I302 - Aprendizaje Automático y Aprendizaje Profundo

Trabajo Práctico 2: Clasificación y Ensemble Learning

Agustín Ezequiel Amblard

15 de abril de 2025

Ingeniería en Inteligencia Artificial

1. Diagnóstico de Cáncer de Mama

Resumen

En este trabajo se buscó modelar un modelo de Regresión logística con regularización L2 para predecir de la forma mas efectiva posible si un tumor presente es benigno o maligno a partir de una base de datos con multiples características de cada paciente. Para llevarlo a cabo, se realizó un análisis de la base de datos proporcionada y se limpió dicha base. Hecho esto, se entrenaron diversos modelos de Regresión Logística, cada uno con una técnica de rebalanceo diferente y se analizó el rendimiento de cada modelo mediante métricas de performance. El modelo implementado performó acorde a lo que se esperaban y las métricas indicaron una alta eficacia en la mayoría de los casos evaluados. Se llega a la conclusión de que el mejor modelo para el contexto de diagnóstico médico es el de Regresión Logistica con regularizacion L2, con rebalanceo mediante Undersampling con valor λ de 0.

1.1. Introducción

El cáncer de mama es una de las principales causas de mortalidad en mujeres a nivel mundial. Un diagnóstico temprano y preciso es fundamental para aumentar las probabilidades de recuperación. En este trabajo, se aborda el problema del diagnóstico de cáncer de mama como una tarea de clasificación binaria en el marco del aprendizaje supervisado.

La entrada del algoritmo está compuesta por un vector de atributos numéricos que representan características morfológicas de células mamarias extraídas de imágenes histopatológicas. Estas características incluyen, por ejemplo, el tamaño de la celula, el tamaño del citoplasma de las células, la densidad del núcleo celular, el tipo celular, el indice de vascularización celular y otros indicadores derivados de imágenes digitales. La salida del algoritmo es una predicción del tipo de tumor: `benigno` o `maligno`.

Para la resolución del problema, se utilizó un modelo de **regresión logística binaria con regularización L2**, implementado desde cero utilizando exclusivamente `NumPy`.

Se realizó un análisis exploratorio del conjunto de datos `cell_diagnosis_balanced_dev.csv`, que incluye la verificación de valores faltantes, identificación de outliers, y la visualización de la distribución de atributos. Además, se evaluó la correlación entre las variables para entender su relación con la clase objetivo.

Para entrenar y validar el modelo, se dividió el conjunto de desarrollo en subconjuntos de entrenamiento (80 %) y validación (20 %). El hiperparámetro de regularización λ fue ajustado con el objetivo de maximizar el *F-score* en el conjunto de validación.

El modelo fue evaluado utilizando distintas métricas. Posteriormente, se evaluó el desempeño final sobre el conjunto de test `cell_diagnosis_balanced_test.csv`.

Por último, se analizaron estrategias de re-balanceo de clases mediante la modificación de la función de pérdida con pesos inversamente proporcionales a la frecuencia de cada clase, así como el uso de técnicas de sobremuestreo y submuestreo. Se discutieron sus impactos sobre las métricas y se propusieron recomendaciones para la elección de umbrales de decisión en casos con fuerte desbalanceo de clases.

1.2. Métodos

1.2.1. Modelos de clasificación

Para abordar el problema de diagnóstico de cáncer de mama se implementó un modelo de **regresión logística binaria con regularización L2**, utilizando exclusivamente la librería `NumPy`. La función de hipótesis del modelo se define como:

$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$

donde θ es el vector de parámetros del modelo y x es el vector de características de entrada. La función de costo utilizada fue la entropía cruzada regularizada:

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\log(h_\theta(x^{(i)})) + (1-y^{(i)})\log(1-h_\theta(x^{(i)}))\right] + \frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2$$

donde m es la cantidad de ejemplos, n la cantidad de características, y λ el coeficiente de regularización.

