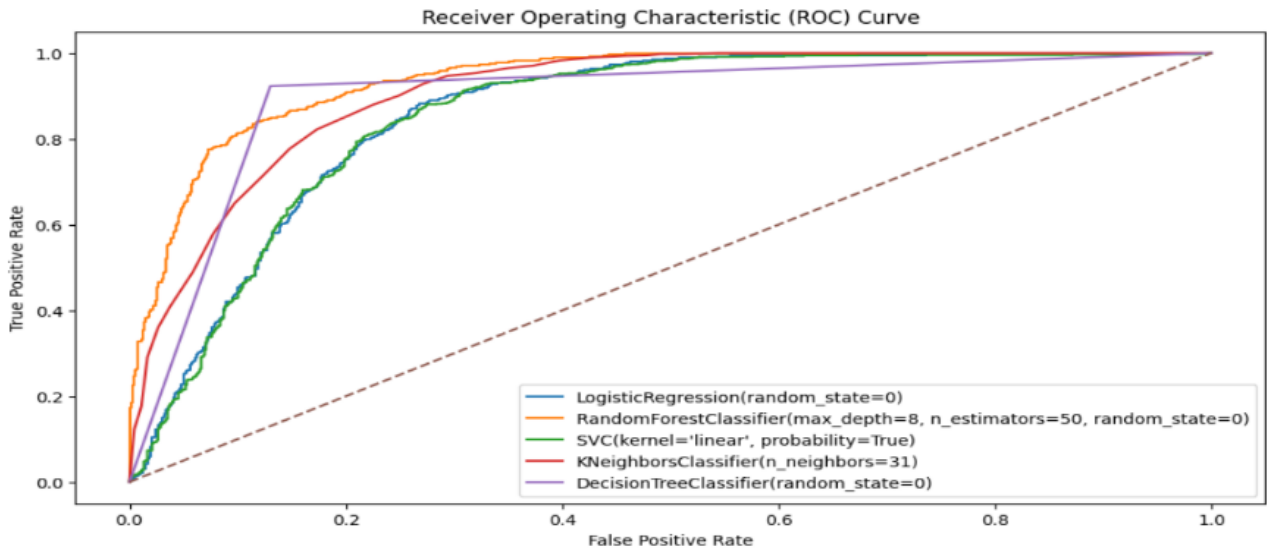
Modelos	f1-score	Recall	AUC	Parámetros
Regresión Logística	0.80	0.82	0.85	random_state = 0
Random Forest	0.86	0.9	0.93	n_estimators = 50, random_state = 0, max_depth = 8
Support Vector Machine	0.81	0.84	0.85	kernel = 'linear', probability = True
K Nearest Neighbor	0.84	0.92	0.9	n_neighbors = 31
Decision Tree	0.90	0.86	0.86	random_state = 0

Como se puede evidenciar fueron modelos básicos, pero arrojaron mejores medidas al aplicar la técnica de SMOTE, logrando que el modelo pudiera entrenarse mejor con mayor variedad de registros, en este caso los árboles de decisión obtienen 0.9 de la métrica f1-score siendo la mejor en este sentido.

Para la curva ROC – AUC, (ver **Figura 16**), los modelos también muestran un mejor rendimiento por lo antes ya explicado.

Figura 16

Curva ROC para todos los modelos con variable de salida balanceada



En este caso, el modelo random forest sigue mostrando el mejor puntaje con un AUC de 0.93 pero es importante notar que los modelos sean capaces de generalizar con muestras reales, los resultados de esta prueba se encuentran en la **Tabla *XI***.

Tabla XI

Resultado de los datos test de la segunda iteración

Modelos	**Clase**	
	0	1
Regresión Logística	0.775	0.8
Random Forest	0.8	0.6
Support Vector Machine	0.7583	0.8
K Nearest Neighbor	0.7125	0.5
Decision Tree	0.875	0.2

Nota: Los datos test corresponden a 240 muestras de la clase 0 y 10 muestras de la clase 1 que corresponden a menos del 5 %

Los resultados muestran que los modelos bajaron la capacidad de predecir y puede ser entendible debido a que se crearon datos sintéticos en gran cantidad y se aumentaron la cantidad de parámetros en los modelos, adicional la base de datos tiene poca dimensionalidad y esto puede alterar la manera de como el modelo generaliza, en este caso la regresión logística junto con el modelo support vector machine tienen buenos resultados al momento de clasificar muestras reales.

