

UNIVERSIDAD PRIVADA ANTENOR ORREGO
FACULTAD DE INGENIERÍA



“Sistema de Inteligencia Artificial para la Detección Temprana de Accidentes Cerebrovasculares”

Carrera: Ing. Sistemas e Inteligencia Artificial

Grupo:

- Barrios Capa, Brad Janer
- Calderon Vera, Brayan Samuel
- Cubas Torres, Fabian
- Rodriguez Ruiz, Alessandro Paul
- Salas Torres, Juan
- Tapia Bravo, Ian
- Vergara Escobedo, Juan

Docente: Sagastegui Chigne, Teobaldo Hernan

Curso: Aprendizaje Estadístico

Trujillo - Perú

2025

Índice

I. Introducción	4
1.1 Título del Proyecto	4
1.2 Antecedentes	4
1.3 Problema a Resolver	5
1.4 Objetivos	5
II. Requerimientos	6
2.1 Definición del Dominio	6
2.2 Determinación de Requisitos	7
III. Preprocesamiento y Normalización	9
3.1 Medición	9
3.1.1 Dispositivos	9
3.1.2 Medidas y Base de Datos	9
3.2 Preprocesamiento	9
3.2.1 Filtrado y limpieza de datos	9
3.2.2 Extracción de características	9
3.2.3 Normalización	9
3.3 Normalización	10
3.3.1 Características de selección de normalización	10
3.3.2 Características de proyección	11
3.3.3 Reducción de dimensionalidad	11
IV. Aprendizaje	13
4.1 Planteamiento del Modelo de Aprendizaje	13
4.2 Desarrollo e Implementación del Modelo	15
V. Comprobación	18
5.1 Aplicación al Modelo: uso del Data-Set de Entrenamiento y de Prueba	18
VI. Evaluación	20
6.1 Evaluación del Desempeño del Modelo	20
6.2 Análisis de Resultados	21
6.3 Comparación con Otros Modelos	21
6.4 Evaluación de la Interpretabilidad	21
6.5 Validación en Entornos Reales	22
6.6 Limitaciones del Sistema	22
6.7 Mejoras Futuras	22
Referencias Bibliográficas	23

I. Introducción

1.1 Título del Proyecto

Sistema de Inteligencia Artificial para la Detección Temprana de Accidentes Cerebrovasculares.

1.2 Antecedentes

El accidente cerebrovascular (ACV) es un trastorno neurológico que ocurre cuando se interrumpe o se reduce el flujo sanguíneo al cerebro, lo que provoca la muerte de células nerviosas en minutos. Se clasifica principalmente en **isquémico** (por obstrucción de un vaso sanguíneo) y **hemorrágico** (por ruptura de un vaso sanguíneo). El primero representa cerca del 85% de los casos a nivel mundial.

Según la Organización Mundial de la Salud (OMS, 2023), el ACV es la **segunda causa de muerte** y la **tercera causa de discapacidad** en el mundo. Cada año afecta a millones de personas, sin distinción de nivel socioeconómico, aunque su impacto es mayor en países en desarrollo debido a la falta de acceso a servicios de salud especializados.

En América Latina, los ACV constituyen un problema creciente de salud pública. En Perú, por ejemplo, el Ministerio de Salud reporta que los ACV se encuentran entre las primeras cinco causas de muerte, con un incremento sostenido en la última década debido a factores como el envejecimiento de la población, el aumento de la hipertensión arterial, la diabetes y los hábitos poco saludables.

Los avances tecnológicos en las últimas décadas han permitido un mejor diagnóstico mediante imágenes médicas (tomografía computarizada, resonancia magnética), pero estas herramientas presentan limitaciones: su alto costo, la necesidad de infraestructura avanzada y la dependencia de especialistas entrenados. Esto provoca que en muchas regiones rurales el diagnóstico oportuno sea poco accesible, aumentando la desigualdad en la atención.

Por otro lado, la Inteligencia Artificial (IA) ha demostrado un gran potencial en el campo de la salud. Aplicaciones de aprendizaje automático y aprendizaje profundo ya se utilizan en la detección temprana de **cáncer, enfermedades cardíacas y trastornos neurológicos**, alcanzando niveles de precisión comparables o superiores a los diagnósticos tradicionales. En el caso de los ACV, la IA permite analizar grandes volúmenes de información clínica y biomédica en segundos, identificando patrones de riesgo invisibles para el ojo humano.

Diversos estudios recientes evidencian que la combinación de IA con información médica puede alcanzar precisiones superiores al 85% en la detección de riesgo de ACV, lo que abre la puerta a sistemas predictivos aplicables en la práctica clínica. Esta integración representa una oportunidad para **reducir la mortalidad, minimizar secuelas y mejorar la equidad en el acceso a diagnósticos tempranos**, sobre todo en regiones donde el tiempo de respuesta es determinante para salvar vidas.

1.3 Problema a Resolver

La detección tardía de los accidentes cerebrovasculares impide el inicio oportuno de tratamientos, lo que incrementa la mortalidad, provoca secuelas neurológicas irreversibles y genera altos costos de atención y rehabilitación. Además, la falta de acceso a diagnósticos rápidos y confiables, especialmente en zonas rurales, limita la equidad en la atención sanitaria.

El reto consiste en desarrollar soluciones innovadoras que automaticen la identificación temprana de ACV, reduzcan la dependencia exclusiva del juicio clínico y permitan un diagnóstico más rápido y preciso.

