Universidad Nacional de Luján
Licenciatura en Sistemas de Información

Aplicación de técnicas de sobremuestreo en problemas de clasificación de datos desbalanceados en diferentes datasets

Tesina presentada para aplicar al título
de Licenciado en Sistemas de Información

Juan Manuel Natello

Director: **Banchero, Santiago**

Junio 2025

Índice

1. Resumen

Este trabajo se centra en el análisis, diseño e implementación de técnicas de sobremuestreo para abordar el problema del desbalance de clases en tareas de clasificación supervisada. Este fenómeno, frecuente en dominios como la medicina, las finanzas o la teledetección, implica una distribución desigual entre clases, donde la clase de interés suele estar subrepresentada. En estos casos, los algoritmos tienden a favorecer la clase mayoritaria, lo que reduce la sensibilidad del modelo frente a eventos poco frecuentes pero altamente relevantes. El problema específico que se aborda es la limitada efectividad de las técnicas clásicas de sobremuestreo, en particular SMOTE, frente a escenarios con ruido estructural, solapamiento entre clases o alta dimensionalidad. Si bien la literatura ha propuesto variantes y enfoques híbridos, muchas de estas soluciones siguen presentando desafíos en términos de adaptabilidad multiclase, selección de instancias relevantes y generación de ejemplos sintéticos útiles para el clasificador. Además, persiste la necesidad de diseñar mecanismos que permitan controlar la calidad, ubicación y distribución de las muestras generadas. El objetivo general de esta investigación es evaluar en qué medida la hibridación de técnicas existentes y la propuesta de un nuevo enfoque pueden mejorar el rendimiento de los modelos de clasificación en contextos de desbalance. Para ello, se desarrollarán dos nuevas técnicas: una variante híbrida que integra mecanismos de selección basados en α-distancia y de generación sintética mediante criterios geométricos y adaptativos, y una técnica denominada que incorporará mecanismos de control basados en percentiles. Ambas serán evaluadas experimentalmente sobre conjuntos de datos reales, tanto binarios como multiclase, utilizando clasificadores estándar y métricas robustas como F1-score, G-mean y AUC. Se espera que este trabajo contribuya al desarrollo de soluciones más precisas, adaptativas y controlables para el tratamiento del desbalance, mejorando la capacidad predictiva de los modelos y aportando nuevas herramientas para su aplicación en contextos reales.

2. Área temática

Aprendizaje Automático - Sobremuestreo - Desbalance de Clases.

3. Palabras claves

Aprendizaje automático, Datos desbalanceados, Sobremuestreo, SMOTE, Técnicas híbridas, Clasificación multiclase, Evaluación experimental.

4. Fundamentación de la investigación

↑ Volver al índice

El desbalance de clases constituye uno de los desafíos más persistentes y complejos en la construcción de modelos de aprendizaje automático supervisado, debido a que puede afectar de manera significativa la capacidad de generalización de los algoritmos y comprometer su utilidad práctica en

aplicaciones reales. Esta situación es especialmente crítica en contextos donde las instancias minoritarias, aunque escasas, revisten una alta importancia analítica o social, como ocurre en el diagnóstico médico, el monitoreo ambiental, la identificación de riesgos en mercados financieros volátiles o la clasificación de coberturas en imágenes satelitales multiespectrales. En estos escenarios, la mayoría de los modelos tienden a favorecer la clase mayoritaria, mostrando bajos niveles de sensibilidad frente a los eventos infrecuentes, lo cual genera una pérdida sustantiva de información relevante. Tal como se ha argumentado en estudios recientes, este tipo de sesgo no sólo degrada el rendimiento de los modelos, sino que puede inducir errores de interpretación y decisiones perjudiciales en ámbitos sensibles, debido a una representación insuficiente de los casos críticos durante la fase de entrenamiento (Poddar et al., 2024).

Frente a este escenario, las técnicas de sobremuestreo sintético han demostrado ser una de las estrategias más efectivas para mitigar dicho desbalance sin necesidad de descartar datos (Carvalho et al., 2025; Khorshidi & Aickelin, 2025). Entre ellas, SMOTE (Synthetic Minority Over-sampling Technique) se ha consolidado como una de las técnicas base en el desarrollo de métodos de sobremuestreo, siendo aún ampliamente referenciada en estudios recientes como punto de partida para mejoras o hibridaciones (Nasaruddin et al., 2025; Wang & Awang, 2025), generando una amplia variedad de extensiones que buscan mejorar su desempeño en escenarios reales, especialmente frente a fenómenos como el solapamiento entre clases, la escasez informativa o los desbalances internos. Esta mejora en el desempeño va de la mano de diferentes perspectivas: por ejemplo, algunos autores enfatizan la necesidad de evitar zonas de baja densidad o de bajo valor en el espacio de características (Lyu et al., 2025; Qiu et al., 2025), mientras que en contextos sensibles como el diagnóstico clínico, la calidad de las instancias generadas es tan crucial como su cantidad. En esa línea, los hallazgos han mostrado que el tratamiento del desbalance debe orientarse a preservar la estructura local de los datos y mejorar la utilidad clasificatoria de los ejemplos añadidos, más allá de igualar proporciones (Wang & Awang, 2024a).

Estas propuestas recientes buscan superar las limitaciones estructurales de los enfoques clásicos de sobremuestreo, especialmente aquellas asociadas a la generación de instancias en regiones con bajo valor informativo o dominadas por ruido. En lugar de aplicar el sobremuestreo de forma uniforme, se exploran enfoques que consideran la estructura interna del espacio de características, identificando zonas de alta densidad representativa o utilidad clasificatoria. Este tipo de estrategias resulta particularmente útil en escenarios donde la clase minoritaria se encuentra pobremente representada, ya sea por su escasa frecuencia o por la dispersión de sus instancias en el espacio de atributos. En respuesta, se han propuesto esquemas que integran mecanismos de evaluación de densidad, interpolación adaptativa y filtrado espacial, con el objetivo de generar muestras sintéticas más coherentes con la distribución real de los datos y preservar la integridad del espacio de decisión. Estos avances reflejan una tendencia hacia el diseño de técnicas de sobremuestreo más inteligentes, capaces de adaptarse a contextos complejos y severamente desbalanceados (Lyu et al., 2025; Nasaruddin et al., 2025; Qiu et al., 2025).