La **tercera iteración** se usaron los mismos modelos, pero esta vez buscando los mejores hiperparámetros y además usando la técnica de la validación cruzada para obtener unas métricas más acertadas y realistas (ver **Tabla *XII***). En la búsqueda de los hiperparámetros se usó como en la primera iteración el scoring f1 y además la cantidad de pliegues fue de 5 para usar el 20 % como datos de validación. Adicional se usaron dos modelos de máquina de soporte de vectores, en uno con el kernel linear o rbf y en el segundo con el kernel poly, esto debido a que este último tiene mayor costo computacional.

Tabla XII
Métrica y parámetros de la tercera iteración

Modelos	f1-score	Recall	AUC	CV_f1	Parámetros
Regresión Logística	0.81	0.88	0.84	0.8	C = 0.01, penalty = 'l1', solver = 'liblinear', multi_class = 'ovr', random_state = 1
Random Forest	0.93	0.96	0.98	0.92	max_features = 5, n_estimators = 150, random_state = 123, criterion = 'gini', max_depth = 20
SVM	0.93	0.95	0.95	0.85	C = 7, gamma = 10, kernel = 'rbf', random_state = 123, probability = True
SVM1 (poly)	0.87	0.89	0.92	0.9	C = 7, degree = 3, kernel = 'poly', random_state = 0, probability = True
K Nearest Neighbor	0.91	0.95	0.95	0.88	n_neighbors = 5, metric = 'manhattan'
Decision Tree	0.91	0.91	0.9	0.87	criterion = 'entropy', max_depth = 30, min_samples_split = 5

Los resultados muestran que, en esta iteración, el f1-score en los modelos random forest y máquinas de soporte de vectores tienen un puntaje de 0.93 y cuando se configura con el cross validation sus métricas se reducen en 0.92 y 0.85 respectivamente. En este caso random forest tiene su mejor medida con el cross validation en 0.92.

Cabe notar algo al observar la curva ROC- AUC, (ver **Figura 17**), es que los modelos obtienen curvas altas cercanas a uno (1) a excepción de la regresión logística que se ajusta en 0.84. Debido a esto se validarán los datos de prueba y los resultados se encuentran en la **Tabla *XIII*** para descartar un posible sobreajuste de los modelos.

Figura 17

Curva ROC para todos los modelos con variable de salida balanceada y con hiperparámetros

