I302 - Aprendizaje Automático y Aprendizaje Profundo

Trabajo Práctico 2: Clasificación y Ensemble Learning

Ilan Nomberg
Ingeniería en Inteligencia Artificial

1. Diagnóstico de Cáncer de Mama

Resumen

En este trabajo se desarrolló un sistema de clasificación binaria para predecir si una célula presenta características compatibles con un diagnóstico médico, a partir de datos morfológicos y bioquímicos. Para ello, se implementó desde cero un modelo de regresión logística con regularización L2. Se exploraron distintas técnicas de preprocesamiento, normalización, imputación de valores faltantes y re-balanceo de clases. El modelo fue entrenado y validado en múltiples escenarios, incluyendo datos balanceados y desbalanceados. Los resultados muestran que la técnica de undersampling permitió alcanzar el mejor compromiso entre precisión y recall, con un F1-score superior al resto de las variantes evaluadas.

1.1. Introducción

El objetivo de esta parte del trabajo fue desarrollar un clasificador binario que permita predecir si una célula presenta características compatibles con un diagnóstico médico, a partir de un conjunto de variables morfológicas, bioquímicas y genéticas obtenidas de muestras celulares. La tarea se basa en un conjunto de datos recopilado por laboratorios biomédicos, e incluye tanto variables numéricas como categóricas, tales como el tamaño de la célula, la densidad nuclear, la tasa de mitosis, el tipo celular o la presencia de mutaciones genéticas.

El algoritmo desarrollado toma como entrada un vector de características extraídas de cada célula y estima la probabilidad de que la muestra corresponda a una célula con características anómalas. Para ello, se implementó desde cero un modelo de regresión logística binaria con regularización L2. Esta elección se basó en la capacidad del modelo para manejar problemas de clasificación binaria, interpretabilidad y eficiencia computacional.

El problema se enmarca dentro del ámbito del aprendizaje supervisado, utilizando etiquetas binarias (‘0‘: célula normal, ‘1‘: célula anómala) y un enfoque experimental que evalúa el rendimiento del modelo bajo distintos esquemas de balanceo de clases.

1.2. Métodos

El modelo implementado fue una regresión logística binaria con regularización L2. Este modelo busca estimar la probabilidad de que una instancia pertenezca a la clase positiva utilizando la función sigmoidea:

$$\hat{y} = \sigma(z) = \frac{1}{1 + e^{-z}}, \quad \text{con } z = \mathbf{x}^\top \mathbf{w} + b$$

La función de pérdida utilizada fue la entropía cruzada penalizada, con regularización L2:

$$\mathcal{L}(\mathbf{w}, b) = -\frac{1}{m} \sum_{i=1}^{m} \left[y^{(i)} \log(\hat{y}^{(i)}) + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)}) \right] + \frac{\lambda}{2m} \|\mathbf{w}\|^2$$

Para la selección del hiperparámetro de regularización λ, se implementó validación cruzada estratificada de $k = 10$ folds. En este esquema, el conjunto de entrenamiento se divide aleatoriamente en k subconjuntos del mismo tamaño. Luego, se repite el siguiente proceso k veces:

1. Se retiene uno de los k subconjuntos como conjunto de validación.

2. Se entrena el modelo sobre los $k-1$ subconjuntos restantes.

3. Se evalúa el modelo en el fold de validación y se guardan las predicciones y valores reales.

Una vez recorridos los k folds, se concatenan todas las predicciones $\hat{y}^{(1)}, \ldots, \hat{y}^{(k)}$ y los valores verdaderos correspondientes $y^{(1)}, \ldots, y^{(k)}$. Se calcula una única métrica global (en nuestro caso, el F1-score) sobre todas las muestras validadas:

$$\mathrm{F1}_\lambda = F_1\left(\bigcup_{i=1}^{k} y^{(i)}, \bigcup_{i=1}^{k} \hat{y}^{(i)}\right)$$

Finalmente, se selecciona el mejor valor de regularización como:

$$\lambda^* = \arg\max_{\lambda} \ \mathrm{F1}_\lambda$$

Preprocesamiento. Los valores faltantes fueron imputados mediante dos estrategias:

- Imputación básica: por media (variables numéricas) o moda (categóricas), definida exclusivamente en el conjunto de entrenamiento.
- Imputación por KNN: se identificaron $k = 5$ vecinos más cercanos a cada muestra incompleta, utilizando distancia euclídea sobre las variables numéricas normalizadas, y se completó el valor faltante con la media (para numéricas) o moda (para categóricas) entre esos vecinos.

