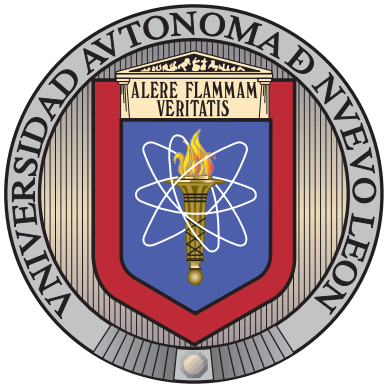
Universidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica

Subdirección de Estudios de Posgrado



Selección y construcción de características agrupadas mediante un algoritmo genético por bloques

por

Rafael Alfredo Cavazos Martínez

como requisito parcial para obtener el grado de

doctorado en ingeniería

con Orientación en Tecnologías de la Información

Septiembre 2020

Universidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica

Subdirección de Estudios de Posgrado

Selección y construcción de características agrupadas mediante un algoritmo genético por bloques

por

Rafael Alfredo Cavazos Martínez

como requisito parcial para obtener el grado de

doctorado en ingeniería

con Orientación en Tecnologías de la Información

Septiembre 2020

Universidad Autónoma de Nuevo León
Facultad de Ingeniería Mecánica y Eléctrica
Subdirección de Estudios de Posgrado

Los miembros del Comité de Tesis recomendamos que la Tesis "Selección y construcción de características agrupadas mediante un algoritmo genético por bloques", realizada por el alumno Rafael Alfredo Cavazos Martínez, con número de matrícula 1120821, sea aceptada para su defensa como requisito para obtener el grado de Doctorado en Ingeniería con Orientación en Tecnologías de la Información.

El Comité de Tesis

Dra. Sara Elena Garza Villarreal
Director

Dr. Romeo Sánchez Nigenda
Revisor

Dr. Luis Martín Torres Treviño
Revisor

Dr. Francisco Javier Barreto Trujillo
Revisor

Dr. Héctor Gibrán Ceballos Cancino
Revisor

Vo. Bo.

Dr. Simón Martínez Martínez
Subdirector de Estudios de Posgrado

SUBDIRECCIÓN DE ESTUDIOS DE POSGRADO FIME 050

San Nicolás de los Garza, Nuevo León, septiembre de 2020

Ciudad Universitaria Pedro de Alba s/n, C.P. 66455. A.P. 076 Suc. "F"
San Nicolás de los Garza, Nuevo León, México. Tels: (81) 8332 0903 /
Conm.: 8329 4020 / Fax: (81) 8332 0904

El presente trabajo de investigación lo dedico con mucho cariño a mi esposa y mis hijas quienes me apoyaron en todo momento con su amor y comprensión son fuente de mi inspiración, también así a mi director de tesis la doctora Sara Elena Garza Villarreal, gracias a su apoyo y guía me ha orientado durante este proceso.

Índice general

Índice de figuras

Lista de Tablas

Agradecimientos

Agradezco antes que nada a Dios por darme salud y fuerza, agradezco también a mi querida esposa Karla Mendoza Ramos y mis hermosas hijas Karla Daniela y Ana Sofía, ya que su amor y comprensión son motor de mi corazón, a mis padres Humberto Javier y Eduvijes, quienes me educaron por el camino el bien y sentaron las bases de mi formación.

Quiero agradecer de forma muy especial a mi director de tesis, la doctora Sara Elena Garza Villarreal con su guía, su paciencia, comprensión y su apoyo este proyecto logró llevarse a cabo. Agradezco también de forma especial al doctor Francisco Torres Guerrero por creer en mí e impulsarme a ser una mejor persona.

Agradezco a todos los doctores del programa DITI, que de forma muy particular me proporcionaron los conocimientos y me guiaron en el desarrollo de mi tesis.

Agradezco también a mis compañeros de trabajo con quienes se desarrollaron muchas de las ideas de este proyecto, en especial a Griselda Hernández, Magdaleno Zarazúa, Ramiro Siller, Fernando Castillo y todos los que de alguna forma colaboraron en la realización de este proyecto.

Resumen

Rafael Alfredo Cavazos Martínez.

Candidato para obtener el grado de Doctorado en Ingeniería con Orientación en Tecnologías de la Información.

Universidad Autónoma de Nuevo León.

Facultad de Ingeniería Mecánica y Eléctrica.

Título del estudio: Selección y construcción de características agrupadas mediante un algoritmo genético por bloques.

Número de páginas: 102.

Objetivos y método de estudio: En el sector educativo, la predicción del abandono escolar es un tema abierto y que implica el uso de múltiples características relativas al desempeño de los estudiantes. Las técnicas de aprendizaje automático se han utilizado para realizar tareas en este sentido, y en este campo se ha determinado la importancia de hacer selección de características para mejorar la calidad de la predicción, así como re-muestreo cuando el conjunto de datos cuenta con clases desbalanceadas. Se presenta un algoritmo genético por bloques para selección y construcción de características, el cual permita mejorar la calidad de predicción en conjuntos de datos donde las características están conformadas por grupos de forma anticipada. En el algoritmo genético por bloques, cada cromosoma se subdivide en grupos de características (bloques), que a su vez están divididos en cuatro secciones

de genes (donde cada gen representa una característica): genes individuales apagados, genes individuales prendidos, genes compuestos apagados y genes compuestos prendidos. Los genes compuestos representan el proceso de construcción y se manejan a través de árboles de operaciones con notación postfija. En cuanto al algoritmo genético en sí, la población inicial, la evaluación, la cruza y la mutación también son adaptadas para hacerse bajo el esquema de bloques.

CONTRIBUCIONES Y CONCLUSIONES: El algoritmo genético por bloques obtuvo una medida F promedio de 0.94 y muestra diferencia significativa en comparación con el conjunto original, con método del alpinista—tanto en su versión de selección como en su versión de construcción de características—y también con el algoritmo genético sin bloques para selección de características. Concluimos que cuando el conjunto de datos puede ser separado por bloques, el algoritmo genético propuesto logra una calidad satisfactoria de clasificación.

Firma de la asesora: Sara Elena Garza V.

Dra. Sara Elena Garza Villarreal

Capítulo 1

Introducción

El aprendizaje automático se utiliza en una gran cantidad de campos como biología, química, física, medicina, finanzas y educación, entre otros. El objetivo del aprendizaje automático es descubrir conocimiento a partir de un conjunto de datos. Se puede estudiar en dos enfoques: supervisado y no supervisado. El enfoque supervisado consiste en modelar, mediante datos etiquetados, una función que pueda predecir mejor un objetivo, el cual puede ser un valor numérico continuo (*regresión*) o un valor numérico discreto o categórico (*clasificación*).

Los conjuntos de datos se componen de instancias, las cuales a su vez se componen de características. De estas características depende la calidad de la clasificación. Sin embargo, los conjuntos de datos pueden (a) contener características irrelevantes o redundantes, e igualmente (b) contener características cuya combinación puede mejorar la calidad de la predicción. Es por esta razón que la *selección de características* estudia cómo incluir solamente las mejores características del conjunto de datos y la *construcción de características* estudia cómo combinar características para elevar la calidad de la predicción.

Existen conjuntos de datos en particular en los cuales las características se encuentran organizadas por *bloques*, es decir, están agrupadas. Lo anterior es común, por ejemplo, en conjuntos de datos que representan encuestas con diferentes cate-

gorías. En ese sentido, la selección y construcción de características debiera respetar esta organización por bloques para realizar una búsqueda más certera y eficiente. El presente trabajo está enfocado a la selección y construcción de características por bloques, aplicando estas técnicas al contexto de detección de abandono escolar en una escuela de nivel medio superior.

En cuanto al contexto de aplicación, en la minería de datos enfocada a la educación (*educational data mining*), la predicción del desempeño de estudiantes es una de las áreas más comunes según la literatura (Bakhshinategh et al., 2018). Dentro de la predicción del desempeño, la predicción del abandono escolar es una tarea similar. Aunque este tipo de casos se pueden tratar como un problema de clasificación binaria (Márquez-Vera et al., 2016), determinar las características de los estudiantes puede resultar en un extenso número de atributos de diversos factores. Por lo tanto, los conjuntos de datos pueden contener información redundante e irrelevante que afecta directamente la calidad de la clasificación (Xue et al., 2016). Además, en algunos casos presentan desbalance de clases (Márquez-Vera et al., 2016). Para eliminar estas características redundantes e irrelevantes se han propuesto las técnicas de reducción de la dimensionalidad, como la selección de características y la construcción de características, que son el foco de estudio de nuestra investigación. Como veremos más adelante, en nuestro caso de estudio, el conjunto de datos se encuentra fraccionado por *bloques*, pues existen grupos de características relacionadas. Por ejemplo, existe un grupo de características relacionadas al examen de ingreso y otro grupo de características relacionadas a los hábitos de estudio de cada alumno (entre otros grupos).

1.1 Definición del problema

En este caso, consideramos que el problema está dividido en dos apartados: (1) selección y construcción de características y (2) detección de abandono escolar. El primer subproblema estaría al servicio del segundo, y el segundo más bien podría

verse como nuestro caso de estudio.

1.1.1 Selección y construcción de características

El problema de selección de características consiste en, dado el conjunto de características de un conjunto de datos, obtener un subconjunto propio de características de tal manera que la calidad de predicción sea igual o mejor utilizando este subconjunto. De manera similar, el problema de construcción de características implica obtener un nuevo conjunto de características a partir el original, de tal manera que cada característica de este nuevo conjunto combine características del original utilizando un operador (por ejemplo, un operador aritmético); la intención es asimismo aumentar la calidad de predicción.

Formalmente, sea $F = \{f_1, f_2, \dots f_m\}$ el conjunto original de características y sea $F' = \{F_s, F_c\}$ el resultado de la selección y combinación de estas, donde $F_s \subset F$ y $F_c = \{f_i \oplus f_j : f_i, f_j \in F \wedge \oplus \in \{+, -, *, \text{máx}, \text{mín}, \dots\}\}$. Consideremos, además, una función de selección $S(F) = F_s$ y una función de construcción $C(F) = F_c$. Finalmente, dentro de estas definiciones, consideraremos $Q(X)$ como la calidad de predicción utilizando el conjunto de características X.

Definición 1. *El problema de selección y construcción consiste, entonces, en encontrar funciones $S(F)$ y $C(F)$, de tal manera que el resultado $F' = \{S(F), C(F)\}$ obtenga una calidad de predicción mayor a la calidad obtenida con el conjunto original de características F: $Q(F') > Q(F)$.*

Definición 2. *Si el conjunto de características está agrupado por bloques, de tal manera que $F = \{B_1, \dots B_n\}$ donde $B_i = \{f_j \dots f_k\}$, el problema de selección y construcción de características agrupadas podría plantearse como el problema de encontrar funciones $S_B(B_i)$ y $C_B(B_i)$ a nivel bloque. De este modo, el resultado final incorporaría el resultado de cada bloque: $\bigcup S_B(B_i) = F_s$ y $\bigcup C_B(B_i) = F_c$.*

1.2 Motivación y justificación

El método presentando en esta investigación surge del estudio un conjunto de datos enfocado a predecir el abandono escolar. En este contexto, el problema puede extenderse a un número elevado de variables, además que los valores de las variables no siempre tienen el mismo efecto ya que las personas pueden tomar decisiones diferentes ante circunstancias similares, de tal manera que se vuelve complicado utilizar pocas variables para determinar la clase a la que pertenece. El caso de estudio presentado es tan solo un área de aplicación que puede tener un beneficio enorme debido a la cantidad de alumnos que no concluyen sus estudios cada año.

Su aplicación trasciende más allá de una escuela de nivel medio superior, ya que es posible aplicar a todo el sistema educativo, otros niveles de estudio e incluso otras áreas de investigación, ya que la propuesta de investigación trabaja directamente sobre la combinación de características y su ajuste con los datos reales de las clases que se desean predecir. De esta manera, es posible incluir más características para ayudar a mejorar la predicción.

Es importante mencionar que en los últimos años se han realizado esfuerzos por diferentes investigadores en este campo con la finalidad de obtener mejores sistemas de predicción. En la investigación de Amrieh et al. (2016) enfocada a predecir el desempeño de los estudiantes, hace énfasis en el uso de técnicas de selección y construcción de características como fase del proceso para mejorar los resultados de la clasificación. De igual manera en la investigación presentada por Velmurugan y Anuradha (2016) se hace estudio de diferentes técnicas de selección de características.

1.3 Modelo de solución

Se propone utilizar un *método de envoltura* para selección y construcción de características, es decir, un método que escoja subconjuntos de características basándo-

se en la calidad obtenida mediante un clasificador. En específico, se propone utilizar un *algoritmo genético*. El rasgo principal de este algoritmo sería que estaría preparado para trabajar con los bloques de características definidos de antemano. Por lo tanto, tanto la estructura del cromosoma como los operadores genéticos serían modificados para lidiar con bloques de genes.

1.4 Protocolo

A continuación, se describe el protocolo que se utilizará en la tesis. Este incluye la hipótesis, las preguntas de investigación, los objetivos, el objeto de estudio, las variables de estudio, el alcance y las contribuciones.

1.4.1 Hipótesis

En un conjunto de datos donde las características se encuentran agrupadas de manera lógica, es posible realizar selección y construcción de características utilizando un algoritmo genético que trabaje por bloques.

1.4.2 Preguntas de investigación

1. ¿Es posible realizar la selección y construcción por bloques independientes cuando las características conforman grupos de características similares?

2. ¿Es posible representar en un genotipo (cromosoma), la información de la selección y la construcción de características para cada uno de los grupos (bloques) de características?

3. ¿Puede encontrarse mediante un algoritmo genético un nuevo conjunto de características con mejor calidad que el conjunto original?

1.4.3 Objetivos

Objetivo principal: Diseñar un algoritmo de construcción y selección de características con enfoque en grupos de características basado en un algoritmo genético capaz de optimizar la calidad de la clasificación.

Objetivos secundarios:

1. Diseñar la representación genética del conjunto de datos
2. Diseñar la función de aptitud
3. Determinar los métodos de selección, cruza y mutación
4. Selección de conjuntos de datos con características en los que se tenga conocimiento del agrupamiento de ellas.
5. Comparar los resultados contra otros métodos.

1.4.4 Objeto de estudio

El objeto de estudio es el espacio muestral de un conjunto de datos que presenta las siguientes particularidades:

- Sus características presentan una estructura de grupos.
- Se cuenta con suficientes características para realizar un proceso de selección de características
- La calidad de un clasificador utilizando el conjunto de datos completo presenta bajos resultados.

1.4.5 Variables de estudio

Variables de trabajo: Número de generaciones, porcentaje de mutación, porcentaje de cruza, balance de clases.

Variable dependiente: Cantidad de características seleccionadas, resultado de la clasificación de la clase minoritaria

Variables intervinientes: Ambiente estudiantil, situaciones, programas de intervención académica. En la educación escolarizada en México en la UANL, la modalidad presencial implica que el estudiante tiene que asistir a clases en un horario y grupo asignado, de tal manera que entendemos por *ambiente escolar* a la relación entre los actores del proceso de enseñanza, tales como el docente, compañeros de estudio, personal administrativo de la dependencia educativa, la familia, entre otros, que puedan beneficiar o afectar el proceso de enseñanza.

1.4.6 Alcance

El alcance de la investigación se encuentra en el desarrollo de un algoritmo genético, que aplica un proceso de selección y construcción de características simultáneamente, con el objetivo de mejorar la clasificación de conjuntos que presenten las siguientes características:

1. Se cuenta con suficientes atributos para requerir un proceso de reducción de la dimensionalidad
2. Se identifican previamente los grupos o categorías de características.
3. El clasificador original tiene resultados bajos en la clasificación.

La aplicación de esta técnica se centra en conjuntos de datos relativos a rendimiento estudiantil, específicamente a la detección de estudiantes en riesgo de aban-

dono escolar.

1.5 Contribuciones

Se presenta una técnica para mejorar la clasificación de conjuntos de datos en los que es posible agrupar características similares. El proceso toma cada grupo e identifica las características individuales y, de existir, las combinaciones de ellas que ayudan a predecir mejor un objetivo. El proceso evolutivo presentado permite desarrollar tanto la selección como la construcción de características en un solo proceso, mediante un cromosoma dividido en grupos de características y para cada grupo en cuatro secciones, dos para características individuales y dos para características combinadas.

Aún y cuando este proceso se basa en la clasificación de estudiantes en riesgo de abandonar sus estudios de nivel medio superior, este proceso se puede aplicar a diferentes áreas en donde el conjunto de datos cumpla con tres características: la primera es que se debe contar con suficientes características para requerir un proceso de selección y construcción, la segunda es que se identifiquen de manera previa los grupos de características (por lo menos dos características por cada grupo) y la tercera es que presente bajo rendimiento en la clasificación.

El resultado final es el cromosoma que mejor se adaptó al clasificador, con el cual se crea el nuevo conjunto de datos, y el cual tiene una dimensionalidad menor al conjunto original pero con mejores resultados en su clasificación.

1.6 Organización del documento

El presente documento se encuentra distribuido de la siguiente manera: en el capítulo 2 se explicarán las bases fundamentales en el área de aprendizaje automático, árboles de clasificación, desbalance de clases y algoritmos genéticos; después, en el capítulo 3 se describen las investigaciones relacionadas con selección de características, construcción de características, la combinación de ambas y la clasificación en el ámbito educativo; en el capítulo 4 se explicará a detalle el algoritmo genético desarrollado para la selección y construcción de características por bloques; después, en el capítulo 5 se explica el conjunto de datos utilizado para la investigación; así, en el capítulo 6 se muestran los resultados y su interpretación, por último en el capítulo 7 se discutirán las conclusiones del proceso desarrollado y el trabajo a futuro.

Capítulo 2

Marco Teórico

En este capítulo, se describen conceptos y teoría básica de los temas relacionados con el tema de investigación. Primero abordaremos los conceptos de aprendizaje automático de manera general, para después concentrarnos en el tema de clasificación con énfasis en las técnicas de árboles de decisión y bosque aleatorio. Aunado a estos temas, se abordará el tema de clases desbalanceadas con énfasis en la técnica de generación de instancias sintéticas, además de los conceptos de selección y construcción de características. Por último, se abordarán conceptos de algoritmos genéticos.

2.1 Aprendizaje automático

El aprendizaje automático se aplica en una gran cantidad de problemas de la vida, tal que se ha explorado en gran medida su desarrollo y aplicaciones. El aprendizaje automático consiste en extraer o descubrir conocimiento a partir de datos, y los campos que integran esta área son la estadística, la inteligencia artificial y las ciencias computacionales (Müller y Guido, 2016). De forma natural, las personas aprenden a partir de la experiencia, por lo que el aprendizaje básicamente es utilizar la información de la experiencia para acciones futuras. Un médico, por ejemplo, aprende a identificar un tipo de enfermedad (como cáncer) basado en la experiencia

de pacientes anteriores que se han diagnosticado; en este caso, los datos obtenidos de las experiencias previas pueden recopilarse de fuentes como imágenes de los diferentes tipos de cáncer y así a partir de ellos aprender a determinar si el paciente presenta cáncer, incluso si este es benigno o maligno (Cheng et al., 2010). En el área financiera, un banco puede tomar datos de clientes, estudiar su comportamiento en el manejo de sus cuentas y así identificar comportamientos extraños que pueden resultar en un fraude (Ryman-Tubb et al., 2018). En el contexto educativo, las instituciones de esta índole pueden tomar datos de los estudiantes, entender su comportamiento e identificar patrones que lleven a un riesgo académico (Márquez-Vera et al., 2016). Existe una gran cantidad de ejemplos y áreas de aplicación del aprendizaje automático que han generado grandes aportaciones en áreas de conocimiento como biología, medicina, astronomía, economía, física y química, entre otras.

En un sistema de clasificación del rendimiento de los estudiantes, el proceso educativo dará como resultado una calificación, de tal manera que es posible construir un modelo de las características que definen al estudiante (entrada) y compararlo con el rendimiento del estudiante (salida). Cuando se conoce la salida del proceso, el sistema puede aprender de los ejemplos reales del proceso. A este tipo de tareas en donde se conoce la salida se le llama *aprendizaje supervisado*. En ese sentido, el sistema crea un modelo predictivo a partir de los datos proporcionados (entradas) y supervisa que el modelo creado pueda clasificar otros datos de la misma especie para obtener una aproximación aceptable. Por otra parte, existen algunas tareas en los que no se conoce la salida; sin embargo, es relevante obtener información a partir de los datos. Por ejemplo, en un sitio web en donde se han registrado usuarios y estos generan información que puede indicar tendencias, es posible generar similitud con otros usuarios y así agruparlos como estrategia de negocio. A este tipo de algoritmos de aprendizaje automático que asocian los datos sin conocer la relación entre ellos se les llama algoritmos de *aprendizaje no supervisado*.

La salida de la clasificación en algunos casos se presenta como un dato continuo numérico (*regresión*) y en otras situaciones como un dato numérico discreto o

categórico (*clasificación*). En este último caso, la clasificación puede ser binaria (dos clases) o multiclase (más de dos clases). Por ejemplo, al predecir el rendimiento de un estudiante, la salida puede ser la calificación final comprendida como un número entre 0 y 100, siendo así un modelo de regresión, o una letra entre la A y la F, correspondiendo así a un modelo de clasificación multiclase; en el caso de predecir el abandono escolar como *bueno* (1) o *en riesgo de abandono* (0), correspondería a un modelo de clasificación binaria. Como podemos ver, cada uno de ellos tienen características distintas en sus estudios. La investigación presentada se centra en la clasificación binaria.

Formalmente, nos referiremos a un *conjunto de datos* como una estructura $D = (\mathcal{I}_1, \mathcal{I}_2, \ldots \mathcal{I}_n)^T$ compuesta de n *instancias*, donde cada instancia tiene la forma $\mathcal{I}_k = (f_1, f_2, \ldots f_m, c_i)$. En este caso, f_j representa una *característica* y c_i representa la *clase* de la instancia. Esta clase forma parte de un conjunto $C = \{c_1, c_2, \ldots c_p\}$ de posibles clases. En un esquema de clasificación binaria, $p = 2$. La Tabla 2.1 muestra un ejemplo de conjunto de datos con formato de tabla.

Tabla 2.1:

Conjunto de datos en forma de tabla

Instancia	f_1	f_2	f_3	f_4	**Clase**
$\mathcal{I}_1$	5.1	3.5	1.4	0.2	Iris-setosa
$\mathcal{I}_2$	4.9	3.0	1.4	0.2	Iris-setosa
$\mathcal{I}_3$	4.7	3.2	1.3	0.2	Iris-setosa
$\mathcal{I}_4$	7.0	3.2	4.7	1.4	Iris-versicolor
$\mathcal{I}_5$	6.4	3.2	4.5	1.5	Iris-versicolor
$\mathcal{I}_6$	7.1	3.0	5.9	2.1	Iris-virginica
$\mathcal{I}_7$	6.3	2.9	5.6	1.8	Iris-virginica

Nota: Extraído del conjunto *Iris* (Dua y Graff, 2017).

f_1 = largo del sépalo, f_2 = ancho del sépalo, f_3 = largo del pétalo, f_4 = ancho del pétalo

El proceso de clasificación depende directamente del algoritmo de aprendizaje. Entre los más comunes se encuentran las *redes neuronales*, los *árboles de decisión* y las *máquinas de soporte vectorial*. Las redes neuronales pretender emular la estructura del cerebro de una persona, idea que se remonta a los años 30's. Dichas redes consisten en una estructura que permite recibir estímulos (entradas) que activan ciertas áreas de nuestra estructura cerebral (neuronas); la determinación de la activación de las neuronas es el punto central del trabajo de aprendizaje de la red neuronal, y por último — basado en los estímulos y las neuronas activas — se emite una salida. Sin embargo, la red neuronal que llamó la atención de la comunidad científica fue el *perceptrón*, propuesto por Frank Rosenblatt en 1958, y a partir de él, se han desarrollado una gran cantidad de extensiones. Una de las más populares es el *perceptrón multicapa con retro propagación* (Abiodun et al., 2018). Las investigaciones más recientes han transformado este concepto en lo que hoy se conoce como *aprendizaje profundo*. En esta tesis, se utilizará el clasificador conocido como *bosque aleatorio* (*random forest*), el cual se describe en la sección 2.1.2.

