Universidad Autónoma de Sinaloa

facultad de informática culiacán
facultad de ciencias de la tierra y el espacio

maestría en ciencias de la información

ANÁLISIS DE LA ESTRUCTURA DE CLASES MINORITARIAS EN PROBLEMAS DE DESBALANCE DE CLASES

Tesis

como requisito para obtener el grado de

MAESTRO EN CIENCIAS DE LA INFORMACIÓN

presenta

Cecilia Angulo Domínguez

directores de tesis
dr. arturo yee rendón
MC. gerardo beltrán gutiérrez

culiacán, sinaloa, méxico a 27 de Enero de 2022

Universidad Autónoma de Sinaloa

facultad de informática culiacán
facultad de ciencias de la tierra y el espacio

maestría en ciencias de la información

ANÁLISIS DE LA ESTRUCTURA DE CLASES MINORITARIAS EN PROBLEMAS DE DESBALANCE DE CLASES

Tesis

como requisito para obtener el grado de

MAESTRO EN CIENCIAS DE LA INFORMACIÓN

presenta

Cecilia Angulo Domínguez

directores de tesis
dr. arturo yee rendón
MC. gerardo beltrán gutiérrez

culiacán, sinaloa, méxico a 27 de Enero de 2022

Dedico mi presente tesis.

A mis padres:

Juan Cecilio Angulo Armenta y Lourdes Domínguez Castro por todo su apoyo y enseñanzas de vida que me han brindado siempre.

A mi esposo:

Emmanuel Felix Valenzuela por su paciencia, su apoyo incondicional, cariño y amor que me ha brindado.

A mi hermana:

Maria de Lourdes Angulo Domínguez, por motivarme a seguir estudiando y ser un gran ejemplo a seguir.

Agradecimientos

A la Universidad Autónoma de Sinaloa y al Posgrado en Ciencias de la Información por ser mi casa de estudios y ayudarme a crecer de manera académica y personal.

Al CONACyT por otorgarme la beca con la cual pude dedicarme por completo a mis estudios de maestría.

A mis directores de tesis el Dr. Arturo Yee Rendón y Mc. Gerardo Beltrán Gutiérrez, por compartir sus conocimientos, comprensión y apoyo incondicional durante el trabajo de tesis, así como por guiarme en mi formación académica y personal.

A los profesores del posgrado, en especial el Dr. Inés Fernando Vega López y al Dr. Jorge Adalberto Navarro Castillo que influyeron en gran medida en mi formación académica.

Resumen

El Aprendizaje de Máquina es una subárea de la Inteligencia Artificial, que se encarga del desarrollo de algoritmos computacionales, que tienen la capacidad de inferir y deducir el conocimiento a partir de un conjunto de datos, con el fin de obtener modelos capaces de evaluar datos similares de los cuales no se tenia información previa, como ocurre con modelos de regresión y clasificación.

En el mundo real, muchos conjuntos de datos presentan el problema de desbalance de clases, lo que significa que una de las clases contiene un número mucho menor de instancias que la(s) otra(s) clase(s). En particular, en los modelos de clasificación, este problema de desbalance de clases pudiera provocar que la capacidad de los modelos para inferir y deducir sean pobres, debido a que los modelos están sesgados hacia las clases mayoritarias, clasificando incorrectamente ejemplos de las clases minoritarias. Sin embargo, el pobre desempeño de los modelos, sobre los ejemplos de las clases minoritarias, podría ser contribuido no solo al problema de desbalance de clases, sino también a una variedad de

factores, por ejemplo, el ruido de datos, traslape de instancias entre las clases, casos raros y pequeñas disyunciones [1]. Esto significa que el problema de desbalance de clases puede no ser un problema en sí mismo y contrarrestar el desbalance de clases disminuyendo o aumentando instancias en las clases de manera arbitraría, no siempre conducirá a un correcto desempeño de los modelos de clasificación.

Es importante estudiar la estructura en clases con desbalance, es decir, la distribución de las instancias en el espacio de características de las clases y su impacto en el desempeño de los modelos de clasificación. En esta investigación, se propone un análisis de la estructura de la clase minoritaria para identificar subgrupos de instancias mediante la técnica de los k-vecinos. Además, se proponen dos técnicas de creación de instancias sintéticas que permitan mitigar los problemas relacionados con el desbalance de clases y otros factores a nivel de datos en problemas de clasificación binaria. Para medir el desempeño de las técnicas propuestas, se crearon modelos de clasificación que utilizan las técnicas de creación de instancias propuestas. Estos modelos se compararon con modelos que utilizan técnicas del estado del arte que mitigan el problema de desbalance de clases. Como consecuencia de esto, los resultados experimentales mostraron que nuestras técnicas propuestas mitigaron el problema de desbalance de clases, permitiendo que los modelos de clasificación obtuvieran instancias representativas de los conjuntos de datos y un alto desempeño.

Abstract

Machine learning is a subarea of Artificial Intelligence (AI) that provides computational algorithms the ability to infer and deduce knowledge from a dataset. Machine learning focuses on building predictive models that can evaluate similar data with no previous information, as occurs with regression and classifications models.

In the real world, many datasets present the class imbalance problem. An imbalanced dataset is a dataset where there are many more instances for one class than others. In classification models, this class imbalance problem could cause that the models' ability to infer and deduce to be poor, because the models will be biased toward one or more of the classes with frequent instances against other classes with infrequent instances, incorrectly classifying examples of minority classes. However, the poor performance of the models, on the minority class instances, could be contributed not only to the class imbalance problem, but also to a variety of factors such as data noise, an overlap of instances between classes, cases rare, and small disjunctions[1]. This means that the

class imbalance problem may not be unique in itself and preventing the class imbalance problem decreasing or increasing instances in the classes in an arbitrary, way will not always lead to correct performance of the classification models.

It is important to study the structure of the class imbalance, that is, the distribution of instances in space the characteristics of classes and their impact on the performance of the classification models. In this investigation, we identify subgroups of minority class instances by analyzing the neighborhood, using a k-Nearest Neighbor approach. We propose two instances creation techniques that allows mitigating class imbalance problem and other factors at the data level in binary classification problems. We build an ensemble classification model that used the proposed instances creation technique to be compared with the state-of-art techniques. Experimental results show that the proposed instances creation techniques remove class imbalance problem, allowing that the classification model obtained representative instances of data sets and high performance.

Índice general

Lista de Figuras

Lista de Tablas

Capítulo 1

Introducción

1.1. Descripción del problema

En diversas áreas de la ciencia como la astronomía, medicina, biología, ingeniería civil, física, así como en la vida cotidiana, surge la necesidad de clasificar objetos de una manera rápida y automatizada. Los objetos se clasifican dentro de clases definidas previamente, a esto, en el área de aprendizaje de máquina, se le conoce como clasificación supervisada[2].

La tarea de clasificación se vuelve difícil de realizar cuando en los conjuntos de datos se presenta el problema de desbalance de clases. El desbalance de clases es un obstáculo para el aprendizaje de los modelos de clasificación. Durante el entrenamiento de los modelos de clasificación, éstos tienden a sesgarse hacia la clase mayoritaria y obtener una

clasificación errónea para las instancias de las clases minoritarias. Aunque, se han propuesto diversos enfoques para mejorar las técnicas de clasificación, aún es un problema extenso a investigar. En la literatura especializada, existen diferentes estudios sobre el desbalance de clases, los cuales proponen técnicas a nivel de datos [3, 4], a nivel de algoritmos [5, 6] e híbridos [7, 8] para contrarrestar este problema.

El problema de desbalance de clases en un problema de clasificación binaria, se define en términos de la razón de desbalance (R_D), ver Ecuación 1.1:

$$R_D = \frac{|N|}{|P|} \tag{1.1}$$

Donde R_D indica el grado de desbalance que hay en un conjunto de instancias, entre más alejado este del valor 1 significa que el conjunto de instancias tiene un alto grado de desbalance [9]. $|P|$ hace referencia a la cardinalidad del conjunto de instancias de la clase minoritaria (clase positiva) y $|N|$ es la cardinalidad del conjunto de instancias en la clase mayoritaria (clase negativa).

Un problema de clasificación binaria consiste en distinguir entre instancias de dos clases diferentes. Formalmente, un problema de clasificación binaria se define de la siguiente manera:

Sea $S = \{(x_i, y_i),\ i = 1,...,m\}$ un conjunto de m instancias etiquetadas, donde $x_i \in \mathbb{R}^n$ y $y_i \in \{0, 1\}$. El objetivo es encontrar una función

$f : \mathbb{R}^n \rightarrow \{0, 1\}$, tal que minimice el error de clasificación $f(x_i) \neq y_i$, $i = 1,...,m$.

Además del problema de desbalance de clases, existen otros factores que degradan el desempeño de los modelos de clasificación. Por ejemplo, los diferentes subgrupos de instancias que se forman en las clases, el traslape de instancias entre clases, el ruido en la obtención de las instancias, entre otros. Cuando estos factores ocurren junto con el desbalance de clases afectan el desempeño de los modelos en la clasificación. Particularmente, en la clasificación de las instancias de la clase minoritaria [10, 11, 12].

Como se mencionó anteriormente, además del problema de desbalance, existen otros factores que degradan el desempeño de los modelos de clasificación. Particularmente, la presencia de instancias de ambas clases que comparten una región común en el espacio de características. Estas instancias tienen similitudes en sus valores de las características, pero pertenecen a clases diferentes, a esto se le conoce como traslape de clases. Una disyunción es un subgrupo de instancias de la clase minoritaria aisladas, que pudieran estar en una región del espacio de características con mayor presencia de instancias de la clase mayoritaria [13]. De la misma manera, existen los casos raros que son aquellas instancias aisladas en una región de la clase mayoritaria [14]. Las instancias en zona frontera son aquellas que están en el límite entre ambas clases. Es decir, estas instancias están rodeadas por una misma cantidad de instancias

de ambas clases. Finalmente, las instancias en zona segura son aquellas que están rodeadas de instancias de la misma clase minoritaria [15].

En la Figura 1.1 se ilustra un ejemplo de una distribución de instancias. Subgrupos de instancias que pertenecen a disyunciones, casos raros, zona frontera y en zonas seguras.

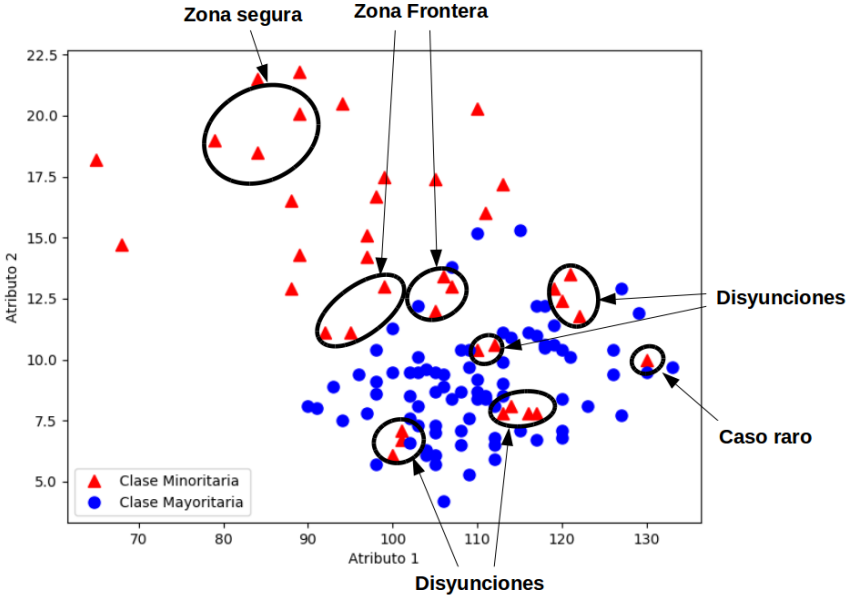


Figura 1.1: Distribución de instancias entre clases

En este trabajo de investigación se pretende mitigar las causas que degradan el desempeño de los modelos de clasificación, en presencia de conjunto con desbalance de clases. Para ello, se propone un análisis de la distribución de las instancias en el espacio de características, y con base

en este análisis, desarrollar técnicas de creación de instancias a nivel de datos.

1.2. Objetivos

El objetivo principal de esta tesis es desarrollar dos técnicas de creación de instancias a nivel de datos, que nos permitan contrarrestar el desbalance de clases, con base en un análisis de la estructura de la clase minoritaria.

Los objetivos específicos son los siguientes:

1.- Analizar la estructura de la clase minoritaria para identificar los subgrupos a los cuales pertenecen las instancias de dicha clase, es decir, disyunciones, casos raros, zona frontera y zonas seguras.

2.-Proponer dos técnicas a nivel de datos, que nos permitan crear instancias artificiales de la clase minoritaria.

3.-Desarrollar un método de ensamble de modelos de clasificación, que utilice las técnicas de creación de instancias propuestas en esta tesis.

4.-Realizar un estudio comparativo del desempeño de las técnicas propuestas con las técnicas para mitigar el desbalance de clases existentes en el estado del arte.

1.3. Hipótesis

Al realizar un análisis de la estructura de la clase minoritaria se puede definir una técnica de creación de instancias de la clase, que nos permita mitigar los problemas relacionados con el desbalance de clases y la distribución de las instancias para aumentar los índices de precisión (*precision*), sensibilidad (*recall*), el valor-F ($F - score$) y exactitud (*accuracy*) de los modelos de clasificación.