Formalmente, para una muestra $\mathbf{x}$ incompleta, y un conjunto de referencia $\{\mathbf{x}_j\}$, se define:

$$\mathrm{dist}(\mathbf{x}, \mathbf{x}_j) = \sqrt{\sum_{l \in \mathrm{valid}} (x_l - x_{j,l})^2}$$

donde "valid" son los índices de las características no nulas en $\mathbf{x}$, y luego se imputan los valores de acuerdo a los k vecinos más cercanos.

Métricas utilizadas. Se evaluó el rendimiento con métricas estándar para clasificación binaria, resumidas en la siguiente tabla:

Fórmula	**Accuracy**	**Precision**	**Recall**	**F1-score**
Expresión	$\dfrac{TP+TN}{TP+TN+FP+FN}$	$\dfrac{TP}{TP+FP}$	$\dfrac{TP}{TP+FN}$	$2 \cdot \dfrac{\mathrm{Precision} \cdot \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}}$

Cuadro 1: Métricas de evaluación utilizadas.

También se utilizaron las curvas ROC y PR, y sus respectivas áreas bajo la curva (AUC-ROC y AUC-PR), para evaluar el modelo en distintos umbrales de decisión.

Técnicas de rebalanceo. Se implementaron distintas estrategias para tratar el desbalanceo de clases, todas aplicadas antes o durante el entrenamiento del modelo:

- **Undersampling**: se seleccionan al azar n instancias de la clase mayoritaria, donde n es la cantidad de muestras de la clase minoritaria. Esto produce un dataset balanceado de tamaño $2n$. Sea C_0 y C_1 las clases, y $|C_0| > |C_1|$, entonces se retiene:

$$D' = C_1 \cup \text{Sample}(C_0,\, |C_1|)$$

- **Oversampling (duplicación)**: se duplican aleatoriamente muestras de la clase minoritaria hasta igualar la cantidad de la clase mayoritaria:

$$D' = C_0 \cup \text{Sample}(C_1,\, |C_0|, \text{con reemplazo})$$

- **SMOTE (Synthetic Minority Over-sampling Technique)**: se generan instancias sintéticas de la clase minoritaria mediante interpolación entre cada punto x_i y uno de sus k vecinos más cercanos x_j:

$$x_{\text{new}} = x_i + \delta \cdot (x_j - x_i), \quad \delta \sim \mathcal{U}(0, 1)$$

  Esto permite generar nuevas muestras en la vecindad de la clase minoritaria sin duplicarlas exactamente.

- **Cost Re-weighting**: se modifica la función de pérdida agregando un peso distinto a los errores según la clase. Si π_1 y π_0 son las proporciones de la clase 1 y 0 respectivamente, se pondera el error de la clase minoritaria con un factor:

$$\alpha = \frac{\pi_0}{\pi_1}$$

  Y se redefine la pérdida como:

$$\mathcal{L}_{\text{weighted}} = -\frac{1}{m} \sum_{i=1}^{m} \alpha_i \cdot \left[y^{(i)} \log(\hat{y}^{(i)}) + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)}) \right]$$

  donde $\alpha_i = \alpha$ si $y^{(i)}$ pertenece a la clase minoritaria, y 1 en caso contrario.

1.3. Resultados

Manejo de valores faltantes y outliers

Durante el análisis exploratorio se detectó la presencia de valores faltantes y valores atípicos (outliers) en múltiples variables numéricas. Para abordar esta problemática, se definieron intervalos válidos para cada variable utilizando la extensión *Data Wrangler* [1], una herramienta de visualización y preprocesamiento interactivo que permite detectar regiones sospechosas de datos no confiables.

Estos intervalos fueron preferidos por sobre enfoques tradicionales como el análisis de cuantiles (por ejemplo, el criterio de $[Q_1 - 1{,}5 \cdot IQR,\ Q_3 + 1{,}5 \cdot IQR]$) y técnicas estadísticas multivariadas como *Multiple Discriminant Analysis* (MDA), ya que ofrecieron mejores resultados empíricos al conservar estructura de clase y estabilidad en métricas posteriores.

Los valores fuera de los rangos plausibles definidos fueron reemplazados por `NaN`, y tratados como faltantes.

El tratamiento de los valores faltantes se realizó en dos etapas:

- En una primera instancia, se imputaron con la media (para variables numéricas) y la moda (para categóricas), utilizando únicamente estadísticas del conjunto de entrenamiento.
- Luego, se aplicó imputación mediante **K-vecinos más cercanos** ($k = 5$), utilizando distancia euclídea sobre variables numéricas previamente normalizadas. Esta técnica permitió completar de forma más contextual los valores faltantes en observaciones con múltiples ausencias.

Este enfoque híbrido permitió reducir completamente la cantidad de valores faltantes sin descartar muestras ni introducir sesgos por duplicación o interpolación lineal. El conjunto final de datos quedó completo, estandarizado y estructuralmente íntegro, listo para el entrenamiento de modelos. Esto puede visualizarse en la Figura 1, donde se muestra un resumen gráfico del conjunto de datos luego del preprocesamiento.

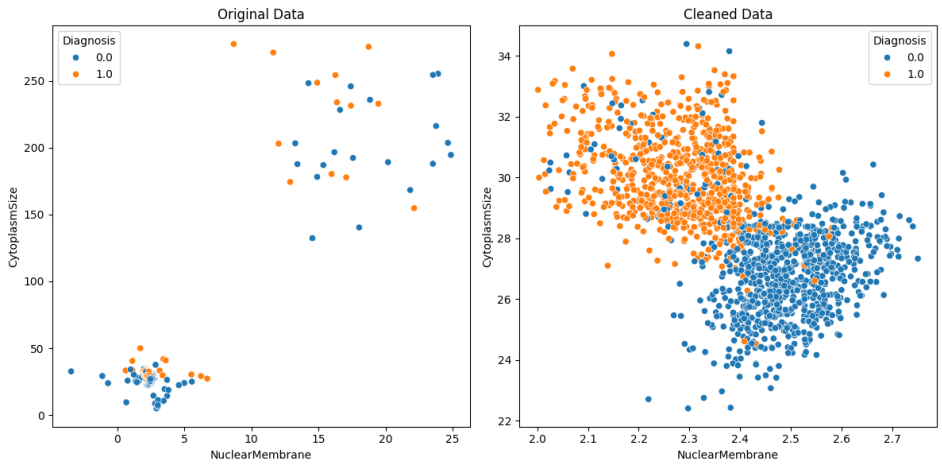

Figura 1: Distribución y completitud del dataset luego del tratamiento de outliers y valores faltantes.

Entrenamiento y selección del hiperparámetro λ

Una vez completado el preprocesamiento del conjunto de datos, se procedió al entrenamiento del modelo de regresión logística con regularización L2. El objetivo era encontrar el valor óptimo del hiperparámetro λ que regula la penalización aplicada a los coeficientes del modelo y permite controlar el sobreajuste.

Para ello, se exploraron dos enfoques complementarios:

- **Barrido de valores**: se entrenó el modelo para un rango logarítmico de valores de λ, utilizando una partición fija (80 % entrenamiento, 20 % validación). Se midió el F1-score en el conjunto de validación para cada valor de λ.

- **Validación cruzada ($k = 10$ folds)**: se aplicó validación cruzada estratificada repitiendo el entrenamiento y evaluación en diferentes particiones, concatenando las predicciones para obtener una única estimación global del F1-score por cada valor de λ (ver sección Métodos).

Ambos métodos coincidieron en una región cercana del espacio de búsqueda. Sin embargo, el valor óptimo determinado por el barrido simple arrojó una leve mejora en el rendimiento final del modelo en el conjunto de validación. Por esta razón, se optó por continuar utilizando ese valor de λ en los experimentos posteriores.

La Figura 2 muestra el comportamiento del F1-score a medida que varía el valor de λ.

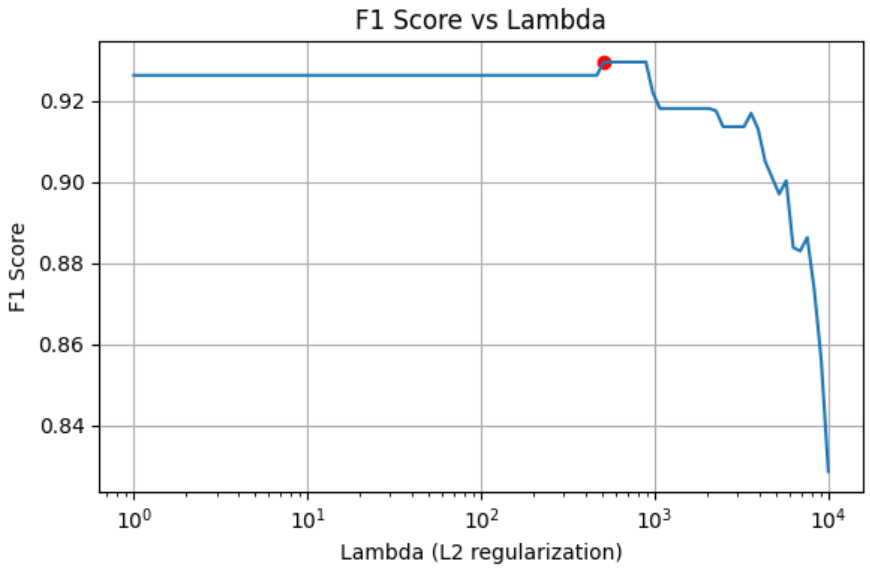

Figura 2: F1-score sobre el conjunto de validación para distintos valores de λ explorados mediante barrido logarítmico.

Evaluación sobre el conjunto de test

Una vez identificado el valor óptimo del hiperparámetro λ a través del procedimiento de barrido y validación cruzada, se procedió a entrenar el modelo definitivo sobre el **conjunto completo de desarrollo balanceado**. A continuación, se evaluó su desempeño sobre el **conjunto de test balanceado**, que permaneció completamente independiente durante el proceso de ajuste.

Las métricas obtenidas se muestran en la Tabla 2. El modelo logró un rendimiento satisfactorio, manteniendo la coherencia respecto a lo observado en validación. Las curvas ROC y PR correspondientes, junto a la matriz de confusión se presentan en la Figura 3.

Accuracy	Precision	Recall	F1-Score
0.9622	0.9419	0.9759	0.9586

Cuadro 2: Desempeño del modelo de regresión logística sobre el conjunto de test balanceado.

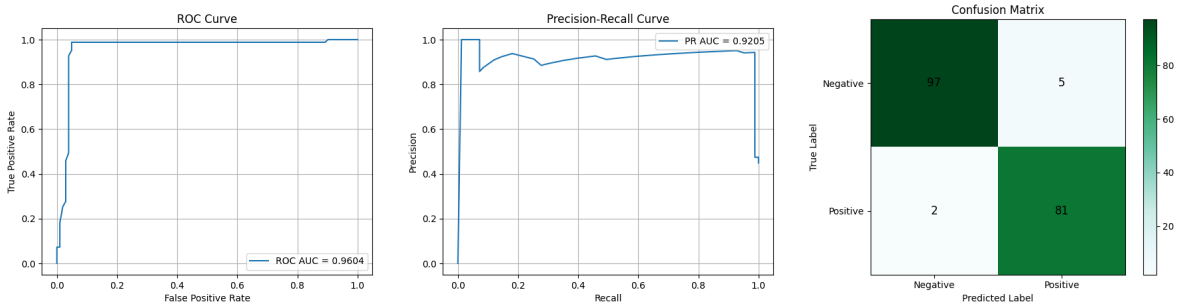

Figura 3: Curvas ROC (izquierda), Precision-Recall (centro) y Matriz de Confusión (derecha) del modelo evaluado sobre el conjunto de test.

Rebalanceo de datos en conjunto desbalanceado

Para evaluar la robustez del modelo frente al desbalance de clases, se repitió el entrenamiento utilizando el conjunto de desarrollo **desbalanceado**. En este caso, la proporción entre clases estaba significativamente sesgada, lo que perjudica la capacidad del modelo de detectar correctamente la clase minoritaria.

Se probaron las cinco estrategias de rebalanceo previamente mencionadas, incluyendo el modelo sin rebalanceo.

Evaluación final sobre el conjunto de test

Luego de seleccionar los hiperparámetros óptimos para cada estrategia de rebalanceo mediante validación cruzada y barrido de λ, se entrenaron modelos finales utilizando los conjuntos de entrenamiento correspondientes. Estos modelos fueron evaluados sobre el conjunto de test, que se mantuvo completamente separado durante el proceso de ajuste.

La Tabla 3 presenta los valores obtenidos de las métricas principales. Se observa que todos los métodos alcanzan un **accuracy** idéntico de 0.956, aunque difieren levemente en términos de **precision**, **AUC-PR** y **AUC-ROC**. En particular, la estrategia de *undersampling* logró el mayor F1-score y AUC-PR, mostrando un buen compromiso entre sensibilidad y precisión.

Modelo	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**AUC-ROC**	**AUC-PR**
Sin rebalanceo	0.956	0.868	0.971	0.917	0.971	0.819
Undersampling	0.956	0.850	1.000	0.919	0.970	0.825
Oversampling duplicado	0.956	0.850	1.000	0.919	0.968	0.797
Oversampling SMOTE	0.956	0.850	1.000	0.919	0.970	0.820
Cost re-weighting	0.956	0.850	1.000	0.919	0.969	0.815

Cuadro 3: Métricas de desempeño final sobre el conjunto de test para distintos métodos de rebalanceo, utilizando los hiperparámetros previamente seleccionados.

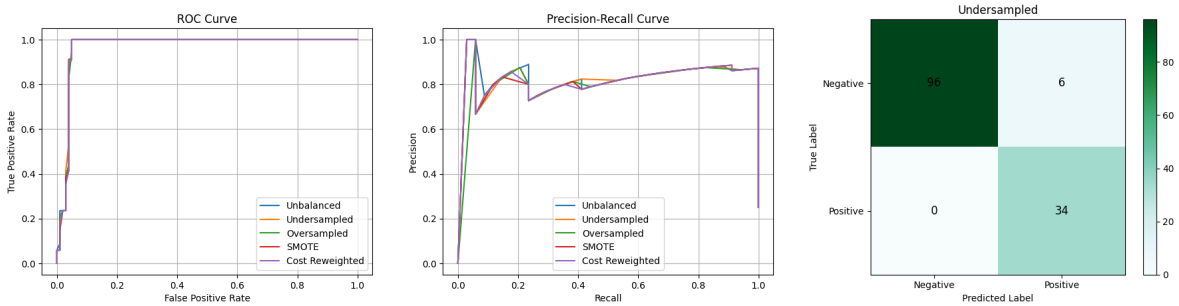

Figura 4: Comparación de métricas visuales para las estrategias de rebalanceo evaluadas sobre el conjunto de validación desbalanceado: curva ROC (izquierda), curva Precision-Recall (centro) y matriz de confusión del mejor modelo (derecha).

Un aspecto llamativo es que el modelo entrenado **sin rebalanceo** obtuvo resultados muy similares, e incluso superiores en precisión, respecto a las técnicas específicamente diseñadas para combatir el desbalance. Esta observación se explica por la alta separabilidad inherente de las clases en el espacio de atributos.

Algunas variables presentan una distribución claramente bimodal entre clases, lo que facilita que el clasificador pueda discriminar correctamente incluso en presencia de un fuerte desbalance. Esta estructura puede observarse en la Figura 8, incluida en el Apéndice, donde se visualiza la distribución de dos variables numéricas coloreadas por clase.

2. Predicción de Rendimiento de Jugadores de Basketball

Resumen

En esta segunda parte del trabajo se abordó un problema de clasificación multiclase, cuyo objetivo fue predecir la categoría de rendimiento (*WAR class*) de un jugador de basketball a partir de sus estadísticas individuales. Se trabajó con un conjunto de datos reales compuesto por múltiples features numéricas y categóricas. Se implementaron y evaluaron distintos clasificadores, incluyendo regresión logística multiclase, LDA y Random Forest.

El conjunto de datos fue preprocesado mediante imputación por KNN y normalización gaussiana. Se utilizaron técnicas de validación cruzada y búsqueda de hiperparámetros para optimizar el rendimiento. Finalmente, se compararon los modelos utilizando métricas multiclase como macro-F1, precisión y recall. El modelo de Random Forest obtuvo el mejor desempeño global, superando a los clasificadores lineales tanto en precisión como en robustez.

2.1. Introducción

El objetivo de esta parte del trabajo fue desarrollar un modelo de clasificación multiclase para predecir el rendimiento de jugadores de basketball, a partir de un conjunto reducido pero representativo de estadísticas avanzadas. La variable objetivo es `WAR class` (Wins Above Replacement class), que segmenta a los jugadores en tres categorías según su impacto en el equipo: rendimiento negativo, nulo o positivo.

Cada instancia del dataset representa el desempeño de un jugador durante una temporada específica, y está caracterizada por las siguientes variables:

Variable	Descripción
`poss`	Número de posesiones en la temporada
`mp`	Minutos jugados
`off_def`	Impacto ofensivo/defensivo global
`pace_impact`	Influencia del jugador en el ritmo de juego

Cuadro 4: Variables seleccionadas y su descripción.

Estas métricas combinan aspectos individuales y contextuales del juego, proporcionando una visión condensada del valor del jugador. Dado que la variable objetivo es discreta y toma tres posibles valores, el problema se abordó como una tarea de clasificación supervisada multiclase.

La presencia de desbalance entre clases y la baja dimensionalidad del dataset plantean desafíos tanto para el modelado como para la evaluación. Por ello, se consideraron múltiples arquitecturas, desde modelos lineales hasta métodos basados en árboles, y se utilizaron métricas específicas para clasificación multiclase como el F1-score macro, precisión y recall por clase.

2.2. Métodos

Además del modelo de regresión logística multiclase previamente extendido, se incorporaron dos nuevas arquitecturas para abordar el problema de clasificación multiclase:

Análisis Discriminante Lineal (LDA). Este método asume que las clases siguen distribuciones normales con una matriz de covarianza común, y busca encontrar combinaciones lineales de las variables que maximicen la separabilidad entre clases. Para una observación $\mathbf{x}$, el clasificador LDA calcula una función discriminante por clase:

$$\delta_k(\mathbf{x}) = \mathbf{x}^\top \Sigma^{-1} \mu_k - \frac{1}{2} \mu_k^\top \Sigma^{-1} \mu_k + \log \pi_k$$

donde μ_k es la media de la clase k, Σ es la matriz de covarianza común estimada (pooled), y π_k es la proporción de muestras de la clase k. La predicción se asigna a la clase con mayor $\delta_k(\mathbf{x})$.

Random Forest. Se implementó un ensemble de árboles de decisión entrenados sobre distintos subconjuntos bootstrap del conjunto de entrenamiento. Cada árbol fue construido utilizando la entropía como criterio de división, y se combinaron las predicciones por votación mayoritaria.

El clasificador final está compuesto por T árboles $\{h_t\}_{t=1}^{T}$, y la predicción se obtiene como:

$$\hat{y} = \text{mode}\left(h_1(\mathbf{x}), h_2(\mathbf{x}), \ldots, h_T(\mathbf{x})\right)$$

Este enfoque introduce aleatoriedad tanto en la selección de datos como en la selección de umbrales de división, lo que reduce el sobreajuste y mejora la generalización.

Ambos modelos fueron evaluados sobre los mismos datos preprocesados que los modelos lineales. En todos los casos, las predicciones probabilísticas fueron utilizadas para computar métricas como ROC y PR por clase, bajo el esquema *one-vs-rest*.

2.3. Resultados

Previo al entrenamiento de los modelos, se realizó un análisis exploratorio del conjunto de datos utilizando nuevamente la extensión *Data Wrangler* [1]. A partir de esta herramienta, se verificó que no existían datos faltantes ni filas duplicadas, lo cual simplificó la etapa inicial de limpieza.

Sin embargo, sí se identificaron valores fuera de los rangos coherentes en varias de las variables numéricas, en particular en `poss` y `mp`, las cuales tenian valores negativos (imposibles en este contexto). Estos valores atípicos no eran frecuentes, pero podían comprometer el desempeño de modelos lineales sensibles a escalas y dispersión.

Al igual que en la primera parte del trabajo, se definieron rangos válidos para cada variable basados en la visualización y comprensión de la distribución. Los valores fuera de esos límites fueron marcados como faltantes. Luego, se imputaron utilizando una estrategia por K-vecinos más cercanos ($k = 5$), considerando únicamente las instancias válidas como referencia. Esto permitió preservar la estructura del conjunto de datos sin eliminar observaciones ni introducir sesgos arbitrarios.

Durante el análisis exploratorio también se evaluó la relevancia de cada feature respecto al target `WAR_class` utilizando la métrica de **información mutua**. Esta métrica permite cuantificar cuánta información aporta cada variable predictora sobre la variable objetivo, sin asumir relaciones lineales.

El análisis reveló que la columna `WAR_total` presentaba una altísima dependencia con la variable objetivo, lo cual era esperable, ya que `WAR_class` es una discretización directa de esa misma variable continua. Incluirla como feature implicaría una fuga directa de información (*data leakage*), ya que el modelo estaría aprendiendo a reproducir una transformación conocida del target.

Por esta razón, `WAR_total` fue excluida del conjunto de datos antes del entrenamiento de los modelos.

Una vez completado el preprocesamiento, se entrenaron los tres modelos sobre el conjunto de desarrollo y se evaluaron sobre el conjunto de test. Para cada modelo se utilizó el mejor conjunto de hiperparámetros encontrado previamente:

- En el caso de la regresión logística multiclase, se utilizó el valor óptimo de regularización λ previamente seleccionado mediante validación cruzada/barrido.
- Para el modelo de Random Forest, se realizó una búsqueda en malla (*grid search*) sobre una grilla de hiperparámetros, evaluando diferentes combinaciones de número de árboles, profundidad máxima y tamaño mínimo de los splits. La combinación que maximizó el F1 macro sobre el conjunto de validación fue utilizada para la evaluación final.
- El modelo LDA no posee hiperparámetros, por lo que se utilizó directamente sobre los datos normalizados.

A continuación se presentan los resultados obtenidos por cada modelo:

Regresión Logística Multiclase

Accuracy	Precision (macro)	Recall (macro)	F1 Score (macro)
0.8850	0.8882	0.8934	0.8852

Cuadro 5: Métricas de desempeño del modelo de regresión logística multiclase sobre el conjunto de test.

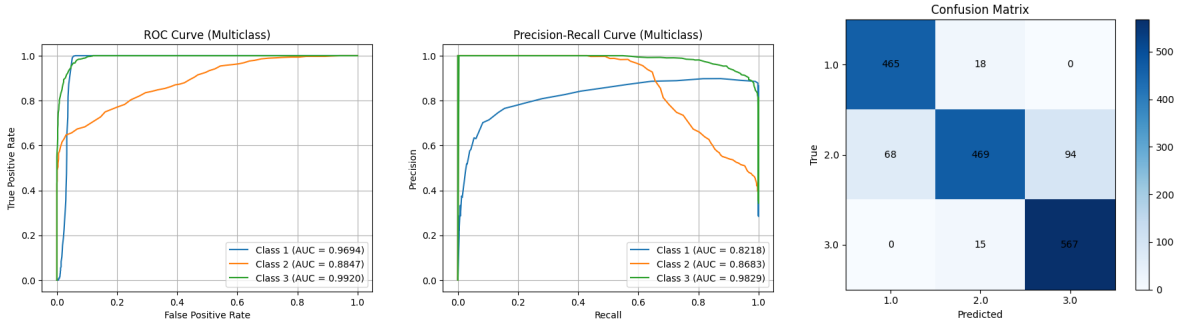

Figura 5: Curva ROC (izquierda), curva PR (centro) y matriz de confusión (derecha) del modelo de regresión logística multiclase.

Análisis Discriminante Lineal (LDA)

Accuracy	Precision (macro)	Recall (macro)	F1 Score (macro)
0.9057	0.9114	0.9150	0.9047

Cuadro 6: Métricas de desempeño del modelo LDA sobre el conjunto de test.

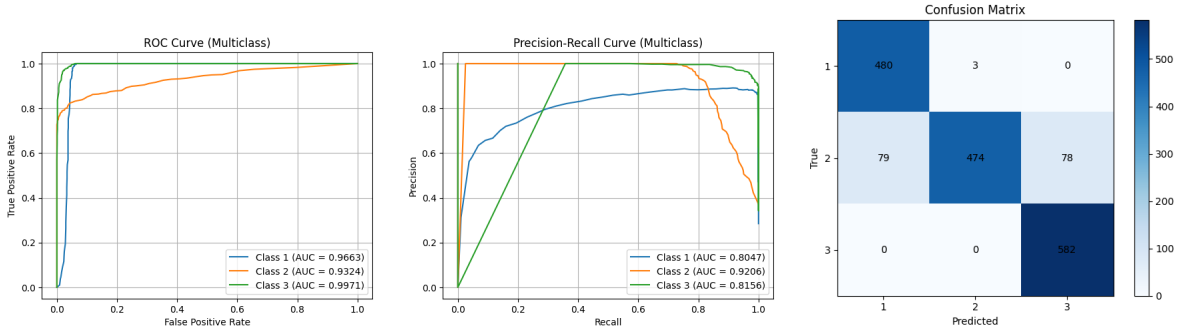

Figura 6: Curva ROC (izquierda), curva PR (centro) y matriz de confusión (derecha) del modelo LDA.

Random Forest

Accuracy	Precision (macro)	Recall (macro)	F1 Score (macro)
0.9587	0.9583	0.9611	0.9595

Cuadro 7: Métricas de desempeño del modelo Random Forest sobre el conjunto de test.

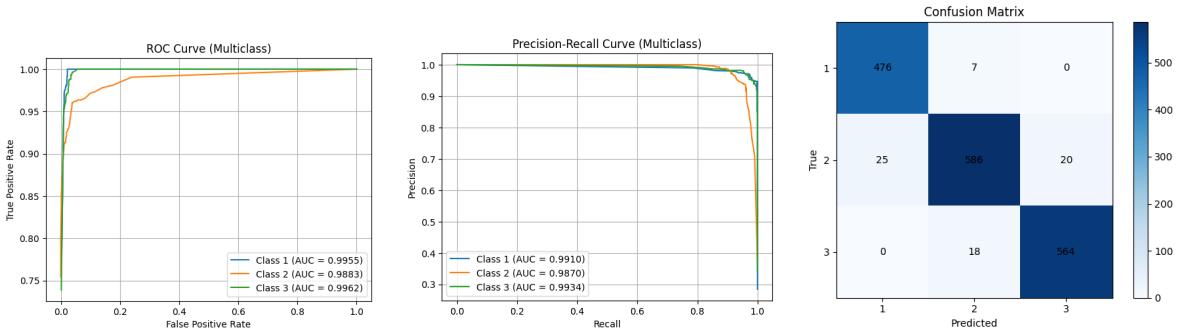

Figura 7: Curva ROC (izquierda), curva PR (centro) y matriz de confusión (derecha) del modelo Random Forest.

Conclusión

Entre los modelos evaluados, el Random Forest demostró ser el más efectivo para la tarea de clasificación del rendimiento de jugadores de basketball. Obtuvo el mayor F1-score macro, precisión y recall, superando tanto a los modelos lineales como al LDA. Su capacidad para capturar relaciones no lineales y manejar interacciones entre variables resultó clave en este escenario, a pesar de contar con un conjunto de features reducido.

Referencias

[1] Microsoft. Data wrangler extension for exploratory data cleaning, 2023. Disponible en: `https://marketplace.visualstudio.com/items?itemName=ms-toolsai.datawrangler`.

A. Apéndice

A.1. Distribución de variables por clase para datos desbalanceados

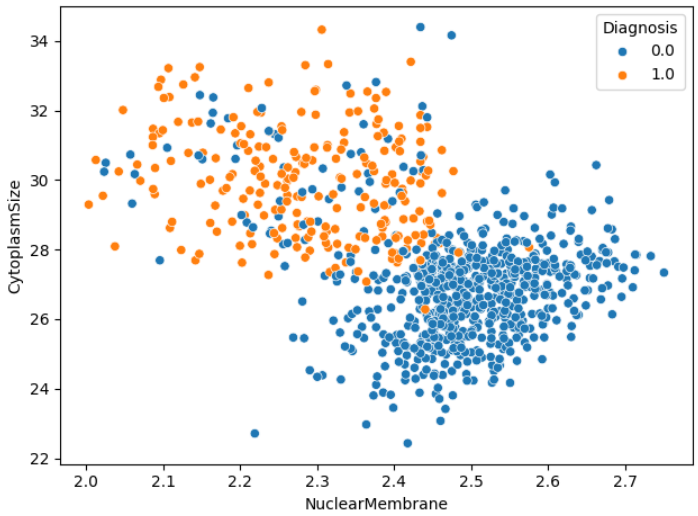

Figura 8: Visualización de la separación de clases en dos dimensiones. Las clases presentan fronteras bien definidas, lo que explica el alto rendimiento del modelo incluso sin rebalanceo.