2.1.1 Evaluación de modelos predictivos

En cuanto a la evaluación de un modelo predictivo generado por un clasificador, se han investigado distintas maneras de llevar a cabo esta evaluación. Una de ellas es separar el conjunto de datos en dos partes: una para entrenamiento (*training set*) y otra para prueba (*test set*). En el conjunto de prueba, adicionalmente, se selecciona un porcentaje de instancias de cada una de las clases para un subconjunto de validación (*validation set*). El clasificador utiliza los datos del conjunto de entrenamiento para generar el modelo y, una vez que este se ha creado, se utiliza el conjunto de prueba para medir el grado con que el modelo predictivo puede aproximarse a la situación real.

Existen también otros métodos similares de validación. Uno de los más utilizados es la *validación cruzada* (*cross validation*). En ella se selecciona un numero

de grupos (*folds*) y el conjunto de datos se divide en k de estos grupos, donde cada uno de estos contiene un subconjunto de prueba distinto. De esta manera, para cada grupo se realiza un entrenamiento y prueba con todas las instancias. Al finalizar los k grupos, se utilizaron todas las instancias de prueba para la clasificación.

Independientemente del método seleccionado, las instancias pueden ser correctamente clasificadas o caer en varias categorías de errores. Las instancias que han sido clasificadas de acuerdo a la salida esperada conforman los *verdaderos positivos* (que denotaremos por VP), mientras que las instancias que se habían clasificado como pertenecientes a la clase y no lo son conforman los *falsos positivos* (que denotaremos por FP); de manera análoga, las instancias que pertenecen a la clase, pero faltaron de etiquetar conforman los *falsos negativos*, y las instancias que no pertenecen a la clase y efectivamente no fueron etiquetadas así, conforman los *verdaderos negativos* (que denotaremos como VN). Con estas distintas categorías, se pueden calcular métricas como la exactitud (Ec. 2.1), la exhaustividad (Ec. 2.2), la precisión (Ec. 2.3) y la medida F (Ec. 2.4) — que combina exhaustividad y precisión, es decir, la completez y la correctitud de la clasificación. A continuación se muestran las métricas antes mencionadas:

$$E = \frac{\mathrm{VP} + \mathrm{VN}}{(\mathrm{VP} + \mathrm{VN} + \mathrm{FP} + \mathrm{FN})}, \tag{2.1}$$

$$R = \frac{\mathrm{VP}}{(\mathrm{VP} + \mathrm{FN})}, \tag{2.2}$$

$$P = \frac{\mathrm{VP}}{(\mathrm{VP} + \mathrm{FP})}, \tag{2.3}$$

$$F_1 = \frac{2 * P * R}{P + R}. \tag{2.4}$$

La métrica a seleccionar debe ser la adecuada para el análisis del conjunto de

datos. Por ejemplo, en un conjunto de datos de 3000 instancias distribuidas en dos clases de $C_{\text{neg}} = 1500$ y $C_{\text{pos}} = 1500$, al analizar los datos de la matriz de confusión con las métricas mencionadas, encontraremos que cuando todas las instancias de C_{pos} se clasifiquen incorrectamente, el resultado al utilizar la ecuación 2.1 es de $E = 0.5$. Como el conjunto de datos está perfectamente balanceado, esta métrica puede ser interpretada de forma fácil, en el sentido de que si todas las instancias están clasificadas erróneamente su resultado será 0 y si todas las instancias están correctamente clasificadas será de 1, así como si la mitad de las instancias se clasifica correctamente su resultado será 0.5; sin embargo, esto no sucede así cuando el conjunto de datos presenta desbalance de clases.

En cambio, si las clases se encuentran distribuidas en $C_{\text{may}} = 2700$ y $C_{\text{min}} = 300$ instancias y, si todas las instancias de C_{min} se clasifican incorrectamente, el resultado al utilizar la ecuación 2.1 es de $E = 0.9$, lo cual puede interpretarse erróneamente como un buen resultado. En cambio la fórmula 2.3 da $P = 0$, así también la ecuación 2.2 da $R = 0$. Por consecuente, al utilizar la ecuación 2.4 da $F = 0$, es decir, el valor da $F = 0$, cuando no se logra clasificar correctamente ninguna de las instancias de la C_{min}. Por otro lado, su resultado será 1 si se clasifican correctamente la totalidad de las instancias, así que su resultado aumenta a medida que ambas clases se clasifican correctamente, pero con atención especial a C_{min}.

La tabla 2.2 muestra diferentes posibles resultados en una clasificación de tipo binaria y sus métricas. Si se considera un método de clasificación en donde simplemente se elige al azar la clase a la que pertenece, se puede aproximar a clasificar la mitad de los resultados correctamente siguiendo las leyes de probabilidad, de tal manera que cuando se menciona un bajo rendimiento en la clasificación en la medida F, tomando este argumento podemos decir que una *baja clasificación* en un conjunto balanceado es por debajo del $F = 0.5$ y, en un conjunto desbalanceado, está en razón del nivel de desbalance. Aun así, en este trabajo se considerará por debajo de 0.5.

Tabla 2.2:

Ejemplo de resultados de clasificación.

VP	VN	FP	FN	E	P	R	F
300	2700	0	0	1	1	1	1
250	2700	50	0	0.983	1	0.833	0.909
200	2700	100	0	0.967	1	0.667	0.8
150	2700	150	0	0.95	1	0.5	0.667
100	2700	200	0	0.933	1	0.333	0.5

2.1.2 Clasificadores basados en árboles de decisión

Un árbol de decisión es una técnica de aprendizaje automático que construye y utiliza un árbol basado en preguntas para realizar predicción. Para construir el árbol, se utiliza el conjunto de entrenamiento; en este caso, tanto el nodo raíz como los nodos internos corresponden a características del conjunto, y los nodos hojas corresponden a las clases. Cada rama, entonces, se compondría de instancias con valores particulares a las características y la clase con que están etiquetadas. Por ejemplo, en la figura 2.1, se puede apreciar un fragmento de un árbol construido a partir del conjunto de entrenamiento mostrado en la Tabla 2.3.

Tabla 2.3:

Fragmento de conjunto de entrenamiento: espera en restaurante.

Instancia	Alt	Bar	Vier.	Ham.	Gente	Precio	Lluvia	Res	Tipo	T. Esp.	**¿Esperar?**
$\mathcal{I}_1$	Sí	No	No	Sí	Poca	$$$	No	Sí	Francés	0-10	Sí
$\mathcal{I}_2$	Sí	No	No	Sí	Lleno	$	No	No	Thai	30-60	No
$\mathcal{I}_3$	No	Sí	No	No	Poca	$	Sí	No	Hamb.	0-10	Sí
$\mathcal{I}_4$	Sí	No	Sí	Sí	Lleno	$	Sí	No	Thai	10-30	Sí
$\mathcal{I}_5$	Sí	No	Sí	No	Lleno	$$$	No	Sí	Francés	¿60	No
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Vier= Viernes, Ham.= Hambriento, T. Esp.= Tiempo de Espera, Hamb.= Hamburguesa. Fuente: Russell y Norvig (2010)

Existen varias maneras de construir un árbol de decisión, pero por lo general se busca minimizar la cantidad de niveles, es decir, que el árbol no sea muy profundo.

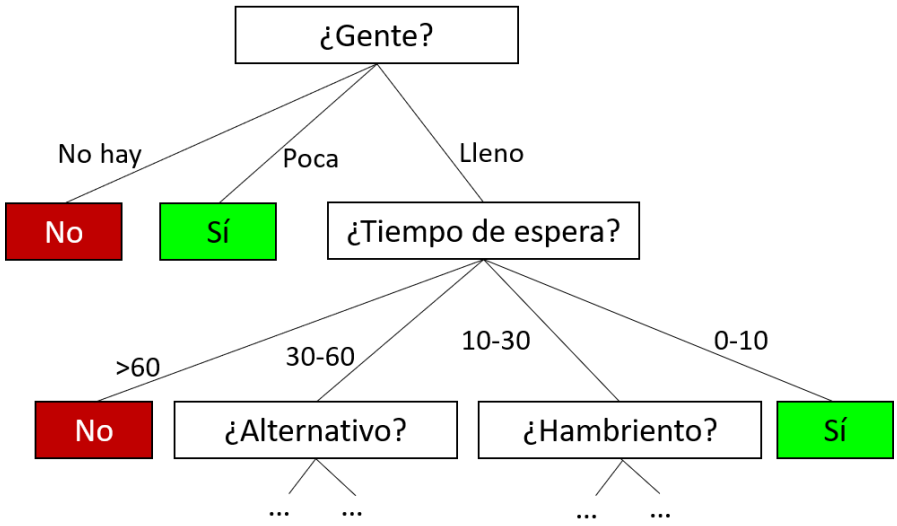


Figura 2.1: Fragmento de árbol de decisión: Tiempo de espera en restaurante. Fuente: Russell y Norvig (2010)

Una técnica consiste en utilizar el *índice Gini* (Ec. 2.5), el cual mide la impureza de cada característica (si una característica fuera "pura", ella sola podría dividir las instancias en clases). Por ejemplo, tomemos la Tabla 2.4 con un fragmento del conjunto de entrenamiento para enfermedad del corazón (Starmer, 2018). Para decidir qué característica irá en la raíz del árbol de decisión, se puede calcular el índice Gini; previamente, es necesario contabilizar las instancias de cada clase cuando la característica tiene valor *Verdadero* y cuando tiene valor *Falso* (ver figura 2.2), considerando que estamos tratando con características de tipo binario (más adelante veremos cómo calcular el índice Gini con otro tipo de características). El índice Gini se calcula de la siguiente manera:

$$\text{Gini} = 1 - \sum_{i=1}^{m} p_i^2, \tag{2.5}$$

donde p_i es la probabilidad de la clase i. Por tanto, para la hoja izquierda de la figura 2.2, la impureza según el índice Gini se calcula como

$$\begin{aligned} \text{Gini(hoja izq.)} &= 1 - (\text{probabilidad de } \textit{Sí})^2 - (\text{probabilidad de } \textit{No})^2 \\ &= 1 - \left(\tfrac{105}{105+39}\right)^2 - \left(\tfrac{39}{105+39}\right)^2 \\ &= 0.395, \end{aligned}$$

Tabla 2.4:

Ejemplo árbol de decisión: Enfermedad del corazón.

Dolor Pecho	Buena circulación	Arterias bloqueadas	Enfermedad corazón
No	No	No	No
Sí	Sí	Sí	Sí
Sí	Sí	No	No
Sí	No	No	Sí
⋮	⋮	⋮	⋮

Fuente: Starmer (2018)

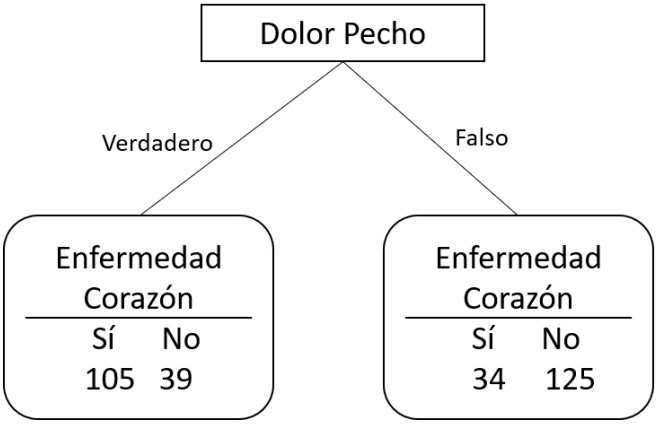


Figura 2.2: Ejemplo de construcción de árbol de decisión. Fuente: Starmer (2018)

y de manera análoga se calcula la impureza de la hoja derecha (0.336). Para calcular la impureza total Gini del nodo con esta característica, tomamos el promedio ponderado de ambas hojas. Este promedio ponderado se calcula tomando en cuenta el total de instancias que existe en cada hoja ($105 + 39 = 144$ para la hoja izquierda y $34 + 125 = 159$ para la hoja derecha):

$$\begin{aligned} \text{Gini(Dolor Pecho)} &= \left(\tfrac{144}{144+159}\right) 0.395 + \left(\tfrac{159}{144+159}\right) 0.336 \\ &= 0.364. \end{aligned}$$

Si consideramos que Gini(Buena Circulación) = 0.36 y
Gini(Arterias Bloqueadas) = 0.381, entonces la característica escogida para ser la

raíz del árbol sería *Buena Circulación*, dado que tiene la menor impureza. En general, para construir un árbol de decisión mediante el índice Gini, se siguen estos dos pasos:

1. Calcular el índice Gini para todas las características no utilizadas hasta el momento.
2. Para los nodos internos:
   a) Si no existe mejoría al utilizar nuevas características, dejar así tal cual (se convierte en hoja).
   b) De otra manera, utilizar la característica que presente el índice Gini más bajo (para el nodo raíz siempre se toma esta alternativa).

Ahora bien, las características no siempre son binarias. Cuando las características son numéricas, las instancias del conjunto de entrenamiento se ordenan de manera ascendente y se obtiene el promedio entre cada par de instancias para cada característica. El índice Gini se calcula considerando $¿f_i \leq a_j? =$ `Verdadero` y $¿f_i \leq a_j? =$ `Falso`, donde f_i representa la característica i y a_j representa el promedio j (por ejemplo: *¿Altura ≤ 3.5m?*). De modo similar, cuando las características son ordinales, el índice Gini se calcula considerando $¿f_i \leq o_j? =$ `Verdadero` y $¿f_i \leq o_j? =$ `Falso`, donde f_i representa la característica i y o_j representa el elemento j (por ejemplo: *¿Satisfecho ≤ 3?*). También cuando son categóricas se calcula como $¿f_i$ es $k_j? =$ `Verdadero` y $¿f_i$ es $k_j? =$ `Falso`, donde f_i representa la característica i y k_j representa la categoría j, y además deben considerarse combinaciones como $¿f_i$ es k_a o $k_b? =$ `Verdadero` (por ejemplo: *¿Tipo = rayado o moteado?*).

Los árboles de decisión son una técnica utilizada desde la década de los 70's. Los primeros modelos (AID, MAID, THAID y CHAID) se centraban en procesos estadísticos. CHAID es una extensión de AID y THAID ampliamente utilizada por su simplicidad y fácil interpretación de los resultados, ya que es posible transformar el árbol en una estructura de reglas de decisión. Por otra parte, el método propuesto por Quinlan (1987) denominado ID3 se caracteriza por utilizar como criterio de

división la métrica *ganancia de la información*. Se basa en el concepto de *entropía*, la cual mide el grado de impureza de la característica y se mide como

$$\mathcal{E} = -\sum_{i=1}^{m} p_i \log_2 p_i, \tag{2.6}$$

donde m es la cantidad de características. La ecuación 2.6 genera un resultado entre 0 y 1, en donde el 0 representa una característica con impureza y el 1 una característica pura. El árbol continúa creciendo hasta que las hojas separan completamente las clases o cuando la ganancia de la información ya no es mayor que cero. Años más adelante, el mismo autor (Quinlan, 2014) presenta una extensión a este proceso denominado C4.5, también conocido como J45, el cual asimismo utiliza la ganancia de información como criterio de división; sin embargo, el criterio para detener el crecimiento del árbol concluye cuando el número de instancias a dividir está por debajo de un umbral, y además se agrega un proceso de poda basado en errores durante el crecimiento del árbol.

Por otra parte, el algoritmo denominado *árbol de clasificación y regresión* (CART), desarrollado por Breiman et al. (1984), tiene como característica principal la construcción de árboles binarios, es decir que cada nodo interno tiene exactamente dos nodos de salida. En cuanto al criterio para realizar la división de las características se puede utilizar la ganancia de información y el índice Gini (Ec. 2.5), el cual se explicó anteriormente. Con este índice, se mide el grado o la probabilidad de que una variable particular se clasifique erróneamente cuando se elige al azar, de tal manera que si todos los elementos pertenecen a una sola clase, entonces puede llamarse puro. Su resultado se encuentra entre 0 y 1, donde 0 denota que todos los elementos pertenecen a una clase y 1 denota que los elementos se distribuyen aleatoriamente en varias clases; un 0.5 denota elementos igualmente distribuidos en algunas clases. El proceso de crecimiento CART incorpora la poda mediante el concepto de costo-complejidad, el cual corresponde a una relación entre el costo en el error de la clasificación ($\mathcal{E}/N$) y la complejidad N_T multiplicada por un valor α que es asignado por el usuario como se muestra en la ecuación 2.7:

$$C = (\mathcal{E}/N) + \alpha * N_T. \tag{2.7}$$

La figura 2.3 muestra un ejemplo de un árbol de decisión de tipo CART desarrollado en `Python 3` con la librería de `scikit learn`. El conjunto de datos utilizado para el ejemplo es del repositorio de datos UCI conocido como Haberman (Haberman, 1976), el cual corresponde a datos de la supervivencia de personas basada en tres características. En la Tabla 2.5 se muestra un fragmento de los datos; este conjunto de datos presenta desbalance de clases, por lo que se vuelve complicado identificar la clase de menor representación. Primero el nodo raíz se identifica por la división con menor índice Gini de las características, que corresponde a la variable de "número de ganglios axilares positivos detectados < 4.5", la cual divide las 214 instancias en dos hijos con 169 y 45 instancias. El proceso se repite en el nodo izquierdo y el derecho hasta que las clases se separen completamente; sin embargo, la profundidad del árbol puede provocar un sobre ajuste, por lo que al utilizar el modelo en el conjunto de prueba puede no generar los mejores resultados. En este ejemplo se obtiene una exactitud de 0.695 (donde la exactitud se mide como en la Ec. 2.1), y para evitar el sobre ajuste, se pueden establecer otros criterios de terminación, como determinar la profundidad máxima para el árbol. En la figura 2.4 se puede ver el mismo conjunto de datos con un criterio de profundidad máxima de 5 con una exactitud de 0.728. Si se cambia a una profundidad máxima de 2, la exactitud llega hasta 0.793.

2.1.3 Bosque aleatorio (*Random Forest*)

El algoritmo de bosque aleatorio (*random forest*), propuesto por Breiman (2001), se compone por un conjunto de árboles de decisión de tipo CART (de ahí parte su nombre "bosque"). Cada uno de los árboles es creado con un subconjunto de características seleccionadas de forma aleatoria, de tal forma que para determinar la decisión final en la clasificación de una instancia, se recorre en cada uno de los árboles creados y por votación se decide cuál es la clase a la que pertenece la instancia. Por basarse en un esquema de votación, al bosque aleatorio se le considera

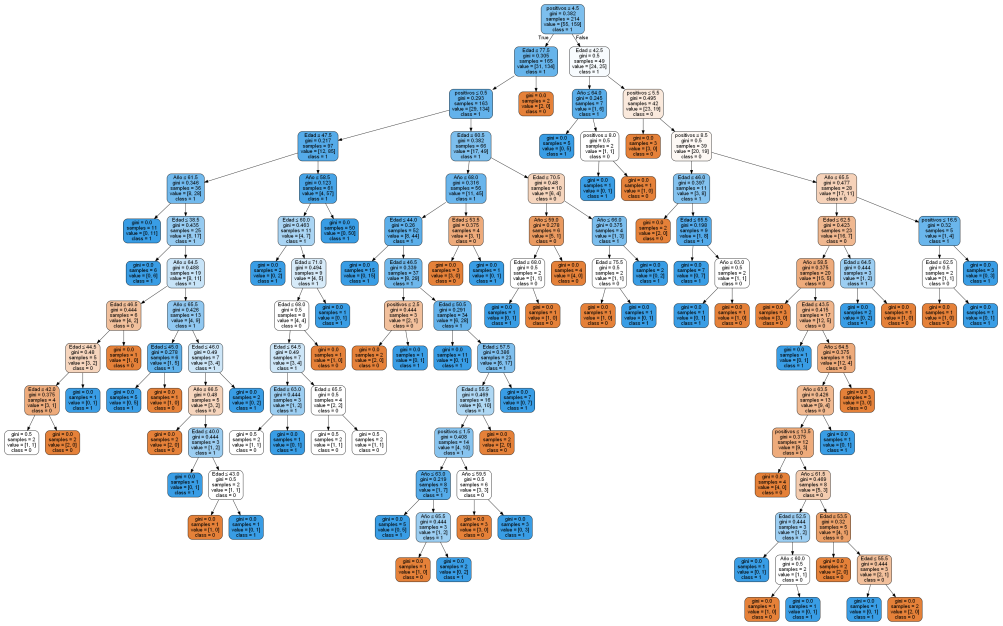


Figura 2.3: Árbol de decisión para conjunto de datos Haberman

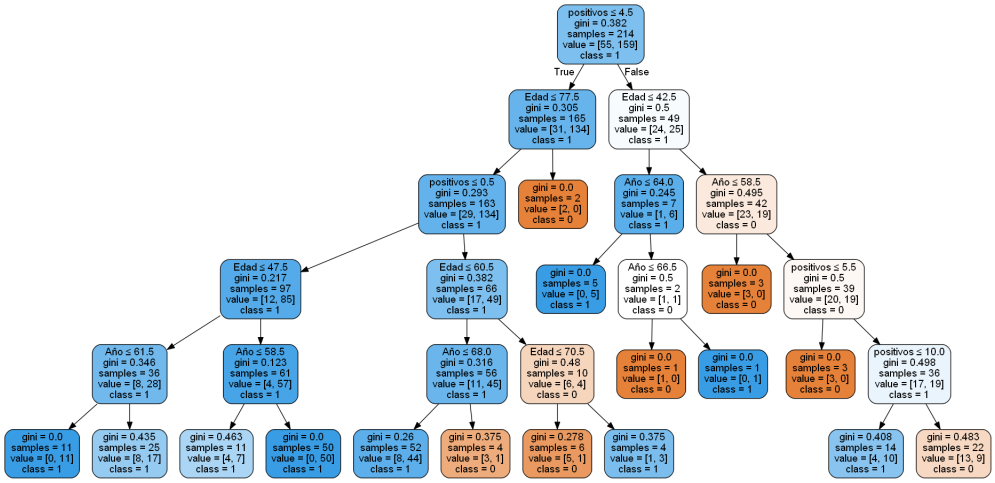


Figura 2.4: Árbol de decisión para conjunto de datos Haberman con profundidad máxima = 5

Tabla 2.5:

Fragmento de conjunto de entrenamiento: supervivencia de Haberman.

Edad	Año	Ganglios	**¿Sobrevive más de 5 años?**
38	59	2	1
39	63	4	1
49	62	1	1
53	60	2	1
45	63	0	0
52	61	0	1
⋮	⋮	⋮	⋮

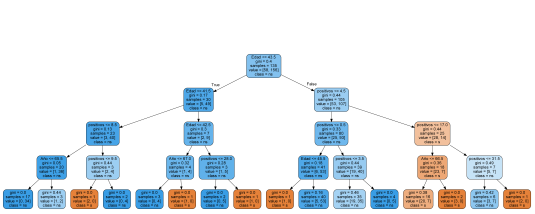
Nota: Edad = La edad del paciente al momento de la operación, Año = El año en el momento de la operación, Ganglios = Número de ganglios axilares positivos detectados, ¿Sobrevive más de 5 años? = 1 para si o 0 para no. Fuente: Haberman (1976)

un *clasificador de ensamble*.