Para ajustar el hiperparametro de regularización λ, se utilizo como metrica de evaluación el F-score: cuanto más alto, mejor era ese valor de lambda para el modelo. Esto lo obtuvimos mediante validación cruzada. La *validación cruzada* es una técnica para evaluar el rendimiento de un modelo dividiendo los datos en varios subconjuntos (folds), entrenando el modelo en algunos y probándolo en los restantes, repitiendo este proceso para obtener una estimación más robusta de su desempeño. Utilizamos 5 folds para cada set, y se evaluaron 10 valores de λ diferentes.

1.2.2. Limpieza del dataset

El conjunto de datos empleado para el entrenamiento fue `cell_diagnosis_balanced_dev.csv`, el cual contiene una distribución balanceada entre muestras benignas y malignas. El conjunto fue dividido en un 80 % para entrenamiento y un 20 % para validación. Posteriormente, el modelo entrenado fue evaluado sobre el conjunto de test `cell_diagnosis_balanced_test.csv`.

Se realizó un preprocesamiento que incluyó:

- **Reemplazo de variables textuales a numéricas:** La base de datos poseía algunas columnas de características, como CellType o GeneticMutation, que tenían sus datos escritos y no numéricos. Esto dificultaba el uso de modelos numericos, como KNN o la Regresión Logistica, entonces se reemplazaron los valores mediante una categorización. Por ejemplo, para la característica de GeneticMutation, se reemplazaron los valores Present por 1 y los valores Absent por 0.

- **Análisis de outliers y datos mal medidos:** Se inspeccionaron valores atípicos en la base de datos para eliminar ruido y poder entrenar un mejor modelo. Primero se extrayeron los datos que se consideraron mal medidos. Para esto, se analizó de las características de cada feature como debían de ser los datos y en base a eso se borraron los datos que estaban fuera de los rangos de cada feature. Para el caso de los outliers, se eliminaron de la base de datos mediante el uso de IQR(Interquartile Range). El **rango intercuartílico (IQR)** es una medida robusta de dispersión que se utiliza comúnmente para detectar valores atípicos (outliers) en un conjunto de datos. Se define como la diferencia entre el tercer cuartil (Q_3) y el primer cuartil (Q_1):

  $$IQR = Q_3 - Q_1$$

  Un valor se considera un *outlier* si se encuentra fuera del rango:

  $$[x_{\min}, x_{\max}] = [Q_1 - 1{,}5 \cdot IQR,\ Q_3 + 1{,}5 \cdot IQR]$$

  Cualquier dato fuera de ese intervalo se puede clasificar como atípico y, dependiendo del análisis, ser eliminado o tratado especialmente. Para este caso se utilizó un intervalo de entre el 25 y el 75 porciento de los datos.

- **Completar datos faltantes:** Una vez que se borraron todos los datos no deseados, se completaron las celdas vacías mediante KNN(k-nearest neighbours). Este algoritmo es un método de clasificación supervisado que asigna a una nueva muestra la clase más común entre sus k vecinos más cercanos en el conjunto de entrenamiento. La cercanía se mide generalmente con la distancia euclídea:

$$d(x, x') = \sqrt{\sum_{j=1}^{n} (x_j - x'_j)^2}$$

Dado un valor de k, se seleccionan los k ejemplos más cercanos a la muestra y se vota por la clase mayoritaria. Es un método no paramétrico, simple de implementar y efectivo para datos bien distribuidos. En este caso, se utilizo un valor de k igual a 5.

- **Normalización de características:** Se aplicó normalización estándar (z-score) sobre cada atributo, es decir:

$$x'_j = \frac{x_j - \mu_j}{\sigma_j}$$

donde μ_j y σ_j representan la media y el desvío estándar de la característica j sobre el conjunto de entrenamiento. Cabe resaltar que el set de entrenamiento, validación y de evaluación están normalizados con la media y el desvío estandar de el set de entrenamiento.

1.2.3. Métricas de performance

Las métricas utilizadas para evaluar el desempeño del modelo fueron:

- **Accuracy**: indica qué fracción de las predicciones fueron correctas respecto al total.

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

- **Precision**: indica cuántas de las predicciones que se realizaron como positivas realmente lo son.