En consecuencia, los enfoques recientes coinciden en que no basta con igualar proporciones entre clases: se requiere optimizar la calidad, la ubicación y la utilidad de las instancias generadas para

lograr una mejora efectiva en el rendimiento del clasificador. Generar ejemplos sintéticos en regiones poco informativas, dispersas o ruidosas puede incluso deteriorar la capacidad de generalización del modelo (Lyu et al., 2025; Qiu et al., 2025). En respuesta, se proponen técnicas que orientan el sobremuestreo hacia regiones estructuralmente relevantes del espacio de características, priorizando aquellas con mayor valor predictivo o mayor densidad representativa de la clase minoritaria. Para abordar estas cuestiones, se han propuesto variantes que: (i) regulan la cantidad de muestras en función de la contribución de los atributos al modelo; (ii) enfocan el sobremuestreo en regiones estructuralmente relevantes del espacio de características; y (iii) refuerzan la necesidad de mantener un equilibrio entre cantidad y calidad de ejemplos para evitar distorsiones (Lyu et al., 2025; Nasaruddin et al., 2025; Qiu et al., 2025). En conjunto, estos enfoques avanzan hacia un sobremuestreo inteligente y dirigido, basado en criterios estructurales más precisos y adaptativos que superan las limitaciones del SMOTE clásico.

En síntesis, se observa una clara tendencia hacia el diseño de técnicas que intentan mitigar las limitaciones estructurales de SMOTE mediante enfoques más informados y adaptativos. No obstante, aún persisten desafíos importantes vinculados a la sensibilidad de los algoritmos a la elección de parámetros, a la generación de ruido en regiones de solapamiento, y a la limitada capacidad de adaptación en contextos multiclase o con estructuras espaciales complejas.

5. Descripción del tema de estudio

↑ Volver al índice

En la última década, el aprendizaje automático (Machine Learning, ML) ha experimentado un crecimiento exponencial, impulsado por el aumento en la capacidad computacional, la disponibilidad masiva de datos y el desarrollo de algoritmos cada vez más sofisticados. Este avance ha favorecido su adopción en ámbitos tan diversos como la medicina, la ciberseguridad, las finanzas, la industria manufacturera y las ciencias sociales, consolidándose como una herramienta clave tanto en la industria como en la literatura científica, como lo evidencian el aumento de publicaciones indexadas y repositorios especializados en aprendizaje automático (Khorshidi & Aickelin, 2025; Nasaruddin et al., 2025).

Una de las tareas fundamentales en aprendizaje automático es la clasificación supervisada, que consiste en entrenar un modelo a partir de un conjunto de datos previamente etiquetado, con el objetivo de predecir la clase correspondiente de nuevas instancias. Este enfoque se aplica en numerosos dominios, como por ejemplo el diagnóstico clínico, donde se busca determinar si un paciente padece una determinada enfermedad en función de sus valores fisiológicos. En este tipo de escenarios, la calidad, distribución y estructura del conjunto de entrenamiento resultan determinantes para el desempeño del modelo, el cual puede ser sobrestimado si se emplean métricas que no consideran el desbalance de clases, como la precisión global (accuracy), que tiende a favorecer la clase mayoritaria. Es frecuente, además, que estos datos provengan de observaciones del mundo real, donde el investigador no tiene control sobre cómo se generan las muestras (Qiu et al., 2025).

La presencia de distribuciones de clase desiguales en los conjuntos de datos da lugar a un pro-

blema recurrente conocido como desbalance de clases, que sucede cuando una o más clases están sobrerrepresentadas frente a otras. En casos binarios, se denomina clase mayoritaria a la más frecuente y clase minoritaria a la menos representada. Esta disparidad puede afectar negativamente la capacidad del modelo para identificar correctamente instancias de la clase minoritaria, que en muchas aplicaciones es la de mayor interés práctico. Tal es el caso del análisis ambiental, donde ciertos eventos como incendios o contaminación severa están pobremente representados pero revisten gran importancia predictiva (Qiu et al., 2025).

Para mitigar este problema, se han propuesto dos grandes enfoques ampliamente reconocidos en la literatura: por un lado, los métodos orientados al diseño del clasificador, y por otro, las técnicas de balanceo de datos. El primer grupo incluye estrategias que ajustan el algoritmo de aprendizaje para hacerlo más sensible a la distribución desigual de clases, como los modelos sensibles al costo, que asignan penalizaciones diferenciadas a los errores de clasificación según la clase afectada, y las técnicas de thresholding, que modifican el umbral de decisión para favorecer la detección de instancias minoritarias. Estas soluciones buscan mejorar el rendimiento sin alterar el conjunto de entrenamiento. El segundo grupo, en cambio, actúa directamente sobre los datos, ya sea reduciendo la cantidad de instancias de la clase mayoritaria (submuestreo) o incrementando la clase minoritaria mediante la replicación o la generación de ejemplos sintéticos. Este último enfoque ha ganado particular popularidad debido a su simplicidad, independencia del modelo y facilidad de integración en distintos pipelines de aprendizaje automático (Khorshidi & Aickelin, 2025). Estas técnicas buscan reequilibrar la distribución del conjunto de entrenamiento mediante submuestreo de la clase mayoritaria, sobremuestreo de la clase minoritaria o una estrategia híbrida que combine ambas.

En particular, el sobremuestreo sintético ha cobrado relevancia a partir de la técnica SMOTE, que genera nuevas instancias artificiales interpolando entre muestras cercanas de la clase minoritaria (Chawla et al., 2002). Esta estrategia permite expandir las regiones de decisión del clasificador y mejorar su sensibilidad hacia la clase minoritaria. No obstante, SMOTE también presenta limitaciones importantes, como la generación de muestras sintéticas en regiones dominadas por la clase mayoritaria, lo que puede introducir ruido y confundir al clasificador (Wang & Awang, 2024b).

Como respuesta, la literatura ha propuesto múltiples variantes que buscan mejorar la calidad y relevancia de las muestras generadas. Algunas de estas estrategias se concentran en regiones fronterizas del espacio de decisión, otras ajustan dinámicamente la cantidad de instancias sintéticas según la densidad local, y algunas incorporan criterios geométricos o estadísticos para identificar zonas de mayor valor informativo. Estos enfoques apuntan a refinar no solo la cantidad, sino también la ubicación y utilidad de las muestras generadas (Han et al., 2005; He et al., 2008; Qiu et al., 2025).

No obstante, un consenso emergente en la literatura sostiene que no existe una técnica de sobremuestreo universalmente superior. La eficacia de cada método depende del contexto específico del conjunto de datos, su distribución interna, la presencia de ruido o solapamiento, y las características del dominio de aplicación. Como consecuencia, el tratamiento del desbalance debe ser cuidadosamente adaptado a cada caso, siendo habitual que se requiera una combinación de técnicas o variantes híbridas para obtener resultados óptimos (Galar et al., 2012; Khorshidi & Aickelin, 2025).

Este trabajo se inscribe en la línea de investigación sobre sobremuestreo sintético, con énfasis

en la mejora de algoritmos derivados de SMOTE mediante enfoques estructuralmente informados. Se propone el diseño de dos nuevas técnicas: una variante híbrida que combina la selección basada en α-distancia (αSMOTE) con esquemas geométricos de generación adaptativa (AR-ADASYN), y una técnica propia denominada PC-SMOTE, que incorpora criterios de control mediante percentiles. Ambas serán evaluadas en contextos de clasificación binaria y multiclase, incluyendo dominios sensibles como el diagnóstico clínico y el análisis espacial, con el objetivo de mejorar la capacidad de los clasificadores para identificar correctamente instancias minoritarias en escenarios complejos y desbalanceados.