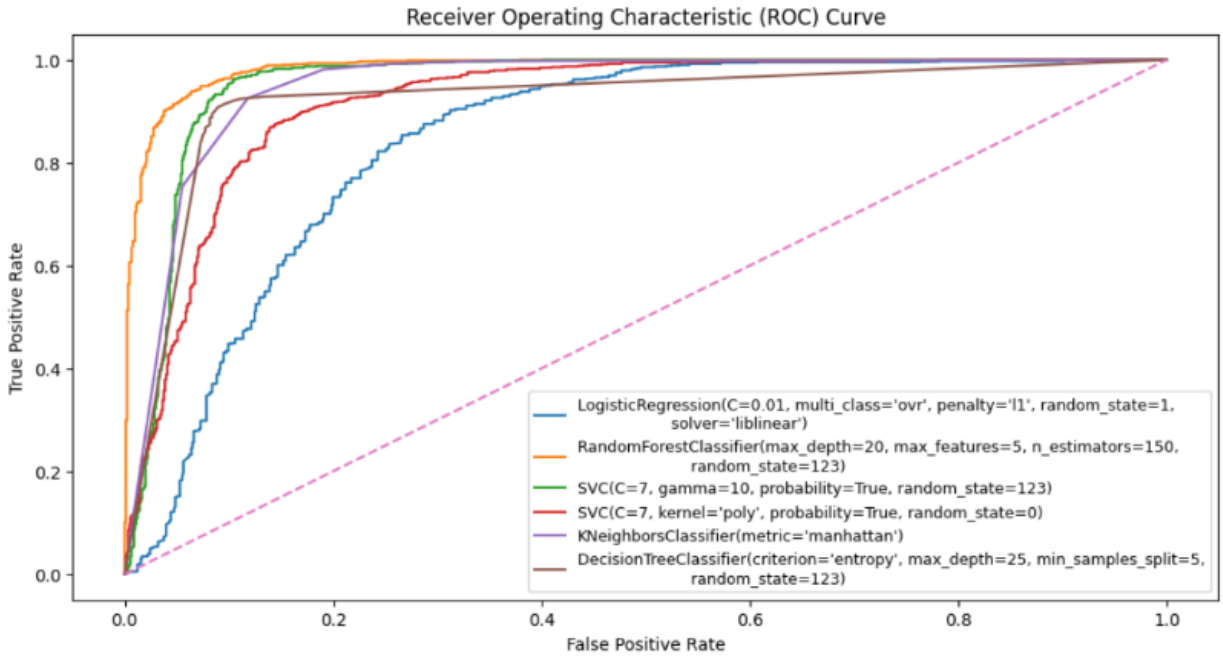


Tabla XIII

Resultado de los datos de prueba de la tercera iteración

Modelos	**Clase**	
	0	1
Regresión Logística	0.729	0.8
Random Forest	0.891	0.2
Support Vector Machine	0.8708	0.2
Support Vector Machine (Poly)	0.8375	0.2
K Nearest Neighbor	0.8208	0.2
Decision Tree	0.9041	0.4

Nota: Los datos test corresponden a 240 muestras de la clase 0 y 10 muestras de la clase 1 que corresponden a menos del 5 %

En este punto se comprueba lo observado anteriormente y es que los modelos identificaron pocos registros reales de la clase *1* de la variable de salida, posiblemente por sobreajuste, a diferencia de la regresión logística que clasifica mejor las dos clases. La regresión logística obtuvo una medida de 0.8 en el cross validation.

La **cuarta iteración** se llevó a cabo con la eliminación de dos clases, *Never_worked* de la variable *work_type* debido a que solo tenía 22 registros de los cuales ninguno correspondía a la clase *1* de la variable de salida y segundo de la matriz de correlación que se generó al aplicar dummies la variable *work_type_children* posee una correlación negativa de -0.63 contra *age* por lo cual se descarta esta variable y se obtienen los siguientes resultados mostrados en la **Tabla *XIV***.

Tabla XIV

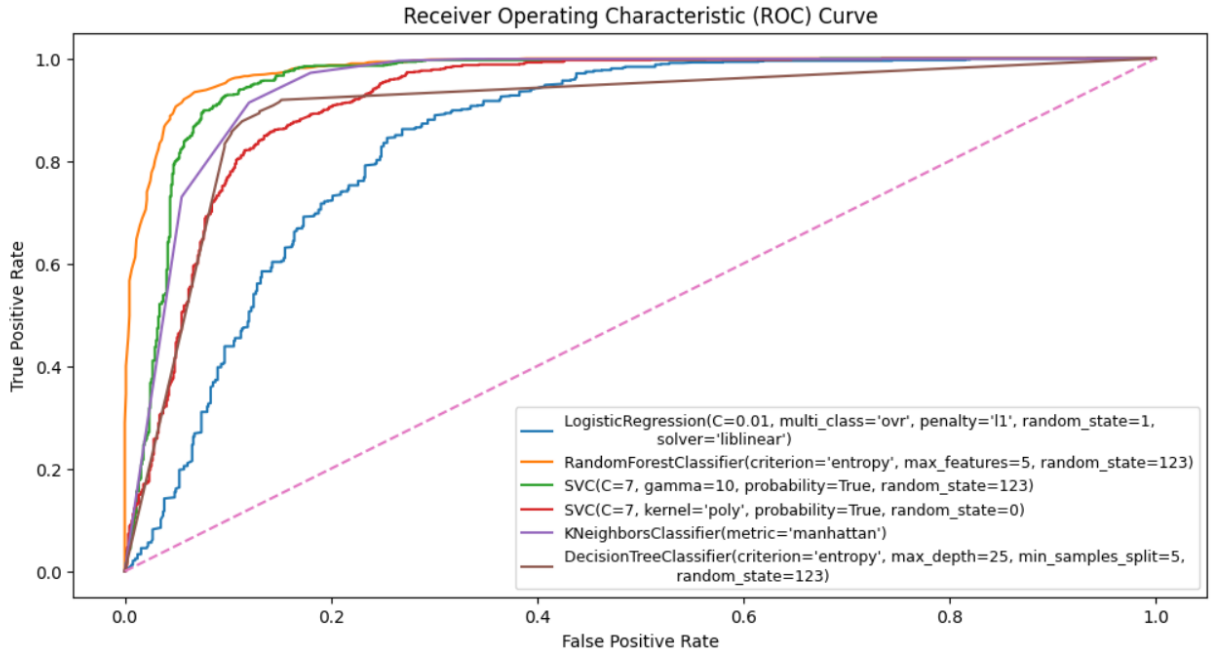
Métricas y parámetros de la cuarta iteración