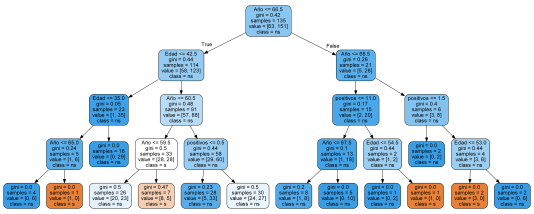
El proceso del bosque aleatorio tiene ventajas en comparación con los arboles de decisión. La primera de ellas es que ayuda a reducir el sobre ajuste, pues un árbol de decisión con gran profundidad puede sobre ajustarse a los datos de entrenamiento; sin embargo, los bosques aleatorios tienen múltiples árboles de decisión creados con subconjuntos de características. Otra ventaja es su capacidad para dar solución a conjuntos de datos extremadamente grandes, ya que al ser de ensamble hace posible tratar con este tipo de volúmenes.

Sus desventajas en comparación a los árboles de decisión principalmente son dos. La primera de ellas es que al utilizar una cierta cantidad de árboles de decisión para crear el modelo, su complejidad computacional se incrementa. La segunda desventaja se presenta en la interpretación de los datos, ya que un árbol de decisión es fácil de interpretar incluso para personas con poco conocimiento de las ciencias computacionales, no así los bosques aleatorios, pues comprenden un conjunto de árboles y el resultado de la clasificación depende del resultado de cada uno de estos árboles, lo que vuelve difícil su interpretación.

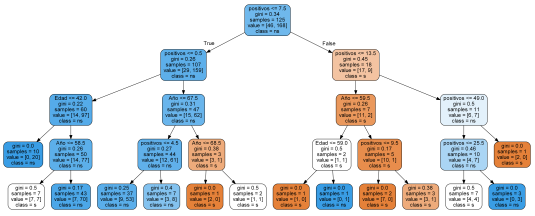
El algoritmo de bosque aleatorio abarca los siguientes pasos:



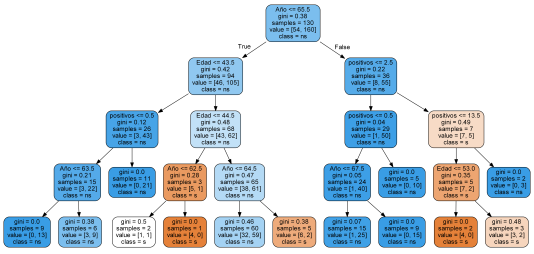
(a) Primer de 10 árbol del bosque aleatorio



(b) Segundo de 10 árbol del bosque aleatorio



(c) Tercer de 10 árbol del bosque aleatorio



(d) Cuarto de 10 árbol del bosque aleatorio

Figura 2.5: Bosque aleatorio para conjunto de datos Haberman con 10 árboles y profundidad máxima = 4

1. Del conjunto de datos se selecciona de forma aleatoria con reemplazo una cierta cantidad de instancias que conforman subconjuntos para la construcción de los árboles de decisión.

2. Para cada subconjunto, se selecciona de forma aleatoria también una cantidad de características a utilizar en el subconjunto.

3. Se construyen los árboles de decisión.

4. Se vota para seleccionar la clase a la que pertenece la instancia.

La figura 2.5 muestra cuatro árboles desarrollados en el proceso del bosque aleatorio. Como se puede apreciar en las figuras, cada uno de los árboles son diferentes: dos de ellos toman como nodo raíz la variable del año de atención, mientras que los otros seleccionan otras variables. La exactitud de este ejemplo fue de 0.75 utilizando 10 árboles de decisión y una profundidad máxima de 4.

2.1.4 El problema de clases desbalanceadas

Los algoritmos de aprendizaje automático tienden a clasificar incorrectamente cuando el conjunto de datos se encuentra desbalanceado. Por *desbalanceado*, nos referimos a que la cantidad de instancias de una clase es muchísimo mayor que la de otra(s) clase(s). Nos referiremos a la clase con mayor cantidad de instancias como *clase mayoritaria* y a la clase con menor número de instancias como *clase minoritaria*.

Cuando existen clases desbalanceadas, el modelo de predicción difícilmente puede aprender la clase minoritaria. Sin embargo, por cuestiones de probabilidad, al medir la exactitud (Ec. 2.1), esta resulta alta, aún cuando todas las instancias de la clase minoritaria sean clasificadas incorrectamente. Este efecto es conocido como *problema de clases desbalanceadas* (García y Herrera, 2009). Se presenta en conjuntos datos reales en áreas como la médica (Bach et al., 2017; Cohen et al., 2006), los negocios (Hyun-Jung et al., 2016), los desastres naturales (Kubat et al., 1998), la biología (Mani y Zhang, 2003) y la educación (Kotsiantis, 2009), en los que generalmente la clase de mayor interés es la de menor representación. Conforme aumenta el grado de desbalance (Ec. 2.8), es más difícil clasificar correctamente y la exactitud aparenta un resultado aceptable; por ejemplo, si la proporción cuenta con 1000 instancias de una clase y 30 instancias de otra, aún y cuando clasifique incorrectamente las 30 instancias de la clase minoritaria, la exactitud llega a un nivel de 97 %. El grado de desbalance IR se calcula como

$$\mathrm{IR} = \frac{|M|}{|m|}, \tag{2.8}$$

donde $|M|$ es la cantidad de instancias en la clase mayoritaria y $|m|$ es la cantidad de instancias en la clase minoritaria. El IR representa la cantidad promedio de instancias en la clase mayoritaria por cada instancia de la clase minoritaria.

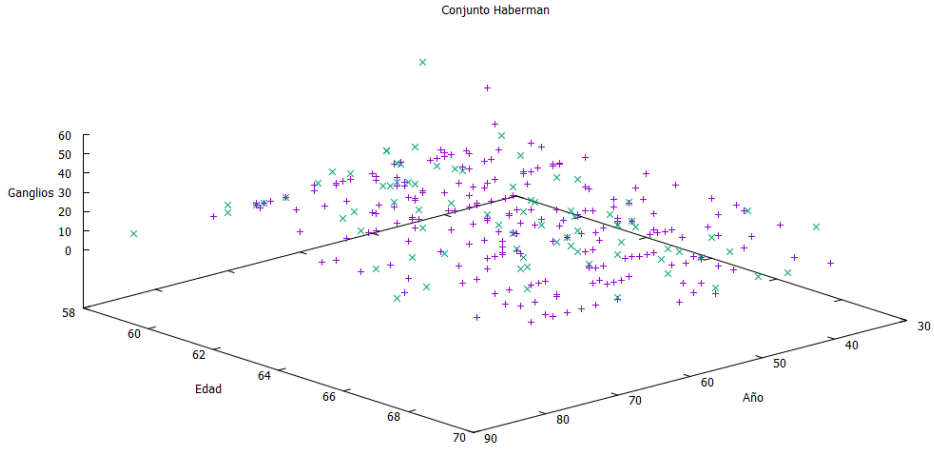
Se han desarrollado propuestas para mejorar la exactitud en la clase minoritaria de diferentes maneras: modificación de los algoritmos de aprendizaje, pre-

procesamiento de los datos, post-procesamiento de los datos y modelos que combinan las técnicas anteriores (Branco et al., 2016). El pre-procesamiento de los datos se ha estudiado en dos líneas principales: el cambio en la distribución de los datos y la asignación de peso al espacio de los datos. La primera de estas líneas cuenta con mucho mayor interés por los investigadores, y a su vez se divide en dos estudios: el re-muestreo estratificado y la generación de instancias sintéticas. Uno de los métodos que más ha llamado la atención de los investigadores es SMOTE (*Synthetic Minority Over-sampling Technique*), el cual consiste en generar instancias sintéticas utilizando la técnica vecinos más cercanos e interpolación (Chawla et al., 2002). Al 2018 existen más de 85 variantes o extensiones de este algoritmo (Fernández et al., 2018).

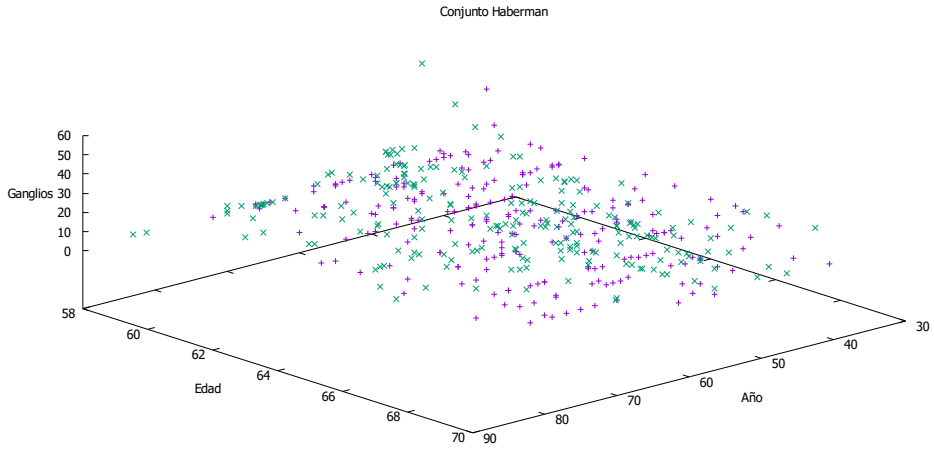
En el algoritmo original de SMOTE propuesto por Chawla et al. (2002) tiene como propósito incrementar la cantidad de instancias de la clase minoritaria en un porcentaje determinado por el usuario. Para este proceso utiliza la técnica de los vecinos más cercanos (KNN) — por defecto se recomienda cinco vecinos. Aun así es un parámetro que se puede ajustar. Una vez detectados los vecinos más cercanos, se utiliza un procedimiento de interpolación para crear nuevas instancias entre las existentes y sus vecinos. Esto aumenta las regiones poco pobladas con las nuevas instancias. En la figura 2.6 se puede apreciar un ejemplo en un conjunto de datos de tres dimensiones.

2.2 Selección y construcción de características

Los algoritmos de aprendizaje automático desarrollan un modelo para predecir un estado desconocido basado en un conjunto de características y ejemplos; sin embargo, en los problemas del mundo real es posible enfrentarse con situaciones en donde la cantidad de atributos es muy extensa, tanto que se vuelve complejo



(a) Haberman original



(b) Haberman con sobremuestreo

Figura 2.6: Distribución de datos en conjunto de datos supervivencia de Haberman

para un clasificador común el obtener resultados en un tiempo razonable. Aún con los grandes avances tecnologías continúa siendo un tema para investigar. Aún y cuando incrementar el número de características parece una buena idea con el fin de mejorar la clasificación, esto puede tener un costo alto, ya que se aumenta la dimensionalidad del conjunto de datos, y también existe la posibilidad de agregar características irrelevantes o redundantes que pueden generar ruido en la clasificación (Bolón-Canedo et al., 2015).

El problema de la alta dimensionalidad es estudiado por los investigadores en dos formas: la *selección* y *construcción* de características (*feature selection* y *feature construction*, respectivamente en inglés). El método de selección de características consiste en la determinación de un subconjunto de características relevantes, es decir, eliminar las características redundantes o irrelevantes para aumentar la calidad de la clasificación. Por otra parte, el método de construcción de características consiste en construir nuevas características a partir de las características originales.

El método de selección de características se ha explorado en tres formas: métodos de filtro, métodos de envoltura y métodos embebidos (Bolón-Canedo et al., 2015). El primero de ellos tiene bases en la estadística y se evalúa la relevancia de cada característica con la clase. La mayoría de ellos utilizan un sistema de puntuación para la característica, es decir, para cada una de las características del conjunto de datos se establece un valor numérico, y las puntuaciones diferentes de 0 pueden ser consideradas para construir el subconjunto. Un ejemplo de un método de filtro es la correlación Pearson. Por otro lado, los métodos de envoltura generan un conjunto de datos con un subconjunto particular de características, y este conjunto de datos se da como entrada a un clasificador; alguna métrica de calidad del clasificador se utiliza para mejorar el subconjunto de características actual hasta obtener un buen desempeño. Un ejemplo de método de envoltura sería un algoritmo genético, como el que se presenta en este trabajo, o programación genética. El método embebido es la incorporación de la selección de características dentro del proceso de clasificación. Un ejemplo de este tipo de métodos es un árbol de decisión (nótese que un árbol de

decisión también podría utilizarse en combinación con un algoritmo genético para producir un método de envoltura).

Se han identificado algunas ventajas de los métodos de envoltura sobre los métodos de filtro. Una de ellas es que se toma en cuenta la interacción entre las características, esto es, cuando dos características están relacionadas de alguna forma, al retirar una de ellas puede incrementar o disminuir su relevancia con la clase. Otra ventaja es que su resultado está basado en el desempeño del clasificador. Su desventaja principal es que requiere de un mayor tiempo computacional.

Los algoritmos evolutivos se han utilizado por los investigadores para atender el problema de la alta dimensionalidad. Lo anterior debido a que, al incrementarse la cantidad de características, el costo computacional se eleva. Por lo tanto, gran parte de los investigadores optan por métodos metaheurísticos como algoritmos genéticos (GA's), programación genética(GP), optimización de partículas (PSO) y optimización por colonia de hormigas (ACO), por mencionar algunos.

2.3 Algoritmos genéticos

La técnica de algoritmos genéticos corresponde a una serie de algoritmos estocásticos para hacer optimización (de ahí que pertenezca a la familia de los *metaheurísticos*), y su particularidad es que están basados en la teoría de la evolución de las especies.

En un algoritmo genético, el espacio de búsqueda se ve como un universo de individuos, donde cada individuo está representado por un vector de características llamado *cromosoma* y a cada característica de este cromosoma se le conoce como *gen*. Los cromosomas se representan típicamente como cadenas binarias (ej. '0110'). Sin embargo, existen otras formas de representación para los cromosomas, dependiendo del problema a tratar y sus restricciones. Por ejemplo, para el problema del viajero (Neapolitan y Naimipour, 1998), los cromosomas pueden consistir en cadenas o listas

de números enteros que representan permutaciones de ciudades.

Cada cromosoma está ligado a un *valor de aptitud*, el cual proviene de una *función de aptitud*, la cual varía dependiendo del problema que se esté abordando. Es decir, es una cuestión de diseño.

2.3.1 Algoritmo genético canónico

El algoritmo genético canónico (que se puede encontrar en referencias como el libro de Sait y Youssef (1999)) está representado en el Algoritmo 1. Se parte de una *población inicial* (normalmente escogida al azar), la cual representa una muestra del espacio de búsqueda; el *tamaño* de esta población también es un parámetro a considerar. A esta población se le aplica una *evaluación* para conocer los individuos más aptos y al mismo tiempo realizar la selección de algunos de los individuos para conformar parejas. A estas parejas se les aplica una operación de *cruza* para generar *cromosomas hijos*. A cada cromosoma hijo se le aplica el operador de *mutación*. Al paso de una generación le llamamos *iteración* y al paso de x generaciones le llamanos *corrida*. Para cada operador, existen diferentes opciones de implementación. Por ejemplo, para la selección las opciones más comunes corresponden a la *rueda de ruleta* (algoritmo 2) y la *selección por torneo*. Mientras que la primera considera la selección aleatoria proporcional, la segunda considera tomar un par de individuos al azar y escoger como miembro de una pareja al individuo con la mayor aptitud de estos dos. En cuanto a la cruza, un método convencional es el *punto de cruce*, en el cual se escoge un punto p entre dos genes y se generan dos hijos por pareja: ambos hijos tienen una parte diferente de cada cromosoma progenitor. Para la mutación, normalmente se escoge una probabilidad de mutación p_m, y de acuerdo con esta probabilidad se decide, para cada gen, si será o no mutado. La mutación consiste en intercambiar el gen por su complemento: si era 0 se cambia por 1 y viceversa. El algoritmo genético canónico también es conocido como *algoritmo genético simple*.

Algoritmo 1 Algoritmo genético canónico

```
1: function AG
2:     for n iteraciones do
3:         ℙ=Generar-Población-Inicial()  ▷ Escoger individuos del universo
   al azar
4:         while Criterio de terminación do
5:             𝔸=Evaluar(ℙ)                ▷ Calcular aptitudes de la población
6:             𝕄=Seleccionar(ℙ,𝔸)          ▷ Seleccionar parejas de individuos
7:                                          ▷ de acuerdo a aptitud
8:             ℍ=Cruzar(𝕄)                 ▷ Generar hijos de parejas seleccionadas
9:             ℍ′=Mutar(ℍ)                 ▷ Mutar (modificar) hijos de parejas
10:            ℙ = ℍ′                      ▷ Reemplazar vieja población por la nueva
11:        end while
12:    end for
13:    return individuo más apto
14: end function
```

Algoritmo 2 Selección por rueda de ruleta

Descripción: Recibe la población $\mathbb{P}$ y la lista de aptitudes Aptitudes. Calcula la probabilidad de selección de cada individuo de la población (de acuerdo a su aptitud), simula un giro de ruleta y devuelve al individuo seleccionado de acuerdo con este giro de ruleta.

1: **function** Ruleta($\mathbb{P}$, Aptitudes)
2: **for** $i \leftarrow 0, |\mathbb{P}|$ **do**
3: $p_{\text{sel}}(i) \leftarrow \frac{\text{Aptitudes}_i}{\sum \text{Aptitudes}}$
4: **end for**
5: $r \leftarrow$ obtener-aleatorio$(0, 1)$
6: **while** $r < p$ **do**
7: $p = p + p_{\text{sel}}(i)$
8: $i = i + 1$
9: **end while**
10: **return** $\mathbb{P}_i$
11: **end function**

Consideramos los términos *población*, *generación* e *iteración* como intercambiables.

2.4 Resumen

En este capítulo, se trataron tres temas principales: (1) aprendizaje automático, (2) selección y construcción de características y (3) algoritmos genéticos. Del tema de aprendizaje automático, se vieron conceptos básicos, tales como *aprendizaje supervisado*, *clasificación binaria*, *conjunto de entrenamiento*, *conjunto de prueba* y medidas de evaluación para un modelo predictivo, tales como la medida F. Asimismo, se ahondó en la técnica de *bosque aleatorio*, que consiste en múltiples árboles de decisión con subconjuntos de características que realizan un proceso de votación para escoger la clase correspondiente a cada instancia del conjunto de datos. Dentro

del tema de aprendizaje automático también se abordó el problema de las clases desbalanceadas, que consiste en tener una cantidad muy pequeña de instancias de una clase (a comparación de la otra clase en un esquema binario), donde esta clase es la clase de interés.

En cuanto a la selección y construcción de características, estas técnicas se utilizan para reducir las dimensiones del conjunto de datos. Por una parte, la selección busca eliminar características redundantes o irrelevantes. Por otra parte, la construcción busca explorar características nuevas a través de la combinación de características existentes. Los tres principales paradigmas para seleccionar y construir características son *filtro*, *envoltura* y *embebido*. Mientras que el filtro trata de descartar características a través de relaciones matemáticas (como correlación), el enfoque de envoltura se vale de los resultados de un clasificador para establecer el conjunto óptimo de características. Por último, el enfoque embebido utiliza al mismo clasificador para escoger características.

En cuanto a algoritmos genéticos, es una técnica metaheurística basada en la evolución de las especies y la selección natural. Se basa en generar una codificación para un problema de optimización, donde cada fragmento de código recibe el nombre de *cromosoma*, y en encontrar al mejor cromosoma dentro de un espacio de búsqueda. Para encontrar este cromosoma, se toma una población (muestra) inicial de cromosomas, los cuales son evaluados de acuerdo a una *función de aptitud* y se seleccionan parejas de acuerdo con esta aptitud. Dichas parejas se cruzan para generar hijos, y estos hijos sufren mutaciones de acuerdo a una pequeña probabilidad. La población inicial es reemplazada por sus hijos y el procedimiento se repite por varias iteraciones hasta cumplir con un criterio de terminación. Se toma el mejor cromosoma encontrado (que se espera sea el mejor del espacio de búsqueda).

Capítulo 3

Estado del Arte

Las técnicas de selección y construcción de características han llamado la atención de muchos investigadores en la actualidad. Con los avances tecnológicos hoy en día es posible recopilar una gran cantidad de variables y ejemplos; sin embargo, a medida que el conjunto de datos crece también lo hace su complejidad. Un número mayor de variables implica un aumento en la probabilidad de utilizar variables redundantes o irrelevantes, provocando que la clasificación se vea limitada debido a estos atributos. En este sentido, los investigadores en este tema han propuesto técnicas que alteran el espacio de búsqueda para resolver dos posibles problemas: el primero es tratar con conjuntos extremadamente grandes (reducción de la dimensionalidad) y el segundo es aumentar la exactitud de la clasificación. En ambos casos se busca un subconjunto que reemplace el conjunto original con la suficiente información para mantener o aumentar los resultados en la clasificación (Chandrashekar y Sahin, 2014).

La *selección de características* se encarga específicamente de proporcionar un subconjunto de características, mientras que la *construcción de características* busca la combinación de estas mediante operaciones aritméticas. Este tipo de problemas se clasifican como *problemas de optimización combinatoria*, ya que a medida que aumenta el número de características se incrementa grandemente el número de posibles respuestas. Debido a lo anterior, los algoritmos del cómputo evolutivo son una

excelente opción para este tipo de problemas, ya que es posible generar una respuesta aceptable en espacios de búsqueda extremadamente amplios. Por tal motivo, la revisión de la literatura pertinente se concentra en este enfoque. En el artículo presentado por Xue et al. (2016) se hace una distinción del tema de selección de características de acuerdo a las siguientes categorías:

1. Por su evaluación
    - *a*) Filtros
    - *b*) Envoltura
    - *c*) Combinación de ambos
2. Por paradigma en el cómputo evolutivo
    - *a*) Algoritmos evolutivos
        1) Algoritmos genéticos
        2) Programación genética
    - *b*) Inteligencia de enjambre
        1) Optimización por inteligencia de partículas
        2) Optimización por colonia de hormigas
    - *c*) Otros
3. Por número de objetivos
    - *a*) Un objetivo
    - *b*) Multi-objetivo

Además de las categorías presentadas, agregamos la técnica de reducción de la dimensionalidad, que corresponde específicamente a (a) selección de características, (b) construcción de características y (c) la combinación de ambas.

Por otra parte, en el ambiente educativo se ha incrementado el interés de la aplicación de técnicas de aprendizaje automático. Dentro de las tareas más comunes se encuentra la predicción del desempeño y predicción del abandono escolar; en ambos casos se toma un conjunto de características propia del proceso educativo y se busca clasificar en base a estas características el comportamiento de los estudiantes. El trabajo presentado tiene como caso de estudio una escuela de educación media superior, por lo cual es de gran interés comprender los avances en este tema.

Considerando lo anterior, la literatura a presentar en este capítulo se divide en dos partes: la primera son artículos relacionados con cómputo evolutivo, desarrollados por un método de evaluación de tipo envoltura o combinado, con un objetivo y los tres tipos de reducción de la dimensionalidad; la segunda parte corresponde a las investigaciones desarrolladas en el entorno educativo correspondientes a la clasificación del desempeño o predicción del abandono escolar.

3.1 Selección y construcción de características

La selección de características es una tarea importante en el campo de la ciencia de datos y se ha estudiado desde diferentes metodologías y paradigmas de trabajo (Wang et al., 2016), principalmente con técnicas estadísticas en sus orígenes; sin embargo, los métodos evolutivos han alcanzado gran popularidad en el desarrollo de técnicas de selección de características, donde el objetivo es mantener o mejorar el rendimiento de los resultados de un clasificador utilizando un subconjunto de datos. Aunado a este objetivo normalmente sigue el minimizar la cantidad de características y reducir el tiempo computacional. Particularmente en el área de cómputo evolutivo, se han presentado propuestas que utilizan como base los algoritmos genéticos con algunos rasgos que los diferencian de los demás.