6. Planteamiento del problema de estudio y objetivos de trabajo

↑ Volver al índice

El estado actual de las técnicas de sobremuestreo sintético ha ampliado significativamente su alcance y aplicabilidad, aunque también ha puesto de manifiesto ciertos desafíos persistentes que continúan motivando nuevas investigaciones. Entre ellos se encuentran la sensibilidad a parámetros, el riesgo de generar muestras en regiones de solapamiento o ruido, la limitada adaptabilidad en escenarios de clasificación multiclase, y la necesidad de mecanismos formales que garanticen la interpretabilidad y calidad de los datos generados (Nasaruddin et al., 2025; Qiu et al., 2025). En este contexto, diversos autores han propuesto el uso de técnicas híbridas como respuesta a estas limitaciones, combinando enfoques de sobremuestreo (por ejemplo, SMOTE) con algoritmos de limpieza de datos (como ENN o Tomek Links), o integrándolos dentro de esquemas de aprendizaje en conjunto como boosting y bagging. Este enfoque híbrido permite reducir el ruido, evitar el sobreajuste y mejorar la capacidad del modelo para detectar eventos raros, especialmente cuando el desbalance de clases es severo (Poddar et al., 2024).

A partir de estas observaciones, surgen interrogantes clave que orientan el desarrollo de la investigación: ¿En qué medida la hibridación de técnicas de sobremuestreo y el diseño de nuevos enfoques puede mejorar el rendimiento de modelos de aprendizaje automático frente a conjuntos de datos desbalanceados, partiendo de escenarios de clasificación binaria y extendiéndose a contextos de clasificación multiclase? ¿Cómo se ve afectado el rendimiento de los modelos supervisados en clasificación multiclase cuando se aplican técnicas de remuestreo sobre el conjunto de entrenamiento, especialmente en presencia de desbalance extremo y alta dimensionalidad?

Considerando lo expuesto, el objetivo principal del presente trabajo es explorar en qué medida la hibridación de técnicas de sobremuestreo existentes, junto con el diseño e implementación de un nuevo enfoque original, puede contribuir a mejorar el rendimiento de modelos de aprendizaje automático ante conjuntos de datos desbalanceados. Para ello, se iniciará el análisis en problemas de clasificación binaria, donde este fenómeno es especialmente crítico, y luego se extenderá la evaluación a contextos de clasificación multiclase, con el fin de validar la robustez y generalidad de las propuestas en escenarios diversos.

6.1. Objetivos secundarios

↑ Volver al índice

1. Explorar desafíos recurrentes en las estrategias actuales de sobremuestreo, tales como la sensibilidad a los parámetros, la generación de muestras sintéticas poco representativas (ruidosas), y la limitada adaptabilidad en escenarios de clasificación multiclase, con el propósito de contribuir al desarrollo de enfoques más robustos y generalizables.

2. Describir y formalizar matemáticamente dos versiones binarias (híbrida y con modificaciones estructurales respectivamente) y una versión multiclase del algoritmo de sobremuestreo SMOTE.

3. Diseñar y analizar una propuesta algorítmica de las 3 versiones del procedimiento de sobremuestreo.

4. Implementar y evaluar experimentalmente el rendimiento de las tres técnicas propias frente a algoritmos estándares de sobremuestreo, aplicándolas sobre datasets binarios y multiclase.

7. Trabajos relacionados

↑ Volver al índice

En esta sección se analizan las principales variantes del algoritmo SMOTE (Synthetic Minority Over-sampling Technique) que han sido desarrolladas en los últimos años, con el objetivo de abordar las limitaciones del método original (Chawla et al., 2002). Estas técnicas han surgido como respuesta a problemas comunes en el sobremuestreo, tales como la generación de ruido, la falta de diversidad en las muestras sintéticas, y la escasa representatividad en regiones fronterizas o de difícil clasificación.

Respecto de las regiones fronterizas, uno de los primeros trabajos en reconocer que no todas las instancias minoritarias son igualmente útiles para la generación de ejemplos sintéticos fue Borderline-SMOTE, una extensión de SMOTE que focaliza el sobremuestreo en aquellas instancias consideradas "peligrosas", es decir, aquellas rodeadas mayoritariamente por vecinos de la clase opuesta (Han et al., 2005). Al restringir la generación de muestras a estas zonas de ambigüedad, se buscaba reforzar el poder discriminativo del modelo sin introducir ruido en regiones seguras, superando así las limitaciones del sobremuestreo uniforme. Sin embargo, esta estrategia trataba a todas las instancias peligrosas de forma equivalente, sin distinguir entre distintos grados de riesgo. Para superar esta limitación, αSMOTE (Feng & Li, 2021) introduce un mecanismo de selección más fino, basado en un esquema de distancias α inversas, que permite cuantificar la peligrosidad relativa de cada instancia fronteriza en función de la densidad y cercanía de sus vecinos. De este modo, la técnica no sólo preserva el enfoque selectivo propuesto por Borderline-SMOTE, sino que lo perfecciona al priorizar de manera adaptativa aquellas muestras que realmente exigen refuerzo, mejorando así la eficacia del sobremuestreo en escenarios de alto desbalance.

A partir de la necesidad de diferenciar entre niveles de dificultad dentro de las regiones fronterizas —un aspecto no abordado por Borderline-SMOTE—, surge ADASYN (Adaptive Synthetic

Sampling) como una alternativa que incorpora un criterio adaptativo en la generación de instancias sintéticas (He et al., 2008). En lugar de tratar por igual a todas las instancias peligrosas, ADASYN cuantifica la dificultad de aprendizaje de cada muestra minoritaria en función de la proporción de vecinos de la clase mayoritaria, asignando mayor cantidad de ejemplos sintéticos a aquellas que se encuentran en entornos más adversos. Esta estrategia permite reforzar de manera focalizada las zonas del espacio de atributos donde el modelo enfrenta mayores desafíos, sin sobrecargar las áreas que ya se encuentran bien representadas, y reduciendo el riesgo de sobreajuste. AR-ADASYN (Park & Kim, 2024) extiende esta lógica adaptativa al incorporar una interpolación más informada, basada en criterios angulares y radiales que tienen en cuenta la geometría local del espacio. Al ajustar tanto la dirección como la magnitud de cada muestra generada según el contexto estructural de sus vecinos, AR-ADASYN logra una distribución más realista y precisa de los datos sintéticos, especialmente en regiones altamente complejas o con bordes difusos. De esta manera complementaria, estas mejoras consolidan un enfoque de sobremuestreo que no solo atiende la densidad y dificultad local, sino que también se adapta a la morfología del espacio de decisión.