1.4. Organización de la tesis

El presente documento está organizado de la siguiente manera. En el Capítulo 2 se describe el problema de desbalance de clases, también se definen los conceptos básicos del área de investigación y se realiza el estudio del estado del arte de las principales técnicas para contrarrestar el problema de desbalance de clases. En el Capítulo 3 se describe la metodología a utilizar para dar solución a nuestro problema, en particular, se presenta el análisis y diseño de los algoritmos. En el Capítulo 4 se describen los resultados experimentales de la propuesta realizada en la presente tesis. Finalmente en el Capítulo 5 se presentan las conclusiones del presente trabajo de investigación, así como, algunas ideas de trabajos a futuro. La principal contribución del presente trabajo de investigación es desarrollar dos técnicas de creación de instancias a nivel de datos, con base en un análisis de la distribución de las instancias en el espacio de características.

Capítulo 2

Marco teórico y trabajos relacionados

2.1. Aprendizaje de máquina

El aprendizaje de máquina es una subárea de la inteligencia artificial que optimiza algún criterio de aprendizaje, usando instancias de ejemplo o experiencia previa [16]. Para optimizar un criterio de aprendizaje, se utiliza algún modelo definido con ciertos parámetros para predecir o describir información.

Las técnicas de aprendizaje de máquina son clasificadas taxonómicamente por Kononenko et al. [17] dependiendo de cómo obtienen o inducen el conocimiento. Estos son: aprendizaje supervisado, no supervisado y aprendizaje por refuerzo.

En este trabajo de investigación nos centraremos en el aprendizaje supervisado, el cual es un tipo de aprendizaje de máquina donde se desea estimar una función desconocida con instancias de entrada, que permiten obtener una variable de salida. El valor de la variable de salida puede ser un valor real (número) para problemas de regresión, o puede ser una valor discreto (clase) para problemas de clasificación [18]. En la presente tesis, nos enfocamos en los problemas de clasificación.

2.2. Desbalance

En la clasificación supervisada frecuentemente aparecen problemas donde la cantidad de instancias de una clase es significativamente mayor que la cantidad de instancias de otra clase. A este tipo de problemas los llamamos problemas con desbalance de clases.

Comúnmente, la clase con menos instancias, representa el concepto más importante que hay que aprender y es difícil identificarlo, ya que podría estar asociado a casos excepcionales pero significativos, o porque la adquisición de estas instancias es muy difícil [19]. El problema de desbalance de clases, como lo mencionamos en el Capítulo anterior, se define en términos de la razón de desbalance. La razón de desbalance expresa el grado de desbalance de un conjunto de datos y se calcula como el cociente de $|N|$ sobre $|P|$, donde $|P|$ es la cantidad de instancias de la clase minoritaria y $|N|$ es la cantidad de instancias de la clase mayoritaria.

A manera de ejemplo, imaginemos un nuevo tipo de cáncer denominado A que, por su rareza y fallas en su diagnóstico, se tienen muy pocas instancias. La poca cantidad de instancias para el nuevo cáncer A sería demasiado pequeña en comparación con la cantidad de instancias para un cáncer similar y muy conocido denominado B. Al usar este conjunto de datos para obtener un modelo de clasificación. El modelo podría tener problemas para clasificar correctamente las instancias del cáncer A, debido a la diferencia en el número de instancias de una clase y otra. En consecuencia, existen algunas técnicas que ayudan a incrementar la cantidad de instancias de la clase minoritaria, cáncer A, y con ese incremento, mitigar el problema de desbalance de clases en el conjunto de datos.

2.3. Distribución de instancias en la clase minoritaria.

En diversos trabajos del estado del arte [20], [21], [22], [23], [24] describen que el pobre desempeño de un modelo de clasificación, no sólo está vinculado al problema de desbalance de clase, sino también a otros factores relacionados con el problema de traslape de clases, así como la formación de pequeños subgrupos de instancias de la clase minoritaria en diferentes regiones del espacio de características. En este sentido, estos trabajos afirman que no importa el tamaño del conjunto de entrenamiento, ni el grado de desbalance de clases, si las clases son linealmente

separables o muestran subgrupos de cada clase bien definidos, con un bajo grado de traslape, no debería existir una degradación significativa en el desempeño de los modelos de clasificación.

A continuación, se describen los subgrupos de la clase minoritaria. Los cuales son objeto de interés en este trabajo de investigación.

Disyunciones

Una disyunción es un pequeño grupo de instancias de una misma clase, es decir, es un grupo de 2 a 5 instancias muy semejantes entre ellas, que cubren pocas instancias de una clase. Las disyunciones pudieran estar en regiones donde predominan instancias de una clase diferente a ellas. En un conjunto de datos, a menudo estas disyunciones dificultan la separabilidad entre clases [13]. Como lo hemos mencionado anteriormente, en conjuntos de datos no balanceados existe una clase minoritaria con mucho menos instancias que la clase mayoritaria y las disyunciones inducidas a partir de esta clase minoritaria tienden a cubrir aún menos instancias. Por lo tanto, se considera una causa importante para una pérdida significativa en el desempeño de los modelos de clasificación. Esta es la razón, por la cual una pobre representación de la clase minoritaria podría ser un obstáculo para la inducción de buenos modelos de clasificación.

Casos raros

Los casos raros son un pequeño número de instancias que se encuentran aisladas unas de las otras, en regiones particulares del espacio de características de una clase diferente a ellas. Podríamos esperar que en los conjuntos de instancias con presencia de desbalance de clases (debido a su naturaleza) la clase minoritaria tenga un mayor número de casos raros que la clase mayoritaria [25].

Zonas frontera

Las zonas frontera son aquellas instancias que están cerca de la frontera o límite entre los espacios de características de las clases, es decir, son instancias que se encuentran muy cercanas entre si, con instancias de su misma clase e instancias de una clase diferente a la de ellas. [26].

Zonas seguras

Las zonas seguras se encuentran ubicadas en regiones homogéneas pobladas por instancias de una sola clase, es decir, las instancias en zonas seguras son aquellas que están en regiones del espacio de características con mayor presencia de instancias de su misma clase [25].

2.4. Técnicas de aprendizaje de máquina tradicional

El aprendizaje de máquina es una área muy activa de la inteligencia artificial que, durante el transcurso de los años, se han desarrollado diversas técnicas para múltiples aplicaciones en el mundo real. En este trabajo de investigación, se utilizaron cinco técnicas de aprendizaje de máquina, redes neuronales artificiales, máquinas de vectores de soporte, k-vecinos más cercanos, árboles de decisión y bosque aleatorio. La Figura 2.1 ilustra las técnicas de aprendizaje de máquina utilizadas.

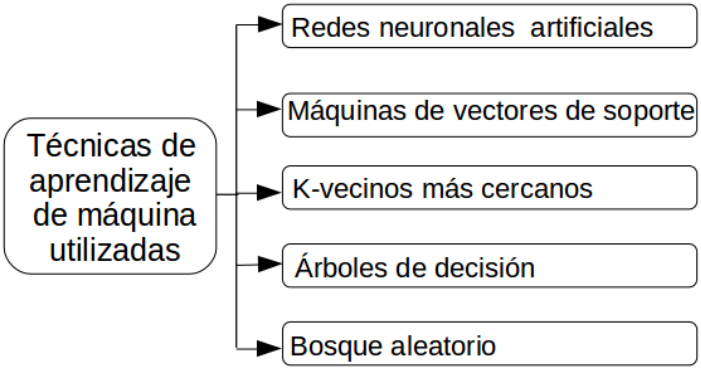


Figura 2.1: Técnicas de aprendizaje de máquina utilizadas

2.4.1. Redes neuronales artificiales

La redes neuronales artificiales imitan el comportamiento de las neuronas humanas. Una red neuronal artificial está compuesta por unidades de procesamiento que están agrupadas en estructuras interconectadas

llamadas capas. El conjunto de una o más capas forman un red neuronal artificial. La literatura especializada distingue tres tipos de capas con base en su funcionalidad: capa de entrada, intermedia y salida. Una capa es un conjunto de nodos (neuronas) cuyos valores de entradas provienen de una capa anterior y cuyos valores de salidas, comúnmente, son la entrada de una capa siguiente. Los nodos de la capa de entrada reciben las instancias a procesar. Los valores de salida de los nodos de la última capa son el resultado visible de la red, por lo que, la última capa se conoce como la capa de salida. Las capas que se sitúan entre la capa de entrada y la capa de salida se conocen como capas intermedias, ya que se desconocen los valores de entrada como los de salida y proporciona cierto grado de libertad a la red para crecer en profundidad [27]. Las conexiones entre nodos tienen asociados pesos que ayudan a ponderan el vínculo entre un par de nodos [28].

Se puede definir una red neuronal artificial como un grafo dirigido, con las siguientes propiedades:

- A cada nodo i se le asocia una variable de estado x_i.
- A cada conexión (i, j) entre los nodos se le asocia un peso $w_{ij} \in \mathbb{R}$.
- A cada nodo i se le asocia un umbral $\theta_i \in \mathbb{R}$. El umbral θ_i representa el grado de inhibición del nodo, es decir, una forma de fijar un nivel mínimo de actividad a la neurona para considerarse como activa.

- Para cada nodo i se define una función de activación f_i que produce un valor de salida de la neurona y su estado de activación. Algunos ejemplos de funciones de activación son función identidad, función escalón, función sigmoidea, entre otras.

En la Figura 2.2 se ilustra un ejemplo de una red neuronal que tiene una estructura de tres capas. La capa de entrada (nodos x_1 y x_2), capa intermedia (nodos x_3, x_4 y x_5) y capa de salida (nodos x_6 y x_7). Para relacionar los nodos de una capa con la capa siguiente, se definen pesos entre los nodos. Por ejemplo, $w_{x_1x_3}$ indica el peso de la conexión entre el nodo x_1 en la capa de entrada y el nodo x_3 en la capa intermedia.

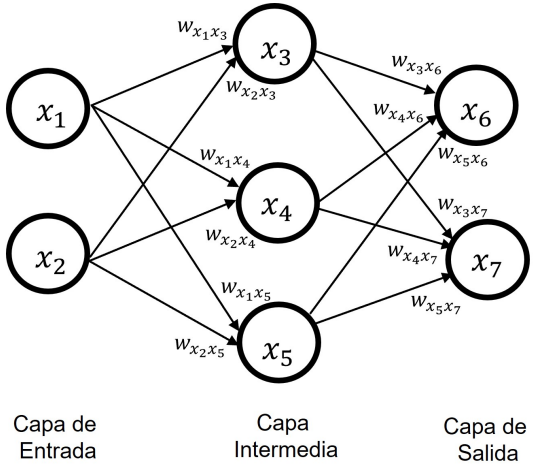


Figura 2.2: Estructura de una Red Neuronal Artificial.

2.4.2. Máquinas de vectores de soporte

Las máquinas de vectores de soporte (MVS) han sido, en los últimos años, una de las técnicas de aprendizaje de máquina más utilizadas para problemas de clasificación binaria, pero también se han extendido a problemas de clasificación múltiple y de regresión. La idea principal de las MVS es encontrar entre los diversos hiperplanos que separan las clases, un hiperplano que maximiza el margen entre los vectores de soporte. Los vectores de soporte son un subconjunto de datos, que son las instancias (puntos) más cercanas de las clases que definen la posición del hiperplano [29].

Supongamos que tenemos un problema de clasificación binaria linealmente separable. Partimos de un conjunto de vectores $\{(x_i, y_i)$,....,$(x_m, y_m)\}$ donde $x_i \in \mathbb{R}^d$ e $y_i \in \{+1, -1\}$ para $i = 1, .., m$. Existe un hiperplano $H : w \cdot x_i + b = 0$ que separa los vectores x_i. Donde $w \in \mathbb{R}^d$ es un vector normal que define la frontera y b es el sesgo. Además, H es el hiperplano medio entre los dos hiperplanos H_1 y H_2 que contiene los vectores de soporte (instancias de entrenamiento más cercanas al hiperplano H) y son paralelas a H, como se observa en las Ecuaciones 2.1 y 2.2.

$$H_1 : w \cdot x_i + b = 1 \text{ Para } x_i \text{ con valor de salida } y_i = +1 \tag{2.1}$$

$$H_2 : w \cdot x_i + b = -1 \text{ Para } x_i \text{ con valor de salida } y_i = -1 \tag{2.2}$$

La condición de clasificación es la Ecuación 2.3.

$$y_i(w \cdot x_i + b) - 1 \geq 0, i = 1, ..., n \tag{2.3}$$

Construida a partir de las zonas más allá de los margenes, como se muestra en las Ecuaciones 2.4 y 2.5.

$$w \cdot x_i + b \geq +1 \text{ para } y_i = +1 \tag{2.4}$$

$$w \cdot x_i + b \leq -1 \text{ para } y_i = -1 \tag{2.5}$$

Además, la distancia de H_1 al origen es $\frac{|1-b|}{||w||}$ donde $||w||$ es la norma euclidea de w y la distancia de H_2 es $\frac{|-1-b|}{||w||}$. Así se tiene que los hiperplanos H_1 y H_2 son paralelos y el margen entre ellos es $\frac{2}{||w||}$, el cual se requiere maximizar.