Los primeros artículos de cómputo evolutivo en el campo de selección de ca-

racterísticas se presentan a finales de la década de los 80. Al representar el estado de las características, convencionalmente se ha empleado el 1 para características a utilizar (activas) y el 0 para características que no se utilizan (inactivas); esto representa una solución mucho más efectiva en tiempo computacional que la búsqueda exhaustiva. En recientes años, se continúan explorando diferentes formas de mejorar este proceso. En el trabajo de Emmanouilidis et al. (2000) se propone un algoritmo genético para hacer selección de características, donde se combinan dos objetivos: disminuir la cantidad de características y aumentar la calidad del clasificador. Se caracteriza su trabajo por incorporar dentro del proceso la mejora con Pareto; utilizan el método del torneo con selección aleatoria y el método propuesto lo prueban en dos conjuntos de datos de clasificación binaria, uno de 34 y otro de 60 características; los resultados del método propuesto (MOEA) se comparan contra el método *Sequential*.

La selección de características tiene un papel importante, ya que su aplicación puede generalizarse en varios contextos; en el área de finanzas se presenta un problema denominado *predicción de dificultades financieras*, en el cual se han desarrollado estudios que involucran selección de características. El modelo propuesto por Lin et al. (2014) denominado HARC involucra la aplicación de selección de características basada en un algoritmo genético, aplicada para un modelo de predicción de problemas financieros. En él crea 60 individuos de manera aleatoria, donde después — por método de torneo — selecciona a los individuos que continuarán el proceso evolutivo; sin embargo, dentro de él se establece una probabilidad de cruza de 70 %, por lo que no todos los individuos se cruzan y la probabilidad de mutar es muy baja, tan solo de 0.5 %, por lo que muy pocos individuos sufren este proceso, es decir, en este caso se le da más prioridad a la selección y la cruza que a la mutación. En su criterio de terminación establece un máximo de 120 generaciones.

Por otra parte, en el trabajo de Soufan et al. (2015) se desarrolló una propuesta de selección de características basada en un algoritmo genético. La característica principal de este trabajo es la presentación de una interfaz web, en la cual permite establecer los valores de probabilidad de cruza, mutación y la cantidad de generacio-

nes del algoritmo genético. Asimismo, para el proceso de clasificador se presentan las opciones de *Naïve Bayes* (NB), vecinos más cercanos (KNN) y una combinación de ambos (NBKNN), además de seleccionar entre cuatro diferentes métricas de rendimiento. Otro punto particular de esta investigación es que incorpora un mecanismo de selección de características de tipo filtro cuando el conjunto de datos es muy grande, previo a la selección de características por el método de envoltura.

Además de algoritmos genéticos se han desarrollado técnicas con enfoque en cómputo evolutivo para la selección de características como la planteada por Xue et al. (2013), quienes presentan dos algoritmos utilizando la optimización por enjambre de partículas (PSO) multi-objetivo para realizar selección de características; sus bases las fundamentan en las ideas propuestas por Deb et al. (2002) y Li (2003), formando dos propuestas denominadas NSPSOFS y CMDPSOFS. Sus objetivos son minimizar el error en la clasificación y utilizar una menor cantidad de características. Los resultados muestran una mejora considerable en cuanto a la disminución del error en comparación a utilizar todas las características; el método LMS y GSBS son utilizados como clasificadores. El mismo año se presenta otro trabajo de optimización por enjambre de partículas con la variación de realizar la construcción de características en el trabajo de Xue et al. (2013) llamado PSOFC, en donde cada elemento de la partícula es un número real entre 0 y 1; cuando es 1 es utilizado para la construcción y si es 0 no se utiliza. La combinación se realiza al inicio con el operador de "+" y al finalizar el proceso genera como respuesta un atributo de alto nivel, de tal manera que la dimensionalidad del conjunto de datos aumenta en uno, pero los resultados presentados no son prometedores.

En el método propuesto por Dai et al. (2014) mejora el proceso obteniendo como resultado dos nuevos algoritmos: PSOFCPair y PSOFCArray, incluyendo la posibilidad de seleccionar operadores distintos, mismos que son utilizados en la propuesta presentada por Mahanipour y Nezamabadi-pour (2017) el cual incluye como paso previo la selección de características hacia adelante (FFS) antes de la construcción de características con el fin de determinar las instancias más relevantes y en

ellas aplicar la construcción de características con PSOFCPair.

Por otra parte, Zhang et al. (2015) presentan un diseño de PSO para hacer selección de características. Al inicio se crea de manera aleatoria el enjambre de partículas, las cuales están compuestas por elementos entre 0 o 1 que corresponden a la probabilidad de utilizar la posición de la característica o no. Para este proceso no se requiere el concepto de velocidad; en cada una de las iteraciones la partícula actualiza sus valores de mejor individual y global. El proceso es evaluado utilizando la técnica de vecinos más cercanos (KNN) — específicamente un vecino más cercano (1NN). La métrica utilizada para evaluar las partículas está dada por la relación entre los clasificados correctamente y el número de características seleccionadas, de tal manera que encuentra un equilibrio entre estas dos.

Por otra parte, la programación genética ha tomado gran popularidad, sobre todo cuando se trata de la construcción de características. Lo anterior, debido a que su estructura evolutiva permite explorar una gran cantidad de posibilidades. El árbol se construye con operadores y los valores de las características del conjunto de datos, en donde las características son las hojas finales del árbol y las operaciones la unión entre ellos; el proceso evolutivo intercambia fragmentos de los árboles formados, los cuales son evaluados con una regla de aprendizaje para lograr un objetivo definido (convergencia) o llegar a un máximo de iteraciones. Además, el mismo se convierte en el clasificador, lo que le permite ahorrar tiempo computacional. En este sentido, Krawiec (2002) propone un método elitista, en donde además de generar las características crea un repositorio de los mejores para conservar durante el proceso evolutivo. De la misma manera, Ahmed et al. (2014) proponen un método basado en programación genética para construir múltiples características nuevas, con la característica de utilizar la misma programación genética para evaluar el nuevo conjunto de datos. Para realizar la evaluación utiliza la F de Fisher y el valor P. La idea principal de su propuesta es tomar los nodos internos de un árbol construido como la nueva característica.

Los objetivos en la literatura son reducir la dimensionalidad y mantener o mejorar el rendimiento de la clasificación con la construcción de nuevas características; sin embargo, algunas de ellas se enfocan en conjuntos de datos extremadamente grandes. Tal es el caso de la propuesta presentada por Tran et al. (2016), donde los conjuntos de datos utilizados en sus experimentos contienen entre $2,000$ y $15,154$ características. Utilizando programación genética para la construcción de características y métodos de filtro y envoltura para la evaluación, logran disminuir en un gran porcentaje el conjunto de datos y obtener resultados superiores a comparación de utilizar todas las características en tres clasificadores: vecinos más cercanos, *Naïve Bayes* y árbol de decisión. El siguiente año, Tran et al. (2017) presentan una nueva propuesta en donde utilizan la técnica de agrupamiento de características: se forman grupos de atributos redundantes. Para determinar esto utilizan la correlación, en donde dos atributos completamente correlacionados son redundantes; para determinar si se agruparán o no se determina un umbral. Una vez agrupadas las características realizan la construcción de características utilizando la programación genética. En relación con sus investigaciones anteriores, dos años después Tran et al. (2019) presentan un nuevo modelo. El objetivo de su investigación es la construcción de varias características de alto nivel (construcción de características) en un mismo proceso. Se basan en la teoría de que a mayor número de clases es necesaria una mayor cantidad de características, por lo que utilizan la fórmula $m = r * c$ donde r es un radio definido (2 o 3) y c es el número de clases. Otra mejora a sus propuestas anteriores es la función objetivo, fitness $= \alpha * \text{BalAcurracy} + (1 - \alpha) * \text{Distancia}$. Mientras BalAccurracy se enfoca en la relación entre los verdaderos positivos y los clasificados correctamente, Distancia se utiliza para maximizar la distancia entre clases y minimizar la distancia entre instancias de cada clase. Sus resultados muestran ser superiores a los métodos anteriores utilizando vecinos más cercanos (KNN), Naïve Bayes (NB) y árbol de decisión (DT).

En el trabajo de Tsoulos et al. (2019) se realiza la construcción de características mediante un algoritmo genético; su rasgo principal es el desarrollo de un software

con la capacidad de ejecutar en paralelo en varias máquinas, lo que ayuda a reducir el efecto provocado por el problema del espacio de búsqueda que se deriva en tiempo computacional. Como clasificador utiliza una red neuronal de tipo RBF (*radial basis function*). Cada una de las máquinas reporta sus resultados a un servidor. El algoritmo genético diseñado requiere de parámetros iniciales, y de entre ellos se destacan algunos particulares de su proceso, como $f_l = \infty$, que es un parámetro que aplica en su criterio de terminación, número de generaciones que deben pasar para aplicar un método de búsqueda local y el tamaño de la población que participa en la búsqueda local. En sus resultados concluyen que su método hace más eficiente la clasificación usando 4 y hasta 8 características construidas.

Una combinación de programación genética y algoritmos genéticos es presentada por Smith y Bull (2005). Al inicio crea una población de 101 individuos de manera aleatoria, en donde cada gen del cromosoma corresponde a un árbol, de tal manera que el árbol puede estar compuesto de un simple elemento (característica original) o de una combinación de características (construcción). Las operaciones en el árbol son $*$, $/$, $+$ y $\%$, y el clasificador utilizado es basado en un árbol CJ4.5. En sus resultados concluye que no se requieren más de 50 generaciones para encontrar un resultado aceptable.

Por otra parte, el trabajo de Nguyen et al. (2017) propone un método híbrido entre el algoritmo genético y la programación genética para realizar los procesos de selección y construcción de características de manera simultánea, de tal forma, que el clasificador evalúa el subconjunto de datos incluyendo las características originales seleccionadas y las características construidas por la programación genética. En este proceso cada individuo de la población consta de un conjunto de bits y un árbol. El método de selección utilizado es el de torneo y la operación de cruza es modificada. Para la parte del cromosoma binario se realiza una cruza de un punto; de igual manera, en el árbol se selecciona un punto de cruza y en la mutación se selecciona un bit al azar y se cambia de 0 a 1 o de 1 a 0 según sea el caso; sin embargo, en el árbol se selecciona un punto en particular y puede cambiar agregando elementos a

este.

La diferencia principal de nuestro método con el propuesto por Nguyen et al. (2017) se puede apreciar principalmente en:

1. La selección de características permite el uso de características simples; sin embargo, estas características activas no pueden participar en la construcción de características, es decir, no se permite repetir el uso de características.
2. El proceso de cruza se realiza por bloques.
3. La mutación puede alterar por completo la estructura del cromosoma.

Cada método presentado tiene sus aspectos particulares. Aún así, en la Tabla 3.1 se muestran algunas características propias de la propuesta presentada en esta investigación en comparación con la literatura.

3.2 Predicción del rendimiento estudiantil

En el ámbito educativo, el uso de minería de datos se ha investigado desde diferentes perspectivas. En el trabajo de Romero y Ventura (2010) se identifican diferentes grupos involucrados en el ámbito educativo, como estudiantes, maestros, cursos, organización educativa y procesos administrativos. Además, existen diferentes tipos de ambientes de aprendizaje como el tradicional y basados en Web, entre otros. De igual manera, Bakhshinategh et al. (2018) separan en enfoques las investigaciones desarrolladas en el ámbito educativo, resaltando entre ellas el modelado de estudiantes, la cual se centra en dos tareas: predicción y agrupamiento. En esta sección se presenta una selección de artículos cuyo objetivo se basa en la predicción en el ámbito educativo.

En Kotsiantis et al. (2004) utilizan información de un curso a distancia y el objetivo es identificar el abandono escolar. Ellos utilizan información general como

Tabla 3.1:

Tabla de comparación de propuestas

Características	Combina selección y construcción de características	Divide el conjunto de características en grupos o bloques	Cromosoma con información de la selección y la construcción	Objetivo de aumentar el rendimiento	Tres niveles de mutación	Cada característica solo se puede utilizar una vez en selección y construcción	Cantidad de características construidas adaptativa
FSCGA	✓	✓	✓	✓	✓	✓	✓
Nguyen et al. (2017)	✓	X	X	✓	X	X	X
Tsoulos et al. (2019)	X	X	X	✓	X	X	X
Tran et al. (2019)	X	X	X	✓	X	X	✓
Tran et al. (2017)	X	✓	X	✓	X	X	X
Smith y Bull (2005)	✓	X	X	✓	X	X	✓
Ahmed et al. (2014)	X	X	X	✓	X	✓	✓
Krawiec (2002)	X	X	X	✓	X	X	X
Zhang et al. (2015)	X	X	X	✓	X	X	X
Mahanipour y Nezamabadi-pour (2017)	X	X	X	✓	X	✓	X
Dai et al. (2014)	X	X	X	✓	X	✓	X
Xue et al. (2013)	X	X	X	✓	X	✓	X
Xue et al. (2013)	X	X	X	✓	X	X	X
Soufan et al. (2015)	X	X	X	✓	X	X	X
Lin et al. (2014)	X	X	X	✓	X	X	X
Emmanouilidis et al. (2000)	X	X	X	✓	X	X	X

el sexo, edad, estado civil, entre otros y clasifica en dos formas: buenos y fallos. El modelo de predicción se genera utilizando árboles de decisión, redes neuronales, máquinas de soporte vectorial y clasificadores bayesianos. La predicción la realiza en dos etapas al inicio del curso con un 62 % en la precisión y a mitad del curso con un 83 %. De igual manera, en Ramaswami y Bhaskaran (2009) se anexa al estudio una exploración de las técnicas de selección de atributos. Ellos utilizan información obtenida a través de encuestas de 1969 estudiantes de nivel superior a secundaria de distintas escuelas de la India. Su propuesta desarrolla un algoritmo voraz para explorar el uso de los características seleccionadas por cada una de las técnicas de la selección de características y las compara utilizando el área bajo la curva (ROC) y la medida F. Concluyen que el omitir algunas de las características del conjunto de datos ayuda a mejorar la precisión de la clasificación.

En el mismo año, en el trabajo presentado por Wook et al. (2009) utilizan información personal, demográfica y antecedentes académicos de estudiantes del departamento de ciencias computacionales de la facultad de ciencia y defensa tecnología en la Universidad de Malasia, para predecir el rendimiento académico de los estudiantes separándolos en tres clases; bajo riesgo, medio riesgo y alto riesgo, utiliza un modelo predictivo basado en redes neuronales artificiales, agrupamiento y árbol de decisiones para generar su clasificación. Por otra parte en Abuteir y El-Halees (2012) se desarrolla una configuración para los graduados del colegio de ciencias y tecnología de Khanyounis, con un total de 18 atributos y 3360 graduados, ellos proponen el uso de clasificadores naive bayes y rules induction, para predecir el promedio en 4 categorías: excelente, muy bueno, bueno y promedio. Sin embargo, los resultados de su clasificación son muy bajos sobre todo en las clases que están menormente representadas como la excelente.

De igual manera, en el trabajo de Velmurugan y Anuradha (2016) se comparan el clasificador bayesiano, *Bayes Net*, OneR, C4.5, IBK, JRip y J48 utilizando información de obtenida de estudiantes del Colegio de Ciencias en el estado de Tamil Nadu, India. También se comparan las técnicas de selección de características Cfs,

evaluación de subconjuntos, chi cuadrada, *info gain*, y *relief attribute evalation*. Su objetivo es predecir el promedio al finalizar el semestre mediante características demográficas de los estudiantes. Concluyen que el mejor método de selección para este caso fue CFS. El uso de la selección de características es común en este contexto, ya que las características recolectadas por los estudiantes pueden ser muy amplias, así como en el trabajo de Amrieh et al. (2016), en donde a través de una plataforma de aprendizaje web obtienen información demográfica, antecedentes académicos, la participación de los padres, y el comportamiento de los estudiantes durante el proceso académico para predecir el desempeño de estudiantes. Ellos realizan la selección de características de tipo filtro; consideran este paso muy importante para la predicción. Además, comparan sus resultados con los clasificadores J48, redes neuronales, el clasificador bayesiano y tres métodos de ensamble: *bagging*, *boosting* y bosque aleatorio.

Un enfoque similar se realiza en la universidad de inOndo en Nigeria, pues ellos toman atributos de maestros tanto personales como académicos para crear un modelo de predicción del desempeño del docente. Utilizan el método de selección de atributos *gain ratio* y los clasificadores ID3, C4.5 y MLP. Sus resultados muestran una buena clasificación y concluyen con las características más relevantes para el desempeño del docente (Asanbe et al., 2016). Otro enfoque se puede apreciar en el trabajo de Mueen et al. (2016), pues ellos toman información de estudiantes utilizando una plataforma de aprendizaje en línea y la separan en tres tipos: general, de la plataforma de aprendizaje y la académica. Su objetivo es predecir el rendimiento académico y reducir el número de características. Utilizando los algoritmos de selección de características del software WEKA de tipo clasificación (*Ranker*) concluyen las características más significativas para la predicción en su modelo.

Por otro lado, en el trabajo de Saarela et al. (2016) se utiliza información de la prueba PISA y un cuestionario con 53 preguntas referentes a actitudes con respecto al aprendizaje de matemáticas, con el cual se busca predecir el desempeño de matemáticas. Se aplican cuatro técnicas de selección de características y los clasifica-

dores de vecinos más cercanos, el clasificador bayesiano, LDA, máquinas de soporte vectorial y el bosque aleatorio. Los resultados muestran una buena clasificación para los primeros niveles; sin embargo, conforme aumenta el nivel la clase tiene una menor representación de instancias lo que hace difícil el entrenamiento del modelo. En cambio, en el trabajo de Márquez-Vera et al. (2016) se propone un algoritmo especializado en la predicción del abandono escolar temprano basado en un algoritmo denominado *Interpretable Classifcation Rule Mining* (ICRM) y una combinación con algoritmos evolutivos, además proponen una clasificación en diferentes etapas del proceso educativo. El sentido de esto es utilizar la información disponible en el momento adecuado para determinar la predicción. El método propuesto por ellos se basa en la generación de reglas, por lo que es posible extraer el conjunto de reglas para entender el modelo de predicción, la información que toman corresponde a estudiantes de preparatoria de la Universidad de Zacatecas, identifican el problema de clases desbalanceadas y utilizan SMOTE. Sus resultados son comparados con clasificadores como máquinas de soporte vectorial, C4.5 y el clasificador bayesiano obteniendo resultados competitivos.

3.3 Resumen

En este capítulo presentamos una revisión de la literatura en dos partes: la primera correspondiente a los métodos de selección y construcción de características y la segunda con respecto al caso de estudio presentado (el uso de minería de datos para temas relacionados con la educación). Podemos concluir de la primera parte que aún y cuando las técnicas de selección y construcción de características se han estudiado desde los 80's, ha habido desarrollos continuamente. Hoy en día tienen una función muy particular debido a que los avances tecnológicos permiten una mayor captación de datos, hasta lo que actualmente se conoce como "big data". Los artículos presentados en esta sección tienen en común el uso de estrategias evolutivas, siendo las tres principales algoritmos genéticos, programación genética y optimiza-

ción de partículas. Además, una de las características más frecuente es manejar las técnicas como dos cosas diferentes, es decir, se realiza el proceso de selección y de forma independiente el proceso de construcción de características; por otra parte, los artículos que utilizan programación genética se presentan para la construcción de una o varias características, las combinaciones de estas no necesariamente distinguen entre las características utilizadas, por lo que una característica puede usarse en repetidas ocasiones.

En cambio, la segunda parte de este capítulo muestra una secuencia de investigaciones con respecto a la predicción del desempeño en el campo de la educación, la literatura muestra el uso de clasificadores en su forma regular, los investigadores en este campo se enfocan en dos ideas, los datos que se utilizan para modelar la situación y obtener una mejor clasificación y cual es el clasificador que mejor se ajusta a este campo, los datos utilizados varían acorde al contexto educativo en el que se presenta el aprendizaje, de igual manera en la mayoría de ellos se centra en conocer situaciones particulares de los alumnos, si no se cuenta con información académica previa se toman datos que caracterizan al estudiante y si se cuenta con información académica esta es de mucho mayor utilidad, sin embargo, esta normalmente se toma después de iniciado el proceso. Esto ha llevado a tomar en cuenta más reactivos para mejorar la clasificación, llegando al punto de provocar redundancia y poca relevancia de algunos de los datos, por lo cual en los investigadores ha despertado el interés por técnicas de selección y construcción de características. Así se toman los datos más importantes, aunque muchos se enfocan en técnicas por método de filtro. Es un campo que puede explorarse en mayor medida.

Capítulo 4

Metodología

En este trabajo, se abordan dos problemas relacionados: *selección* y *construcción* de características. Mientras que en el primero se busca el subconjunto $\mathcal{C} \in \mathbb{C}$ de características más relevantes, en el segundo se buscan combinaciones de características con este mismo objetivo. Sin embargo, a diferencia de la selección y construcción convencionales, donde cualquier combinación es válida, tratamos el caso donde existen diferentes *grupos relacionados* de características.

Debido a que el problema de selección y construcción de características es un problema de optimización combinatoria en un espacio discreto, la técnica de algoritmos genéticos nos parece naturalmente apropiada para abordar dicho espacio, a pesar de que también existen técnicas evolutivas que se han binarizado para lidiar con este tipo de espacio .

Debido a que las características en los diferentes grupos solo se pueden mezclar entre ellas, hemos optado por diseñar un algoritmo que trabaja por bloques, donde cada bloque es una parte del cromosoma que representa a un grupo de características. En ese sentido, trabajamos en cinco niveles:

1. Nivel gen
2. Nivel sección

3. Nivel bloque

4. Nivel cromosoma

5. Nivel población

Tomamos como base el algoritmo genético canónico y hemos acoplado los diferentes operadores considerando que la población está compuesta por cromosomas, los cromosomas están compuestos por bloques, los bloques se dividen en secciones y las secciones contienen genes; esta representación se muestra en la figura 4.1. De igual manera, en el algoritmo 3 se puede ver en términos generales la estructura del algoritmo diseñado.

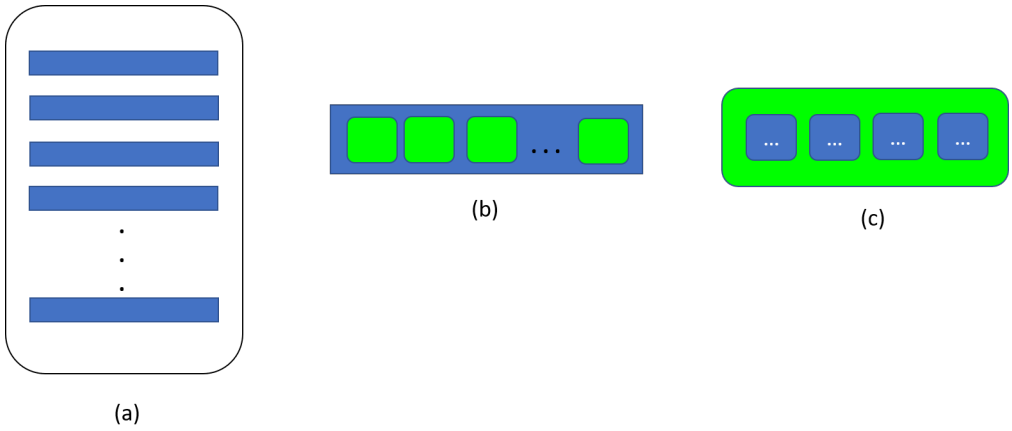


Figura 4.1: Representación utilizada: (a) la población se divide en cromosomas, (b) cada cromosoma se divide en bloques, (c) cada bloque se divide en cuatro secciones de genes: individuales apagados, individuales prendidos, compuestos apagados y compuestos prendidos.