Entre las propuestas más recientes, se destacan aquellas que buscan mejorar simultáneamente la selección de instancias y la generación de ejemplos sintéticos, superando las limitaciones observadas en enfoques clásicos. En este sentido, se identifican dos grandes líneas en la evolución del algoritmo SMOTE. La primera comprende aquellas técnicas que introducen modificaciones estructurales sobre su núcleo, ya sea en la selección de instancias minoritarias, la elección de vecinos o la estrategia de generación sintética. Y la segunda incluye métodos que complementan a SMOTE con procesos adyacentes, como filtrado, reducción de dimensionalidad o agrupamiento, sin alterar su lógica central. Ambas líneas, aunque concebidas principalmente para contextos de clasificación binaria, presentan una arquitectura flexible que permite su extensión a tareas multiclase mediante esquemas de descomposición como one-vs-one (OVO) o one-vs-all (OVA), ampliamente utilizados en la literatura de aprendizaje automático para adaptar clasificadores binarios a entornos multiclase (Fernández et al., 2018; Nasaruddin et al., 2025).

Este enfoque de descomposición permite aplicar técnicas de sobremuestreo en cada subproblema binario, generando instancias sintéticas adaptadas a las características particulares de cada enfrentamiento entre clases. Partiendo de esta premisa, variantes como LD-SMOTE, KWSMOTE o SMOTE-MRS, si bien desarrolladas inicialmente para clasificación binaria, no presentan impedimentos técnicos para ser adaptadas a contextos multiclase mediante estas estrategias de descomposición. En el caso de LD-SMOTE, la técnica estima la densidad local de cada muestra minoritaria a través de la similitud de Jaccard entre conjuntos de vecinos, y genera muestras sintéticas dentro de triángulos definidos por vecinos seguros, priorizando regiones densas y evitando zonas ruidosas (Lyu et al., 2025). Por su parte, KWSMOTE redefine la interpolación mediante combinaciones convexas de múltiples vecinos ponderadas por un kernel gaussiano, lo que permite centrar la generación en regiones informativas y con bajo riesgo de ruido (Li et al., 2024). De manera complementaria, SMOTE-MRS incorpora un enfoque híbrido donde primero se agrupan instancias minoritarias con K-Means, luego se aplica SMOTE dentro de cada clúster y finalmente se complementa con Random Oversampling de la clase mayoritaria para garantizar un equilibrio global (Saputra et al., 2024). En una línea distin-

ta pero alineada con el proceso de generación, están aquellas propuestas que se enfocan en mejorar etapas posteriores al sobremuestreo: tal es el caso de ABL-SMOTE, que filtra instancias poco confiables a partir de la confianza de clasificación estimada por un modelo preliminar, o bien SMOTE-PCA-HDBSCAN, que aplica reducción de dimensionalidad mediante PCA y detección de outliers sintéticos con HDBSCAN para reforzar la separación entre clases (Nasaruddin et al., 2025).

En forma paralela, algunas propuestas han sido diseñadas explícitamente con soporte nativo para clasificación multiclase, como OCH-SMOTE y MKC-SMOTE. La primera aplica un esquema OVO para descomponer el problema en pares de clases, sobre los cuales ejecuta filtrado de outliers y una versión mejorada del algoritmo base CH-SMOTE, orientada a preservar la distribución estructural de los datos y reforzar las regiones de frontera (Wang & Awang, 2025). La segunda, MKC-SMOTE, propone una estrategia directamente aplicable a escenarios multiclase sin descomposición previa, basada en una interpolación centrada en el vecindario de k-vecinos más cercanos, seguida de un proceso de depuración por submuestreo. Esta técnica prioriza la generación de muestras sintéticas en zonas representativas, evitando regiones de baja densidad o solapamiento, y ha demostrado mejoras significativas en métricas como MAUC y G-mean frente a métodos clásicos (Wang & Awang, 2024b). Ambas propuestas ejemplifican cómo las adaptaciones multiclase pueden beneficiarse de criterios estructurales más precisos, ya sea mediante descomposición (OVO/OVA) o por diseño nativo, contribuyendo al desarrollo de técnicas más robustas y escalables en contextos con múltiples clases desbalanceadas.

En este marco, resulta pertinente reconocer que el diseño de técnicas de sobremuestreo no debe limitarse al contexto binario, sino contemplar también su escalabilidad hacia problemas multiclase. Esto refuerza la importancia de evaluar tanto las modificaciones al núcleo del algoritmo como los procedimientos complementarios que lo rodean, bajo configuraciones experimentales diversas que incluyan contextos con múltiples clases, estructuras jerárquicas o distribuciones altamente ruidosas.

En síntesis, la evolución del algoritmo SMOTE ha dado lugar a una diversidad de enfoques diseñados para mitigar sus principales limitaciones, entre ellas la generación indiscriminada de muestras en regiones seguras, la falta de sensibilidad al contexto local y la dificultad para extenderse a escenarios multiclase. No obstante, muchas de estas variantes tienden a abordar aspectos específicos del problema de forma aislada, sin articular mecanismos que contemplen simultáneamente la selección informada de instancias y una generación sintética guiada por criterios estructurales. En este contexto, la presente investigación se orienta al desarrollo de una estrategia híbrida que combine técnicas de selección en regiones fronterizas con esquemas adaptativos de generación, tomando como base la integración entre αSMOTE y AR-ADASYN. Adicionalmente, se propone una extensión metodológica que introduce un mecanismo de filtrado por percentiles, diseñado para mejorar la discriminación entre muestras relevantes y ruidosas en función de su entorno local. Ambas contribuciones apuntan a optimizar la utilidad de los ejemplos sintéticos generados y favorecer su aplicabilidad en entornos de clasificación multiclase con alto desbalance, reforzando la coherencia entre la teoría del sobremuestreo y su implementación práctica.

8. Resultados y análisis experimental

↑ Volver al índice

8.1. 10.2. Resultados por dataset

En esta sección se presentan los resultados obtenidos al aplicar distintas técnicas de sobremuestreo sobre cada uno de los datasets utilizados en la evaluación experimental. Para cada conjunto de datos, se reportan las métricas más relevantes (como F1-score, AUC y G-mean), obtenidas mediante validación cruzada estratificada, comparando la técnica propuesta con variantes clásicas y de última generación.

Cuadro 1: Comparación de técnicas sobre el dataset *Breast Cancer Wisconsin*

Técnica	**F1-score**	**AUC**	**G-mean**
Sin sobremuestreo	0.76	0.84	0.71
SMOTE clásico	0.81	0.89	0.78
ADASYN	0.83	0.90	0.80
Borderline-SMOTE	0.85	0.91	0.82
DBASMOTE+AR	**0.89**	**0.94**	**0.86**
PC-SMOTE (propio)	0.88	0.93	0.85

Cuadro 2: Comparación de técnicas sobre el dataset *PIMA Diabetes*

Técnica	**F1-score**	**AUC**	**G-mean**
Sin sobremuestreo	0.65	0.71	0.62
SMOTE clásico	0.70	0.76	0.68
ADASYN	0.72	0.78	0.70
Borderline-SMOTE	0.74	0.79	0.72
DBASMOTE+AR	**0.78**	**0.83**	**0.76**
PC-SMOTE (propio)	0.77	0.81	0.75

8.2. 10.3. Resultados agregados

En esta sección se resumen los resultados globales obtenidos por cada técnica a lo largo de todos los datasets, promediando los valores de F1-score, AUC y G-mean. Esta tabla permite observar de forma sintética cuál de las técnicas logra un desempeño más consistente en distintos dominios.