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

- **Recall**: indica cuántos de los positivos reales los predijo correctamente.

$$\text{Recall (true positive rate)} = \frac{VP}{VP + FN}$$

- **F-score**: El F-score es la media armónica entre precisión y recall, útil cuando hay un desbalance de clases.

$$F\text{-score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

- **AUC-PR**: métrica que evalúa el rendimiento de un clasificador en tareas desbalanceadas, considerando la relación entre precisión y recuperación a diferentes umbrales de decisión.

$$\text{AUC-PR} = \int_0^1 \text{Precision}(r)\, d\text{Recall}(r)$$

- **AUC-ROC**: métrica que evalúa el rendimiento de un clasificador en función de su capacidad para discriminar entre clases positivas y negativas a diferentes umbrales de decisión.

$$\text{AUC-ROC} = \int_0^1 \text{TPR}(f)\, d\text{FPR}(f)$$

Estas métricas permiten evaluar no solo el desempeño general del modelo, sino también su capacidad para distinguir correctamente entre ambas clases en escenarios clínicos donde los falsos negativos tienen un costo elevado.

Para visualizar mejor estas métricas, graficamos lo siguiente:

- **Matriz de confusión**: Una matriz de confusión es una tabla que muestra el desempeño de un modelo de clasificación, comparando las predicciones del modelo con los valores reales en términos de verdaderos positivos, falsos positivos, verdaderos negativos y falsos negativos.
- **Curva ROC**: es una gráfica que muestra la relación entre la tasa de verdaderos positivos (TPR) y la tasa de falsos positivos (FPR) para diferentes umbrales de decisión de un clasificador.
- **Curva PR (Precision-Recall) y AUC-PR**: es una gráfica que muestra la relación entre la precisión y la recuperación de un clasificador para diferentes umbrales de decisión.

Primero se realizó un entrenamiento del modelo para una base de datos balanceada, es decir que tiene una cantidad similar de muestras para cada clase. Luego, se realizó el mimso estudio para una base de datos desbalanceada, por lo que se implementron diversas técnicas de rebalanceo previo al entrenamiento del modelo.

Las técnicas utilizadas fueron las siguientes:

- **Undersampling**: Consiste en reducir el número de muestras de la clase mayoritaria para balancear el dataset.
- **Oversampling mediante duplicación**: Aumenta las muestras de la clase minoritaria duplicando instancias existentes para igualar la distribución.
- **Oversampling mediante SMOTE**: Genera nuevas instancias sintéticas de la clase minoritaria interpolando entre una muestra y sus vecinos más cercanos.
  Fórmula:
  $$x_{\text{nuevo}} = x_i + \delta \cdot (x_{nn} - x_i)$$
  donde x_i es una muestra minoritaria, x_{nn} uno de sus vecinos cercanos, y $\delta \in [0, 1]$ un valor aleatorio.
- **Cost reweighting**: Asigna un mayor peso a los errores de la clase minoritaria en la función de pérdida. En la función de costo, se multiplican los términos que dependen de las muestras de la clase minoritaria por un factor
  $$C = \frac{\pi_2}{\pi_1}$$
  donde π_1 es la probabilidad a priori de la clase minoritaria y π_2 la de la clase mayoritaria. Esto efectivamente re-balancea la importancia de cometer errores de clasificación en ambas clases.

1.3. Resultados

Luego de realizar validación cruzada en cada uno de los modelos de rebalanceo, se llegó a que los mejores valores de λ para el funcionamiento de cada modelo son siguientes:

- **Sin rebalanceo:** $\lambda = 22{,}22$
- **Undersampling:** $\lambda = 0$
- **Oversampling duplicate:** $\lambda = 22{,}22$
- **Oversampling SMOTE:** $\lambda = 88{,}89$
- **Cost re-weighting:** $\lambda = 0$

Modelo	Accuracy	Precision	Recall	F-Score	AUC-ROC	AUC-PR
Sin rebalanceo	0.9057	0.7500	0.8947	0.8160	0.9085	0.6206
Undersampling	0.9221	0.7639	0.9649	0.8527	0.9130	0.6341
Oversampling duplicate	0.9139	0.7432	0.9649	0.8397	0.9093	0.6224
Oversampling SMOTE	0.9098	0.7397	0.9474	0.8308	0.9087	0.6199
Cost re-weighting	0.9057	0.7429	0.9123	0.8189	0.8927	0.5790

