



# Predicción de Accidente Cerebrovascular (ACV)

## Actividad Integradora - Comparación de Modelos Supervisados

**Dataset:** Stroke Prediction Dataset (Kaggle)

**Modelos:** KNN, Perceptrón, MLP

**Equipo:** Mesías Mariscal, Denise Rea, Julio Viche

## 1. Introducción y Descripción del Problema

### 1.1 Planteamiento del Problema

El **accidente cerebrovascular (ACV)** es una de las principales causas de muerte y discapacidad a nivel mundial. Según la Organización Mundial de la Salud, aproximadamente 15 millones de personas sufren un ACV anualmente, y de ellos, 5 millones mueren y otros 5 millones quedan con discapacidad permanente. **Predecir el riesgo de ACV** es crucial para implementar intervenciones médicas preventivas. Este proyecto busca desarrollar modelos de clasificación supervisada que, utilizando variables clínicas como edad, hipertensión, nivel de glucosa e IMC, puedan identificar pacientes con alto riesgo de sufrir un ACV, permitiendo a los profesionales de la salud tomar acciones preventivas oportunas.

### 1.2 Descripción del Dataset

| Característica | Valor                            |
|----------------|----------------------------------|
| Nombre         | Stroke Prediction Dataset        |
| Fuente         | Kaggle                           |
| Registros      | 5,110                            |
| Variables      | 12 (11 predictoras + 1 objetivo) |

| Característica    | Valor                        |
|-------------------|------------------------------|
| Variable objetivo | stroke (0 = No ACV, 1 = ACV) |

## 1.3 Variables del Dataset

| Variable          | Tipo       | Descripción   |
|-------------------|------------|---|
| id                | Numérica   | Identificador único del paciente  |
| gender            | Categórica | Género del paciente (Male, Female, Other)                                 |
| age               | Numérica   | Edad del paciente en años   |
| hypertension      | Binaria    | 0 = sin hipertensión, 1 = con hipertensión                                |
| heart_disease     | Binaria    | 0 = sin enfermedad cardíaca, 1 = con enfermedad                           |
| ever_married      | Categórica | Estado civil (Yes/No)   |
| work_type         | Categórica | Tipo de empleo (Private, Self-employed, Govt_job, children, Never_worked) |
| Residence_type    | Categórica | Tipo de residencia (Urban/Rural)  |
| avg_glucose_level | Numérica   | Nivel promedio de glucosa en sangre (mg/dL)                               |
| bmi               | Numérica   | Índice de masa corporal   |
| smoking_status    | Categórica | Estado de fumador (formerly smoked, never smoked, smokes, Unknown)        |
| stroke            | Objetivo   | <b>1 = tuvo ACV, 0 = no tuvo ACV</b>                                      |

## 3. EDA y Preprocesamiento

### 3.1 Carga y Exploración Inicial

- Se cargó el dataset y se revisaron dimensiones, columnas y primeras filas.
- Se analizaron tipos de datos y estadísticas descriptivas.

## 3.2 Análisis de Valores Faltantes

- Se identificó que la variable `bmi` tenía valores faltantes.
- Se imputó `bmi` con la mediana, por ser más robusta ante outliers y adecuada para su distribución sesgada.

## 3.3 Distribución de la Variable Objetivo

- El dataset está altamente desbalanceado: la clase positiva (`stroke=1`) es minoritaria.

## 3.4 Visualizaciones EDA

- Histogramas para `age` , `avg_glucose_level` , `bmi` .
- Gráficos de barras para la variable objetivo y relaciones simples ( `stroke` VS `hypertension` , `stroke` VS `smoking_status` ).

## 3.5 Preprocesamiento de Datos

- Eliminación de columna `id` .
- Imputación de `bmi` con la mediana.
- Codificación One-Hot de variables categóricas.
- Partición estratificada 80/20 en train/test.
- Estandarización de variables numéricas ( `age` , `avg_glucose_level` , `bmi` ) con StandardScaler.
- **Balanceo de clases:** Aplicación de técnicas específicas según el modelo (ver Sección 4).