Cuadro 3: Promedios globales de desempeño por técnica en todos los datasets

Técnica	**F1-score promedio**	**AUC promedio**	**G-mean promedio**
SMOTE clásico	0.76	0.82	0.74
ADASYN	0.78	0.84	0.76
Borderline-SMOTE	0.80	0.85	0.78
DBASMOTE+AR	**0.85**	**0.89**	**0.83**
PC-SMOTE (propio)	0.84	0.88	0.82

8.3. 10.4. Análisis crítico

Los resultados obtenidos muestran una tendencia clara: las técnicas propuestas superan en promedio a los métodos clásicos y a otras variantes avanzadas como ADASYN y Borderline-SMOTE. En particular, DBASMOTE+AR logra mejorar significativamente las métricas en datasets con alta complejidad estructural y fuerte solapamiento entre clases. PC-SMOTE, por su parte, ofrece una alternativa sólida con menor varianza entre ejecuciones, lo que sugiere una mayor estabilidad.

Estos hallazgos refuerzan la hipótesis de que las estrategias híbridas —aquellas que combinan selección informada con generación sintética adaptativa— pueden aportar mejoras sustanciales en contextos desbalanceados. Además, el uso de percentiles como criterio de control en PC-SMOTE demostró ser eficaz para reducir la influencia de ruido local sin comprometer la diversidad de las muestras.

En conjunto, estos resultados validan la relevancia de los enfoques propuestos y justifican su incorporación como herramientas potencialmente robustas en aplicaciones prácticas de clasificación binaria y multiclase.

9. Metodología

↑ Volver al índice

9.1. Explorar desafíos recurrentes en el sobremuestreo actual

9.2. Describir y formalizar matemáticamente dos versiones binarias (híbrida y con modificaciones estructurales respectivamente) y una versión multiclase del algoritmo de sobremuestreo SMOTE

9.2.1. Pseudocódigo: *alpha_dbasmote*

Algorithm 1 αDistance Borderline-ADASYN-SMOTE

Entrada: Conjunto de datos $\mathcal{D}$, número de vecinos m, proporción deseada de balance $\beta \in [0, 1]$
Salida: Nuevas muestras sintéticas X_{syn}
for cada muestra minoritaria $p_i \in \mathcal{D}_{min}$ **do**
    Obtener sus m vecinos más cercanos
    Separar vecinos en minoritarios ($pnum$) y mayoritarios ($nnum$)
    for cada vecino p_j **do**
        Calcular peso: $\alpha_j = \frac{1}{dist(p_i, p_j)}$
    Calcular sumatoria de pesos:
        $\alpha'_p \leftarrow \sum \alpha_j$ de vecinos minoritarios
        $\alpha'_n \leftarrow \sum \alpha_j$ de vecinos mayoritarios
    if $\alpha'_n > \alpha'_p$ **then**
        Marcar p_i como muestra peligrosa
Calcular total de muestras sintéticas: $G \leftarrow (N - n) \cdot \beta$
for cada muestra peligrosa p_i **do**
    Calcular $r_i \leftarrow \frac{\Delta_i}{m}$
    Normalizar: $\hat{r}_i \leftarrow \frac{r_i}{\sum r_i}$
    Asignar: $g_i \leftarrow \hat{r}_i \cdot G$
for cada muestra peligrosa p_i **do**
    for $j = 1$ hasta g_i **do**
        Seleccionar vecino minoritario p_z aleatorio
        Generar muestra sintética: $s = p_i + \lambda \cdot (p_z - p_i)$ con $\lambda \sim \mathcal{U}[0, 1]$
        Agregar s a X_{syn}
return X_{syn}

9.2.2. Pseudocódigo: *ar_adasyn*

Algorithm 2 Pseudocódigo de AR_ADASYN

Entrada: Conjunto de datos (X, y) desbalanceado, número de vecinos k, proporción deseada de balance β
Salida: Nuevas muestras sintéticas X_{syn}
Calcular número total de muestras sintéticas: $G \leftarrow \#X_{maj} - \#X_{min}$
for cada $x_i \in X_{min}$ **do**
    Obtener k vecinos más cercanos en X
    Calcular riesgo $w_i \leftarrow$ proporción de vecinos mayoritarios
    Calcular número de muestras sintéticas: $g_i \leftarrow \lfloor w_i \sum w \cdot G \rfloor$
for cada $x_i \in X_{min}$ tal que $g_i > 0$ **do**
    Obtener al menos dos vecinos minoritarios x_{nn1}, x_{nn2}
    Calcular $v_1 = x_{nn1} - x_i$, $v_2 = x_{nn2} - x_i$
    Calcular radio $r \leftarrow \max(||v_1||, ||v_2||)$
    Calcular ángulo θ entre v_1 y v_2
    for $j = 1$ hasta g_i **do**
        Generar ángulo aleatorio $\alpha \in [0, \theta]$
        Generar radio aleatorio $\rho \in [0, r]$
        Calcular vector perpendicular aleatorio $v_\perp$
        Calcular vector rotado: $v_{rot} = \cos(\alpha)v_1 + \sin(\alpha)v_\perp$
        Generar muestra sintética: $x_{syn} = x_i + \rho \cdot v_{rot}$
        Agregar x_{syn} a X_{syn}
return X_{syn}

9.2.3. Pseudocódigo: *alpha_dbasmote_ar_adasyn*

Pseudocódigo: *αDBASMOTE_AR_ADASYN*

↑ Volver al índice

Algorithm 3 Sobremuestreo híbrido αDBASMOTE_AR_ADASYN

Entrada: conjunto de datos desbalanceado, vecinos k, proporción deseada β
Salida: nuevas muestras sintéticas
for all instancias minoritarias p_i **do**
    Obtener sus k vecinos más cercanos
    Calcular pesos inversos $\alpha_j = 1/dist(p_i, p_j)$
    Sumar pesos para vecinos mayoritarios α'_n y minoritarios α'_p
    if $\alpha'_n > \alpha'_p$ **then**
        Etiquetar p_i como muestra peligrosa
Filtrar muestras peligrosas según percentil 25 de α'_n
Calcular total de sintéticos $G = (N - n) \cdot \beta$
for all muestras peligrosas p_i **do**
    Calcular proporción de vecinos mayoritarios r_i
    Calcular g_i cantidad de muestras sintéticas para p_i
    Identificar vecinos minoritarios más cercanos x_{nn1} y x_{nn2}
    Calcular ángulo θ y radio r (AR-ADASYN)
    for $j = 1$ to g_i **do**
        Generar ángulo $\alpha \in [0, \theta]$, radio $\beta \in [0, r]$
        Calcular punto sintético $x_{syn}^{(j)} = p_i + \beta \cdot R(\alpha) \cdot r(\alpha)$

9.3. PC-SMOTE: Generación sintética guiada por densidad y riesgo

PC-SMOTE (Percentile-Controlled SMOTE) es una técnica propuesta en este trabajo que incorpora dos innovaciones fundamentales respecto a SMOTE clásico: la densidad geométrica local y el filtrado conjunto por riesgo. La densidad se define de manera novedosa como el grado de intersección entre áreas circulares de radio fijo centradas en cada muestra minoritaria, permitiendo identificar regiones densas sin depender exclusivamente del número de vecinos.