Cuadro 1: Métricas de performance de cada modelo, evaluando en el set de validación

Modelo	Accuracy	Precision	Recall	F-Score	AUC-ROC	AUC-PR
Sin rebalanceo	0.8676	0.7105	0.7941	0.7500	0.7872	0.5061
Undersampling	0.8676	0.7000	0.8235	0.7568	0.7612	0.4839
Oversampling duplicate	0.8676	0.7000	0.8235	0.7568	0.7618	0.4916
Oversampling SMOTE	0.8676	0.7000	0.8235	0.7568	0.7587	0.4850
Cost re-weighting	0.8603	0.6923	0.7941	0.7397	0.7771	0.4871

Cuadro 2: Métricas de performance de cada modelo, evaluando en el set de evaluación

Una vez obtenidos estos valores, entrenamos el modelo de Regresion Logistica con este valor y con los valores del set de entrenamiento. Una vez entrenado, se realizan predicciones sobre el set de validación y se reportan las métricas previamente enunciadas.

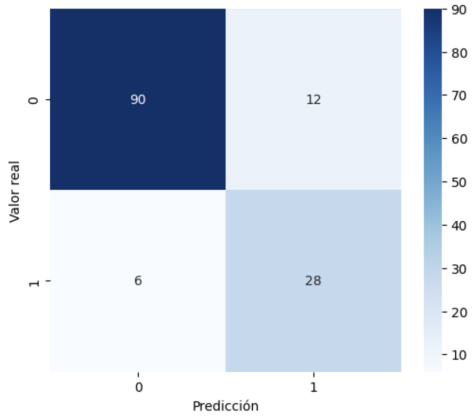


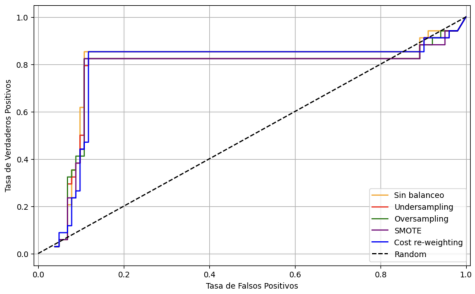
Figura 2: Matriz de confusión del modelo rebalanceado mediante undersampling evaluado en test

Viendo el Cuadro 1 presentado debajo, podemos sacar las siguientes conclusiones. Dado que el objetivo clínico principal en el diagnóstico de cáncer de mama es minimizar los **falsos negativos** —es decir, reducir al máximo la probabilidad de no detectar un caso positivo—, se priorizan métricas como **Recall**, **F1-Score** y **AUC-PR**.

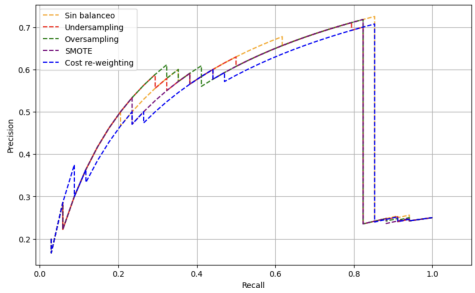
Bajo estos criterios, el modelo entrenado con la técnica de **undersampling** demostró el mejor desempeño, alcanzando un **Recall de 0.9649**, un **F1-Score de 0.8527** y el **mayor valor de AUC-PR (0.6341)** entre todos los modelos evaluados.

Por lo tanto, este modelo se considera el más adecuado para ser implementado en un entorno de producción clínica, donde la sensibilidad del sistema es crítica para asegurar la detección temprana y oportuna de casos positivos.

Al evaluar los modelos en el **set de evaluación** (Cuadro 2), observamos que los modelos con **undersampling**, **oversampling duplicate** y **SMOTE** obtienen el valor más alto de Recall (**0.8235**). Sin embargo, para una selección más robusta, también se deben considerar métricas complementarias



(a) Curvas ROC de cada modelo en set de evaluación



(b) Curva PR de cada modelo en set de evaluación

Figura 1: Comparación de curvas ROC y PR en el set de evaluación

como el **F-Score** y el **AUC-PR**, que capturan el balance entre precisión y recall, y el rendimiento en problemas desbalanceados, respectivamente.