4.1 Población inicial

El conjunto de datos esta compuesto por p instancias y q características, donde el total de características (q) se divide en agrupaciones que llamamos bloques. Se

Algoritmo 3 Algoritmo genético por bloques para la selección y construcción de características

Descripción: Recibe el conjunto de datos D con las instancias y características; la tupla $E_D = (b_0, b_1 \dots b_k)$ representando la estructura de bloques donde b_i es la cantidad de características en el bloque B_i, el tamaño de población t y la cantidad n de iteraciones, la probabilidad de mezcla p_{mz} y las probabilidades de mutación p_m, p_{ind} y p_f de acuerdo a diferentes modalidades. Regresa el mejor individuo encontrado considerando todas las poblaciones generadas.

1: **procedure** FSCGA($D, E_D, t, n, p_{mz}, p_m, p_{ind}, p_f$)
2: $\mathbb{P} \leftarrow$ Generar-Población-Inicial(t, E_D, p_{mz})
3: **while** no se ha llegado a n **do**
4: Aptitudes $\leftarrow$ Evaluar($\mathbb{P}, D$)
5: Parejas $\leftarrow$ Seleccionar($\mathbb{P}$, Aptitudes)
6: Hijos $\leftarrow$ Cruzar(Parejas)
7: $\mathbb{P}' \leftarrow$ Mutar(Hijos, p_m, p_{ind}, p_f)
8: $\mathbb{P} \leftarrow \mathbb{P}'$
9: **end while**
10: **return** individuo más apto de las poblaciones generadas
11: **end procedure**

pueden definir dos o más bloques en el conjunto de datos, donde este proceso depende del conocimiento previo de los datos. El algoritmo genético trabaja de forma independiente en cada uno de los bloques, es decir, las opciones de utilizar o no una característica o combinar un conjunto de estas se desarrolla en cada bloque. Por lo tanto, el cromosoma es un conjunto de bloques, como lo muestra la ecuación 4.1:

$$\mathbb{C} = (B(1), B(2), B(3), \ldots, B(k)) \tag{4.1}$$

Cada uno de los bloques está conformado por un determinado número de características. Algunas de estas características se aplican individualmente y otras se combinan para formar características compuestas. Por lo tanto, podemos tener (i) genes individuales y (ii) genes compuestos. Por otro lado, cada gen puede estar ya sea (a) apagado, pues la característica no se aplica o (b) prendido, pues la característica se aplica. Es por esta razón que consideramos que cada bloque está dividido en cuatro secciones diferentes: genes individuales apagados (I_{ap}), genes individuales prendidos (I_{pr}), genes compuestos apagados (C_{ap}) y genes compuestos prendidos (C_{pr}). La Ec. 4.2 muestra esta estructura:

$$B(i) = (I_{ap}, I_{pr}, C_{ap}, C_{pr}) \tag{4.2}$$

La población inicial es un conjunto de cromosomas conformado por bloques con (a) las características que no se utilizarán en la clasificación (I_{ap}), (b) las características que se utilizarán de forma natural (I_{pr}) y (c) las características que se utilizarán de forma combinada (C_{pr}). En cuanto a las características combinadas pero sin utilizar (C_{ap}), esta sección permanece vacía en la población inicial, puesto que consideramos que es más natural que esta sección contenga genes una vez iniciada la mutación.

El algoritmo 4 resume el proceso para generar la población inicial. Este proceso comienza con la representación de características mediante un cromosoma de tipo binario (por ejemplo, 100101010) donde el 1 se refiere a una característica prendida y el 0 a una característica apagada y la posición del gen representa la posición de la característica en el conjunto de datos. Después se toman las características prendidas y, basado en la probabilidad de combinación p_{mz}, se seleccionan las características

que se combinan y aleatoriamente se forman grupos de estas, donde las características seleccionadas para la construcción de características conforman un árbol binario de operaciones. Este árbol lo consideramos como un *gen compuesto*.

Algoritmo 4 Generar población inicial utilizando una estructura de bloques

Descripción: Recibe el tamaño t de la población, la probabilidad de mezcla p_{mz} y la estructura por bloques E_D. Devuelve la lista de cromosomas $\mathbb{P}$ (población inicial).

function Generar-Población-Inicial(t, E_D, p_{mz})
  for $i \leftarrow 0, t$ **do**
    for all $e \in E_D$ **do**
      $x \leftarrow$ Generar-Aleatorio-Binario$(1, 2^e - 1)$
      agregar posiciones de ceros en x a I_{ap}
      agregar posiciones de unos en x a Prendidos
    end for
    for $k \leftarrow 0, |\text{Prendidos}|$ **do**
      if Generar-Aleatorio$((0, 1)) < p_{mz}$ **then**
        Combinados $\leftarrow$ Prendidos(k)
      else
        $I_{pr} \leftarrow$ Prendidos(k)
      end if
    end for
    $C_{pr} =$ generar-compuestos(Combinados)
    $C_{ap} = ()$
    $B \leftarrow (I_{ap}, I_{pr}, C_{ap}, C_{pr})$
    agregar B a $\mathbb{C}$
    agregar $\mathbb{C}$ a $\mathbb{P}$
  end for
  return $\mathbb{P}$
end function

Para representar los árboles binarios de operaciones se utiliza la notación polaca postfija. Si se tienen dos o más características para combinar, primero se forman

Algoritmo 5 Generar gen compuesto representado por árbol binario de operaciones

Descripción: Recibe el conjunto *Combinados* de genes individuales a combinar y genera un conjunto de grupos con diferentes tamaños cada uno; para cada grupo, a su vez se genera un árbol binario de operaciones que considera $\{+, -, *, \text{máx}, \text{mín}\}$ como operadores. Cada grupo representa un gen compuesto que se agrega a la tupla *Sección* y esta se devuelve como respuesta.

```
function Generar-compuestos(Combinados)
    inicializar Sección
    while |Combinados| > 0 do
        Grupo ← Escoger-Grupo-Aleatorio(2, |Combinados|)
        Árbol ← Generar-Árbol-de-Operaciones(Grupo)
        agregar Árbol a Sección
        quitar Grupo de Combinados
    end while
    return Sección
end function
```

parejas, donde cada pareja se combina al emplear un operador (suma, resta, multiplicación, máximo o mínimo). Cada pareja forma un nuevo resultado, y este proceso continúa hasta obtener un solo resultado para cada grupo a combinar. Para ilustrar este proceso se presenta el siguiente ejemplo: si se cuenta con un conjunto de datos de 60 características organizadas en bloques de 20, 10, 15 y 25 características, entonces el primer paso es generar un bloque de números binarios de tamaño del número de características del bloque

$$B(1) = (1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1, 1).$$

Después, tomando como base el orden de las características se separan en genes individuales apagados $I_{ap} = (4, 5, 6, 10, 13, 15, 16)$ y los prendidos
$(0, 1, 2, 3, 7, 8, 9, 11, 12, 14, 17, 18, 19)$. De acuerdo con la probabilidad de combinar, podrían separarse de forma aleatoria en $I_{pr} = (3, 7, 14, 18)$ y los seleccionados para combinar en $(0, 1, 2, 8, 9, 11, 12, 17, 19)$. También de forma aleatoria los seleccionados

para combinar podrían separarse en grupos $C_{pr} = ((2, 8, 17, 19), (0, 1, 9), (11, 12))$ y por último, se genera el árbol en notación postfija para cada uno de los grupos de combinados prendidos: $(2, 8, +, 17, 19, -, *)$, $(0, 1, -, 9, \text{máx})$, $(11, 12, *)$.

Por lo tanto, el bloque $B(1)$ del cromosoma está compuesto por

$$\begin{aligned} I_{ap} &= (4, 5, 6, 10, 13, 15, 16), \\ I_{pr} &= [3, 7, 14, 18), \\ C_{ap} &= (), \\ C_{pr} &= ((2, 8, +, 17, 19, -, *), (0, 1, -, 9, \text{máx}), (11, 12, *)). \end{aligned}$$

4.2 Evaluación

Una vez generada la población inicial, el siguiente paso corresponde a la evaluación, en la cual se genera un conjunto de datos por cada cromosoma y se ingresa este conjunto de datos a un clasificador. Este proceso se muestra en el algoritmo 6, donde cada gen individual prendido $g \in I_{pr}$ indica características que se toman directamente del conjunto de datos; sin embargo, no es así con cada gen compuesto $g \in C_{pr}$, ya que que con ellos se realizan las operaciones matemáticas designadas por el árbol de operaciones respectivo en las características correspondientes del conjunto de datos.

Continuando con el ejemplo de la sección anterior, si se cuenta con tres genes compuestos encendidos, $(2, 8, +, 17, 19, -, *)$, $(0, 1, -, 9, \text{máx}), (11, 12, *)$, el proceso de evaluación convierte cada gen compuesto en una sola característica. El proceso se puede ejemplificar de la siguiente manera:

$$\begin{aligned} f_0 &= \text{característica } 2 + \text{característica } 8 \\ f_1 &= \text{característica } 17 - \text{característica } 19 \\ f_3 &= f_0 * f_3. \end{aligned}$$

Entonces la nueva característica del conjunto de datos es f_3. Para el segundo gen

compuesto, se desarrolla el mismo proceso:

$$\begin{aligned} f_0 &= \text{característica } 0 - \text{característica } 1 \\ f_1 &= \text{máx}(f_0, \text{característica } 9) \end{aligned}$$

Por lo tanto, la siguiente nueva característica es f_1, y por último se aplica para el tercer gen compuesto:

$$f_0 = \text{característica } 11 * \text{característica } 12$$

De esta manera, nuevo conjunto de datos a clasificar contiene las características $\{4, 5, 6, 10, 13, 15, 16, f_3, f_1, f_0\}$.

Como función de aptitud se utiliza la medida F (Ec. 2.4, pág. 14), considerando que existe desbalance en el conjunto de datos y que es relevante mejorar la predicción en la clase minoritaria. De esta manera, el proceso de evaluación genera un listado de aptitudes para el conjunto de cromosomas, Aptitudes $= (a_1, a_2, a_3, ..., a_{\mathbb{P}})$.

4.3 Selección

El método de selección (ilustrado en el algoritmo 7) empleado es una combinación de ruleta y torneo. El proceso consiste en obtener la probabilidad de selección de cada uno de los cromosomas (Ec. 4.3). Se gira la ruleta para obtener un par de cromosomas de los cuales solo pasa el más apto hasta obtener la mitad de cromosomas de la población inicial agrupados en pares (progenitor$_a$, progenitor$_b$). Debido a que la selección se realiza a nivel cromosoma, no se requiere hacer un manejo especial por bloques, es decir, no se requiere adaptar esta operación para el contexto de características agrupadas.

$$p_{sel}(i) = \frac{a_i}{\sum_0^{\mathbb{P}} a} \tag{4.3}$$

Algoritmo 6 Evaluación

Descripción: Recibe un conjunto de datos $D = (\mathcal{I}(1), \mathcal{I}(2), \mathcal{I}(p))^T$, con p instancias, donde cada instancia tiene la forma $\mathcal{I}_k = (f(1), f(2), \dots f(q))$ y cada $f(j)$ representa una característica. Se genera un nuevo conjunto de datos D' con las características prendidas de cada cromosoma y con características que se generan a partir de las características prendidas combinadas de cada bloque del cromosoma. D' se clasifica y se obtiene una aptitud, la cual se guarda en la lista *Aptitudes*. Esta lista es retornada como resultado.

```
1: function Evaluación(ℙ, D)
2:     for i ← 0, |ℙ| do
3:         ℂ ← ℙ(i)
4:         for j ← 0, ℂ do
5:             (I_ap, I_pr, C_ap, C_pr) ← ℂ(j)
6:             for k ← 0, p do
7:                 ℐ(k) ← I_pr
8:                 for z ← 0, |C_pr| do
9:                     Árbol ← C_pr(z)
10:                    f' ← Realizar-Operación(Árbol)      ▷ Se crea la característica
11:                    agregar f' a ℐ(k)                   ▷ Se agrega a la instancia
12:                end for
13:                agregar ℐ(k) a D'
14:            end for
15:        end for
16:        Aptitud(i) ← Clasificar(D')
17:        agregar Aptitud(i) a Aptitudes
18:    end for
19:    return Aptitudes
20: end function
```

Algoritmo 7 Selección combinando ruleta y torneo

Descripción: Recibe la población $\mathbb{P}$ y la lista de aptitudes *Aptitudes*, y devuelve la lista *Parejas*, la cual contiene pares de individuos a cruzar.

```
1: procedure SELECCIONAR(ℙ, Aptitudes)
2:     numParejas ← |ℙ|/2
3:     for i ← 0, numParejas do
4:         for j ← 0, 2 do ▷ Quedarán dos progenitores en Pareja al terminar ciclo
5:             competidor1 ← GIRARRULETA(ℙ, Aptitudes)
6:             competidor2 ← GIRARRULETA(ℙ, Aptitudes)
7:             if APTITUD(competidor1) > APTITUD(competidor2) then   ▷ Torneo
8:                 progenitor ← competidor1
9:             else
10:                progenitor ← competidor2
11:            end if
12:            agregar progenitor a Pareja
13:        end for
14:        agregar Pareja a Parejas
15:    end for
16:    return Parejas
17: end procedure
```

4.4 Cruza

El método de cruza recibe la selección en forma de pares (progenitor$_a$, progenitor$_b$). Durante este proceso se identifica el cromosoma con mejor aptitud. Si alguno de los progenitores es el individuo de mayor aptitud se genera una probabilidad de pasar sin modificaciones; si este individuo pasa a la siguiente generación, solo se genera un hijo que sustituye al otro padre. Si ninguno de los individuos es el de mayor aptitud, se realiza la cruza generando dos hijos.

Para la cruza se considera el intercambio de bloques entre cromosomas. En este caso, es necesario decidir la cantidad r de bloques que se intercambiarán entre el cromosoma progenitor$_a$ y el cromosoma progenitor$_b$, considerando que debe ser una cantidad significativa, pero no una cantidad excesiva. Optamos por escoger $r = \lceil n/2 \rceil$ (donde n es la cantidad de bloques del cromosoma), considerando que sería una cantidad suficiente de bloques para intercambiar.

4.5 Mutación

La mutación es uno de los procesos clave, ya que a través de ella se modifican los genes del cromosoma para generar nuevos individuos diferentes a los padres y así buscar en espacios no explorados posibles soluciones con aspectos mejores que no se hayan identificado anteriormente. La mutación se realiza para cada uno de los bloques que conforma el cromosoma. Ya que cada bloque está conformado por cuatro secciones, cada una de las secciones presenta una forma de mutación distinta (aunque análoga en varios casos). Este proceso se detalla en el algortimo 8.

Individuales apagados Para mutar los genes individuales apagados $g \in I_{ap}$, se utiliza la probabilidad de mutación p_m. De acuerdo con esta probabilidad, el gen puede mutar en un gen individual prendido I_{pr}, en un gen con posibilidad de combi-

Algoritmo 8 Mutación basada en bloques Parte I: Algoritmo general

```
1: function MUTAR(ℙ, p_m, p_ind, p_f)
2:     inicializar ℙ′
3:     for i ← 0, |ℙ| do
4:         inicializar ℂ′
5:         ℂ ← ℙ_i
6:         for j ← 0, |ℂ| do
7:             B ← ℂ_j
8:             B′ ← MUTAR-IAP(B, p_m)
9:             B′ ← MUTAR-IPR(B′, p_m)
10:            B′ ← MUTAR-CAP(B′, p_m, p_ind, p_f)
11:            B′ ← MUTAR-CPR(B′, p_m, p_ind, p_f)
12:            agregar B′ a ℂ
13:        end for
14:        agregar ℂ′ a ℙ′
15:    end for
16:    return ℙ′
17: end function
```

narse con otros genes individuales para conformar un gen compuesto, o quedar igual (no se muta), como se muestra en el algoritmo 9.

Algoritmo 9 Mutación Parte II: mutar individuales apagados

Descripción: Recibe un bloque B y una probabilidad de mutación p_m. Si el gen se muta, puede pasar a individuales apagados (I_{ap}) o a combinados prendidos (C_{pre}). Se genera un nuevo bloque con los resultados de la mutación.

```
1: function MUTAR-IAP(B, p_m)
2:     B = (I_ap, I_pr, C_ap, C_pre)
3:     for i ← 0, |I_pr| do
4:         if OBTENER-ALEATORIO(0, 1) < p_m then                ▷ Se muta
5:             if OBTENER-ALEATORIO(0, 1) > 0.5 then
6:                 agregar I_pr(i) a I'_ap                      ▷ Se apaga
7:             else
8:                 Combinados ← I_pr(i)                         ▷ Se combina
9:             end if
10:        else
11:            agregar I_pr(i) a I'_pr                          ▷ No se muta
12:        end if
13:        if |Combinados| < 2 then
14:            agregar Combinados a I'_pr
15:        else
16:            C'_pr ← GENERAR-COMPUESTOS(Combinados)
17:        end if
18:    end for
19:                                  ▷ Solamente compuestos apagados no se modifican
20:    return (I'_ap, I'_pr, C_ap, C'_pre)
21: end function
```

Individuales prendidos El proceso de mutación para un gen individual prendido $g \in I_{pr}$ es análogo al proceso de mutación de los genes individuales apagados.

En este caso, de acuerdo a p_m el gen puede mutar a individual apagado, combinarse junto con otros en un gen compuesto o no mutar.

Los genes individuales que mutaron a genes con posibilidad de combinar deben ser más de dos. Si esta condición no se cumple, el gen pasa a su estado original, ya sea individual apagado o individual prendido. Los genes individuales que pasan a formar parte de genes compuestos se anexan aleatoriamente a distintos árboles binarios de operaciones, de manera similar a como se explicó en la generación de la población inicial.

Combinados prendidos Cada gen compuesto prendido $g \in C_{pr}$ aparece en notación postfija y se somete a una probabilidad de mutar p_m. La primer forma de mutar es cambiar el gen a *compuesto apagado (C_{ap})*, de tal manera que se toma a todo el grupo incluyendo su notación postfija y se cambia a un estado inactivo. En el ejemplo que hemos explicado, la sección C_{ap} se encuentra vacía y este tipo de mutación es la que permite que estos aparezcan en generaciones futuras:

$$\begin{aligned} C_{ap} &= () \\ C_{pr} &= ((2, 8, +, 17, 19, -, *), (0, 1, -, 9, \text{máx}), (11, 12, *)). \end{aligned}$$

Por lo tanto, se puede mutar un gen compuesto prendido a compuesto apagado, como se muestra a continuación:

$$\begin{aligned} C'_{ap} &= (0, 1, -, 9, \text{máx}) \\ C_{pr} &= ((2, 8, +, 17, 19, -, *), (11, 12, *)), \end{aligned}$$

donde C'_{ap} muestra el resultado de la mutación para la sección C_{ap}. De esta forma, el gen compuesto $(0, 1, -, 9, max)$ ya no estará activo al realizar la evaluación. En cambio, los genes compuestos prendidos que no mutaron entran en una nueva probabilidad de mutar por escaneo de genes individuales o por función.

Mutación por escaneo de genes individuales: el algoritmo 10 muestra este proceso en el cual, de acuerdo a la probabilidad de mutación por escaneo de genes individuales p_{ind}, se genera un recorrido de cada uno de los genes que integra el gen

compuesto. Cada uno de estos genes puede cambiar a individuales prendidos I_{pr} o pasar a formar parte de una nueva combinación. Si ningún gen mutó, la combinación permanece prendida; en cambio, si al menos uno de los genes individuales mutó, entonces, el gen compuesto original desaparece. En este caso, se revisa si quedaron dos o más genes para formar un nuevo gen compuesto; si es así, se genera un nuevo árbol de operaciones; de lo contrario, se reasigna el gen solitario a los individuales prendidos. En el ejemplo que hemos manejado, $C_{pr} = ((2, 8, +, 17, 19, -, *), (11, 12, *))$. Si el primer gen compuesto entra en el proceso de mutación por escaneo de genes individuales, entonces existe la posibilidad de cambiar genes individuales a I_{pr} y a la combinación de atributos restantes del gen compuesto se les asigna una nueva operación en notación postfija:

$$\begin{aligned} I'_{pr} &= (2, 7, 13, 14, 17) \\ C'_{pr} &= (8, 19, *), \end{aligned}$$

donde I'_{pr} y C'_{pr} muestran el resultado de la mutación para las secciones de individuales prendidos y compuestos prendidos, respectivamente.

Mutar por función: Cuando el gen compuesto prendido no mutó por grupo o por escaneo de genes individuales, entonces entra a un proceso de mutar por función de acuerdo a una probabilidad p_f, de tal manera que se recorren cada uno de los operadores del gen compuesto y aleatoriamente se cambia el operador (también puede permanecer sin cambios). Al finalizar este proceso, se obtienen las nuevas secciones del bloque:

$$\begin{aligned} B(1) &= (I'_{ap}, I'_{pr}, C'_{ap}, C'_{pr}) \\ I'_{ap} &= (4, 6, 10, 15) \\ I'_{pr} &= (2, 7, 13, 14, 17) \\ C'_{ap} &= (0, 1, -, 9, \text{máx}) \\ C'_{pr} &= ((3, 16, +), (5, 18, \text{máx}), (8, 19, *), (11, 12, +)) \end{aligned}$$

Note que el último gen cambió de signo $*$ a $+$; esto, debido al último proceso de mutación posible. Sin embargo, existe una posibilidad de no realizar ningún cambio

en el cromosoma. Es decir, aunque existen diferentes modalidades de mutación, cada gen — ya sea individual o compuesto — al final del día puede no mutarse.

Combinados apagados El proceso de mutación para los genes compuestos apagados es análogo al de los genes compuestos prendidos, pues pueden mutar por grupo, por escaneo de genes individuales o por función.

4.6 Resumen

Nuestra propuesta se basa en la combinación de dos procesos: selección y construcción de características. Esto, con el objetivo de mejorar la predicción en la clasificación de conjuntos de datos binarios, ya que el problema abordado se puede representar como un problema de optimización combinatoria. Se propuso el desarrollo de un algoritmo genético, donde la característica de nuestro algoritmo genético es que el proceso de selección y construcción están combinados en un mismo proceso; además, nuestro diseño se enfoca en conjuntos de datos en los que se manejan *grupos* de características, es decir, cuando se conocen la estructura de bloque de las características del conjunto de datos. Otro de los puntos que resaltan de nuestro diseño es el proceso de mutación, ya que este se acopla al trabajo por grupos y a la estructura de los bloques diseñada.

En términos generales, nuestro diseño es capaz de tomar un conjunto de datos y con ellos realizar una gran cantidad de pruebas con un clasificador determinado de la selección de características (utilizar o no una característica del conjunto de datos) y construcción de características (mezclar características con funciones matemáticas) y obtener una combinación de atributos para mejorar la clasificación.

Algoritmo 10 Mutación parte III: mutar compuestos prendidos

Descripción: Recibe un bloque con forma $(I_{ap}, I_{pr}, C_{ap}, C_{pr})$, así como probabilidades de mutación por grupo p_m, mutación por escaneo de genes individuales p_s y mutación por función p_f. Devuelve un nuevo bloque $(I'_{ap}, I'_{pr}, C'_{ap}, C'_{pr})$ con los resultados.

