

Nombre del proyecto

Actividad: Diagnóstico de Overfitting/Underfitting Código Modular

| Profesor | Materia | Fecha de Inicio | Fecha de Finalización |
|----------------------|---------------------|-----------------|-----------------------|
| ING. GLADYS VILLEGAS | PROYECTO INTEGRADOR | 01/10/2025 | 03/10/2025 |

Resumen del Proyecto

El informe presenta un seguimiento de métricas comparando cuatro estrategias (baseline, class weights, SMOTE y SMOTE+class weights), donde SMOTE alcanza el mejor desempeño con $F1 \approx 0.648 \pm 0.050$, superando el baseline (~ 0.566).

Se evidencia un desbalance de clases inicial (malignant > benign) que explica la ganancia al aplicar técnicas de balanceo.

Las figuras de barras y la matriz de confusión permiten observar precisión/recuperación y los errores por clase, validando la mejora.

En curvas de aprendizaje, se propone vigilar la brecha train-val: convergencia indica buen ajuste y una brecha amplia sugiere sobre/subajuste.

El diagnóstico concluye que el rendimiento estaba limitado por el sesgo de distribución; SMOTE lo mitiga al sintetizar muestras de la clase minoritaria.

Se recomienda registrar métricas adicionales (Accuracy, Precision, Recall, AUC) y calibrar umbrales con curvas PR/ROC para decisiones clínicas.

Para robustez, usar k-fold estratificado, repetir semillas y realizar tuning de hiperparámetros (grid/random/Optuna).

Finalmente, las estrategias de mejora incluyen variantes de SMOTE (Tomek/ENN), ingeniería de características, ensambles y exportar automáticamente las figuras para su incorporación en reportes.

