Aplicación de Técnicas de Aprendizaje Supervisado en Enfermedades Cardiovasculares

Nicolás Seivane

UNAHUR

4 de Diciembre de 2025

Introducción

- Las enfermedades cardiovasculares son la causa número uno de muerte globalmente, con un estimado de 17.9 millones de vidas cada año, aproximadamente el 31 % de todas las muertes globales.

Introducción

- Las enfermedades cardiovasculares son la causa número uno de muerte globalmente, con un estimado de 17.9 millones de vidas cada año, aproximadamente el 31 % de todas las muertes globales.
- El objetivo general de este trabajo es comparar el rendimiento de diversas técnicas de Aprendizaje Automático Supervisado con el fin de recomendar aquella que presente el mejor desempeño al aplicarse sobre un conjunto de datos cardiológicos, con el fin de realizar predicciones de si un paciente tiene altas probabilidades de tener insuficiencia cardíaca.

Motivación

- El proceso de diagnóstico médico puede ser extenso, incluso contando con la mejor disposición del personal de salud, ya que con frecuencia requiere la recopilación y análisis de datos provenientes de distintos estudios.

Motivación

- El proceso de diagnóstico médico puede ser extenso, incluso contando con la mejor disposición del personal de salud, ya que con frecuencia requiere la recopilación y análisis de datos provenientes de distintos estudios.
- El propósito de este trabajo es contribuir a agilizar dicho proceso, identificando técnicas que puedan ser utilizadas por los profesionales médicos como herramientas complementarias para realizar diagnósticos.

Aprendizaje Automático en Cardiología

El Aprendizaje Automático se ha vuelto central en la investigación cardiovascular. Isaksen et al. [4] destacan desafíos como fuga de datos y desbalance de clases. Kumar y Kumar [5] revisan técnicas no invasivas basadas en ML para diagnóstico cardíaco.

Las variables cardiológicas suelen agruparse en:

- **Parámetros clínicos estructurados** (demografía, laboratorio).
- **Señales cardíacas** (ECG, PCG).
- **Imágenes médicas** (ECG, CMR, CCT, SPECT).

Modelos y Herramientas Diagnósticas

Los modelos más usados en cardiología incluyen:

- **SVM**: fuerte rendimiento en análisis de ECG y datos clínicos.
- **k-NN**: útil en detección de arritmias y riesgo.
- **Naïve Bayes y árboles de decisión**: empleados en datos tabulares y señales procesadas.

El diagnóstico clínico tradicional continúa basado en:

- Electrocardiograma (ECG)
- Prueba de esfuerzo
- Ecocardiograma
- Análisis de laboratorio (troponinas, colesterol, glucosa)

Estas herramientas siguen siendo el estándar y los modelos computacionales buscan complementarlas.

Metodología General

- Dataset clínico con múltiples variables relacionadas al estado cardíaco.
- Proceso seguido:
  1. Preprocesamiento y limpieza:

Metodología General

- Dataset clínico con múltiples variables relacionadas al estado cardíaco.
- Proceso seguido:
  1. Preprocesamiento y limpieza:
     - Se eliminaron los registros con datos faltantes.

Metodología General

- Dataset clínico con múltiples variables relacionadas al estado cardíaco.
- Proceso seguido:
  1. Preprocesamiento y limpieza:
     - Se eliminaron los registros con datos faltantes.
     - Se eliminaron los datos con valores anormales.

Metodología General

- Dataset clínico con múltiples variables relacionadas al estado cardíaco.
- Proceso seguido:
  1. Preprocesamiento y limpieza:
     - Se eliminaron los registros con datos faltantes.
     - Se eliminaron los datos con valores anormales.
     - Se estandarizaron los datos.

Metodología General

- Dataset clínico con múltiples variables relacionadas al estado cardíaco.
- Proceso seguido:
  1. Preprocesamiento y limpieza:
     - Se eliminaron los registros con datos faltantes.
     - Se eliminaron los datos con valores anormales.
     - Se estandarizaron los datos.
     - Se codificaron los datos categóricos.

Metodología General

- Dataset clínico con múltiples variables relacionadas al estado cardíaco.
- Proceso seguido:
  1. Preprocesamiento y limpieza:
     - Se eliminaron los registros con datos faltantes.
     - Se eliminaron los datos con valores anormales.
     - Se estandarizaron los datos.
     - Se codificaron los datos categóricos.
  2. Selección de características.

Metodología General

- Dataset clínico con múltiples variables relacionadas al estado cardíaco.
- Proceso seguido:
  1. Preprocesamiento y limpieza:
     - Se eliminaron los registros con datos faltantes.
     - Se eliminaron los datos con valores anormales.
     - Se estandarizaron los datos.
     - Se codificaron los datos categóricos.
  2. Selección de características.
  3. Entrenamiento de modelos supervisados.

Metodología General

- Dataset clínico con múltiples variables relacionadas al estado cardíaco.
- Proceso seguido:
  1. Preprocesamiento y limpieza:
     - Se eliminaron los registros con datos faltantes.
     - Se eliminaron los datos con valores anormales.
     - Se estandarizaron los datos.
     - Se codificaron los datos categóricos.
  2. Selección de características.
  3. Entrenamiento de modelos supervisados.
  4. Evaluación y comparación usando métricas estándar.

Metodología General

- Dataset clínico con múltiples variables relacionadas al estado cardíaco.
- Proceso seguido:
  1. Preprocesamiento y limpieza:
     - Se eliminaron los registros con datos faltantes.
     - Se eliminaron los datos con valores anormales.
     - Se estandarizaron los datos.
     - Se codificaron los datos categóricos.
  2. Selección de características.
  3. Entrenamiento de modelos supervisados.
  4. Evaluación y comparación usando métricas estándar.

Origen del Dataset

El dataset *HeartFailure* [2] se construyó combinando cinco conjuntos de datos cardiovasculares, unificados en 11 atributos comunes. Esto permite obtener uno de los conjuntos más grandes usados en investigación clínica.

Datasets incluidos:

- Cleveland (303)
- Hungarian (294)
- Switzerland (123)
- Long Beach VA (200)
- Stalog (Heart) (270)

Dataset Binario: Características Generales

Cuadro: Cantidad de registros utilizados.

Cantidad de registros	**918**
Cantidad de atributos	**11**
Atributos Categóricos	**5**
Atributos Numéricos	**6**

Dataset Binario: Características Generales

Cuadro: Cantidad de registros utilizados.

Cantidad de registros	**918**
Cantidad de atributos	**11**
Atributos Categóricos	**5**
Atributos Numéricos	**6**

Tipos de Atributos (Resumen)

Los atributos incluyen datos demográficos, clínicos, señales del ECG y medidas relacionadas al esfuerzo físico.

Atributos del Dataset Binario

Tipo de atributo del conjunto binario.

Atributo	Tipo de dato	¿Está codificado?	Unidad
Age	Numérico (`int`)	No	Años
Sex	Categórico (`string`)	No	-
ChestPainType	Categórico (`string`)	No	-
RestingBP	Numérico (`int`)	No	mm Hg
Cholesterol	Numérico (`int`)	No	mm/dl
FastingBS	Numérico (`int`)	Sí	mg/dl
RestingECG	Categórico (`string`)	No	-
MaxHR	Numérico (`int`)	No	-
ExerciseAngina	Categórico (`string`)	No	-
Oldpeak	Numérico (`float`)	No	ST en depresión
ST_Slope	Categórico (`string`)	No	-
HeartDisease	Numérico (`int`)	Sí	-

Descripción de Atributos (I)

Demográficos y Clínica General

- **Age:** Los pacientes tienen una media de edad de 53 años, con una edad máxima de 77 y de edad mínima de 28, con proporciones de edad bastante bien distribuidas, siendo la menor de 0.11 % para algunas edades y la mayor de 4.14 % para otras edades.
- **Sex:** Refiere al sexo de los pacientes; hay una distribución de sexo del 78.98 % masculinos y el 21.02 % femeninos.
- **ChestPainType::** Tipo del dolor en el pecho, con exactitud, dolor torácico causado por isquemia cardíaca.Tiene una distribución 18.85 % de angina atípica, luego un 22.11 % de dolor no anginoso, un 54.03 % asintomático, y un 5.01 % de dolor de pecho anginoso típico.