Problema formulado:

¿Cómo diseñar un sistema de inteligencia artificial que, a partir de información clínica y biomédica, detecte de forma temprana los accidentes cerebrovasculares con la precisión y rapidez necesarias para apoyar la toma de decisiones médicas y reducir las consecuencias negativas de este tipo de eventos?

1.4 Objetivos

Objetivo General

Diseñar e implementar un sistema de inteligencia artificial que permita la detección temprana de accidentes cerebrovasculares, mejorando el diagnóstico oportuno y contribuyendo a la reducción de la mortalidad y la discapacidad asociadas.

Objetivos Específicos

1. Identificar y analizar las variables clínicas y biomédicas asociadas a la ocurrencia de ACV.
2. Desarrollar procesos de preprocesamiento y normalización de datos para garantizar su calidad y consistencia.
3. Implementar modelos de inteligencia artificial que permitan la detección temprana de patrones de riesgo.
4. Validar el desempeño del sistema mediante métricas de precisión, sensibilidad, especificidad y AUC.
5. Proponer un marco de aplicación clínica que facilite la integración del sistema en entornos de salud, tanto urbanos como rurales.

II. Requerimientos

2.1 Definición del Dominio

Propósito:

Desarrollar y evaluar un sistema automático de inteligencia artificial para la **detección temprana de accidentes cerebrovasculares (ACV)**. El sistema busca identificar patrones de riesgo a partir de información clínica, biomédica y demográfica de pacientes, con el fin de apoyar la toma de decisiones médicas y reducir los tiempos de diagnóstico en entornos hospitalarios y rurales.

Entradas Principales:

- Datos clínicos (antecedentes de hipertensión, diabetes, enfermedades cardíacas, historial de ACV previo).
- Datos biomédicos (niveles de glucosa, índice de masa corporal, presión arterial, frecuencia cardíaca, entre otros).
- Datos demográficos (edad, género, hábitos de vida, residencia).

Enfoque Metodológico:

Aprendizaje supervisado y, en etapas exploratorias, técnicas de aprendizaje no supervisado. Los modelos de IA identificarán patrones en los datos clínicos y biomédicos mediante algoritmos como regresión logística, Random Forest, Gradient Boosting y redes neuronales profundas. El sistema incluirá un proceso de preprocesamiento de datos (limpieza, normalización y codificación) y se validará con métricas estándar de clasificación.

Salidas Principales:

- Predicción binaria: *riesgo de ACV o no riesgo de ACV*.
- Métricas de desempeño del modelo: **Accuracy, Precision, Recall, F1-Score y AUC**.
- Reportes visuales interpretables (curvas ROC, matrices de confusión, gráficos de importancia de variables).

Alcance:

- Recolección, limpieza y preprocesamiento de datos clínicos y biomédicos.
- Implementación de algoritmos de inteligencia artificial para detección temprana de ACV.
- Entrenamiento, validación y prueba de los modelos en entornos controlados.

- Generación de reportes de rendimiento y visualizaciones que permitan interpretar resultados.
- Propuesta de aplicación clínica como sistema de apoyo al diagnóstico.

Fuera de Alcance:

- Emisión de diagnósticos médicos definitivos sin supervisión profesional.
- Implementación en tiempo real en hospitales sin validación clínica previa.
- Uso de dispositivos médicos especializados para adquisición de datos (se asume que los datos ya están recolectados).
- Clasificación de subtipos de ACV (isquémico, hemorrágico).
- Desarrollo de interfaces gráficas avanzadas o sistemas web para despliegue inmediato.

2.2 Determinación de Requisitos

Requisitos Funcionales (FR):

- FR1: El sistema debe permitir la entrada y procesamiento de información clínica, biomédica y demográfica de los pacientes.
- FR2: El sistema debe realizar limpieza, transformación y normalización de los datos antes de su análisis.
- FR3: El sistema debe implementar modelos de inteligencia artificial capaces de identificar patrones predictivos de riesgo de ACV.
- FR4: El sistema debe generar predicciones binarias (riesgo / no riesgo) de manera automática.
- FR5: El sistema debe calcular métricas de desempeño (Accuracy, Precision, Recall, F1-Score, AUC).
- FR6: El sistema debe presentar resultados mediante reportes gráficos y numéricos interpretables.
- FR7: El sistema debe permitir la comparación entre distintos algoritmos de IA para seleccionar el más eficiente.
- FR8: El sistema debe contar con un módulo de validación cruzada para minimizar el sobreajuste y garantizar la generalización.

- FR9: El sistema debe almacenar los modelos entrenados para su reutilización futura.
- FR10: El sistema debe permitir la actualización periódica con nuevos datos para mejorar su precisión.
- FR11: El sistema debe ofrecer interpretabilidad, indicando las variables más influyentes en la predicción.
- FR12: El sistema debe generar informes finales que faciliten la toma de decisiones clínicas.

Requisitos No Funcionales (NFR):

- NFR1: El sistema debe alcanzar una precisión mínima del 85% y buscar métricas superiores al 90%.
- NFR2: El sistema debe entregar predicciones en menos de 5 segundos por paciente.
- NFR3: El sistema debe ser reproducible en diferentes entornos, controlando versiones y semillas aleatorias.
- NFR4: El sistema debe ser escalable para manejar desde cientos hasta miles de registros.
- NFR5: El sistema debe cumplir buenas prácticas de programación, documentación y pruebas de software.
- NFR6: El sistema debe garantizar la confidencialidad y privacidad de los datos médicos (ej. anonimización).
- NFR7: El sistema debe ser portable y ejecutarse en distintos entornos (local, servidor, nube).
- NFR8: El sistema debe ser robusto ante valores atípicos y datos incompletos.
- NFR9: El sistema debe garantizar interpretabilidad, evitando que sea una “caja negra” inentendible.
- NFR10: El sistema debe cumplir principios éticos, evitando sesgos por edad, género o condición socioeconómica.
- NFR11: El sistema debe integrarse en flujos de trabajo clínicos de forma sencilla y no intrusiva.
- NFR12: El sistema debe contar con logs de auditoría para rastrear decisiones y predicciones.