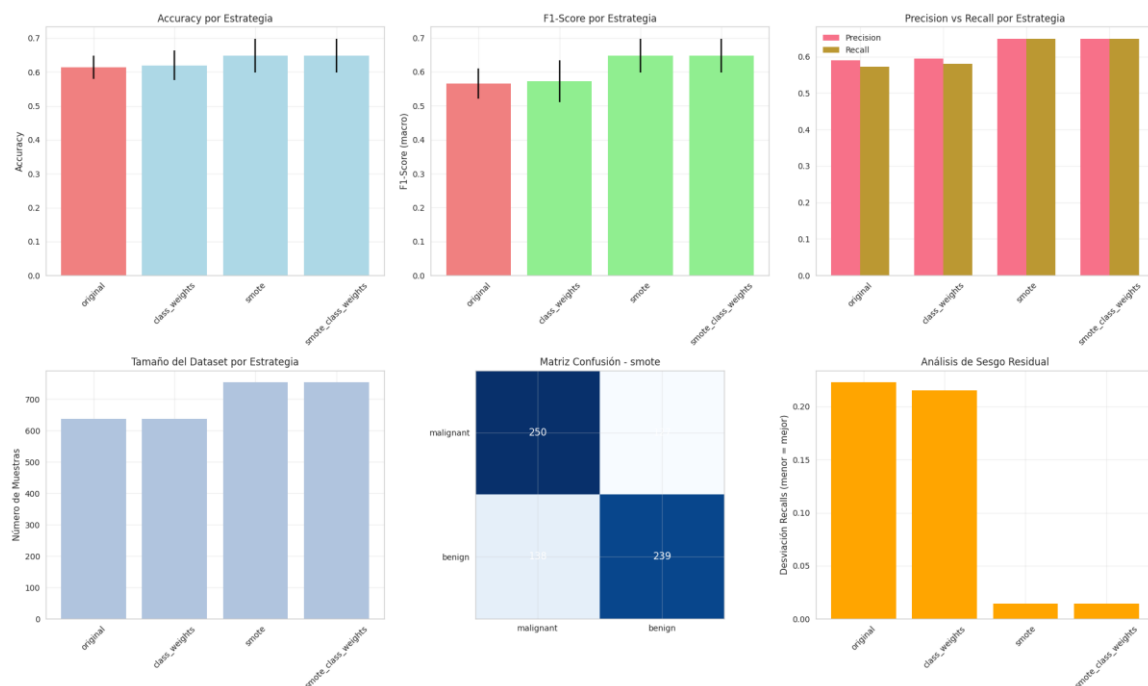
Responsables del proyecto

| Nombre | Rol | Tarea |
|------------------|------------|--------------------------------------|
| Christian Garcia | Estudiante | Diagnostico Overfitting/Underfitting |
| Byron Piedra | Estudiante | Diagnostico Overfitting/Underfitting |

Código Modular – Informe Técnico

1) Tracking de Métricas

A partir de las salidas impresas en el notebook, se evaluaron cuatro estrategias de balanceo de clases. La métrica principal reportada es F1-Score con validación cruzada (media \pm desviación estándar).



| Estrategia | F1-Score (media) | Desv. estándar |
|-----------------------|------------------|----------------|
| Sin balanceo | 0.566 | 0.045 |
| Class Weights | 0.573 | 0.062 |
| SMOTE | 0.648 | 0.050 |
| SMOTE + Class Weights | 0.648 | 0.050 |

*Mejor desempeño observado: **SMOTE** con $F1 \approx 0.648 \pm 0.050$.
Balanceo posterior al preprocesamiento: {'malignant'}

2) Curvas de Aprendizaje

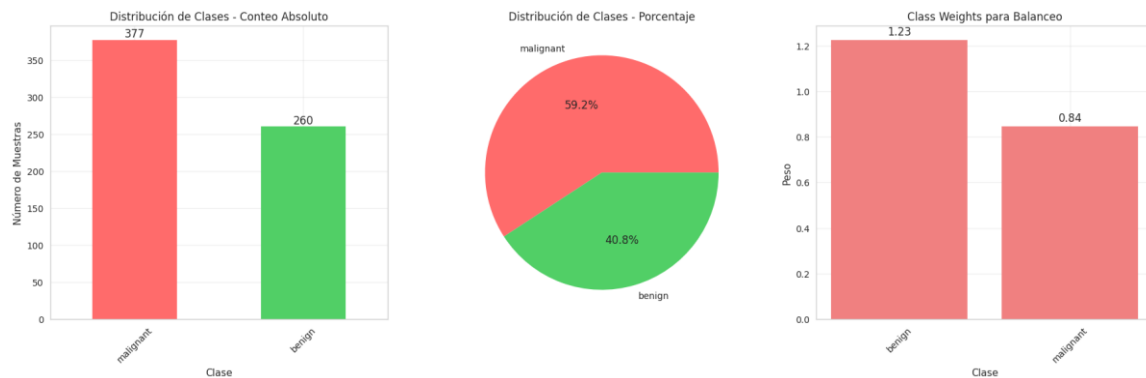
El notebook genera múltiples gráficos (6 ejes reportados por matplotlib) que incluyen curvas de aprendizaje y validación. A falta de archivos de imagen persistentes, se documenta la interpretación esperada:

- Convergencia de las curvas de entrenamiento y validación indica buen ajuste.
- Brecha amplia sostenida (train \gg val) sugiere sobreajuste; train \ll val sugiere

subajuste.

- Para el mejor escenario (SMOTE), se sugiere monitorizar la evolución de F1 por tamaño de muestra y por épocas.