Las curvas ROC y Precision-Recall permiten visualizar mejor el rendimiento de los modelos. En la curva ROC (Figura 1) se observa que los métodos de *SMOTE*, *Oversampling* y *Undersampling* muestran una buena capacidad de discriminación entre clases. En particular, los métodos *SMOTE* y *Cost re-weighting* alcanzan una alta tasa de verdaderos positivos con baja tasa de falsos positivos.

En la curva Precision-Recall(Figura 2), se evidencia que el modelo sin balanceo presenta un buen equilibrio entre precisión y sensibilidad (recall), aunque técnicas como *Oversampling*, *SMOTE* y *Undersampling* logran curvas competitivas, especialmente en valores altos de recall. Por otro lado, *Cost re-weighting* muestra una menor precisión en rangos bajos de recall, indicando posibles falsos positivos.

En resumen, las curvas sugieren que los métodos de *Undersampling SMOTE* y *Oversampling* ofrecen un buen compromiso entre precisión y recall, mejorando la capacidad del modelo para detectar la clase minoritaria sin sacrificar demasiado la precisión.

En base a estos resultados, y priorizando la capacidad del modelo para detectar todos los casos positivos posibles, consideramos que el modelo de undersampling es el más recomendable, ya que ofrece el mejor Recall y presenta buenos valores de AUC-PR y F-Score. Aunque pierde algo de precisión con respecto a otros modelos, eso es aceptable porque es preferible tener falsos positivos que falsos negativos en este contexto.

La matriz de confusión obtenida en el set de evaluación (Figura 3) muestra que el modelo predijo correctamente 28 de los 34 casos positivos. Esto implica que sólo se produjeron **6 falsos negativos**, lo cual es fundamentalmente bajo considerando el contexto clínico, donde es crucial detectar todos los casos de cáncer.

Además, se observa un total de 12 **falsos positivos**. Aunque este valor es moderado, es aceptable ya que en este tipo de aplicación se prefiere cometer falsos positivos antes que falsos negativos.

En resumen, la matriz respalda la selección del modelo ya que cumple adecuadamente con el objetivo principal: **minimizar los falsos negativos**.

2. Predicción de Rendimiento de Jugadores de Basketball

Resumen

En este experimento se buscó encontrar el mejor modelo para la predicción del rendimiento de jugadores de Basketball. Para esto, se realizó un análisis de la base de datos proporcionada y se aplicaron técnicas de preprocesamiento para acomodar dicha base. Luego, se evaluaron tres modelos: Análisis discriminante lineal(LDA), Regresión logística multi-clase con regularización y Random Forest. Se entrenaron los modelos con el set de entrenamiento y se predicieron valores con los sets de validación y de evaluación. También se ajsutaron hiperparámetros para ciertos modelos como Random Forest. Finalmente, se reportaron las mismas métricas de performance que en el

ejercicio anterior. Concluímos que el modelo de Random Forest es el mejor para predecir en este caso, y que las métricas de performance entre los sets de validación y de evaluación son similares, representando que no hay overfitting.

2.1. Introducción

En esta sección se aborda el problema de estimar el rendimiento de jugadores profesionales de basketball a partir de métricas individuales extraídas de sus estadísticas en distintas temporadas. Para ello, se emplea como variable objetivo la métrica **WAR (Wins Above Replacement)**, que indica cuántas victorias adicionales aporta un jugador en comparación con un jugador suplente promedio.

La tarea se plantea como un problema de clasificación supervisada multiclase, donde se busca predecir la clase a la que pertenece cada jugador: *Negative WAR* (clase 1), *Null WAR* (clase 2) o *Positive WAR* (clase 3). Cada instancia del conjunto de datos representa a un jugador en una temporada particular, e incluye una serie de atributos numéricos que reflejan su desempeño. Estos atributos incluyen minutos jugados, número de posesiones y el impacto en ofensa/defensa de ese jugador.