Por lo tanto, se puede encontrar el hiperplano óptimo H resolviendo el problema de optimización cuadrático, en la Ecuación 2.6.

$$min\frac{1}{2}||w||^2 \tag{2.6}$$

Sujeto a: $y_i(w \cdot x_i + b) - 1 \geq 0$

En la Figura 2.3 se muestra la representación geométrica del problema de programación cuadrática mostrando H (separador óptimo) y los

hiperplanos H_1 y H_2.

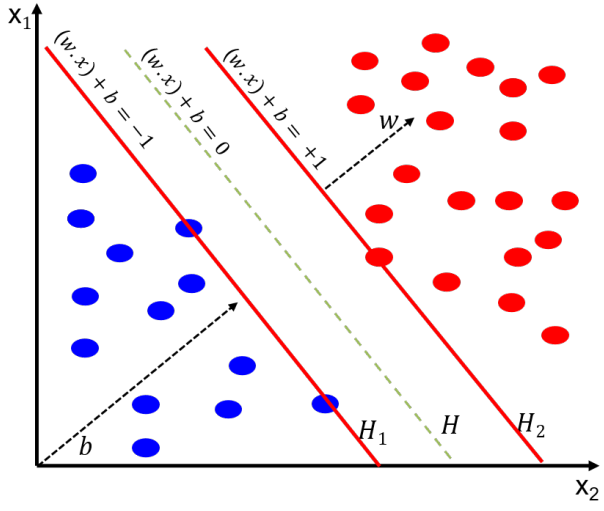


Figura 2.3: Hiperplano H óptimo, en un problema linealmente separable.

Cuando el problema de clasificación no es linealmente separable. La técnica de MVS utiliza una función de kernel para mapear las instancias a un espacio de características de mayor dimensión, de modo que las instancias sean linealmente separables. Algunas de las funciones de kernel más utilizadas son el polinomio, la base radial y la función sigmoidal.

2.4.3. $K-$vecinos más cercanos

La técnica de $k-$vecinos más cercanos ($k-NN$ por sus siglas en inglés) es una de las técnicas de clasificación supervisada más básicas y sencillas. La idea es calcular las distancias de una instancia nueva con las instancias existentes y ordenar dichas distancias de menor a mayor

para determinar la clase a la que pertenece. Esta clase será la de mayor frecuencia con menores distancias [30]. Se dice que las instancias que están cerca unas de otras son vecinos. Se puede especificar el número de vecinos más cercanos a examinar, este valor es determinado por k. En caso de que se produzca un empate entre dos o más clases, conviene tener una regla heurística para su ruptura, por ejemplo, seleccionar la clase que contiene al vecino más próximo, seleccionar la clase con distancia media menor, entre otras.

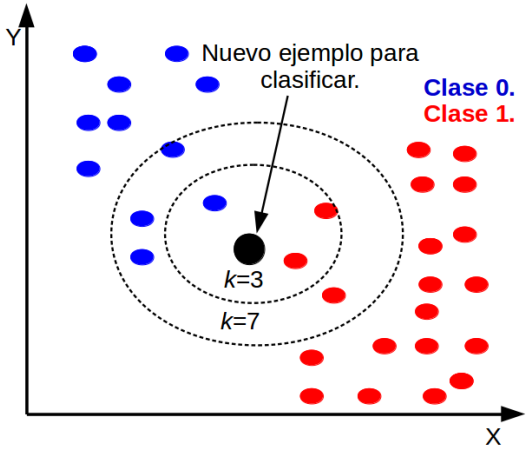


Figura 2.4: Ejemplo de $k-$vecinos más cercanos.

En la Figura 2.4 se observa un ejemplo de clasificación. Cuando k=3 el nuevo ejemplo es clasificado en la clase 1, pero cuando k=7, el nuevo ejemplo es clasificado en la clase 0.

2.4.4. Árboles de decisión

Los árboles de decisión generan decisiones secuenciales que se organizan de forma jerárquica, por este motivo reciben el nombre de árboles. Los elementos básicos de los árboles de decisión son los nodos de decisión y los nodos hoja. Los nodos de decisión, representan las características de las instancias del conjunto de datos. De los nodos de decisión se generan tantas ramas como valores posibles tengan las características, cada rama se conecta a otro nodo de decisión o a un nodo hoja. Los nodos hoja representan las clases a las que pertenecen las instancias. En esta técnica, se emplea la estrategia divide y vencerás, para partir el problema en subproblemas [31].

Los árboles de decisión se pueden representar como conjuntos de reglas $IF - THEN$. Algunos de los algoritmos más conocidos son $ID3$ y $C4.5$ [32].

Se puede construir un árbol de decisión, que determine si una persona debe o no debe jugar al golf [33]. El tiempo, la temperatura, la humedad y el viento son las características utilizadas para la creación del árbol de decisión. La Figura 2.5 ilustra un ejemplo de un árbol de decisión para determinar si se juega al golf.

Una vez que se tiene generado el árbol, es posible generar fácilmente un conjunto de reglas del tipo: si $cond_1 \wedge \text{cond}_2 \wedge ....\wedge \text{cond}_n$, entonces tendríamos una predicción donde cada rama puede ser considerada como

una regla. Los antecedentes de la regla son definidos por cada nodo interno, desde la raíz hasta las hojas, y el nodo hoja correspondiente define el consecuente de la regla. Así, por ejemplo, las reglas de la Tabla 2.1 son obtenidas a partir del árbol de la Figura 2.5.

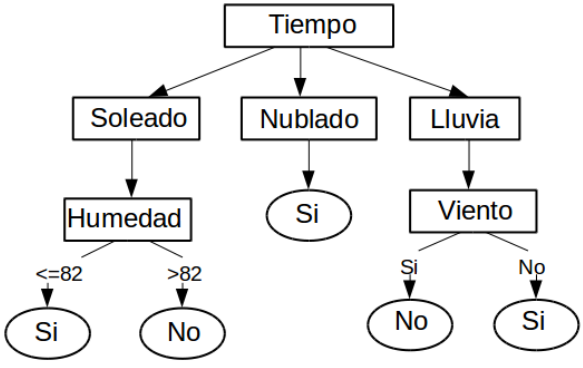


Figura 2.5: Árbol de decisión para determinar si se juega al golf.

Tabla 2.1: Ejemplo de reglas para determinar si se juega al golf

a) Si tiempo = soleado $\wedge$ humedad = es <= 82 entonces jugar = si.
b) Si tiempo = soleado $\wedge$ humedad = es > 82 entonces jugar = no.
c) Si tiempo = nublado entonces jugar = si.
d) Si tiempo = lluvia $\wedge$ viento = si entonces jugar = no.
e) Si tiempo = lluvia $\wedge$ viento = no entonces jugar = si.

2.4.5. Bosque aleatorio

Bosque aleatorio es una técnica que consiste en combinar un conjunto de árboles de decisión entrenados de forma independiente [34]. Donde cada árbol de decisión contiene distintas evaluaciones en sus nodos. Para

un problema de clasificación, el objetivo es obtener mejores predicciones. El valor de predicción se obtiene al seleccionar la clase que logra la mayoría de votos entre todos los árboles del bosque.

La técnica de bosque aleatorio genera una serie de árboles de decisión. Cada modelo de árbol de decisión es entrenado utilizando un conjunto de instancias de entrenamiento. Posteriormente, las predicciones de estos modelos son combinadas mediante voto mayoritario. En la Figura 2.6 podemos ver un ejemplo del funcionamiento de bosque aleatorio.

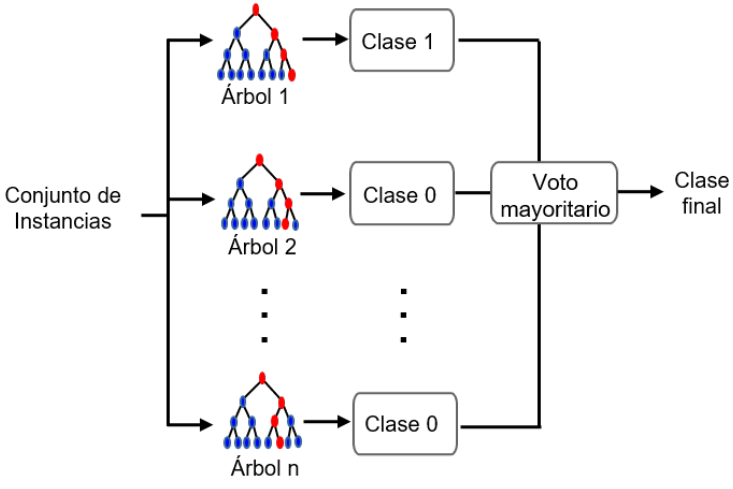


Figura 2.6: Ejemplo del funcionamiento de bosque aleatorio.

2.5. Métodos de ensamble de modelos

Los métodos de ensamble de modelos combinan varías técnicas de aprendizaje de máquina para producir un modelo de clasificación con alto índice de precisión al clasificar. Los métodos de ensamble de modelos se dividen en *bagging* y *boosting* [35]. En la presente tesis, utilizamos *bagging* conjuntamente con las técnicas de aprendizaje de máquina: redes neuronales artificiales, máquinas de vectores de soporte, $k-$vecinos más cercanos, árboles de decisión y bosque aleatorio.

Bagging

Bagging es un método de ensamble más utilizado para reducir la varianza, que se basa en la utilización de una estrategia de selección con reemplazo basada en *bootstrap* [36]. Una muestra *bootstrap* es generada por muestreo de m instancias del conjunto de entrenamiento. Cada instancia del conjunto de entrenamiento tiene una probabilidad de $\frac{1}{m}$ de ser seleccionada. Se generan diferentes muestras *bootstrap* a partir del conjunto de entrenamiento, y después se entrenan diferentes modelos de clasificación a partir de cada uno de estas muestras. En el proceso de deducción, se utiliza el criterio de voto mayoritario entre los modelos para decidir las predicciones finales. Un esquema general de la técnica de *bagging* se muestra en la Figura 2.7.

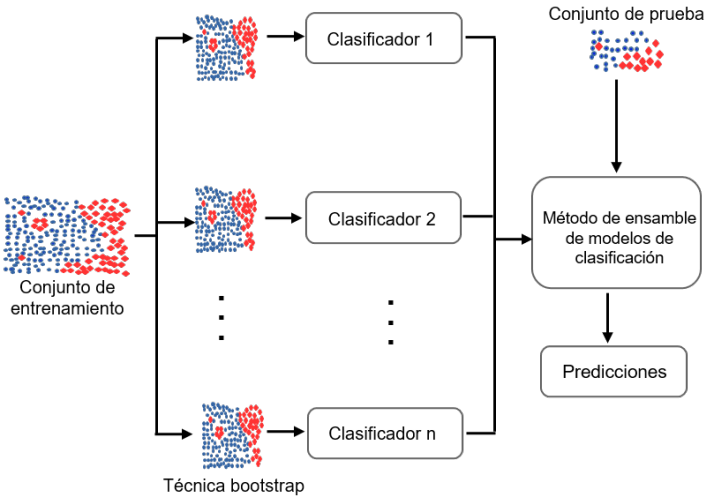


Figura 2.7: Esquema general del método *bagging*.

2.6. Métricas de desempeño

El criterio de evaluación es un factor clave a la hora de medir el desempeño de los modelos de clasificación. En un problema de clasificación binaria, la matriz de confusión permite conocer el desempeño de los modelos de clasificación, es decir, esta matriz nos muestra la relación de las predicciones realizadas por un modelo y los resultados reales (correctos). Por lo tanto, la matriz de confusión registra los resultados de las instancias clasificadas correctamente e incorrectamente en cada clase [37].

En concreto, podemos obtener la fracción de instancias bien clasificadas en la clase positiva (TP), la fracción de instancias bien clasificadas

en la clase negativa (TN), la fracción de instancias mal clasificadas en la clase negativa (FP) y la fracción de instancias mal clasificadas en la clase positiva (FN). A continuación describimos las métricas de desempeño utilizadas en este trabajo de investigación. En la Tabla 2.2, se observa la matriz de confusión.

Tabla 2.2: Matriz de confusión para problema de clases binaria

	Positiva Real	Negativa Real
Positiva Predicha	Verdaderos Positivos (TP)	Falsos Positivos (FP)
Negativa Predicha	Falsos Negativos (FN)	Verdaderos Negativos (TN)

Precisión $=\frac{TP}{TP+FP}$ es la proporción de verdaderos positivos entre todos los resultados positivos (verdaderos positivos y falsos positivos).

Sensibilidad $=\frac{TP}{TP+FN}$ es la proporción de casos positivos que fueron correctamente clasificados.

Valor-F $=\frac{2*Precisión*Sensibilidad}{Precisión+Sensibilidad}$ combina la precisión con la sensibilidad.

Exactitud es la proporción de predicciones correctas que el modelo realizó [38]. La exactitud ha sido la medida comúnmente más utilizada para evaluar la eficacia de un modelo predictivo ver Ecuación 2.7. Sin embargo, por ser una medida global, no considera los resultados por clase.

$$Exactitud = \frac{TP + TN}{TP + FN + FP + TN} \tag{2.7}$$

2.7. Trabajos Relacionados

En esta sección realizamos una revisión del estado del arte de los trabajos que investigan el problema de desbalance de clases. En general, las principales estrategias para resolver el problema de desbalance de clases, se distinguen en tres enfoques [19].

