10/12/2025

ML: SUPERVISED LEARNING

GRADO CIENCIA DE DATOS E INTELIGENCIA ARTIFICIAL – 3º

APRENDIZAJE AUTOMÁTICO I

DANIEL RIVAS HIDALGO (daniel.rivas@alumnos.upm.es)

Contents

1. Introducción

El objetivo clínico de este proyecto es desarrollar un modelo predictivo capaz de detectar Diabetes Mellitus tipo 2 utilizando datos de salud de los pacientes. Dado el gran impacto de los Falsos Negativos en medicina (pacientes enfermos no diagnosticados), el estudio prioriza la sensibilidad del modelo, intentando no sacrificar por completo su precisión.

2. Descripción del Problema

Utilizamos el dataset *Diabetes Health Indicators* de Kaggle, compuesto por 21 variables de salud. El desafío técnico central es el desbalanceo severo de clases: el 86% de la población es sana (Clase 0) frente a solo un 14% de diabéticos (Clase 1). Este desequilibrio provoca que los modelos tradicionales tiendan a ignorar la clase minoritaria para maximizar el acierto global (Accuracy), lo cual es inadmisible en un contexto médico. Nuestro reto es forzar al algoritmo a aprender patrones complejos de la clase enferma evitando el sobreajuste.

3. Metodología

3.1. Preprocesamiento y Escalado

Para garantizar la convergencia de algoritmos basados en distancia (KNN, SVM) y gradiente, aplicamos un preprocesamiento diferenciado:

- Discretización: Las variables con distribuciones sesgadas (MentHlth, PhysHlth) fueron categorizadas para reducir el ruido.
- Escalado: MinMaxScaler para variables ordinales/binarias y una combinación de StandardScaler y MaxAbsScaler para variables continuas (BMI), homogeneizando los rangos.

3.2. Estrategia de Muestreo Híbrido

Para mitigar el desbalanceo (86/14), diseñamos una arquitectura híbrida con diferentes técnicas de muestreo dentro del bucle de entrenamiento:

- **Random Under-sampling**: Reducción controlada de la clase mayoritaria hasta una proporción 65/35. Esto preserva información valiosa que se perdería en un balanceo 50/50 puro.
- **SMOTE** (Synthetic Minority Over-sampling Technique): Generación de datos sintéticos sobre la clase minoritaria para cubrir la brecha restante hasta el equilibrio perfecto.

Esta combinación minimiza el riesgo de overfitting (propio de un SMOTE puro) y la pérdida de información relevante (propia del Under-sampling agresivo).

3.3. Diseño Experimental y Métricas

Para la selección de los parámetros óptimos en cada modelo, se utilizó *Stratified Cross-Validation* con 10 *k-folds*, para garantizar que cada pliegue mantenga la proporción real de clases. Algo a

destacar sobre la implementación es que, tanto el escalado como el muestreo híbrido, se incluyen en el propio *cross-validation*. De esta manera, dentro de cada *fold* los escaladores se entrenan (*fit*) solo con los datos de entrenamiento, y a su vez estos son los únicos sobre los que se aplican las técnicas de muestreo (nunca en validación). El objetivo de esta metodología es evitar la filtración de datos (***data leakage***).

En cuanto a la métrica de optimización para la selección del modeloTuvimos dudas en cdecidir cuál sería la mejor. Nuestra estrategia de muestreo híbrido ya actúa aumenta significativamente el Recall: al entrenar con datos equilibrados (50/50), el modelo tiende naturalmente a sobre-predecir la clase minoritaria, aumentando la sensibilidad a costa de la precisión. Si hubiéramos optimizado por *Recall* o *F2*, el volumen de Falsos Positivos predichos sería inaceptable. Por ello, asumiendo que el modelo ya tiene una buena sensibilidad, fruto de haber equilibrado las clases en el entrenamiento, buscamos imponer un control de calidad sobre la Precisión y por ello elegimos **F1**.