El objetivo principal es desarrollar modelos que sean capaces de predecir de manera precisa y confiable a qué clase pertenece un jugador, en función de sus estadísticas individuales. Esta predicción resulta valiosa tanto para equipos como para analistas deportivos, ya que permite identificar jugadores con alto potencial, evaluar incorporaciones, o estimar el impacto de un jugador en una temporada.

A lo largo del trabajo se exploran distintos modelos de clasificación, junto con técnicas de evaluación, buscando seleccionar el enfoque que mejor generalice sobre nuevos jugadores no vistos durante el entrenamiento.

2.2. Métodos

2.2.1. Modelos de clasificación

Para predecir la clase de cada jugador, se emplearon tres modelos de clasificación. El primero de ellos es el **Análisis Discriminante Lineal (LDA)**. Este es un método supervisado de reducción de dimensionalidad y clasificación que busca proyectar los datos a un espacio de menor dimensión maximizando la separabilidad entre clases. LDA asume que las diferentes clases generan datos a partir de distribuciones normales con la misma matriz de covarianza, pero diferentes medias.

La función discriminante lineal para una clase k se define como:

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

donde:

- x es el vector de características de entrada,
- μ_k es el vector de medias de la clase k,
- Σ es la matriz de covarianza común entre las clases,
- π_k es la probabilidad a priori de la clase k.

Para clasificar una observación x, se evalúan las funciones discriminantes para todas las clases y se asigna aquella que tenga el mayor valor de $\delta_k(x)$. LDA es especialmente útil cuando el número de variables es alto en relación a la cantidad de observaciones y cuando las clases son aproximadamente linealmente separables.

Otro de los modelos usados fue la **Regresión Logsística multi-clase con regularización L2**. La regresión logística multiclase es una extensión del modelo de regresión logística binaria para problemas donde la variable objetivo puede tomar más de dos clases. Una de las estrategias más comunes es la de *one-vs-rest* (OvR), donde se entrena un clasificador binario por cada clase, separando esa clase del resto.

Para una observación x, la probabilidad de que pertenezca a la clase k se calcula como:

$$P(y = k \mid x) = \frac{e^{\theta_k^T x}}{\sum_{j=1}^{K} e^{\theta_j^T x}}$$

donde:

- θ_k es el vector de parámetros asociado a la clase k,
- K es el número total de clases.

El modelo se entrena minimizando la función de pérdida de entropía cruzada regularizada, definida como:

$$J(\Theta) = -\frac{1}{m} \sum_{i=1}^{m} \sum_{k=1}^{K} y_k^{(i)} \log \left(\frac{e^{\theta_k^T x^{(i)}}}{\sum_{j=1}^{K} e^{\theta_j^T x^{(i)}}} \right) + \frac{\lambda}{2m} \sum_{k=1}^{K} \|\theta_k\|^2$$

donde:

- m es la cantidad de ejemplos de entrenamiento,
- λ es el parámetro de regularización (L2),
- $\|\theta_k\|^2$ representa la norma cuadrada de los coeficientes de la clase k.

La regularización L2 ayuda a prevenir el sobreajuste, penalizando los modelos con coeficientes excesivamente grandes. Se implementó validación cruzada para evaluar el mejor lambda para el modelo en la instancia de evaluar toda la base de datos y de evaluarla en el set de evaluación.

El último de los modelos implementados es el de **Random Forest**. Random Forest es un algoritmo de ensamblado basado en árboles de decisión, que combina múltiples clasificadores para mejorar la precisión y robustez del modelo. En este trabajo se implementó un Bosque Aleatorio utilizando la **entropía** como criterio de división en cada nodo. La entropía mide la impureza o aleatoriedad de un conjunto de datos y se define como:

$$H(S) = -\sum_{i=1}^{C} p_i \log_2(p_i)$$

donde:

- S es el conjunto de datos en un nodo dado,
- C es el número de clases,
- p_i es la proporción de instancias de la clase i en S.