Modelos	f1-score	Recall	AUC	CV_f1	Parámetros
Regresión Logística	0.82	0.88	0.84	0.8	C = 0.01, penalty = 'l1', solver = 'liblinear', multi_class = 'ovr', random_state = 1
Random Forest	0.93	0.97	0.98	0.92	max_features = 5, n_estimators = 100, random_state = 123, criterion = 'entropy', max_depth = None
SVM	0.92	0.96	0.96	0.86	C = 7, gamma = 10, kernel = 'rbf', random_state = 123, probability = True
SVM poly	0.86	0.92	0.92	0.9	C = 7, degree = 3, kernel = 'poly', random_state = 0, probability = True
K Nearest Neighbor	0.91	0.98	0.95	0.88	n_neighbors = 5, metric = 'manhattan'
Decision Tree	0.89	0.89	0.89	0.88	criterion = 'entropy', max_depth = 25, min_samples_split = 5, random_state = 123

Los resultados son similares a la tercera iteración donde se ve el modelo de random forest con un puntaje de cross validation de 0.92 en la métrica f1-score. Si se observa la curva ROC-AUC, (ver **Figura *18***), también es similar a la **Figura 17** al notar que las curvas se acercan a uno (1).

Figura 18

Curva ROC para todos los modelos en la cuarta iteración



Para medir la capacidad de los modelos de generalizar en muestras reales se hizo la prueba con los registros dispuestos para esto. Los resultados se pueden ver en la **Tabla *XV***.

Tabla XV

Resultado de los datos de prueba de la cuarta iteración

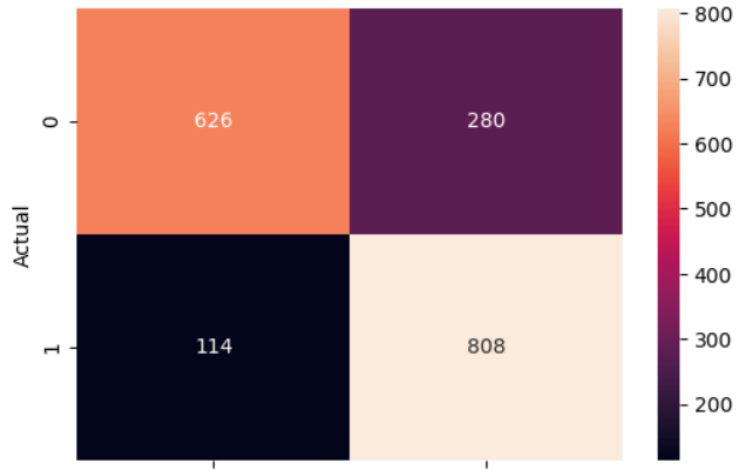
Modelos	**Clase**	
	0	1
Regresión Logística	0.616	0.8
Random Forest	0.86	0.2
Support Vector Machine	0.85	0.1
Support Vector Machine (Poly)	0.78	0.2
K Nearest Neighbor	0.791	0.3
Decision Tree	0.825	0.2

Nota: Los datos test corresponden a 240 muestras de la clase 0 y 10 muestras de la clase 1 que corresponden a menos del 5 %

Los resultados anteriores sugieren que el modelo que mejor generaliza tanto muestras de clase *0* como *1* es la regresión logística. Su medida f1-score con cross validation fue de 0.8.