- Técnicas a nivel de datos, modifican la recopilación de instancias para balancear las clases, al agregar o eliminar instancias.
- Técnicas a nivel de algoritmo, modifican directamente los algoritmos de aprendizaje para mitigar el sesgo hacia la clase mayoritaria.
- Técnicas híbridas, combinan las técnicas a nivel de datos y las técnicas a nivel de algoritmos.

Técnicas a nivel de datos

El enfoque a nivel de datos emplea el preprocesamiento para balancear la distribución de clases. Esto se hace modificando el conjunto de instancias de entrenamiento para los modelos de clasificación. En este enfoque, se generan nuevas instancias para la clase minoritaria (sobremuestreo) o se eliminan instancias de la clase mayoritaria (submuestreo)

para reducir la razón de desbalance entre las clases. Al utilizar técnicas aleatorias para la creación o eliminación de instancias, se conduce a la perdida de información relevante de las clases mayoritarias o a duplicar información de las clases minoritarias, modificando la estructura de las clases.

Técnicas a nivel de algoritmos

Las técnicas a nivel de algoritmo emplean algoritmos dedicados que aprenden directamente la distribución de desbalance de clases en los conjuntos de instancias. Los algoritmos están basados en el reconocimiento de clasificación de aprendizaje por clases, aprendizaje sensible al costo y técnicas de conjunto.

Técnicas híbridas

Para mejorar el desempeño de los modelos de clasificación en presencia de desbalance de clases, a menudo se emplean las técnicas híbridas, es decir, combinar las técnicas a nivel de datos y a nivel de algoritmos. Este tipo de técnicas ayudan a mitigar los problemas en el muestreo, selección de subconjuntos de características, optimización de la matriz de costos y ajuste del aprendizaje de algoritmos.

Trabajos a nivel de datos

En 2002, Chawla et al. [39] desarrollaron una técnica de sobremuestreo SMOTE (*Synthetic Minority Oversampling Technique*). SMOTE

genera instancias sintéticas en diferentes regiones del espacio de características de la clase minoritaria. SMOTE primero calcula el número de instancias a generar, después selecciona aleatoriamente instancias de la clase minoritaria. Para cada instancia seleccionada se elige aleatoriamente uno de sus k-vecinos más cercanos y se calcula la diferencia entre las características de la instancia seleccionada y la instancia vecina más cercana seleccionada. Se prosigue con multiplicar la diferencia por un número aleatorio entre 0 y 1. Finalmente se crea una instancia sintética. Para los resultados experimentales, se utilizó el conjunto de datos *Oil*, que cuenta con 896 instancias de la clase mayoritaria y 41 instancias de la clase minoritaria, en total cuenta con 937 instancias. La técnica propuesta, SMOTE, obtuvo una precisión de 0.86, mientras que la técnica comparativa de submuestreo aleatorio obtuvo una precisión de 0.76.

En 2018, Georgios et al. [40] utilizaron los algoritmos de $k-$medias y SMOTE para desarrollar una técnica de sobremuestreo. En esta técnica se agrupan instancias mediante el algoritmo $k-$medias, después se calcula la razón de desbalance en cada agrupación y finalmente se utiliza el algoritmo SMOTE para realizar un sobremuestreo en cada agrupación. El objetivo de la técnica es eliminar el desbalance de clases y a la vez combatir el problema de las disyunciones. Al mitigar la escasez de instancias en zonas minoritarias y evitar la generación de casos raros eliminan el problema de las disyunciones. Para medir el desempeño de su técnica, los autores utilizaron la técnica de $k-$vecinos más cercanos para generar un modelo de clasificación. Las técnicas con las cuáles se

compararon fueron SMOTE, Borderline-SMOTE y sobremuestreo aleatorio. Los autores utilizaron el conjunto de datos *wine* el cual tiene 178 instancias, donde cada instancia cuenta con 13 características. La técnica propuesta obtuvo la mejor precisión, 0.94. En el mismo año, Zhou et al. [4] desarrollaron una técnica mejorada de esteganografía segura basada en redes generativas antagónicas (SSGAN). La técnica propuesta utiliza un modelo de redes generativas antagónicas (GAN) para generar instancias sintéticas en el conjunto de entrenamiento. Los experimentos comparativos se realizaron entre el algoritmos SSGAN y bosque aleatorio, utilizando la métrica de precisión. La precisión del modelo propuesto fue de 0.92, el algoritmo bosque aleatorio obtuvo una precisión inferior del 0.86. Se utilizo el conjunto de datos $AlibabaMIFD$, el cual tiene 12000 instancias, donde cada instancia cuenta con 721 características. Hazanin et al. [41] estudiaron la perdida de información que se produce al aplicar submuestreo a grandes conjuntos de datos. Los conjuntos de datos contaban con una razón de desbalance de clases en diferentes porcentajes: 10 %, 1 %, 0.1 %, 0.01 % y 0.001 %. Los autores utilizaron el algoritmo bosque aleatorio para entrenar un modelo de clasificación. Los resultados mostraron que, para conjuntos con desbalance de clases del 0.1 % al 1.0 % el desempeño del modelo fue de 0.81 a 0.94 en precisión. Sharma et al. [42] propusieron una técnica para mitigar el problema de desbalance de clases, en conjuntos de datos con desbalance de clases alto. La técnica se llama sobremuestreo con base en la clase mayoritaria (SWIM). SWIM utiliza la información de la estructura de

la clase mayoritaria para generar instancias sintéticas en la clase minoritaria. Primero SWIM utiliza la medida de distancia Mahalanobis. Esta distancia determina la similitud entre dos instancias, tomando en cuenta la correlación entre las características, después SWIM identifica las instancias que se encuentren en zonas límite entre clases y genera instancias sintéticas de la clase minoritaria en zonas límite. Para esta investigación, se utilizaron 26 conjuntos de datos, para el caso del conjunto de datos *Abalone*, el cual cuenta con 4177 instancias, donde cada instancia tiene ocho características. El algoritmo SWIM obtuvo resultados de 0.72 de precisión, mientras que la técnica comparativa de sobremuestreo aleatorio obtuvo 0.61 de precisión.

En 2020, Sun et al. [43] desarrollaron un algoritmo de agrupación, denominado estimación de subregiones minoritarias basadas en la distribución de direcciones (DDMSE). Lo primero que realiza el algoritmo es una selección de instancias de referencia de la clase mayoritaria, después forma subregiones para agrupar las instancias de la clase minoritaria. Para ello, DDMSE realiza un análisis de la distribución de las instancias en la clase minoritaria y considera el factor de dirección como criterio para agrupar las instancias en regiones seguras y regiones inseguras. Finalmente utilizan la técnica MWMOTE (*Majority Weighted Minority Oversampling Technique*). Para identificar las instancias atípicas de la clase minoritaria. A cada instancia atípica le asigna un valor con base en la distancia euclidiana con su instancia más cercana, después MWMOTE utiliza el algoritmo SMOTE para la creación de las nuevas

instancias sintéticas a partir de ponderar el valor asignado a cada instancia. DDMSE utiliza el algoritmo de redes neuronales para entrenar un modelo de clasificación. Sus resultados experimentales demostraron que, su propuesta puede determinar las regiones de la estructura de la clase minoritaria más adecuadas para realizar un sobremuestreo. En la etapa experimental, se utilizó el conjunto de datos $Webpage$, el cual tiene 4177 instancias, donde cada instancia cuenta con 5 características. DDMSE obtuvo la mejor precisión del 0.89.

En 2021, Puri et al. [44] propusieron una técnica que combina el submuestreo con el sobremuestreo en la clase minoritaria. Para realizar el submuestreo utilizaron la técnica de ENN ($Editing\ Nearest\ Neighbor$). La técnica ENN elimina instancias que se encuentran aisladas de instancias de su misma clase. Para realizar el sobremuestreo aplicaron el algoritmo de $k-$medias junto con SMOTE. Finalmente, los autores utilizaron $AdaBoost$ y el algoritmo de árboles de decisión para generar un modelo de clasificación. Los resultados experimentales mostraron una precisión de 0.91 utilizando la técnica propuesta y una precisión de 0.83 utilizando únicamente la técnica de SMOTE. Se utilizó el conjunto de datos $Glass$, el cual tiene 214 instancias y cada instancia cuenta con 10 características.

Trabajo a nivel de algoritmos

En 2016, Elkarami et al. [45] desarrollaron un clasificador sensible al costo basado en la técnica de bosque aleatorio que mitiga el problema de

desbalance de clases. Primero el modelo crea N subconjuntos de instancias utilizando la técnica de selección *boostrap*. Para cada subconjunto creado se utiliza un clasificador sensible al costo, el cual agrega un factor a la clase con más riesgo de ser clasificada erróneamente. Los resultados experimentales mostraron que, el modelo propuesto obtuvo 0.92 en la métrica de sensibilidad.

En 2017, Zhang et al. [6] propusieron un codificador automático basado en redes neuronales para conjuntos de datos con desbalance de clases (SDAE). El algoritmo propuesto agrega ruido aleatorio a las instancias originales, para permitir que la red neuronal aprenda las instancias con ruido adicional y mejore la capacidad de aprendizaje. En la etapa experimental, se realizaron comparativas entre la técnica propuesta y la técnica que utiliza codificadores automáticos dispersos (SAE). Los resultados mostraron un 80 % de exactitud para la técnica SAE y 83 % de exactitud para la técnica propuesta.

En 2019, Zhang et al. [46] implementaron un método de ensamble evolutivo basado en submuestreo. El método utiliza el algoritmo de optimización de enjambre de partículas. Para realizar un submuestreo en la clase mayoritaria, los autores utilizaron un algoritmo evolutivo para encontrar un subconjunto de instancias de la clase mayoritaria, que fuera de la misma cardinalidad que el conjunto de instancias de la clase minoritaria. Los resultados experimentales demostraron que la técnica propuesta obtuvo 0.86 de sensibilidad, mientras que la técnica compara-

tiva ROS-Bag, que es una técnica de sobremuestreo aleatorio basado en el método de Baggin, obtuvo 0.75 de sensibilidad para el conjunto de datos $Ecoli5$. El conjunto de datos $Ecoli5$ contiene 336 instancias, donde cada instancia cuenta con 7 características, el conjunto de datos contiene una razón de desbalance de clases de 8.6. Para el conjunto de datos $Abalone18vs9$ que tiene 731 instancias, donde cada instancia cuenta con 8 características y el conjunto de datos cuenta con una razón de desbalance de clases de 16.4. La técnica propuesta obtuvo 0.76 de sensibilidad y la técnica comparativa ROS-Bag obtuvo 0.64 de sensibilidad.

Trabajos híbridos

En 2019 Rustogi et al. [8] propusieron una técnica híbrida para clasificar instancias con desbalance de clases. La técnica utiliza SMOTE para generar instancias sintéticas en la clase minoritaria. Para medir el desempeño de su técnica, los autores utilizaron máquinas de aprendizaje extremo (ELM) para generar un modelo de clasificación. ELM son redes de neuronas artificiales con una sola capa oculta, conectadas con pesos aleatorios a la capa de entrada y en la que solo se entrenan los pesos que la conectan con la capa de salida mediante un algoritmo de mínimos cuadrados. En los resultados experimentales, se utilizó el conjunto de datos *Pima Indians*, el cual tiene 768 instancias, con 8 características cada instancia y con una razón de desbalance de clases de 1.87. El modelo propuesto obtuvo un valor-F de 0.68 y el modelo comparativo que utiliza SMOTE obtuvo 0.63 de valor-F.

En 2020, Junhai Zhai et al. [47] desarrollaron una técnica de sobremuestreo modificada basada en el modelo de redes discriminatorias generativas de doble discriminador (D2GAN), con un enfoque de clasificación de instancias basado en la fusión de clasificador e integral difusa. D2GAN es un modelo que tiene tres redes neuronales, una red neuronal es llamada generador y las otras dos redes neuronales son llamadas discriminadores. El generador crea muestras de datos y los dos discriminadores catalogan si las muestras creadas por el generador son reales o no. La modificación al modelo D2GAN, se caracteriza en agregar un clasificador al modelo para aprender la diferencia entre muestras positivas y muestras negativas, a fin de mitigar la superposición entre instancias. Los resultados mostraron que, generar instancias sintéticas con buena diversidad y separabilidad controlable mejora el aprendizaje del clasificador. Para el conjunto de datos $Vowel$, que tiene 988 instancias, con 13 atributos cada instancia y el conjunto de datos tiene una razón de desbalance de clases de 9.9. La técnica propuesta obtuvo una precisión de 0.93, mientras que la técnica comparativa SMOTE obtuvo una precisión de 0.84.

Capítulo 3

Metodología

En este Capítulo se presenta la metodología utilizada en este trabajo de investigación. Además, se describe el análisis y el diseño de los algoritmos implementados para la creación de instancias sintéticas. Así mismo, se especifican las técnicas de aprendizaje de máquina utilizadas para crear el modelo de clasificación.

3.1. Estrategia de creación de instancias sintéticas

La solución propuesta es una técnica que integra el análisis de la distribución de las instancias en el espacio de características de la clase minoritaria, con técnicas de sobremuestreo para generar instancias sintéticas. La solución propuesta en el presente trabajo de tesis parte de tres ideas básicas:

1. Analizar el espacio de características para agrupar las instancias de la clase minoritaria en cuatro tipos de subgrupos: disyunciones, casos raros, zona frontera y zonas seguras.