En cada división, el árbol selecciona la característica y el umbral que generan la mayor **ganancia de información**, definida como:

$$\text{Gain}(S, A) = H(S) - \sum_{v \in \text{valores}(A)} \frac{|S_v|}{|S|} H(S_v)$$

donde A es una característica y S_v es el subconjunto de S con valor v para la característica A.

Para construir el modelo, se experimentó con distintas configuraciones de hiperparámetros, entre ellos:

- **n_estimators**: número de árboles en el bosque.
- **max_depth**: profundidad máxima de cada árbol.
- **max_features**: número de características consideradas al dividir un nodo.

- **min_samples_split**: cantidad mínima de muestras necesarias para dividir un nodo.

La selección de la mejor combinación de hiperparámetros se realizó mediante Grid search, que es un tipo de búsqueda exhaustiva que evalúa sistemáticamente todas las combinaciones posibles de hiperparámetros predefinidos para encontrar la configuración que optimiza el rendimiento de un modelo.

El uso de entropía como criterio permite capturar mejor la incertidumbre en escenarios donde la distribución entre clases no es uniforme, y suele resultar en divisiones muy informativas para el caso.

2.2.2. Limpieza del dataset

Muy similar a como se hizo en el primer experimento, solo que realizado sobre el conjunto de datos de `WAR_class_dev.csv`. Leer el inciso 1.2.2 para mayor explicación. El único cambio a diferencia al procedimiento del experimento anterior es que se elimina la columna `war_total`, debido a que representaba lo mismo que la columna `war_class`. Tambíen se estudió el balance de los datos, y se concluyó que el 30 porciento de los datos pertenecen a la clase 1, el 33 porciento a la clase 2 y un 37 porciento a la clase 3. Esto indica que la base de datos está suficientemente balanceada y que no es necesario aplicar técnicas de rebalanceo para el entrenamiento posterior.

2.2.3. Métricas de performance

Las métricas de performance utilizadas son las mismas que se usaron en el primer experimento, leer inciso 1.2.3 para más detalle. El único cambio es que se reportan de manera macro, esto quiere decir que se toma el valor de la métrica para cada clase de la variable objetivo y luego se toma el valor promedio total.

2.3. Resultados

Para analizar el funcionamiento y la predicitibilidad de cada mmodelo, se realizó lo siguiente. Primero, se evaluó cada modelo en su set de entrenamiento y se evaluaron métricas de rendimiento en el set de validación. Luego, se entrenó una nueva instancia de cada uno de estos modelos, pero con ciertos cambios. Esta vez, se entrenó con el set de desarrollo entero (es decir, WAR_class_dev.csv) y se ajustó el modelo para encontrar los mejores parámetros posibles, ya sea mediante validación cruzada o haciendo Grid-Search. Luego, estos modelos se evaluaron en el conjunto de evaluación. A continuación se presentan los resultados de estos procedimientos.

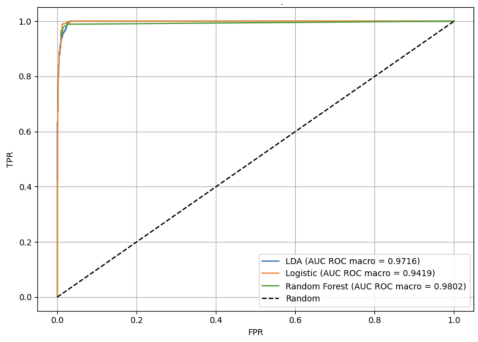
Modelo	Accuracy	Precision	Recall	F-Score	AUC-ROC	AUC-PR
LDA	0.8923	0.9035	0.9009	0.8897	0.9716	0.9316
Regresión Logistica multi-clase	0.9152	0.9153	0.9189	0.9148	0.9419	0.8962
Random Forest	0.9631	0.9629	0.9647	0.9636	0.9802	0.9806

Cuadro 3: Métricas de performance de cada modelo multi-clase, evaluando en el set de validación

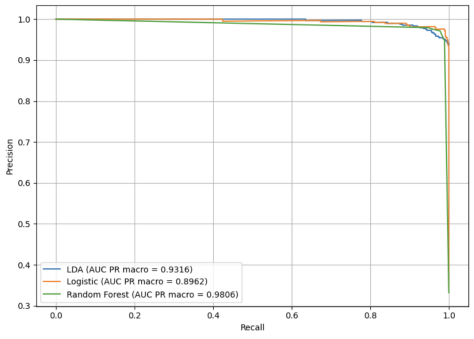
Modelo	Accuracy	Precision	Recall	F-Score	AUC-ROC	AUC-PR
LDA	0.8838	0.8949	0.8959	0.8812	0.965	0.9099
Regresión Logistica multi-clase	0.8980	0.8995	0.9060	0.8977	0.9364	0.8689
Random Forest	0.9629	0.9628	0.9652	0.9639	0.9898	0.9884

Cuadro 4: Métricas de performance de cada modelo multi-clase, evaluando en el set de evaluación

A partir del análisis de los resultados obtenidos en los Cuadros 3 y 4, así como de las curvas ROC y PR correspondientes (Figuras 3 y 4), es posible concluir que el modelo de **Random Forest** es el

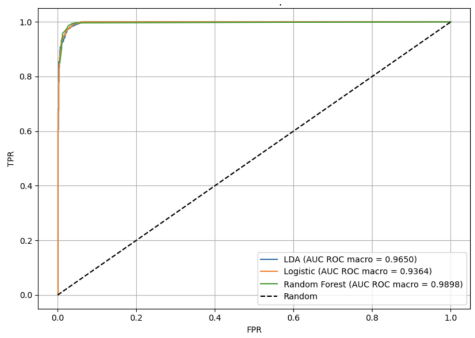


(a) Curvas ROC de cada modelo en set de evaluación

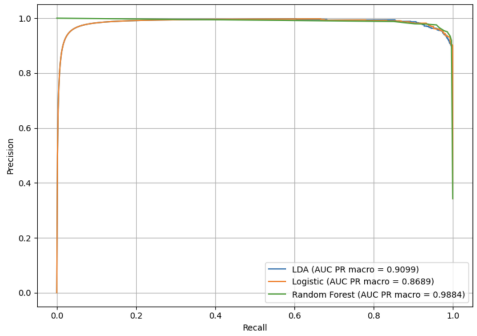


(b) Curva PR de cada modelo en set de entrenamiento

Figura 3: Comparación de curvas ROC y PR en el set de entrenamiento



(a) Curvas ROC de cada modelo multi-clase en set de evaluación



(b) Curva PR de cada modelo multi-clase en set de evaluación

Figura 4: Comparación de curvas ROC y PR en el set de evaluación

más adecuado para llevar a producción en el contexto de predicción del rendimiento de jugadores de basketball.

Este modelo presenta consistentemente el mejor desempeño en todas las métricas evaluadas tanto en el set de validación como en el de test. En particular, alcanza un **F-Score de 0.9636 en validación** y **0.9639 en test**, lo que evidencia una excelente capacidad de generalización. Asimismo, obtiene los **valores más altos de AUC-ROC (0.9802 y 0.9898)** y **AUC-PR (0.9806 y 0.9884)**, lo cual indica un comportamiento robusto frente a distintas configuraciones de umbrales, y una notable capacidad para distinguir entre las tres clases de rendimiento.

Al comparar las métricas de validación con las obtenidas sobre el conjunto de test, se observa que son muy similares en todos los modelos, lo cual sugiere que la estimación de desempeño mediante validación cruzada fue confiable y no hubo *overfitting* significativo. Esta consistencia refuerza la validez del proceso de selección del modelo y la confianza en los resultados reportados.

Aunque los otros modelos (LDA y Regresión Logística Multiclase) muestran también un desempeño aceptable — especialmente LDA en AUC-ROC y AUC-PR — sus valores de F-Score, Accuracy y AUC-PR son inferiores a los del modelo Random Forest, particularmente en el set de evaluación, lo cual justifica la elección del modelo de bosque aleatorio como el más eficaz.

Por todo lo anterior, y considerando tanto la calidad predictiva como la estabilidad del modelo, **Random Forest** se destaca como la mejor opción para ser desplegada en un entorno de producción donde se busque predecir de manera confiable el rendimiento de jugadores en base a sus estadísticas individuales.