1: **function** MUTAR-CPR($(I_{ap}, I_{pr}, C_{ap}, C_{pr}), p_m, p_s, p_f$)
2: **for** $i \leftarrow 0, |C_{pr}|$ **do** ▷ Recorre cada grupo
3: **if** OBTENER-ALEATORIO$(0, 1) < p_m$ **then** ▷ Mutar por grupo
4: **agregar** $C_{pr}(i)$ a C'_{ap}
5: **else**
6: **if** OBTENER-ALEATORIO$(0, 1) < p_s$ **then**
7: **for** $j \leftarrow 0, |C_{pr}(i)|$ **do** ▷ Por genes individuales
8: **quitar** $C_{pr}(i, j)$ de $C_{pr}(i)$
9: $C^n_{pr}(i) \leftarrow$ GENERAR-COMBINACION$(C_{pr}(i))$
10: **if** OBTENER-ALEATORIO$(0, 1) < 0.5$ **then**
11: **agregar** $C^n_{pr}(i)$ y $C_{pr}(i, j)$ a C'_{ap} e I'_{ap}, respectivamente
12: **else**
13: **agregar** $C^n_{pr}(i)$ y $C_{pr}(i, j)$ a C'_{pr} e I'_{pr}, respectivamente
14: **end if**
15: **end for**
16: **else**
17: **if** OBTENER-ALEATORIO$(0, 1) < p_f$ **then**
18: Arbol$'$ $\leftarrow$ CAMBIAR-OPERADORES$(C_{pr}(i))$ ▷ Por función
19: **agregar** Arbol$'$ a C'_{pr}
20: **else**
21: **agregar** $C_{pr}(i)$ a C'_{pr} ▷ No mutar
22: **end if**
23: **end if**
24: **end if**
25: **end for**
26: **return** $(I'_{ap}, I'_{pr}, C'_{ap}, C'_{pr})$
27: **end function**

Capítulo 5

Caso de Estudio

El uso de técnicas de minería de datos en educación es un área reciente que ha llamado la atención de muchos investigadores enfocados a resolver problemáticas en el ámbito educativo. La tendencia de la minería de datos para la educación se ha especializado en cinco categorías: predicción, agrupamiento, minería de relaciones, purificación de datos para el juicio de las personas y descubrimiento con modelos Baker y Yacef (2009). En este último, los autores identifican como una de las claves el modelado del estudiante para determinar las características que puedan clasificar con mayor eficiencia a los estudiantes en recientes años. Esta clave sigue siendo uno de los objetivos de la minería de datos en la educación (Bakhshinategh et al., 2018).

Uno de los principales objetivos en el proceso educativo es lograr que la mayor cantidad de estudiantes obtengan el aprendizaje requerido para egresar en el tiempo establecido. Para cumplir este objetivo se debe considerar una gran cantidad de factores que intervienen en el proceso formativo del estudiante y que pueden provocar desde un bajo desempeño hasta posiblemente la interrupción de sus estudios. Estos factores dependen directamente del escenario en el que se desarrolla el proceso formativo, por lo que es necesario entenderlos de forma distinta (Romero y Ventura, 2010).

El caso de estudio corresponde a una escuela de nivel medio superior técnica, la

cual está ubicada en el área metropolitana de Monterrey en el estado de Nuevo León, en México. Sus programas educativos tienen enfoque en el área industrial y de servicios, y actualmente es pertenece al Sistema Nacional de Educación Media Superior[1]. Este organismo en sus inicios se sustentó en la Reforma Integral de la Educación Media Superior (RIEMS) que establece el modelo educativo basado en competencias[2], el cual, dicho en forma simplificada es que los programas educativos están diseñados para el desarrollo de competencias *genéricas* (para la vida),*disciplinares básicas y extendidas* (específicas de un área de estudio, por ejemplo física) y las *profesionales* (específicas para el trabajo). Por lo tanto, los programas educativos de la escuela son basados en competencias y se conforman por unidades de aprendizaje de bachillerato general y de formación para el trabajo, sumando un total de 35 horas por semana, distribuidas en un promedio de 11 asignaturas por semestre, con una duración de 6 semestres.

La población total es superior a los siete mil alumnos, con un ingreso anual promedio de dos mil alumnos. Su porcentaje de egreso generacional[3] se encuentra en el 68.8 % en el 2018, lo cual supera la media nacional y estatal que se encuentra en 64.2 % y 68.1 % respectivamente en el ciclo 2018/2019 según las cifras publicadas por el INEGI [4]. El porcentaje de alumnos de abandono al finalizar el plan de estudios (3 años) es de 31.2 % y, si lo reducimos a el abandono durante el primer año de estudios, el porcentaje es de 15.2 % por lo cual, la cantidad de alumnos de abandono escolar se presenta en una menor cantidad que los estudiantes que continúan de forma regular su proceso académico.

En conocimiento que el ambiente escolar, las situaciones propias de cada estudiante y el contexto social en el que habitan puede afectar su desempeño académico

[1]Información disponible en: `http://educacionmediasuperior.sep.gob.mx/sinems`

[2]Información disponible en: `http://dof.gob.mx/nota_detalle.php?codigo=5061936&fecha=26/09/2008`

[3] Es el porcentaje de alumnos que abandonan las actividades escolares durante el ciclo escolar y al finalizar éste, respecto al total de alumnos inscritos en el ciclo escolar.

[4]Información disponible en: `https://www.inegi.org.mx/temas/educacion/`

la escuela cuenta con programas de atención preventiva, las cuales apoyan al estudiante y ayudan a reducir el abandono escolar, entre ellas se encuentras programas de asesoría académica, tutorías individuales y grupales, orientación psicológica y nutrición.

Tanto en la escuela del caso de estudio, como en las investigaciones presentadas en el capítulo 3, se tiene conocimiento que existe una gran cantidad de variables que afectan el rendimiento académico y cada una de las escuelas en la literatura muestra diferentes formatos para modelar el estudiante, tales como información económica, cultural y académica, entre otras, sin embargo, se ha tomado la información historia disponible al momento para este estudio. La escuela marcó como uno de sus objetivos prioritarios el disminuir el abandono académico. Por tal motivo, se desarrollaron herramientas para captar información de los estudiantes y se mejoraron los informes de resultados para llegar a procesos de análisis de la información. Al inicio de cada generación, se toman datos de los estudiantes inscritos, lo cual se describe a continuación.

La Tabla 5.1 muestra las seis categorías del instrumento estandarizado para la evaluación de la aptitud académica de los aspirantes (EXANI I) [5], la cual forma parte de la *Información académica del proceso de asignación de espacios a Nivel Medio Superior* y es aplicada a todos los aspirantes a ingresar a la dependencia. Cada una de las pruebas evalúa un conjunto de habilidades y conocimientos mediante una cantidad de reactivos. El resultado obtenido es un puntaje entre 0 y 100 para cada una de ellas. Por sí mismo, este resultado permite visualizar el conocimiento adquirido en los niveles básicos (primaria y secundaria) antes del ingreso al nivel medio superior, aunque esto no es suficiente para discriminar a los estudiantes de abandono escolar. Con la finalidad de identificar otros factores, la dependencia educativa ha aplicado un cuestionario durante el proceso de inscripción, mismos que se muestran en la Tabla 5.2. En este cuestionario se toma información referente al núcleo familiar, hábitos de alimentación, hábitos de salud, hábitos de estudio, percepción de sí

[5]Información disponible en: `https://www.ceneval.edu.mx/exani-i`

Tabla 5.1:

Prueba de habilidades y conocimientos

Tipo de Prueba	**Rango**
Habilidad Verbal	0-100
Habilidad Numérica	0-100
Español	0-100
Matemáticas	0-100
Ciencias Sociales	0-100
Ciencias Naturales	0-100

Tabla 5.2:

Cuestionario aplicado durante inscripción

Sección de la encuesta	**Preguntas**	**Rango**
Núcleo familiar	4	0-100
Hábitos de salud	5	0-100
Hábitos de alimentación	6	0-100
Hábitos de estudio y percepción de si mismo	15	0-100
Aspectos económicos	7	0-100

mismo y aspectos económicos.

5.1 Resumen

Para generar los conjuntos de datos con los que se experimentará, se hará uso de información anonimizada[6] sobre el rendimiento escolar y diversos aspectos de los alumnos de una preparatoria técnica pública del área metropolitana de Monterrey Nuevo León. Esta información concentra los resultados del examen de aptitud

[6]Aunque se contó con el permiso de las autoridades se omitieron datos de identificación de los estudiantes y el nombre de la escuela.

(EXANI I) aplicado a los estudiantes para el ingreso a la preparatoria y los resultados de la encuesta aplicada al momento de inscripción. Mientras que el examen de aptitud evalúa habilidad matemática, habilidad verbal, matemáticas, español, ciencias sociales y ciencias naturales, la encuesta evalúa aspectos del núcleo familiar, hábitos de salud, hábitos de estudio, hábitos de alimentación, percepción de sí mismo y aspectos económicos.

Capítulo 6

Experimentos y Resultados

En este capítulo presentamos los resultados obtenidos al comparar nuestra propuesta contra diferentes métodos, tanto para selección como para construcción de características. Cabe destacar que el algoritmo genético por bloques superó a estos métodos.

6.1 Configuración experimental

El caso de estudio presentado en la sección 5 se alinea con los requerimientos de la metodología propuesta en cuanto a la estructura del conjunto de datos, que está dividida en bloques. Lo anterior, debido a que es posible separar los datos en dos: Concurso de Ingreso y Cuestionario. A su vez, el cuestionario se puede dividir en cinco grupos debido a los factores evaluados en él. Por otro lado, se tienen registros para los años 2015, 2016, 2017 y 2018, por lo que se cuenta con cuatro conjuntos de datos.

Cada uno de los conjuntos de datos del caso de estudio utilizados presenta desbalance de clases en distintas proporciones; en la Tabla 6.1 se muestra la distribución de las clases, donde el IR se calcula como se estableció en el Capítulo 2, en la Ecuación 2.8 (página 25). La proporción de la clase minoritaria es tan solo de

Tabla 6.1:

Distribución de clases de los conjuntos de datos

Conjunto de Datos	Minoritaria	Mayoritaria	IR
2015	58	717	9.1923
2016	62	925	14.9194
2017	99	1076	10.8686
2018	63	1305	20.7142

un 5 % a un 9 % en comparación a la mayoritaria. Por tal motivo, se contempló el uso del método de sobre muestreo SMOTE (Chawla et al., 2002), de tal manera que cada uno de los experimentos tiene como parámetro el porcentaje de sobre muestreo aplicado a s en un intervalo de $0 \leq s \leq 1$, en donde 0 se refiere a no aplicar sobre muestreo y 1 se refiere a un sobre muestro hasta igualar la cantidad de instancias para cada clase.

En cada experimento se realizaron 30 corridas con la finalidad de demostrar la confiabilidad del método, mientras que el 70 % de cada conjunto de datos se utilizó para entrenamiento y el 30 % restante para prueba. El método utilizado para generar los conjuntos fue el *Train Test Split* de la librería de `Scikit Learn Model Selection` en `Python 3.0`, la cual consiste en un proceso aleatorio de selección de instancias de entrenamiento y prueba, respetando un porcentaje determinado de instancias para cada clase.

Como método de clasificación se utilizó el bosque aleatorio. Al ser un método basado en árboles de decisión, este método tiene una buena discriminación de datos categóricos; además, su estructura de ensamble ayuda a disminuir el problema de clases desbalanceadas. Sin embargo, uno de los factores críticos por los cuales este método fue escogido es el tiempo de respuesta, ya que tarda aproximadamente $\frac{1}{3}$ del tiempo requerido por una red neuronal de tipo multicapa, de acuerdo con nuestros resultados preliminares. Los parámetros utilizados en nuestros experimentos son los establecidos por defecto en la librería `SciKit Learn Ensemble`, los cuales se descri-

Tabla 6.2:

Parámetros del bosque aleatorio

Parámetro	**Valor**
Número de árboles en el bosque	100
Máximo de características	auto = `sqrt`(número de características)
Profundidad Máxima	Ninguna
Número de ejemplos mínimos para separar un nodo interno	2
Criterio	Gini
Bootstrap	`true`

ben en la Tabla 6.2.

Los métodos contra los cuales se comparó fueron los siguientes:

Conjunto de datos completo.- No se realizó ni selección ni construcción, sino que se trabajó con todas las características del conjunto de datos.

Método de Alpinista Aleatorio (*Random Hill Climber*).- En este método, existen tres pasos básicos: (a) construir una solución inicial, (b) evaluar la solución y (c) construir una solución vecina de manera aleatoria, donde (b) y (c) se repiten hasta llegar a un criterio de terminación (Morales et al., 2013). Tomamos dos variantes: (1) una donde se hace solo selección de características, con siglas RHCFS y (2) otra donde se hacen ambas selección y construcción, con siglas RHCFSC. Mientras que la primera no trabaja por bloques, la segunda sí. Para este algoritmo, se consideraron 200 iteraciones, 30 corridas y probabilidad de mutación de 0.1.

Algoritmo genético con solo selección.- Es el algoritmo genético canónico para selección de características. Para este algoritmo genético, se utilizaron los mismos parámetros que para el algoritmo genético por bloques para selección y construcción, excepto lo referente a la construcción.

Ya que—como se mencionó anteriormente—los conjuntos de datos presentan desbalance de clases, se utiliza la medida F (Ec. 2.4, pág. 14) de la clase minoritaria como medida de evaluación, pues brinda una perspectiva más clara sobre la calidad de la clasificación.

6.2 Resultados

A continuación se muestran los resultados para diferentes tipos de experimentos realizados. El primer experimento (denominado *Experimento 1*), consistió en no hacer sobre muestreo a los conjuntos de datos, mientras que en el segundo experimento (denominado *Experimento 2*) sí se utilizó generación de instancias sintéticas con la técnica SMOTE.

6.2.1 Resultados del Experimento 1

En la presente sección, los conjuntos de datos se utilizaron en su forma original (es decir, sin sobre muestreo). En la Tabla 6.3 se muestran los resultados promedio para cada conjunto de datos, donde se puede apreciar un incremento desde $F = 0.0803$ con el conjunto completo de características hasta una $F = 0.4318$ para la propuesta y un incremento de un 0.3715. De la misma manera, en la figura 6.1, se muestra la distribución de los resultados siendo el GAFSC superior.

Configuración del experimento 1

1. grupos = [6, 4, 5, 6, 15, 7]
2. operaciones = $\{+, -, *, \text{máx}, \text{mín}\}$
3. tamaño de la población $(t) = 50$
4. probabilidad de combinar $(p_{mz}) = 0.5$

Tabla 6.3:

Resultados promedio de Experimento 1

Conjunto de Datos	Completo	RHCFS	RHCFSC	GAFS	GAFSC
2015	0.0775	0.2990	0.2965	0.2558	**0.4863**
2016	0.0748	0.2691	0.2671	0.2279	**0.4200**
2017	0.1316	0.3033	0.3037	0.2697	**0.4290**
2018	0.0372	0.2233	0.2305	0.1779	**0.3920**
Promedio	0.0803	0.2737	0.2592	0.2328	**0.4318**

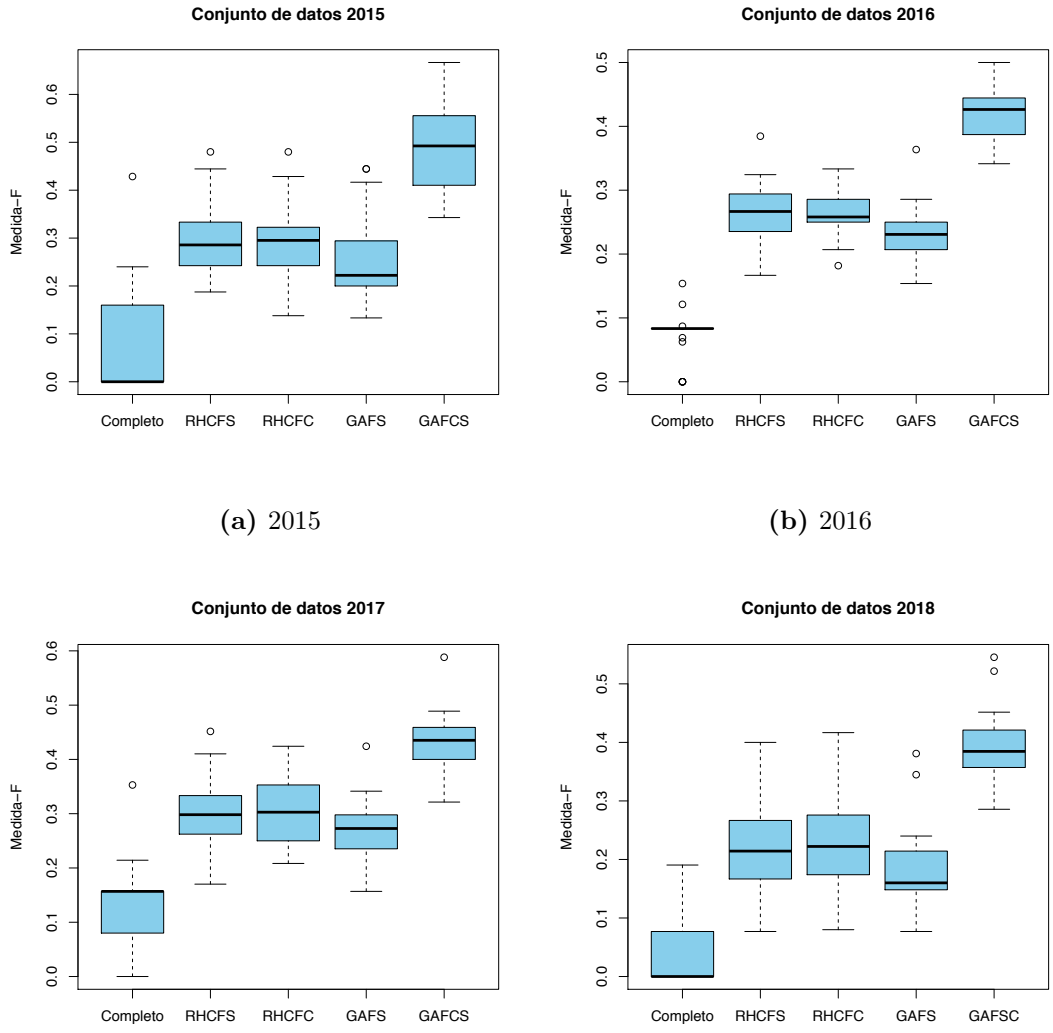
nota: Los mejores puntajes se muestran en negritas

5. probabilidad de mutar (p_m) = 0.1
6. iteraciones (n) = 200
7. corridas = 30
8. $s = 0$ (no hay balanceo de clases)

El algoritmo inicia con una población aleatoria del espacio de búsqueda y al paso de las generaciones los resultados mejoran considerablemente. El máximo en la medida F obtenido en cada una de las generaciones en las pruebas realizadas están representados en la figura 6.2, en donde es notable que en las primeras 100 generaciones se identifica el mayor incremento.

6.2.2 Resultados del Experimento 2

El modelo por seguir en la presente sección inicia al aplicar la técnica de sobre muestreo SMOTE al conjunto de datos, la cual consiste en generar nuevas instancias a partir de los elementos de la clase minoritaria utilizando la interpolación entre los k vecinos más cercanos para generar x cantidad de instancias sintéticas. La variable s es la responsable de calcular la cantidad de instancias sintéticas, donde



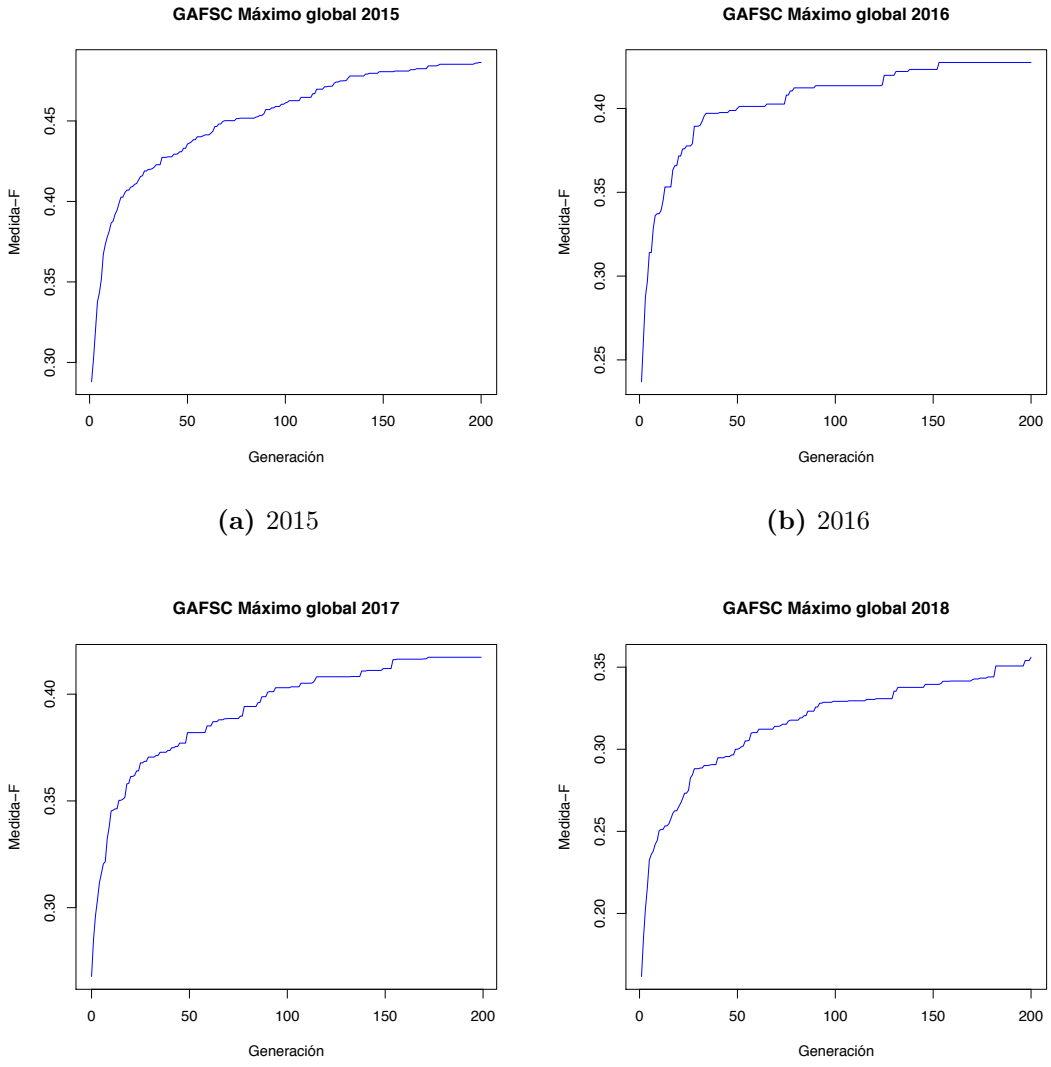
(a) 2015

(b) 2016

(c) 2017

(d) 2018

Figura 6.1: Distribución de los resultados del Experimento 1.



(a) 2015

(b) 2016

(c) 2017

(d) 2018

Figura 6.2: Experimento 1

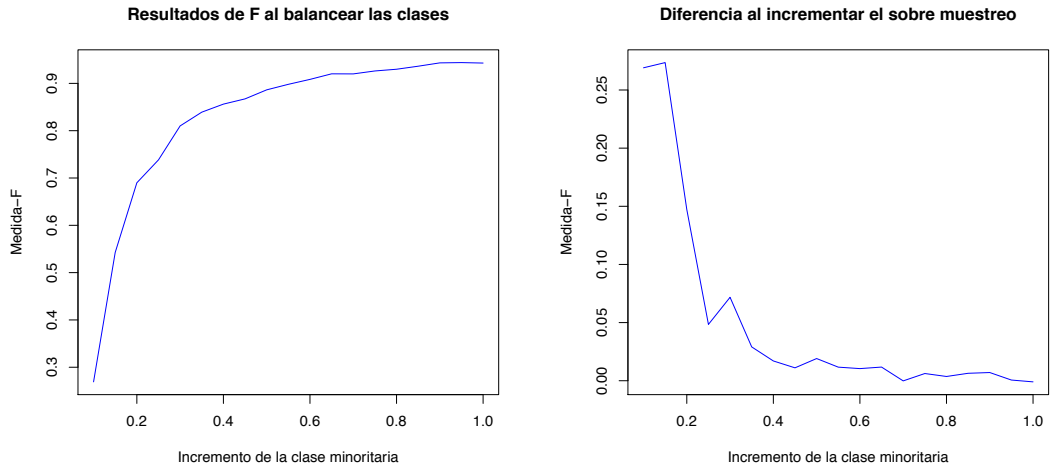
$s = N_{rm}/N_M$, N_{rm} es el número de instancias de la clase minoritaria y N_M es el número de instancias de la clase mayoritaria. Los detalles de este método se pueden ver en el capítulo 2; después, se desarrolla la técnica de selección y construcción de características, y por último se aplica la clasificación con bosque aleatorio.