III. Preprocesamiento y Normalización

3.1 Medición

3.1.1 Dispositivos

Los datos son clínicos y demográficos ya recolectados, por lo que no se usan dispositivos para medición directa en este proyecto. Las variables provienen de historias clínicas y mediciones médicas previas, como mediciones de presión arterial, análisis de laboratorio, etc.

3.1.2 Medidas y Base de Datos

- Se utiliza el **Stroke Prediction Dataset** (Kaggle, fedesoriano). [Kaggle](#)
- Las instancias incluyen datos demográficos como edad, género, tipo de residencia, tipo de trabajo, así como condiciones médicas: hipertensión, enfermedades cardíacas, nivel de glucosa, etc. [Kaggle](#)
- La variable objetivo es binaria: si el paciente ha sufrido un ACV ("stroke") o no. [Kaggle](#)

3.2 Preprocesamiento

3.2.1 Filtrado y limpieza de datos

- Detección y tratamiento de valores faltantes.
- Verificación de consistencia de datos (por ejemplo, edades improbables, valores extremos).
- Eliminación o corrección de valores atípicos si se detectan que distorsionan significativamente el dataset.

3.2.2 Extracción de características

Aunque el dataset ya contiene variables clínicas/tabulares, se pueden generar variables derivadas, tales como interacciones entre variables (por ejemplo, hipertensión × frecuencia cardíaca, o glucosa normalizada), para mejorar los modelos.

3.2.3 Normalización

- Aplicar **Min-Max Scaling** a variables numéricas para llevarlas al rango [0,1].
- Aplicar **Estandarización (Z-score)** para otras variables numéricas si es necesario, especialmente si hay dispersión desigual.

- Codificación de variables categóricas (One-Hot Encoding o Label Encoding) para opciones como género, residencia, tipo de trabajo.

3.3 Normalización

3.3.1 Características de selección de normalización

Como se indicó en la sección de preprocesamiento (3.2.3), la normalización de los datos es un paso esencial para garantizar que todas las variables aporten de manera equilibrada al entrenamiento de los modelos de inteligencia artificial. En este proyecto se emplean dos enfoques principales:

- **Escalado Min-Max:** se aplica a variables numéricas continuas (por ejemplo, edad, niveles de glucosa, índice de masa corporal) con el fin de llevarlas a un rango común [0,1]. La fórmula utilizada es:

$$X' = (X - X_{\min}) / (X_{\max} - X_{\min})$$

$X \rightarrow$ valor original de la variable.

$X' \rightarrow$ valor normalizado en el rango [0,1].

$X_{\min} \rightarrow$ valor mínimo de la variable en el dataset.

$X_{\max} \rightarrow$ valor máximo de la variable en el dataset.

Esto asegura que ninguna variable con valores absolutos más grandes domine el proceso de aprendizaje.

- **Estandarización (Z-score):** utilizada cuando las variables presentan una dispersión desigual. En este caso, los datos se transforman para que tengan media 0 y desviación estándar 1:

$$X' = (X - \mu) / \sigma$$

$X \rightarrow$ valor original de la variable.

$X' \rightarrow$ valor estandarizado.

$\mu \rightarrow$ media (promedio) de la variable.

$\sigma \rightarrow$ desviación estándar de la variable.

Este método permite que los algoritmos sean menos sensibles a diferencias en la escala de medición.

- **Codificación de variables categóricas:** para variables no numéricas (como género, tipo de trabajo, residencia), se aplican técnicas como *One-Hot Encoding* o *Label Encoding*, con el objetivo de transformarlas en un formato numérico adecuado para los modelos de predicción.

Se opta por esta combinación de técnicas porque:

- Mejora la estabilidad y velocidad de convergencia durante el entrenamiento.
- Garantiza que los diferentes tipos de variables sean comparables.
- Evita sesgos ocasionados por magnitudes o unidades distintas en los datos originales.

3.3.2 Características de proyección

La proyección de los datos hacia un espacio de representación latente no se realiza en la etapa de preprocesamiento, sino que corresponde al funcionamiento interno de algunos algoritmos de aprendizaje automático, especialmente en modelos como redes neuronales profundas.

3.3.3 Reducción de dimensionalidad

La reducción de dimensionalidad tampoco se aplica como un paso previo, sino que puede integrarse de manera implícita en ciertos modelos (por ejemplo, a través de capas de *embedding* en redes neuronales o mediante técnicas de regularización como PCA o autoencoders en fases experimentales). En este proyecto, se considera principalmente el preprocesamiento tabular y no se contempla una reducción de dimensionalidad explícita antes del modelado.