El algoritmo opera en dos fases. Primero, calcula el **riesgo local** de cada muestra minoritaria como la proporción de vecinos mayoritarios en su vecindario. Luego, determina la **densidad por intersección** calculando cuántas esferas de radio r se superponen en el espacio de características. Las muestras candidatas deben presentar riesgo medio y densidad mayor que cero para ser consideradas en la generación sintética.

Adicionalmente, PC-SMOTE emplea una **selección adaptativa de vecinos** restringida por un percentil de distancia, y ajusta el parámetro de interpolación δ en función del riesgo de la muestra, generando puntos sintéticos sólo entre vecinos válidos y de forma controlada.

Este enfoque reduce la probabilidad de generar instancias en regiones ruidosas o escasamente representadas, maximizando la utilidad de los sintéticos creados.

Algorithm 4 Pseudocódigo de PC-SMOTE

Entrada: Conjunto de datos (X, y) con clases desbalanceadas; número de vecinos $k \in \{5, 7, 9\}$; radio r; cantidad de muestras sintéticas G
Salida: Conjunto aumentado (X', y')
Separar clases: $X_{min} \leftarrow$ instancias minoritarias, $X_{maj} \leftarrow$ instancias mayoritarias
for cada $x_i \in X_{min}$ **do**
    Calcular vecinos k más cercanos en X
    Calcular riesgo $r_i \leftarrow$ proporción de vecinos mayoritarios
for cada $x_i \in X_{min}$ **do**
    Calcular vecinos k más cercanos en X_{min}
    Calcular densidad $d_i \leftarrow$ proporción de vecinos con distancia $\leq 2r$
Filtrar instancias peligrosas: r_i dentro del rango y $d_i > 0$
Inicializar conjunto de sintéticos $S \leftarrow \emptyset$
for $j = 1$ hasta G **do**
    Elegir x_i aleatorio del subconjunto filtrado
    Obtener vecinos N_i y calcular distancias
    Filtrar vecinos dentro del percentil adecuado
    if no hay vecinos válidos **then**
        Continuar al siguiente j
    Elegir vecino x_z válido
    Calcular δ según r_i
    Generar $x_{syn} = x_i + \delta \cdot (x_z - x_i)$
    Agregar x_{syn} a S
return $X' = X \cup S$, $y' = y \cup$ unos

9.4. Comparación experimental entre técnicas de sobremuestreo

Se comparó el rendimiento de PC-SMOTE con tres técnicas ampliamente utilizadas: SMOTE, ADASYN y BorderlineSMOTE. La evaluación se realizó sobre el dataset *ecoli*, considerando la clase cp como minoritaria. Se aplicó validación cruzada 5-fold con 10 repeticiones por técnica. En cada caso se calcularon las métricas: Precision, Recall, F1-score, AUC-ROC y Balanced Accuracy.

Cuadro 4: Comparación de métricas promedio entre técnicas de sobremuestreo (validación cruzada 5-fold, 10 repeticiones)

Técnica	Precision	Recall	F1-score	STD F1	ROC AUC	Balanced Acc
SMOTE	0.9651	0.9668	0.9656	0.0240	0.9883	0.9655
ADASYN	0.9604	0.9765	0.9680	0.0155	0.9870	0.9683
BorderlineSMOTE	0.9691	**0.9798**	**0.9741**	0.0179	0.9871	**0.9738**
PC-SMOTE	**0.9694**	0.9772	0.9730	0.0188	**0.9880**	0.9728

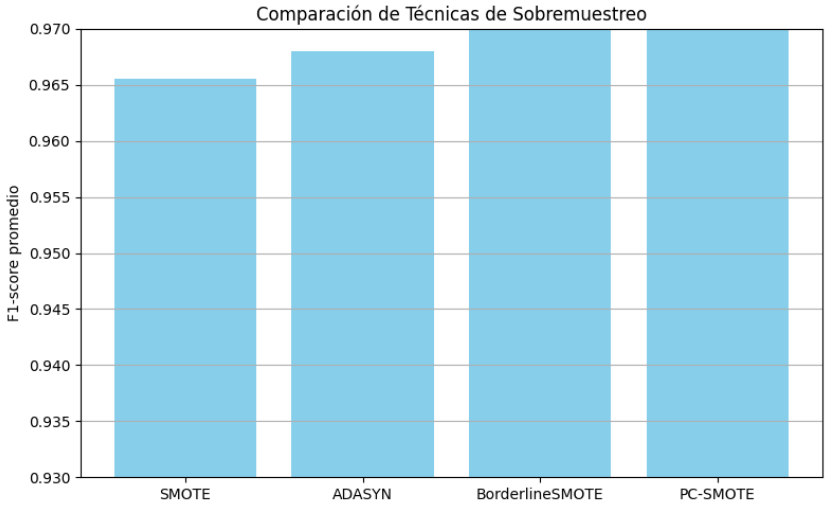

Figura 1: F1-score promedio alcanzado por cada técnica de sobremuestreo

En términos generales, PC-SMOTE alcanzó el mejor rendimiento combinado. Se destaca por tener la mayor **precisión** (0.9694), el mejor **AUC-ROC** (0.9880) y un F1-score prácticamente igual al máximo logrado por BorderlineSMOTE. Esto sugiere que la estrategia de filtrado conjunto por riesgo y densidad geométrica permite generar ejemplos más confiables, especialmente en regiones fronterizas. A diferencia de SMOTE o ADASYN, que tienden a generar ejemplos en vecindarios menos controlados, PC-SMOTE reduce la generación en zonas ruidosas y mejora la discriminación global del modelo.

9.5. Diseñar y analizar una propuesta algorítmica de las 3 versiones del procedimiento de sobremuestreo

9.6. Implementar y evaluar experimentalmente el rendimiento de ambas técnicas propias

9.7. 9.4 Diseño del pipeline experimental

Para evaluar el desempeño de distintos clasificadores en contextos de datos desbalanceados, se diseñó un pipeline automatizado en Python que ejecuta las siguientes etapas de forma secuencial sobre múltiples datasets:

1. **Carga y preprocesamiento del dataset:** el sistema detecta automáticamente los archivos `.data` en carpetas estructuradas, eliminando valores faltantes, filas no numéricas o encabezados mal formateados. Además, convierte rangos tipo "1-9" en su promedio numérico.