Por ahora el modelo que más estabilidad tuvo en todas las iteraciones fue la regresión logística de la tercera iteración que pudo clasificar bien los registros de prueba que se separaron al principio del problema. En la clase *1* pudo clasificar bien 8 de los 10 registros, que es de los mejores en este proyecto, además se obtuvo buenos resultados en la clase *0* con un 72.5 %, lo anterior teniendo en cuenta que se usaron métricas enfocadas en la clase minoritaria como el f1-score en GridSearchCV para mejorar esta clase. Adicional, se probó que el modelo no tuviera sobreajuste midiendo el rendimiento de los datos de entrenamiento y validación basados en el f1-score y se encuentra que los datos de entrenamiento tienen un rendimiento de 0.88 y los datos de validación tienen un rendimiento de 0.87, (ver **Tabla *XVII***), por lo cual se evidencia que generaliza bien. Para conocer los detalles completos de este modelo se muestra la matriz de confusión correspondiente a la regresión logística de la tercera iteración en la **Figura 19**.

Figura 19
Matriz de confusión del modelo principal



	precision	recall	f1-score	support
0.0	0.85	0.69	0.76	906
1.0	0.74	0.88	0.80	922
accuracy			0.78	1828
macro avg	0.79	0.78	0.78	1828
weighted avg	0.79	0.78	0.78	1828

Para responder a la pregunta de cuáles son las características que más aportan al modelo se hace una búsqueda por medio de la importancia de las características (ver **Tabla *XVI***). La idea es conocer cuáles son las variables que más influyen en el problema de los accidentes cerebro vasculares teniendo en cuenta las variables que se usaron en este proyecto. En este caso se usaron

la importancia de las características obtenidas en el modelo random forest de la tercera iteración. Esto no es determinante en el área médica debido que no se tuvo en cuenta muchas otras características que no se incluyeron en esta base de datos y no se deben generalizar estos resultados. El único objetivo es conocer en esta base de datos, cuáles son las características que más influyen al momento de generar los modelos.

Tabla XVI
Importancia de las características

Características	Importancia
age	0.4020
avg_glucose_level	0.1758
bmi	0.1749
hypertension	0.0355
gender_Male	0.0266
heart_disease	0.0266
Residence_type_Urban	0.0254
smoking_status_never smoked	0.0240
ever_married_Yes	0.0221
work_type_Private	0.0221
work_type_Self-emplyed	0.0176
smoking_status_formerly smoked	0.0166
smoking_status_smokes	0.0165
work_type_children	0.0141
work_type_Never_worked	0.000

La anterior tabla muestra que la característica que más influyen en esta base de datos es *age*, junto con las otras dos variables numéricas que son *avg_glucose_level* y *bmi*. Por último, lugar se encuentran *work_type_children* y *work_type_Never_worked* que precisamente son las dos variables que se eliminaron en la cuarta iteración porque según lo observado no aportaban al modelo.

6.1. Evaluación cualitativa

De las 4 iteraciones realizadas se puede destacar que a medida que se aumentaban las especificaciones de los modelos con las técnicas de búsqueda de hiperparámetros mediante GridSearchCV, los modelos iban perdiendo poder de predicción en la clase minoritaria. Esto se notó cuando se usaron los datos de prueba para que los modelos clasificaran con muestras reales pero su capacidad de generalizar disminuyó. Se hizo una prueba de rendimiento de los datos de entrenamiento y los datos de validación y se encontró que muchos de los modelos donde se pensaba que había sobreajuste, los dos rendimientos se acercaban bastante logrando mostrar que podrían generalizar bien. Al final se decidió el modelo de regresión logística de la tercera iteración debido a que por un lado tenía buenas medidas entre datos de entrenamiento y validación, también podía generalizar bien la prueba con 10 registros positivos. En la **Tabla *XVII*** se encuentra la información descrita para cada modelo.