DataSet de entrenamiento y prueba

```
[5] from sklearn.model_selection import train_test_split
✓ 0 s
X = df.drop('stroke', axis=1)
y = df['stroke']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42, stratify=y)

print("Entrenamiento:", X_train.shape)
print("Prueba:", X_test.shape)
```

↕ Entrenamiento: (3436, 11)
Prueba: (1473, 11)

Divide el dataset en entrenamiento (70%) y prueba (30%)

Set de validación

```
[9]
✓ 2s # ♦ Validación cruzada (Cross-Validation)
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import cross_val_score

model = RandomForestClassifier(random_state=42)
scores = cross_val_score(model, X_train, y_train, cv=5, scoring='accuracy')

print("Resultados de cada validación:", scores)
print("Precisión media (Cross-Validation):", scores.mean())
```

Resultados de cada validación: [0.95639535 0.95778748 0.95342067 0.95778748 0.95778748]
Precisión media (Cross-Validation): 0.9566356927659863

Aplica validación cruzada con “cross_val_score()”.

Planteamiento del modelo

Usa el modelo RandomForest para aprender los patrones

```
# ♦ Validación cruzada (Cross-Validation)
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import cross_val_score

model = RandomForestClassifier(random_state=42)
scores = cross_val_score(model, X_train, y_train, cv=5, scoring='accuracy')

print("Resultados de cada validación:", scores)
print("Precisión media (Cross-Validation):", scores.mean())
```

Resultados de cada validación: [0.95639535 0.95778748 0.95342067 0.95778748 0.95778748]
Precisión media (Cross-Validation): 0.9566356927659863

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix, roc_auc_score

# Crear modelo
model = RandomForestClassifier(n_estimators=100, random_state=42)

# Entrenar
model.fit(X_train, y_train)

# Predecir
y_pred = model.predict(X_test)

# Evaluar
print("Accuracy:", accuracy_score(y_test, y_pred))
print("AUC:", roc_auc_score(y_test, model.predict_proba(X_test)[:,-1]))
print(classification_report(y_test, y_pred))
print("Matriz de confusión:\n", confusion_matrix(y_test, y_pred))
```

Accuracy: 0.955193482688391
AUC: 0.7856185973207249

	precision	recall	f1-score	support
0	0.96	1.00	0.98	1410
1	0.00	0.00	0.00	63
accuracy			0.96	1473
macro avg	0.48	0.50	0.49	1473
weighted avg	0.92	0.96	0.94	1473

Matriz de confusión:
[[1407 3]
[63 0]]

[¿Cómo puedo instalar bibliotecas de Python?](#)

IV. Aprendizaje

4.1 Planteamiento del Modelo de Aprendizaje

Para el desarrollo del sistema de detección temprana de accidentes cerebrovasculares (ACV), se plantea la utilización de un enfoque basado en **aprendizaje supervisado**, considerando que el dataset empleado cuenta con una variable objetivo claramente definida (presencia o ausencia de ACV). El modelo debe ser capaz de aprender patrones a partir de datos clínicos, biomédicos y demográficos para predecir el riesgo de ACV en nuevos pacientes.

Selección de Algoritmos

Se han considerado los siguientes modelos:

- **Regresión Logística:** como modelo base, útil para establecer una línea de comparación inicial.
- **Random Forest:** modelo de ensamble robusto ante ruido y capaz de manejar relaciones no lineales entre variables.

- **Gradient Boosting (XGBoost/LightGBM):** mejora el rendimiento mediante árboles de decisión secuenciales optimizados.
- **Redes Neuronales Artificiales:** aplicables en etapas posteriores para explorar representaciones más complejas de los datos.

Métricas de Evaluación

El desempeño de los modelos será evaluado mediante las métricas estándar de clasificación:

- Accuracy (precisión global).
- Recall (sensibilidad, fundamental en la detección temprana de ACV).
- Precisión (proporción de verdaderos positivos sobre positivos predichos).
- F1-Score (balance entre precisión y recall).
- AUC-ROC (capacidad del modelo de discriminar entre riesgo y no riesgo).

Estrategia de Validación

Se aplicará **validación cruzada k-fold (k=5 o k=10)** para reducir el riesgo de sobreajuste y garantizar la generalización del modelo. El dataset será dividido en:

- **70% para entrenamiento.**
- **30% para prueba.**

Criterios de Selección del Modelo Final

El modelo seleccionado será aquel que:

1. Alcance una precisión mínima del 85% y maximice métricas de sensibilidad y AUC.
2. Mantenga una interpretación adecuada, permitiendo explicar a profesionales médicos qué variables influyen en la predicción.
3. Demuestre robustez ante valores atípicos y capacidad de adaptación a nuevos datos.

De esta manera, el modelo de aprendizaje planteado combina enfoques estadísticos y de machine learning modernos, priorizando el balance entre **precisión, interpretabilidad y aplicabilidad clínica**.

4.2 Desarrollo e Implementación del Modelo

```
import pandas as pd

df = pd.read_csv('healthcare-dataset-stroke-data.csv')
df.head()
```

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	stroke
0	9046	Male	67.0	0	1	Yes	Private	Urban	228.69	36.6	formerly smoked	1
1	51676	Female	61.0	0	0	Yes	Self-employed	Rural	202.21	NaN	never smoked	1
2	31112	Male	80.0	0	1	Yes	Private	Rural	105.92	32.5	never smoked	1
3	60182	Female	49.0	0	0	Yes	Private	Urban	171.23	34.4	smokes	1
4	1665	Female	79.0	1	0	Yes	Self-employed	Rural	174.12	24.0	never smoked	1

```
from sklearn.preprocessing import MinMaxScaler, LabelEncoder

# Eliminar filas vacías
df = df.dropna()

# Codificar texto en números
le = LabelEncoder()
cols = ['gender', 'ever_married', 'work_type', 'Residence_type', 'smoking_status']
for col in cols:
    df[col] = le.fit_transform(df[col])