2. Determinar la cantidad de instancias sintéticas a crear para cada subgrupo formado.

3. Generar instancias sintéticas con base en la información de los subgrupos. En este punto se proponen dos técnica de creación de instancias sintéticas: Generación de instancias artificiales con base en centroides (GIAC) y Generación de instancias artificiales con base en SMOTE (GIAC-S).

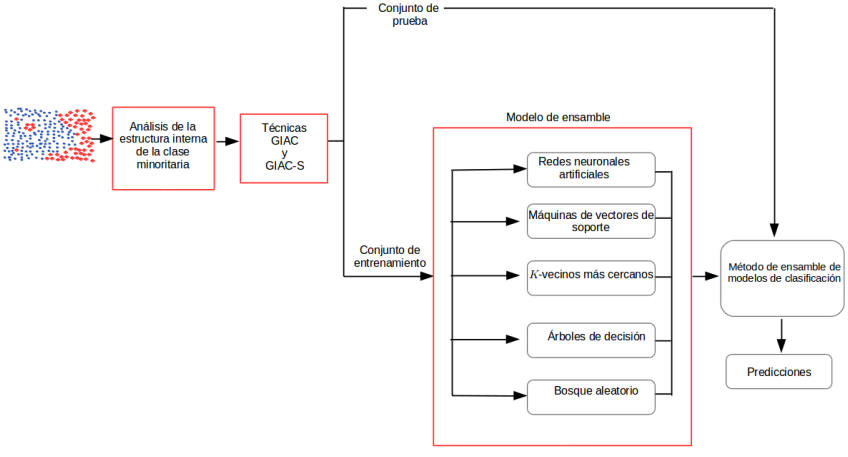


Figura 3.1: Esquema general de la metodología propuesta

En la Figura 3.1 se ilustra el esquema general de nuestra metodología. Para demostrar el funcionamiento correcto de nuestra propuesta

de creación de instancias generamos un modelo de clasificación basado en un método de ensamble de modelos. Nuestro modelo utiliza cinco algoritmos de aprendizaje de máquina, redes neuronales artificiales, máquinas de vectores de soporte, $k-$vecinos más cercanos, árboles de decisión y bosque aleatorio. Además, utilizamos validación cruzada y reportamos los índices de precisión, sensibilidad, valor-F y exactitud del modelo de clasificación.

3.2. Pre-análisis de los datos

El primer paso de esta metodología es realizar un pre-análisis de los conjuntos de datos para conocer la distribución de las instancias en el espacios de características, debido a que existen diferentes escenarios en los que se puede encontrar un conjunto de datos. En la Figura 3.2 se observan los procesos para realizar un pre-análisis de los datos en un conjunto de datos.

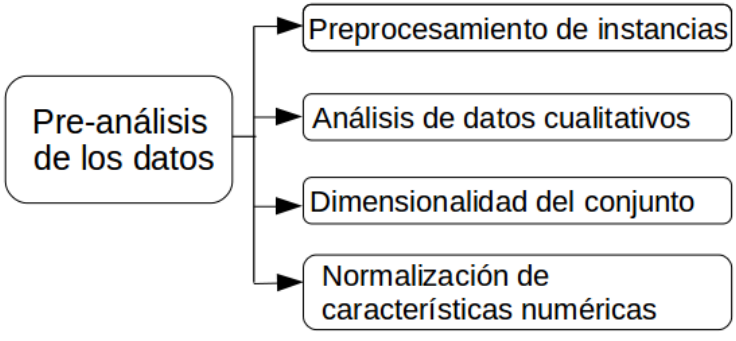


Figura 3.2: Pre-análisis de los datos

Escenario 1: cuando las instancias se encuentran duplicadas o con valores ausentes (faltantes), se realiza un preprocesamiento. El preprocesamiento consiste en eliminar instancias que se encuentran duplicadas o con características con valores ausentes. Este preprocesamiento se realiza justificando que, si esas instancias no se eliminan, afectarían directamente al proceso de identificación de subgrupos, debido a que utilizamos una técnica basada en distancia ($k-$vecinos más cercanos).

Escenario 2: en caso de que las características sean cualitativas, es decir, no numéricas que adquieran valores de un número limitado de clases o categorías [48]. Las características cualitativas pueden ser de dos tipos:

- Ordinales: sus valores pueden ser ordenados jerárquicamente, por ejemplo, situación socio-económica (ingresos bajos, ingresos medios, ingresos altos). Para este tipo de características se utilizó una codificación ordinal, que consiste en remplazar cada valor de la característica con un número entero distinto.

- Nominales: sus valores representan categorías que no obedecen a una clasificación intrínseca. Además, no se puede establecer un orden entre sus categorías. Algunos ejemplos son el sexo de una persona (hombre o mujer) o el estado civil de una persona (soltero, casado, viudo, divorciado, unión libre). En términos generales, consiste en crear una nueva característica binaria por cada categoría existente de la característica a codificar. Así, estas nuevas

características contendrán 1 en aquellas instancias que pertenezcan a esa categoría y 0 en el resto.

Escenario 3: en conjuntos de datos con un gran número de características (alta dimensionalidad), existe la posibilidad de tener el problema de la maldición de la dimensionalidad (*Curse of Dimensionality*) [49]. Por lo tanto, se requiere realizar un análisis, ver Apéndice A, para descartar el problema de la maldición de la dimensionalidad o para determinar si es necesario aplicar una técnica para mitigar este problema, por ejemplo, análisis de componentes principales (PCA).

Escenario 4: la normalización de las características numéricas es importante para evitar la sobre ponderación de una característica sobre otra, en este trabajo se utilizo la normalización $Z - score$ [50].

3.3. Análisis de la estructura de la clase minoritaria

Para realizar el análisis de la estructura de la clase minoritaria se identifican los vecinos más cercanos a cada instancia de la clase minoritaria. Estos vecinos más cercanos se determinan a partir de la comparación entre los valores de sus características para establecer un grado de cercanía entre ellos. Para ello, utilizamos la técnica de k-vecinos más cercanos.

En la Figura 3.3 se observan los pasos a realizar en el análisis de

la estructura de la clase minoritaria. En esta metodología se utilizó la distancia *Manhattan*, la cual determina la distancia de dos vectores numéricos a través de la suma de las diferencias absolutas de sus elementos.

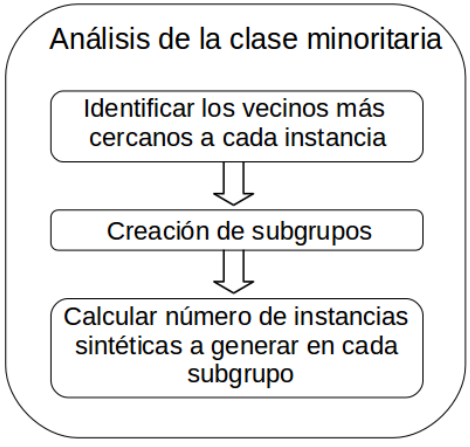


Figura 3.3: Análisis de la estructura de la clase minoritaria

En diversos artículos del estado del arte [51, 52, 53], se menciona que la distancia *Manhattan* es ampliamente utilizada para conjuntos de datos con alta dimensionalidad, además es menos susceptible a instancias atípicas (*outliers*) [54]. La formula matemática de la distancia Manhattan se presenta en la Ecuación 3.1.

$$d = \sum_{i=1}^{n} |x_i - y_i| \tag{3.1}$$

donde x e y son instancias, i es la i-ésima característica de cada instancia y n es el número de características.

Para determinar los diferentes tipos de subgrupos es importante decidir el valor de k. Si el valor de k es grande se corre el riesgo de hacer la clasificación de acuerdo a las instancias de la clase mayoritaria. Si el valor de k es pequeño puede haber imprecisión en la clasificación a causa de los pocas instancias seleccionadas. Por lo tanto es recomendable probar con distintos valores de k.

A continuación, se describe las especificaciones que se tienen que cumplir para cada tipo de subgrupo. Se definen tres umbrales α, β, γ. Estos umbrales fueron obtenidos de manera experimental. El proceso de experimentación que se realizó para determinar los umbrales adecuados para todos los conjuntos de datos se desarrollo de la siguiente manera. Se comenzó en un rango de 0 a 1, incrementando los valores en 0.1. Con base en los resultados que mejor se obtuvieron se delimitaron los umbrales para los cuatro tipos de subgrupos. Para que se cumpla que una instancia pertenece a un tipo de subgrupo, las especificaciones de sus vecinos se definen de la siguiente manera:

$$\alpha = \lfloor 0.2 \times k \rfloor \qquad \beta = \lfloor 0.4 \times k \rfloor \qquad \gamma = \lfloor 0.6 \times k \rfloor$$

- Subgrupo disyunciones: donde el número de instancias de la clase minoritaria es mayor que α y menor o igual que β.

- Subgrupo casos raros: donde el número de instancias de la clase minoritaria es menor o igual que α.
- Subgrupo frontera: donde el número de instancias de la clase minoritaria es mayor que β y menor o igual que γ.
- Subgrupo seguro: donde el número de instancias de la clase minoritaria es mayor que γ.

Una vez que se tienen las especificaciones de los subgrupos, cada instancia de la clase minoritaria se agrupa con sus $k-$vecinos más cercanos que pertenezcan a la clase minoritaria, y se agregan al subgrupo que correspondan. Una vez formados los diferentes subgrupos, se determina el número de instancias sintéticas a generar en la clase minoritaria. Para ello definimos (G) como la cantidad total de instancias sintéticas a generar, que se determina con base en la razón de desbalance (R_D), como se muestra en la Ecuación 3.2, donde $|P|$, determina la cantidad de instancias de la clase minoritaria.

$$G = (R_D * |P|) - |P| \tag{3.2}$$

Para determinar el número de instancias sintéticas ha generar en cada subgrupo (T), es necesario que la generación de instancias sintéticas se haga de manera equitativa con base en la proporción de instancias que contenga cada subgrupo. Por lo tanto, se propone la expresión descrita en la Ecuación 3.3, donde $|s|$ hace referencia a la cardinalidad del

subgrupo que se esté analizando.

$$T = (G/|P|) * |s| \tag{3.3}$$

3.4. GIAC

La técnica de creación de instancias sintéticas GIAC es una técnica de sobremuestreo. Esta técnica utiliza el análisis de la distribución de las instancias en el espacio de características de la clase minoritaria. GIAC genera instancias sintéticas basadas en representantes del grupo. Esta técnica utiliza las instancias de cada tipo de subgrupo para generar instancias sintéticas. A continuación se presenta el Algoritmo 1 GIAC.

GIAC recibe como valores de entrada: el subconjunto de instancias de la clase minoritaria, el subconjunto de instancias de la clase mayoritaria y el valor de la cantidad de vecinos a considerar. La salida del algoritmo es un conjunto de datos sin desbalance de clases. En el paso uno, se calcula el número de instancias a generar en la clase minoritaria. En el paso dos se crean los tipos de subgrupos. La creación de los subgrupos es mediante el análisis de la estructura de la clase minoritaria de acuerdo a la Sección 3.3. Para el paso tres se obtiene el resultado de la cantidad de instancias a generar en cada tipo de subgrupo creado. En el paso cuatro cada instancia sintética es generada con base en la media de las características de las $n - 1$ instancias de cada agrupación que conforman un tipo de subgrupo. Finalmente se obtiene un conjun-

to de datos sin desbalance de clases, que contiene las instancias de la clase mayoritaria, las instancias de la clase minoritaria y las instancias sintéticas.

Algoritmo 1 GIAC

Entrada:	N subconjunto de instancias del la clase mayoritaria. P subconjunto de instancias de la clase minoritaria. k cantidad de vecinos más cercanos
Salida:	$D = \{P \cup N \cup N'\}$ conjunto de datos sin desbalance de clases.
Paso 1:	Calcular instancias a generar G=(R_D*$\|P\|$) - $\|P\|$
Paso 2:	Efectuar la función para calcular los subgrupos
Paso 3:	**Para** cada subgrupo (s) de la clase minoritaria **hacer**: Calcular instancias a generar en cada subgrupo T=($G/\|P\|$) * $\|s\|$
Paso 4:	**Para** cada agrupación (A) de cada subgrupo **hacer**: **while** número de instancias creadas <= T **hacer**: Seleccionar aleatoriamente una instancia (l) de A. Calcular la media (M) de cada característica i de las instancias de A, sin tomar en cuenta l, ($M_i = \mu(\{A_{1i}, A_{2i}, ..., A_{ni}\} - l_i)$). Asignar cada media del punto anterior a cada característica de la nueva instancia a generar. Agregar la nueva instancia generada a A.
Paso 5:	Definir N' como el conjunto de instancias generadas.
Paso 6:	$D = \{P \cup N \cup N'\}$
Paso 7:	**Retorna** D.