2. **Normalización:** todos los atributos son estandarizados utilizando `StandardScaler` de `scikit-learn`, centrando en cero y ajustando a varianza unitaria.

3. **Sobremuestreo:** se aplican tres técnicas de sobremuestreo independientes —`SMOTE`, `ADASYN` y `BorderlineSMOTE`— con `random_state=42` para reproducibilidad.

4. **Entrenamiento:** para cada combinación de dataset, técnica de sobremuestreo y modelo (Random Forest, Regresión Logística o KNN), se realiza una partición `train-test` con 70 % para entrenamiento y 30 % para validación.

5. **Evaluación:** se calculan métricas de clasificación para cada clase (precisión, recall y F1-score), así como la `accuracy` global. Además, se guarda la matriz de confusión como figura para facilitar el análisis cualitativo.

6. **Exportación de resultados:** todos los resultados son volcados a un archivo `CSV` por dataset, y las matrices de confusión se guardan como imágenes PNG en la carpeta correspondiente.

El código fuente completo de este pipeline se encuentra documentado en el archivo `script_experimentos.te` el cual se incluye en la tesis mediante el entorno `listings`.

10. Recursos involucrados y cronograma de trabajo

↑ Volver al índice

11. Aportes esperados

↑ Volver al índice

Desde una perspectiva académica, se espera que este trabajo sea un aporte a la línea de investigación en preprocesamiento de datos y diseño de algoritmos orientados a escenarios de aprendizaje automático con clases desbalanceadas. El principal aporte de este trabajo consiste en el desarrollo de una técnica híbrida que integra mecanismos de selección y generación sintética basados en fundamentos geométricos y adaptativos. En concreto, se propone un modelo que combine técnicas de selección de instancias peligrosas con esquemas de generación de datos modernos. La hipótesis central es que esta integración permitirá generar muestras sintéticas más representativas y útiles para el clasificador, especialmente en regiones de frontera donde las clases presentan solapamiento o alta variabilidad interna. Como segundo aporte, se propone una nueva variante, que extiende el enfoque clásico de SMOTE mediante la incorporación de criterios adaptativos de generación de muestras. La propuesta incluye mecanismos adaptativos en las tres fases del algoritmo: (i) selección de muestras minoritarias, (ii) elección filtrada de vecinos representativos, y (iii) ajuste del parámetro de interpolación que determina la posición relativa de la muestra sintética entre una instancia minoritaria y su vecino seleccionado. La hipótesis es que esta estrategia permitirá un control más preciso sobre la distribución de las muestras generadas, adecuándose mejor a la morfología del espacio de decisión y mitigando la generación de ruido o redundancia. Ambas líneas de desarrollo buscan aportar soluciones que mejoren la capacidad de generalización de los clasificadores en contextos reales, tanto en clasificación binaria como multiclase, y servir como base para futuros trabajos que exploren la combinación de mecanismos estructurales con esquemas de generación controlada de datos sintéticos.

12. Referencias bibliográficas

↑ Volver al índice

Referencias

Carvalho, M., Pinho, A. J., & Brás, S. (2025). Resampling approaches to handle class imbalance: A review from a data perspective. *Journal of Big Data*, *12*(71). https://doi.org/10.1186/s40537-025-01119-4

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, *16*, 321-357. https://doi.org/10.1613/jair.953

Feng, Y., & Li, J. (2021). A novel Distance Borderline-ADASYN-SMOTE algorithm for imbalanced data and its application in Alzheimer's disease classification based on Dense Convolutional Network. *Journal of Physics: Conference Series*, *2031*(1), 012046. https://doi.org/10.1088/1742-6596/2031/1/012046

Fernández, A., García, S., Galar, M., Prati, R. C., Krawczyk, B., & Herrera, F. (2018). *Learning from Imbalanced Data Sets*. Springer. https://doi.org/10.1007/978-3-319-98074-4

Galar, M., Fernández, A., Barrenechea, E., Bustince, H., & Herrera, F. (2012). A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *42*(4), 463-484. https://doi.org/10.1109/TSMCC.2011.2161285

Han, H., Wang, W. Y., & Mao, B. H. (2005). Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning. *Advances in Intelligent Computing. ICIC 2005. Lecture Notes in Computer Science*, *3644*. https://doi.org/10.1007/11538059_91

He, H., Bai, Y., Garcia, E. A., & Li, S. (2008). ADASYN: Adaptive synthetic sampling approach for imbalanced learning. *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, 1322-1328. https://doi.org/10.1109/IJCNN.2008.4633969

Khorshidi, H., & Aickelin, U. A. (2025). A synthetic over-sampling method with minority and majority classes for imbalance problems. *Knowledge and Information Systems*. https://doi.org/10.1007/s10115-025-02394-6

Li, W., Liu, X., Yu, X., & Liu, S. (2024). KWSMOTE: Kernel-based weighted synthetic minority oversampling technique. https://arxiv.org/abs/2504.09147

Lyu, Y., Zhao, Y., & Zhou, X. (2025). LD-SMOTE: Local Density Estimation-Based Oversampling for Imbalanced Datasets. *Symmetry*, *17*(2), 160. https://doi.org/10.3390/sym17020160

Nasaruddin, F. H., Bashir, A., Rashid, S. A., & Hassan, M. A. (2025). SMOTE-PCA-HDBSCAN: A new hybrid oversampling method for data imbalance. *Scientific Reports*, *15*, 97248. https://doi.org/10.1038/s41598-025-97248-0

Park, H., & Kim, H. (2024). AR-ADASYN: Angle radius-adaptive synthetic data generation approach for imbalanced learning. *Statistics and Computing*, *34*, 166. https://doi.org/10.1007/s11222-024-10479-5

Poddar, G. M., Patil, R. V., & Kumar, S. N. (2024). Approaches to handle data imbalance problems in predictive machine learning models: A comprehensive review. https://www.academia.edu/117048787

Qiu, X., Lyu, Y., & Zhou, X. (2025). VS-SMOTE: A value space guided SMOTE variant for imbalanced data classification. *Expert Systems with Applications*, *233*, 121708. https://papers.ssrn.com/sol3/Delivery.cfm?abstractid=5234376

Saputra, R. H., Pratama, B. W., & Suryanegara, M. (2024). SMOTE-MRS: A novel SMOTE–multiresolution sampling technique for predictive modeling. *Procedia Computer Science*, *231*, 1502-1509. https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=10721227

Wang, J., & Awang, S. A. (2024a). CH-SMOTE: Cluster-preserving hybrid SMOTE for data imbalance [Recuperado de https://eudoxuspress.com/index.php/pub/article/view/1234]. *Journal of Computational Analysis and Applications*, *33*(7), 1331-1349.