La técnica de SMOTE incrementa los resultados de la clasificación en la clase minoritaria, a medida que se incrementa el porcentaje de balanceo los resultados alcanzan una $F = 0.94$. En la figura 6.3 se muestran los resultados de la medida F al incrementar el porcentaje de balanceo; es apreciable que el resultado aumenta rápidamente. Después de un determinado número de instancias creadas sintéticamente, los resultados se estabilizan y la diferencia entre el resultado actual y el anterior cada vez es menor. Cuando el conjunto de datos se encuentra extremadamente desbalanceado, la cantidad de instancias sintéticas se vuelve excesiva. La Tabla 6.4 muestra los resultados de la cantidad de instancias generadas con diferentes porcentajes y su resultado de F para cada una de ellas; tan solo con un $s = 0.30$, la cantidad de instancias de la clase minoritaria es superior hasta a un 200 % de la clase minoritaria original.

Como parte de los objetivos, para demostrar la efectividad del método de selección y construcción de características, se estableció seleccionar un porcentaje bajo de sobre muestreo que ayude a disminuir el problema del desbalance de clases, en otras palabras, utilizar la menor cantidad de instancias sintéticas para crear el modelo de predicción.

Configuración del experimento 2

1. grupos = [6, 4, 5, 6, 15, 7]
2. operaciones = $\{+, -, *, \text{máx}, \text{mín}\}$
3. tamaño de la población $(t) = 50$
4. probabilidad de combinar $(p_{mz}) = 0.5$



(a) 2015 **(b)** 2015

Figura 6.3: Resultados de F al incrementar la clase minoritaria con SMOTE.

Tabla 6.4:

Incremento de la clase minoritaria

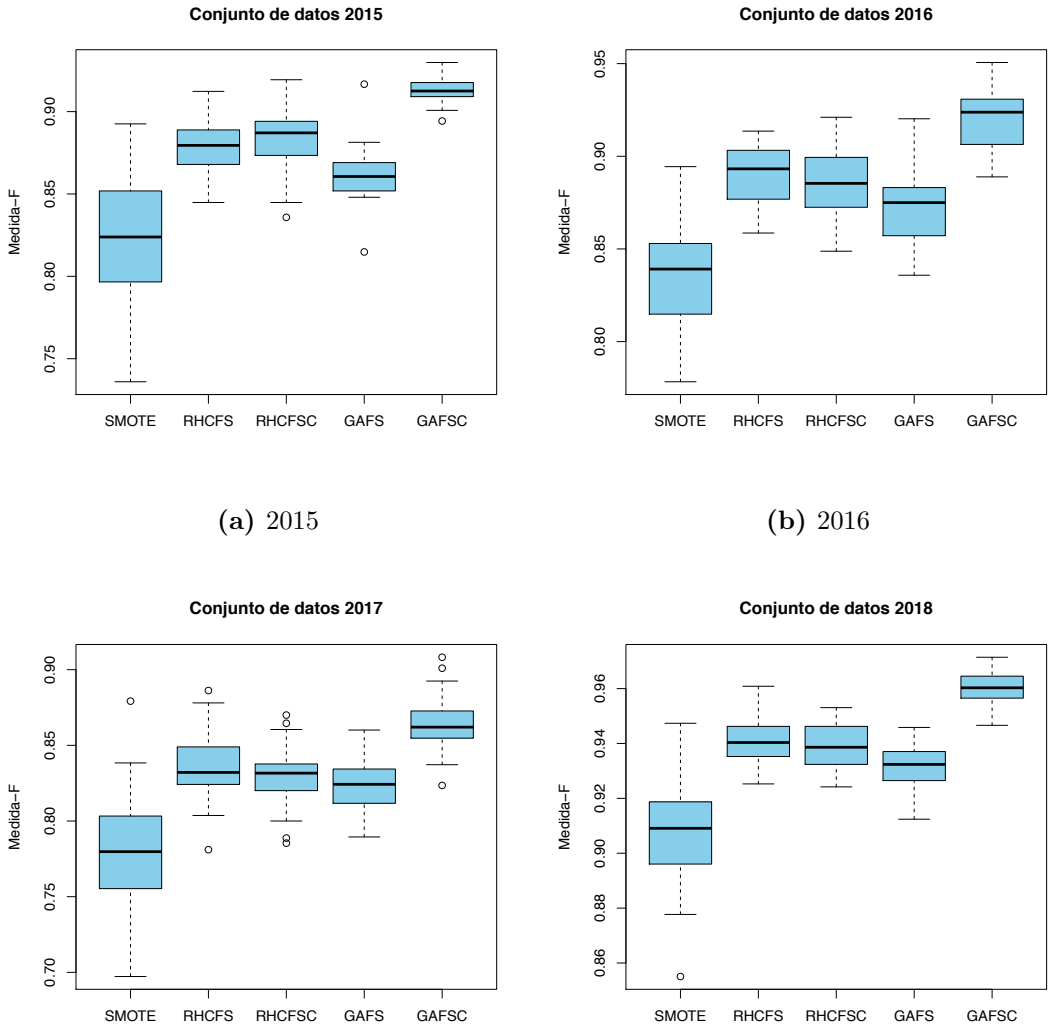
Porcentaje de sobre muestreo (s)	Medida F	Instancias sintéticas
0.1	0.164	9
0.2	0.664	116
0.3	0.776	224
0.4	0.854	331
0.5	0.876	439
0.6	0.906	547
0.7	0.922	654
0.8	0.927	762
0.9	0.948	869
1	0.943	977

5. probabilidad de mutar (p_m) = 0.1
6. iteraciones (n) = 200
7. corridas = 30
8. $s = 0.3$

Configuración del experimento 3

1. grupos = [6, 4, 5, 6, 15, 7]
2. operaciones = $\{+, -, *, \text{máx}, \text{mín}\}$
3. tamaño de la población (t) = 50
4. probabilidad de combinar (p_{mz}) = 0.5
5. probabilidad de mutar (p_m) = 0.1
6. iteraciones (n) = 200
7. corridas = 30
8. $s = 0.4$

Después de aplicar SMOTE, cuando $s = 0.3$, $F = 0.8364$ y para $s = 0.4$, el valor promedio de $F = 0.8859$ para el conjunto de datos completo. La comparación en la Tabla 6.5 muestra los resultados de los conjuntos de datos con sobre muestreo, a los cuales se les aplicó el proceso de selección y construcción de características con los diferentes métodos obteniendo un resultado en el método propuesto promedio de $F = 0.9149$ y $F = 0.9433$ para los parámetros de $s = 0.3$ y $s = 0.4$ respectivamente. El incremento provocado por la disminución de la dimensionalidad es de $F = 0.0785$ y $F = 0.0574$; además, en la figura 6.4 se puede apreciar la distribución de los resultados del método GAFSC es superior a los demás en los cuatro conjuntos de datos.



(a) 2015

(b) 2016

(c) 2017

(d) 2018

Figura 6.4: Experimento 2

Tabla 6.5:

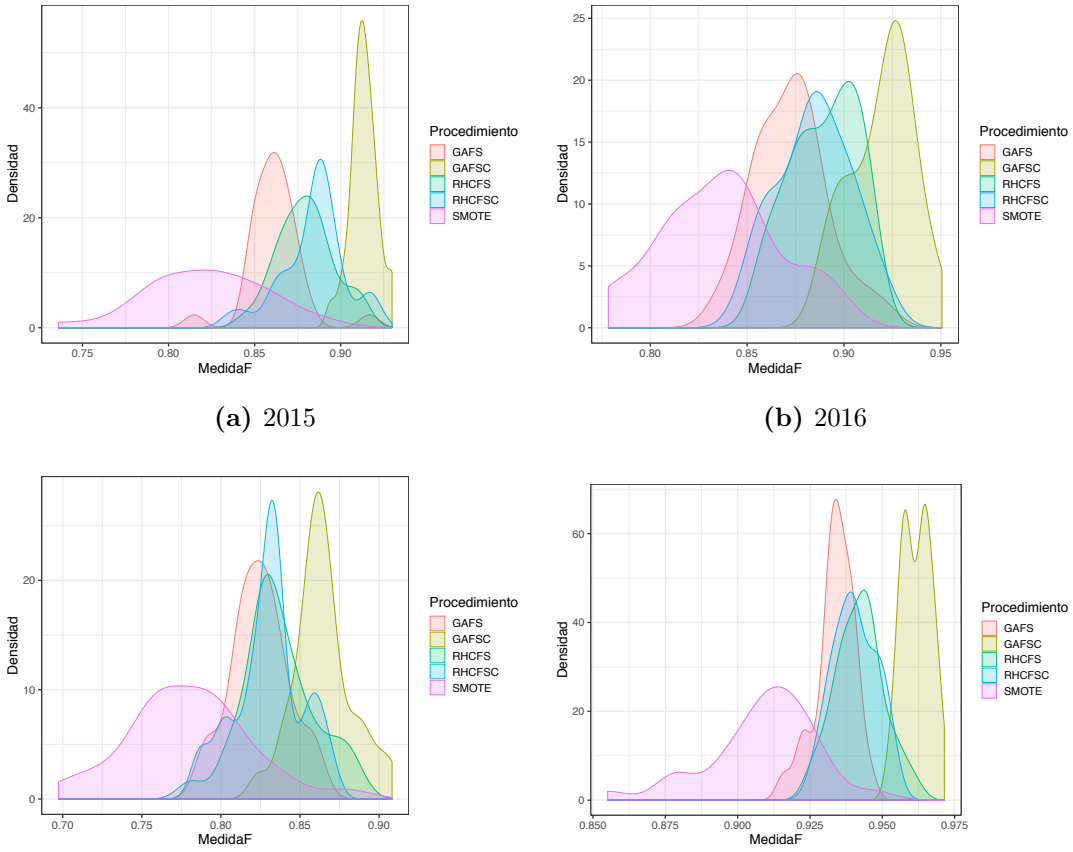
Resultados promedio de Experimento 2 y 3

Conjunto de Datos	s	Completo	RHCFS	RHCFSC	GAFS	GAFSC
2015	0.3	0.8219	0.8797	0.8842	0.8617	**0.9135**
	0.4	0.8706	0.9160	0.917	0.9039	**0.9425**
2016	0.3	0.8362	0.8898	0.8848	0.8727	**0.9198**
	0.4	0.89.60	0.9340	0.9323	0.9218	**0.9563**
2017	0.3	0.7789	0.8365	0.8302	0.8246	**0.8646**
	0.4	0.8451	0.8806	0.8760	0.8694	**0.9055**
2018	0.3	0.9087	0.9421	0.9405	0.9337	**0.9619**
	0.4	0.9318	0.9544	0.9511	0.9458	**0.9689**
Promedio	0.3	0.8364	0.8870	0.8839	0.8732	**0.9149**
	0.4	0.8859	0.9213	0.9192	0.9103	**0.9433**

6.2.3 Prueba de validez estadística

En este aparatado realizamos una prueba de t-student para demostrar que existe diferencia significativa. Considerando $\mu = \frac{\sum f-\text{FSCGA}}{n}$ y $\mu_0 = \frac{\sum f-\text{competidor}}{n}$, se tienen las hipótesis $H_0 : \mu = \mu_0$ y $H_1 : \mu > \mu_0$. En la figura 6.5 se muestra que los datos se distribuyen de forma normal en cada uno de los métodos, además la Tabla 6.6 muestra los resultados de la prueba de Shapiro Wilk, en la cual indica que existe normalidad en los datos para cada uno de los métodos.

En todos los casos el valor de P es mayor que el α de 0.05, con lo cual se comprueba la normalidad de los datos. Además, se aplicó una prueba de varianza a las muestras, en la cual se puede apreciar que las varianzas de las pruebas son significativamente iguales solo en algunos casos. Por lo tanto, la prueba t para comparar la igualdad de las medias utilizando para varianzas iguales o desiguales según la Tabla 6.7.



(a) 2015 **(b)** 2016

(c) 2017 **(d)** 2018

Figura 6.5: Experimento 2: Prueba de validez estadística

Tabla 6.6:

Resultados del valor de P en la Prueba de Shapiro Wilk en el Experimento 2

Conjunto de Datos	Completo	RHCFS	RHCFSC	GAFS	GAFSC
2015	0.9617	0.9029	0.129	0.1453	0.8288
2016	0.6342	0.0947	0.6337	0.7117	0.3915
2017	0.918	0.6523	0.4409	0.6436	0.5913
2018	0.5998	0.8116	0.2594	0.1404	0.2564

Tabla 6.7:

Prueba de F para varianzas de dos muestras $\alpha = 0.01$ *Experimento 2*

Conjunto de Datos	Completo/GAFSC	RHCFS/GAFSC	RHCFSC/GAFSC	GAFS/GAFSC
2015	$2.84 \cdot 10^{-12}$	$9.54 \cdot 10^{-5}$	$4.62 \cdot 10^{-6}$	$1.02 \cdot 10^{-4}$
2016	$3.65 \cdot 10^{-4}$	0.3808	0.1911	0.2010
2017	$8.95 \cdot 10^{-5}$	0.1028	0.2199	0.4632
2018	$7.13 \cdot 10^{-11}$	0.0027	0.0141	0.0500

Tabla 6.8:

Prueba de t comparar medias $\alpha = 0.01$ *Experimento 2*

Conjunto de Datos	Completo/GAFSC	RHCFS/GAFSC	RHCFSC/GAFSC	GAFS/GAFSC
2015	$8.71 \cdot 10^{-16}$	$2.33 \cdot 10^{-13}$	$7.13 \cdot 10^{-10}$	$1.4 \cdot 10^{-19}$
2016	$5.2 \cdot 10^{-17}$	$1.7 \cdot 10^{-9}$	$1.16 \cdot 10^{-10}$	$4.35 \cdot 10^{-15}$
2017	$2.65 \cdot 10^{-14}$	$1.49 \cdot 10^{-6}$	$4.91 \cdot 10^{-9}$	$4.75 \cdot 10^{-12}$
2018	$1.35 \cdot 10^{-16}$	$3.15 \cdot 10^{-15}$	$3.21 \cdot 10^{-18}$	$2.01 \cdot 10^{-19}$

Por último, los resultados de la prueba t-student presentados en la Tabla 6.8 demuestra la comparación del método GAFSC contra los demás métodos obteniendo para cada uno de ellos diferencia significativa. Por esta razón, se rechaza H_0, es decir, no existe evidencia suficiente para demostrar que las medias son iguales. En consecuencia, la media de GAFSC es mayor que los demás métodos.

6.3 Discusión

El Experimento 1 consistió en la aplicación del método de selección y construcción de características a los Conjuntos 2015, 2016, 2017 y 2018 sin resolver el problema de desbalance de clases con el objetivo de analizar el comportamiento del método en el conjunto real. Podemos notar que el problema de las clases desbalanceadas afecta directamente la clasificación de la clase minoritaria de tal manera que los resultados del clasificador para su conjunto completo es muy cercano a cero en la medida F. En el experimento se comprobó la hipótesis: al reducir la dimensionalidad

del conjunto de datos con el método propuesto los resultados en la clasificación aumentan, y el algoritmo genético por bloques obtiene un resultado superior al método del alpinista—incluso mejores que el algoritmo genético desarrollado solo para selección de atributos. En promedio, los conjuntos de datos logran una $F = 0.0803$ y la propuesta GAFSC una $F = 0.4318$, logrando así un incremento de 0.3515. Por otra parte, el algoritmo del alpinista para selección y construcción de características obtuvo un resultado de $F = 0.2592$ el cual fue superado por 0.1726. En otras palabras, el método propuesto identifica un subconjunto de los datos originales y otro subconjunto creado a partir de la combinación de atributos, que en combinación se adaptan bien para obtener una mejor clasificación.

Las características del método propuesto en comparación a los otros métodos son las siguientes: la selección se realiza con la combinación del método de ruleta y torneo, el cual, selecciona de manera probabilística la población dando mayor oportunidad a los mejores individuos en cada generación para iniciar el proceso de torneo, por consecuente, los mejores individuos pasan al proceso de cruza y mutación. Al utilizar como base un algoritmo genético, este se encarga de decidir la cantidad de atributos a utilizar como parte de su proceso, por lo que no se requiere definir previamente el número de atributos de salida deseados, además de ser más eficiente en tiempo computacional que utilizar un método exhaustivo.

Debido a que aún y cuando se sabe que los datos de la clase mayoritaria se clasificarán de manera correcta, la mayoría de los datos de la clase minoritaria no. Por lo anterior, en el Experimento 2 y 3, se utilizó la técnica de SMOTE. Se aplicó sobre muestreo en el conjunto de datos; por lo tanto, la clase minoritaria tiende a obtener una mejor clasificación. Al ingresar instancias sintéticas en el proceso de clasificación, algunas de ellas se utilizan para entrenamiento y otras para prueba, de tal forma que los espacios poco poblados de la clase minoritaria aumentan su relevancia para el clasificador. De esta manera, a medida que se aumenta la cantidad de instancias sintéticas hasta igualar la cantidad de datos de la clase mayoritaria, aumenta la probabilidad de obtener mejores resultados; sin embargo, una sobre po-

blación de instancias sintéticas puede provocar un sobre entrenamiento. El riesgo se incrementa cuando el nivel de desbalance es grande (IR > 9); por lo tanto, se realizó sobre muestreo hasta completar un porcentaje de instancias de la clase minoritaria utilizando $s = 0.3$ y $s = 0.4$, es decir, se crearon las instancias sintéticas de la clase minoritaria suficientes para que la proporción entre la clase mayoritaria y la minoritaria sea de un 30/100 y 40/100 con la finalidad de no saturar el conjunto de instancias sintéticas.

Los resultados de este experimento muestran que aún y cuando la técnica de SMOTE aumenta significativamente los resultados, la propuesta presentada identifica las mejores combinaciones de atributos presentando un subconjunto de datos que mejor se adapta. La cantidad de atributos en los cromosomas ganadores ronda entre 20 y 39 atributos y con una media de 30 atributos. Sin embargo, en promedio 20 atributos se utilizan de manera individual y 7 atributos se utilizan para realizar combinaciones.

6.4 Comprobación de hipótesis

La prueba de validez estadística se realizó para el experimento 2 con una muestra de 30 resultados para cada conjunto de datos. Se efectuó la prueba Shapiro Wilk para comprobar la normalidad de los datos, después se realizó la prueba de igualdad de varianzas y por último la prueba t de student. Los resultados indicaron que existe diferencia significativa tomando un $\alpha = 0.01$ al comparar el método propuesto (GAFSC) contra los demás métodos, lo que demuestra la efectividad del método en la búsqueda de las características más relevantes y las combinaciones con mayor interacción entre ellas.

6.5 Resumen

Se presentó una propuesta para identificar un subconjunto de datos originales y otro subconjunto de datos creados a partir de los originales, que en conjunto puedan incrementar los resultados de la clasificación para casos en los cuales, la cantidad de características del conjunto de datos es suficiente para formar grupos o categorías y el uso de todas las características provoca resultados bajos en la clasificación. La propuesta toma el conjunto original y mediante un algoritmo genético crea un conjunto de posibles soluciones, las cuales son evaluadas con un clasificador. Las pruebas realizadas muestran que es posible identificar un subconjunto con mejor ajuste.

Los resultados de la propuesta se compararon con los resultados de la clasificación utilizando todas las características, aplicando selección de características con método del alpinista y algoritmo genético y aplicando selección y construcción de características con el método del alpinista. En todos los casos se identificó diferencia significativa en el método propuesto.

Capítulo 7

Conclusiones y Trabajo futuro

Este capítulo tiene como objetivo el dar un cierre al presente trabajo, así como brindar posibles mejoras y extensiones que pudieran abordarse en un futuro. A manera de cierre, se provee un resumen con los puntos más sobresalientes de la tesis, se exponen comentarios finales, se da respuesta a las preguntas de investigación, se enumeran las contribuciones de la tesis y se describen diferentes aplicaciones posibles de este trabajo. Asimismo, como ya se mencionó, se discuten algunas líneas de trabajo futuro.

7.1 Resumen

Se presentó un algoritmo genético por bloques para selección y construcción de características, el cual permitiera mejorar la calidad de clasificación en conjuntos de datos donde las características estuviesen conformadas por grupos de forma anticipada. En el algoritmo genético por bloques, cada cromosoma se subdividió en grupos de características (bloques), que a su vez fueron divididos en cuatro secciones de genes (donde cada gen representó una característica): individuales apagados, individuales prendidos, compuestos apagados y compuestos prendidos. Los genes compuestos representan el proceso de construcción y se manejaron a través de árboles de operaciones con notación posfija. En cuanto al algoritmo genético en sí, la

población inicial, la evaluación, la cruza y la mutación fueron también adaptadas para hacerse por bloques.

Los conjuntos de datos utilizados para probar esta propuesta correspondieron a un caso de estudio de una escuela de nivel medio superior que incluye educación técnica. Estos conjuntos de datos fueron tomados de (1) encuestas desarrollas por la institución por su departamento interno de orientación psicológica durante los años del 2015 a 2018 y (2) de un examen de conocimientos de dividido en 6 áreas, conformando un total de 43 características para cada ejemplo, las cuales se dividieron en 6 grupos: 6 características para el examen de conocimientos, 4 correspondientes a núcleo familiar, 5 para hábitos de salud, 6 para hábitos de alimentación, 15 para hábitos de estudio y percepción de sí mismo y 7 para aspectos económicos.

El resultado que se logró al aplicar el algoritmo genético muestra un incremento la calidad de la clasificación (también comparado contra otros métodos), pero es también necesario abordar el problema de las clases desbalanceadas. Por este motivo, se aplicó generación de instancias sintéticas utilizando la técnica SMOTE. Utilizando en promedio 20 características de manera individual y 7 características combinadas, ya considerando balanceo de clases, se obtuvo una medida F de 0.94. Como se vio en el capítulo 6, la diferencia en comparación con métodos rivales fue significativa.

Este trabajo se distingue de otros trabajos similares en tres aspectos principales. El primer aspecto es el trabajar por grupos de características, lo que hace que las combinaciones de características (construcción) solo se realicen en el grupo y no se combinen características que pertenecen a un fenómeno de estudio diferente. El segundo aspecto es el uso exclusivo de las características, es decir, cada característica se utiliza solo una vez para selección o para construcción, pero no en ambas. Consideramos que esto reduce el espacio de búsqueda, evitar la sobre-explotación de características y evitar también el aumento de la cantidad de características (más bien lo que interesa es la reducción). El tercer aspecto es el diseño del cromosoma propuesto que combina los grupos de características y las secciones de individua-

les y compuestos con el objetivo de realizar un solo proceso para la selección y la construcción de características.

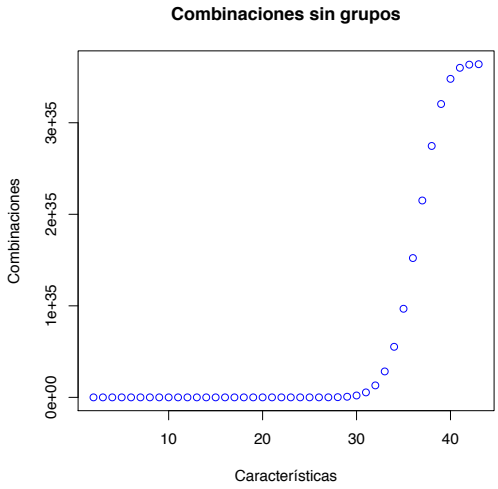
En este sentido es importante resaltar que en el caso de la selección de características el espacio de búsqueda no es reducido a través de la conformación de los grupos de características, ya que la cantidad de combinaciones posibles con o sin grupos es la misma, sin embargo, En las Gráficas 7.1 se puede apreciar que la construcción de características si se ve afectada por la conformación de bloques, ya que solo se busca la combinación entre los miembros del grupo, a medida que se conforman grupos más pequeños se puede reducir el espacio de búsqueda.