3.5. GIAC-S

La técnica de sobremuestreo GIAC-S al igual que la técnica GIAC, utiliza el análisis de la distribución de las instancias en el espacio de características de la clase minoritaria. La técnica GIAC-S hace uso de la información de los tipos de subgrupos pero a diferencia de GIAC, GIAC-S aplica la técnica de SMOTE. SMOTE consiste en generar un nueva instancia sintética con base en la media de dos instancias seleccionadas aleatoriamente. Para el caso de GIAC-S cada instancia sintética es generada con base en la media de las características de las dos instancias aleatorias seleccionadas. Las instancias aleatorias seleccionadas forman parte de una misma agrupación. Cada agrupación pertenece a un tipo de subgrupo de la clase minoritaria. Finalmente la técnica GIAC-S obtiene un conjunto de datos que contiene las instancias de la clase mayoritaria, las instancias de la clase minoritaria y las instancias sintéticas. A continuación se presenta el Algoritmo 2 GIAC-S.

Algoritmo 2 GIAC-S

Entrada:	N subconjunto de instancias del la clase mayoritaria. P subconjunto de instancias de la clase minoritaria. k cantidad de vecinos más cercanos
Salida:	$D = \{P \cup N \cup N'\}$ conjunto de datos sin desbalance de clases.
Paso 1:	Calcular instancias a generar G=(R_D*$\|P\|$) - $\|P\|$
Paso 2:	Efectuar la función para calcular los subgrupos
Paso 3:	**Para** cada subgrupo (s) de la clase minoritaria **hacer**: Calcular instancias a generar en cada subgrupo T=($G/\|P\|$) * $\|s\|$
Paso 4:	**Para** cada agrupación (A) de cada subgrupo **hacer**: **while** número de instancias creadas <= T **hacer**: Seleccionar aleatoriamente dos instancias (l, q) de A. Calcular la media (M) de cada característica i de las dos instancias seleccionadas (l, q), $(M_i = \mu(\{l_i, q_i\}))$. Asignar cada media del punto anterior a cada característica de la nueva instancia a generar. Agregar la nueva instancia generada a A.
Paso 5:	Definir N' como el conjunto de instancias generadas.
Paso 6:	$D = \{P \cup N \cup N'\}$
Paso 7:	**Retorna** D.

3.6. Consideraciones finales

Con lo expuesto en las secciones previas de este Capítulo, podemos identificar que nuestras técnicas propuestas se distinguen en dos aspectos. El primer aspecto es que nuestras técnicas utilizan la información del análisis de la estructura de la clase minoritaria para generar una

cantidad de instancias sintéticas en cada subgrupo de la clase minoritaria de manera equitativa. El segundo aspecto es que nuestras técnicas propuestas no eliminan instancias aisladas, para no perder información real del conjunto de datos. A diferencia de técnicas existentes en el estado del arte, que generan o eliminan instancias de forma aleatoria como lo son: el sobremuestreo aleatorio, submuestreo aleatorio, $NearMiss$, $SMOTEENN$, $SMOTETomek$, entre otros.

Capítulo 4

Experimentación y resultados

En este capítulo se presentan los resultados experimentales de las comparaciones de las técnicas propuestas GIAC y GIAC-S contra las técnicas del estados del arte. Además, se describen los conjuntos de datos utilizados en los experimentos.

4.1. Conjuntos de datos

Para los experimentos de este trabajo de investigación, se utilizaron siete conjuntos de datos obtenidos del repositorio de aprendizaje de máquina, de la Universidad de California en Irvine (*machine learning repository* UCI) [55]. Los conjuntos de datos presentan diferentes razones de desbalance que van desde 1.14 hasta 14.2. El número de instancias

por conjunto de datos varía desde 185 hasta 2584. En la Tabla 4.1 se muestra información a detalle de cada conjunto de datos.

Tabla 4.1: Descripción de conjunto de datos.

Conjunto de datos	Número de instancias	Número de atributos	Clase mayoritaria	Clase minoritaria	Razón desbalance (R_D)
Haberman's Survival	306	4	225	81	2.78
German Credit	1000	21	700	300	2.33
Pima Indians Diabetes	768	9	500	268	1.86
Seismic-bumps	2584	12	2414	170	14.2
Connectionist Bench	208	61	111	97	1.14
Thyroid Disease	185	6	150	35	4.29
Blood Transfusion	748	5	570	178	3.20

Haberman's Survival [56] es un conjunto de datos que contiene casos de un estudio que se llevó a cabo entre 1958 y 1970 en el Hospital Billings de la Universidad de Chicago, sobre la supervivencia de pacientes que se habían sometido a una cirugía de cáncer de mama. El número de instancias que contiene el conjunto de datos es de 306. El número de características por instancias es de 4. El conjunto de datos tiene una razón de desbalance de 2.78.

German Credit [57] fue proporcionado por el profesor Hofmann en Hamburgo. Este conjunto de datos caracteriza a 1000 personas, sobre el riego crediticio. Cada individuo (instancias) está descrito con 21 ca-

racterísticas. Este conjunto de datos tiene una razón de desbalance de 2.33.

Pima Indians Diabetes[58] es un conjunto de datos que contiene información sobre los pacientes que sufren la enfermedad de diabetes. El número de instancias que contiene el conjunto de datos son 768, el número de características por instancias son 9 y tiene una razón de desbalance de 1.86.

Seismic-bumps[59] es un conjuntos de datos de clasificación binaria menos conocido que, captura las condiciones geológicas utilizando sistemas sísmicos y sismo-acústicos en minas de carbón de tajo largo para evaluar si son propensos a que el estallido de rocas cause riesgos sísmicos o no. El conjunto de datos cuenta con 2584 instancias, donde cada instancia tiene 12 características y una razón de desbalance es de 14.2.

Connectionist Bench[60] es un conjunto de datos que contiene señales de respuestas obtenidas por frecuencias de sonda, enviadas contra un campo minado con el objeto de identificar si es una roca o una mina. El número de instancias que contiene el conjunto de datos es 208, el número de características por instancias es 61 y tiene una razón de desbalance de 1.14.

Thyroid Disease[61] es un conjunto de datos de clasificación para la detección de enfermedades de la tiroides, donde los pacientes diagnosti-

cados con hipotiroidismo tienen anomalías frente a pacientes normales. El conjunto de datos tiene un total de 185 instancias, el número de características por instancias es 6 y tiene una razón de desbalance de 4.29.

Blood Transfusion[62] este conjunto de datos contiene información de donantes del centro de servicios de transfusión de sangre en la ciudad de Hsin-Chu en Taiwán, donde determina si el donante realizó o no realizo, la donación de sangre. El número de instancias que contiene el conjunto de datos es 748, cada instancia cuenta con 5 características y la razón de desbalance es de 3.20.

4.2. Descripción de experimentos

Los experimentos fueron realizados utilizando una estación de trabajo con sistema operativo Ubuntu 18.04, con procesador Intel core i5 9 generación, con 8 GB de memoria de acceso aleatorio (RAM) y 256 GB de almacenamiento de estado solido. Para el desarrollo de los modelos de clasificación, se utilizó el lenguaje de programación Python 3.7.6. Además se utilizaron las bibliotecas $imbalanced-learn$ [63] y $scikit-learn$ [64]. Se implementaron siete técnicas del estado del arte que mitigan el desbalance de clases, las cuales son: sobremuestreo, submuestreo, SMOTE, SMOTETomek Links, SMOTEENN, NearMiss y SMOTETomek. Los modelos de clasificación utilizados fueron generados por un método de ensamble de modelos, que implementó cinco

técnicas de aprendizaje de máquina: redes neuronales, máquinas de vectores de soporte, $k-$vecino más cercano, árboles de decisión y bosque aleatorio.

Para la evaluación de los modelos de clasificación utilizamos un esquema de validación cruzada $k-folds$. Para los resultados de este trabajo a cada algoritmo se le realizaron cinco ejecuciones utilizando $k-fold$ *cross validation* con un valor de $k = 5$. Lo que significa que cada conjunto de datos fue dividido en cinco particiones distintas en las cuales se obtuvieron y promediaron los valores de las métricas de clasificación: precisión ($precision$), sensibilidad ($recall$), el valor-F ($F1-score$) y la exactitud ($accuracy$).

4.3. Resultados del desempeño de las propuestas.

Para la creación de los subgrupos de nuestra propuesta, se utilizó la distancia $Manhatan$ para calcular los $k-$vecinos más cercanos. Se ha usado $k = 9$, con base en experimentos que se realizaron con diferentes valores de k. En la Tabla 4.2 se puede observar el número de agrupación que obtuvo cada tipo de subgrupo en la clase minoritaria. Para cada conjunto de datos, después de realizar el análisis de la estructura de la clase minoritaria.

Tabla 4.2: Descripción del número de agrupaciones de cada tipo de subgrupo en cada conjunto de datos.

Conjunto de datos	Disyunciones	Casos raros	Zona frontera	Zona segura
Haberman's Survival	14	5	8	11
German Credit	19	14	38	12
Pima Indians Diabetes	17	10	29	22
Seismic-bumps	11	21	11	8
Connectionist Bench	6	4	9	15
Thyroid Disease	2	0	5	7
Blood Trans-fusion	11	8	14	21

Como hemos descrito anteriormente, para la evaluación de los modelos de clasificación utilizamos: precisión, sensibilidad, valor-F y exactitud. Los resultados para cada métrica se expresan como el par (a,b), en donde a representa la clase mayoritaria y b representa la clase minoritaria. Cada experimento se realizó cinco veces y cada vez se utiliza un esquema 5-*fold cross validation*.

En las Tablas 4.3 a la 4.9 se pueden observar los resultados obtenidos en los diferentes conjuntos de datos donde se resalta el valor más alto de la clase minoritaria obtenido en cada métrica de desempeño. En nuestros experimentos, nuestras técnicas mostraron ser superiores o similares a las técnicas del estado del arte en términos de precisión, sensibilidad, valor-F y exactitud de clasificación.

En los resultados del conjunto de datos *Haberman's Survival* descritos en la Tabla 4.3. El mejor resultado en precisión para la clase minoritaria lo obtuvo nuestra técnica GIAC con una precisión de 0.55.

El mejor comportamiento entre las técnicas del estado del arte lo obtuvo la técnica SMOTEENN con una precisión de 0.53 en la clase minoritaria. El peor comportamiento lo tienen las técnicas de submuestreo y NearMiss con un 0.43 de precisión. En las demás métricas, el resultado fue el siguiente. Para sensibilidad, nuestra propuesta GIAC-S obtuvo el mejor resultado de 0.67. Para las métricas valor-F y exactitud, GIAC obtuvo el mejor resultado 0.59 y 75 %, respectivamente.

Tabla 4.3: Resultados de las métricas del desempeño de la clasificación del conjunto de datos *Haberman's Survival*.

Técnica	Precisión	Sensibilidad	Valor-F	Exactitud
Caso base	0.74 , 0.48	0.88 , 0.26	0.80 , 0.33	70 %
GIAC	0.85 , **0.55**	0.80 , 0.63	0.82 , **0.59**	**75** %
GIAC-S	0.85 , 0.52	0.74 , **0.67**	0.79 , 0.58	72 %
Sobremuestreo	0.83 , 0.47	0.71 , 0.64	0.76 , 0.54	69 %
Submuestreo	0.76 , 0.43	0.76 , 0.43	0.76 , 0.43	66 %
SMOTE	0.83 , 0.54	0.79 , 0.60	0.81 , 0.56	74 %
SMOTETomek Links	0.78 , 0.49	0.81 , 0.44	0.80 , 0.46	71 %
SMOTEENN	0.83 , 0.53	0.77 , 0.62	0.80 , 0.56	72 %
Nearmiss	0.82 , 0.43	0.65 , 0.65	0.72 , 0.52	65 %
SMOTETomek	0.85 , 0.53	0.75 , 0.67	0.79 , 0.59	73 %

Los resultados de los modelos de clasificación para el conjunto de datos *German Credit*, descritos en la Tabla 4.4, muestran que nuestras propuestas GIAC y GIAC-S obtuvieron los mejores resultados en las métricas de clasificación en la clase minoritaria. Nuestras técnicas propuestas obtuvieron 0.66 y 0.64 en precisión, 0.72 y 0.78 en sensibilidad, y 0.69 y 0.70 en valor-F, respectivamente.

Tabla 4.4: Resultados de las métricas del desempeño de la clasificación del conjunto de datos *German Credit*.

Técnica	Precisión	Sensibilidad	Valor-F	Exactitud
Caso base	0.78 , 0.49	0.81 , 0.44	0.80 , 0.46	76 %
GIAC	0.88 , **0.66**	0.84 , 0.72	0.86 , 0.69	**80** %
GIAC-S	0.90 , 0.64	0.81 , **0.78**	0.85 , **0.70**	**80** %
Sobremuestreo	0.82 , 0.60	0.83 , 0.58	0.83 , 0.59	76 %
Submuestreo	0.84 , 0.61	0.82 , 0.64	0.83 , 0.62	77 %
SMOTE	0.86 , 0.64	0.83 , 0.68	0.84 , 0.65	79 %
SMOTETomek Links	0.81 , 0.62	0.86 , 0.54	0.84 , 0.57	76 %
SMOTEENN	0.88 , 0.60	0.79 , 0.74	0.83 , 0.66	77 %
Nearmiss	0.86 , 0.46	0.62 , 0.77	0.72 , 0.58	66 %
SMOTETomek	0.88 , 0.65	0.83 , 0.73	0.85 , 0.68	**80** %

En la Tabla 4.5 se puede observar que en el conjunto de datos *Pima Indians Diabetes*, nuestro modelo GIAC obtuvo el mejor resultado en la métrica de precisión con un valor de 0.70, mientras que en las metricas sensibilidad y valor-F, las técnicas SMOTEENN y SMOTETomek obtuvieron los mejores resultados 0.85 y 0.73, respectivamente. Para conjuntos de datos con una alta razón de desbalance. Por ejemplo, *Seismic-bumps*, que tiene una razón de desbalance de 14.2. Los resultados obtenidos por nuestras técnicas fueron competentes. En la Tabla 4.6, se puede observar que la técnica propuesta GIAC, obtuvo una precisión de 0.44. El modelo que utilizó la técnica SMOTEEN obtuvo el mejor valor en sensibilidad con 0.84. Destacamos los resultados del modelo de clasificación del caso base (sin utilizar ninguna técnica que mitigue el problema de desbalance), con 0.05 en precisión, 0 para sensibilidad, 0.01 para el valor-F, pero 92 % en exactitud. Esto es un claro ejemplo

que el modelo de clasificación está sesgado hacia la clase mayoritaria, clasificando incorrectamente las instancias de la clase minoritaria, pero su valor de exactitud es alto.