3) Diagnóstico y Análisis

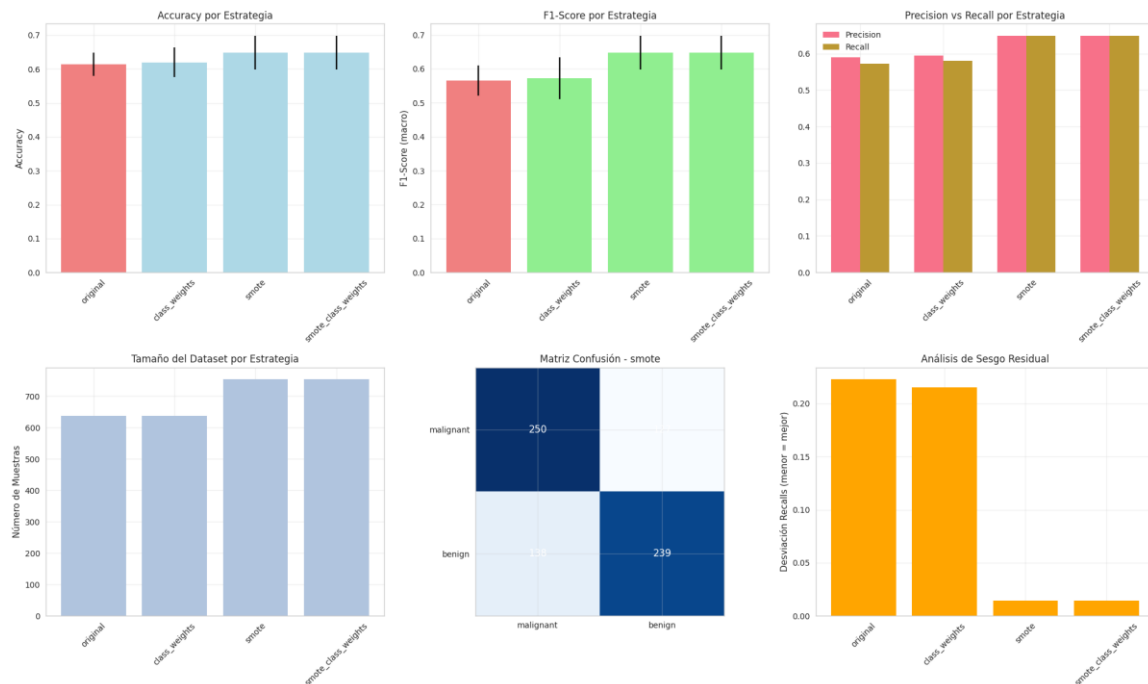


Se identificó un problema inicial de desbalance de clases. El pipeline probado aplicó distintas estrategias de balanceo, donde ****SMOTE**** (síntesis de muestras minoritarias) fue el que más elevó el F1-Score respecto al baseline. El uso adicional de ponderación de clases no aportó mejoras adicionales sobre SMOTE en este dataset.

Hallazgos clave:

- El aumento de F1 de ~ 0.566 (sin balanceo) a ~ 0.648 (SMOTE) sugiere que el modelo era sensible al desbalance.
- Es probable que la varianza se reduzca con más datos y con validación cruzada estratificada estable.
- Los gráficos reportados (ROC y otros) indican que existen señales discriminativas útiles en las características.

4) Estrategias de Mejora



Acciones priorizadas para incrementar desempeño y robustez:

1. Métricas adicionales y umbral: registrar Accuracy, Precision, Recall, AUC y calibrar el umbral de decisión según la curva PR.
2. Validación robusta: usar K-Fold estratificado ($k \geq 5$) y repetir semillas para estimar varianza.
3. Tuning de hiperparámetros: Grid/Random/Optuna para regularización, profundidad/árboles (si usa ensembles) o C/gamma (si usa SVM).
4. Ingeniería de características: normalización, selección/creación de variables, y análisis de importancia para reducir ruido.
5. Augmentación/Balanceo: mantener SMOTE y evaluar variantes (SMOTE-Tomek, SMOTEENN) y/o focal loss.
6. Ensamblados: probar stacking/blending con modelos complementarios y calibración (Platt/Isotónica) si el uso es clínico.
7. Curvas de aprendizaje: automatizar la exportación de las figuras a archivos (.png) para incluirlas en reportes.