7.2 Comentarios finales

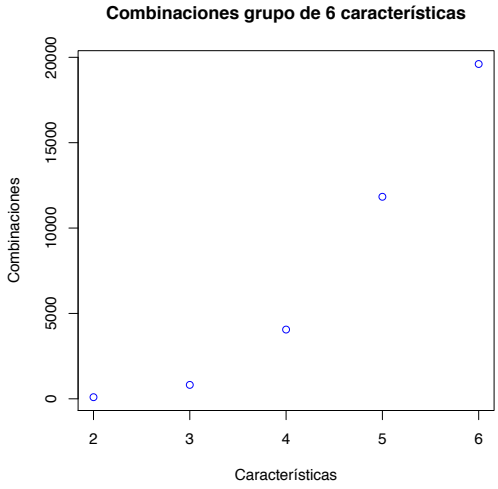
La propuesta de combinar la selección y la construcción en un solo proceso utilizando un algoritmo genético por bloques presentó una diferencia significativa a comparación de utilizar el conjunto de datos completo (con todas las características) y de utilizar otros métodos, como el de alpinista — tanto para selección como para construcción de características. Consideramos que esta es una aportación al estado del arte.

Los cromosomas de salida permiten ver las características que de manera individual o en combinación con otras obtuvieron el mejor desempeño entre todas las posibilidades exploradas. Consideramos que este tipo de salida es claro, pues el usuario del método puede apreciar sin dificultad qué características se van a utilizar y cómo.

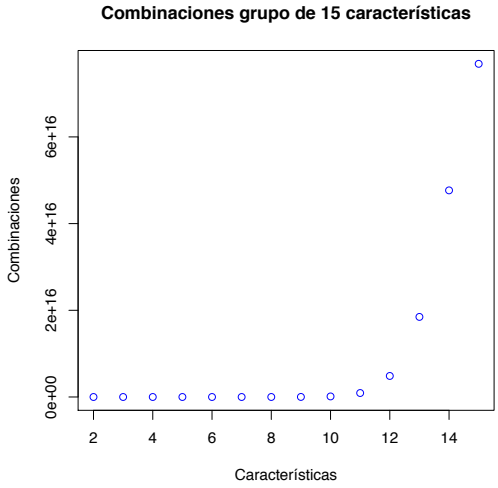
En general, los métodos de envoltura tienen un alto costo computacional, así que al aumentar el tamaño de la población inicial, implica un aumento en este tiempo. Para mitigar este problema, existen algunas técnicas presentadas en la revisión de la literatura que proponen el trabajo con máquinas conectadas en paralelo, aunque esto no asegura la mejora en los resultados.



(a) Sin Grupos



(b) Grupo de 15 Características



(c) Grupo de 6 Características

Figura 7.1: Combinaciones de caracteristicas

Es importante ver que se pueden tomar más datos de los estudiantes que puedan ayudar a mejorar la clasificación. En la revisión de la literatura se muestra una propuesta en la cual la clasificación se hace en diferentes momentos, antes del inicio de clases, y otras durante el periodo académico con información reciente.

7.3 Respuesta a las preguntas de investigación

1. *¿Es posible realizar la selección y construcción por bloques independientes cuando las características conforman grupos de características similares?*

   Sí es posible. Como se vio en el capítulo 4, el algoritmo genético por bloques inicia con la generación de una población inicial identificando de manera aleatoria una cantidad establecida de cromosomas, donde cada cromosoma creado identifica para cada grupo las características individuales y las combinadas; las combinaciones se realizan con una estructura similar a la programación genética, ya que a través de una estructura de árbol binario se crea una combinación de las características y guardadas en el cromosoma con notación postfija. Después se desarrolla el proceso de evaluación mediante un clasificador (en este caso se seleccionó el método de bosque aleatorio). La selección se realizó mediante una combinación del método de ruleta y torneo; de esta manera se selecciona a los cromosomas que pasan a cruza, la cual en términos básicos es un intercambio de bloques. El proceso de mutación, aún y cuando la probabilidad es baja, permite crear los cambios entre la selección y construcción de características dentro de cada grupo.

2. *¿Es posible representar en un genotipo (cromosoma), la información de la selección y la construcción de características para cada uno de los grupos (bloques) de características?*

   Sí es posible. En general el cromosoma representa el proceso de selección y construcción. En nuestro caso, el proceso de selección está representado por

las secciones de *individuales apagados* (I_{ap}) e *individuales prendidos* (I_{pre}) y el proceso de construcción está representado por las secciones de *compuestos apagados* (C_{ap}) y *compuestos prendidos* (C_{pre}).

3. *¿Puede encontrarse mediante un algoritmo genético un nuevo conjunto de características con mejor calidad que el conjunto original?*

   Sí. Como se mostró en el capítulo 6, el algoritmo genético por bloques fue capaz de encontrar conjuntos de datos con menos características (27, por ejemplo) que el conjunto original (43) y con una mejor calidad de clasificación.

7.4 Contribuciones

En resumen, nuestras aportaciones fueron las siguientes:

1. Se presentó una técnica de selección y construcción de características para mejorar la clasificación para conjuntos de datos en los que es posible agrupar características. El proceso toma cada grupo e identifica las características individuales y de existir, las combinaciones de ellas que ayudan a predecir mejor un objetivo.

2. El proceso evolutivo presentado permitió desarrollar tanto la selección como la construcción de características en un solo proceso, mediante un cromosoma dividido en grupos de características y para cada grupo en cuatro secciones, dos para características individuales y dos para características combinadas.

3. Se aportó el diseño de un cromosoma conformado por bloques y dentro de cada bloque cuatro secciones que agrupan las características del conjunto de datos.

4. Se aportó a la dependencia educativa una técnica para identificar estudiantes en riesgo de abandonar sus estudios de nivel medio superior, la cual puede

apoyar en la preparación de estrategias preventivas para confirmar el riesgo y tomar acciones oportunas.

7.5 Posibles aplicaciones

Aún y cuando el diseño se realizó bajo el caso de estudio de una escuela de nivel medio superior con educación técnica en el país de México en la modalidad presencial, la aplicación de este método puede ser extenderse para la modalidad no presencial o mixta, así como a distintos sectores educativos de nivel superior. Inclusive, puede ser aplicado para cualquier conjunto de datos que cumpla las siguientes características:

1. Clasificarse de manera binaria.
2. Tener al menos dos grupos de características identificadas previamente.
3. Tener baja calidad en la clasificación.

7.6 Trabajo futuro

Como trabajo futuro se contemplan varios escenarios. El primero de ellos es el incorporar en el proceso de cruza un método para que el cambio no se realice por bloques sino por características, ya que cuando los grupos contienen una gran cantidad de características el cambio entre un cromosoma y otro puede disminuir la calidad de la clasificación. El segundo es contemplar la posibilidad de trabajar con características discretas, ya que el proceso de la combinación de características se realiza solo mediante operaciones matemáticas. Se han propuesto algunas técnicas de agrupamiento de características, aunque estas trabajan bajo la identificación de similitud de los datos y no de un conocimiento previo de ellos con respecto a su naturaleza. Por tanto, un tercer trabajo futuro podría ser incorporar una técnica

para identificar los grupos de características cuando no se tenga este conocimiento previo. Al ser un proceso de selección y construcción de tipo envoltura el tiempo computacional requerido para obtener la respuesta se ve afectado en comparación a los métodos de filtro; sin embargo, este se ve afectado por los valores iniciales del proceso evolutivo, principalmente el número máximo de generaciones y el tamaño de la población inicial, encontrar los valores más adecuados para obtener un equilibrio entre eficiencia y tiempo computación es algo que es necesario investigar con mayor profundidad.

Referencias

Abiodun, O. I., Jantan, A., Omolara, A. E., Dada, K. V., Mohamed, N. A., y Arshad, H. (2018). State-of-the-art in artificial neural network applications: A survey. *Heliyon*, 4(11):e00938. https://doi.org/10.1016/j.heliyon.2018.e00938.

Abuteir, M. y El-Halees, A. (2012). Mining educational data to improve students' performance: a case study. *International journal of information and communication technology research*, 2(2):140–146.

Ahmed, S., Zhang, M., Peng, L., y Xue, B. (2014). Multiple feature construction for effective biomarker identification and classification using genetic programming. In *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation*, GECCO '14, pages 249–256, New York, NY, USA. Association for Computing Machinery. https://doi.org/10.1145/2576768.2598292.

Amrieh, E. A., Hamtini, T., y Aljarah, I. (2016). Mining educational data to predict student's academic performance using ensemble methods. *International Journal of Database Theory and Application*, 9(8):119–136. http://dx.doi.org/10.14257/ijdta.2016.9.8.13.

Asanbe, M., Osofisan, A., y William, F. (2016). Teachers' performance evaluation in higher educational institution using data mining technique. *International Journals of Applied Information System (IJAIS)*, 10(7):10–15. http://dx.doi.org/10.5120/ijais2016451524.

Bach, M., Werner, A., Żywiec, J., y Pluskiewicz, W. (2017). The study

of under- and over-sampling methods' utility in analysis of highly imbalanced data on osteoporosis. *Information Sciences*, 384:174 – 190. 20 citas, https://doi.org/10.1016/j.ins.2016.09.038.

Baker, R. S. y Yacef, K. (2009). The state of educational data mining in 2009: A review and future visions. *JEDM— Journal of Educational Data Mining*, 1(1):3–17. https://doi.org/10.5281/zenodo.3554657.

Bakhshinategh, B., Zaiane, O. R., ElAtia, S., y Ipperciel, D. (2018). Educational data mining applications and tasks: A survey of the last 10 years. *Education and Information Technologies*, 23(1):537–553. https://doi.org/10.1007/s10639-017-9616-z.

Bolón-Canedo, V., Sánchez-Maroño, N., y Alonso-Betanzos, A. (2015). *Feature Selection for High-Dimensional Data*. Springer International Publishing. 10.1007/978-3-319-21858-8.

Branco, P., Torgo, L., y Ribeiro, R. P. (2016). A survey of predictive modeling on imbalanced domains. *ACM Comput. Surv.*, 49(2):31:1–31:50. https://doi.org/10.1145/2907070.

Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

Breiman, L., Friedman, J., Stone, C. J., y Olshen, R. A. (1984). *Classification and regression trees*. CRC press.

Chandrashekar, G. y Sahin, F. (2014). A survey on feature selection methods. *Computers & Electrical Engineering*, 40(1):16 – 28. 40th-year commemorative issue, https://doi.org/10.1016/j.compeleceng.2013.11.024.

Chawla, N. V., Bowyer, K. W., Hall, L. O., y Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357. 5, https://doi.org/10.1613/jair.953.

Cheng, H., Shan, J., Ju, W., Guo, Y., y Zhang, L. (2010). Automated breast cancer detection and classification using ultrasound images: A survey. *Pattern Recognition*, 43(1):299 – 317. https://doi.org/10.1016/j.patcog.2009.05.012.

Cohen, G., Hilario, M., Sax, H., Hugonnet, S., y Geissbuhler, A. (2006). Learning from imbalanced data in surveillance of nosocomial infection. *Artificial Intelligence in Medicine*, 37(1):7 – 18. Intelligent Data Analysis in Medicine, https://doi.org/10.1016/j.artmed.2005.03.002.

Dai, Y., Xue, B., y Zhang, M. (2014). New Representations in PSO for Feature Construction in Classification. In Esparcia-Alcázar, A. I. y Mora, A. M., editors, *Applications of Evolutionary Computation*, pages 476–488, Berlin, Heidelberg. Springer Berlin Heidelberg.

Deb, K., Pratap, A., Agarwal, S., y Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197. https://doi.org/10.1109/4235.996017.

Dua, D. y Graff, C. (2017). Uci machine learning repository.

Emmanouilidis, C., Hunter, A., y MacIntyre, J. (2000). A multiobjective evolutionary setting for feature selection and a commonality-based crossover operator. In *Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No.00TH8512)*, volume 1, pages 309–316 vol.1. doi: 10.1109/CEC.2000.870311.

Fernández, A., Garcia, S., Herrera, F., y Chawla, N. V. (2018). Smote for learning from imbalanced data: Progress and challenges, marking the 15-year anniversary. *Journal of Artificial Intelligence Research*, 61:863–905. https://doi.org/10.1613/jair.1.11192.

García, S. y Herrera, F. (2009). Evolutionary undersampling for classification with imbalanced datasets: Proposals and taxonomy. *Evolutionary Computation*, 17(3):275–306. https://doi.org/10.1162/evco.2009.17.3.275.

Haberman, S. J. (1976). Generalized residuals for log-linear models. In *Proceedings of the 9th international biometrics conference*, pages 104–122.

Hyun-Jung, K., Nam-Ok, J., y Kyung-Shik, S. (2016). Optimization of cluster-based evolutionary undersampling for the artificial neural networks in corpo-

rate bankruptcy prediction. *Expert Systems with Applications*, 59:226 – 234. https://doi.org/10.1016/j.eswa.2016.04.027.

Kotsiantis, S. (2009). Educational data mining: a case study for predicting dropout-prone students. *International Journal of Knowledge Engineering and Soft Data Paradigms*, 1(2):101–111. https://doi.org/10.1504/IJKESDP.2009.022718.

Kotsiantis, S., Pierrakeas, C., y Pintelas, P. (2004). Predicting students' performance in distance learning using machine learning techniques. *Applied Artificial Intelligence*, 18(5):411–426. https://doi.org/10.1080/08839510490442058.

Krawiec, K. (2002). Genetic programming-based construction of features for machine learning and knowledge discovery tasks. *Genetic Programming and Evolvable Machines*, 3(4):329–343. https://doi.org/10.1023/A:1020984725014.

Kubat, M., Holte, R. C., y Matwin, S. (1998). Machine learning for the detection of oil spills in satellite radar images. *Machine Learning*, 30(2):195–215. https://doi.org/10.1023/A:1020984725014.

Li, X. (2003). A non-dominated sorting particle swarm optimizer for multiobjective optimization. In *Genetic and Evolutionary Computation — GECCO 2003*, pages 37–48, Berlin, Heidelberg. Springer Berlin Heidelberg.

Lin, F., Liang, D., Yeh, C.-C., y Huang, J.-C. (2014). Novel feature selection methods to financial distress prediction. *Expert Systems with Applications*, 41(5):2472 – 2483. https://doi.org/10.1016/j.eswa.2013.09.047.

Mahanipour, A. y Nezamabadi-pour, H. (2017). Improved pso-based feature construction algorithm using feature selection methods. In *2017 2nd Conference on Swarm Intelligence and Evolutionary Computation (CSIEC)*, pages 1–5. https://doi.org/10.1109/CSIEC.2017.7940173.

Mani, I. y Zhang, I. (2003). kNN approach to unbalanced data distributions: a case study involving information extraction. In *Proceedings of workshop on learning from imbalanced datasets*, volume 126, pages 1–7.

Márquez-Vera, C., Cano, A., Romero, C., Noaman, A. Y. M., Mousa Fardoun, H., y Ventura, S. (2016). Early dropout prediction using data mining: a case study with high school students. *Expert Systems*, 33(1):107–124. https://doi.org/10.1111/exsy.12135.

Morales, Á. F. K., Aldana-Bobadilla, E., y Lopez-Peña, I. (2013). The best genetic algorithm i-a comparative study of structurally different genetic algorithms. In *Proceedings of the Mexican International Conference on Artificial Intelligence 2013*, pages 16–29.

Mueen, A., Zafar, B., y Manzoor, U. (2016). Modeling and predicting students' academic performance using data mining techniques. *International Journal of Modern Education and Computer Science*, 8(11):36. https://doi.org/10.5815/ijmecs.2016.11.05.

Müller, A. C. y Guido, S. (2016). *Introduction to machine learning with Python: a guide for data scientists*. O'Reilly Media, Inc.

Neapolitan, R. y Naimipour, K. (1998). *Foundations of algorithms using C++ pseudocode*. Jones and Bartlett Publishers International, Massachussetts, Estados Unidos, segunda edition.

Nguyen, H. B., Xue, B., y Andreae, P. (2017). A Hybrid GA-GP Method for Feature Reduction in Classification. In Shi, Y., Tan, K. C., Zhang, M., Tang, K., Li, X., Zhang, Q., Tan, Y., Middendorf, M., y Jin, Y., editors, *Simulated Evolution and Learning*, pages 591–604, Cham. Springer International Publishing. https://doi.org/10.1007/978-3-319-68759-9_48.

Quinlan, J. (2014). *C4.5: Programs for machine learning*. Ebrary online. Elsevier Science.

Quinlan, J. R. (1987). Simplifying decision trees. *International journal of man-machine studies*, 27(3):221–234.

Ramaswami, M. y Bhaskaran, R. (2009). A study on feature selection techniques in educational data mining. *CoRR*, abs/0912.3924.

Romero, C. y Ventura, S. (2010). Educational data mining: A review of the state of the art. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 40(6):601–618. https://doi.org/10.1109/TSMCC.2010.2053532.

Russell, S. y Norvig, P. (2010). *Artificial Intelligence: A Modern Approach*. Prentice Hall, New Jersey, EUA, tercera edition.

Ryman-Tubb, N. F., Krause, P., y Garn, W. (2018). How artificial intelligence and machine learning research impacts payment card fraud detection: A survey and industry benchmark. *Engineering Applications of Artificial Intelligence*, 76:130 – 157. https://doi.org/10.1016/j.engappai.2018.07.008.

Saarela, M., Yener, B., Zaki, M., y Kärkkäinen, T. (2016). Predicting math performance from raw large-scale educational assessments data: a machine learning approach. In *JMLR Workshop and Conference Proceedings; 48*, pages 1–8. JMLR.

Sait, S. M. y Youssef, H. (1999). *Iterative computer algorithms: and their applications in engineering*. IEEE Computer Society, Los Alamitos, California.

Smith, M. G. y Bull, L. (2005). Genetic programming with a genetic algorithm for feature construction and selection. *Genetic Programming and Evolvable Machines*, 6(3):265–281. https://doi.org/10.1007/s10710-005-2988-7.

Soufan, O., Kleftogiannis, D., Kalnis, P., y Bajic, V. B. (2015). Dwfs: a wrapper feature selection tool based on a parallel genetic algorithm. *PloS one*, 10(2). https://doi.org/10.1371/journal.pone.0117988.

Starmer, J. (2018). StatQuest: Decision Trees. `https://www.youtube.com/watch?v=7VeUPuFGJHk`.

Tran, B., Xue, B., y Zhang, M. (2017). Using feature clustering for GP-based feature construction on high-dimensional data. In McDermott, J., Castelli, M., Sekanina, L., Haasdijk, E., y García-Sánchez, P., editors, *Genetic Programming*, pages 210–226, Cham. Springer International Publishing. https://doi.org/10.1007/978-3-319-55696-3_14.

Tran, B., Xue, B., y Zhang, M. (2019). Genetic programming for multiple-feature construction on high-dimensional classification. *Pattern Recognition*, 93:404 – 417. https://doi.org/10.1016/j.patcog.2019.05.006.

Tran, B., Zhang, M., y Xue, B. (2016). Multiple feature construction in classification on high-dimensional data using gp. In *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–8. https://doi.org/10.1109/SSCI.2016.7850130.

Tsoulos, I., Tzallas, A., y Tsalikakis, D. (2019). Genetic feature construction genetic feature construction: a parallel implementation of a genetic programming tool for feature construction. *European Journal of Engineering Research and Science*, 4(5):5–11. https://doi.org/10.24018/ejers.2019.4.5.1272.

Velmurugan, T. y Anuradha, C. (2016). Performance evaluation of feature selection algorithms in educational data mining. *Performance Evaluation*, 5(02).

Wang, L., Wang, Y., y Chang, Q. (2016). Feature selection methods for big data bioinformatics: A survey from the search perspective. *Methods*, 111:21 – 31. Big Data Bioinformatics, https://doi.org/10.1016/j.ymeth.2016.08.014.

Wook, M., Yahaya, Y. H., Wahab, N., Isa, M. R. M., Awang, N. F., y Seong, H. Y. (2009). Predicting NDUM student's academic performance using data mining techniques. In *2009 Second International Conference on Computer and Electrical Engineering*, volume 2, pages 357–361. https://doi.org/10.1109/ICCEE.2009.168.

Xue, B., Zhang, M., y Browne, W. N. (2013). Particle swarm optimization for

feature selection in classification: A multi-objective approach. *IEEE Transactions on Cybernetics*, 43(6):1656–1671. https://doi.org/10.1109/TSMCB.2012.2227469.

Xue, B., Zhang, M., Browne, W. N., y Yao, X. (2016). A survey on evolutionary computation approaches to feature selection. *IEEE Transactions on Evolutionary Computation*, 20(4):606–626. https://doi.org/10.1109/TEVC.2015.2504420.

Xue, B., Zhang, M., Dai, Y., y Browne, W. N. (2013). PSO for Feature Construction and Binary Classification. In *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation*, GECCO '13, page 137–144, New York, NY, USA. Association for Computing Machinery. https://doi.org/10.1145/2463372.2463376.

Zhang, Y., Gong, D., Hu, Y., y Zhang, W. (2015). Feature selection algorithm based on bare bones particle swarm optimization. *Neurocomputing*, 148:150 – 157. https://doi.org/10.1016/j.neucom.2012.09.049.

Resumen autobiográfico

Rafael Alfredo Cavazos Martínez

Candidato para obtener el grado de
Doctorado en Ingeniería
con Orientación en Tecnologías de la Información

Universidad Autónoma de Nuevo León
Facultad de Ingeniería Mecánica y Eléctrica

Tesis:
Selección y construcción de características agrupadas mediante un algoritmo genético por bloques

Nací en la ciudad de Monterrey Nuevo León, México en el año 1984. Obtuve el grado de Ingeniero Administrador de Sistemas en la Facultad de Ingeniería Mecánica y Eléctrica en el año 2008. Posteriormente en el 2014 obtuve el grado de Maestría en Ingeniería de la Información con Orientación en Inteligencia Artificial, trabajo en la Universidad Autónoma de Nuevo León desde el año 2004, en el año 2009 me incorporé a la planta docente en la academia de Matemáticas y al cuerpo docente del Bachillerato Técnico en Sistemas computacionales (actualmente Bachillerato Técnico en Programación WEB), con más de 10 años de experiencia como docente he participado en la elaboración de planes de estudio del área de formación

técnica como manuales, guías de aprendizaje, programas analíticos entre otras actividades, además desde el 2009 he trabajado en el área administrativa, conformando así experiencia en sistemas de gestión de calidad con énfasis en estadística educativa, áreas que me han permitido tener la experiencia de participar activamente en reconocimientos de calidad de la dependencia como el premio Nuevo León a la Competitividad, Premio COPARMEX a la excelencia en la educación Técnica, Premio Nacional a la Calidad y premio Iberoamericano a la calidad, mismos en los que la organización a la que laboro ha sido reconocida. La participación en los modelos de calidad por parte de la dependencia educativa, me motivo en buscar cada día más opciones para desarrollar herramientas que apoyen a la toma de decisiones, los proyectos de mejora me han llevado a buscar nuevos retos, entre ellos la incorporación de sistemas para la educación motivo por el cual decidí continuar mis estudios en la Facultad de Ingeniería Mecánica y Eléctrica en el Doctorado en Ingeniería con Orientación en Tecnologías de Información.