Tabla 4.5: Resultados de las métricas del desempeño de la clasificación del conjunto de datos *Pima Indians Diabetes*.

Técnica	Precisión	Sensibilidad	Valor-F	Exactitud
Caso base	0.79 , 0.69	0.85 , 0.58	0.82 , 0.63	75 %
GIAC	0.86 , **0.70**	0.82 , 0.74	0.84 , 0.71	79 %
GIAC-S	0.83 , 0.69	0.83 , 0.68	0.83 , 0.68	78 %
Sobremuestreo	0.82 , 0.66	0.82 , 0.65	0.81 , 0.65	76 %
Submuestreo	0.83 , 0.68	0.82 , 0.68	0.82 , 0.67	77 %
SMOTE	0.87 , 0.67	0.79 , 0.77	0.82 , 0.71	78 %
SMOTETomek Links	0.83 , 0.69	0.82 , 0.68	0.82 , 0.68	77 %
SMOTEENN	0.90 , 0.62	0.72 , **0.85**	0.80 , 0.72	77 %
Nearmiss	0.84 , 0.63	0.77 , 0.71	0.80 , 0.67	75 %
SMOTETomek	0.88 , 0.68	0.80 , 0.79	0.83 , **0.73**	**80** %

Tabla 4.6: Resultados de las métricas del desempeño de la clasificación del conjunto de datos *Seismic-bumps*.

Técnica	Precisión	Sensibilidad	Valor-F	Exactitud
Caso base	0.92 , 0.05	0.98 , 0.00	0.96 , 0.01	92 %
GIAC	0.95 , **0.44**	0.98 , 0.24	0.96 , 0.31	**93** %
GIAC-S	0.98 , 0.34	0.89 , 0.81	0.93 , **0.48**	88 %
Sobremuestreo	0.95 , 0.24	0.94 , 0.26	0.94 , 0.24	90 %
Submuestreo	0.97 , 0.17	0.76 , 0.68	0.86 , 0.27	76 %
SMOTE	0.96 , 0.43	0.96 , 0.45	0.96 , 0.43	92 %
SMOTETomek Links	0.94 , 0.25	1.00 , 0.04	0.96 , 0.07	**93** %
SMOTEENN	0.99 , 0.25	0.82 , **0.84**	0.90 , 0.39	82 %
Nearmiss	0.98 , 0.11	0.53 , 0.83	0.69 , 0.19	55 %
SMOTETomek	0.98 , 0.34	0.89 , 0.80	0.93 , 0.48	88 %

Para conjuntos de datos con una baja razón de desbalance de clases, como es el caso del conjunto de datos *Connectionist Bench*, que cuenta con una razón de desbalance de 1.14. El modelo de clasificación que utilizó nuestra propuesta GIAC fue el que en general obtuvo los mejores valores de las métricas, 0.89, 0.81, 0.84 en precision, sensibilidad y valor-F, respectivamente, como se muestra en la Tabla 4.7. En estos resultados experimentales, se observó que, en general, todas las técnicas del estado del arte para conjuntos de datos con razón de desbalance pequeño (menor a dos) son capaces de brindar instancias representativas del conjunto de datos y así, poder generar modelos de clasificación correctos.

Tabla 4.7: Resultados de las métricas del desempeño de la clasificación del conjunto de datos *Connectionist Bench*.

Técnica	Precisión	Sensibilidad	Valor-F	Exactitud
Caso base	0.80 , 0.87	0.90 , 0.75	0.84 , 0.80	83 %
GIAC	0.85 , **0.89**	0.91 , 0.81	0.88 , **0.84**	**86** %
GIAC-S	0.83 , 0.88	0.90 , 0.78	0.86 , 0.82	85 %
Sobremuestreo	0.83 , 0.88	0.90 , 0.77	0.86 , 0.81	84 %
Submuestreo	0.81 , 0.86	0.89 , 0.76	0.85 , 0.80	83 %
SMOTE	0.83 , 0.87	0.90 , 0.77	0.86 , 0.81	84 %
SMOTETomek Links	0.81 , 0.86	0.89 , 0.75	0.85 , 0.80	83 %
SMOTEENN	0.80 , 0.88	0.92 , 0.72	0.85 , 0.79	83 %
Nearmiss	0.85 , 0.81	0.84 , **0.82**	0.84 , 0.81	83 %
SMOTETomek	0.83 , 0.86	0.89 , 0.78	0.85 , 0.81	84 %

El conjunto de datos *Thyroid Disease*, contiene una razón de desbalance de 4.29, sin embargo dicha razón no afecta el desempeño de los

modelos de clasificación. Al realizar el análisis de la estructura de la clase minoritaria, se observó que este conjunto de datos no contiene casos raros y además el número de instancias en subgrupos de zonas seguras era grande. Por lo tanto, podemos concluir que, las clases están bien separadas. En los resultados que se muestran en la Tabla 4.8 se aprecia que el modelo de clasificación del caso base obtuvo buenos resultados 0.99, 0.91 y 0.94 en precisión, sensibilidad y valor-F, respectivamente. Destacamos en estos resultados que, a pesar de que el conjunto de datos tiene un problema de desbalance de clases. Este problema no fue impedimento para obtener modelos de clasificación correctos.

Tabla 4.8: Resultados de las métricas del desempeño de la clasificación del conjunto de datos *Thyroid Disease*.

Técnica	Precisión	Sensibilidad	Valor-F	Exactitud
Caso base	0.98 , 0.99	1.00 , 0.91	0.99 , 0.94	98 %
GIAC	0.98 , 0.99	1.00 , 0.92	0.99 , 0.96	**99** %
GIAC-S	1.00 , 0.97	0.99 , **0.99**	1.00 , **0.98**	**99** %
Sobremuestreo	0.98 , 0.97	0.99 , 0.90	0.98 , 0.92	97 %
Submuestreo	0.99 , 0.92	0.98 , 0.96	0.98 , 0.93	97 %
SMOTE	1.00 , 0.97	0.99 , 0.99	0.99 , 0.98	**99** %
SMOTETomek Links	0.98 , **1.00**	1.00 , 0.90	0.99 , 0.94	98 %
SMOTEENN	0.98 , 0.98	0.99 , 0.91	0.99 , 0.93	98 %
Nearmiss	0.98 , 0.97	0.99 , 0.92	0.99 , 0.94	98 %
SMOTETomek	1.00 , 0.97	0.99 , 0.99	1.00 , 0.98	**99** %

Para el conjunto de datos *Blood Transfusion*, el comportamiento de nuestra propuesta GIAC fue similar a la de la técnica del estado del arte SMOTEENN. En la Tabla 4.9, se muestran los resultados, donde

nuestra técnica propuesta, GIAC, obtuvo una precisión de 0.56, un valor de sensibilidad de 0.73, un valor-F de 0.63 y 75 % en exactitud.

Tabla 4.9: Resultados de las métricas del desempeño de la clasificación del conjunto de datos *Blood Transfusion*.

Técnica	Precisión	Sensibilidad	Valor-F	Exactitud
Caso base	0.79 , 0.37	0.54 , 0.65	0.64 , 0.47	57 %
GIAC	0.87 , **0.56**	0.76 , **0.73**	0.81 , **0.63**	**75** %
GIAC-S	0.85 , 0.55	0.77 , 0.68	0.81 , 0.61	74 %
Sobremuestreo	0.82 , 0.50	0.75 , 0.59	0.78 , 0.54	71 %
Submuestreo	0.85 , 0.55	0.77 , 0.67	0.81 , 0.60	74 %
SMOTE	0.86 , 0.51	0.72 , 0.71	0.78 , 0.60	72 %
SMOTETomek Links	0.83 , 0.56	0.81 , 0.59	0.82 , 0.57	74 %
SMOTEENN	0.87 , 0.56	0.76 , 0.72	0.81 , 0.63	**75** %
Nearmiss	0.78 , 0.49	0.81 , 0.44	0.80 , 0.46	74 %
SMOTETomek	0.83 , 0.54	0.79 , 0.60	0.81 , 0.57	73 %

Las técnicas GIAC y GIAC-S están disponibles para su descarga y uso en el sitio WEB: https://github.com/Cecyad/GIAC-y-GIAC-S.

Capítulo 5

Conclusiones y trabajos futuros

El problema del desbalance de clases es un obstáculo para los modelos de clasificación, ya que, puede ocasionar un deterioro en su capacidad de generalización. En particular, este problema provoca que los modelos tiendan a ignorar la clase minoritaria y sesgar su resultado hacia la clase mayoritaria.

En este trabajo de tesis, se propusieron dos nuevas técnicas a nivel de datos, basadas en el análisis de la estructura de la clase minoritaria. Se evaluaron las técnicas propuestas GIAC y GIAC-S contra otras técnicas a nivel de datos reportadas en el estado del arte. En nuestros experimentos utilizamos siete conjuntos de datos con distintos grados de desbalance. La experimentación con dichos conjuntos de datos, nos

permitió analizar el desempeño de nuestras propuestas GIAC y GIAC-S para mitigar el problema de desbalance de clases a nivel de datos. Para el conjunto de datos *Seismic-bumps* el cual tiene una alta razón de desbalance nuestra propuesta GIAC obtuvo el mejor valor en precisión con respecto a las técnicas comparativas. Para conjuntos de datos con razón de desbalance menor a dos, nuestras propuestas al igual que las técnicas del estado del arte generaron modelos de clasificación correctos. En general los modelos de clasificación que utilizaron nuestras técnicas obtuvieron instancias representativas de los conjuntos de datos y así, consiguieron buenos rendimientos en términos de precisión, sensibilidad y valor-F en la clase minoritaria. Como trabajo futuro de esta tesis proponemos extender nuestro análisis a conjuntos de datos multiclase y de alta dimensionalidad.

Apéndice A

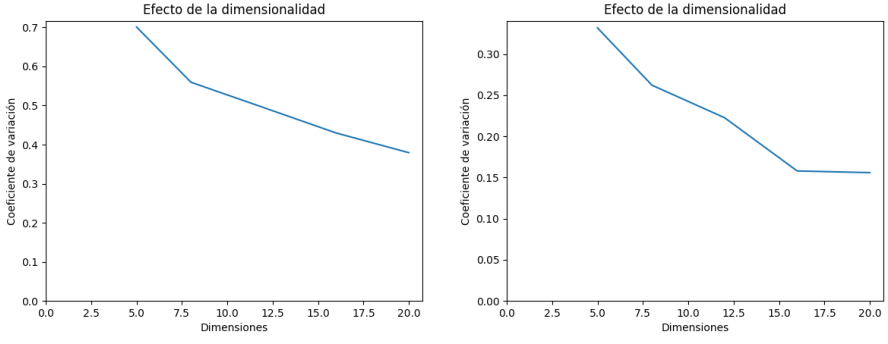
Dimensionalidad del conjunto de datos

El aumento en la dimensionalidad en un conjunto de datos conduce a un aumento en volumen de espacio de instancias, en general, las instancias no se dispersan uniformemente en el espacio de características. Por medio de la medida de similitud *Manhatan*, podemos identificar, las distancias de los vecinos de un punto en un espacio de características y de esas distancias calcular el coeficiente de variación.

El coeficiente de variación (CV), nos ofrece información respecto de la dispersión relativa del conjunto de datos, es decir es la relación entre la desviación típica (σ) de una muestra con su media ($\bar{X}$) y se calcula utilizando la Ecuación A.1.

$$CV = \frac{\sigma}{\bar{X}} \tag{A.1}$$

Para observar el efecto de la dimensionalidad se evaluó el respectivo coeficiente de variación de dos conjuntos de datos, con 5,8,12,16 y 20 dimensiones, por lo que se puede apreciar, que a medida de que las dimensiones de un conjunto de datos aumentan, el coeficiente de variación disminuye, debido a que existe poca variabilidad entre las instancias, es decir el efecto es que las instancias rara vez están más cerca de las instancias que consideraríamos similares, ver Figura A.1.



(a) Conjunto de datos Sonar. (b) Conjunto de datos German.

Figura A.1: Efecto del aumento de dimensiones en conjuntos de datos.

Bibliografía

[1] Bartosz Krawczyk. Learning from Imbalanced Data: Open Challenges and Future Directions. *Progress in Artificial Intelligence*, 5:221–232, 2016.