Descripción de Atributos (I)

Variables Clínicas

- **RestingBP:** Se está describiendo la presión sanguínea en reposo, tiene un valor medio de 133.02, con un valor máximo de 200.00 y valor mínimo de 92.00.
- **Cholesterol:** Este atributo es el colesterol sérico, la medida total de colesterol en sangre; tiene un valor medio en los pacientes de 199.02, con un valor máximo de 603.00 y un valor mínimo de 85.00. Se encuentra en miligramos por decilitro.

Descripción de Atributos (I)

Variables Clínicas

- **FastingBS:** Es la Glucosa en sangre en ayuno; hay un 76.66 % de registros con valores de glucosa en sangre menores a 120 mg/dl, codificado en 0, y un 23.34 % con valores mayores a 120 mg/dl, codificado en 1.
- **RestingECG:** Son los resultados de electrocardiogramas en reposo; hay 60.09 % codificado en Normal, un 19.41 % codificado en ST (tiene una anormalidad en el estudio) y, por último, un 20.50 % codificado en LVH (probablemente una hipertrofia en el ventrículo izquierdo).

Descripción de Atributos (I)

Esfuerzo, Señales y Variable Objetivo

- **MaxHR:** Este atributo es el máximo ritmo cardíaco registrado, tiene una media en los pacientes de 136.79, con un valor máximo de 202.00 y un valor mínimo de 60.00.
- **ExerciseAngina:** Es la angina producido por ejercicio, dolor en el pecho; donde hay un 59.54 % que no tenían dolor, codificado en N, y hay un 40.46 % que sí tenían dolor, codificado en Y.
- **Oldpeak:** Valor máximo de depresión del segmento ST (en milímetros) registrado en todas las derivaciones contiguas durante una prueba de esfuerzo. Forma parte del cálculo del riesgo de un paciente de isquemia o infarto de miocardio; valores más altos indican un mayor riesgo de enfermedad coronaria; tiene una media de 0.90, valor máximo de 6.20 y valor mínimo de -0.10.

Descripción de Atributos (V)

Esfuerzo, Señales y Variable Objetivo

- **ST_Slope:** La pendiente del segmento ST durante el ejercicio máximo; hay un 43.08 % en Up, un 50.05 % en Flat y luego un 6.87 % en Down [Up: pendiente ascendente, Flat: pendiente plana, Down: pendiente descendente].
- **HeartDisease (objetivo):**Variable de salida si posee una enfermedad cardíaca; donde hay un 44.71 % que no tiene enfermedad cardíaca, codificado en 0, y hay un 55.29 % que sí tienen enfermedad cardíaca, codificado en 1. Siendo ésta la **variable objetivo**.

Hiperparámetros

Los modelos requieren **hiperparámetros**: valores fijados antes del entrenamiento que controlan complejidad, regularización o velocidad de aprendizaje. A diferencia de los parámetros internos, no se aprenden de los datos y deben seleccionarse mediante:

- experimentación,
- validación cruzada,
- búsqueda sistemática de hiperparámetros.

Implementación

Se utilizaron las implementaciones estandarizadas de `scikit-learn`, que definen los hiperparámetros principales de cada modelo.
La búsqueda de combinaciones se realizó mediante **Grid Search**, evaluando:

- kernels y grados polinomiales en SVM,
- regularización L1/L2 en Regresión Logística,
- profundidad, criterios de partición y número de árboles en Random Forest,
- cantidad de vecinos en modelos basados en distancia.

Criterio General

Se partió de los valores por defecto y se ajustaron manualmente los hiperparámetros más influyentes para analizar su impacto en el desempeño, priorizando:

Criterio General

Se partió de los valores por defecto y se ajustaron manualmente los hiperparámetros más influyentes para analizar su impacto en el desempeño, priorizando:

- reproducibilidad,

Criterio General

Se partió de los valores por defecto y se ajustaron manualmente los hiperparámetros más influyentes para analizar su impacto en el desempeño, priorizando:

- reproducibilidad,
- consistencia comparativa,

Criterio General

Se partió de los valores por defecto y se ajustaron manualmente los hiperparámetros más influyentes para analizar su impacto en el desempeño, priorizando:

- reproducibilidad,
- consistencia comparativa,
- compatibilidad de cada hiperparámetro.

Modelos Utilizados

Los modelos fueron seleccionados por ser ampliamente usados en diagnóstico médico.

Modelos Utilizados

Los modelos fueron seleccionados por ser ampliamente usados en diagnóstico médico.

- **Regresión Logística** Modelo lineal para clasificación. Calcula la probabilidad de pertenecer a una clase mediante la función sigmoide. Ideal para relaciones aproximadamente lineales.

Modelos Utilizados

Los modelos fueron seleccionados por ser ampliamente usados en diagnóstico médico.

- **Regresión Logística** Modelo lineal para clasificación. Calcula la probabilidad de pertenecer a una clase mediante la función sigmoide. Ideal para relaciones aproximadamente lineales.
- **Naïve Bayes** Modelo probabilístico basado en el Teorema de Bayes. Asume independencia entre atributos. Muy eficiente computacionalmente.

Modelos Utilizados

Los modelos fueron seleccionados por ser ampliamente usados en diagnóstico médico.

- **Regresión Logística** Modelo lineal para clasificación. Calcula la probabilidad de pertenecer a una clase mediante la función sigmoide. Ideal para relaciones aproximadamente lineales.
- **Naïve Bayes** Modelo probabilístico basado en el Teorema de Bayes. Asume independencia entre atributos. Muy eficiente computacionalmente.
- **SVM (Support Vector Machines)** Busca maximizar el margen entre clases. Permite fronteras no lineales mediante kernels (RBF, polinómico).

Modelos Utilizados

Los modelos fueron seleccionados por ser ampliamente usados en diagnóstico médico.

- **Regresión Logística** Modelo lineal para clasificación. Calcula la probabilidad de pertenecer a una clase mediante la función sigmoide. Ideal para relaciones aproximadamente lineales.
- **Naïve Bayes** Modelo probabilístico basado en el Teorema de Bayes. Asume independencia entre atributos. Muy eficiente computacionalmente.
- **SVM (Support Vector Machines)** Busca maximizar el margen entre clases. Permite fronteras no lineales mediante kernels (RBF, polinómico).
- **Random Forest** Agrupación de múltiples árboles de decisión. Reduce varianza y mejora la generalización. Muy robusto a ruido y datos clínicos heterogéneos.

Clasificador Naïve Bayes (I)

Idea General

El **Clasificador Naïve Bayes** [3] es un método supervisado probabilístico basado en el *Teorema de Bayes*. Asume **independencia condicional** entre los atributos dado la clase, lo cual rara vez se cumple estrictamente, pero aun así funciona bien en la práctica.

Regla General

Para una clase C_l y atributos $X_1, \ldots, X_d$:

$$P(Y = C_l \mid X_1 = x_1, \ldots, X_d = x_d) = \frac{P(X_1 = x_1, \ldots, X_d = x_d \mid Y = C_l)\, P(Y = C_l)}{P(X_1 = x_1, \ldots, X_d = x_d)}$$

Probabilidad A Priori

La probabilidad previa de cada clase se estima como:

$$\hat{\pi}_l = P(Y = C_l) = \frac{|\{Y = C_l\}|}{n}$$

- $\hat{\pi}_l$: frecuencia relativa de la clase.
- n: tamaño total del conjunto de datos.

Supuesto Gaussiano

Para atributos continuos, NB usa la densidad Normal:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right]$$

Regla de Decisión

$$\hat{Y} = \underset{C_l}{\operatorname{argmax}} \left[\log P(Y = C_l) + \sum_{j=1}^{d} \log f(x_j \mid \mu_{jl}, \sigma_{jl}^2)\right]$$

$$\hat{\mu}_{jl} = \frac{1}{|C_l|} \sum_{i:Y_i=C_l} x_{ij} \qquad \hat{\sigma}_{jl}^2 = \frac{1}{|C_l|} \sum_{i:Y_i=C_l} (x_{ij} - \hat{\mu}_{jl})^2$$

Representación de Bayes Gaussiano

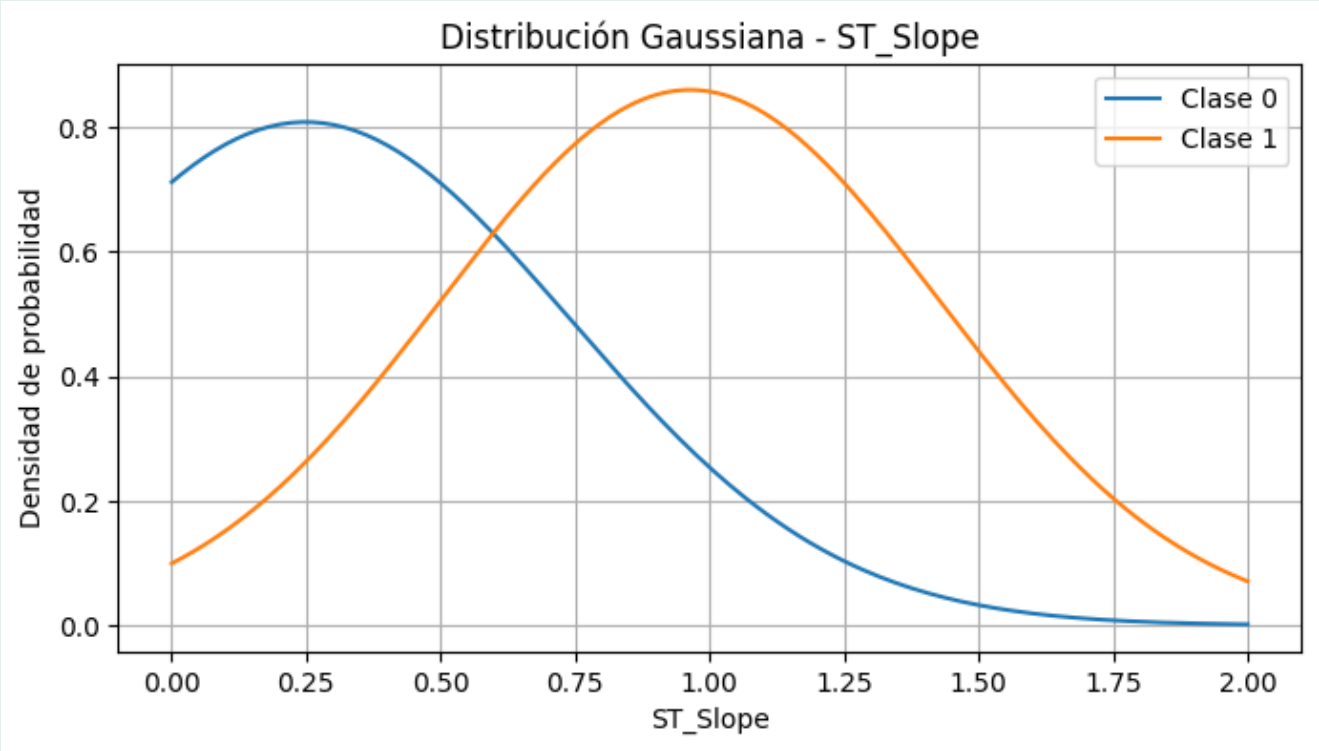

Figura: Comparación de ST_Slope en clase 1 y 0

¿Qué es la Regresión Logística? [?]

Técnica supervisada utilizada para modelar la probabilidad de pertenencia a una clase binaria.

$$P(I_i = 1 \mid \mathrm{x}_i)$$

Es un modelo:

- Lineal en los parámetros.
- No lineal en la salida (usa función logística).

Motivación

- Permite interpretar coeficientes como cambios en las *odds*.
- Robusto y eficiente para datasets con atributos numéricos y categóricos.

Interpretación

Si el modelo aumenta el valor lineal

$$z_i = \beta_0 + \sum_j \beta_j x_{ij},$$

la probabilidad crece de forma sigmoidal.

Inversa de la Sigmoide

La función logit transforma probabilidades en valores reales:

$$\text{logit}(p) = \ln\left(\frac{p}{1-p}\right) = \beta_0 + \sum_{j=1}^{D} \beta_j x_{ij}.$$

- Hace el modelo lineal en las *odds*.
- Permite interpretar coeficientes:

$$e^{\beta_j} = \text{cambio multiplicativo en las odds.}$$

¿Por qué es importante?

- Facilita el entrenamiento (problema convexo).
- Evita probabilidades fuera del rango $[0, 1]$.

Entrenamiento: Máxima Verosimilitud

Objetivo

Encontrar parámetros β que maximizan:

$$\ell(\beta) = \sum_{i=1}^{n} \left[l_i \ln(p(\mathrm{x}_i)) + (1 - l_i) \ln(1 - p(\mathrm{x}_i)) \right].$$

- Problema convexos $\rightarrow$ solución única.
- No existe solución cerrada $\rightarrow$ se usan métodos iterativos.

Por qué es log-verosimilitud

- Evita productos numéricamente inestables.
- Convierte productos en sumas $\rightarrow$ más fácil de optimizar.

Métodos Utilizados

- **Newton-CG**:

$$\beta^{(t+1)} = \beta^{(t)} - H^{-1}\nabla \tilde{J}$$

  - Utiliza información de **segunda derivada** (Hessiano).
  - Converge muy rápido cuando la función es suave.
  - Más costoso en memoria y tiempo porque requiere calcular o aproximar H^{-1}.
  - Adecuado para modelos con pocas características pero alta precisión.

- **L-BFGS** (Quasi-Newton):
  - No calcula el Hessiano completo; construye una **aproximación eficiente**.
  - Menor uso de memoria que Newton-CG $\rightarrow$ ideal para datasets medianos/grandes.
  - Muy estable para problemas bien condicionados.
  - Soporta regularización L2.

Métodos Utilizados

- **SAGA**:
  - Variante moderna de **Stochastic Gradient Descent**.
  - Actualiza los parámetros utilizando un muestreo por observación.
  - Excelente para:
    - datasets extremadamente grandes,
    - regularización L1 (Lasso),
    - modelos que requieren sparsidad en los coeficientes.
  - Convergencia rápida incluso en funciones no estrictamente suaves.

Tipos de Regularización

- L1 $\rightarrow$ elimina coeficientes.
- L2 $\rightarrow$ coeficientes pequeños pero no cero.
- EN $\rightarrow$ combinación útil práctica.

L1 (Lasso):

$$\tilde{J}_{L1} = J + \alpha \sum_{j=1}^{D} |\beta_j|$$

L2 (Ridge):

$$\tilde{J}_{L2} = J + \frac{1}{2}\alpha \sum_{j=1}^{D} \beta_j^2$$

Elastic Net:

$$\tilde{J}_{EN} = J + \alpha \left(\rho \sum_{j} |\beta_j| + \frac{1-\rho}{2} \sum_{j} \beta_j^2 \right)$$

Parámetro de Regularización (C)

La regresión logística incluye un término de regularización para controlar la complejidad del modelo y evitar sobreajuste. El hiperparámetro C es el **inverso de la fuerza de regularización**:

$$\alpha = \frac{1}{C}$$

- C **grande (poca regularización)**:
  - El modelo se ajusta más a los datos de entrenamiento.
  - Los coeficientes pueden tomar valores grandes.
  - Mayor riesgo de *overfitting*.
- C **pequeño (alta regularización)**:
  - Se penalizan los coeficientes grandes.
  - Reduce la complejidad del modelo.
  - Produce un modelo más estable y con mejor generalización.

Estrategias Multiclase

One-vs-Rest (OvR)

- Entrena un clasificador por clase.
- Cada modelo distingue: clase *i* vs resto.
- Simple y eficiente.

Multinomial

- Optimiza una única función de verosimilitud para todas las clases.
- Usado con solvers `lbfgs` y `newton-cg`.
- Mejora desempeño cuando las clases compiten entre sí.

Idea General

Los **árboles de decisión** son modelos no paramétricos que construyen reglas de decisión del tipo *if–else* para clasificar ejemplos. Mediante particiones jerárquicas, el árbol divide el espacio de atributos buscando **maximizar la pureza** en los nodos hijos.

- Se evalúan divisiones sobre atributos y umbrales.
- El cómputo pesado ocurre en la construcción del árbol; la predicción es muy rápida.
- Cada nodo terminal representa una clase estimada.

Probabilidad de Clase en un Nodo

Estimación de Probabilidades

La probabilidad de que un ejemplo en el nodo t pertenezca a la clase C_l se estima como:

$$p(l \mid t) = \frac{N_l(t)}{N(t)},$$

donde $N(t)$ es la cantidad total de ejemplos y $N_l(t)$ los de la clase C_l.

- Es un cálculo computacionalmente muy barato.
- Representa probabilidades empíricas ("equiprobables por frecuencia").

Requisitos de una función de impureza ϕ

Debe cumplir:

- $\phi(\mathrm{p}) \geq 0$ (no negatividad)
- $\phi(\mathrm{p}) = 0$ si el nodo es puro
- $\phi(\mathrm{p})$ máxima cuando todas las clases son equiprobables

La impureza del nodo t se define como:

$$\iota(t) = \phi\big(p(1|t), \ldots, p(K|t)\big).$$

Entropía de Shannon

$$H(D) = -\sum_{l=1}^{L} \frac{N_l(t)}{N(t)} \log_2 \left(\frac{N_l(t)}{N(t)} \right).$$

Índice de Gini

$$\text{Gini}(t) = 1 - \sum_{l=1}^{L} \left(\frac{N_l(t)}{N(t)} \right)^2.$$

- Ambos son 0 en nodos puros.
- Crecen cuando las clases están mezcladas.

Reducción de Impureza

La **reducción de impureza** generada al dividir el nodo t en dos nodos hijos t_1 y t_2 mediante una partición s se calcula como:

$$\Delta\iota(s,t) = \iota(t) - q_1\iota(t_1) - q_2\iota(t_2),$$

donde $q_j = \frac{N(t_j)}{N(t)}$.

- Se elige la división que maximiza $\Delta\iota$.
- Esta métrica es el núcleo del algoritmo CART [?].

Random Forest: Idea General

Principio del Método

Un **Random Forest** [?] combina muchos árboles independientes, cada uno entrenado sobre:

- un subconjunto *bootstrap* de los datos,
- un subconjunto aleatorio de atributos en cada división.

Esto reduce la varianza y mejora la generalización.

Voto Mayoritario

La clase predicha se obtiene mediante:

$$\hat{y} = \underset{l}{\operatorname{argmax}} \sum_{a=1}^{A} \mathbb{I}\big(\widehat{y_a}(\mathrm{x}) = l\big).$$

La **función indicadora**, definida como

$$\mathbb{I}(\widehat{y_a}(\boldsymbol{x}) = l) = \begin{cases} 1, & \text{si el árbol } a \text{ asigna la clase } l \text{ al ejemplo } \boldsymbol{x}, \\ 0, & \text{en caso contrario.} \end{cases}$$

Esta función contabiliza cuántos árboles votan por cada clase l, permitiendo que el modelo escoja aquella con mayor cantidad de votos.

- Cada árbol vota una clase.
- La predicción final es la clase más votada.

Hiperparámetros Principales

- **Criterio** (`gini`, `entropy`): mide la impureza.
- **n_estimators**: número de árboles.
- **max_depth**: profundidad máxima del árbol.
- **max_features**: número de atributos candidatos por división.
- **min_samples_split**: mínimo de muestras para dividir.
- **min_samples_leaf**: mínimo en una hoja.
- **bootstrap**: usar o no remuestreo con reemplazo.

Idea General

Las Máquinas de Soporte Vectorial [1] buscan una función de decisión que permita clasificar correctamente los datos, construyendo un **hiperplano óptimo** que maximice el **margen** entre clases.

- Operan sobre atributos numéricos $\rightarrow$ requieren codificación previa.
- Sólo los **vectores de soporte** determinan la frontera.
- El objetivo: maximizar la separación y mejorar la generalización.

Objetivo

Entre todos los hiperplanos que separan las clases, SVM elige aquel que:

$$\textbf{maximiza el margen } \varphi$$

la distancia mínima entre el hiperplano y los puntos más cercanos de cada clase.

Ecuación del Hiperplano

$$\langle \boldsymbol{w}, \boldsymbol{x} \rangle + \beta = 0$$

donde $\boldsymbol{w}$ es el vector normal al hiperplano y β es el término independiente.

Margen Rígido (Hard Margin)

Caso Ideal: Datos Separables

Se busca un hiperplano que clasifique perfectamente:

$$y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + \beta) \geq 1.$$

Problema de Optimización

$$\text{Minimizar } \frac{1}{2}\|\boldsymbol{w}\|^2 \quad \text{sujeto a } y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + \beta) \geq 1.$$

Margen Suave (Soft Margin)

Motivación

En la práctica los datos no son separables. Se permiten violaciones mediante **variables de holgura** $\xi_i \geq 0$.

Optimización con Penalización

$$\text{Minimizar } \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n}\xi_i$$

$$\text{sujeto a } y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i\rangle + \beta) \geq 1 - \xi_i.$$

- C grande $\rightarrow$ penaliza errores $\rightarrow$ margen chico.
- C chico $\rightarrow$ permite errores $\rightarrow$ margen grande.

Problema Dual

$$\text{Maximizar } -\frac{1}{2}\sum_{i,\ell}\alpha_i\alpha_\ell y_i y_\ell K(\boldsymbol{x}_i,\boldsymbol{x}_\ell)+\sum_i \alpha_i$$

$$\text{sujeto a } 0\le\alpha_i\le C,\quad \sum_i \alpha_i y_i = 0.$$

- Sólo los puntos con $\alpha_i > 0$ son **vectores de soporte**.
- La solución depende sólo de productos internos $\rightarrow$ clave para kernels.

Motivación

Cuando las clases no son separables linealmente, se proyectan los datos a un espacio de mayor dimensión.

Idea Central

Sin calcular la transformación explícita:

$$\langle \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j) \rangle \quad \Rightarrow \quad K(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

donde K es una **función kernel**.

- SVM se vuelve capaz de aprender fronteras no lineales.
- El costo computacional se mantiene manejable.

Kernels Comunes

- **Lineal**:

$$K(\boldsymbol{a}, \boldsymbol{b}) = \langle \boldsymbol{a}, \boldsymbol{b} \rangle$$

- **RBF / Gaussiano**:

$$K(\boldsymbol{a}, \boldsymbol{b}) = \exp\left(-\gamma \|\boldsymbol{a} - \boldsymbol{b}\|^2 \right)$$

- **Polinómico**:

$$K(\boldsymbol{a}, \boldsymbol{b}) = (\langle \boldsymbol{a}, \boldsymbol{b} \rangle + r)^q$$

- **Sigmoide**:

$$K(\boldsymbol{a}, \boldsymbol{b}) = \tanh(\gamma \langle \boldsymbol{a}, \boldsymbol{b} \rangle + r)$$

Principales

- **C**: regula el balance margen–errores.
- **kernel**: lineal, rbf, poly, sigmoid.
- γ: controla influencia local del punto.
- **degree** (d): sólo para kernel polinómico.
- **coef0** (r): usado por `poly` y `sigmoid`.

Escalas de γ en scikit-learn

$$\texttt{gamma = scale} : \frac{1}{D \cdot \mathrm{Var}(X)} \qquad \texttt{gamma = auto} : \frac{1}{D}$$

Objetivo

Evaluar el desempeño del calificador de cada modelo de Aprendizaje Automático mediante métricas de rendimiento que cuantifican la capacidad de clasificación.

Importancia

El objetivo no es solo un buen rendimiento en datos de entrenamiento, sino la **capacidad de generalización** a entradas nuevas no vistas.

Validación Cruzada K-fold

Se divide el conjunto de datos en K pliegues $\{G_1, G_2, \ldots, G_K\}$. Cada pliegue sirve como conjunto de prueba una vez y como entrenamiento $K-1$ veces. Métrica promedio:

$$\widehat{M} = \frac{1}{K} \sum_{i=1}^{K} M_i$$

Caso Binario: Matriz de Confusión

Una matriz de confusión, que se puede observar en la Tabla 47, es una forma simple de saber de qué forma está clasificando el algoritmo, donde una clase es considerada **positiva** P y la otra **negativa** N.

Matriz de Confusión

		Predicción	
		Positivo	**Negativo**
Verdad	**Positivo**	TPcolor **Verdadero Positivo (TP)**	FNcolor **Falso Negativo (FN)**
	Negativo	FNcolor **Falso Positivo (FP)**	TPcolor **Verdadero Negativo (TN)**

Caso Binario: Matriz de Confusión

Definición

Clasifica las predicciones en:

- **Verdaderos Positivos (TP):** Casos positivos correctamente clasificados.
- **Verdaderos Negativos (TN):** Casos negativos correctamente clasificados.
- **Falsos Positivos (FP):** Casos negativos clasificados como positivos.
- **Falsos Negativos (FN):** Casos positivos clasificados como negativos.

Accuracy

Definición

Proporción de instancias correctamente clasificadas:

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} = \frac{TP + TN}{\text{Total}}$$

Conjunto de predicciones

$$Accuracy(y, \hat{y}) = \frac{1}{n} \sum_{I=0}^{n-1} \mathbb{I}(\hat{y}_I = y_I), \quad \mathbb{I}(\hat{y}_I = y_I) = \begin{cases} 1, & \text{si } \hat{y}_I = y_I \\ 0, & \text{en caso contrario} \end{cases}$$

Resumen

$$\text{Accuracy} = \frac{\text{Número de predicciones correctas}}{\text{Número total de muestras}}$$

Precision

Definición

Mide la probabilidad de que la predicción positiva sea correcta:

$$\text{Precision} = \frac{TP}{TP + FP}$$

Recall / Sensibilidad

Definición

Mide la probabilidad de detectar un caso positivo real:

$$\text{Recall} = \text{TPR} = \frac{TP}{TP + FN} = \frac{TP}{P}$$

Definición general

Media armónica ponderada de precision y recall:

$$F_\beta = \frac{(1+\beta_f^2)\,\text{precision} \cdot \text{recall}}{\beta_f^2 \text{precision} + \text{recall}}$$

F1-score ($\beta_f = 1$)

$$F1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

$$F1 = \frac{2\,TP}{2\,TP + FP + FN}$$

Área Bajo la Curva ROC (AUC)

Definición

Mide la capacidad de un clasificador para distinguir entre clases.
La curva ROC grafica TPR vs FPR al variar el umbral de decisión.

Interpretación

- Ideal: $(0, 1) \Rightarrow \text{AUC} = 1$
- Aleatorio: $\text{TPR} = \text{FPR} \Rightarrow \text{AUC} = 0.5$
- Razonable: $0{,}5 < \text{AUC} \leq 1$

Ejes del gráfico

- Eje Y: TPR
- Eje X: FPR

Definición

Sea un modelo predictivo $\mathcal{M}$ entrenado sobre un conjunto de datos X, y sea M la métrica de referencia del modelo sobre los datos originales.

Procedimiento: Paso 1

Iteración por atributo

Para cada atributo j del conjunto de datos:

1. Repetir el proceso K veces (para reducir la varianza de la estimación):

Permutación de la característica

Para cada repetición k:

1. Generar una versión alterada del conjunto de datos $X^{(k,j)}$ permutando aleatoriamente la columna j, manteniendo las demás columnas sin cambios.
2. Calcular la métrica M del modelo sobre los datos permutados:

$$M_{k,j} = \text{valor de la métrica } M \text{ del modelo } \mathcal{M} \text{ con } X^{(k,j)}$$

Cálculo de importancia

Calcular la importancia de la característica j como la disminución promedio del puntaje respecto del puntaje de referencia:

$$I_j = M - \frac{1}{K} \sum_{k=1}^{K} M_{k,j}$$

Significado de I_j

I_j mide la pérdida de desempeño al romper la relación entre la característica j y la variable objetivo.
Valores más altos de $I_j \Rightarrow$ características más relevantes para el modelo.

Resultados – Dataset Binario

Descripción general

- Se evaluaron NB, Regresión Logística, Random Forest y SVM.
- Métricas: Accuracy, Recall, F1-Score, AUC y tiempo de entrenamiento.
- Se probó **undersampling**, pero redujo la generalización.
- Se decidió mantener la distribución original del dataset.

Objetivo

Evaluar cuál modelo obtiene el mejor rendimiento clasificando entre dos clases.

Desempeño general

- Modelo muy eficiente computacionalmente, con una muy buena capacidad clasificadora.
- Los distintos valores de suavizado (*var smoothing*) no generó diferencias.
- Mantiene un rendimiento competitivo pese al supuesto de independencia condicional entre atributos.

Hiperparámetros evaluados

Hiperparámetro	**Valores evaluados**
Suavizado	10^{-9}, 10^{-8}, 10^{-7}, 10^{-6}, 10^{-5}

Desempeño general

- Modelo muy eficiente computacionalmente, con una muy buena capacidad clasificadora.
- Los distintos valores de suavizado (*var smoothing*) no generó diferencias.
- Mantiene un rendimiento competitivo pese al supuesto de independencia condicional entre atributos.

Resultados del modelo

Configuración	*Accuracy*	Recall	F1-Score	*AUC*	Tiempo (s)	Precisión
Todas las consideradas	0.8420	0.8420	0.8420	0.9138	0.10	0.8433

Características del modelo

- Modelo lineal, rápido y estable.
- Obtuvo una buena capacidad clasificadora.
- Opción eficiente en cuestión de tiempo.

Hiperparámetros evaluados

Hiperparámetro	**Valores evaluados**
C	0, 0.1, 0.01
Penalidad	Ninguna, Lasso, Ridge, Elastic Net
Solver	L-BFGS, SAGA, Newton-CG
Multiclase	OvR, *multinomial*

Características del modelo

- Modelo lineal, rápido y estable.
- Obtuvo una buena capacidad clasificadora.
- Opción eficiente en cuestión de tiempo.

Resultados del modelo

Configuración	*Accuracy*	Recall	F1-Score	*AUC*	Tiempo (s)	Precisión
$C = 1$ Penalidad = Lasso *Solver* = SAGA Multiclase = OvR	0.8485	0.8485	0.8481	0.9050	0.13	0.8495

Desempeño general

- Modelo con mejor rendimiento global.
- Excelente capacidad de generalización.
- Mayor costo computacional debido a la construcción del bosque, pero predicción más sencilla.

Hiperparámetros evaluados

Hiperparámetro	**Valores evaluados**
Criterio	`gini`, `entropy`
`max_depth`	Ninguna, 3, 5, 7, 9
`min_samples_split`	2, 5, 10
`min_samples_leaf`	1, 2, 4
`max_features`	`None`, `sqrt`, `log2`

Random Forest – Resultados

Desempeño general

- Modelo con mejor rendimiento global.
- Excelente capacidad de generalización.
- Mayor costo computacional debido a la construcción del bosque, pero predicción más sencilla.

Resultados del modelo

Configuración	*Accuracy*	Recall	F1-Score	*AUC*	Tiempo (s)	Precisión
Criterio = `entropy` `max_depth` = 7 `min_samples_split` = 5 `min_samples_leaf` = 1 `max_features` = `sqrt`	0.8780	0.8780	0.8775	0.9315	1.38	0.8731

Comportamiento del modelo

- Alta capacidad discriminatoria.
- Kernel RBF fue el de mejor desempeño, por lo cual fue necesario cambiar la dimensionalidad para encontrar el hiperplano.
- Tiempo de entrenamiento moderado.

Hiperparámetros evaluados

Hiperparámetro	**Valores evaluados**
C	0.001, 0.01, 0.1, 1, 10, 15, 20, 25
`kernel`	`linear`, `poly`, `rbf`, `sigmoid`
γ	`scale`, `auto`, 0.001, 0.01, 0.1, 1
`degree`	$2, 3, \ldots, 10$

Comportamiento del modelo

- Alta capacidad discriminatoria.
- Kernel RBF fue el de mejor desempeño, por lo cual fue necesario cambiar la dimensionalidad para encontrar el hiperplano.
- Tiempo de entrenamiento moderado.

Resultados del modelo

Configuración	*Accuracy*	Recall	F1-Score	*AUC*	Tiempo (s)	Precisión
$C = 1$ `kernel = rbf` $\gamma = 0.1$	0.8616	0.8616	0.8609	0.9232	0.60	0.8628

Conclusiones por desempeño

- Se observa que Random Forest obtiene la mayor puntuación en todas las métricas, seguido por la Regresión Logística y SVM.

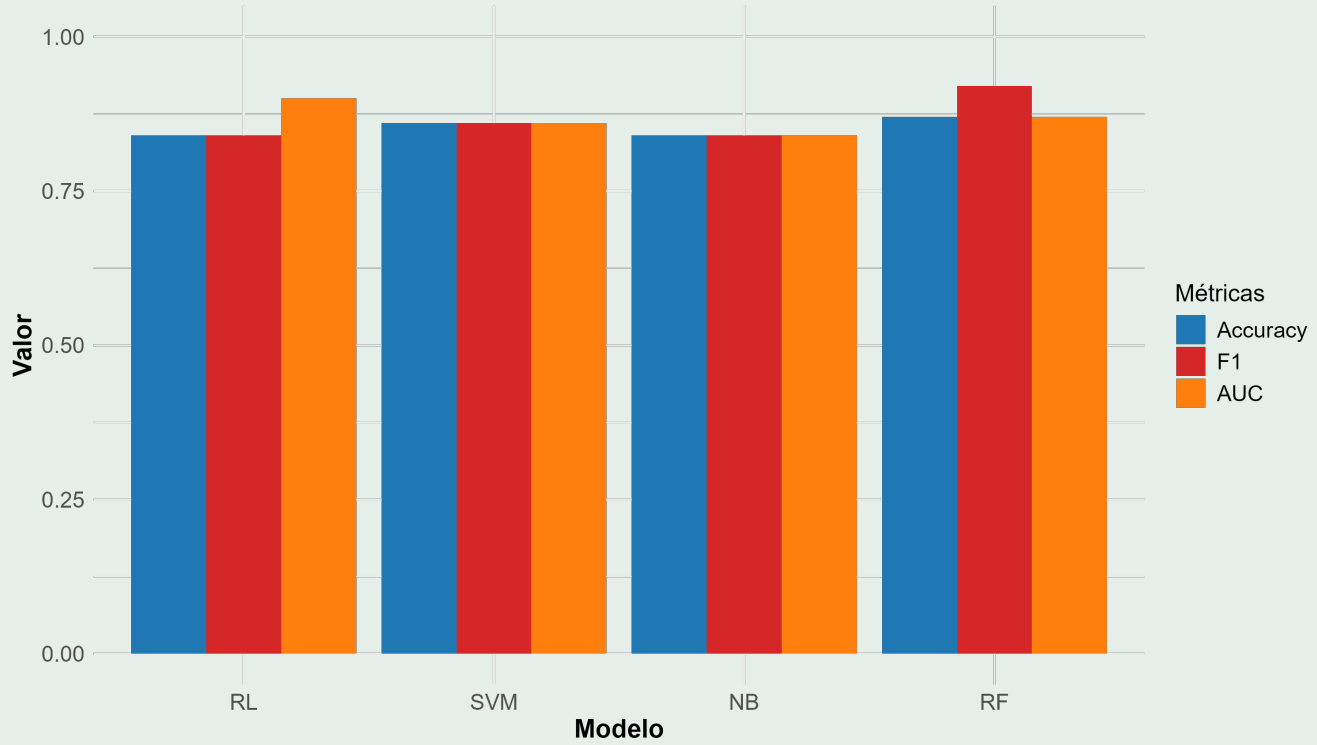

Figura: Comparación del desempeño de los mejores modelos en el caso binario.

Descripción

- Se midió cómo varía la métrica del modelo al permutar cada feature.
- Permite identificar las variables más influyentes.
- Se calculó para NB, RL, RF y SVM.

Hallazgo general

ST_Slope aparece como la característica más relevante en los cuatro modelos.

Comparación de los Mejores Modelos

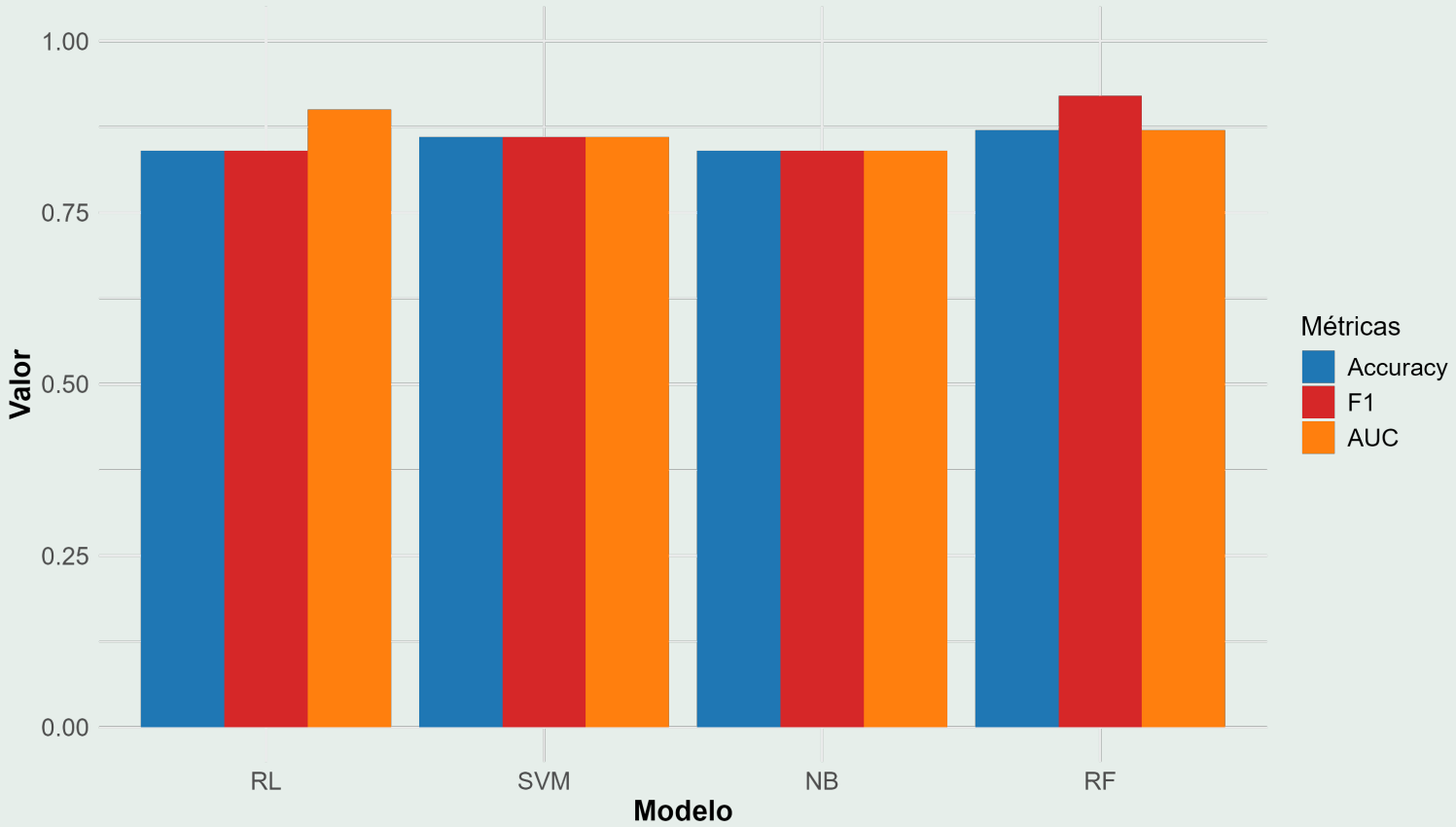

Figura: Comparación del desempeño de los mejores modelos en el caso binario.

Observaciones

- Todos los modelos mejoran al incluir las variables más importantes.
- RF y SVM muestran el patrón más estable y creciente.
- Con pocas características ya alcanzan valores competitivos.

Conclusión

Es posible reducir el número total de características sin pérdida significativa de desempeño.

Importancia de las características para NB en el caso binario.

Característica	Importancia (Permutación)
ST_Slope	0.027015
ExerciseAngina	0.023747
Oldpeak	0.018736
ChestPainType	0.018519
Cholesterol	0.014815
Sex	0.014270
FastingBS	0.004575
RestingBP	0.001852
MaxHR	-0.000218
RestingECG	-0.001198
Age	-0.003595

Importancia de las características para RL en el caso binario.

Característica	Importancia (Permutación)
ST_Slope	0.072440
ExerciseAngina	0.026797
ChestPainType	0.020806
Sex	0.011329
FastingBS	0.010240
Cholesterol	0.009150
Oldpeak	0.006100
Age	0.003922
MaxHR	0.003268
RestingBP	0.000545
RestingECG	0.000218

Importancia de las características para RF en el caso binario.

Característica	Importancia (Permutación)
ST_Slope	0.254265
ChestPainType	0.127319
Oldpeak	0.113156
ExerciseAngina	0.105952
Cholesterol	0.099872
MaxHR	0.088635
Age	0.065807
RestingBP	0.055053
Sex	0.040916
FastingBS	0.030069
RestingECG	0.018956

Importancia de las características para SVM en el caso binario.

Característica	Importancia (Permutación)
ST_Slope	0.106209
Cholesterol	0.031808
Oldpeak	0.024510
ChestPainType	0.023312
Sex	0.011438
MaxHR	0.008715
ExerciseAngina	0.008388
Age	0.007952
RestingBP	0.005773
RestingECG	0.004902
FastingBS	0.004575

Evolución al incorporar métricas.

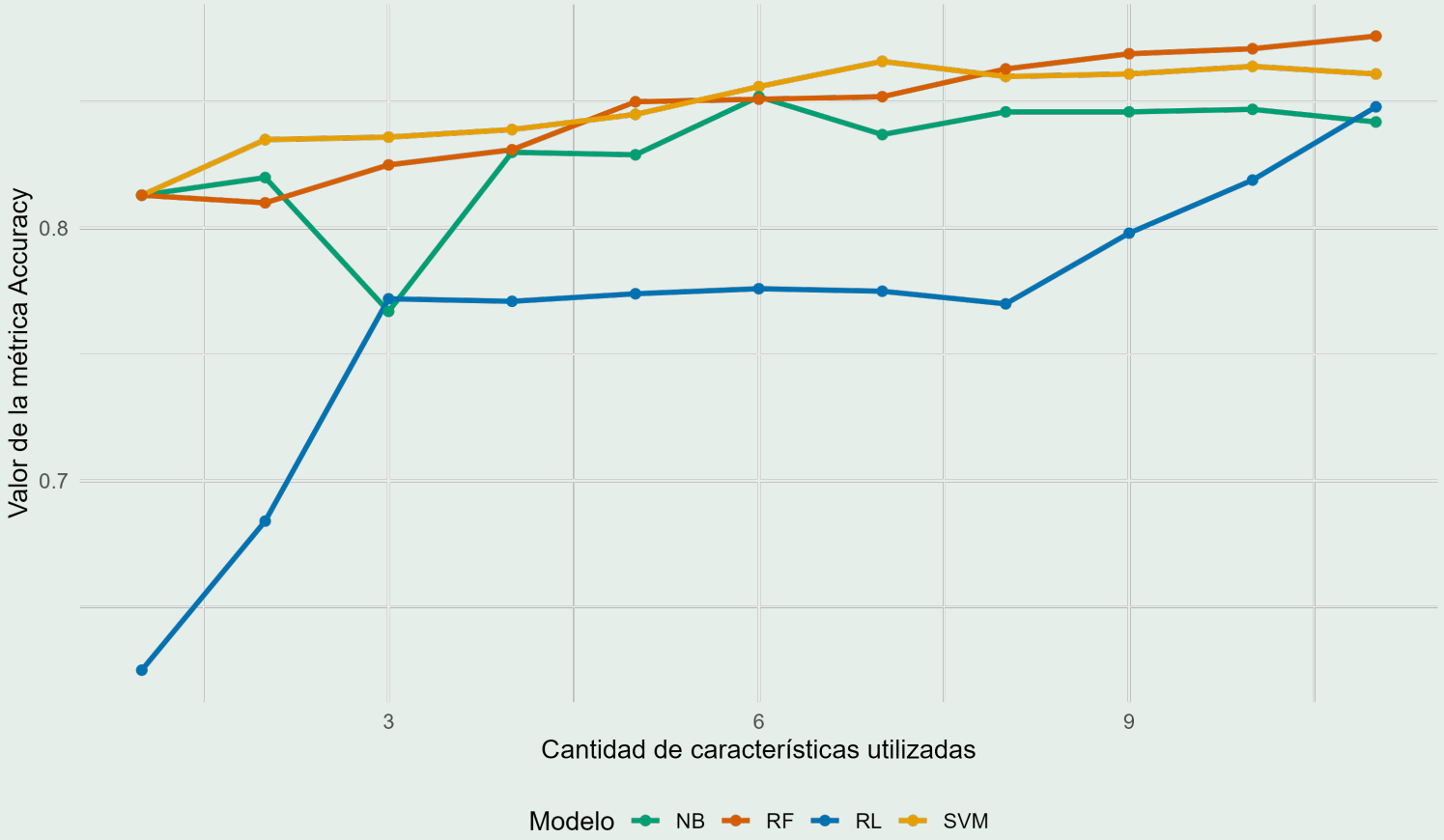

Figura: Valores de *Accuracy* según la cantidad de características en el caso binario.

Evolución al incorporar métricas.

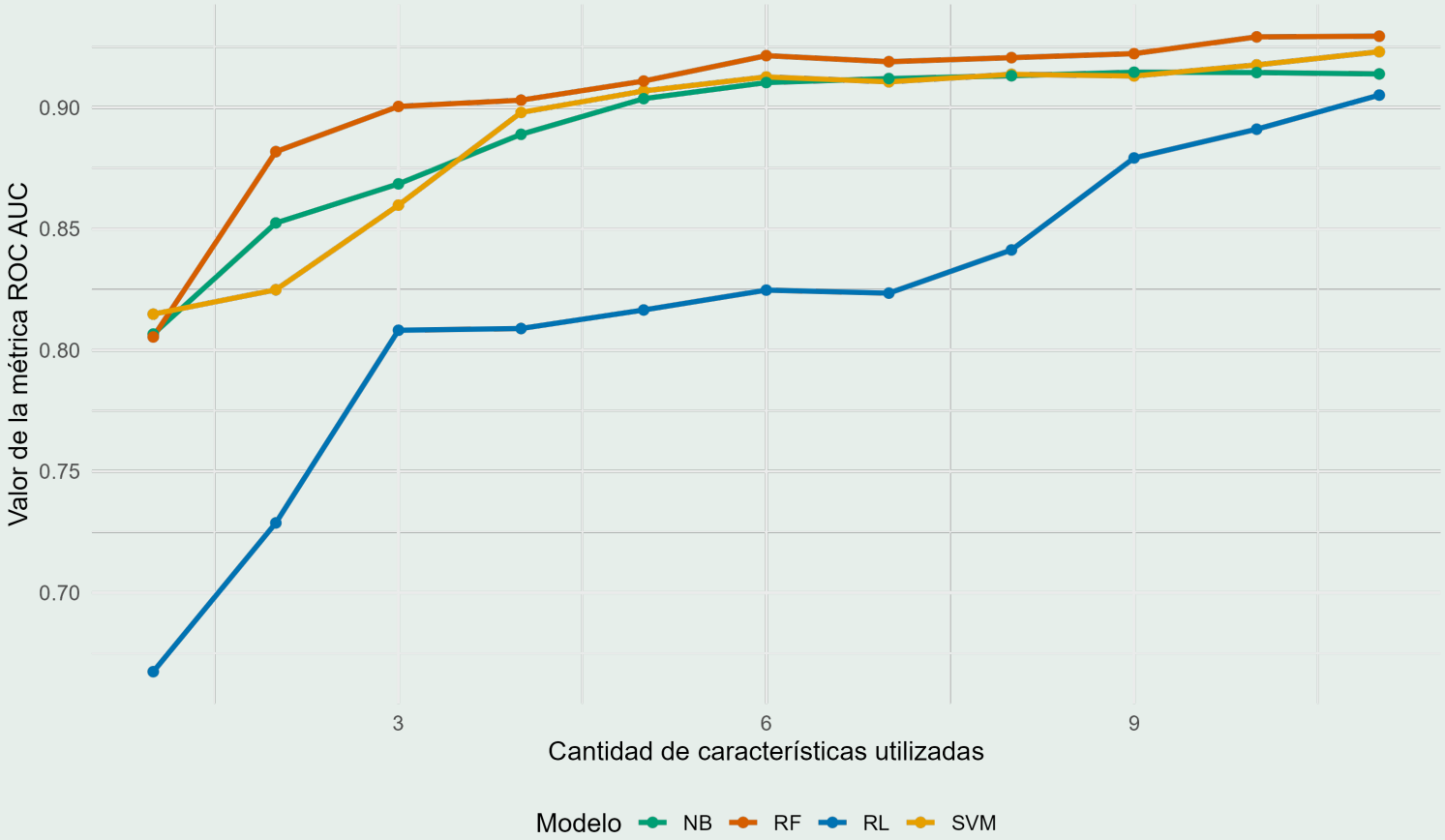

Figura: Valores de *AUC* según la cantidad de características en el caso binario.

Evolución al incorporar métricas.

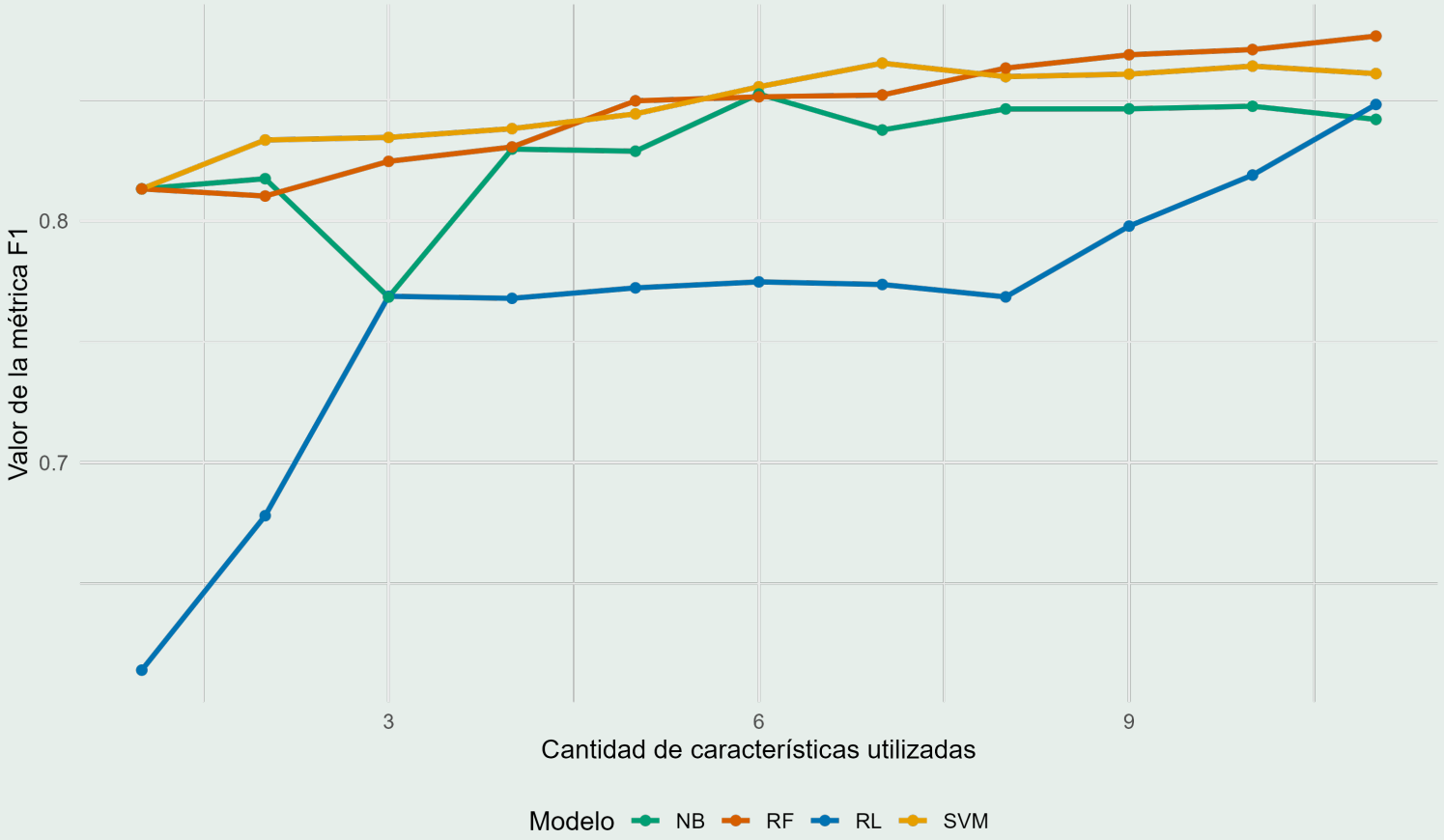

Figura: Valores de F1-Score según la cantidad de características en el caso binario.

Random Forest

- Mayor rendimiento en ambas tareas (binaria y multiclase).
- Reduce el sobreajuste gracias al voto mayoritario.
- Modela relaciones no lineales e interacciones complejas.

Máquinas de Soporte Vectorial

- Excelente rendimiento con kernels **RBF** y **polinómicos**.
- El *kernel trick* permite fronteras no lineales complejas.
- Muy útil cuando los datos no son separables linealmente.

Regresión Logística

- Modelo lineal interpretable y de buena generalización.
- Capta relaciones simples entre las características y la clase.
- Menor capacidad que RF y SVM para fronteras no lineales.
- Útil como modelo base por su estabilidad y claridad interpretativa.

Naive Bayes

- Extrema eficiencia computacional — entrenamiento casi instantáneo.
- Supone independencia condicional entre atributos.
- Aun cuando el supuesto no se cumple, mantiene buen rendimiento.
- Muy competitivo como modelo inicial y para datasets pequeños.

Interpretabilidad vs. Desempeño

- RF y SVM obtienen las mejores métricas, pero son menos interpretables.
- RL y NB permiten explicaciones claras, valiosas en entornos clínicos.

Coherencia con el conocimiento médico

- Caso binario: destacan *ST_Slope*, *ChestPainType*, *Oldpeak*.
- Caso multiclase: resaltan *ASTV*, *ALTV*, *MSTV*.
- Las variables importantes coinciden con indicadores usados en práctica clínica.

Proyección a Implementaciones Médicas

- Los modelos pueden integrarse a sistemas de apoyo al diagnóstico.
- Necesitan validación clínica, calibración y análisis interpretativo.
- La selección de características mejora eficiencia y confianza del sistema.

Estrategias de Optimización

- **Ajuste más fino de hiperparámetros:** uso de *Randomized Search* o *Bayesian Optimization* para explorar más eficientemente el espacio, y luego un *Grid Search* focalizado.

Estrategias de Optimización

- **Ajuste más fino de hiperparámetros:** uso de *Randomized Search* o *Bayesian Optimization* para explorar más eficientemente el espacio, y luego un *Grid Search* focalizado.
- **Manipulación de características:** aplicar técnicas como PCA o generar variables sintéticas para mejorar la calidad del espacio de representación.

Estrategias de Optimización

- **Ajuste más fino de hiperparámetros:** uso de *Randomized Search* o *Bayesian Optimization* para explorar más eficientemente el espacio, y luego un *Grid Search* focalizado.
- **Manipulación de características:** aplicar técnicas como PCA o generar variables sintéticas para mejorar la calidad del espacio de representación.
- **Validación cruzada más robusta:** incrementar la cantidad de particiones o emplear CV estratificada más exhaustiva.

Mejoras Potenciales y Consideraciones

Estrategias de Optimización

- **Ajuste más fino de hiperparámetros:** uso de *Randomized Search* o *Bayesian Optimization* para explorar más eficientemente el espacio, y luego un *Grid Search* focalizado.
- **Manipulación de características:** aplicar técnicas como PCA o generar variables sintéticas para mejorar la calidad del espacio de representación.
- **Validación cruzada más robusta:** incrementar la cantidad de particiones o emplear CV estratificada más exhaustiva.

Conclusión General

Los modelos muestran un desempeño altamente satisfactorio, con margen de mejora a través de un refinamiento de hiperparámetros y un análisis más profundo de la estructura de los datos.

Muchas gracias

Preguntas

[1] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[2] fedesoriano. Heart failure prediction dataset. https://www.kaggle.com/fedesoriano/heart-failure-predictionhttps://www.kaggle.com/fedesoriano/heart-failure-prediction, September 2021.

[3] David J Hand and Keming Yu. Idiot's bayes—not so stupid after all? *International statistical review*, 69(3):385–398, 2001.

[4] Jonas L Isaksen, Malene Nørregaard, Martin Manninger, Dobromir Dobrev, Thomas Jespersen, Ben Hermans, Jordi Heijman, Gernot Plank, Daniel Scherr, Thomas Pock, et al. Evaluating artificial intelligence-enabled medical tests in cardiology: Best practice. *IJC Heart & Vasculature*, 60:101783, 2025.

[5] Narender Kumar and Dharmender Kumar. Machine learning based heart disease diagnosis using non-invasive methods: A review. In *Journal of Physics: Conference Series*, volume 1950, page 012081. IOP Publishing, 2021.