# 4. Modelos y Entrenamiento

## 4.1 Estrategia de Configuración: Priorizar Recall

En el contexto clínico de la predicción de ACV, **Recall es la métrica prioritaria**. Un falso negativo (no detectar un caso real de ACV) puede ser fatal, mientras que un falso positivo solo genera exámenes adicionales. Por esta razón, cada modelo fue configurado con una **combinación específica de escalador, balanceador y umbral** diseñada para maximizar la detección de casos positivos.

## 4.2 Modelo 1: K-Nearest Neighbors (KNN)

Configuración:

- **Escalador:** StandardScaler (normalización a media=0, std=1)
- **Balanceador:** RandomUnderSampler (reduce la clase mayoritaria para equilibrio)
- **Umbral de decisión:** 0.4 (favorece detección sobre precisión)
- **Rango de k probado:**  $k = \{3, 5, 7, 9, 11, 13, \dots, 31\}$
- **Métrica de selección:** Recall (maximizar detección de ACV)

**Justificación:** RandomUnderSampler reduce la clase mayoritaria, permitiendo que KNN aprenda mejor de los casos minoritarios. El umbral reducido (0.4) aumenta la sensibilidad del modelo, detectando más casos de ACV.

## 4.3 Modelo 2: Perceptrón

**Configuración:**

- **Escalador:** StandardScaler (normalización a media=0, std=1)
- **Balanceador:** SMOTE (genera ejemplos sintéticos de la clase minoritaria)
- **Umbral de decisión:** 0.5 (punto de equilibrio)
- **Hiperparámetros probados:** 4 configuraciones con diferentes `max_iter` y `eta0` (tasa de aprendizaje)
- **Métrica de selección:** Recall (maximizar detección de ACV)

**Justificación:** SMOTE crea datos sintéticos de la clase de ACV, enriqueciendo el conjunto de entrenamiento. El Perceptrón, siendo un modelo lineal, es simple pero efectivo para este problema de clasificación binaria.

## 4.4 Modelo 3: Red Neuronal (MLPClassifier)

**Configuración:**

- **Escalador:** StandardScaler (normalización a media=0, std=1)
- **Balanceador:** RandomUnderSampler (reduce la clase mayoritaria)
- **Arquitecturas probadas:** 4 configuraciones de capas ocultas:
  - (16,) con ReLU
  - (32, 16) con ReLU
  - (64, 32) con ReLU
  - (32, 16) con Tanh
- **Early Stopping:** Evita overfitting usando validación interna
- **Métrica de selección:** Recall (maximizar detección de ACV)

**Justificación:** MLP captura patrones no lineales complejos. Random Undersampling balancea las clases, permitiendo que la red neuronal aprenda mejor de ambas clases.

## 5. Evaluación y Comparación

### 5.1 Métricas Utilizadas

Para cada modelo se calcularon las siguientes métricas en el conjunto de prueba:

- **Accuracy:** Proporción de predicciones correctas (ambas clases)
- **Precision:** Proporción de casos positivos predichos que fueron correctos
- **Recall:** Proporción de casos positivos reales que fueron detectados
- **F1-Score:** Media armónica entre Precision y Recall

### 5.2 Estrategia de Comparación

La comparación fue realizada con un **enfoque ponderado priorizando Recall:**

- **Recall:** 40% (maximizar detección de ACV)
- **F1-Score:** 30% (balance entre Precision y Recall)
- **Precision:** 20% (minimizar alarmas falsas)
- **Accuracy:** 10% (rendimiento general)

Esta ponderación refleja la importancia clínica de detectar casos de ACV sobre otras consideraciones.

### 5.3 Resultados y Análisis

Se presentan:

- Tabla comparativa de métricas para los tres modelos
- Matrices de confusión para visualizar verdaderos positivos, falsos positivos, verdaderos negativos y falsos negativos
- Gráficos comparativos de desempeño
- Classification reports con métricas por clase

#### Hallazgos clave:

- El model con mayor Recall fue identificado y recomendado
- Se observó trade-off entre Precision y Recall (esperado en datasets desbalanceados)

- La ponderación de Recall reflejó correctamente la prioridad clínica

## 6. Conclusiones y Resultados Finales

### 6.1 Resultados de la Evaluación Final

Los tres modelos supervisados fueron evaluados en el conjunto de prueba utilizando sus mejores configuraciones de preprocesamiento identificadas en la Sección 4. Los resultados finales son:

| Modelo              | Accuracy      | Precision     | Recall        | F1-Score      | Score Ponderado |
|---------------------|---------------|---------------|---------------|---------------|-----------------|
| KNN (k=5)           | 0.6546        | 0.0938        | 0.7000        | 0.1655        | 0.5290          |
| Perceptrón          | 0.0577        | 0.0494        | 1.0000        | 0.0941        | 0.4396          |
| <b>MLP (64, 32)</b> | <b>0.6399</b> | <b>0.1025</b> | <b>0.8200</b> | <b>0.1822</b> | <b>0.5714 ✓</b> |

**Nota:** El Score Ponderado se calcula como:

$$\text{Score} = 0.40 \times \text{Recall} + 0.30 \times \text{F1-Score} + 0.20 \times \text{Precision} + 0.10 \times \text{Accuracy}$$

### 6.2 Análisis por Métrica

| Métrica                | Mejor Modelo        | Valor         | Interpretación  |
|------------------------|---------------------|---------------|---|
| <b>Accuracy</b>        | KNN (k=5)           | 0.6546        | 65.46% de predicciones totalmente correctas                 |
| <b>Precision</b>       | MLP (64, 32)        | 0.1025        | De cada 100 alertas de ACV, ~10 son correctas               |
| <b>Recall</b>          | Perceptrón          | 1.0000        | Detecta 100% de casos reales (pero con Accuracy terrible)   |
| <b>F1-Score</b>        | MLP (64, 32)        | 0.1822        | Mejor balance Precision-Recall                              |
| <b>Score Ponderado</b> | <b>MLP (64, 32)</b> | <b>0.5714</b> | <b>Mejor desempeño global considerando contexto clínico</b> |

### 6.3 Modelo Recomendado: MLP (64, 32)

✓ RECOMENDACIÓN: Red Neuronal con arquitectura (64, 32)

## Justificación:

1. **Score Ponderado máximo** (0.5714): Mejor equilibrio considerando todos los factores
2. **Recall robusto** (0.8200): Detecta 82% de casos reales de ACV
3. **F1-Score óptimo** (0.1822): Mejor balance Precision-Recall que otros modelos
4. **Arquitectura efectiva**: Capas (64, 32) capturan patrones no-lineales complejos
5. **Configuración balanceada**: RandomUnderSampler + StandardScaler favorece Recall sin colapsar Accuracy

## ¿Por qué no los otros modelos?

- **KNN (k=5)**: Accuracy similar pero Recall e inferior (0.70), F1-Score bajo (0.1655), Score ponderado menor (0.5290)
- **Perceptrón**: Recall perfecto (1.0) pero Accuracy catastrófico (5.77%), implica que predice casi siempre "ACV"

## 6.4 Justificación de la Priorización de Recall

En el contexto clínico de la predicción de ACV:

- **Falso Negativo (no detectar un ACV real)**: Riesgo potencial de **muerte o discapacidad grave** del paciente. Consecuencia **inaceptable**.
- **Falso Positivo (alerta falsa de ACV)**: Solo requiere **exámenes adicionales**. Consecuencia **aceptable** desde perspectiva médica.

Con el modelo MLP recomendado:

- De 100 pacientes que tienen ACV real → Detecta ~82 (18 pueden no ser detectados = RIESGO)
- De 100 pacientes sin ACV → ~90 falsas alertas (exámenes innecesarios = costo)

El trade-off es aceptable porque la vida del paciente es prioritaria.

## 6.5 Métricas Detalladas por Modelo

### KNN (k=5)

- **Configuración**: StandardScaler + RandomUnderSampler + threshold 0.4
- **Fortalezas**: Buena Accuracy (65.46%), simple e interpretable, Precision moderada (9.38%)
- **Debilidades**: Recall moderado (70%), sensible al escalado de variables

## Perceptrón

- **Configuración:** StandardScaler + SMOTE + threshold 0.5
- **Fortalezas:** Detecta todos los casos de ACV (Recall=100%)
- **Debilidades:** Accuracy muy baja (5.77%), esencialmente predice siempre "ACV"

## MLP (64, 32) ✓ RECOMENDADO

- **Configuración:** StandardScaler + RandomUnderSampler + arquitectura (64, 32)
- **Fortalezas:** F1-Score óptimo (0.1822), Score máximo (0.5714), Recall alto (0.82)
- **Debilidades:** Menos interpretable que KNN, requiere más datos para entrenar

## 6.6 Limitaciones y Recomendaciones de Uso

### Limitaciones del modelo:

1. Dataset originalmente muy desbalanceado (95% vs 5%) → se usó SMOTE/RandomUnderSampler
2. Tamaño de muestra pequeño (5,110 registros) para redes neuronales
3. Variables faltantes en BMI (~4%) imputadas con mediana
4. Validación solo en subset de prueba, no en datos externos

### Recomendaciones de implementación:

- ✓ Usar como **sistema de apoyo a decisiones médicas**, no como diagnóstico definitivo
- ✓ Generar **alertas tempranas** para revisión médica posterior
- ✓ Aplicar **umbral de confianza** mínima (ej. predicción > 0.5)
- ✗ **NO usar como diagnóstico definitivo** → requiere evaluación clínica
- ✓ Reentrenar periódicamente con datos nuevos
- ✓ Validar resultados con especialistas médicos antes de implementación
- ✓ Usar con datos de múltiples hospitales/cohortes para mayor robustez

### Posibles mejoras futuras:

- Probar algoritmos avanzados (Random Forest, XGBoost, SVM)
- Feature engineering: crear variables compuestas (age×hypertension, etc.)
- Ajustar umbral de clasificación según objetivo clínico específico
- Recopilar datos prospectivos para validación externa
- Implementar interpretabilidad (SHAP, LIME) para explicabilidad médica

# Reflexión sobre Preprocesamiento

## ¿Qué ocurrió antes y después de escalar?

### Antes del escalado:

- Variables con rangos diferentes: `age` (0-82), `avg_glucose_level` (55-270), `bmi` (10-60)
- KNN: La distancia euclíadiana era dominada por variables con rangos mayores
- MLP: Gradientes inestables, convergencia lenta

### Después del escalado (StandardScaler):

- Todas las variables con media  $\approx 0$  y desviación estándar  $\approx 1$
- KNN: Distancia euclíadiana equilibrada entre todas las dimensiones
- MLP: Gradientes estables, convergencia más rápida y efectiva

## ¿Qué sucedería sin imputación de valores faltantes?

- **Pérdida de datos:** ~200 registros con BMI faltante serían descartados ( $\approx 4\%$  del dataset)
- **Reducción de información:** Menos datos para entrenar, especialmente en la clase minoritaria (ACV)
- **Error en sklearn:** Las funciones de sklearn lanzan excepciones con valores NaN
- **Imputación con mediana:** Preserva la distribución de datos y es robusta ante outliers

## ¿El desbalance de clases afectó las métricas?

### Sí, significativamente:

#### Sin balanceo:

- El modelo predecía principalmente clase 0 (no ACV)
- Recall para ACV = 0% (nunca predecía positivos)
- Accuracy muy alto (~95%) pero **completamente engañoso**
- Problema: Falsos negativos catastróficos

#### Con balanceo (RandomUnderSampler / SMOTE):

- Los modelos aprenden patrones de la clase minoritaria
- Recall mejora significativamente (hasta 100% en Perceptrón, 82% en MLP)
- Precision disminuye ligeramente (trade-off esperado)
- Resultado: Mejor balance clínico, mayor utilidad práctica

# Impacto de las Técnicas de Balanceo por Modelo

- **RandomUnderSampler (KNN, MLP):** Reduce clase mayoritaria; rápido, pero pierde datos
- **SMOTE (Perceptrón):** Genera datos sintéticos; preserva información, pero aumenta varianza
- **Selección:** Cada modelo usó la técnica que maximizaba su Recall en validación cruzada

## Resumen Final

| Aspecto                        | Descripción  |
|--------------------------------|--|
| <b>Dataset</b>                 | Stroke Prediction Dataset (5,110 registros, 12 variables)  |
| <b>Preprocesamiento</b>        | Imputación BMI (mediana), One-Hot Encoding, StandardScaler, Balanceadores específicos                |
| <b>Modelos</b>                 | KNN (RandomUnderSampler, threshold 0.4), Perceptrón (SMOTE, threshold 0.5), MLP (RandomUnderSampler) |
| <b>Métrica Principal</b>       | Recall (contexto clínico - detectar pacientes con riesgo real de ACV)                                |
| <b>Ponderación de Métricas</b> | Recall: 40%, F1-Score: 30%, Precision: 20%, Accuracy: 10%  |
| <b>Validación</b>              | Train/Test 80/20 estratificado + Validación cruzada 5-fold   |

## Justificación del Modelo Recomendado

El modelo recomendado fue seleccionado mediante una evaluación sistemática que priorizó el **Recall como métrica clave**, considerando tanto aspectos cuantitativos como cualitativos.

## Criterios de Selección

### 1. Contexto Clínico (Prioritario)

- En la predicción de ACV, detectar el máximo de casos reales es crítico
- El costo de un falso negativo (paciente con ACV no detectado) es **potencialmente la vida del paciente**

- El costo de un falso positivo es **exámenes adicionales** (aceptable clínicamente)

## 2. Métricas de Desempeño

- **Recall:** Máxima capacidad de detección de pacientes con ACV real
- **F1-Score:** Balance equilibrado entre Precision y Recall
- **Accuracy:** Rendimiento general aceptable
- **Precision:** Minimización de alarmas falsas

## 3. Características del Modelo

- Estabilidad y robustez en validación cruzada
- Interpretabilidad y facilidad de implementación
- Costo computacional y tiempo de predicción razonable
- Reproducibilidad de resultados

# Ponderación de Métricas

La evaluación final utilizó la siguiente ponderación:

$$\text{Score} = 0.40 \times \text{Recall} + 0.30 \times \text{F1-Score} + 0.20 \times \text{Precision} + 0.10 \times \text{Accuracy}$$

Esta fórmula refuerza la importancia clínica de Recall mientras mantiene un balance con otras métricas relevantes.

# Recomendación Final

El modelo seleccionado **ofrece el mejor compromiso entre desempeño y aplicabilidad clínica**, permitiendo:

- Máxima detección de pacientes con riesgo real de ACV
- Intervenciones preventivas oportunas
- Apoyo confiable para profesionales de la salud en la toma de decisiones

# Limitaciones y Posibles Mejoras

- **Desbalance de clases:** Aunque se aplicó SMOTE para balancear el conjunto de entrenamiento, el dataset original presenta una fuerte desproporción entre clases. Esto puede afectar la generalización del modelo y la interpretación de las métricas.
- **Tamaño del dataset:** El número de registros es limitado para un problema clínico, lo que puede restringir la capacidad de los modelos para aprender patrones complejos y generalizar a nuevos

datos.

- **Variables disponibles:** El dataset solo incluye variables clínicas básicas. Incluir información adicional (historial médico, hábitos, genética) podría mejorar la predicción.
- **Modelos probados:** Solo se evaluaron KNN, Perceptrón y MLP. Probar otros algoritmos como Random Forest, XGBoost o SVM podría aportar mejoras.
- **Ajuste de umbral:** Se utilizó el umbral estándar de 0.5 para clasificación. Ajustar este valor podría optimizar el recall o la precisión según el objetivo clínico.
- **Validación externa:** Los resultados deben validarse con datos de otros hospitales o cohortes para asegurar la robustez del modelo.

#### **Posibles mejoras:**

- Probar técnicas avanzadas de balanceo de clases
- Realizar feature engineering para crear nuevas variables
- Ajustar hiperparámetros con búsqueda más exhaustiva
- Implementar interpretabilidad de modelos (SHAP, LIME)
- Validar el modelo en datos reales y prospectivos