4. Modelos, Resultados y Análisis

A continuación, se detalla el desempeño de cada algoritmo tras la optimización de los distintos parámetros mediante *GridSearch* con validación cruzada estratificada.

4.1. Regresión Logística

La Regresión Logística se estableció como el modelo base debido a su naturaleza lineal. Además ofrece una muy buena interpretabilidad, gracias a que todas las variables están escaladas en el mismo rango y, por tanto, sus coeficientes en el modelo nos indican su importancia.

Para este modelo, vamos a comparar el rendimiento en dos escenarios distintos de gestión del balanceo de datos.

Definición de Escenarios:

- Escenario A (*Class Weights*): Se utilizan los datos originales (escalados pero no remuestreados). Paracombatir el desbalance de los datos se usará el parámetro *class_weight='balanced'*, que hace que el modelo ajuste su función de coste penalizando más severamente los errores en la clase minoritaria-
- Escenario B (Muestreo Híbrido): Se aplica el Pipeline de Muestreo Híbrido (Undersampling + SMOTE) definido en la metodología. Esto altera físicamente la distribución de entrenamiento antes de que el modelo "vea" los datos.

Estudio de Hiperparámetros

Para ambos escenarios, se ha explorado el mismo espacio de configuración:

- **Regularización (*C*):** Se evaluó el rango [0.01, 0.1, 1, 10, 100, 1000]. Valores pequeños implican una regularización fuerte para evitar *overfitting*. Valores grandes permiten al modelo ajustarse fielmente a los datos de entrenamiento.
- **Solvers + Penalización: *lbfgs* + l2** son los parámetros por defecto de regresión logística. Probamos también **saga + l1 (Lasso)**, que con dicha penalización es capaz de llevar

coeficientes a cero (selección de variables), lo cual puede útil si el muestreo introduce ruido.

Resultados y Comparativa de Escenarios

Para el Escenario A, los mejores parámetros fueron: {'C': 100, 'penalty': 'l2', 'solver': 'lbfgs'}. Es decir, los parámetros por defecto, con una regularización de 100, obteniendo un **F1 = 0.4406**.

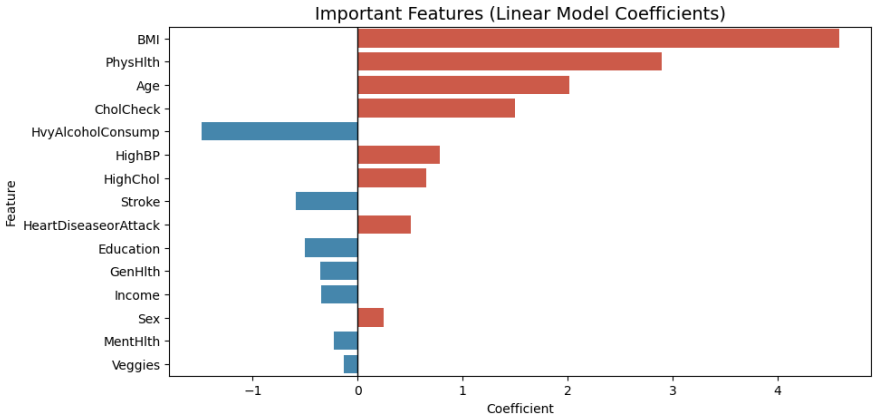
En el Escenario B, los mejores parámetros fueron: {'C': 100, 'penalty': 'l1', 'solver': 'saga'}, con un **F1= 0.4416.**

Aunque la métrica es muy similar, la configuración óptima cambia. El modelo prefiere el *solver* ***saga*** con penalización **L1 (Lasso)**. Esto nos indica que, al introducir datos sintéticos (SMOTE), el modelo se beneficia de la capacidad de L1 para eliminar ruido y seleccionar solo las características más robustas. Aunque la diferencia numérica es marginal (+0.001), el Muestreo Híbrido demuestra ser ligeramente superior.