[2] K. Thirunavukkarasu, A. S. Singh, P. Rai, and S. Gupta. Classification of Iris Dataset Using Classification Based KNN Algorithm in Supervised Learning. In *In Proceedings of the Fourth International Conference on Computing Communication and Automation (ICCCA)*, pages 1–4, 2018.

[3] C. Guo, Y. Ma, Z. Xu, M. Cao, and Q. Yao. An Improved Oversampling Method for Imbalanced Data–SMOTE Based on Canopy and K-means. In *2019 Chinese Automation Congress (CAC)*, pages 1467–1469, 2019.

[4] T. Zhou, W. Liu, C. Zhou, and L. Chen. Gan-based Semi-supervised for Imbalanced Data Classification. In *In Proceedings of*

the Fourth International Conference on Information Management (ICIM), pages 17–21, 2018.

[5] Z. Lee, C. Lee, S. Chou, W. Ma, F. Ye, and Z. Chen. A Distributed Intelligent Algorithm Applied to Imbalanced Data. In *2019 IEEE International Conference of Intelligent Applied Systems on Engineering (ICIASE)*, pages 136–138, 2019.

[6] C. Zhang, W. Gao, J. Song, and J. Jiang. An Imbalanced Data Classification Algorithm of Improved Autoencoder Neural Network. In *2016 Eighth International Conference on Advanced Computational Intelligence (ICACI)*, pages 95–99, 2016.

[7] Y. Pristyanto and A. Dahlan. Hybrid Resampling for Imbalanced Class Handling on Web Phishing Classification Dataset. In *In Proceedings of the Fourth International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*, pages 401–406, 2019.

[8] Rishabh Rustogi and Ayush Prasad. Swift Imbalance Data Classification Using SMOTE and Extreme Learning Machine. In *2019 International Conference on Computational Intelligence in Data Science (ICCIDS)*, pages 1–6, 2019.

[9] A. Amin, S. Anwar, A. Adnan, M. Nawaz, N. Howard, J. Qadir, A. Hawalah, and A. Hussain. Comparing Oversampling Techni-

ques to Handle the Class Imbalance Problem: A Customer Churn Prediction Case Study. *IEEE Access*, 4:7940–7957, 2016.

[10] Nathalie Japkowicz and Shaju Stephen. The Class Imbalance Problem: A Systematic Study. *Intelligent Data Analysis*, pages 429–449, 2002.

[11] Krystyna Napierała, Jerzy Stefanowski, and Szymon Wilk. Learning from Imbalanced Data in Presence of Noisy and Borderline Examples. In *Rough Sets and Current Trends in Computing*, pages 158–167, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[12] Victoria López, Alberto Fernández, Salvador García, Vasile Palade, and Francisco Herrera. An Insight Into Classification with Imbalanced Data: Empirical Results and Current Trends on Using Data Intrinsic Characteristics. *Information Sciences*, 250:113 – 141, 2013.

[13] Y. Frayman, Kai Ming Ting, and Lipo Wang. A Fuzzy Neural Network for Data Mining: Dealing with the Problem of Small Disjuncts. In *IJCNN'99. International Joint Conference on Neural Networks. Proceedings (Cat. No.99CH36339)*, volume 4, pages 2490–2493 vol.4, 1999.

[14] Gary M. Weiss. Learning with Rare Cases and Small Disjuncts. In Armand Prieditis and Stuart Russell, editors, *Machine Learning*

Proceedings 1995, pages 558 – 565. Morgan Kaufmann, San Francisco (CA), 1995.

[15] V. García, J.S. Sánchez, A.I. Marqués, R. Florencia, and G. Rivera. Understanding the Apparent Superiority of Over-sampling Through an Analysis of Local Information for Class-imbalanced Data. *Expert Systems with Applications*, 158:113026, 2020.

[16] O. Simeone. A Very Brief Introduction to Machine Learning with Applications to Communication Systems. *IEEE Transactions on Cognitive Communications and Networking*, 4(4):648–664, 2018.

[17] Igor Kononenko and Matjaz Kukar. *Machine Learning and Data Mining: Introduction to Principles and Algorithms*. Horwood Publishing Limited, 2007.

[18] S. R. Guruvayur and R. Suchithra. A Detailed Study on Machine Learning Techniques for Data Mining. In *2017 International Conference on Trends in Electronics and Informatics (ICEI)*, pages 1187–1192, 2017.

[19] Aida Ali, Siti Mariyam Hj. Shamsuddin, and Anca L. Ralescu. Classification with Class Imbalance Problem: A Review. In *SOCO 2015*, 2015.

[20] Nathalie Japkowicz. Concept-learning in the Presence of Between-Class and Within-Class Imbalances. pages 67–77, 06 2001.

[21] Z. Liu, B. Xiang, W. Guo, Y. Chen, K. Guo, and J. Zheng. Overlapping Community Detection Algorithm Based on Coarsening and Local Overlapping Modularity. *IEEE Access*, 7:57943–57955, 2019.

[22] Han Kyu Lee and Seoung Bum Kim. An Overlap-sensitive Margin Classifier for Imbalanced and Overlapping Data. *Expert Systems with Applications*, 98:72–83, 2018.

[23] Pattaramon Vuttipittayamongkol, Eyad Elyan, and Andrei Petrovski. On the Class Overlap Problem in Imbalanced Data Classification. *Knowledge-Based Systems*, 212:106631, 2021.

[24] Haitao Xiong, Ming Li, Tongqiang Jiang, and Shouxiang Zhao. Classification Algorithm Based on NB for Class Overlapping Problem. *Applied Mathematics and Information Sciences*, 7:409–415, 06 2013.

[25] Krystyna Napierala and Jerzy Stefanowski. Types of Minority Class Examples and their Influence on Learning Classifiers from Imbalanced Data. *Journal of Intelligent Information Systems*, 46(3):563–597, jul 2015.

[26] M. Kubat. Addressing the Curse of Imbalanced Training Sets: One-Sided Selection. *Fourteenth International Conference on Machine Learning*, 06 2000.

[27] C.M. Bishop. *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer, 2006.

[28] Y. Ozbay and B. Karlik. A Recognition of ECG Arrhytihemias Using Artificial Neural Networks. In *2001 Conference Proceedings of the 23rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, volume 2, pages 1680–1683 vol.2, 2001.

[29] Yu Wanjun and Song Xiaoguang. Research on Text Categorization Based on Machine Learning. *ICAMS 2010 - In proceedings of 2010 IEEE International Conference on Advanced Management Science*, 2, 07 2010.

[30] J. M. Keller, M. R. Gray, and J. A. Givens. A Fuzzy K-nearest Neighbor Algorithm. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-15(4):580–585, 1985.

[31] S. S. Gavankar and S. D. Sawarkar. Eager Decision Tree. In *2017 2nd International Conference for Convergence in Technology (I2CT)*, pages 837–840, 2017.

[32] J. R. Quinlan. Induction of Decision Trees. *Mach. Learn.*, 1(1):81–106, March 1986.

[33] S. L. Fink, C. Ratke, H. Hoffmann, and H. Nehring. Gaussian Decicion Tree Induction Case Study. In *2018 1st International Conference on Computer Applications Information Security (ICCAIS)*, pages 1–6, 2018.

[34] Leo Breiman. Random Forests. *Mach. Learn.*, 45(1):5–32, October 2001.

[35] Charu C. Aggarwal and Saket Sathe. Theoretical Foundations and Algorithms for Outlier Ensembles. *SIGKDD Explor. Newsl.*, 17(1):24–47, September 2015.

[36] Leo Breiman. Bagging Predictors. *Mach. Learn.*, 24(2):123–140, aug 1996.

[37] Victoria López, Alberto Fernández, and Francisco Herrera. On the Importance of the Validation Technique for Classification with Imbalanced Datasets: Addressing Covariate Shift when Data is Skewed. *Information Sciences*, 257:1–13, 02 2014.

[38] Ludmila I. Kuncheva. *Combining Pattern Classifiers: Methods and Algorithms*. Wiley-Interscience, USA, 2004.

[39] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. SMOTE: Synthetic Minority Over-sampling Technique. *J. Artif. Intell. Res. (JAIR)*, 16:321–357, 2002.

[40] Georgios Douzas, Fernando Bação, and Felix Last. Improving Imbalanced Learning Through a Heuristic Oversampling Method Based on K-Means and SMOTE. *Information Sciences*, 465, 06 2018.

[41] T. Hasanin and T. Khoshgoftaar. The Effects of Random Undersampling with Simulated Class Imbalance for Big Data. In *2018*

IEEE International Conference on Information Reuse and Integration (IRI), pages 70–79, 2018.

[42] S. Sharma, C. Bellinger, B. Krawczyk, O. Zaiane, and N. Japkowicz. Synthetic Oversampling with the Majority Class: A New Perspective on Handling Extreme Imbalance. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 447–456, 2018.

[43] Y. Sun, L. Cai, B. Liao, and W. Zhu. Minority Sub-region Estimation-based Oversampling for Imbalance Learning. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–1, 2020.

[44] Arjun Puri and Manoj Gupta. Improved Hybrid Bag-Boost Ensemble with K-Means-SMOTE–ENN Technique for Handling Noisy Class Imbalanced Data. *The Computer Journal*, 05 2021.

[45] B. Elkarami, A. Alkhateeb, and L. Rueda. Cost-sensitive Classification on Class-balanced Ensembles for Imbalanced Non-coding RNA Data. In *2016 IEEE EMBS International Student Conference (ISC)*, pages 1–4, 2016.

[46] Y. ZHANG, R. LU, J. HUANG, and D. GAO. Evolutionary-based Ensemble Under-Sampling for Imbalanced Data. In *In Proceedings of the Sixteen International Computer Conference on Wavelet Active Media Technology and Information Processing*, pages 212–216, 2019.

[47] J. Zhai, J. Qi, and S. Zhang. Binary Imbalanced Data Classification Based on Modified D2GAN Oversampling and Classifier Fusion. *IEEE Access*, 8:169456–169469, 2020.

[48] Cati Olmo, German Sanchez, Francesc Prats, Nutria Agell, and Monica Sanchez. Using Orders of Magnitude and Nominal Variables to Construct Fuzzy Partitions. In *2007 IEEE International Fuzzy Systems Conference*, pages 1–6, 2007.

[49] N. Sharma and K. Saroha. A Novel Dimensionality Reduction Method for Cancer Dataset Using PCA and Feature Ranking. In *2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 2261–2264, 2015.

[50] V. N. G. Raju, K. P. Lakshmi, V. M. Jain, A. Kalidindi, and V. Padma. Study the Influence of Normalization/Transformation Process on the Accuracy of Supervised Classification. In *2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT)*, pages 729–735, 2020.

[51] M. D. Malkauthekar. Analysis of Euclidean Distance and Manhattan Distance Measure in Face Recognition. In *In Proceedings of the Third International Conference on Computational Intelligence and Information Technology (CIIT 2013)*, pages 503–507, 2013.

[52] Shruti Kapil and Meenu Chawla. Performance Evaluation of K-means Clustering Algorithm with Various Distance Metrics. In *In*

Proceedings of the First International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES), pages 1–4, 2016.

[53] Haviluddin, Muhammad Iqbal, Gubtha Mahendra Putra, Novianti Puspitasari, Hario Jati Setyadi, Felix Andika Dwiyanto, Aji Prasetya Wibawa, and Rayner Alfred. A Performance Comparison of Euclidean, Manhattan and Minkowski Distances in K-Means Clustering. In *In Proceedings of the Six International Conference on Science in Information Technology (ICSITech)*, pages 184–188, 2020.

[54] M. Mahin, M. J. Islam, A. Khatun, and B. C. Debnath. A Comparative Study of Distance Metric Learning to Find Sub-categories of Minority Class from Imbalance Data. In *2018 International Conference on Innovation in Engineering and Technology (ICIET)*, pages 1–6, 2018.

[55] Dheeru Dua and Casey Graff. UCI Machine Learning Repository, 2017.

[56] Tjen-Sien Lim. Haberman's Survival Datasets. url : https://archive.ics.uci.edu/ml/datasets/Haberman's+Survival, 1999.

[57] Hans Hofmann. Statlog (German Credit Data) Datasets. url : https://archive.ics.uci.edu/ml/datasets/statlog+(german+ cre-

dit+data), 1994.

[58] UCI Machine Learning. Pima Indians Diabetes Datasets. url : https://www.kaggle.com/uciml/pima-indians-diabetes-database, 2016.

[59] Marek Sikora. Seismic-bumps Datasets. url : https://archive.ics.uci.edu/ml/datasets/seismic-bumps, 2013.

[60] Terry Sejnowski. Connectionist Bench (Sonar, Mines vs. Rocks) Datasets. url : http://archive.ics.uci.edu/ml/datasets/connectionist+bench+(sonar,+mines+vs.+rocks), 1988.

[61] Ross Quinlan. Thyroid Disease Datasets. url : https://archive.ics.uci.edu/ml/datasets/Thyroid+Disease, 1987.

[62] I-Cheng Yeh. Blood Transfusion Service Center Datasets. url : https://archive.ics.uci.edu/ml/datasets/Blood+Transfusion+Service+Center, 2008.

[63] Guillaume Lemaître, Fernando Nogueira, and Christos K. Aridas. Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning. *Journal of Machine Learning Research*, 18(17):1–5, 2017.

[64] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer,

Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. API Design for Machine Learning Software: Experiences from the Scikit-learn Project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122, 2013.