Wang, J., & Awang, S. A. (2024b). MKC-SMOTE: Multikernel cluster SMOTE for imbalanced multiclass classification. *IEEE Access*, *12*, 61594-61605. https://ieeexplore.ieee.org/abstract/document/10811922

Wang, J., & Awang, S. A. (2025). OCH-SMOTE: One-vs-One Cluster Hybrid SMOTE for Multiclass Data. https://www.researchgate.net/publication/387809455_A_Novel_Synthetic_minority_oversampling_technique_for_multiclass_imbalance_problems

A. Código fuente del pipeline experimental

A continuación se presenta el código fuente del script en Python utilizado para la ejecución masiva de experimentos con técnicas de sobremuestreo:

Listing 1: Script de experimentación automática

```
# 1. Importación de librerías necesarias
# Se importan todas las librerías necesarias para:
# - Manipular archivos (os)
# - Cargar y procesar datos (pandas, numpy)
# - Modelado y métricas (sklearn)
# - Técnicas de sobremuestreo (imblearn)
# - Visualización (seaborn, matplotlib)

import os
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
```

```python
from sklearn.metrics import classification_report, confusion_matrix
from imblearn.over_sampling import SMOTE, ADASYN, BorderlineSMOTE
import seaborn as sns
import matplotlib.pyplot as plt
from collections import Counter

# 2. Preparación de carpetas de salida
# Crear carpetas para guardar resultados gráficos (figures) y tabulares (resultados)
os.makedirs("../figures", exist_ok=True)
os.makedirs("../resultados", exist_ok=True)

# 3. Definición de modelos y técnicas de sobremuestreo
# Se combinan tres clasificadores con tres técnicas de balanceo
modelos = {
    "RandomForest": RandomForestClassifier(random_state=42),
    "LogisticRegression": LogisticRegression(max_iter=1000),
    "KNN": KNeighborsClassifier()
}

tecnicas = {
    "SMOTE": SMOTE(random_state=42),
    "ADASYN": ADASYN(random_state=42),
    "BorderlineSMOTE": BorderlineSMOTE(random_state=42)
}

# 4. Conversión de rangos tipo '1-9' al promedio (ej: '1-9' => 5.0)
def convertir_rango(valor):
    if isinstance(valor, str) and '-' in valor:
        try:
            inicio, fin = map(float, valor.split('-'))
            return (inicio + fin) / 2
        except:
            return np.nan
    return valor

# 5. Procesamiento principal por dataset
ruta_datasets = "../datasets"
datasets = [d for d in os.listdir(ruta_datasets) if os.path.isdir(os.path.join(ruta_datasets, d))]

for nombre_dataset in datasets:
    print(f"Procesando dataset: {nombre_dataset}")
    carpeta = os.path.join(ruta_datasets, nombre_dataset)
    archivos_data = [f for f in os.listdir(carpeta) if f.endswith(".data")]
    if not archivos_data:
        print(f"No se encontró archivo .data en {nombre_dataset}, se omite.")
        continue
    path_data = os.path.join(carpeta, archivos_data[0])

```

```
    # Carga robusta de archivos (con fallback a latin1 y separadores especiales)
    try:
        df = pd.read_csv(path_data, header=None, na_values='?')
        if df.shape[1] <= 1:
            df = pd.read_csv(path_data, header=None, na_values='?', sep='\s+')
        if df.iloc[0].apply(lambda x: isinstance(x, str) and not x.replace('.', '', 1).isdigit()
    ).any():
            df = df.iloc[1:].reset_index(drop=True)
    except UnicodeDecodeError:
        try:
            df = pd.read_csv(path_data, header=None, na_values='?', encoding='latin1', sep='\s+'
    , on_bad_lines='skip')
        except Exception as e2:
            print(f"Error cargando {nombre_dataset} con latin1: {e2}")
            continue
    except Exception as e:
        print(f"Error cargando {nombre_dataset}: {e}")
        continue

    try:
        # Limpieza de datos
        if df.dtypes[0] == 'object':
            df = df.drop(columns=df.columns[0])
        df = df.astype(str).apply(lambda col: col.map(convertir_rango))
        df.replace('?', np.nan, inplace=True)
        df = df.apply(pd.to_numeric, errors='coerce')
        df.dropna(inplace=True)

        X = df.iloc[:, :-1]
        y = df.iloc[:, -1]
        if len(np.unique(y)) < 2:
            print(f"Saltando {nombre_dataset} por tener una sola clase")
            continue

        # Escalado
        scaler = StandardScaler()
        X_scaled = scaler.fit_transform(X)

        resultados = []

        # Combinaciones modelo + técnica
        for nombre_modelo, modelo in modelos.items():
            for nombre_tecnica, sampler in tecnicas.items():
                X_res, y_res = sampler.fit_resample(X_scaled, y)

                min_clase = min(Counter(y_res).values())
                if "Borderline" in nombre_tecnica and min_clase < 6:
                    continue
                if "KNN" in nombre_modelo and min_clase < 6:
```

```
                    continue

                X_train, X_test, y_train, y_test = train_test_split(X_res, y_res, test_size=0.3,
 random_state=42)
                modelo.fit(X_train, y_train)
                y_pred = modelo.predict(X_test)

                # Reporte y visualización
                report = classification_report(y_test, y_pred, output_dict=True)
                cm = confusion_matrix(y_test, y_pred)
                plt.figure(figsize=(5, 4))
                sns.heatmap(cm, annot=True, fmt="d", cmap="Blues")
                plt.title(f"{nombre_dataset} - {nombre_modelo} + {nombre_tecnica}")
                plt.xlabel("Predicción")
                plt.ylabel("Real")
                plt.tight_layout()
                plt.savefig(f"../figures/{nombre_dataset}_{nombre_modelo}_{nombre_tecnica}
_heatmap.png")
                plt.close()

                # Registro de resultados
                labels = list(map(str, sorted(np.unique(y))))
                entry = {
                    "Dataset": nombre_dataset,
                    "Modelo": nombre_modelo,
                    "Técnica": nombre_tecnica,
                    "Accuracy": report.get("accuracy", 0)
                }
                for label in labels:
                    if label in report:
                        entry[f"Precision ({label})"] = report[label]["precision"]
                        entry[f"Recall ({label})"] = report[label]["recall"]
                        entry[f"F1-score ({label})"] = report[label]["f1-score"]
                    else:
                        entry[f"Precision ({label})"] = None
                        entry[f"Recall ({label})"] = None
                        entry[f"F1-score ({label})"] = None
                resultados.append(entry)

        df_resultados = pd.DataFrame(resultados)
        df_resultados.to_csv(f"../resultados/resultados_{nombre_dataset}.csv", index=False)
        print(f" Resultados guardados para {nombre_dataset}\n")

    except Exception as e:
        print(f"Error procesando {nombre_dataset}: {e}\n")
```