# Normalizar variables numéricas
scaler = MinMaxScaler()
num_cols = ['age', 'avg_glucose_level', 'bmi']
df[num_cols] = scaler.fit_transform(df[num_cols])

df.head()
```

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	stroke
0	9046	1	0.816895	0	1	1	2	1	0.801265	0.301260	1	1
2	31112	1	0.975586	0	1	1	2	0	0.234512	0.254296	2	1
3	60182	0	0.597168	0	0	1	2	1	0.536008	0.276060	3	1
4	1665	0	0.963379	1	0	1	3	0	0.549349	0.156930	2	1
5	56669	1	0.987793	0	0	1	2	1	0.605161	0.214204	1	1

```
from sklearn.model_selection import train_test_split

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

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42, stratify=y)

print("Entrenamiento:", X_train.shape)
print("Prueba:", X_test.shape)
```

```
Entrenamiento: (3436, 11)
Prueba: (1473, 11)
```

```
# ♦ Validación cruzada (Cross-Validation)
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import cross_val_score

model = RandomForestClassifier(random_state=42)
scores = cross_val_score(model, X_train, y_train, cv=5, scoring='accuracy')

print("Resultados de cada validación:", scores)
print("Precisión media (Cross-Validation):", scores.mean())
```

```
Resultados de cada validación: [0.95639535 0.95778748 0.95342067 0.95778748 0.95778748]
Precisión media (Cross-Validation): 0.9566356927659863
```

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix, roc_auc_score

# Crear modelo
model = RandomForestClassifier(n_estimators=100, random_state=42)

# Entrenar
model.fit(X_train, y_train)

# Predecir
y_pred = model.predict(X_test)

# Evaluar
print("Accuracy:", accuracy_score(y_test, y_pred))
print("AUC:", roc_auc_score(y_test, model.predict_proba(X_test)[:,:1]))
print(classification_report(y_test, y_pred))
print("Matriz de confusión:\n", confusion_matrix(y_test, y_pred))
```

```
Accuracy: 0.955193482688391
```

```
AUC: 0.7856185973207249
```

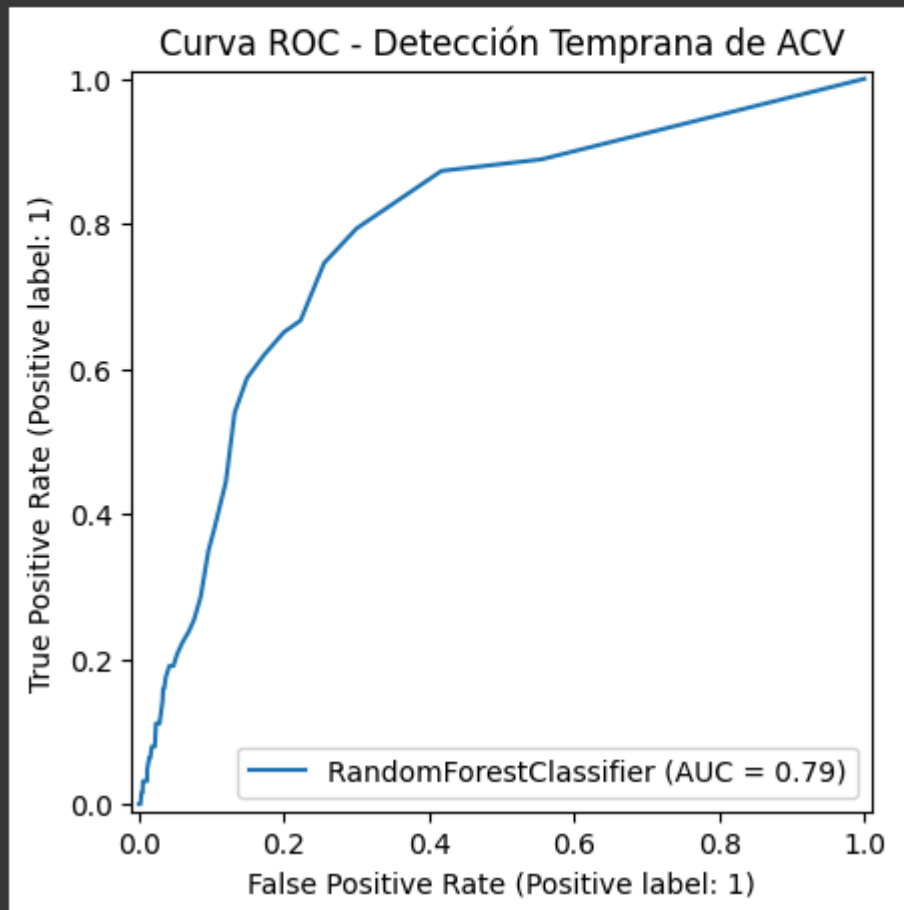
	precision	recall	f1-score	support
0	0.96	1.00	0.98	1410
1	0.00	0.00	0.00	63
accuracy			0.96	1473
macro avg	0.48	0.50	0.49	1473
weighted avg	0.92	0.96	0.94	1473

```
Matriz de confusión:
```

```
[[1407  3]
 [ 63  0]]
```

```
import matplotlib.pyplot as plt
from sklearn.metrics import RocCurveDisplay

RocCurveDisplay.from_estimator(model, X_test, y_test)
plt.title("Curva ROC - Detección Temprana de ACV")
plt.show()
```