Tabla XVII

Resumen de todas las iteraciones y sus métricas

Modelos	Iteración	F1	Recall	Test_Clase_0	Test_Clase_1	AUC	Rendimiento f1		CVf1 f1
							Entren.	Valid.	
RL	1	0,23	0,74	0.76	0.8	0,81	0,23	0,23	-
RF	1	0,24	0,66	0.83	0.8	0,82	0,29	0,24	-
SVM	1	0,18	0,52	0.8	0.8	0,76	0,25	0,18	-
KNN	1	0,04	0,02	0.99	0	0,57	0,29	0,03	-
DTC	1	0,21	0,48	0.85	0.4	0,66	0,41	0,21	-
RL	2	0,8	0,82	0.77	0.8	0,85	0,78	0,8	-
RF	2	0,86	0,9	0.8	0.6	0,93	0,87	0,85	-
SVM	2	0,81	0,84	0.75	0.8	0,85	0,76	0,8	-
KNN	2	0,84	0,92	0.71	0.5	0,9	0,84	0,83	-
DTC	2	0,9	0,86	0.87	0.2	0,86	1	0,86	-
RL	3	0,81	0,88	0.73	0.8	0,84	0,88	0,87	0,8
RFf1	3	0,93	0,96	0.89	0.2	0,98	1	0,95	0,92
SVM	3	0,93	0,95	0.87	0.2	0,95	0,97	0,95	0,85
SVM1	3	0,87	0,89	0.83	0.2	0,92	0,9	0,88	0,9
KNN	3	0,91	0,95	0.82	0.2	0,95	0,98	0,95	0,88
DTC	3	0,91	0,91	0.90	0.4	0,9	0,98	0,9	0,87
RL	4	0,82	0,88	0.61	0.8	0,84	0,88	0,87	0,8
RFf1	4	0,93	0,97	0.86	0.2	0,98	1	0,96	0,92
SVM	4	0,92	0,96	0.85	0.1	0,96	0,97	0,96	0,86
SVM1	4	0,86	0,92	0.78	0.2	0,92	0,91	0,91	0,9
KNN	4	0,91	0,98	0.79	0.3	0,95	0,98	0,98	0,88
DTC	4	0,89	0,89	0.82	0.2	0,89	0,97	0,89	0,88

7. Conclusiones

De las 4 iteraciones realizadas en este trabajo, los desempeños obtenidos son bastantes buenos incluso en la primera iteración donde se usa el parámetro class_weight = balanced para otorgar mayor peso a la clase minoritaria sin necesidad de balancear la variable de salida, esto se comprueba al usar los registros de prueba y obtener buenos resultados con registros reales.

En la segunda iteración el modelo de la regresión logística fue el que mejor generalizó los datos de prueba, aunque el mayor f1-score se consiguió con los árboles de decisión. A partir de la tercera iteración el modelo de la regresión logística fue el que mantuvo buenos resultados al momento de generalizar con registros reales. De las 4 iteraciones, la regresión logística, solo en la primera iteración no obtuvo resultados del f1-score por encima del 80 % y se debe a que en esa iteración todavía no se había balanceado la variable de salida.

La gran limitante que se evidenció a lo largo de este trabajo fue obtener una buena muestra de registros de prueba para conocer que el modelo no se estuviera sobre ajustando. En este proyecto hubo modelos con mejores métricas que la regresión logística, pero al medirlo con muestras reales no clasificaron bien. Lo ideal es una base de datos con mayor cantidad de registros, pero al conocer que la variable de salida está desbalanceada no se pueden usar muchos datos para prueba porque se perdería información importante al momento de partir los registros y usarlos como entrenamiento y validación. Los pocos registros que se usaron corresponden a la realidad de la variable y por lo tanto se usó un número adecuado a la necesidad de conocer el comportamiento de los modelos a la hora de clasificar.

Acerca de los objetivos que se plantearon en este proyecto, se cumplieron todos menos el principal que era predecir por encima del 85 % en sus métricas acordadas. Hubo modelos que en sus medidas lograron los objetivos, pero como se dijo anteriormente no hubo buena impresión cuando interactuaron con registros reales. Por lo cual no se decidió tomarlos como modelos principales.

8. Recomendaciones

De acuerdo con lo encontrado en este trabajo la recomendación que se hace es no pretender buscar un solo modelo perfecto o un método de preprocesamiento, lo ideal es ir usando las diferentes técnicas y modelos con el fin de evaluar múltiples soluciones. Si se decide ir por un solo camino es posible sesgarse y no obtener los resultados esperados.

Es importante balancear las clases de la variable de salida ya que esta técnica mejora el rendimiento de los modelos y se pueden tener mejores resultados. Esto permite que los modelos no se sesguen con la clase mayoritaria y obtener métricas de igual manera sesgadas.

Medir con registros reales puede ser uno de los métodos para conocer si el modelo presenta sobreajuste. Esta práctica, como en muchos otros casos es una manera de permitir conocer que también predice o clasifica un modelo.

Referencias

Bolaños Vaillant, S., Gómez García, Y., Dosouto Infante, V., & Rodríguez Cheong, M. (12 de 03 de 2009). *scielo*. Obtenido de Tomografía axial computarizada en pacientes con enfermedades cerebrovasculares hemorrágicas: http://scielo.sld.cu/pdf/san/v13n5/san11509.pdf

Breunig, M. M., Kriegel, H.-P., Ng, R. T., & Sander, J. (2000). LOF: identifying density-based local outliers. En *Proceedings of the 2000 ACM SIGMOD international conference on Management of data* (págs. 93-104). New York: ACM.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artifiacial Inteligence Research*, 321-357.

Guerrero Agámez, D., Pestana Utria, G., Diaz Arrieta, B., Vargas Moranth, R., & Alvis Guzmán, N. (2021). Mortalidad por enfermedades cerebrovasculares en Colombia: 30 años de observación. En *Acta Neurológica Colombiana* (págs. 173-188).

Instituto Nacional de Salud, Observatorio Nacional de Salud. (2013). *Segundo Informe ONS. Mortalidad 1998-2011 y situación de salud en los municipios de frontera terrestre en Colombia*. Bogotá, D.C.: Imprenta Nacional de Colombia.

Ministerio de Salud y Protección Social. (30 de 10 de 2020). *minsalud.gov.co*. Obtenido de minsalud.gov.co: minsalud.gov.co

Ministerio de Sanidad y Consumo. (2008). *semg.es*. Obtenido de semg.es: https://www.semg.es/doc/documentos_SEMG/estrategias_ictus_SNS.pdf

Nasiriany, S., Thomas, G., Wang, W., & Yang, A. (2019). *A Comprehensive Guide to Machine Learning*. Berkeley.

National Health Services. (13 de 09 de 2022). *www.nhs.uk*. Obtenido de www.nhs.uk: https://www.nhs.uk/conditions/stroke/

OMS. (17 de 05 de 2017). *who.int*. Recuperado el 08 de 06 de 2023, de https://www.who.int/es/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)

OMS. (09 de 12 de 2020). *who.int*. Recuperado el 08 de 06 de 2023, de https://www.who.int/es/news-room/fact-sheets/detail/the-top-10-causes-of-death

RecaVar. (s.f.). *recavar.org*. Obtenido de recavar.org: www.recavar.org/quienes-somos-recavar

Saifuk Bahari, N. I., Ahmad, A., & Aboobaider, B. M. (2014). Application of support vector machine for classification of multispectral data. *IOP science*.

VanderPlas, J. (2017). *Python Data Science Handbook.* Sebastopol: O'Reilly.

Zaki, M. J., & Meira, W. J. (2014). *Data Mining and Analysis: Fundamental Concepts and Algorithms.* New York: Cambridge University.

Zheng, A., & Casari, A. (2018). *Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists.* Sebastopol: O'Reilly.