Por tanto, **se selecciona la Estrategia de Muestreo Híbrido como el estándar de preprocesamiento** para los siguientes modelos, asegurando así que las comparaciones futuras se realicen bajo las mismas condiciones de datos, y poder así atribuir las posibles mejorías o empeoramientos a los propios modelos, y no a los datos.

Interpretación de Variables:

Dado que el modelo es lineal, podemos analizar los coeficientes resultantes para interpretar el modelo y entender los factores de riesgo:



1. **Factores Determinantes:** Al igual que se observará posteriormente en otros modelos, el **BMI (IMC)** y la **Salud Física (PhysHlth)** dominan la predicción con coeficientes positivos altos, sugiriendo que ligeros cambios en el índice de masa corporal o en la salud física, son muy sensibles a la frontera de decisión.

2. **Relaciones Inversas:** El coeficiente negativo del *Alto consumo de alcohol* es un ejemplo perfecto de **causalidad inversa:** los pacientes suelen **dejar de beber después** de recibir un diagnóstico o sufrir un susto relacionado con la salud; no es que el alcohol prevenga la enfermedad.

4.2. K-Nearest Neighbors (K-NN)

Estudio de Hiperparámetros:

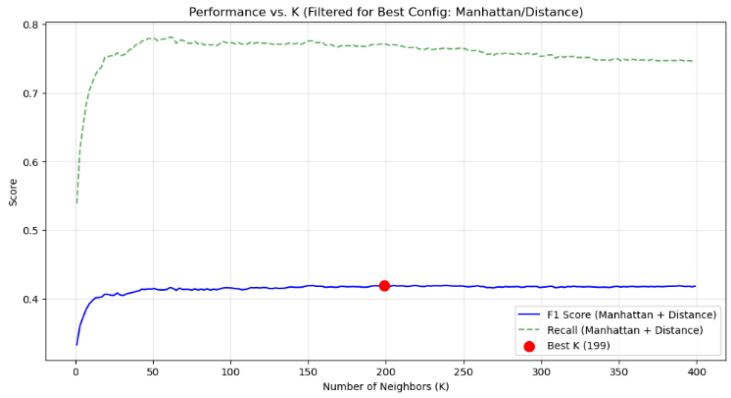
1. **Número de Vecinos (*n_neighbors*):** Evaluado en un rango amplio (1 a 400). Un número bajo de vecinos captura patrones locales (sensible al ruido), mientras que uno alto suaviza la frontera de decisión.
2. **Métrica de Distancia (*metric*):** Comparamos **Euclídea** vs. **Manhattan**. Se estudian ambas ya que la euclídea es la clásica, y manhattan puede ser interesante, ya que con grandes dimensiones de datos puede llegar a ser más robusta ante outliers o variables menos importantes
3. **Estrategia de Pesos (*weights*):** Estudiamos **'uniform'** (democracia pura) y **'distance'** (ponderación inversa a la distancia 1/d). Esta última puede ser más interesante en nuestro caso, dando prioridad a los pacientes con perfiles casi idénticos frente a aquellos que son meramente "vecinos".

Configuración Óptima

La validación cruzada seleccionó la siguiente combinación: *n_neighbours* = **199**, *metric* = **Manhattan** y *weights* =**'distance'**

- **Interpretación:** La necesidad de consultar casi 200 vecinos para una predicción fiable es un hallazgo estructural. Confirma la **ausencia de clústeres locales definidos**; la "señal" de la diabetes es difusa y requiere un suavizado global masivo. En cuanto a la métrica,el uso de la distancia Manhattan valida nuestra hipótesis sobre su mejor comportamiento en espacios dispersos.

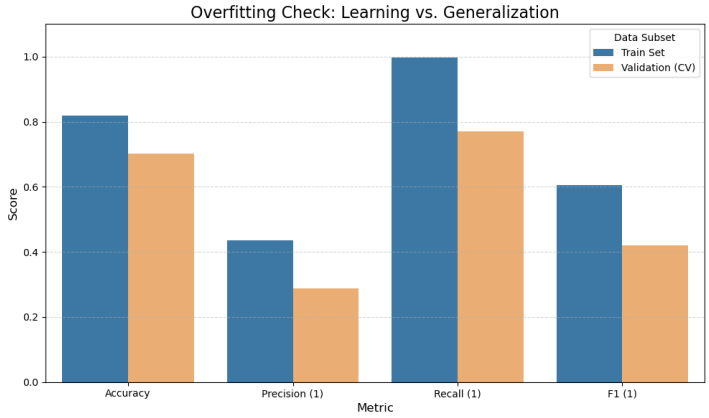
Respecto al parámetro ***n_neighbors***, la gráfica siguiente evidencia una notable estabilidad del F1-Score (línea azul) incluso con valores extremos. Esto confirma que la estrategia weights='distance' funciona: la influencia de los vecinos lejanos se anula.



Evaluación de Rendimiento y Generalización

Al comparar el rendimiento del modelo entre los datos de entrenamiento y validación, vemos que en estos últimos el Recall es casi perfecto (99,7%) no indica sobreajuste, sino que es una consecuencia directa de usar el parámetro *weights='distance'*, el cual asigna un peso infinito al propio dato, ya que el modelo lo tiene memorizado, garantizando el acierto. La métrica real es la validación (0,77), que confirma que el modelo generaliza correctamente. No obstante, su F1

inferior sugiere que se capturan mejor los diabéticos mediante las fronteras establecidas por el modelo lineal anterior, que a través de la lógica de vecindad local del KNN.



4.3. Árboles de Decisión

Estudio de Hiperparámetros

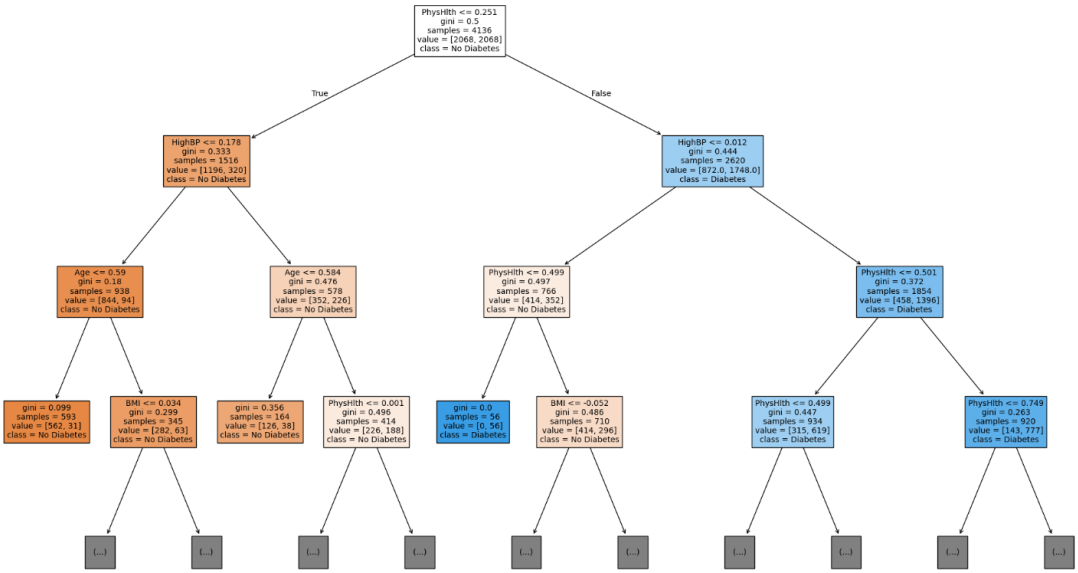
1. **Criterio de División (*criterion*):** Se probaron ***gini*** y **entropy**, para ver cual funcionaba mejor a la hora de separar los nodos.
2. **Profundidad y Poda (*max_depth* y *ccp_alpha*):** La profundidad (rango 4-7) se limitó para evitar ramas demasiado específicas que memoricen ruido. Por otra parte, se probó una poda suave (hasta 0.005) para simplificar el árbol final eliminando ramas que no aporten suficiente ganancia de información.
3. **Restricciones de Hoja (*min_samples_split*/*leaf*):** Se forzó a que los nodos contengan un volumen alto de datos (45 o 90) así como mínimos altos para dividir un nodo (90 o 180). De esta manera garantizamos robustez en cada regla y suavizamos la frontera.
4. **Aleatoriedad (*max_features*):** Se probaron distintas fracciones de variables (0.5 y 0.75) así como libertad al modelo (None), para las diferentes divisiones.

Configuración Óptima

La validación cruzada seleccionó un árbol con profundidad moderada (**max_depth=6**), criterio **Gini** y una poda leve (**ccp_alpha=0.001**). En cuanto a la importancia de las variables, el modelo priorizó tres variables: **PhysHlth (65.5%)**, **HighBP (19.1%)** y **BMI (8.0%)**, que acumulan más del 92% de la capacidad predictiva.

Visualización del Árbol e Interpretación

Solo se muestran los 3 primeros niveles del árbol para que la interpretación de este sea más sencilla, ya que con 6 niveles es complicado leer el contenido de los nodos.



Analizamos los primeros niveles del árbol, donde se consolidan las variables más discriminantes (**PhysHlth** y **HighBP**).

1. **Nodo Raíz:** La decisión comienza por **PhysHlth** (Salud Física). El modelo separa a la población entre quienes reportan varios días de mala salud física y quienes no, estableciendo la salud física como el predictor más importante.
2. **Rama Izquierda (Bajo Riesgo):** Para pacientes con buena salud física, en la mayoría de los casos, el paciente estará sano. El árbol consulta **HighBP** (Hipertensión), edad y otros parámetros.
3. **Rama Derecha (Alto Riesgo):** Para pacientes con mala salud física, **HighBP** vuelve a ser el discriminador. La combinación de "Mala Salud Física + Hipertensión" dispara la probabilidad de diabetes, dirigiendo al paciente a los nodos de mayor riesgo del árbol.

Conclusión: Esta estructura refleja una lógica clínica coherente: el modelo prioriza primero la percepción de salud física, luego verifica la hipertensión y finalmente utiliza el IMC (BMI) para afinar el diagnóstico en los casos dudosos.

4.4. Support Vector Machines (SVM)

Búsqueda de Hiperparámetros (Dos Fases)

1. **Fase I: Kernel Lineal**
    - **Tipo (l1, l2):** Se contrasta la norma estándar L2 frente a L1, con el objetivo de verificar si inducir L1 (forzando coeficientes a cero para selección de variables) beneficia al modelo.
    - **Regularización C (0.1, 1, 10, 100)**: Se exploran desde "márgenes suaves" (C<=1) que toleran errores de clasificación para favorecer la generalización, hasta configuraciones más rígidas (C>=10) que priorizan la exactitud en el entrenamiento.
2. **Fase II: Kernels No Lineales**

- **Kernels:** Probamos con los kernels no lineales ***rbf***, ***poly*** y ***sigmoide***. Estos, permiten que el modelo trace fronteras de decisión curvas o complejas, moviéndose a otras dimensiones distintas a las del problema.
- ***Coeficiente del Kernel (gamma***): Fundamental para los kernels no lineales (especialmente RBF y Polinómico). Define el radio de influencia de cada punto de entrenamiento. Se estudian los valores ***scale*** y ***auto*** que calculan valores óptimos, evitando que el modelo sea demasiado restrictivo o general.
- **Grado (*Degree*) y Coeficiente Independiente (*Coef0*):** Parámetros adicionales ajustados específicamente cuando el kernel seleccionado era el polinómico o sigmoide, permitiendo controlar la complejidad del polinomio (grado 2 o 3) o el desplazamiento de la función, respectivamente.

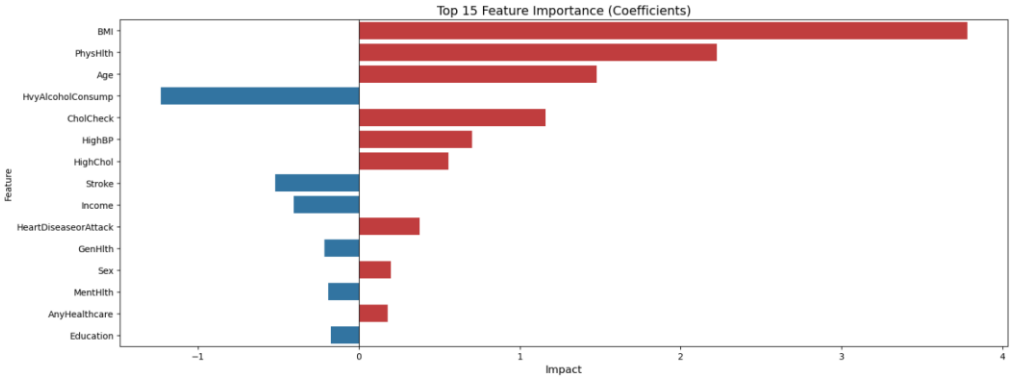
Configuración Óptima

Tras la validación cruzada, el **Kernel Lineal** (C = 10, L1) resultó la opción óptima (F1 = 0.4412). Su superioridad sobre los modelos no lineales confirma que el problema no requiere mayor complejidad geométrica. La clave está en el uso de la norma L1, que es crucial para anular ruido y seleccionar variables, coincidiendo con la Regresión Logística. Por tanto, priorizamos este hiperplano robusto frente a la innecesaria complejidad de los kernels no lineales.

Evaluación de Rendimiento y Generalización

Análisis Train vs. Validation: A diferencia del KNN, el SVM exhibe una notable estabilidad estructural. La consistencia entre las métricas de entrenamiento y validación descarta el sobreajuste. La combinación de un margen medianamente ajustado (C=10) con la penalización L1 ha resultado clave: el modelo logra filtrar el ruido anulando variables irrelevantes, definiendo así una frontera robusta y generalizable sin caer en la memorización.

Interpretación del modelo



Como se observa en la gráfica, el SVM asigna una importancia predominante al **BMI (IMC)**, seguido de la **salud física** y la **edad**. Esto demuestra que la **frontera de decisión** es extremadamente sensible al BMI, por lo que pequeños aumentos en el índice de masa corporal empujan rápidamente al paciente hacia el lado positivo del hiperplano. Además, como pasaba en regresión logística, tenemos el falso “efecto causa” del alto consumo de alcohol (en realidad es una consecuencia).

5. Discusión y conclusiones

5.1 Selección del modelo final

Para la elección del modelo definitivo nos basamos estrictamente en la maximización del **F1-Score**.

Algoritmo	F1-Score (Mean)	Recall (Mean)	Precision (Mean)	Veredicto
Regresión Logística	**0.4416**	0.7594	**0.3114**	**GANADOR**
SVM	0.4412	**0.7674**	0.3096	*Eliminado*
Decision Tree	0.4226	0.7038	0.3035	*Eliminado*
KNN	0.4195	0.7710	0.2883	*Eliminado*

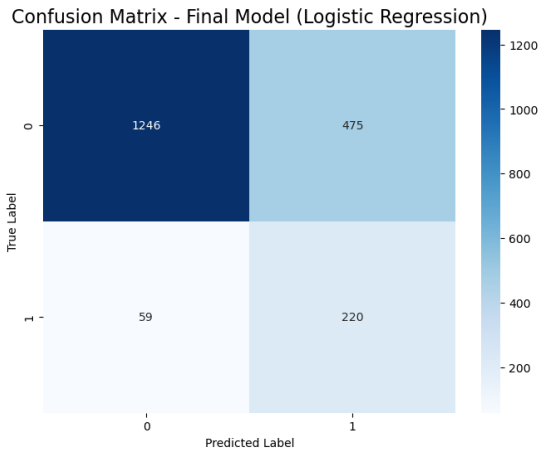
La Regresión Logística (con C=100, y penalización L1) resulta la elegida, con un F1-Score de **0.4416**, superando marginalmente al SVM (0.4412). Aunque el SVM obtuvo un Recall ligeramente superior, la Regresión Logística ofreció la mejor Precisión (31,14%) y por lo tanto el equilibrio entre ambas (F1) es mejor en el modelo de Regresión Logística

La configuración **L1 (Lasso)** con un C alto (100) es reveladora: confirma que el modelo se beneficia de la **selección de variables** (anulando ruido mediante L1) y que prefiere ajustarse fielmente a los datos de entrenamiento (baja regularización), lo que sugiere que la barrera principal es el solapamiento de clases y no el riesgo de sobreajuste.

5.2 Evaluación del modelo

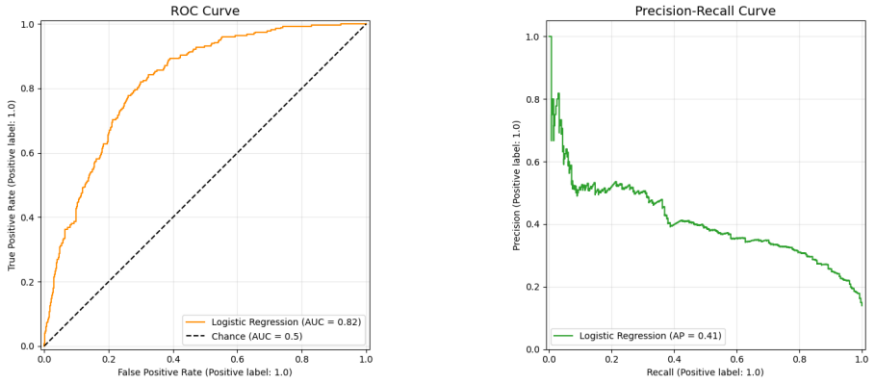
Una vez seleccionado y entrenado el modelo se procede a la evaluación definitiva sobre el conjunto de test (20\% de los datos), el cual se mantuvo aislado desde el inicio.

Accuracy	Recall
0.7330	0.7885
Precision	**F1**
0.3165	0.4517



La matriz de confusión nos muestra que el modelo funciona eficazmente como **herramienta de cribado**. A pesar de que la métrica elegida para seleccionar el modelo, como se explicó al inicio, el recall es lo que más nos interesa, logrando detectar **220 de los 279 casos positivos** (casi el 79%). De esta manera conseguimos minimizar los Falsos Negativos (riesgo real), aceptando como

contrapartida una mayor tasa de Falsos Positivos (475 casos) y una Precisión del 31.6%. A niveles prácticos, esto implica filtrar a la mayoría de los pacientes sanos, asumiendo una segunda revisión para los casos falsos alertados. Se muestran a continuación las gráficas de la *curva ROC* y de la relación *Precision-Recall* que ofrece el modelo:



Estas ases gráficas validan la robustez del modelo. El **AUC de 0.82** confirma una sólida capacidad para distinguir entre clases independientemente del umbral. Por su parte, la curva *Precision-Recall* ilustra el compromiso necesario: para alcanzar la sensibilidad deseada (>75%), es inevitable un descenso en la precisión, validando la estrategia de *Hybrid Resampling* empleada.

5.3 Conclusiones

El modelo valida su aptitud como sistema de soporte clínico, priorizando con éxito la detección de riesgo y demostrando una sólida generalización (AUC 0.82). Se propone como posible mejora, **aumentar el volumen de datos de la clase minoritaria** para mitigar el desbalanceo original y **explorar arquitecturas no lineales** (como Random Forest, *Gradient Boosting*) que permitan refinar la precisión y reducir los falsos positivos sin comprometer la sensibilidad alcanzada.