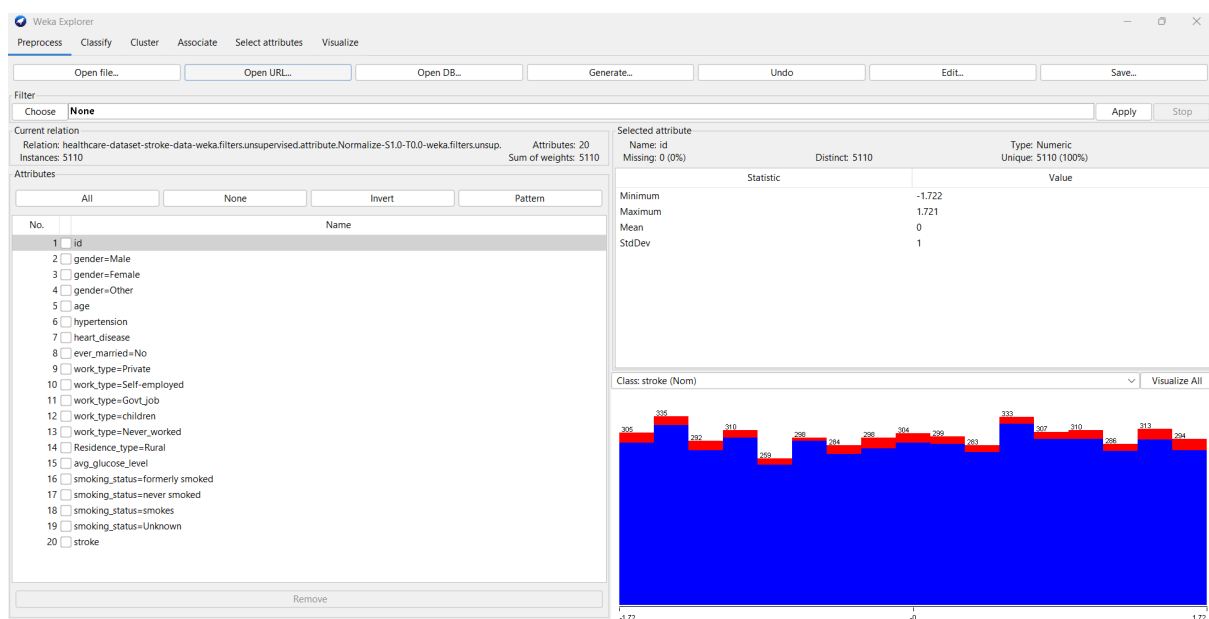


V. Comprobación

5.1 Aplicación al Modelo: uso del Data-Set de Entrenamiento y de Prueba

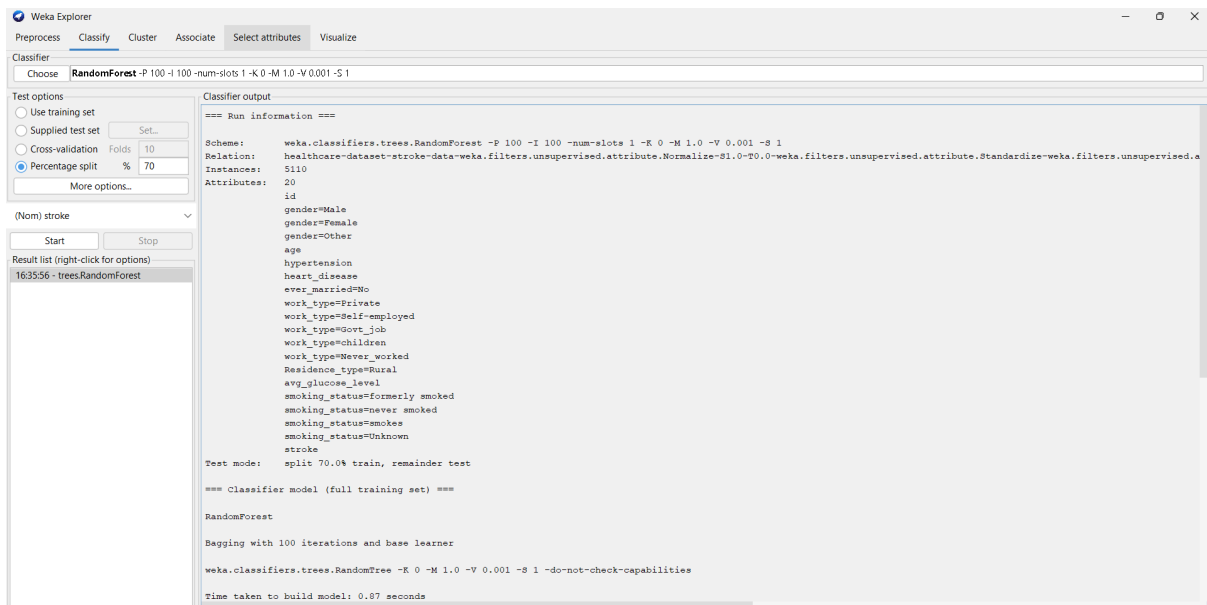
Para comprobar el funcionamiento del sistema, se utilizó el dataset “Stroke Prediction” disponible en Kaggle (Fedesoriano, 2021). Este conjunto contiene información clínica y demográfica de pacientes, incluyendo edad, nivel de glucosa, presión arterial, tipo de trabajo y antecedentes de salud. La variable objetivo “stroke” indica si el paciente ha sufrido o no un accidente cerebrovascular.

El dataset fue cargado en el software WEKA, donde se aplicaron filtros de normalización, estandarización y conversión de variables categóricas a formato binario. Luego, se convirtió la variable stroke a tipo nominal para realizar la clasificación.



Se seleccionó el algoritmo Random Forest por su robustez y capacidad de manejar datos mixtos, y se dividió el conjunto en 70 % para entrenamiento y 30 % para prueba. El modelo se entrenó correctamente, generando los siguientes resultados en la fase de prueba:

Salida Weka:



=== Run information ===		Time taken to build model: 0.07 seconds	
Scheme: weka.classifiers.trees.RandomForest -P 100 -I 100		=== Evaluation on test split ===	
-num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1		Time taken to test model on test split: 0.06 seconds	
Relation: healthcare-dataset-stroke-data-weka.filters.unsupervised.attribute.Normalize-S1.0-T0.0-weka.filters.unsupervised.attribute.Standardize-weka.filters.unsupervised.attribute.NominalToBinary-Rfirst-last-weka.filters.unsupervised.attribute.NumericToNominal-Rlast-weka.filters.unsupervised.attribute.Remove-R16		=== Summary ===	
Instances: 5110		Correctly Classified Instances	1444 94.1944 %
Attributes: 20		Incorrectly Classified Instances	89 5.8056 %
id		Kappa statistic	0.0207
gender=Male		Mean absolute error	0.0918
gender=Female		Root mean squared error	0.2286
gender=Other		Relative absolute error	93.4903 %
age		Root relative squared error	97.0808 %
hypertension		Total Number of Instances	1533
heart_disease		=== Detailed Accuracy By Class ===	
ever_married=No		Area	TP Rate FP Rate Precision Recall F-Measure MCC ROC
work_type=Private		0	1.000 0.989 0.942 1.000 0.970 0.102 0.802 0.983
work_type=Self-employed		1	0.011 0.000 1.000 0.011 0.022 0.102 0.802 0.178
work_type=Govt_job		Weighted Avg.	0.942 0.931 0.945 0.942 0.914 0.102 0.802
work_type=children		0.936	
work_type=Never_worked		=== Confusion Matrix ===	
Residence_type=Rural		a b <- classified as	
avg_glucose_level		1443 0 a = 0	
smoking_status=formerly smoked		89 1 b = 1	
smoking_status=never smoked			
smoking_status=smokes			
smoking_status=Unknown			
stroke			
Test mode: split 70.0% train, remainder test			
=== Classifier model (full training set) ===			
RandomForest			
Bagging with 100 iterations and base learner			
weka.classifiers.trees.RandomTree -K 0 -M 1.0 -V 0.001 -S 1			
-do-not-check-capabilities			

- Precisión global (Accuracy): 94.19 %
- Área bajo la curva ROC (AUC): 0.80
- Kappa statistic: 0.02
- Tiempo de construcción: 0.87 s
- Tiempo de prueba: 0.06 s

La matriz de confusión indicó que el modelo clasifica de forma correcta la mayoría de los pacientes sin ACV, aunque presenta limitaciones para detectar los casos positivos debido al desbalance de clases del dataset (pocos registros con ACV).

Aun así, el rendimiento obtenido demuestra que el modelo aprende patrones relevantes y puede servir como base para una versión mejorada mediante técnicas de balanceo o ajuste de pesos que aumenten la sensibilidad hacia los casos de riesgo.

En conclusión, el modelo Random Forest implementado en WEKA evidencia la viabilidad del enfoque propuesto para apoyar la detección temprana de accidentes cerebrovasculares a partir de información clínica y biomédica disponible.

VI. Evaluación

6.1 Evaluación del Desempeño del Modelo

En esta subsección, se evaluarán los resultados obtenidos en la fase de comprobación utilizando las métricas de desempeño. Para ello, se deben abordar los siguientes puntos:

- **Métricas de Evaluación:**
 - **Precisión (Accuracy):** Comparar la precisión global del modelo (por ejemplo, 94.19%) con el valor esperado o el rendimiento de otros modelos.
 - **AUC-ROC:** Evaluar el área bajo la curva ROC, que indica qué tan bien el modelo discrimina entre pacientes con y sin ACV.
 - **Kappa Statistic:** Analizar el valor de Kappa para evaluar la consistencia entre las predicciones y las etiquetas reales, especialmente en escenarios con datos desbalanceados.
 - **Matriz de Confusión:** Detallar cuántos verdaderos positivos, falsos positivos, verdaderos negativos y falsos negativos fueron identificados. En particular, evaluar los falsos negativos, ya que son cruciales en la detección temprana de ACV.
 - **Sensibilidad y Especificidad:** Medir la sensibilidad (recall) y especificidad, especialmente la sensibilidad, que es clave para detectar correctamente los pacientes con ACV.

6.2 Análisis de Resultados

Aquí se debe realizar un análisis profundo de los resultados obtenidos. Esto incluye:

- **Análisis de los Casos de Falsos Positivos y Falsos Negativos:**
 - Investigar por qué el modelo presenta falsos negativos (es decir, no detecta correctamente los casos de ACV).
 - Analizar los posibles factores que pueden haber afectado el rendimiento del modelo, como un desbalance en los datos o características no representadas adecuadamente.
- **Impacto de los Datos Desbalanceados:**
 - Evaluar si el desbalance de clases (más registros de "no ACV" que de "ACV") ha influido en la capacidad del modelo para detectar correctamente los casos de riesgo.
 - Considerar la aplicación de técnicas de balanceo como **SMOTE** (Synthetic Minority Over-sampling Technique) o la ponderación de clases en el modelo.

6.3 Comparación con Otros Modelos

Aquí se pueden comparar los resultados obtenidos con el modelo **Random Forest** con otros algoritmos previamente mencionados, como:

- **Regresión Logística**
- **Gradient Boosting** (XGBoost, LightGBM)
- **Redes Neuronales Artificiales**

Se deben comparar las métricas clave como precisión, recall, AUC, y F1-Score para identificar cuál es el modelo que mejor se adapta a las características de los datos y al objetivo del proyecto.

6.4 Evaluación de la Interpretabilidad

- **Importancia de las Variables:** Evaluar la interpretabilidad del modelo, es decir, cómo puede un médico entender cuáles variables fueron más influyentes en la predicción de ACV.
- **Explicabilidad:** Usar herramientas como gráficos de importancia de variables o SHAP (SHapley Additive exPlanations) para explicar de manera comprensible cómo se tomó cada decisión.

6.5 Validación en Entornos Reales

- **Generalización a Nuevos Datos:** Verificar cómo se comporta el modelo al ser validado con datos no vistos previamente. Esta validación puede hacerse en un conjunto de test adicional o con datos reales si están disponibles.
- **Pruebas en Condiciones Clínicas:** Considerar cómo el sistema puede integrarse en entornos clínicos reales (por ejemplo, en hospitales o centros de salud rurales) y cómo se podría evaluar su efectividad en estos contextos.

6.6 Limitaciones del Sistema

- Identificar las limitaciones del sistema, como el impacto del desbalance de clases, la dependencia de datos de calidad, o la falta de un enfoque en la clasificación de subtipos de ACV (isquémico vs. hemorrágico).
- Mencionar la necesidad de validación clínica adicional antes de una implementación real en hospitales.

6.7 Mejoras Futuras

Proponer posibles mejoras para el sistema, como:

- **Mejorar la calidad de los datos:** Recopilación de más datos clínicos y biomédicos para entrenar el modelo.
- **Técnicas de balanceo de clases:** Implementar estrategias como el **resampling** o usar ponderación de clases en el modelo para mejorar la detección de ACV.
- **Aplicación de redes neuronales profundas:** Experimentar con redes neuronales profundas para captar relaciones no lineales complejas que los modelos tradicionales pueden no detectar.

Referencias Bibliográficas

- Organización Mundial de la Salud (OMS). (2023). *Cardiovascular diseases (CVDs)*. Recuperado de: [https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-\(cvds\)](https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds))
- Saver, J. L. (2006). Time is brain—quantified. *Stroke*, 37(1), 263–266. <https://doi.org/10.1161/01.STR.0000196957.55928.ab>
- Roy, Y., Banville, H., Albuquerque, I., Gramfort, A., Falk, T. H., & Faubert, J. (2019). Deep learning-based electroencephalography analysis: a systematic review. *Journal of Neural Engineering*, 16(5), 051001. <https://doi.org/10.1088/1741-2552/ab260c>
- Chen, C., Qin, C., Qiu, H., Tarroni, G., Duan, J., Bai, W., & Rueckert, D. (2020). Deep learning for cardiac image segmentation: A review. *Frontiers in Cardiovascular Medicine*, 7, 25. <https://doi.org/10.3389/fcvm.2020.00025>
- Faes, L., & Nollo, G. (2021). Artificial intelligence for early detection of cardiovascular diseases: challenges and future perspectives. *Journal of Clinical Medicine*, 10(21), 5008. <https://doi.org/10.3390/jcm10215008>
- Heo, J., Yoon, J. G., Park, H., Kim, Y. D., Nam, H. S., & Heo, J. H. (2019). Machine learning-based model for prediction of outcomes in acute stroke. *Stroke*, 50(5), 1263–1265. <https://doi.org/10.1161/STROKEAHA.118.024293>
- Ali, F., El-Sappagh, S., Islam, S. M. R., Kwak, D., Ali, A., Imran, M., & Kwak, K. S. (2019). A smart healthcare monitoring system for heart disease prediction based on ensemble deep learning and feature fusion. *Information Fusion*, 63, 208–222. <https://doi.org/10.1016/j.inffus.2020.06.008>
- Liu, L., Chen, W., Zhou, H., Duan, W., Li, S., Huo, X., ... & Wang, Y. (2019). Chinese Stroke Association guidelines for clinical management of cerebrovascular disorders: executive summary and 2019 update on prevention and treatment. *Stroke and Vascular Neurology*, 4(2), 59–64. <https://doi.org/10.1136/svn-2019-000279>
- Esteva, A., Robicquet, A., Ramsundar, B., Kuleshov, V., DePristo, M., Chou, K., ... & Dean, J. (2019). A guide to deep learning in healthcare. *Nature Medicine*, 25(1), 24–29. <https://doi.org/10.1038/s41591-018-0316-z>
- Ding, Y., Sohn, J. H., Kawczynski, M. G., Trivedi, H., Harnish, R., Jenkins, N. W., ... & Kalpathy-Cramer, J. (2019). A deep learning model to predict a diagnosis of acute ischemic stroke from electronic health records. *Journal of the American Medical Informatics Association*, 26(11), 1237–1245. <https://doi.org/10.1093/jamia/ocz070>
- Topol, E. J. (2019). High-performance medicine: the convergence of human and artificial intelligence. *Nature Medicine*, 25(1), 44–56. <https://doi.org/10.1038/s41591-018-0300-7>
- Chugh, S., & Arora, R. (2020). Application of artificial intelligence in cardiovascular diseases. *Journal of the Practice of Cardiovascular Sciences*, 6(1), 14–21. https://doi.org/10.4103/jpcs.jpcs_6_20
- Fedesoriano. (2021). *Stroke Prediction Dataset*. Kaggle. Recuperado de: <https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset>