Técnicas de clasificación para la predicción de tarifas aéreas

Classification techniques for airfares prediction

Marco Antonio Barrón Ortiz

Directores: Dr. Sebastián Ventura Soto

Dr. José María Luna Ariza

Computación Avanzada, Energía y Plasma

Universidad de Córdoba

Esta tesis se presenta para optar al grado de

Doctor

Informática y Análisis Numérico Marzo de 2023

TITULO: *TECNICAS DE CLASIFICACIÓN PARA LA PREDICCION DE TARIFAS AEREAS*

AUTOR: *Marco Antonio Barrón Ortiz*

Campus de Rabanales
Ctra. Nacional IV, Km. 396 A
14071 Córdoba

https://www.uco.es/ucopress/index.php/es/
ucopress@uco.es

TÍTULO DE LA TESIS: Técnicas de clasificación para la predicción de tarifas aéreas

DOCTORANDO/A: Marco Antonio Barrón Ortiz

INFORME RAZONADO DEL/DE LOS DIRECTOR/ES DE LA TESIS

(se hará mención a la evolución y desarrollo de la tesis, así como a trabajos y publicaciones derivados de la misma).

La presente tesis doctoral ha tenido una evolución muy favorable en los últimos años, donde se ha publicado un artículo en una revista de IEEE, así como un artículo en un congreso internacional de reconocido prestigio. Es importante destacar que, por motivos personales y laborales del doctorando, los primeros años de la tesis no produjeron avances significativos. No obstante, a partir del curso 2020/2021 se comenzó a trabajar de manera notoria en la misma con la dificultad añadida de que el doctorando está residiendo en un país extranjero.

Tras mucho esfuerzo, se publicó en el 2022 un artículo en una revista de impacto en el que se propuso una metodología para generar características que permitiesen estudiar, de manera automática, precios competitivos para una aerolínea en base a sus competidores. Normalmente este proceso es manual y muy tedioso, por lo que la metodología propuesta supone un avance significativo en el campo. Además, se publicó un artículo en un congreso internacional para analizar descuentos dinámicos en el campo de las aerolíneas mediante el uso de técnicas de Subgroup Discovery.

En general, el informe de la tesis es muy favorable con dos artículos de alta calidad científica.

Por todo ello, se autoriza la presentación de la tesis doctoral.

Córdoba, 3 de marzo de 2023

Firma del/de los director/es

Firmado digitalmente por VENTURA SOTO SEBASTIAN EMILIO - 30510000V
Fecha: 2023.03.03 16:55:46 +01'00'

LUNA ARIZA JOSE MARIA - 30979388K

Firmado digitalmente por LUNA ARIZA JOSE MARIA - 30979388K
Fecha: 2023.03.03 10:39:00 +01'00'

Fdo.: Sebastián Ventura Soto Fdo.: José María Luna Ariza

Declaración

La memoria titulada “Técnicas de clasificación para la predicción de tarifas aéreas”, que presenta Marco Antonio Barrón Ortiz para optar al grado de Doctor, ha sido realizada dentro del programa de doctorado Computación Avanzada, Energía y Plasma del Departamento de Informática y Análisis Numérico de la Universidad de Córdoba bajo la dirección de los doctores Sebastián Ventura Soto y José María Luna Ariza cumpliendo, en su opinión, los requisitos exigidos a este tipo de trabajos.

Córdoba, Marzo de 2023

El Doctorando

Fdo: Marco Antonio Barrón Ortiz

El Director

Firmado digitalmente por VENTURA SOTO SEBASTIAN EMILIO - 30510000V
Fecha: 2023.03.03 16:55:46 +01'00'

Fdo: Dr. Sebastián Ventura Soto

El Director

LUNA ARIZA JOSE MARIA - 30979388K

Firmado digitalmente por LUNA ARIZA JOSE MARIA - 30979388K
Fecha: 2023.03.03 10:39:00 +01'00'

Fdo: Dr. José María Luna Ariza

Dedicada a la memoria de Yolanda Ortiz Del Río, mi madre.

Agradecimientos

A mis directores de tesis Dr. Sebastián Ventura Soto y Dr. José María Luna Ariza por todo el apoyo que me han brindado, por su disposición a aceptar mi propuesta de investigación, por lo cual tuvieron que adentrarse al mundo de las aerolíneas comerciales. Al Dr. Sebastián Ventura Soto le agradezco su paciencia, sobre todo porque en algún momento parecía más fácil tirar la toalla y pedirme que abandonáramos la pelea antes del doceavo round, por no abandonarme en esos momentos y seguir adelante. Al Dr. José María Luna Ariza porque desde que nos conocimos me brindó su apoyo, primero como un compañero y después porque tomó el reto de dirigir mi tesis. A mis compañeros del grupo KDIS, especialmente a Carlos Márquez Vera y a Hermes Robles Berumen por esas largas y tendidas pláticas acerca de este trabajo, de los cuales he aprendido mucho durante este tiempo. A mi padre, Antonio Barrón Corvera por apoyarme en todo momento a lo largo de mi vida. Finalmente, a mi esposa, Blanca Irene Macias Hiriartt que decidió convertirse en una nómada para emprender una vida juntos en diferentes continentes.

Abstract

This work is focused on the multi-factorial problems that commercial airlines face up, such as the pricing war and the creation of a dynamic discount table through the implementation of evolutionary algorithms and data mining methods. On the one hand, in the airline industry, the Revenue and Pricing teams generally spend a considerable amount of time analysing and interpreting the actions of their competitors. Most of the time the analysts have to use their analytical skills to create ad-hoc methods to interpret or find patterns in the fares. The use of automatic methodologies is key to reducing time and avoiding human errors. This thesis proposes a new methodology to predict, analyze and interpret airline fares which are capable of mimicking manual processes executed by pricing teams. A gene expression programming algorithm is proposed to mimic the manual process carried out by pricing teams by adding new features automatically. The algorithm can explore huge search spaces, which is a daunting process to be done manually as pricing teams do daily. A real scenario was considered in the experimental analysis by considering Air Canada fares in the period December 2019 to January 2020, corresponding to a travel period between December 2019 and April 2020. On the other hand, historically, airlines around the globe have used static pricing structures, which are constrained to discrete price points and there is limited segmentation between their guests. Because of these limitations and constraints, the necessity of novel methods to calculate the willingness to pay and identify potential guests whose propensity to book a flight will increase if they receive a discount to improve their sales is huge. Thus, This thesis proposes a novel methodology to identify interesting subgroups whose chance to book a flight increases if they receive an offer discount. This proposal includes a grammatically evolutionary feature selection algorithm to extract the best subgroups by analyzing the booking behaviour of historical passengers. A real case scenario was considered in the experimental analysis using private data from a commercial airline.

Resumen

Esta memoria de tesis se enfoca en los problemas multifactoriales a los que se enfrentan las aerolíneas comerciales como son la guerra de precios y la creación de una tabla dinámica de descuentos. Por un lado, dentro de la industria aérea, los equipos de precios y ganancias pasan una cantidad de tiempo considerable analizando e interpretando las acciones de sus competidores. La mayoría de las veces, estos analistas tienen que usar sus habilidades para realizar una serie de análisis *ad-hoc* que les permita interpretar o encontrar patrones en las tarifas aéreas. La implementación de metodologías automáticas es clave para reducir los tiempos y evitar errores humanos. Esta tesis propone una nueva metodología para predecir, analizar e interpretar las tarifas de las aerolíneas que es capaz de imitar los procesos manuales ejecutados por los equipos de fijación precios. Para enfrentar esta guerra de precios, se propone un algoritmo de programación de expresión genética que imita el proceso manual llevado a cabo por los equipos de analistas mediante la adición automática de nuevas características o atributos. Para demostrar la capacidad de la metodología, se consideró un escenario real utilizando tarifas publicadas por parte de la aerolínea denominada Air Canada durante el período de diciembre 2019 a enero 2020; correspondiente a un período de viajes entre los meses de diciembre 2019 y abril de 2020.

En segundo lugar, se aborda el problema de crear una tabla de ofertas dinámicas, debido a que, históricamente, las aerolíneas de todo el mundo han utilizado estructuras de precios estáticas, que están restringidas a puntos de precios discretos y existe una segmentación limitada entre sus pasajeros. Debido a estas limitaciones y restricciones, existe una enorme necesidad de métodos novedosos para calcular la disposición a pagar e identificar a los pasajeros potenciales, cuya probabilidad de reservar un vuelo aumenta si estos reciben un descuento con la finalidad de incrementar sus ganancias a través del incremento de las ventas de tarifas aéreas. Se propone un algoritmo de gramáticas evolutivas, el cual funciona como un selector de características para extraer los mejores subgrupos mediante el análisis del comportamiento de reservas que muestran los pasajeros. Se consideró un escenario real en el análisis experimental utilizando datos privados de una aerolínea comercial de talla mundial.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

En la actualidad, diversas industrias a nivel mundial están inmersas en una guerra de precios. Al igual que las aerolíneas, compañías como Amazon [7] y Walmart [75] cuentan con grupos de analistas especializados en el monitoreo de los precios de sus competidores. Para enfrentar esta guerra de precios, las técnicas de minería de datos (*Data Mining*, DM) están siendo utilizadas como técnicas avanzadas de análisis para la creación de estrategias de precios que vayan acorde a los precios existentes en el mercado [151]. A lo anterior, se le conoce como inteligencia de precios o monitoreo competitivo, el cual se refiere al conocimiento de las complejidades de los precios existentes en el mercado y su impacto en las empresas [7]. La inteligencia de precios y uso de las técnicas de DM son de especial importancia en la industria de la aviación comercial, pues a través de las mismas se pueden generar modelos de reglas altamente interpretables que provean una descripción precisa de los cambios que suceden en el mercado frecuentemente. Tal descripción puede permitir mejorar la estrategia de precios y minimizar la pérdida de ingresos debido a las acciones efectuadas por los competidores [172].

Cuando se habla de la industria de las aerolíneas, la fijación de precios se refiere al proceso de determinar las clases de tarifas, junto con diferentes productos, servicios y restricciones en un mercado de origen y destino. Cada tarifa que las aerolíneas publican en el mercado se le adjunta una clase específica, lo que en inglés se le conoce como *fare class*, la cual se puede identificar mediante un código de una sola letra. Debido a esto, existen algunos códigos de las tarifas que son estándar entre las aerolíneas, mientras que otros difieren de acuerdo a cada aerolínea. Para lograr una alta demanda en la práctica, las aerolíneas diferencian las clases de tarifas ofreciendo productos identificables con diferentes precios y calidad de servicio. De este modo, para poder establecer estas clases, las aerolíneas han aplicado una estructura de tarifas altamente diferenciada basada tanto en las comodidades del servicio como en las

restricciones de las mismas. Estas clases de tarifas o estructuras se conocen como tarifas tradicionales o restringidas debido a su uso extensivo de restricciones cada vez más severas sobre tarifas más bajas, las cuales son diseñadas para evitar desvíos en las estrategias de precios [131].

Durante este siglo, se observa un aumento a nivel mundial de nuevas aerolíneas de bajo costo, las cuales están implementando clases de tarifas menos restrictivas en comparación con las aerolíneas convencionales. Estas aerolíneas convencionales tienen que adaptar sus estrategias de precios para competir con estas nuevas modalidades de tarifas, creando una significativa guerra de precios, la cual se ha convertido en el estándar de esta industria [131]. Para enfrentar la guerra de precios, las aerolíneas cuentan con grupos de analistas de precios, los cuales deben clasificar las tarifas de sus competidores, que a su vez, deben de adaptarlas dentro de su propia estructura de clases de tarifas y liberarlas al público. Este proceso es clave para mantener los precios competitivos y proteger sus ganancias. Los analistas encargados de administrar estas estrategias de precios dedican un tiempo considerable analizando e interpretando las acciones efectuadas por sus competidores. Actualmente, existe una necesidad importante de herramientas comerciales que puedan brindar de forma automática una metodología capaz de entregar modelos de alta interpretación para facilitar esta tarea.

Es importante recalcar que tanto la cantidad de restricciones, como un alto número de tarifas por cada clase que se publican en el mercado, hacen que el proceso de análisis e interpretación de tarifas aéreas sea extremadamente complejo. Por tanto, se corre el riesgo de la perdida de identificación de tendencias y patrones significativos, llevando a conclusiones erróneas y/o una comprensión parcial de las acciones reales tomadas por otras aerolíneas. De esta manera, se hace necesaria la implementación de metodologías automatizadas de análisis de precios competitivos y la extracción de reglas interpretables mediante el uso de técnicas de análisis avanzadas. A través del empleo de estas técnicas, las aerolíneas pueden descubrir tendencias significativas en los precios y ciertas características específicas de manera oportuna. Además, la integración de un proceso de monitoreo automático libera tiempo y recursos valiosos, elimina posibles errores humanos, y proporciona información relevante y precisa.

Aunado a la guerra de precios que existe dentro de ámbito de la aviación comercial, las estrategias de precios convencionales se basan en reglas comerciales que están mal optimizadas y, en ocasiones, no responden a las condiciones de cambio que constantemente sufre el mercado [153]. Debido a esto, para complementar los esfuerzos que se realizan para optimizar las estrategias de precios, las aerolíneas comerciales han comenzado a enfocarse en

ofertas dinámicas de tarifas de vuelos y productos como son el equipaje, reserva anticipada de asientos, comidas, opciones de flexibilidad, así como contenido de terceros, como por ejemplo estacionamiento y seguros. Estas ofertas o cambios dinámicos están controlados por *bots* de precios, que son agentes de software que recopilan datos y usan algoritmos para ajustar los precios de acuerdo con las reglas comerciales. Por lo general, estas reglas comerciales tienen en cuenta aspectos tales como la ubicación del cliente, la hora del día, el día de la semana, el nivel de demanda y los precios de la competencia. En este sentido, gracias al uso del *big data* y el avance de las técnicas de DM, las reglas comerciales para los ajustes de precios se pueden hacer más granulares. Sin embargo, en la actualidad a pesar de los avances tecnológicos, dentro de la industria de la aviación la creación de estas ofertas es rudimentaria y se gestiona en procesos, organizaciones y sistemas de tecnologías de información separados [121]. Así, las aerolíneas necesitan implementar métodos efectivos de precios dinámicos (*Dynamic Pricing*, DP), los cuales se pueden definir como el enfoque para establecer el costo de un producto o servicio que es altamente flexible en tiempo real, teniendo como objetivo principal el permitir que una empresa que vende bienes o servicios a través de Internet ajuste los precios sobre la marcha en respuesta a las demandas del mercado. La fijación de precios y el aprendizaje dinámico han recibido una atención considerable en los últimos años por parte de diferentes comunidades científicas [39]. Tres factores han contribuido a este fenómeno: la mayor disponibilidad de datos de demanda, la facilidad para cambiar los precios debido a las nuevas tecnologías y la disponibilidad de herramientas de apoyo a la toma de decisiones en tiempo real [45].

Actualmente, otras industrias han implementado exitosos sistemas de recomendación de DP. Un claro ejemplo lo tenemos con el modelo de la compañía Uber, el cual actualiza los precios de acuerdo a las horas pico, como pueden ser viernes y sábados por la noche, en ciertos días festivos, como Halloween y Nochevieja, y durante eventos particularmente grandes y condiciones climáticas adversas [31]. El objetivo primordial de las técnicas de DP, es la identificación de la disposición a pagar (*Willingness to Pay*, WTP) de los pasajeros potenciales que buscan hacer una reserva de vuelo. Esta identificación del WTP, actualmente es una tarea difícil de realizar dentro de las aerolíneas, debido a la estructura rígida de clases de tarifas que existe actualmente, las cuales se están sujetas a una serie de precios discretos, dificultando la segmentación de pasajeros e identificación de ciertas características para realizar ofertas de forma dinámica y personalizadas.

Regresando a la alta competitividad en la industria de la aviación, y la gran actividad de publicación de tarifas que existe diariamente, se hace obligado que los equipos de analistas de precios dediquen un tiempo significativo analizando e interpretando las acciones de otras ae-

rolíneas para tomar una decisión y poder así determinar sus propias estrategias de precios con base a lo que está sucediendo en el mercado. Así mismo, la diversidad de restricciones que tienen las tarifas aéreas, los diversos puntos de precios, y el alto volumen de tarifas por clase que hay el mercado, hace de este proceso de análisis una tarea extremadamente compleja, generando una pérdida de identificación de tendencias y eventos que están sucediendo en el mercado. Lo anterior podría llevar a falsas conclusiones así como a un parcial entendimiento de la realidad con base a las acciones efectuadas por otras aerolíneas. En base a todo lo anterior, **la hipótesis de partida de esta tesis doctoral es que la utilización de métodos de clasificación basados en reglas y de métodos descriptivos supervisados permiten una mejor comprensión e interpretación de los cambios de precios que suceden frecuentemente en la industria aérea**. Para ello, la presente tesis doctoral propone una metodología que tiene por objetivo la clasificación eficaz de tarifas aéreas así como la obtención de un modelo de alta interpretación. Esta metodología ha sido probada experimentalmente con tarifas reales que han sido lanzadas al público por la aerolínea Air Canada. La automatización de un modelo de clasificación, el cual es fácilmente interpretable, puede generar ganancias, evitar errores humanos, y disminuir el tiempo que se dedica a esta tarea de forma manual, permitiendo a los analistas de precios tener una perspectiva clara de lo que está ocurriendo en el mercado.

Por otro lado, la presente tesis doctoral también propone una segunda metodología enfocada a la creación de un sistema de recomendación de ofertas dinámicas que sirva como un modelo de alta interpretación basado en la tarea de descubrimiento de subgrupos (*Subgroup Discovery*, SD). Este modelo permite identificar subgrupos de interés para el ajuste de precios en base a las características específicas de los pasajeros, teniendo como objetivo principal el incremento de reservas de vuelos a través una página Web. Este sistema de recomendación ha sido probado experimentalmente con datos reales y privados pertenecientes una aerolínea comercial.

1.1. Objetivos

El primer objetivo de esta tesis es obtener un modelo de predicción de tarifas aéreas, el cual, permita adquirir conocimiento novedoso de las estrategias de precios mediante reglas de clasificación. Además, permitirá conocer los cambios que suceden diariamente dentro un entorno competitivo. Para obtener este modelo, se utilizarán técnicas de DM y, particularmente, algoritmos de clasificación. Se propone una metodología que incluye un algoritmo evolutivo basado en programación de expresión genética (*Gene Expression Programming*, GEP) para el aprendizaje de atributos (*Feature Learning*, FL) usando conjuntos

de datos reales de tarifas de la aerolínea Air Canada. Finalmente, para conseguir este objetivo, se deben alcanzar otros sub-objetivos particulares, los cuales se describen a continuación:

- Desarrollar e implementar una metodología automatizada para la extracción de reglas de alta interpretación que se adapte a los cambios que sucedan en el mercado, debido a las acciones tomadas por las aerolíneas competidoras.
- Desarrollar e implementar un algoritmo que pueda imitar el proceso de aprendizaje de atributos que realizan los analistas de precios de forma manual, permitiendo obtener conjuntos de datos que faciliten la tarea de clasificación de tarifas aéreas sin perder eficacia y generando mejores modelos de interpretación.
- Realizar un análisis del rendimiento y los modelos generados de diferentes algoritmos de clasificación al ser aplicados a distintos conjuntos de datos. Verificar y comparar los modelos de interpretación que se generan con base en los conjuntos datos del antes y él después del ser aplicado el algoritmo de GEP antes mencionado.
- Obtener un modelo de alta interpretación de tarifas aéreas que pueda ser utilizado en un entorno industrial.

Como segundo objetivo de esta memoria de tesis se plantea obtener un sistema de recomendación de ofertas dinámicas a través de un modelo interpretación basado en reglas mediante el descubrimiento de subgrupos de interés. En esta metodología se propone un algoritmo de evolución gramatical (*Grammatical Evolution*, GE) para la selección de atributos, obteniendo conjuntos de datos mejorados que permiten a los algoritmos de SD extraer reglas significativas, y minimizando la extracción de reglas redundantes. Para conseguir este objetivo, se deben alcanzar otros sub-objetivos particulares, los cuales se describen a continuación:

- Desarrollar e implementar una metodología de automatización para la extracción de reglas que permitan el descubrimiento de subgrupos de interés que sirva como base para un sistema de recomendación de ofertas dinámicas.
- Desarrollar e implementar un algoritmo de GE para la selección de atributos que permita obtener mejores conjuntos de datos, para facilitar la extracción de reglas significativas a los algoritmos de SD.
- Realizar un análisis del rendimiento y los modelos generados de diferentes algoritmos de SD al ser aplicados a datos históricos pertenecientes a una aerolínea comercial.

Verificar y comparar los modelos de interpretación que se generan con base en los conjuntos datos del antes y él después del ser aplicado el algoritmo de GE.

- Obtener un sistema de recomendación de ofertas dinámicas automatizado.

1.2. Contenido del documento

A continuación se describen brevemente los distintos capítulos en los que está dividida la memoria de esta tesis doctoral:

- El capítulo *Introducción* presenta una visión general de este trabajo. En este capítulo se realiza el planteamiento del problema a resolver y su justificación e importancia. Además, se citan los objetivos que se pretenden conseguir, y la necesidad de aplicar dos metodologías de DM; una primera para la predicción e interpretación de tarifas aéreas, y una segunda para la creación de una tabla dinámica de descuentos.

- El capítulo *Antecedentes* introduce las técnicas, tareas y métodos dentro de la minería de datos, así como los problemas que existen para generar modelos altamente interpretables. Introduce los conceptos de la guerra de precios y precios dinámicos dentro del ámbito de las aerolíneas comerciales. También se realiza una revisión de la literatura sobrre los trabajos más importantes enfocados a estos dos conceptos. Este capítulo concluye con una introducción exhaustiva a la computación evolutiva y sus aplicaciones dentro de la minería de datos.

- El capítulo *Modelo de precios competitivos para enfrentar la guerra de tarifas aéreas* propone una nueva metodología que permite generar modelos altamente interpretable sin perder eficacia en la clasificación de tarifas. Esta metodología inclue el proceso de recopilación de información, creación de conjuntos de datos y su pre-procesado. Se presenta, además, el diseño del algoritmo GEP-FL que se ha desarrollado para la optimización de los atributos que componen los conjuntos de datos; facilitando la tarea de clasificación y la obtención de un modelo de alta interpretación. Este capítulo también describe los diferentes experimentos realizados con diferentes técnicas de DM y los resultados obtenidos. Finalmente, se presentan las reglas de clasificación que forman el modelo de interpretación de tarifas aéreas.

- El capítulo *Modelo de Descuentos Dinámicos* propone una metodología para generar un sistema de recomendación de descuentos dinámicos altamente interpretable. En este capítulo se describe el proceso de recopilación de información, creación del conjunto de datos y su pre-procesado. A su vez, se presenta el diseño del algoritmo GE-FS que se ha desarrollado para la generación de diversos conjuntos de datos; facilitando la tarea del descubrimiento de subgrupos de interés. Además, este capítulo describe los diferentes experimentos realizados con diferentes técnicas de DM y los resultados obtenidos. Por último, se presentan las reglas que forman el modelo de recomendación para asignar las ofertas.

- El capítulo *Conclusiones y trabajo futuro* resume los resultados alcanzados en esta memoria de tesis, y se plantean algunas posibilidades de mejoras y trabajos en el futuro sobre nuevas líneas abiertas a partir de este trabajo.

- Por último, la presente memoria contiene dos *Apéndices* que incluyen información añadida sobre los datos que se han utilizado en la fase experimentación de las dos metodologías propuestas. En ellos se describen el significado, el tipo de dato y la importancia de cada atributo.

Capítulo 2

Antecedentes

En este capítulo, se describe formalmente el marco conceptual utilizado en esta tesis doctoral. En él se incluye una revisión del proceso de extracción de conocimiento y de las principales técnicas de análisis de datos tales como técnicas predictivas, descriptivas y descriptivas supervisadas. Además, el presente capítulo introduce el concepto de modelos de alta interpretación, así como las métricas más interesantes empleadas en esta tesis doctoral. Adicionalmente, se presenta la exposición de los trabajos más notables para enfrentar la guerra de precios y la aplicación de precios dinámicos dentro del área de la aviación comercial. Finalmente, se realiza una revisión histórica de los conceptos básicos de la computación evolutiva, exponiendo las diversas etapas de su desarrollo, profundizando en los paradigmas de Programación de Expresión Genética (*Gene expression programing*, GEP) y Evolución Gramatical (*Grammatical Evolution*, GE), así como sus aplicaciones en diversas áreas del conocimiento.

2.1. Proceso de extracción del conocimiento

El proceso de extracción de conocimiento (*Knowledge Discovery in Databases*, KDD) se define como el proceso que permite extraer automáticamente conocimiento a partir de grandes volúmenes de datos. KDD se organiza entorno a cinco fases [48] tal y como se observa en la Figura 2.1. Es importante destacar que KDD se conoce como un proceso repetitivo (alguna de las fases pueden hacer volver a pasos anteriores) e iterativo (a menudo son necesrias varias iteraciones para extraer conocimiento de calidad). A continuación se describen brevemente cada una de las fases de KDD (ver Figura 2.1):

- *Fase de integración y recopilación*: La primera etapa del KDD consiste en identificar las fuentes de datos y recopilar los datos de todas esas fuentes para poder homogeneizarlos

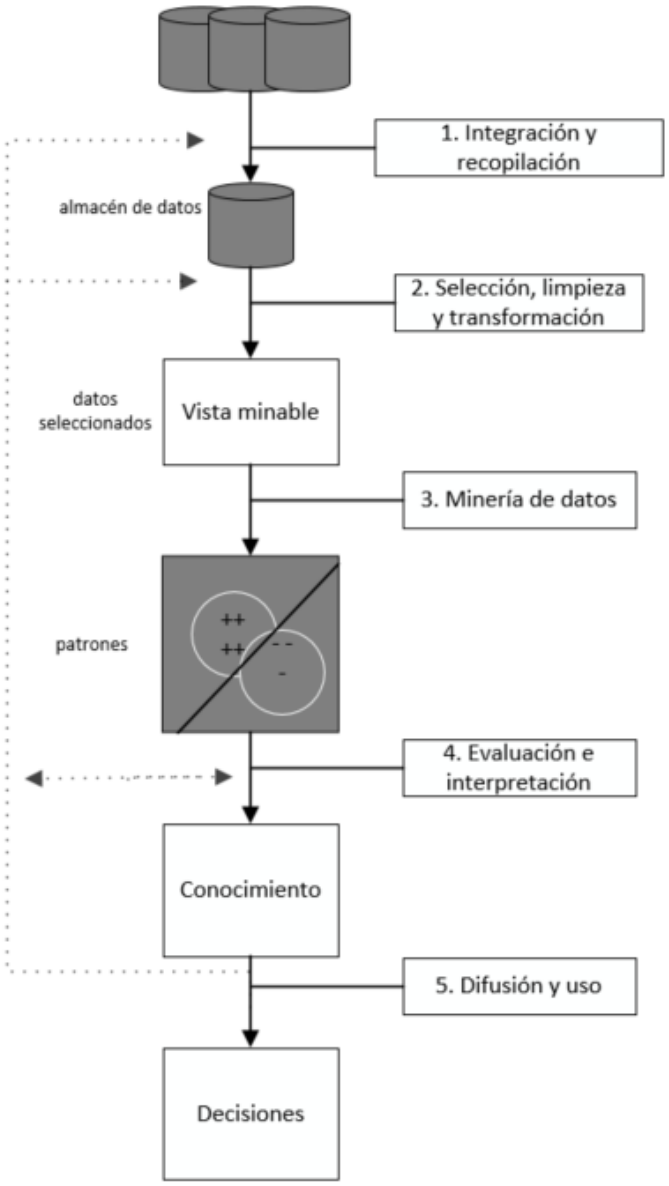

Figura 2.1 Fases del proceso KDD

e integrarlos, produciendo el conjunto de datos inicial sobre el que trabajar. Este conjunto de datos tiene un formato unificado y, además, posee una estructura adecuada para el análisis de los datos.

- *Fase de limpieza y transformación*: La calidad del conocimiento no sólo depende del algoritmo de DM utilizado, sino también de la calidad de los datos que se van a analizar. Por ello, es de suma importancia este paso de limpieza y transformación de los

datos [67]. Por un lado, la limpieza de datos es el proceso que permite detectar, corregir o eliminar registros corruptos o imprecisos dentro del conjunto de datos. Por otro lado, la transformación de datos es el proceso de convertir el formato de origen, el valor o la estructura de los datos en otro formato destino. Esto implica agregar, replicar y eliminar registros, así como estandarizar su formato. También implica identificar el formato actual de la información y la asignación de datos. En términos generales, esta fase suele aparecer descompuesta en tres subfases: selección de datos (técnicas de filtrado de registros y de atributos que permiten eliminar datos); limpieza de datos (control y manejo de valores inexistentes o erróneos); transformación de datos (transformar los datos para obtener el formato necesario para la posterior fase de extracción de conocimiento).

- *Fase de extracción de conocimiento* (*Data Mining*, DM): esta fase [170] es la más importante del proceso de KDD. Tal es su importancia que, en multitud de documentos, aparece el concepto de DM para nombrar todo el proceso KDD. Esto se realiza construyendo un modelo basado en los datos recopilados y tratados para ello. Un modelo, en este contexto, se denomina como una representación simbólica y resumida de los datos que se han analizado, lo que permitirá al experto extraer sus propias conclusiones. Es necesario tomar una serie de decisiones antes de empezar el proceso:

  - Determinar que tipo de técnica es la más apropiada. Por ejemplo, se podría usar la clasificación para predecir en una aerolínea que pasajeros harán una reservación de vuelo.
  - Elegir el tipo de modelo. Por ejemplo, para una tarea de clasificación se podría usar un árbol de decisión, en caso de querer obtener un modelo en forma de reglas.
  - Elegir el algoritmo adecuado para resolver la tarea y obtener el tipo de modelo que se está buscando. Esta elección es pertinente porque existen muchos métodos para construir modelos. Por ejemplo, para crear árboles de decisión de clasificación se puede hacer uso de CART o C5.0, entre otros.

  La siguiente sección (ver Sección 2.1.1) analiza en detalle las técnicas de DM existentes organizadas por tareas.

- *Fase de evaluación e interpretación*: Medir la calidad de los patrones descubiertos por un algoritmo de DM no es un problema trivial, ya que esta medida puede atañer a varios criterios, algunos de ellos bastante subjetivos. Idealmente, los patrones descubiertos deben tener tres cualidades: ser precisos, comprensibles (inteligibles) e interesantes

(útiles y novedosos). Según el campo de aplicación puede interesar mejorar algún criterio y sacrificar ligeramente otro.

- *Fase de difusión y uso*: Una vez construido un modelo, éste puede utilizarse principalmente con dos finalidades: 1) para que un analista recomiende acciones basándose en el modelo y en sus resultados; 2) para aplicar el modelo a diferentes conjuntos de datos. Es importante destacar que el conocimiento extraído a través del modelo debe de integrar el *know-how* de la organización, y el modelo debe evolucionar correctamente en el tiempo según el tipo de organización.

2.1.1. Minería de datos: tareas y métodos

DM es un término que integra numerosas técnicas de análisis de datos y extracción de modelos, teniendo como principales objectivos la extracción de patrones, el descubrimiento de tendencias y la predicción de comportamientos. DM incorpora diferentes técnicas de los campos del aprendizaje automático, la estadística, las bases de datos, los sistemas de toma de decisión, la inteligencia artificial, así como otras áreas de la informática y de la gestión de información. En [170] se define DM como el proceso de extraer conocimiento útil y comprensible, previamente desconocido, desde grandes cantidades de datos almacenados en distintos formatos. Este proceso debe ser automático o semi-automático, utilizando estos patrones descubiertos para la toma de decisiones más seguras, que reporten algún tipo de beneficio dentro de una organización.

De una forma simplista y ambiciosa, se puede decir que el objetivo primordial de DM es convertir datos en conocimiento [78]. Este conocimiento puede ser en forma de relaciones, patrones o reglas inferidas de los datos y (previamente) desconocidos, o bien en forma de una descripción más concisa (resumen de los mismos). Estas relaciones o resúmenes constituyen el modelo de los datos analizados. Existen muchas formas de representación de estos modelos, por lo que cada una de ellas determina el tipo de técnica que puede usarse para inferirlos. En la práctica, las tareas pueden ser de dos tipos: predictivas o descriptivas. Las primeras pretenden determinar valores futuros o desconocidos de variables de interés, que se denomina variables objetivas o dependientes, empleando otras variables dentro de un conjunto de datos, a las que se conocen como variables independientes o predictoras. Por ejemplo, un modelo predictivo sería aquel que permite estimar la demanda de una nueva ruta de vuelo de una aerolínea comercial. Por otra parte, las técnicas descriptivas identifican patrones que explican o resumen los datos, es decir, sirven para explorar las propiedades de los datos examinados. En este caso, el objetivo nunca es predecir el comportamiento de datos futuros. Un ejemplo de esto, puede ser una agencia de viajes que desea identificar grupos de

personas con unos mismos gustos, con el objeto de organizar varias ofertas para cada grupo y poder remitirles esta información.

Con todo lo anterior en mente, podemos decir que las tareas más importantes de DM se pueden agrupar de la siguiente forma:

- **Predictivas:** este tipo de tarea trabaja con aquellos problemas en los que se requiere predecir uno o más valores para uno o más datos de ejemplo [167]. Los ejemplos que componen una evidencia del problema van acompañados de una variable objetivo (coloquialmente conocida como salida) de tipo nominal o numérica. En función del tipo de variable objetivo, se definen varios tipos de tareas (clásicas) predictivas entre las que se encuentran principalmente las siguientes:

  - **Clasificación** [78]: sea E el conjunto de todos los posibles elementos de entrada, y S el conjunto de valores de salida (valores categóricos), el objetivo de esta tarea es el de aprender una función $\lambda : E \rightarrow S$, denominada clasificador, que represente la correspondencia existente en los valores de E y los valores de S. De manera formal, esta tarea puede definirse como, sea δ el conjunto de datos etiquetados, tal que $\delta = \{< e,s >: e \in E, s \in S\}$, y considerando que los ejemplos $e \in E$ van acompañados de un valor de S, la función a aprender $\lambda : E \rightarrow S$ será capaz de determinar la clase (etiqueta) para cada nuevo ejemplo sin etiquetar, es decir, dará un valor de S para cada valor de e. Algunos ejemplos de este tipo de tarea puede ser el de determinar si un mensaje de correo electrónico es *spam* o no, o el de determinar cual es el mejor medicamento para una determinada patología.
  - **Regresión**: de manera análoga a la tarea de clasificación, esta tarea puede definirse foormalmente como la de aprender una función $\lambda : E \rightarrow S$ a partir de un conjunto de todos los posibles elementos de entrada E , y un conjunto de valores de salida S (numéricos en este caso). La función a aprender $\lambda : E \rightarrow S$ será capaz de determinar el valor numérico (entero o real) asociado a cada nuevo ejemplo sin etiquetar, es decir, dará un valor se S para cada valor de e. Algunos ejemplos de este tipo de tarea son el de estimar las ventas de una empresa en un año; predecir el número de unidades defectuosas de una partida de productos; predecir la presión de una válvula a partir de una serie de entradas, etc.

  Gracias al auge de la minería de datos y, más especialmente, de las técnicas de aprendizaje automático, se han desarrollado multitud de propuestas consideradas como "avanzadas" que, si bien están basadas en las técnicas clásicas de clasificación y regresión, permiten resolver diferentes tipos de problemas. En ocasiones, estas técnicas

avanzadas son ligeras modificaciones que, según el foro en el que se debatan, pueden considerarse como independientes o formar parte de las técnicas de clasificación y/o regresión. Un ejemplo claro de este tipo de métodos avanzados de minería de datos son los relativos a la **clasificación suave** donde, se estiman en primer lugar las probabilidades condicionales a la variable de salida y, posteriormente, se obtiene la función $\lambda : E \rightarrow S$ a partir de las probabilidades estimadas. Otra técnica considerada como avanzada es la **clasificación multi-etiqueta**, la cual a diferencia de las tareas de clasificación clásicas en las que las etiquetas de clase se excluyen mutuamente, la clasificación multi-etiqueta requiere algoritmos especializados de aprendizaje automático que permitan predecir múltiples clases o etiquetas mutuamente no exclusivas. Para la descripción formal de la tarea de clasificación multi-etiqueta se considera al conjunto de etiquetas presente en un problema determinado: $L = \{\lambda_i : i = 1..n\}$ siendo n el número máximo de etiquetas que puede tener asociadas un patrón. Por otro lado, se denominará al conjunto $D = \{D_j : j = 1..m\}$ conjunto de datos multi-etiqueta. Cada elemento del conjunto de datos D, también denominado patrón multi-etiqueta, estará compuesto por un par de elementos $D_j = (X_j, Y\,j)$ siendo X_j el vector de características asociado al patrón e $Y_j \subseteq L$ el subconjunto de etiquetas asociadas con el patrón j, por tanto $|Y_j|$ será el número de etiquetas asociadas al patrón j.

- **Descriptivas**: En este tipo de tarea se trabaja con aquellos problemas en los que no se quiere obtener una predicción del comportamiento de una o varias variables de entrada, sino que se considera el análisis del comportamiento de los datos (datos históricos) para comprender sus relaciones y propiedades intrínsecas. En este tipo de técnicas, los datos no están etiquetados y, por tanto, no se pretende definir una función que prediga valores sino un conjunto de patrones y/o reglas independientes que describan los datos. Las tareas descriptivas más importantes son:

  - **Agrupamiento (*clustering*)**: el objetivo de esta tarea es obtener grupos o conjuntos entre los elementos de un conjunto de datos δ, de tal manera que los elementos asignados al mismo grupo sean similares. Uno de los aspectos más importantes a tener en cuenta en un problema de *clustering* es el desconocimiento del número de grupos existentes, así como de la partenencia o no de un registro a un grupo. En ocasiones, el número de grupos es conocido, por lo que el problema consiste en determinar a qué grupo pertenece cada uno de los registros del conjunto de datos en base a las características intrínsecas de los mismos. En esta ocasión, la función a obtener es idéntica a la de la clasificación antes mencionada, es decir, $\lambda : E \rightarrow S$, con la diferencia de que los valores de S y sus miembros se

crean, durante el proceso de aprendizaje. La principal utilidad del agrupamiento es averiguar que las instancias se agrupan en varios segmentos, lo cual resulta de gran utilidad porque se puede determinar el comportamiento de una nueva instancia viendo a que grupo de instancias pertenece. Es decir, si se tiene que una nueva instancia es etiquetada en el grupo 3 significa que, posiblemente, en muchos aspectos se comporte como el resto de elementos del grupo 3. Un ejemplo de *clustering* en marketing es el de agrupar los clientes en segmentos diferenciados, estudiar que grupos se comportan mejor ante determinados productos, y después orientar ciertos productos a ciertos grupos. También se pueden definir variantes para realizar un agrupamiento suave u obtener estimadores de probabilidad de agrupamiento, que propongan más flexibilidad y posibilidades a la hora de interpretar y trabajar con los grupos formados, o permiten construir taxonomías o agrupamientos jerárquicos. Por último, destacar que la tarea general del agrupamiento se puede utilizar con objetivos ligeramente distintos. Por ejemplo, si reducimos un conjunto de datos de miles de ejemplos a media docena de grupos, y se analizan los grupos formados, se podrían entender mejor los datos originales y, en cierto modo, estos grupos pueden servir como resumen de los datos originales. Incluso, muchos autores hoy en día consideran el agrupamiento con este objetivo como una tarea nueva, llamada **sumarización** [78].

- **Minería de patrones y reglas de asociación**: El objetivo último de KDD es el de dar sentido a los datos, con el fin de sacar algún tipo de provecho de los mismos. El descubrimiento de patrones juega un papel fundamental, definiendo un patrón como un conjunto de elementos que representan algún tipo de homogeneidad y regularidad en los datos [3]. En términos generales, podemos indicar que un patrón representa propiedades intrínsecas e importantes de los datos. De manera formal, dado un conjunto de elementos $I = \{i_1, i_2, ..., i_n\}$ pertenecientes a un conjunto de datos, un patrón P se define como un subconjunto de I, es decir, $\{P = \{i_j, ..., i_k\} \subseteq I, 1 \leq j, k \leq n\}$, que describe características importantes de los datos. El análisis de la cesta de la compra es la primera aplicación, y más conocida, de la minería de patrones. Hay que tener en cuenta que, en la mayoría de las ocaciones, las compras se realizan por impulso, por lo que tener conocimiento de productos que se suelen comprar conjuntamente es de enorme utilidad para las empresas en el diseño de campañas de marketing. En ocasiones, el conocimiento otorgado por los patrones no es suficiente, y se requiere un mayor poder descriptivo. Surge así el concepto de reglas de asociación [78], el cual describe asociaciones o relaciones entre elementos en un conjunto de datos. Sea

P un patrón definido en un conjunto de elementos $I = \{i_1, i_2, ..., i_n\}$, y sea X e Y subconjutnos de un patrón de forma que $\{X \subset P \subseteq I \wedge Y = P \setminus X\}$, o también $\{Y \subset P \subseteq I \wedge X = P \setminus Y\}$. Una regla de asociación se define como una implicación de la forma $X \rightarrow Y$ donde X e Y son conjuntos disjuntos (no presentan ningún elemento en común). El significado de una regla de asociación es que una vez que el antecedente X de la regla es satisfecho, es bastante probable que el consecuente Y de la regla también se satisfaga. En otras palabras, una regla de asociación determina la probabilidad de que se cumpla el consecuente de la regla una vez que es conocido que el antecedente se ha cumplido.

- **Detección de valores e instancias anómalas**: la detección de valores anómalos o atípicos (*outlier detection*) puede ser muy útil precisamente para detectar comportamientos anómalos, que puede sugerir fraudes, fallos, intrusos o comportamientos diferenciados. La definición de instancia anómala es más general en el sentido que no solo considera un único atributo, sino que los considera todos. La tarea se define como el objetivo de encontrar aquellas instancias que no son similares a ninguna (o muy pocas) de las otras instancias. La manera de abordar el problema es generalmente la de agrupar los ejemplos y ver aquellas instancias que quedan desplazadas de los grupos mayoritarios. Para ellos son especialmente útiles los agrupadores suaves o estimadores de probabilidad de agrupamiento, ya que si un ejemplo tiene baja probabilidad de agrupamiento con todos los grupos se puede considerar un caso aislado y, por tanto, anómalo. También se utilizan otros métodos no necesariamente basados en la tarea de agrupamiento, como la medición de distancias (aquellas instancias cuyo vecino más próximo este muy lejos puede considerarse, en cierto modo, una instancia anómala). Un ejemplo de tareas de detección de instancias anómalas seria: encontrar, de las compras realizadas con tarjeta, aquellas que sean anómalas.

A partir de estas tres tareas clásicas descriptivas, se han desarrollado nuevas técnicas como son: la minería de patrones secuenciales [59], la minería de patrones de alta utilidad [87], la minería de patrones periódicos [60], entre otras. El problema de la minería de patrones secuenciales fue propuesto por *Agrawal y Srikant* [59], el cual tiene como objetivo la extracción de secuencias frecuentes o interesantes que ocurren en un conjunto de datos donde los elementos están ordenados de alguna forma (generalmente en el tiempo). Por su parte, la minería de patrones de alta utilidad está diseñada para encontrar patrones altamente rentables, de forma que los elementos de un conjunto de datos están relacionados con una utilidad que debe maximizarse [87]. Finalmente, la extracción de patrones periódicos es una tarea importante de minería de datos, ya que

los patrones pueden aparecer periódicamente en todo tipo de datos, y puede ser deseable encontrarlos para comprender los datos para tomar decisiones estratégicas. Los patrones periódicos son conjuntos de eventos o elementos que aparecen periódicamente en una secuencia de eventos o transacciones [60].

- **Descriptivas supervisadas**: El análisis de datos generalmente se divide en dos tipos de tareas, predictivas y descriptivas. Por un lado, las tareas predictivas incluyen técnicas donde los modelos, típicamente inducidos a partir de datos etiquetados, se utilizan para predecir una variable objetivo de ejemplos nunca antes vistos. Por otro lado, las tareas descriptivas, se enfocan en encontrar patrones que denoten cualquier comportamiento interesante en datos no etiquetados. Hasta hace poco, estas tareas han sido desarrolladas e investigadas por dos comunidades diferentes: las tareas predictivas principalmente por la comunidad de ML, y tareas descriptivas principalmente por la comunidad de DM. Sin embargo, diferentes dominios de aplicación a veces requieren que ambos grupos converjan en algún punto, dando lugar al concepto de la minería de patrones descriptivos supervisados. La minería de patrones supervisados tiene como principal objetivo comprender un fenómeno subyacente (según una variable objetivo) en vez de clasificar nuevos ejemplos. Las tareas descriptivas supervisadas más importantes son:

  - **Conjuntos de contraste**: Según la definición proporcionada en [40], los conjuntos de contraste son una agrupación de elementos que describen diferencias y similitudes entre grupos que el usuario desea contrastar. Los grupos bajo contraste son, a menudo, considerados como subconjuntos de un conjunto de datos común. Así, cada grupo a analizar puede obtenerse a partir de un función específica de los datos, o a partir de una o varias condiciones que permiten separar los grupos del conjunto de datos total. Es importante remarcar la independencia de tales subconjuntos, de tal forma que un registro específico sólo puede pertenecer a un único subconjunto, nunca a dos o más. Por lo tanto, mientras que los clasificadores colocan nuevos datos en una serie de categorías discretas, la minería de conjuntos de contraste toma la categoría y proporciona una evidencia estadística que identifica un registro como miembro de una clase (valor de la variable de destino). De manera formal, *Bay, Pazzani* [13] definieron un conjunto de contraste como un patrón P o conjunto de atributos y valores que difieren significativamente en sus distribuciones entre grupos. En términos generales, los conjuntos de contraste cuantifican la diferencia en la frecuencia de aparición (soporte) en cada subconjunto (grupo) S_x. Así, P se considera un conjunto de contraste si y solo si $\exists_{ij} : max(\mid soporte(P,S_i) - soporte(P,S_j) \mid) \geq \alpha \in [0,1]$. $soporte(P,S_i)$ se define

como el soporte de un patrón P en el subconjunto S_i. Los conjuntos de contraste se han aplicado a diferentes campos, incluidas las ventas a minoristas [166]; diseño de programas de seguros personalizados [173]; identificación de patrones en datos de rayos X de sincrotrón que distinguen muestras de tejido de diferentes formas de tumor canceroso [155]; clasificar entre dos grupos de pacientes con isquemia cerebral [102], entre otros.

- **Patrones emergentes**: Los patrones emergentes pueden considerarse un caso especial de conjuntos de contraste en el que se consideran solo dos grupos, con el objetivo de buscar características de un grupo que lo discriminen respecto al otro [127]. Los patrones emergentes fueron definidos por *Dong* [20] como una tarea de minería de datos que busca patrones discriminativos cuyo soporte (frecuencia de aparición) aumenta significativamente de un conjunto de datos a otro. De manera formal, un patrón emergente P sobre dos conjuntos de datos Ω_1 y Ω_2 se define como aquel que cumple $soporte(P\,,\,\Omega_1)/soporte\,(P\,,\,\Omega_2) > \alpha$ o $soporte(P\,,\,\Omega_2)/soporte\,(P\,,\,\Omega_1) > \alpha$. Esta diferencia en el soporte se cuantifica como una tasa de crecimiento, considerándose un patrón como emergente si la tasa es superior a 1. En aquellos casos en los que la tasa de crecimiento es infinito se les considera como *jumping emerging patterns*, generalmente porque $soporte(P\,,\,\Omega_1) = 0$ y $soporte(P\,,\,\Omega_2) \neq 0$, o porque $soporte(P,\Omega_1) \neq 0$ y $soporte(P\,,\,\Omega_2) = 0$. Algunos autores [127] definen a los patrones emergentes como reglas de asociación con un conjunto de elementos (el patrón P) en el antecedente y un consecuente fijo (el conjunto de datos), es decir, $P \rightarrow \Omega$. Por lo tanto, la tasa de crecimiento se calcula como $\text{confianza}(P \rightarrow \Omega_1)/confianza(P \rightarrow \Omega_2) = confianza(P \rightarrow \Omega_1)/1 - confianza(P \rightarrow \Omega_1)$. Los patrones emergentes se han aplicado principalmente al campo de la bioinformática y, más específicamente, al análisis de datos de micromatrices [108]. Algunos autores [109] han centraron sus estudios en la extracción de patrones emergentes para analizar genes relacionados con el cáncer de colon. Los patrones emergentes también se han aplicado para analizar el comportamiento de clientes [156] con la finalidad de descubrir cambios inesperados en los hábitos de compra.
- **Descubrimiento de subgrupos**: Es una de las técnicas más conocidas en el campo de la minería supervisada de patrones descriptivos [10]. El objetivo del descubrimiento de subgrupos es el de identificar un conjunto de patrones de interés en base a una distribución inusual con respecto a una determinada propiedad de interés (concepto o variable objetivo). El concepto de descubrimiento de subgrupos fue introducido por primera vez por *Klösgen* [96] y *Wrobel* [174] de

la siguiente manera: dada una población de individuos (clientes, objetos, etc.) y una propiedad de esos individuos es interesante, la tarea del descubrimiento de subgrupos es encontrar poblaciones de subgrupos que son estadísticamente más interesantes para el usuario, por ejemplo, subgrupos que son tan grandes como sea posible y tener las características estadísticas más inusuales con respecto a un objetivo atributo de interés.

El descubrimiento de subgrupos combina características de tareas descriptivas y predictivas [20] y descubre subgrupos explícitos a través de reglas únicas y simples, es decir, teniendo una clara estructura y pocas variables [79]. Estas reglas, en la forma $P \rightarrow Target$, denotan una distribución estadística inusual de P (patrón que incluye un conjunto de características) con respecto al concepto objetivo o variable de interés. Si bien esta tarea se ha aplicado, de manera general, a objetivos discretos (espacio finito de valores), también ha sido aplicado a dominios continuos en los últimos años [10].

Hoy en día, existen otras tareas adicionales que se pueden agrupar bajo el concepto de minería de patrones descriptivos supervisados, ya que proporcionan cualquier tipo de conocimiento descriptivo a partir de datos etiquetados. Un ejemplo son las reglas de asociación de clase [163], siendo un tipo especial de reglas de asociación donde el consecuente de la regla es la variable objetivo. Este tipo de reglas ha dado lugar a la clasificación asociativa [132], donde un clasificador es generado a partir del conjunto de reglas de asociación de clase extraídas en un proceso previo. Otro ejemplo es la minería de modelos excepcionales [115], la cual se define como una generalización multiobjetivo de descubrimiento de subgrupos. Esta tarea busca subconjuntos de datos interesantes en variables objetivo predefinidas y describe las razones para comprender la causa de las interacciones inusuales entre los objetivos.

Como se ha podido comprobar, algunas tareas están en cierto modo relacionadas, lo que permite que la terminología de alguna de ellas sea bastante diversa y, en ocasiones, confusa en cuanto a los nombres usados en diferentes herramientas y textos de referencia. En términos generales, se suele utilizar el término de “aprendizaje supervisado"para los métodos predictivos y el término “aprendizaje no supervisado"para los métodos descriptivos.

2.1.2. Correspondencia entre tareas y métodos

Cada una de las tareas que se han descrito en este trabajo, como cualquier otro problema, requiere de métodos, técnicas, o algoritmos para resolverlas. De hecho, una tarea puede tener

muchos métodos para ser resuelta y esto no es la excepción dentro de DM. A continuación se hace una pequeña reseña de la variedad de técnicas existentes:

- Técnicas algebraicas y estadísticas: se basan en expresar los modelos y patrones mediante fórmulas algébricas, funciones lineales, funciones no lineales, distribuciones o valores agregados estadísticos tales como medias, varianzas, correlaciones, entre otros [154]. Frecuentemente, estas técnicas, cuando obtienen un patrón, lo hacen a partir de un modelo predeterminado del cual, se estiman unos coeficientes o parámetros. Algunos de los algoritmos más conocidos dentro de este grupo de técnicas son la regresión lineal [71], la regresión logarítmica [98] y la regresión logística [100]. Los discriminantes lineales y no lineales [117], basados en funciones predefinidas, también conocidos como discriminantes paramétricos.

- Técnicas bayesianas: se basan en determinar la probabilidad de pertenencia a una clase de grupo, mediante la estimación de las probabilidades condicionales inversas o a priori, utilizando por ello el teorema de Bayes [159]. Algunos algoritmos muy populares son el clasificador bayesiano naive [142], los métodos basados en máxima verisimilitud y el algoritmo EM [16]. Las redes bayesianas generalizan las topologías de las interacciones probabilísticas entre variables y permiten representar gráficamente dichas interacciones.

- Técnicas basadas en conteo de frecuencias y tablas de contingencia: estas técnicas se basan en contar la frecuencia en la que dos o más sucesos se presenten conjuntamente. Cuando el conjunto de procesos posibles es muy grande, existen algoritmos que comienzan por pares de sucesos, incrementando los conjuntos solo en aquellos casos en que las frecuencias conjuntas superen un cierto umbral. Un ejemplo de estos algoritmos es el popular algoritmo *Apriori* [76].

- Técnicas basadas en árboles de decisión y sistemas de aprendizaje de reglas: son técnicas que, además de su representación en forma de reglas, se basan en dos tipos de algoritmos: los algoritmos denominados “divide y vencerás", como el C4.5 [37], CART [15], o CN2 [34].

- Técnicas relacionales, declarativas y estructurales: la característica principal de este conjunto de técnicas es que representan los modelos mediante lenguajes declarativos, como los lenguajes lógicos, funcionales, o lógico-funcionales. Las técnicas de ILP (programación lógica inductiva) [97] son las más representativas y las que han dado nombre a un conjunto de técnicas denominadas *minería de datos relacional*.

- Técnicas basadas en redes neuronales artificiales: se trata de técnicas que aprenden un modelo mediante el entrenamiento de los pesos que conectan un conjunto nodos o neuronas. La topología de la red y los pesos de las conexiones determinan el patrón aprendido. Existen innumerables variantes de organización: perceptron simple, redes multicapa, redes de base radial, redes de Kohonen, entre otras. Uno de los algoritmos más conocidos es el de retropropagación (*backpropagation*) [143].

- Técnicas basadas en núcleo y máquinas de soporte vectorial: se trata de técnicas que intentan maximizar el margen entre grupos o clases formadas. Para ello se basan en unas transformaciones que pueden aumentar la dimensionalidad. Estas transformaciones se llaman núcleos (*kernels*). Existen muchas variantes, dependiendo del núcleo utilizado y de la manera de trabajar con el margen [78].

- Técnicas estocásticas y difusas: bajo este paraguas se incluyen la mayoría de las técnicas que, junto a las redes neuronales, forman lo que se denomina computación flexible (*soft computing*). Son técnicas en las que o bien los componentes aleatorios son fundamentales, como el *simulated annealing* [17], los métodos evolutivos [90] y genéticos [81], o bien al utilizar funciones de pertenencia difusas (fuzzy) [14].

- Técnicas basadas en casos, en densidad o distancia: son métodos que se basan en distancias al resto de elementos, ya sea directamente, como los vecinos más próximos (los casos similares) [160], de una manera más sofisticada, mediante la estimación de funciones de densidad. Además de los vecinos más próximos, algunos algoritmos muy conocidos son los jerárquicos, como Two-step [94] o COBWEB [29], y los no jerárquicos, como K medias [178].

Aparte de todo lo anterior, existen multitud de híbridos que dificultan realizar una taxonomía óptima y razonable, que aglutine grupos de técnicas de una manera indiscutible y que no confunda tareas con métodos. Por lo tanto, es necesario conocer las capacidades de cada técnica, los ámbitos donde suelen funcionar mejor, la eficiencia, la robustez, entre otros; definiendo las características funcionales con respeto a las demás.

2.1.3. Interpretabilidad: modelos basados en reglas

En la actualidad, diversas ramas de la Inteligencia Artificial (*Artificial Intelligence*, AI), se encuentran en el centro de muchos sectores activos que han adoptado las nuevas tecnologías de la información [146]. Tomando en cuenta que las raíces de la AI se remontan de varias décadas atrás, existe un claro consenso sobre la importancia primordial que hoy en día tienen las máquinas inteligentes, dotadas de capacidades de aprendizaje, razonamiento y adaptación.

Es en virtud de estas capacidades, que los métodos de AI están logrando resolver complicadas tareas computacionales, convirtiéndose en una parte fundamental para el desarrollo de la humanidad [168]. El grado de sofisticación de estos sistemas han logrado llegar hasta el punto que casi no se requiere intervención humana para su diseño y despliegue. De manera que, cuando las decisiones se derivan de estos sistemas; afectando la vida de los humanos, existe la necesidad de comprender cómo estos métodos de AI proporcionan tales decisiones [70]. Los paradigmas que subyacen a este problema caen dentro del llamado campo de la Inteligencia Artificial Explicable (*eXplainable AI*, XAI), que es ampliamente reconocido como una característica crucial para la práctica y despliegue de modelos de AI [9].

En los años recientes se ha observado el surgimiento de sistemas opacos tales como son las redes neuronales, las cuales han demostrado un alto grado de efectividad para resolver diversas tareas de la AI, por lo que los modelos derivados de estas son muy complicados de interpretar. Un factor importante a mencionar es que, por lo general; los humanos son reticentes a adoptar técnicas que no son directamente interpretables, manejables y confiables [185], debido a la demanda creciente de factores referentes a la ética de la AI [70]. A su vez, es costumbre pensar que al enfocarse únicamente en el rendimiento, los sistemas serán cada vez más opacos, lo cual es cierto en el sentido de que existe una compensación entre el desempeño de un modelo y su transparencia [41]. Sin embargo, una mejora en la comprensión de un sistema puede llevar a la corrección de sus deficiencias. Por lo tanto, al desarrollar un modelo ML, la consideración de la interpretabilidad como un controlador de diseño adicional puede mejorar su implementación por tres razones [9]:

- La interpretabilidad ayuda a garantizar la imparcialidad en la toma de decisiones, es decir, a detectar y, en consecuencia, a corregir el sesgo en el conjunto de datos de entrenamiento.
- La interpretabilidad facilita la provisión de robustez al resaltar posibles perturbaciones antagónicas que podrían cambiar la predicción.
- La interpretabilidad puede actuar como un seguro de que sólo las variables significativas infieren en los resultados, es decir, garantizando que una información veraz subyacente existe causalidad en el razonamiento del modelo.

Todo esto implica que la interpretación de un sistema debe ser práctica, proporcionando una comprensión de los mecanismos y predicciones del modelo, una visualización de la discriminación del modelo reglas decisión, o sugerencias sobre lo que podría perturbar el modelo [73]. Para evitar limitar la efectividad la generación actual de sistemas de AI,

XAI [72] propone crear un conjunto de técnicas de ML que (1) produzcan modelos más explicables mientras mantengan un alto nivel de rendimiento de aprendizaje (por ejemplo, precisión de predicción), y (2) permitir que los humanos entiendan y confíen en la generación y adaptación sistemas que contienen una AI.

Es importante mencionar que hasta ahora, no existe una definición matemática para la interpretación [124], de manera que se puede definir el concepto de interpretabilidad [122] como el grado en el que un ser humano puede comprender la causa de una decisión. Otra forma de definir este concepto de interpretabilidad [99] es como el grado en que un ser humano puede predecir consistentemente el resultado del modelo. De tal forma que, a mayor interpretabilidad de un modelo de aprendizaje automático, más fácil es para un humano comprender por qué se toman ciertas decisiones o predicciones. Por ende, un modelo de clasificación tiene mejor interpretabilidad en relación con otro modelo si sus predicciones son más fáciles de comprender que las predicciones del otro modelo. De manera que, un aspecto muy relevante en la extracción del conocimiento es que, los modelos extraídos sean comprensibles o tengan un alto grado de interpretabilidad que faciliten la toma de decisiones.

Dentro de algunos ámbitos de aplicación [145], la interpretabilidad es preferible a la exactitud (*Accuracy*, Acc.), específicamente cuando el objetivo es revelar datos ocultos y patrones que actúan como parte de un ciclo de retroalimentación positivo que pueda ser utilizado por los expertos de dominio y que a través del cual, los expertos puedan aprender nuevas dependencias y correlaciones de aquellos modelos que pueden ser fácilmente interpretados. En efecto, diversos modelos de alta interpretación han sido aplicados en diversas áreas del conocimiento, por ejemplo, los autores en [162] propusieron un modelo que muestra las causas del porqué un medicamento tiene un buen funcionamiento y las causas del porqué falla. En casos específicos, esto es significativo, pues permite a los expertos a mejorar los diseños de los medicamentos terapéuticos. *Bhargava et al.* [19] propusieron una metodología de interpretación y detección de programas maliciosos mediante la predicción de valores atípicos en conjuntos de datos que registran amenazas cibernéticas. Por otra parte, los modelos de clasificación interpretables han tenido éxito en el fracaso escolar, por lo que han sido un foco de atención de algunos investigadores [118]. Regresando al dominio de la medicina, diversos autores han considerado el uso de modelos de alta interpretación en diferentes enfermedades tales como: síndrome metabólico [181], artritis reumatoide [33], diabetes [179] y cáncer de mama [5], entre otros. Adicionalmente, otros investigadores [177] han trabajado en el problema de detección de rostros, pudiendo procesar imágenes con una rapidez excepcional y logrando altas tasas de detección a través de simples modelos de interpretación.

Centrándonos en DM [65], definido como el proceso de extraer conocimiento útil, comprensible y novedoso, vuelve a surgir el concepto de modelos altamente interpretables a partir de los datos. Para que este proceso sea efectivo debe ser automático o semi-automático; el uso de los patrones descubiertos debe ayudar a la toma de decisiones más segura, y que a su vez, reflejen algún beneficio para la organización. Por lo tanto, dos retos importantes de la DM son, por una parte, trabajar con grandes volúmenes de datos procedentes de diversos sistemas de información, con los problemas que esto conlleva, tales como ruido, datos ausentes, intratabilidad, volatilidad de los datos, entre otros. Por otra parte, utilizar técnicas adecuadas para analizar los mismos y extraer conocimiento novedoso, útil, e interpretable. En muchos casos, la utilidad del conocimiento extraído está íntimamente relacionada con la comprensibilidad del modelo inferido. Se debe hacer notar, que en muchas aplicaciones es fundamental hacer que la información descubierta sea más comprensible por los humanos, por ejemplo, usando representaciones gráficas, convirtiendo los patrones en lenguaje natural o empleando técnicas de visualización de los datos. Debido a lo anterior, en [78] de manera simplista pero ambiciosa, determinaron que el objetivo de la DM es convertir datos en conocimiento.

Para generar modelos de interpretación, la extracción de conocimiento se puede abordar, en función del problema a resolver, desde las dos perspectivas distintas que se han mencionado anteriormente en esta memoria de tesis: desde el punto de vista predictivo, como un proceso de inducción predictiva que intenta obtener conocimiento que permita pronosticar el comportamiento futuro según los datos disponibles, o desde el punto de vista descriptivo, cuyo objetivo fundamental es descubrir conocimiento de interés dentro de los datos, intentando obtener información que describa el modelo que existe detrás de los datos.

Cuando el objetivo es obtener un modelo de interpretación similar al lenguaje de los expertos a través de un enfoque de clasificación, se puede recurrir a dos tipos de algoritmos: los algoritmos basados en reglas y los árboles de decisión. Los primeros tratan de generar reglas siguiendo una estrategia de cubrimiento secuencial, utilizando un conjunto de entrenamiento (o aprendizaje) compuesto por objetos descritos por atributos de condición (con los cuales se forman las condiciones p) y el rasgo de decisión (clase) [175]. Los algoritmos basados en reglas que han tenido éxito en diversos ámbitos de aplicabilidad se pueden encontrar los siguientes: JRip [35], OneR [83], PART [61], Ridor [4], entre otros. Por otra parte, los árboles de decisión son quizá el método más fácil de utilizar y entender [78]. Un árbol de decisión [144] es un conjunto de condiciones organizadas en una estructura jerárquica, de tal manera que la decisión final a tomar se puede determinar siguiendo las condiciones que se cumplen, desde la raíz del árbol hasta alguna de sus hojas. Entre los algoritmos de

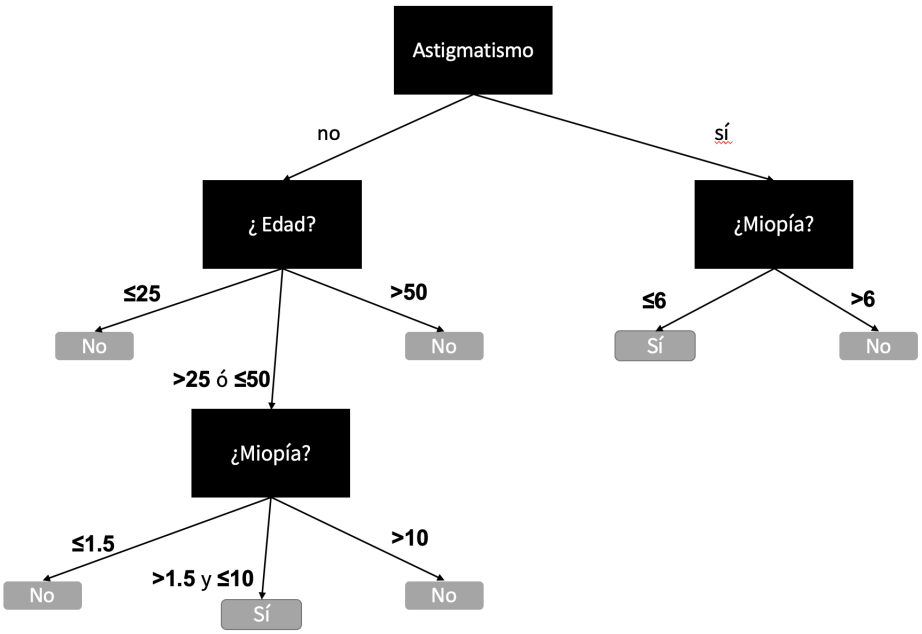

Figura 2.2 Modelo de interpretación generado a través de un árbol de decisión.

árboles de decisión que has sido implementados en diferentes dominios se pueden nombrar los siguientes: C4.5 [148], J48 Consolidated [135], J48 Graft [165], DecisionStump [89], simpleCART [24], BFTree [63], LMT [158], JCDT [1], entre otros.

La Figura 2.2 muestra un modelo de interpretación elaborado a través de un árbol de decisión, el cual podría ser utilizado como un sistema de recomendaciones que puede servir a un oftalmólogo para determinar si una persona cumple con las condiciones necesarias para realizarle o no una cirugía sin tener que gastar demasiados recursos mediante la aplicación de un examen más detallado. Como se puede observar en este modelo, es sencillo aplicar el árbol de decisión a nuevo paciente para saber si se le ha de recomendar o no dicha operación. Basta con realizar las preguntas y seguir las respuestas hasta alguna de las hojas de árbol catalogadas, con un "no" o "sí". Este modelo en concreto funciona como un "clasificador", es decir, dado un nuevo individuo lo clasifica en una de las clases posibles: "no" o "sí".

Por otro lado, los algoritmos basados en reglas son una generalización de los árboles de decisión en el que no se exige exclusión ni exhaustividad en las condiciones de las reglas, es decir, podría aplicarse más de una regla o ninguna, por lo que el orden de las reglas es muy importante para determinar ambiguedades. Retomado el ejemplo del modelo generado por el árbol de decisión, la Figura 2.3 muestra un ejemplo del árbol de decisión expresado como un modelo de interpretación en forma de reglas. Esta representación en general suele ser más

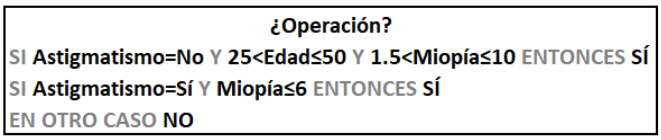

Figura 2.3 Modelo de interpretación generado por un algoritmo basado en reglas.

sucinta que los árboles de decisión, ya que permite englobar condiciones y permite el uso de las reglas por defecto, como la que comienza por "EN OTRO CASO"en el ejemplo anterior.

En [78] se afirma que la tarea de aprendizaje a la que se adecuan mejor los árboles de decisión es la clasificación. De hecho, clasificar es determinar de entre varias clases a que clase pertenece un objeto; la estructura de condición y ramificación de un árbol de decisión es idónea para este problema. La característica más importante del problema de clasificación es que se asume que las clases son disyuntivas, es decir, una instancia es de clase a o de la clase b, pero no puede ser al mismo tiempo de las clases a y b. Un ejemplo de clasificación podría ser determinar si un objeto en la carretera en un autobús de pasajeros o una motocicleta: o es una cosa o es la otra. Debido al hecho que la clasificación trata con clases o etiquetas disjuntos, un árbol de decisión conducirá un ejemplo hasta una y solo una hoja, asignando, por tanto, una única clase al ejemplo. Para ello, las particiones existentes en el árbol deben ser también disjuntas. Es decir, cada instancia cumple o no una condición. En el ejemplo de la Figura 2.2, una persona o bien tiene miopía > 6, o bien tiene miopía ≤ 6, pero no las dos condiciones a la vez. Además, dicha propiedad es exhaustiva, es decir, una de las dos condiciones se debe cumplir.

Como se ha mencionado anteriormente, los árboles de decisión se pueden expresar como un conjunto de reglas, por lo que, de este conjunto de reglas se derivan una serie de particiones en las cuales para cualquier condición, siempre aparece la o las condiciones complementarias. De facto, existen muchos conjuntos de reglas que no cumplen estas condiciones y, sin embargo, son capaces de clasificar la evidencia de una forma conveniente. La Figura 2.4

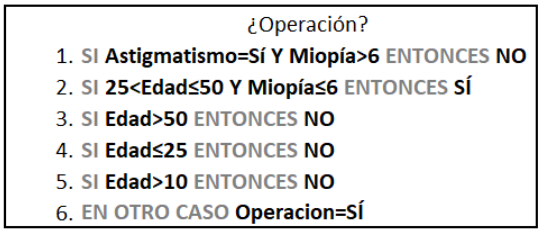

Figura 2.4 Conjunto de reglas para el problema de la cirugía refractaria.

muestra las diferencias, un modelo de reglas aplicado al problema de la cirugía refractaria con respecto a un modelo generado por un árbol de decisión. En este modelo varias reglas pueden aplicarse para un mismo ejemplo, como se observa en las reglas 1, 3, y 5, las cuales se pueden aplicar a la vez. En esta caso, las tres darán la misma predicción. De hecho, aunque en este conjunto de reglas no hay posibilidad de contradicción, por lo general, existen métodos que generan conjuntos de reglas que podrían ser, en algunos casos, contradictorias para algunos ejemplos. La forma de resolver es mediante el ordenamiento de las reglas o la ponderación de las predicciones. Los métodos [2] que generan estos conjuntos van añadiendo reglas una detrás de otra, mientras vayan cubriendo ejemplos de una manera consistente. A diferencia de los árboles de decisión, no se sigue con las condiciones complementarias a la utilizada en la regla anterior (exclusión y exhaustividad), sino que descartan los ejemplos ya cubiertos por las reglas obtenidas y con los ejemplos que quedan se empieza de nuevo. Esto hace que puedan aparecer nuevas condiciones que solapen o no con las reglas anteriores; esta manera de actuar [64] es la que utilizan los métodos de cobertura.

Para concluir esta sección es importante mencionar que independientemente de los recientes avances y desarrollo de diversas técnicas de DM y máquinas de aprendizajes (*Machine Learning*, ML) tales como técnicas de clasificación, reglas de asociación y técnicas descriptivas de aprendizajes supervisadas, entre otras; aún existe una brecha entre el modelado de datos y la extracción de conocimientos que no debe ser ignorada. Debido a esto, *Vellido et al.* [30] plantearon que la interpretabilidad es una cualidad primordial que los métodos de aprendizaje automático deben aspirar a lograr si se van a aplicar en la práctica. La interpretabilidad en el análisis de datos mediante el aprendizaje automático puede verse como un proceso con etapas interactivas, como se describe esquemáticamente en la Figura 2.5. Cuando se genera un modelo de datos se requieren de una serie de métodos o técnicas de DM para interpretarlos. A su vez, la interpretación de estos resultados se debe comunicar en el lenguaje de los expertos de dominio, lo cual sirve como un proceso de adaptación del modelo de datos.

2.2. Guerra de precios

La guerra de precios [77] se define como la situación en la que compañías rivales establecen precios significativamente inferiores a los que generalmente se cobran en la industria por cierto periodo de tiempo con el fin de sacar algún tipo de beneficio. En términos generales, las compañías deciden regularmente bajar el precio de alguno de sus productos para aumentar su competitividad en el mercado, siempre considerando los precios que ofertan sus competidores. Este hecho permite a los clientes menos fieles a determinadas

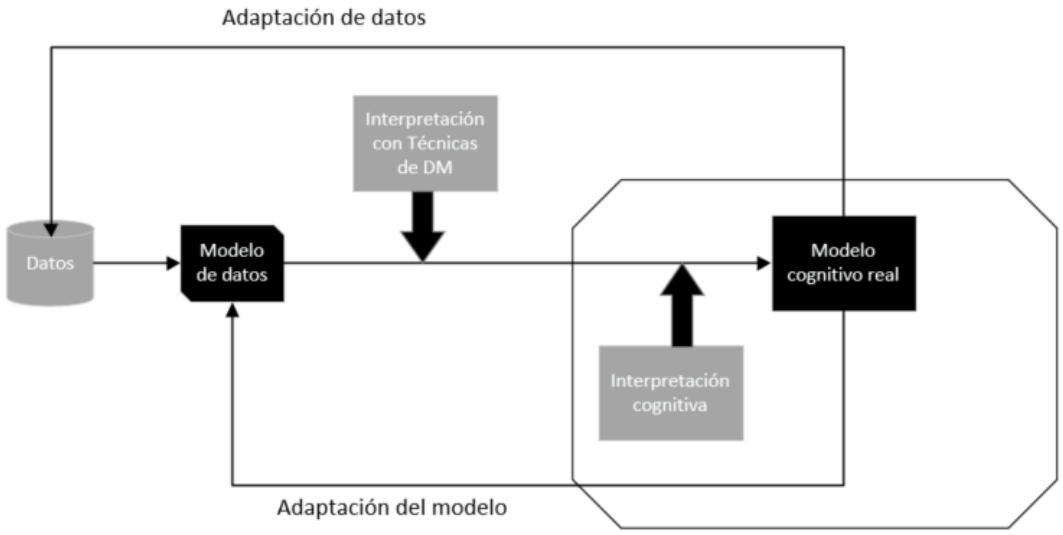

Figura 2.5 Proceso de interpretación para modelos de DM.

marcas u organizaciones la posibilidad de obtener productos más económicos. Un factor importante a tener en cuenta es que los clientes siempre están buscando ofertas, incentivando a una constante guerra de precios. Generalmente, esta guerra [140] sucede en mercados en los cuales las industrias tienen un fuerte enfoque en los competidores, más que en los consumidores. En términos de economía, a este tipo de mercados se le conoce como oligopólicos, como es el caso de la industria de las aerolíneas.

Uno de los principales objetivos de la reducción de precios [113] es debilitar a los competidores disminuyendo su nivel de ventas. Es decir, llevarlo a un nivel tal en el que el otro no pueda continuar la lucha. Así mismo, fijan una estrategia de precios agresiva durante un tiempo determinado con la intención de aumentar sus ventas. Regularmente, estas acciones provocan que los competidores en cuestión también reduzcan sus precios para mitigar la pérdida de clientes. Durante este periodo de tiempo limitado, los consumidores se ven beneficiados, mejorando su ingreso disponible para dedicar a otros productivos o consumir una mayor cantidad del mismo bien. En tanto que las empresas pueden verse severamente afectadas, ya que no necesariamente, una disminución de los precios conlleva al aumento de las ganancias.

Centrándonos en la industria de las aerolíneas, la guerra de precios sucede constantemente debido al número de competidores que existen en el mercado. De esta manera, cada aerolínea es forzada a desarrollar sus propios sistemas o metodologías de monitoreo permanente para

comprender y hacer frente a las estrategias de precios que publican sus competidores. Por lo general, las aerolíneas libran esta batalla desde dos paradigmas diferentes. El primer paradigma se enfocan en el análisis del movimiento de los precios que publican sus competidores, y el segundo, se enfoca en el análisis y optimización de inventarios de acuerdo a los precios que a los precios que han sido publicados y a la demanda que existe en el mercado.

Centrándonos en la industria de las aerolíneas, las series temporales han sido una de las técnicas principalmente utilizadas para analizar los precios que existen en el mercado. *Pitfield et al.* [136] propusieron un modelo de análisis del comportamiento de precios competitivos utilizando tarifas de dos aerolíneas de bajo costo (ULCC). En esta investigación se seleccionaron dos rutas que por sus características son clasificadas como mercados donde los pasajeros en su mayoría viajan debido a negocios y donde compiten más de una aerolínea directamente; aplicando un análisis de correlación cruzada con el objetivo de determinar si existe correlación entre los precios de una aerolínea con respecto a la de sus competidores. *Pets y Rietveld* [134] presentaron una metodología para analizar el comportamiento de precios en la ruta de París a Londres, utilizando tarifas de las aerolíneas Easy Jet y Ryan Air. El objetivo primordial de esta investigación fue responder a la interrogante de sí ambas aerolíneas ajustan sus precios en relación con la otra, es decir, si una de ellas publica un precio la otra reacciona de manera similar, si esta situación es afirmativa se determina que ambas aerolíneas compiten directamente, por lo que se espera que la mayoría de las estrategias de precios entre ambas aerolíneas sean semejantes. Sin embargo, el análisis de series temporales en la guerra de precios puede implicar una bajo conocimiento de los que sucede en el mercado, principalmente debido a la cantidad de restricciones que pueden aparecer en cada una de las tarifas se publican diariamente. *Hofer et al.* [80] presentaron un estudio utilizando análisis de series temporales para analizar los precios de tarifas prémium en mercados de los Estados Unidos, encontrando como resultado que en los mercados donde se compite contra aerolíneas de bajo costo que las aerolíneas las tarifas prémium son más bajas que en los mercados donde no hay aerolíneas de bajo costo. Otros autores han considerado el uso de otras técnicas para incrementar la extracción de conocimiento de las tarifas que existen en el mercado. Al respecto, *Wohlfarth et al.* [172] abordan el problema desde el contexto de los diferentes precios de las tarifas desde la perspectiva de los pasajeros, por lo que diseñaron una metodología de clustering o agrupamiento para la toma de decisiones y selección del mejor precio. Basándose en datos históricos heterogéneos recopilados a través de Internet, describen dos enfoques para pronosticar cambios en los precios de los vuelos en un periodo de tiempo determinado y tomando como variables de entrada una lista de características descriptivas del vuelo, junto con las posibles características que se observan en la evolución de los mismos. Por otra parte, en [104] realizaron un análisis utilizando técnicas de regresión

para comparar el comportamiento de precios en vuelos locales y globales de las aerolíneas Rusas. En los resultados que obtuvieron encontraron grandes diferencias entre las estrategias de precios locales con respecto a las estrategias de precios globales.

Por último, es importante destacar que la mayoría de los trabajos que se han mencionado en este capítulo se centran en utilizar series temporales, enfocándose en el estudio del movimiento de los precios, careciendo de la habilidad para identificar otras características o restricciones que contiene las tarifas aéreas y que son interesantes de determinar por un analista, como por ejemplo, el denominado *advanced purchase* (AP), [1] u otros atributos que se agregan a las tarifas y que son importantes de conocer pues estos determina los precios de cada clase de tarifa.

2.3. Precios dinámicos

En los últimos años, en algunas industrias como las aerolíneas comerciales, hoteles, y empresas eléctricas, donde la oferta de servicios es limitada, y estos servicios suelen ser a corto plazo y perecederos, se han utilizado novedosos métodos de fijación de precios dinámicos (*Dynamic Pricing*, DP). DP se define como la tarea de ajustar los precios de un producto o un servicio continuamente, por lo que, la meta principal de estos cambios de precios es doble: por un lado, las empresas quieren optimizar los márgenes de ganancias y, por otro lado, quieren aumentar sus posibilidades de ventas [46]. Tres factores han contribuido al crecmiento de métodos de DP [8]: la mayor disponibilidad de datos de demanda, la facilidad para cambiar los precios debido a las nuevas tecnologías y la disponibilidad de herramientas de apoyo en la toma de decisiones para analizar datos de demanda y precios dinámicos [45].

En un entorno industrial complejo como es la aviación comercial, donde la competencia tiene un gran impacto en el comportamiento de compra del cliente e impacta en la optimización de las tarifas, es fundamental implementar metodologías de DP, efectivas para estimular la demanda, mitigar la pérdida de clientes y aumentar los ingresos.De acuerdo a la naturaleza de cada industria, DP se puede aplicar de varias formas [47]. La forma más sofisticada es casi instantánea, en la que las aerolíneas puedan identificar a los huéspedes seleccionados que buscan vuelos en sus sitios web para crear ofertas específicas descuentos en tiempo real como incentivo para potenciar las reservas en determinadas rutas. Históricamente, las aerolíneas han utilizado precios estáticos en los que una aerolínea crea su estructura de tarifas utilizando una serie precios discretos y limitados, que son liberados al público en general.

[1] En el cual un usuario tiene acceso a un determinado precio siempre y cuando se realice la reserva del vuelo con un tiempo de anticipado al día que se desea viajar

Cada precio se desarrolla en varios factores como la demanda del mercado, la segmentación de clientes, respuestas competitivas, entre otros. Sin embargo, estas estrategias de precios no son suficientes ya que hay pasajeros sensibles al precio que no han definido una fecha u horario de viaje y tratan de encontrar las tarifas más baratas independientemente de diversas situaciones como pueden ser largas escalas. Otros clientes, que tienen una mayor disposición a pagar (*Willingness to pay*, WTP), podrían considerar factores tales como la hora del vuelo, el día de la semana (*Day of Week*, DoW) o la clase del asiento. Por lo tanto, los analistas de precios e ingresos intentan establecer precios de acuerdo con el tipo de pasajero y su WTP; desafortunadamente, debido a los constantes cambios en el panorama del mercado y a diversas acciones competitivas, como venta de asientos, cambios de horario, agregar o eliminar vuelos, esta segmentación sigue siendo bastante superficial para hacer frente estos problemas, por lo que, en la actualidad hay un aumento en el empleo de método y técnicas de DM y aprendizaje automático (Machine Learning, ML). Gracias a esto, algunas aerolíneas están desarrollando novedosas metodologías para la de construcción de ofertas con la intención de mejorar la segmentación de sus mercados, predecir los precios adecuados en el momento oportuno y extraer conocimiento relevante que les permita comprender el comportamiento de compra de sus pasajeros. Sin embargo, la implementación de un sistema de construcción de ofertas dinámicas no es una tarea fácil de realizar y, de hecho, hoy en día esta construcción de ofertas es rudimentaria [52] y está limitada por las capacidades de los sistemas de distribución, que requieren de un conjunto finito de productos tarifarios con precios fijos o estáticos.

Amazon y Uber son ejemplos de otras industrias que tienen gran éxito al aplicar el ajuste automático de precios mediante el uso de algoritmos DM y ML. Amazon ha sido lider en la aplicacionde metodos de DP durante mucho tiempo [30]; esta empresa es capaz de cambiar el precio de millones de artículos en solo pocos minutos en función de la capacidad de predecir la probabilidad de sus clientes para comprar un artículo; asi mismo, Uber [149], ha implementado un sofisticado algoritmo DP para predecir su demanda basándose en el día y la hora del viaje, siendo capaz de ajustar sus precios de acuerdo a esta demanda. En particular, DP está a punto de convertirse en una de las capacidades centrales que distingue a los ganadores en la industria de las aerolíneas. Por lo tanto, las aerolíneas a pesar de las restricciones tecnológicas, deben implementar métodos de DP capaces de interpretar y extraer patrones útiles para identificar subgrupos interesantes de pasajeros potenciales, que les permita ajustar los precios de acuerdo a sus necesidades.

Volviendo a la industria de las aerolíneas, la necesidad de metodologías de DP automatizadas dentro de la misma es enorme. Se requiere que estas metodologías sean capaces de

generar un modelo reglas interpretables mediante el uso de técnicas de DM capaz de explorar todos los atributos de los datos para permitir el descubrimiento de subgrupos o patrones interesantes y ocultos, con el objetivo de identificar pasajeros potenciales y proporcionar la oferta adecuada, con la finalidad de aumentar sus tasas de conversión. Debido a una serie de anticuadas tecnologías que aún son parte de la industria aérea, la complejidad y la opacidad de los procesos para el análisis y establecimiento de precios ha crecido con el tiempo. Como resultado de esto, los sistemas de gestión de ingresos utilizados por las aerolíneas para la fijación de precios se han convertido en uno de los más arcanos y complejos sistemas de información en el planeta, el cual, a su vez, es un gran componente económico [121]. Los precios de las aerolíneas representan un gran desafío para los análisis económicos modernos porque es tan distante del nivel de análisis de la "ley del precio único". De manera que, DP se ha convertido en una estrategia de precios que tiene como objetivo establecer el precio correcto de acuerdo con ciertas características de los pasajeros potenciales que buscan reservar un vuelo o comprar otro tipo de servicio como una mejora o un complemento, como son los denominados servicios *add-on*.

Existen algunas investigaciones relacionadas con DP en la literatura. *Fiig et al.* [125] introdujeron una serie de estrategias de DP que maximizan ingresos al vender un inventario de artículos idénticos por un precio en un tiempo fijo y donde existe una aerolínea competidora. El modelo implementado incorpora una formulación probabilística de la demanda del cliente, que está influenciado por los precios ofrecidos por la aerolinea y el competidor, y el tiempo restante hasta el final del periodo de venta. *Otero et al.* [129] propusieron un modelo estocástico de DP para resolver el problema de decidir el precio de la tarifa por cada vuelo, que afecta directamente al número de personas que en el futuro intentará comprar un boleto, según el WTP de los pasajeros potenciales. En este trabajo se aplicaron distribuciones de tipo fase y procesos de renovación para modelar el tiempo entre llegadas entre dos clientes que reservan una tarifa y la probabilidad de que un pasajero vaya a comprar un boleto. Este modelo fue probado en un caso real en el que, como resultado, hubo un incremento de reservas hasta un 31 por ciento en promedio de cada vuelo. *Kramer et al.* [103] enfocaron su trabajo en investigar y recomendar varias técnicas para resolver DP incluyendo técnicas de minería de datos y técnicas de análisis predictivas para medir el WTP de los pasajeros. *Shukla et al.* [153] introdujeron y compararon tres enfoques para la fijación dinámica de precios de los misceláneos, con niveles crecientes de sofisticación: (1) se implementa un modelo de dos fases de predicción y optimización utilizando una función de mapeo logístico; (2) se utiliza otro modelo de dos fases combinando una red neuronal profunda de predicción y una técnica de maximización de ingresos mediante una búsqueda exhaustiva discreta; (3) concluye con uno modelo de una sola fase el cual se crea utilizando una red neuronal

profunda de extremo a extremo que recomienda el precio óptimo. En su estudio, *Selcuk et al.* [150] desarrollaron una propuesta de programación dinámica exacta para la gestión de ingresos basada en precios de industria aérea. En este trabajo se realiza una comparación de los resultados con un enfoque alternativo aproximado basado en el uso repetido de un modelo mixto de programación entera y también de métodos intuitivos de discriminación temporal de precios.

2.4. Computación evolutiva

El término computación evolutiva engloba un conjunto de técnicas para solucionar problemas complejos de aprendizaje y búsqueda basadas en la emulación de modelos o comportamientos naturales. Este tipo de computación se inspira en la teoría de selección natural propuesta por Charles Darwin en el siglo XIX; postulando que los organismos vivos evolucionan y se adaptan al entorno con base en mutaciones aleatorias, cruce de miembros de la misma especie, selección y supervivencia de los más aptos [157]. Desde los 1930s la evolución natural ha sido vista como un proceso de aprendizaje, Walter D. Cannon [28] plantea que el proceso evolutivo es algo similar al aprendizaje por ensayo y error que suele manifestarse en los humanos. Alan Mathison Turing en su artículo titulado *"Computing Machinery and Intelligence"* [161] reconoció una conexión "obvia" entre la evolución y el aprendizaje de máquina. A fines de los 1950s y principios de los 1960s, el biólogo Alexander S. Fraser [110] publicó una serie de trabajos sobre la evolución de sistemas biológicos en una computadora digital, dando la inspiración para lo que se convertiría más tarde en el algoritmo genético; incluyendo, entre otras cosas, el uso de una representación binaria, de un operador de cruce probabilístico, de una población de padres que generaban una nueva población de hijos tras recombinarse y el empleo de un mecanismo de selección. De tal forma, que el trabajo de Fraser anticipó la propuesta del algoritmo genético simple de John Holland y la de la estrategia evolutiva de dos miembros de Hans-Paul Schwefel [55]. Fraser además llegó a utilizar el término "aprendizaje" para referirse al proceso evolutivo efectuado en sus simulaciones, y anticipó el operador de inversión, la definición de una función de aptitud y el análisis estadístico de la convergencia del proceso de selección [62].

La computación evolutiva utiliza mecanismos de selección y generación de nuevas soluciones a través de la combinación o cruce de características mostradas en soluciones ya presentes de forma similar a lo que ocurre con los organismos naturales. Dentro de la computación evolutiva se pueden encontrar varios paradigmas como son: la Programación Evolutiva [54], las Estrategias Evolutivas [18], los Algoritmos Genéticos [69], la Programación

Genética [92], la Programación de Expresión Genética [50], la Evolución Gramatical [147], entre otras.

La computación evolutiva tuvo un importante desarrollo durante el siglo XX, diversos investigadores desarrollaron algoritmos inspirados en la evolución natural para resolver diferentes problemas. Por ejemplo, R. M. Friedberg [43] fue pionero en la tarea de evolucionar programas de computadora. El trabajo de Friedberg consistió en generar un conjunto de instrucciones en lenguaje máquina que pudiesen efectuar ciertos cálculos sencillos (por ejemplo, sumar dos números) [55]. George J. Friedman [53] propuso una aplicación de técnicas evolutivas a la robótica en su tesis de maestría en la cual evolucionan una serie de circuitos de control similares a lo que hoy conocemos como redes neuronales, usando lo que él denominaba "retroalimentación selectiva", en un proceso análogo a la selección natural. Friedman [58] también especuló que la simulación del proceso de reproducción sexual (o cruza) y el de mutación nos conduciría al diseño de "máquinas pensantes", remarcando específicamente que podrían diseñarse programas para jugar al ajedrez con este método. Hans Joachim Bremermann [25] fue tal vez el primero en ver a la evolución como un proceso de optimización, además de realizar una de las primeras simulaciones de la evolución usando cadenas binarias que se procesaban por medio de reproducción (sexual o asexual), selección y mutación, en lo que sería otro claro predecesor del algoritmo genético. Bremermann [26] y Bremermann y Rogson [27] utilizaron una técnica evolutiva para problemas de optimización con restricciones lineales. La idea principal de su propuesta era usar un individuo factible el cual se modificaba a través de un operador de mutación hacia un conjunto de direcciones posibles de movimiento. Al extender esta técnica a problemas más complejos, Bremermann y Rogson utilizaron además operadores de recombinación especializados [27]. Bremermann fue uno de los primeros en utilizar el concepto de "población" en la simulación de procesos evolutivos, además de intuir la importancia de la co-evolución [25] , es decir, el uso de dos poblaciones que evolucionan en paralelo y cuyas aptitudes están relacionadas entre sí; visualizando el potencial de las técnicas evolutivas para entrenar redes neuronales [56].

En 1964, Lawrence J. Fogel [57], uno de los primeros profesionales de la metodología de programación genética (*Genetic Programming*, GP), aplica los algoritmos evolutivos para el problema de descubrir autómatas de estado finito. Más tarde, el trabajo relacionado con GP surgió la comunidad de los sistemas de clasificación basados en aprendizaje, la cual desarrolló un conjunto de reglas que describen las políticas óptimas para los procesos de decisión de Markov. La primera declaración de la moderna GP "basado en árboles", es decir, un procedimiento con una estructuración basada en árboles y operadores adecuadamente definidos en algoritmos genéticos (*Genetic algorithms*, GA) fue dada por Nichael L. Cra-

mer [36], este trabajo fue posteriormente ampliado en gran medida por John R. Koza [101], un proponente principal de GP que ha sido pionero en la aplicación de GP en la optimización de diversos y complejos problemas de búsqueda. Gianna Giavelli, un estudiante de Koza, luego fue el pionero en el uso de GP como una técnica para modelar la expresión del ADN. A principios del siglo XXI, Cândida Ferreira [50] propone una técnica evolutiva denominada Programación de Expresión Genética (*Gene Expression Programming*, GEP) en la cada individuo tiene una codificación dual, más adelante en este capítulo se detalla como funciona esta técnica.

Como ya se ha descrito, el término computación evolutiva engloba una serie de técnicas inspiradas en los principios de la teoría de la evolución natural. En términos generales, para simular el proceso evolutivo en una computadora se requiere lo siguiente:

- Codificar una estructura de datos que se utilice para almacenar las soluciones o individuos los cuales se replicarán.
- Operadores genéticos que afecten a los individuos, típicamente, se utilizan el cruce y la mutación.
- Una función de aptitud (*fitness*) que indique qué tan buena es una solución con respecto a las demás.
- Un mecanismo de selección que implemente el principio de ”supervivencia del más apto“ de la teoría de Darwin.

2.4.1. Algoritmos evolutivos

Un algoritmo evolutivo (Evolutionary Algorithm, EA) tiene como propósito genérico guiar una búsqueda estocástica haciendo evolucionar un conjunto de posibles soluciones a un problema, a través de una selección iterativa de las soluciones más adecuadas. La estructura general de un EA aparece en la Figura 2.6. Formalmente se define a los EAs como algoritmos probabilísticos iterativos que mantienen una población de individuos $P(t) = \{x1(t), x2(t), ..., xn(t)\}$ para la iteración o generación t. Cada individuo representa una solución potencial (o parte de una solución potencial) del problema a resolver, la cual se representa mediante una estructura de datos denominada genotipo. Para almacenar el genotipo se han utilizado múltiples estructuras de datos, como vectores de diversos tipos de datos, árboles, grafos, pilas, entre otros. Cada campo del genotipo representa una característica del problema, a la cual se le conoce como gen. El significado de un gen en particular se denomina fenotipo siendo este una solución del problema a resolver [82].

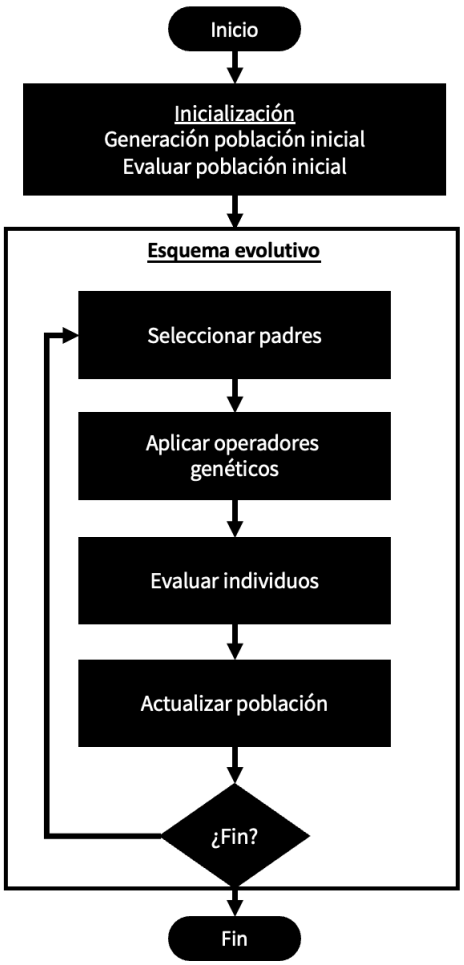

Figura 2.6 Estructura general de un algoritmo evolutivo.

En un proceso evolutivo se evalúa cada solución $xi(t)$, por lo cual, se utiliza una función de aptitud o *fitness* obteniendo un valor numérico para evaluar a cuanto se aproxima ésta a una solución óptima del problema. En la iteración $t+1$ se obtiene una nueva población mediante la selección de los mejores individuos de la población t. A las soluciones o individuos seleccionados se les aplica una serie de operadores evolutivos, que los alteran generando nuevas soluciones candidatas. Después de algún número de generaciones, el algoritmo converge hacia una solución, la cual se encontrara razonablemente cerca de la solución óptima si el algoritmo está bien diseñado.

Un algoritmo evolutivo normalmente analiza cuando se ha obtenido una solución adecuada, es decir una solución óptima o cercana al óptimo, aunque se suele añadir como parámetro un número máximo de generaciones o bien se detiene el algoritmo cuando se observa que las soluciones ya no evolucionan más.

Los operadores genéticos tienen dos finalidades, en ocasiones contrapuestas: por un lado generar variabilidad en la población con el objetivo de ampliar el campo de búsqueda del algoritmo, y por otro lado, tratar de conseguir soluciones cada vez mejores que profundicen en los caminos de búsqueda que sean más satisfactorios. Básicamente existen tres tipos de operadores genéticos:

- **Los operadores de cruce**: que se aplican a dos o más individuos o soluciones.
- **Los operadores de mutación**: que se aplican a un solo individuo o solución.
- **Los operadores de selección**: que aplicados a una población, escogen a un subconjunto de individuos o soluciones.

El cruce tiene como finalidad obtener nuevas soluciones a partir de las características de otras soluciones presentes en la población. Independientemente de la técnica que se utilice tanto para la codificación como para el cruce el objetivo último de este operador es tratar que las características adecuadas para resolver el problema se hereden, y de ser posible se combinen de manera que se potencien entre sí. Con este operador se generan una o varias nuevas soluciones a partir de las características de otras soluciones preexistentes, que se les conoce como padres o progenitores.

Los operadores de mutación realizan una alteración aleatoria de la información contenida en el genotipo de un individuo o solución, para obtener otro individuo denominado mutante. El operador de mutación hace que el algoritmo tenga mecanismos que permitan el desbloqueo del algoritmo si este ha alcanzado un óptimo local en el espacio de búsqueda, en este caso una mutación puede incorporar nuevos genotipos de otras zonas de dicho espacio. Además la implementación de operadores de mutación permite evitar que surjan poblaciones degeneradas, en las que todos o casi todos los individuos tienen el mismo genotipo. Aparte de evitar que se bloquee el proceso evolutivo, el operador de mutación también permite incrementar el número de saltos evolutivos, es decir, la mutación permite explorar nuevos subespacios de soluciones. Sin embargo, si la tasa de mutación es excesivamente alta aparecerá la llamada deriva genética; es decir, la población no converge hacia una solución, sin llegar nunca a un punto estable.

El operador de selección se produce en dos instantes dentro de un algoritmo evolutivo, en primer lugar se seleccionan las soluciones o individuos a los que se van a aplicar los operadores genéticos la cual se le conoce como selección de padres. En segundo lugar, se seleccionan los individuos que formarán la siguiente generación, después de que cada individuo ha sido evaluado y se le ha asignado un fitness. Para llevar a cabo la selección se utilizan una serie de técnicas entre las que se pueden destacar las siguientes:

- **Selección directa**: se seleccionan los individuos de acuerdo a un criterio objetivo, como puede ser los *x* mejores o eliminar los *y* peores.
- **Selección aleatoria**: en el criterio de selección es influido de algún modo el azar. En este tipo de selección se pueden destacar dos técnicas:
  - **Selección por torneos**: en la cual se crea una competición en la que participan un número predeterminado de individuos menor que el tamaño de la población, los cuales son elegidos aleatoriamente; el torneo es ganado por el individuo que posea una mayor aptitud. Se realiza un torneo por cada individuo que sea necesario seleccionar.
  - **Selección por ruleta**: se seleccionan individuos utilizando una metáfora de ruleta en la que se encuentran todos los individuos, ocupando un sector circular con área proporcional a su fitness. La ruleta se gira tantas veces como individuos haya que seleccionar.

Como se ha mencionado anteriormente existen diferentes formas de codificar o representar los algoritmos evolutivos, de tal manera que, una representación tradicional es la binaria (ver Figura 2.7) la cual es ampliamente implementada en los algoritmos genéticos. A la cadena binaria se le denomina *cromosoma*. Al bloque de bits que codifica una sola variable del problema se le denomina *gen* y al calor dentro de cada posición cromosómica se le llama *alelo*.

Otra forma común de codificar las soluciones dentro de los algoritmos evolutivos son las estructuras de árbol que típicamente se implementan en la programación genética (GP) como se muestra en la Figura 2.8. Los árboles pueden ser fácilmente evaluados de forma recursiva. Cada nodo del árbol tiene una función como operador y cada nodo terminal tiene un operando, por lo que las expresiones matemáticas son fáciles de evolucionar y evaluar. Así, tradicionalmente GP favorece el uso de lenguaje de programación que, naturalmente, introduce a través de las estructuras de árbol.

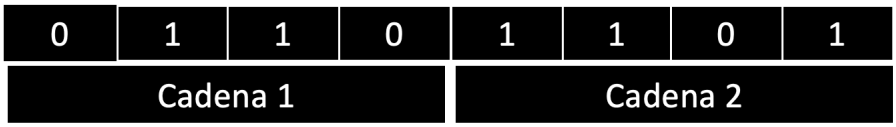

Figura 2.7 Ejemplo de codificación mediante cadenas binarias.

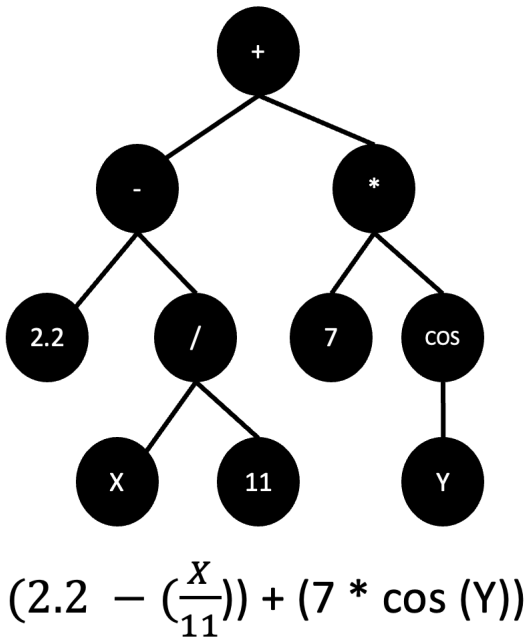

Figura 2.8 Ejemplo de codificación mediante estructuras de árbol.

Finalmente, es importante tener en cuenta que los algoritmos evolutivos son técnicas heurísticas, por tanto, no garantizan que convergerán al óptimo de un problema dado, aunque en la práctica suelen aproximar razonablemente bien el óptimo de un problema en un tiempo promedio considerablemente menor que los algoritmos deterministas.

2.4.2. Programación de Expresión Genética (GEP)

GEP es una técnica evolutiva desarrollada por Cândida Ferreira [50] la cual crea programas o modelos complejos representados por estructuras arbóreas que aprenden y se adaptan por el cambio de sus tamaños, formas y composición, muy parecido a un organismo vivo. GEP es un sistema genotipo-fenotipo, beneficiado de un simple genoma para mantener y trasmitir información genética y un fenotipo complejo para explorar el ambiente y adaptarse. De manera que, cada individuo tiene una codificación dual, el genotipo se compone por un conjunto de genes que se representan mediante una cadena simple de elementos, y el fenotipo correspondiente a cada cadena es un árbol de expresión, formado por funciones en sus nodos no terminales y por valores de entrada o constantes en sus nodos terminales. Aunado a lo anterior, en GEP se propone un modo de transformar las cadenas que representan a los genes en árboles de tal manera que cualquier cadena válida genere un árbol sintácticamente correcto.

En GEP, tanto el genotipo como el fenotipo de los individuos está compuesto de los mismos elementos, siendo diferente la estructura en la que se organizan. Los elementos que aparecen en un individuo son los mismos que aparecen en cualquier otro paradigma de GP, esto son elementos no terminales y elementos terminales:

- **Conjunto de no-terminales**: son funciones que solamente pueden aparecer en los nodos no terminales del árbol de expresión. La aridad de las funciones de cada nodo representará el número de hijos que tendrá el nodo. Como se mostrará a continuación es un factor a tener en cuenta durante el proceso de construcción de árboles válidos.
- **Conjunto de terminales**: es el conjunto de elementos que solamente pueden aparecer en las hojas del árbol. Dentro de este conjunto encontramos tanto valores constantes como parámetros de entrada que reciba el árbol.

El genotipo es una cadena lineal que está formado por varios genes. Cada uno de estos genes se divide en dos partes: la cabeza y la cola. La cabeza del gen tendrá un tamaño elegido *a priori* para cada problema, la cual puede contener elementos terminales y no terminales, pero el tamaño de la cola, solamente puede contener elementos no terminales, esto se determina por la siguiente expresión:

$$t = h(n-1)+1 \tag{2.1}$$

Siendo t el tamaño de la cola, h el tamaño de la cabeza, y n la aridad máxima presente en los nodos no terminales. De esta forma se garantiza que la cola contendrá elementos terminales suficientes como para rellenar el último nivel del árbol de expresión en el peor caso posible. Gráficamente se puede observar en la Figura 2.9 donde se representa un individuo de GEP genérico, siendo las mayúsculas funciones, que se consideran con aridad 3, los valores en minúscula los parámetros y los números que aparecen valores constantes. Como se puede observar, la cola ha de tener un elemento más que el tamaño de la cabeza multiplicado por la aridad máxima menos uno. El objetivo de la limitación de elementos junto con el tamaño de la cola es permitir que cualquier gen pueda ser transformado en un árbol válido, como se aprecia en la Figura 2.9.

El formato en el que se almacena en la cadena-genotipo el árbol de expresión se denomina *K-Expression*. En el cual el modo de construir un árbol válido a partir de la *K-Expression* que lo representa consiste en ir rellenándolo por niveles. De manera que, el primer elemento, será la raíz del árbol. Los siguientes elementos del gen (tantos como indique la aridad del primero) corresponderán a los hijos del nodo raíz, posteriormente se genera el siguiente nivel,

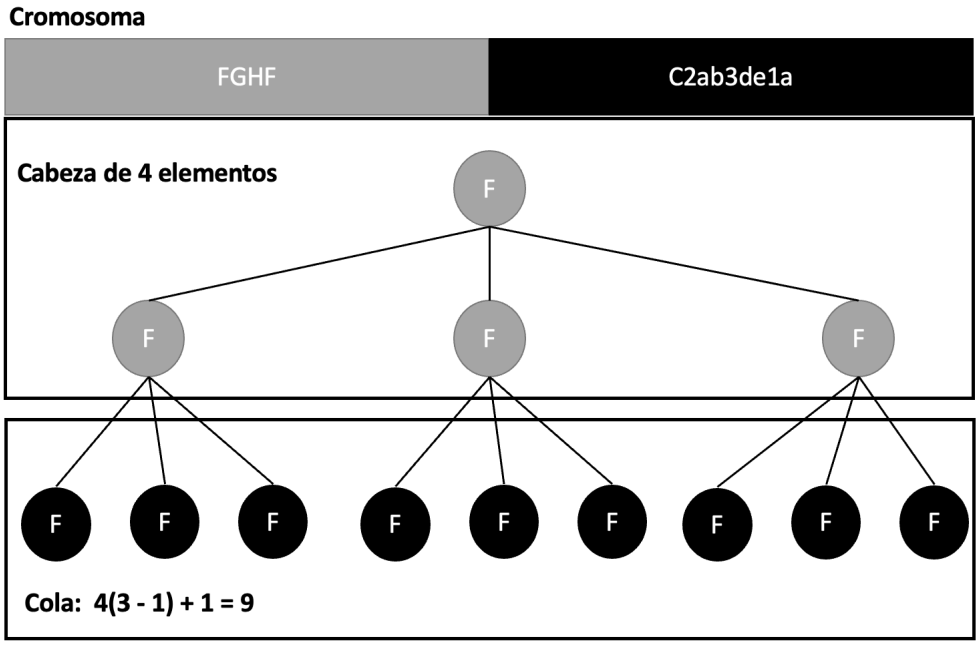

Figura 2.9 Ejemplo de genotipo y fenotipo en GEP.

asignando los hijos que necesite cada nodo según su aridad hasta que todas las hojas de árbol tengan elementos terminales. Al construir el árbol según este procedimiento se garantiza que todos los árboles generados serán válidos, ya que en el peor caso posible habrá elementos terminales en la cola suficientes para completar el árbol.

Por ejemplo consideremos la *K-Expression*: Q*+-abcd. El nodo raíz etiquetado con Q representa a la función raíz cuadrada. El árbol que representa dicha *K-Expression* es el presentado en la Figura 2.10, y la expresión matemática la de la siguiente ecuación:

$$\sqrt{(a+b)*(c-d)} \tag{2.2}$$

Se debe que tener en cuenta que las *K-Expressions* las cuales representan a un árbol pueden ser de tamaño más pequeño que el del gen que las almacena, de hecho puede haber genes distintos que representen la misma expresión matemática. Por ejemplo, si se considera h = 5 y n = 2, cadenas como Q*+-abcd11b o Q*+-abcd12a pueden ser convertidas en el mismo árbol de expresión que la *K-Expression* del ejemplo anterior. Todos los genes cuya *K-Expression* sigan el patrón Q*+-abcd___ almacenan la misma expresión, pudiendo tener en los subrayados (_) cualquier elemento del conjunto de terminales.

Deduciendo del párrafo anterior, el hecho de que un elemento esté en el gen, no implica que aparezca en el árbol. Estos elementos, según Ferreira [51], sirven para aumentar la

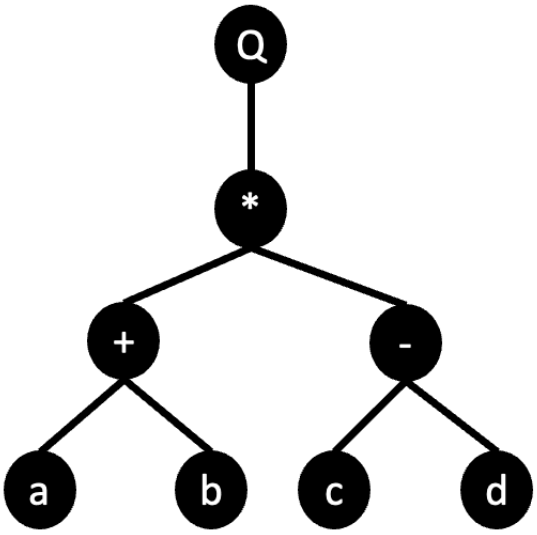

Figura 2.10 Árbol de representación sencilla.

diversidad genética de la población sin que esto implique pérdidas de buenos individuos. Además, estos elementos permiten aumentar la tasa de mutación sin que se produzcan fenómenos de deriva genética.

En caso de que el individuo esté representado por varios genes, su fenotipo consistirá en varios árboles. En determinados problemas esto puede ser adecuado, pero si se desea que cada individuo genere un valor numérico concreto, todos estos árboles deben de conectarse mediante la denominada función de conexión, que recibirá tantos parámetros como genes existan. La función de conexión puede ser un parámetro predeterminado del algoritmo, o bien puede incluirse al principio o al final de los individuos para que evolucione junto con estos. De manera que, el proceso de transformación en este caso se realiza en dos pasos: en primer lugar se transforma cada gen en un árbol, y en segundo lugar se unen todos los subárboles mediante la función de conexión. Por ejemplo, la Figura 2.11 representa a un individuo con tres genes, h = 4 (por tanto t = 4*(3-1)+1) = 9 según la ecuación 2.1) y aridad máxima = 3 (la aridad de las funciones es aridad(I) = 3, aridad(A) = 2 y aridad(N) = 1). En primer lugar cada gen del individuo se transforma en un árbol. Posteriormente los árboles se conectan con una función de conexión I para constituir un único árbol. Todo este proceso aparece recogido en la Figura 2.11.

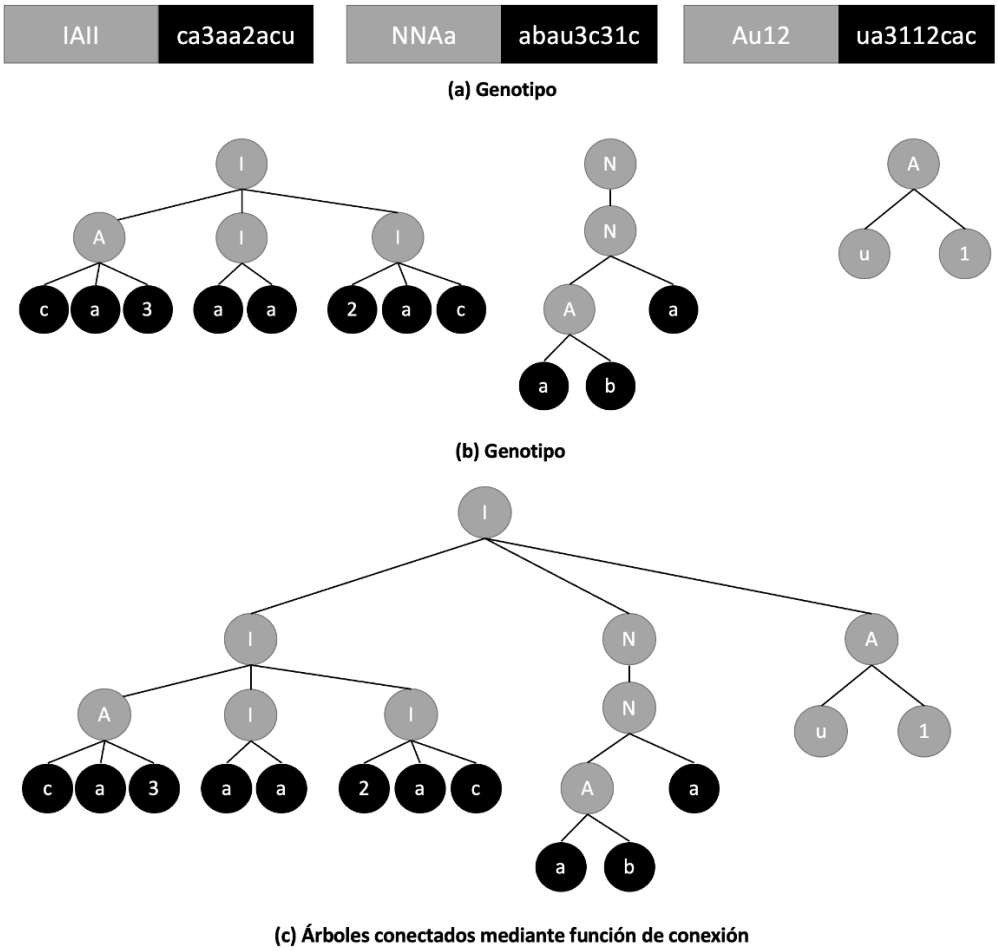

Figura 2.11 Transformación de un individuo con varios genes: tamaño del gen 13 y número de genes 3.

Similar a otros algoritmos evolutivos, el paradigma de la GEP incluye la definición de una serie de operadores genéticos adaptados a la representación dual de los individuos, y formulados de manera que respeten la estructura de los genes. A continuación se comentan los tres tipos de operadores definidos: cruce, transposición y mutación.

En GEP los operadores más comunes de cruce son: la recombinación en un punto, recombinación en dos puntos y recombinación de genes. En todos los casos se escogen aleatoriamente dos padres, y entre ellos intercambian parte de su genotipo para generar dos nuevos hijos, con genes válidos:

- **Recombinación en un punto**: en el cruce en un punto se escoge aleatoriamente una posición del genotipo de los padres. Los hijos resultantes tendrán como genotipo, hasta

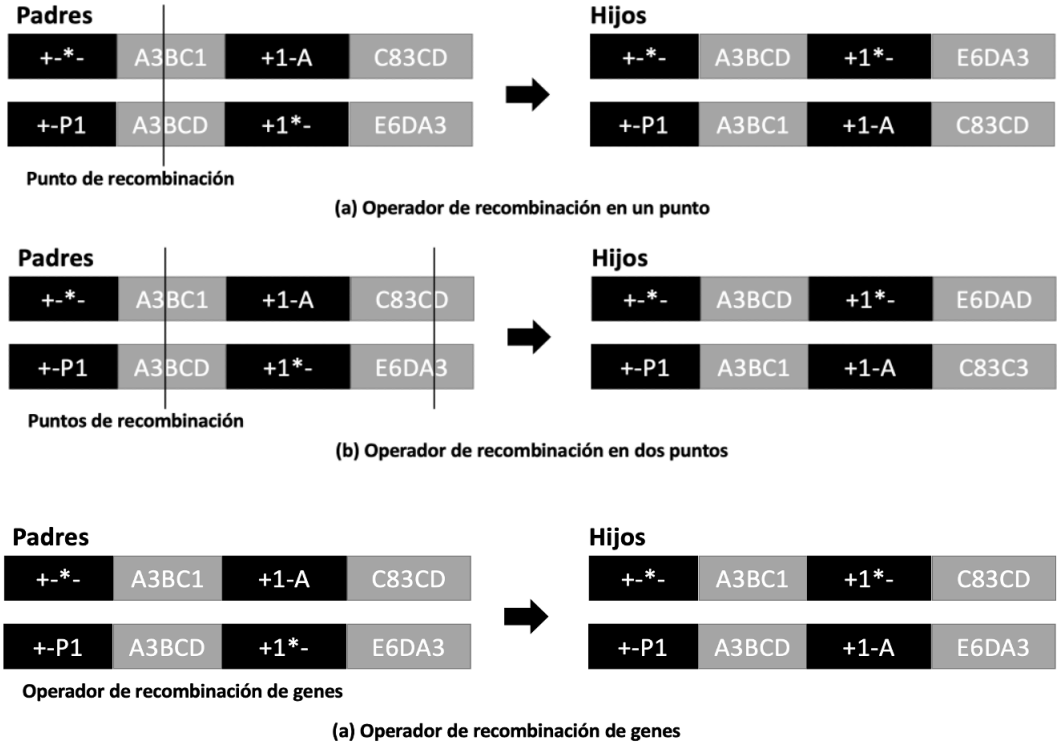

Figura 2.12 Operadores de cruce en GEP.

el punto escogido, el genotipo de un progenitor, y a partir de este, el genotipo del otro progenitor (ver Figura 2.12a).

- **Recombinación en dos puntos**: en esta variante del cruce se escogen dos puntos del genotipo. El genotipo de cada uno de los hijos es alternativamente una copia del cromosoma de un progenitor, hasta el primer punto, una copia de otro progenitor hasta el segundo punto y por último de nuevo copia del primer progenitor (ver Figura 2.12b).
- **Recombinación de genes**: se escoge aleatoriamente un gen del genotipo, y los padres intercambian este para generar los hijos (ver Figura 2.12c). Solamente se puede aplicar a individuos con más de un gen.

El operador de mutación consiste en un cambio aleatorio de un elemento en una posición aleatoria del genotipo, sin embargo, hay que tener en cuenta a la hora de su implementación, que la estructura interna del genotipo ha de mantenerse intacta, esto es, un elemento que pertenezca a la cola del gen solamente cambiará a elementos que pertenezcan al conjunto de terminales, y un elemento que pertenezca a la cabeza del gen podrá alterarse por cualquier

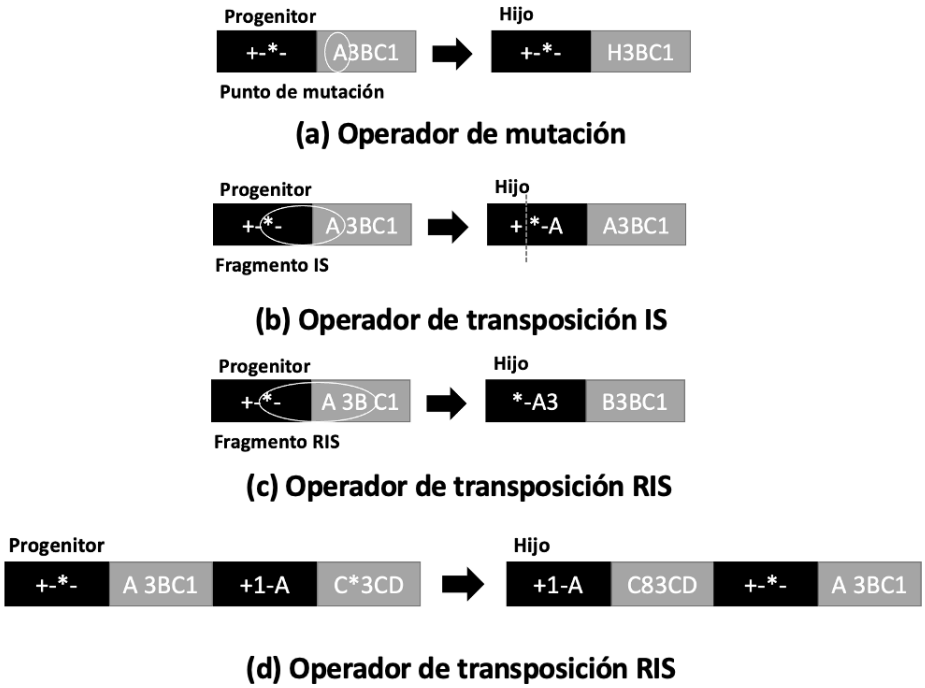

Figura 2.13 Operadores de mutación y transposición en GEP.

elemento del conjunto de terminales y funciones. En la Figura 2.13a se muestra un ejemplo. La mutación de un terminal en una función, o de una función en otra con aridad superior pueden representar un cambio drástico en el fenotipo del individuo, aunque solamente afecte a un elemento. También pueden ocurrir mutaciones neutras que no alteren el fenotipo del individuo.

En la terminología de la GEP [50] aparece un nuevo tipo de operadores, los operadores de transposición, que se corresponden con operadores de mutación modificados para actuar sobre los genes. Los operadores de transposición son un tipo de operadores que seleccionan un fragmento del cromosoma del individuo y lo transfieren a otra posición de este, por tanto podrían catalogarse como operadores de cruce interno. En GEP se definen tres operadores de transposición:

- **La transposición de secuencia de inserción**: también conocida como transposición IS, consiste en seleccionar aleatoriamente un fragmento del cromosoma del individuo, de tamaño también aleatorio, que es insertado en cualquier parte de la cabeza de un gen, salvo en el primer elemento de la cabeza (la raíz del árbol). El fragmento original se copia (se mantiene también en su posición original) y el tamaño del gen no varía (ver Figura 2.13b).

- **La transposición de secuencia de inserción en la raíz**: también conocida como transposición RIS, consiste en seleccionar aleatoriamente un fragmento del cromosoma del individuo, de tamaño también aleatorio, pero con la salvedad de que debe comenzar por una función, el cual es insertado en el primer elemento de la cabeza de un gen, o sea, la raíz del árbol (ver Figura 2.13c).
- **La transposición de genes**: se aplica a individuos con más de un gen y consiste en la modificación de la posición que ocupa un gen en el cromosoma. En esta transposición se escogen aleatoriamente dos genes del individuo, que intercambian sus posiciones en el cromosoma (ver Figura 2.13d).

2.4.3. GEP aplicada a la minería de datos

GEP ha sido aplicado para resolver diversos problemas de optimización dentro de la DM, entre ellos la clasificación. La propia Ferreira en el trabajo en el que plantea el paradigma de GEP [50], apunta a la utilización de esta técnica en tareas de clasificación. Posteriormente, en [51] se sientan las bases para utilizar GEP en problemas de clasificación, utilizando para ello conjuntos de funciones lógicas y esta misma autora resuelve en [49] un problema de clasificación relacionado con el comportamiento de autómatas celulares.

En términos generales, se han utilizado dos enfoques distintos a la hora de abordar problemas de clasificación utilizando GEP, o bien mediante el aprendizaje de reglas o mediante el empleo de funciones discriminantes. Una función discriminante es una función matemática que separa las clases entre sí haciendo uso de un umbral. De esta manera la función recibe como entrada los atributos de un patrón y devuelve una salida numérica. Esta salida se interpreta como la clase a la que pertenece el patrón haciendo uso de uno o varios umbrales. Las funciones discriminantes son bastante robustas y eficientes, pero presentan el problema de que es difícil interpretar el conocimiento que generan. En cuanto a su uso en clasificación mediante GEP merece la pena destacar en esta revisión varios estudios. Uno de los primeros fue el trabajo de *Zhou et al.* [184] donde se abordaron varios problemas de clasificación binaria del repositorio UCI con GEP y conjuntos de funciones aritméticas interpretadas como funciones discriminantes. Las funciones discriminantes se han utilizado también para tratar con problemas multiclase, en [38], utilizando el enfoque de uno-contra-todos. Posteriormente, se han aplicado usando solamente una función discriminante y varios umbrales en [88], donde se trata de buscar los centroides de cada clase utilizando para ello los autovalores de cada clase. El trabajo [176] codifica en un solo individuo varias funciones discriminantes, cada una de ellas asociada a una clase.

Un problema que han tratado de abordar algunos trabajos es la presencia de un gran número de atributos en el espacio de entrada, como en el trabajo [42] en el que los patrones de entrada proceden de DNA microarrays, que se utilizan para la clasificación de moléculas orgánicas. En [95], se propone una modificación del paradigma de la GEP, haciendo un híbrido con un algoritmo de selección clonal. Esta técnica también busca emular un proceso biológico, en este caso el modo en el que el sistema inmunitario clasifica y recuerda las amenazas ante las que se ha enfrentado. En cuanto al uso de reglas de clasificación, los primeros trabajos que tratan de aprender reglas con GEP utilizaron árboles formados solamente por funciones lógicas. Destaca el trabajo antes mencionado [51] en el que los patrones son un automata celular, y se busca clasificar este según su estado final. Otra perspectiva más elaborada es plantear la utilización de funciones lógicas difusas. De esta manera, en [112] se abordan varios problemas de clasificación. Este mismo enfoque se usa en [116] donde cada individuo representa una regla difusa de clasificación, y además se proponen algunos operadores de mutación sobre los atributos difusos. El principal problema del uso de funciones lógicas es que solamente se pueden abordar problemas en que los conjuntos de terminales sean binarios. Para poder trabajar con datos numéricos se han buscado algunas soluciones, como en el trabajo [93] donde cada terminal es un triplete formado por un atributo de entrada, una función relacional y un valor constante. Estos mismos autores proponen en [91] integrar la GEP con un algoritmo celular evolutivo y utilizar ambos combinados como clasificadores débiles en un ensemble mediante la técnica de *boosting*.

GEP también ha sido utilizada para resolver otras tareas de DM, como son las reglas de asociación, *Zuo et al.* [186] introducen un nuevo concepto llamado Predicate Association (PA), proponiendo un nuevo método para descubrir PA por GEP, llamado PAGEP (*mining Predicate Association by GEP*). Los principales resultados son: (1) Se exploran las debilidades inherentes de la asociación tradicional. (2) Se proponen e implementan los algoritmos para extraer PA, decodificar cromosomas y *fitness*. (3) También se demuestra que el procedimiento de decodificación de genes siempre tiene éxito para cualquier gen bien definido. (4) Se realizan extensos experimentos para demostrar que PAGEP puede descubrir alguna regla de asociación que no puede expresarse ni descubrirse mediante el método tradicional. Chen et.al [32] al proponen un algoritmo evolutivo para generar reglas de asociación, implementando un algoritmo de GEP asistido por nichos. Este algoritmo (1) divide a los individuos en varios nichos para evolucionar por separado y fusiona los nichos seleccionados según las similitudes de los mejores individuos para garantizar la dispersabilidad de los cromosomas, y (2) ajusta la función de aptitud para adaptarse a las necesidades de las aplicaciones subyacentes.

2.4.4. Evolución Gramatical (GE)

GE fue propuesta por Ryan, Collins y O'Neill [130] en 1998. Surge como un algoritmo evolutivo capaz de generar automáticamente programas escritos en un lenguaje de programación. El proceso de generación de programas se divide en la evolución de los genotipos de longitud variable y su correspondiente transformación en los fenotipos, que son los programas propiamente dichos. Para realizar esta conversión, cada genotipo representa una secuencia de elecciones en las reglas de producción de una gramática libre de contexto representada en la notación de *Backus-Naur* (BNF). Este procedimiento de generación de programas asegura que siempre son sintácticamente correctos. Por lo tanto, GE se desarrolla en dos partes, de forma similar a los algoritmos genéticos. En primer lugar, existen un conjunto de genotipos, que son evolucionados (como en los algoritmos genéticos). Estos genotipos generan nuevos genotipos, que son transformados en sus correspondientes fenotipos; los cuales son evaluados, asignando un *fitness* a cada uno de los genotipos. En base a esto, se producen nuevas poblaciones.

GE no realiza el proceso evolutivo en los programas reales, lo hace en cadenas binarias de longitud variable, empleando un proceso de mapeo para generar programas usando cadenas binarias para seleccionar reglas de producción en una definición de gramática Backus Naur Form (BNF). El resultado es la construcción de un programa sintácticamente correcto a partir de una cadena binaria que puede luego ser evaluada por una función de aptitud. De manera que, GE se desarrolla en dos partes, en primer lugar, existen un conjunto de genotipos, que son evolucionados de forma similar a los algoritmos genéticos. Estos genotipos generan nuevos genotipos, que son transformados en sus correspondientes fenotipos; los cuales son evaluados, asignando un fitness a cada uno de los genotipos. En base a este proceso se generan nuevas poblaciones.

Backus Normal Form o *Backus-Naur Form* (BNF) es una técnica de notación para gramáticas libres de contexto, la cual se usa frecuentemente para describir los lenguajes de programación. La Evolución Gramatical utiliza una representación BNF para realizar la conversión del genotipo al fenotipo. Así, BNF es una notación para expresar la gramática de un lenguaje en forma de reglas de producción. Las gramáticas BNF consisten en terminales, que son elementos que pueden aparecer en el lenguaje, por ejemplo, $+$, $-$, entre otros, y no terminales, que se pueden expandir en una o más terminales y no terminales. Una gramática se puede representar mediante la tupla N, T, P, S, donde N es el conjunto de no terminales, T representa el conjunto de terminales, P representa el conjunto de reglas de producción que mapea los elementos de N a T, S es el símbolo de comienzo el cual es un miembro de N.

Cuando existen un cierto número de reglas a producir que se pueden aplicar a un elemento de N la elección está delimitada con el símbolo '|'. Por ejemplo:

$$N = \{exp, op, pre-op, var\}$$

$$T = \{sin, cos, +, -, /, *, X, 1, 0, (,)\}$$

$$S = exp$$

el cual se representa como:

```
P = {S = <expr>;
     <expr> =  <expr> <op> <expr> | ( <expr> <op> <expr> ) | <pre-op> ( <expr> ) | <var>;
     <op> = + | - | / | *;
     <pre-op> = sin | cos;
     <var> = X | 1.0}
```

Figura 2.14 Ejemplo de una gramática de libre contexto.

A diferencia del enfoque de [6], no se hace distinción en esta fase, entre las funciones (los operadores en este sentido) y terminales (variables en este ejemplo); sin embargo, esta distinción es una de implementación detalla más que un problema de diseño. Whigham [169] también notó la posible confusión con la terminología y usó los términos *GPFunctions* y *GPTerminals* para mayor claridad.

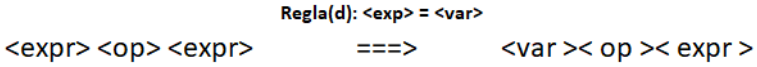

Figura 2.15 Ejemplo de una regla de producción.

En muchos de los ejemplos que siguen, los individuos de una población están representados como pasos de derivación. Un paso de derivación es simplemente la aplicación de la producción de una regla, P, a un elemento del conjunto no terminal, N, como se muestra en la Figura 2.15, la cual muestra un ejemplo de un paso de derivación en el que se puede observar la aplicación de la regla Regla (d): $< expr >=< var >$, el elemento que se encuentra el parte de la izquierda es un elemento del conjunto de no terminal ($< expr >$). Esto da como resultado que $< expr >$ sea reemplazado por $< var >$.

La Figura 2.16 muestra un ejemplo un árbol de derivación el cual representa un individuo. Este árbol de derivación puede ser convertido fácilmente en el formato adoptado por la GP el cual se conoce como un árbol de análisis, el cual se muestra en la Figura 2.17. Sin embargo, es más difícil convertir un árbol de GP en un árbol de derivación equivalente, ya que el proceso suele ser no determinista.

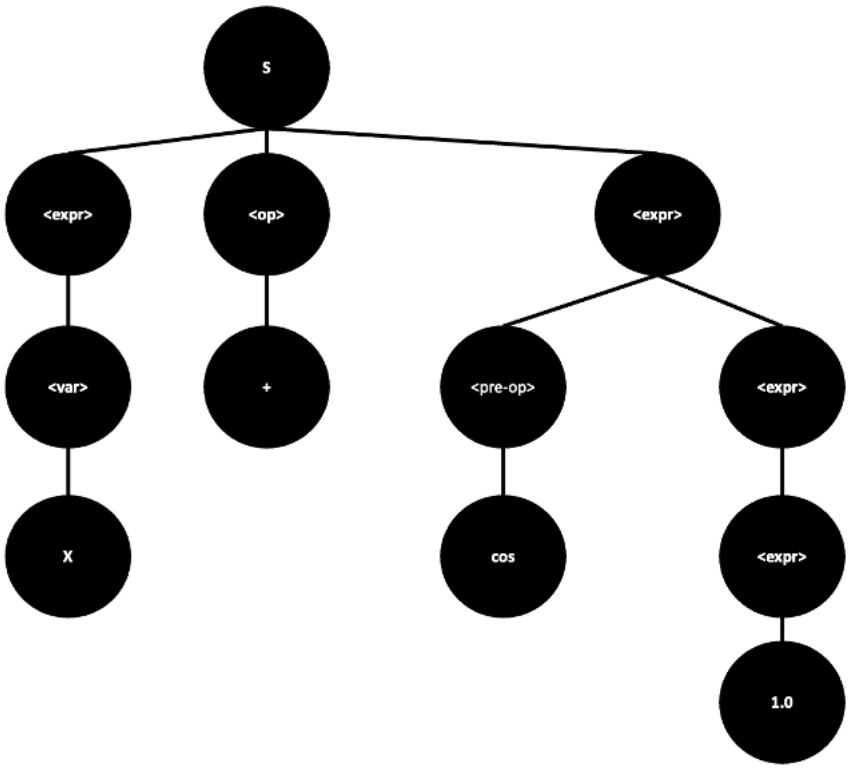

Figura 2.16 Ejemplo de un árbol de derivación de un individuo basado en una gramática evolutiva.

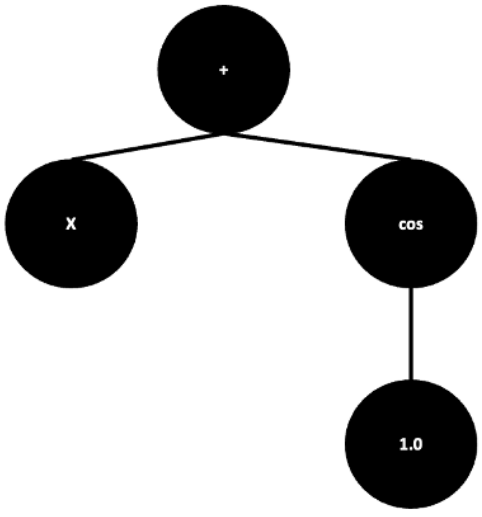

Figura 2.17 Ejemplo del árbol de derivación de la Figura 2.16 convertido a su equivalente árbol de análisis.

2.4.5. GE aplicada a la minería de datos

GE ha sido ampliamente utilizada en diversos dominios del conocimiento. El área las finanzas fue uno de los primeros dominios de su aplicación. Desde el primer estudio de este tipo en 2001, se han publicado más de 100 estudios que emplean a GE para una amplia gama de propósitos que abarcan el comercio financiero, el modelado de riesgo crediticio, la gestión de la cadena de suministro, la detección de incumplimiento fiscal y el modelado de estrategia corporativa [22]. Un ejemplo de lo anterior es el trabajo de Brabazon et al. [23] en el cual examinaron el potencial de la GE para descubrir una serie de reglas útiles que ayudaran a predecir el fracaso corporativo utilizando información extraída de estados financieros. Implementando una metodología de GE capaz de descubrir una estructura útil en la información de índices financieros que se puede usar para predecir fallas.

Mauceri et al. [119] utilizan un método de GE para extraer características de series temporales de acelerómetros con el fin de aumentar el rendimiento de un clasificador de estimación de densidad de kernel. Las series temporales se recopilan a través de nueve acelerómetros de muñeca asignados a otros tantos sujetos. El objetivo es distinguir cada tema de todos los demás en un marco de clasificación de una clase. Las soluciones evolucionadas de GE, denominadas extractores de características, se analizan minuciosamente. Cada solución es una función capaz de apuntar a una subsecuencia específica de una serie de tiempo y reducirla a un solo escalar. De esta forma, una serie de tiempo larga se puede resumir en un número arbitrario de características. Como se mencionó anteriormente, una de las tareas principales y fundamentales de la minería de datos es la inducción automática de reglas de clasificación a partir de un conjunto de ejemplos y observaciones. De manera que, *Mazouni et al.* [120] propusieron un sistema de programación genética gramatical automática (AGGP) que puede desarrollar códigos completos de programas en Java. Estos códigos representan un algoritmo de inducción de reglas que emplea una técnica de GE a través de una gramática definida en BNF asignada a un programa. Para realizar esta tarea, utilizaron cadenas binarias como entradas para el mapeador junto con la gramática de BNF. Tales cadenas binarias representan posibles soluciones potenciales resultantes del componente inicializado y los bloques de construcción de la herramienta Weka, esto facilitaría el proceso de inducción y acortaría los programas inducidos. Geourgolas et al. [68] presentaron un método novedoso basado en GE para discriminar fetos acidémicos de los normales, empleando características extraídas de la señal de la frecuencia cardíaca fetal durante los minutos inmediatamente anteriores al parto. El método propuesto identifica correlaciones lineales y no lineales entre las características extraídas originalmente, construyendo un conjunto de nuevas características que, a su vez, alimentan un clasificador no lineal.

Boutorh et al. [21] introdujeron un algoritmo denominado GEARM, el cual se enfoca en descubrir reglas de asociación empleando GE, con el objetivo de identificar variaciones en las secuencias de ADN de la genética humana, para determinar si existe un aumento o disminución de una determinada enfermedad. *Noaman et al.* [126] proponen algoritmo de descubrimiento de subgrupos basado en gramáticas de GE para extraer conjuntos de elementos y relaciones que representan cualquier tipo de homogeneidad y regularidad en los datos de un contexto supervisado, este algoritmo funciona como un sistema de recomendación para la adecuada toma decisiones por parte de estudiantes de bachillerato y poder decidir cuál es la licenciatura que deben elegir para ingresar a la Universidad, en base a sus habilidades.

2.5. Conclusiones del capítulo

En la primera parte de este capítulo se introdujeron los conceptos y tareas más importantes de DM. Se presentó el proceso KDD y el problema de la extracción de patrones, así como la importancia de la interpretabilidad. Se estableció la diferencia entre tareas y métodos y su correspondencia. Se presenta una descripción detallada del funcionamiento de los métodos basados en reglas.

En la segunda parte de este capítulo se muestró la importancia de la guerra de precios y como el desarrollo tecnológico en las últimas dos décadas, así como el aumento de las aerolíneas de bajo costo han hecho esta guerra precios un factor determinante para el éxito o fracaso de las aerolíneas en general. Se ha mostrado cómo las aerolíneas que no están dispuestas a competir pueden sufrir una pérdida de ingresos significativa. Un ejemplo de esto es Aeromexico, la cual era la aerolínea más importante en el mercado local e internacional en México, y que con la llegada de aerolíneas de bajo costo como Volaris y Viva Aerobus, aunado a una pobre estrategia competitiva, han causado una perdida significativa de pasajeros, siendo ahora Volaris la aerolínea que trasporta más pasajeros a nivel local [106]. Después se introdujo el paradigma de precios dinámicos y su importancia actual en diversos tipos de industria. Se describió el concepto de WTP y cómo diversas compañías aéreas han desarrollado diversas metodologías para incentivar el ingreso de ganancias mediante la utilización de este paradigma.

En este capítulo finalizó con una introducción al mundo de la computación evolutiva, donde se muestraron los pasos importantes de su desarrollo a través de la historia, comenzando por la teoría de selección natural de Darwin postulada en el siglo XIX, la cual sirve como base para la implementación de los primeros algoritmos que se desarrollaron en el siglo XX. Así mismo, se han presentado los conceptos básicos del Neo-Darwinismo y como

han inspirado el desarrollo de diversos paradigmas evolutivos como son: los algoritmos genéticos (GA), la programación genética (GP) y sus variantes como son: la programación de expresión genética (GEP) y la programación de gramáticas evolutivas (GE). Siendo GEP y GE dos de las técnicas que se utilizan en las propuestas que se presentan en los capítulos 3 y 4 de esta memoria de tesis.

Capítulo 3

Modelo de precios competitivos para enfrentar la guerra de tarifas aéreas

Cuando se habla de la industria de las aerolíneas, la fijación de precios se refiere al proceso de determinación de las clases de tarifas, junto con diferentes productos, servicios y restricciones en un origen y un mercado de destino (O-D). Cada punto de precio lanzado al público está vinculado a una clase de tarifa específica que se identifica por códigos de tarifa de una letra establecida por cada aerolínea. Para obtener ingresos, las aerolíneas ofrecen billetes en diferentes clases de tarifas para cada uno de sus vuelos, sin que los pasajeros tengan constancia de que existe una subdivisión de clases de tarifas. Además, estas clases de tarifas se relacionan directamente a lo que se conoce como la clase de servicio como son: clase económica, clase premium, clase ejecutiva y primera clase. Esta estructura desconocida de las clases de tarifas se basa tanto en el servicio, los misceláneos y sus restricciones [131].

En los últimos años, está teniendo lugar una constante guerra de precios [131] debido al rápido crecimiento de las aerolíneas de bajo coste, con clases de tarifas menos restringidas. Varias veces al día, las aerolíneas tienen que clasificar las tarifas de otras aerolíneas y colocarlas dentro de su propia estructura de sus clases de tarifas, el cual es un proceso clave para mantener la competitividad en el mercado [74]. Por lo tanto, los equipos de ingresos y precios pasan una cantidad de tiempo considerable analizando y tratando de interpretar las acciones realizadas de sus competidores. Este proceso es extremadamente complejo debido al alto número de restricciones y puntos de precio que se añaden a cada clase de tarifa del mercado; derivando como resultado, en comprensión parcial de las acciones reales tomadas por otras aerolíneas, y existiendo el riesgo tanto de pérdida de información como de generación de información inútil.

Debido a lo anterior, se propone un algoritmo de programación de expresión génica (GEP) [11], [47]. Diversos algoritmos de GEP han sido aplicados en problemas del mundo real [86], [152], [182], por su sencillez de codificación y su versatilidad para explorar enormes espacios de búsqueda. El algoritmo de GEP propuesto funciona como un algoritmo de aprendizaje de atributos (*Feature Learning*, FL) el cual produce atributos (métricas) usualmente utilizadas en los análisis realizados por los equipos de precios tales como: la media, la tarifa más baja, la moda, y la desviación estándar. Por lo tanto, se forman nuevos conjuntos de datos para mejorar la predicción de algoritmos de clasificación clásicos sin aumentar el número de reglas en los modelos que generan. Los resultados de los experimentos llevados a cabo con 18 algoritmos de clasificación, y considerrando datos reales de una aerolínea, revelaron diferencias estadísticas en términos de precisión, la métrica F1 e interpretabilidad cuando se aplica el algoritmo propuesto.

La novedad de este trabajo se describe a continuación:

- Se propone un algoritmo de programación de expresión génica (GEP) que imita un proceso de aprendizaje y modificación de características de las tarifas que desempeñan los equipos de precios cotidianamente para la extracción de reglas significativas.
- Se propone una metodología automatizada para la extracción de reglas y obtener un modelo de alta interpretabilidad cuando existen cambios en el mercado debido al lanzamiento de tarifas de una aerolínea.

Como se menciona anteriormente, La metodología propuesta para la predicción e interpretación de tarifas aéreas para enfrentar la guerra de precios, que incluye el algoritmo de GEP (ver Figura 3.1), incluye cinco diferentes pasos que se describen a profundidad en la siguiente sección: recopilación y pre-procesado de datos, aprendizaje de atributos (*Feature Learning*, FL), clasificación e interpretación de reglas.

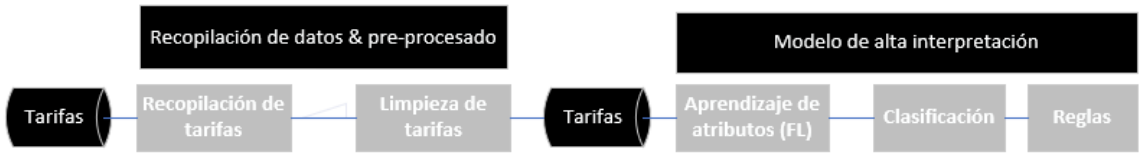

Figura 3.1 Propuesta de metodología para la extracción de un modelo de alta interpretación.

3.1. Recopilación de datos y pre-procesado

En este primer paso de recopilación de datos y pre-procesado se aplica automáticamente un proceso de cotización de tarifas directamente en un base de datos donde diversas aerolíneas publican sus tarifas. Para efectos de esta memoria de tesis, solo se consideran tarifas publicadas por la aerolínea Air Canada, con veinte fechas de captura diferentes entre los meses de diciembre de 2019 y enero de 2020; en un período de viajes entre los meses de diciembre de 2019 hasta abril del 2020. Estos datos crudos conforman 3 GB de archivos de texto (ver Figura 3.2) los cuales contienen información de diez mercados locales del territorio canadiense, un total de más de ciento cincuenta fechas de viaje y los atributos siguientes:

- **Origen (ORG)**: aeropuerto o ciudad donde se inicia un vuelo.
- **Destino (DES)**: aeropuerto o ciudad donde termina un vuelo.
- **Código base de tarifa (*Fare Basis Code*, FBC)**: código alfanumérico que resume las restricciones de las tarifas.
- **Tarifa (*Fare*)**: el precio base de la tarifa.
- **Boleto de viaje (*Travel-ticket*)**: es el último día que la tarifa puede ser reservada al precio mostrado y con las mismas restricciones.
- **Compra avanzada (*Advance Purchase*, AP)**: el número de días antes del inicio del viaje en el que se aplica la tarifa con el mismo precio y mismas restricciones.
- **Estadía mínima y máxima**: esta restricción aplica para tarifas de ida y vuelta, para efectos de esta memoria de tesis solo analizamos tarifas de ida, por lo tanto, estas restricciones no se aplican.
- **Enrutamineto (Routing, RTG)**: número de enrutamiento.
- **Fecha de viaje**: fecha en que aplica una tarifa.

Debido a que los atributos mencionados anteriormente se reciben en formato de texto se requiere un pre-procesamiento de datos para transformarlos al formato de entrada correcto. De manera que, algunas tareas específicas de limpieza y de pre-procesamiento se llevan a cabo en esta etapa, con la finalidad de preparar todos los datos descritos anteriormente. Esta etapa es fundamental para poder llevar a cabo de forma correcta el siguiente paso en la metodología propuesta. Por lo tanto, la limpieza de tarifas es un proceso extensivo el cual

```
YTO-NYC       CXR-AC       THU 14DEC19                    CAD
THE FOLLOWING CARRIERS ALSO PUBLISH FARES YTO-NYC:
AA AS CO DL JJ JU LA NW PD PK UA US WS
//SEE FQHELP FOR INFORMATION ABOUT THE NEW FARE DISPLAYS//
ALL FEES/TAXES/SVC CHARGES INCLUDED WHEN ITINERARY PRICED
SURCHARGE FOR PAPER TICKET MAY BE ADDED WHEN ITIN PRICED
AC     YTONYC          14DEC19
    V FARE BASIS     BK    FARE   TRAVEL-TICKET AP  MINMAX  RTG
  1   WNA7A0TG       W X   192.00        T12JA  7/1  -/  -  636
  2   VNA7A0TG       V X   219.00        T12JA  7/1  -/  -  636
  3   WNA7A0FL       W X   242.00        T12JA  7/1  -/  -  636
  4   QNA3A0TG       Q X   251.00        T12JA  3/1  -/  -  636
  5   VNA7A0FL       V X   269.00        T12JA  7/1  -/  -  636
  6   HNA3A0TG       H X   292.00        T12JA  3/1  -/  -  636
  7   QNA3A0FL       Q X   301.00        T12JA  3/1  -/  -  636
  8   WNA7A0CO       W X   302.00        T12JA  7/1  -/  -  636
  9   VNA7A0CO       V X   329.00        T12JA  7/1  -/  -  636
 10   HNA3A0FL       H X   342.00        T12JA  3/1  -/  -  636
 11   UNA0A0TG       U X   346.00        T12JA  -/‡  -/  -  636‡
```

Figura 3.2 Tarifas en formato de texto.

elimina varios caracteres y convierte los datos en formato de texto a un formato tabular. Así mismo, elimina algunos atributos inutilizables los cuales a su vez determina los valores de los atributos FARE y AP como valores numéricos. De manera que, en esta etapa se extraen automáticamente cinco atributos de los archivos de texto: ORG, DES, FARE, AP y RTG. Adicionalmente se integran los atributos **día del vuelo** (*Travel Day*, TrDay), **mes del vuelo** (*Travel Month*, TrMonth) y **año del vuelo** (*Travel Year*, TrYear), los cuales se obtienen de dividir el atributo **fecha de viaje**. Un cuarto atributo es creado de la unión de los atributos ORG y DES extraídos de los archivos de texto el cual es nombrado **mercado directo** (*Direct Market*, DMKT). Un quinto atributo denominado **día de la semana** (*Day of Week*, DOW) se crea a partir de la **fecha de viaje**. Se continúa agregando el atributo denominado **temporada** (Season), por lo que, previamente se realizó un análisis de los FBC publicados por la aerolínea Air Canada con el objetivo de identificar la temporada en que aplican las tarifas; por lo tanto, el proceso automático lee los resultados de este análisis y asigna la etiqueta de temporada para cada clase de tarifa, las cuales son tres **L** para temporada baja, **H** para temporada alta y **Q** para lo que se conoce como temporada super-alta o super-pico. Finalmente, se integra el attributo **Clase** el cual se crea utilizando el primer carácter del atributo **FBC**.

El proceso termina filtrando el atributo **Clase** considerando solo las cinco primeras clases con base en la estructura de tarifas publicadas por Air Canada (**K, A, L, T, S**). Estas clases de tarifas se seleccionan de acuerdo a los precios, comenzando desde el valor más bajo hasta los cuatro siguientes en orden consecutivo; por ejemplo, la clase **K** tiene un valor de $50, la clase **A** un valor de $70, la clase **L** un valor de $100 y así sucesivamente. Usualmente en la industria de las aerolíneas, las clases de tarifas se ordenan por precios y después de ciertos niveles, dejan de competir con las otras aerolíneas según el desempeño cada ruta o

DMKT	FARE	ORG	DES	AP	RTG	TrDay	TrMonth	TrYear	DOW	Season	Class
YHZYZZ	98	YHZ	YYZ	30	R-900	27	Feb	2020	Thu	A	K
YULYYZ	166	YUL	YYZ	14	R-905	7	Jan	2020	Tue	L	S
YEGYZZ	98	YEG	YYZ	30	R-900	27	Feb	2020	Thu	Q	L
YVRYZZ	221	YVR	YYZ	21	R-900	7	Jan	2020	Tue	L	T
YVRYYC	129	YVR	YYC	30	R-900	5	Apr	2020	Sun	H	L
YYCYYZ	187	YYC	YYZ	14	R-5	20	Jan	2020	Wed	A	A
YYCYZZ	226	YYC	YYZ	21	R-900	15	Apr	2020	Wed	Q	T
YYCYYZ	187	YYC	YYZ	14	R-5	24	Jan	2020	Fri	A	A

Tabla 3.1 Ejemplo del conjunto de datos de entrada en formato tabular

mercado, siendo esta la razón por la cual se seleccionan las primeras cinco clases. Al terminar el proceso los nombres de los atributos son modificados a un formato más corto.

Al final de esta etapa, se crea un conjunto de datos el cual contiene un total de doce atributos y cinco clases de tarifas diferentes (ver Tabla 3.1). Con relación al atributo *DMKT* el cual es la unión de los atributos *ORG* y *DES*, se ha determinado mantener los tres atributos por la posibilidad de obtener reglas que podrían ser aplicadas para cada uno de estos atributos. De hecho, en la práctica, existen rutas que tienen estructuras de tarifas iguales a las cuales se les conocen como rutas de tarifa común. Un ejemplo de estos casos podría ser que la ruta entre Montreal y Toronto tenga una estructura de precios igual a la ruta entre las ciudades de Toronto y Ottawa; por lo tanto, al mantener los tres atributos como parte del conjunto de datos los algoritmos de clasificación podrían encontrar reglas significativas y exclusivas para estos atributos, a su vez, sería interesante obtener reglas que puedan identificar una serie de rutas donde su estructura de precios es igual o muy similar.

3.2. Modelo de alta interpretación

Debido al alto volumen de tarifas en el mercado y a los constantes cambios que suceden en el mercado diariamente, los cuales se generan debido a la guerra de precios; los analistas de las aerolíneas pasan un tiempo considerable ejecutando procesos manuales de modificación y aprendizaje de atributos que les permita analizar de manera eficiente dichas estrategias. De manera que, en esta etapa de la metodología se propone la implementación de un algoritmo de GEP [182], el cual imita estos procesos manuales para el aprendizaje e integración de nuevos atributos de forma automática. GEP se ha aplicado para resolver diversos problemas [182], presentando sencillez en la codificación y la versatilidad para explorar grandes espacios de búsqueda. De manera que, la finalidad del algoritmo de GEP que se propone en esta memoria de tesis es imitar el proceso de aprendizaje de atributos que los equipos de precios dentro de una aerolínea realizan diariamente. Este proceso se realiza seleccionando, agrupando

y modificando algunos de los atributos para adaptar los conjuntos de datos antes de que sean introducidos a los algoritmos de clasificación. El objetivo principal es proporcionar los mejores atributos para que el algoritmo de clasificación pueda encontrar una serie de reglas que permitan una alta interpretación y comprensión de lo que sucede en el mercado. Los siguientes son los pasos que sigue el proceso de aprendizaje propuesto:

- **Codificación**. El algoritmo de GEP propuesto codifica individuos como cadenas simbólicas (longitud fija), que son luego expresados como entidades no lineales de diferentes tamaños y formas (árboles de expresión) [123], considerando el conjunto de funciones formado por nueve atributos del conjunto de datos $F = \{$ORG, DES, AP, RTG, TrDay, TrMonth, TrYear, DOW, Season$\}$, y el conjunto de terminales (métricas) que generalmente son utilizadas por los equipos de precios para el análisis de tarifas $T = \{$Mean, Lowest Fare File (LFF), Mode, Standard Deviation (SD)$\}$. En este punto, cabe destacar que los atributos *DMKT, FARE* y *Class* fueron excluidos del conjunto de funciones, ya que permanecen como atributos fijos o constantes en cada una de las soluciones. El atributo *FARE* está excluido del conjunto de funciones porque cada solución necesita este atributo para calcular la métrica seleccionada del conjunto de terminales. Los atributos *Class* y *DMKT* también se excluyen, el primero para preservar la clase de cada instancia de los nuevos conjuntos de datos y; el segundo, para mantener una buena interpretabilidad cuando se obtienen los resultados de clasificación. En referencia al conjunto de terminales, el uso de la tarifa media (Mean), o su desviación típica (SD), puede ser útil para encontrar estrategias de tarifas especiales que se aplican en algunos de los mercados. La tarifa más baja publicada (*Lowest Fare Filed*), la cual es la tarifa más baja disponible al público por cada clase, puede identificar descuentos específicos en determinadas fechas. Por último, pero no menos importante, la métrica moda (*Mode*) podría identificar estrategias similares o iguales en diversas tarifas que se publican constantemente.

  La Figura 3.3 representa un ejemplo de la codificación de un cromosoma creado por la propuesta que se expone en este capítulo. Los genes variables son los atributos del

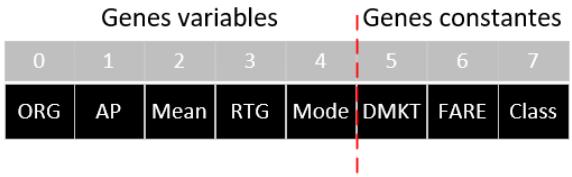

Figura 3.3 Ejemplo de un cromosoma

conjunto de datos y las métricas que han sido seleccionados por un proceso aleatorio y que se utilizará para modificar el conjunto de datos original. Los genes constantes son los que ya se han mencionado anteriormente, los cuales serán parte de cada una de las soluciones generadas por el algoritmo. Este algoritmo tiene como objetivo integrar un proceso de aprendizaje de atributos leyendo y codificando todos los genes variables en el cromosoma. Regresando a la codificación del ejemplo que se muestra en la Figura 3.3, se definen el primer y segundo gen como elementos del conjunto de funciones. De manera que, el algoritmo actúa como un selector de atributos para el primer gen (atributo ORG) que formará parte del nuevo conjunto de datos sin ninguna modificación a sus valores. Después de eso, el algoritmo revisará los dos genes siguientes que en este ejemplo son el atributo AP y la métrica *Mean*. Por lo tanto, el algoritmo agrupará el conjunto de datos original mediante el uso del atributo AP y los dos atributos constantes: DMKT y Class para calcular la media (*Mean*) utilizando el atributo FARE, de tal forma que crea un nuevo atributo. Este proceso se explica con lujo de detalle en el siguiente procedimiento. El proceso continuará revisando nuevamente los siguientes dos genes RTG y Mode, debido a que el primer gen es un elemento que forma parte del conjunto de funciones y segundo gen es un elemento del conjunto de terminales, el proceso de transformación se aplica nuevamente agrupando el conjunto de datos mediante la utilización de los atributos RTG, DMKT y Class, calculando la moda mediante el atributo FARE, por lo que se integra un nuevo atributo. Este proceso se explicará a detalle en el siguiente paso de la metodología. Este proceso de aprendizaje de atributos se ejecuta hasta que todos los genes son revisados. Al final del proceso se obtiene una nueva solución (un nuevo conjunto de datos) el cual también se encuentra codificado en su forma cromosómica (ver Figura 3.4). Es importante recalcar que se requiere satisfacer algunas restricciones. Por ejemplo, la longitud de las soluciones candidatas las cuales se limitan a través de la fórmula $n = F + T = 13$, siendo F el número de elementos que conforman el conjunto de terminales, y T el número de elementos del conjunto de funciones. Esto representa un total de 51,895,935 combinaciones.

- **Conjunto inicial de soluciones**. Para obtener el conjunto inicial de soluciones el algoritmo ejecuta dos procesos secuenciales: el proceso inicial de codificación y el proceso de aprendizaje de atributos. El primer proceso inicializa aleatoriamente el conjunto inicial de soluciones codificando y seleccionando atributos dentro del conjunto de funciones y las métricas dentro del conjunto de terminales hasta que el tamaño del cromosoma se alcance, como se mostró anteriormente en la Figura 3.3 .

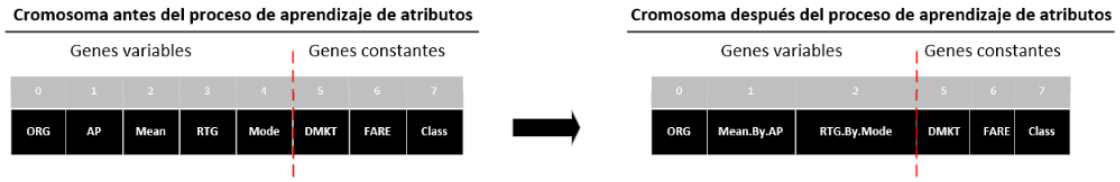

Figura 3.4 Ejemplo del proceso de aprendizaje de una solución

Después de que se ejecuta el primer proceso, el proceso de aprendizaje de atributos se ejecuta para transformar y crear las nuevas soluciones (conjuntos de datos).

El Algoritmo 1 muestra el pseudocódigo para la generación del conjunto inicial de soluciones en su forma de codificada. El algoritmo comienza con la creación de dos variables; la variable s, que contiene los elementos de los conjuntos de funciones y terminales, y la variable *solutions* para guardar las soluciones obtenidas. Los valores en s son los genes que forman los cromosomas después de un proceso aleatorio que se ha ejecutado para seleccionarlos. Las líneas 5 al 11 denotan el proceso aleatorio para seleccionar un valor de s y al cual se le asigna la variable g; esta variable determinará la posición del gen en la posición j del cromosoma. En la línea 7 si el cromosoma ya contiene el valor de g, y si este valor es un elemento del conjunto de funciones, entonces se ejecuta la línea 8, por lo que la variable g se le asignará un valor nulo (null). Esta condición está establecida para evitar atributos repetidos en la misma solución. Finalmente, g se incluye en el cromosoma. Una vez que se forma el cromosoma, los valores se eliminan y se incluye en la lista de soluciones. Cuando el número de cromosomas creados es igual al tamaño predefinido de la población (pop), el proceso finaliza y regresa las soluciones en su forma de codificada.

El Algoritmo 2 define el pseudocódigo del proceso de aprendizaje de atributos. El algoritmo recibe el conjunto inicial de soluciones que fueron creadas previamente en el proceso de codificación por el Algoritmo 1, y el conjunto de funciones (F). La línea 1 representa el bucle general para leer cada cromosoma que se va a decodificar y transformar para obtener los nuevos conjuntos de datos. De las líneas 2 al 13 ejecutan dicha decodificación y los procesos de transformación. Para cada cromosoma en el conjunto de soluciones, el algoritmo analiza cada una de sus genes (ver las líneas de la 5 a la 14). Se revisan los genes en la posición j y la posición $j + 1$ para determinar qué acción debería aplicarse en el algoritmo de aprendizaje de atributos (FL). Si ambos genes son elementos de F, entonces el algoritmo actúa como un selector de atributos

Algoritmo 1 Generación de la población inicial

Input pop-size, chromosome-size, F, T
Output solutions

```
1:  S ← F ∪ T
2:  solutions ← ∅
3:  for i = 1 to pop-size do
4:      chromosome ← ∅
5:      for j = 1 to chromosome-size do
6:          g ← get.random.element(S)
7:          if g ∈ chromosome and g ∈ F then
8:              g = null
9:          end if
10:         chromosome ← chromosome ∪ g
11:     end for
12:     chromosome ← remove.null.values(chromosome)
13:     solutions ← solutions ∪ chromosome
14: end for
15: return solutions
```

dejando el atributo en la posición j sin ninguna modificación para ser parte del conjunto de datos transformado. Si el gen en j pertenece a F, y el gen $j + 1$ pertenece al conjunto de terminales, el algoritmo crea un nuevo atributo agrupando el atributo en la posición j, el elemento en la posición $j + 1$, y los genes constantes. Como previamente se ha explicado la el atributo numérico *FARE* el cual representa el valor base de una tarifa es el único atributo que se puede utilizar para calcular las métricas, el cual permite modificar los cromosomas o soluciones mediante la integración de nuevos atributos. Para comprender mejor ese proceso, si se regresa al ejemplo de la Figura 3.4, la cual muestra una solución codificada creada por el Algoritmo 1 y su transformación por el Algoritmo 2. En este caso, se tiene el atributo ORG en la posición j y el siguiente gen es el atributo AP, ambos son elementos del conjunto de funciones. Debido a lo anterior, el algoritmo selecciona el atributo ORG sin ninguna modificación para integrarlo al conjunto de datos transformado. Cuando la siguiente iteración se ejecuta, AP estará en la posición j del cromosoma, y el siguiente gen es la métrica media (*Mean*). En este caso, el algoritmo agrupa el conjunto de datos utilizando los atributos *Class* y AP para calcular la media de las tarifas utilizando el atributo *FARE*, creando un nuevo atributo denominado *Mean.By.AP*, el cual se integra al nuevo conjunto de datos. Este proceso continúa hasta que todos los genes del cromosoma han sido descodificados (ver línea 12). Continuando con el ejemplo ilustrado en la Figura 3.4, el algoritmo revisa los

Algoritmo 2 Aprendizaje de atributos (FL)

Input solutions, F
Output t-solutions, e-solutions

```
1: for ∀ solution ∈ solutions do
2:    chromosome ← solution
3:    t-chromosome ← ∅
4:    t-dataset ← ∅
5:    while j = 1 to length(chromosome) - 3 do
6:       if chromosome[j] ∈ F and chromosome[j + 1] ∈ F then
7:          apply feature selection to chromosome[j]
8:          t-chromosome[j] ← new encoded attribute
9:          t-dataset[j] ← new attribute
10:      else
11:         create a new feature using chromosome[j] and chromosome[j+1]
12:      end if
13:      j ← j + 1
14:   end while
15:   e-solutions ← e-solutions ∪ t-chromosome
16:   t-solutions ← t-solutions ∪ t-dataset
17: end for
18: return t-solutions, e-solutions
```

siguientes genes RTG y la Mode. Por lo tanto, el algoritmo aplica otra vez el proceso FL, y crea un nuevo atributo denominado RTG.By.Mode. Este proceso se detiene cuando todos los genes variables son analizados. El nuevo cromosoma codificado (*e-solutions* en Algoritmo 2) y el conjunto de datos transformado son también creados (*t-dataset* Algoritmo 2). El algoritmo propuesto finalmente regresa las soluciones (*t-solutions*) como conjuntos de datos y los nuevos cromosomas (*e-solutions*) en su forma codificada. Hay que hacer notar que hasta este punto las nuevas soluciones no han sido evaluadas.

- **Evaluación**. Este procedimiento se encarga de asignar un valor de aptitud (*fitness*) para cada solución (ver Ecuación 3.1). En la metodología propuesta se tiene que evaluar qué tan buenos son los conjuntos de datos resultantes obtenidos a través del proceso de FL. Para ello, se aplica un modelo de clasificación a cada conjunto de datos resultante para proporcionar una función de aptitud que se basa en una combinación de dos métricas: *F-score* (F1) y el tamaño o número de reglas (Nr). F1 representa la media harmónica de los valores de precisión y sensibilidad [85]. Esto formalmente se define

Algoritmo 3 Evaluación de soluciones

Input t-solutions, e-solutions
Output Evaluated-solutions

```
1: for i = 1 to size(t-solutions) do
2:    if n of original features in solution[i] <4 then
3:       fitness = 0
4:    else
5:       create classification model
6:       set fitness
7:    end if
8:    Evaluated-Solutions = [e-solution[i], fitness]
9: end for
10: return Evaluated-solutions
```

como $F1 = \frac{T_P}{T_P + 1/2(F_P + F_N)}$. El tamaño o número de reglas es una medida que considera el número de reglas que forman el modelo. Reducir el tamaño del modelo aumenta su interpretabilidad [66].

$$Fitness = \frac{F_1 * Nr}{2} \tag{3.1}$$

El Algoritmo 3 muestra el pseudocódigo del proceso de evaluación. Este algoritmo recibe un conjunto de soluciones como conjuntos de datos (*t-solutions*), y el conjunto de soluciones en su forma codificada (*e-solutions*). El algoritmo funciona como el bucle principal para evaluar cada solución (*t-solution*) creada en el Algoritmo 2. En cada iteración, la propuesta verifica si en la solución existen por lo menos cuatro de los atributos originales en referencia al conjunto de datos original, para ser evaluada a través de un algoritmo de clasificación. De lo contrario, si en la solución no existen al menos cuatro atributos originales (excluyendo el atributo Class), el algoritmo le asigna una aptitud con valor 0. Este requisito de incluir al menos cuatro o más de los atributos originales es para mantener la interpretabilidad del modelo. Finalmente, el procedimiento propuesto devuelve las soluciones en su nueva forma codificada, junto con su respectivo valor de aptitud. De vuelta al ejemplo que se muestra en la Figura 3.3, el cromosoma contiene solo tres de los atributos originales después del proceso FL: *ORG, DMKT y FARE*. Por lo tanto, un valor de aptitud 0 es asignado a esta solución. Para obtener las reglas y calcular el valor de aptitud, se propone el uso de algoritmos clasificación de caja blanca. En este sentido, el proceso de evaluación se

lleva a cabo considerando una serie de algoritmos clásicos de clasificación disponibles en las librerias Rweka [84] y LAC [133] :

- Cuatro algoritmos de aprendizaje basados en reglas: JRip [35], que es un algoritmo de aprendizaje de reglas proposicionales; OneR [83], que utiliza el atributo de error mínimo para la predicción de clases; PART [61] que utiliza el método de separar y conquistar; y Ridor [4], que es una implementación del algoritmo de aprendizaje de reglas conocido como RIpple-DOwnRule.
- Nueve árboles de decisión: algoritmos para generar árboles podados o sin podar J48 y C4.5 [148]; J48 Consolidated [135]; J48 Graft [165]; Decision Stump [89]; simpleCART [24]; BFTree [63]; LMT [158]; y JCDT [1].
- Cinco algoritmos de clasificación asociativa: CBA [111] descubre un subconjunto de reglas de asociación de clase y produce un modelo de clasificación de las reglas extraídas. FOIL [137] y FOIL2 [139], los cuales son algoritmos codiciosos (greedy algorithms) que aprenden reglas para distinguir los ejemplos positivos de los negativos. CPAR [180] hereda la idea básica de FOIL en la generación de reglas e integra las características de asociativas de clasificación en análisis predictivo de reglas. Finalmente, PMR [138], el cual selecciona la mejor regla entre un conjunto de reglas generadas.

- **Operadores genéticos**. Aquí se describen los operadores genéticos que se utilizan en el algoritmo que se propone en esta memoria de tesis. Primero, en cuanto a la selección, se aplica una selección de ruleta [182], que consiste en asignar un segmento de la ruleta a las soluciones con base en su aptitud y la aptitud total de la población actual. La ruleta se hace girar tantas veces como soluciones haya en la población para mantener la población en un tamaño constante. De manera que, con la selección de la rueda de la ruleta, las soluciones se seleccionan de acuerdo con la aptitud y la suerte del sorteo, lo que significa que a veces los mejores rasgos puedan perderse. Sin embargo, al combinar la selección de la ruleta con la clonación de las mejores soluciones de cada generación, se garantiza que al menos los mejores los rasgos no se pierden. Esta técnica de clonar las mejores soluciones de generación se conoce como elitismo y se utiliza por la mayoría de los esquemas de selección estocásticos. Finalmente, en cuanto a los operadores genéticos de cruce y mutación, el enfoque propuesto sigue un operador de cruce típico de un solo punto que actúa en un solo gen o punto de la solución. El operador de cruce elige aleatoriamente un lugar en el cromosoma de los padres, para intercambiar, para después, intercambiar las subcadenas, creando dos descendientes. Además, un método de mutación típico sobre la base de una mutación

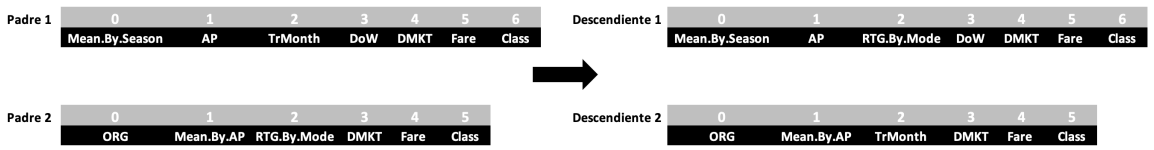

Figura 3.5 Ejemplo del operador de cruce

de un solo punto se realiza. Los elementos a la derecha/izquierda de ese punto se intercambian entre los dos cromosomas progenitores, lo que da como resultado dos descendientes. Cada descendiente está formado por alguna información genética de ambos padres. Así mismo, se realiza una mutación de un solo punto donde se elige un elemento aleatorio del cromosoma y su valor se cambia a un nuevo valor aleatorio.

Se proponen dos operadores genéticos para obtener nuevas soluciones candidatas en cada generación. El operador genético de cruce es propuesto para intensificar la diversidad de la población. Este operador genético elige aleatoriamente un punto de corte entre los genes variables del cromosoma de uno de los padres. El mismo proceso se repite en el siguiente padre. De manera que, dos descendientes se forman por la combinación de los genes de los cromosomas divididos; estos descendientes incluyen genes de ambos padres. Para clarificar, considere los dos ejemplos de padres siguientes (ver Figura 3.5 codificados por el conjunto de datos de Air Canada se ha considerado a lo largo de este capítulo): El padre 1 representa un conjunto de datos que incluye la media por temporada para cada uno de los mercados directos (DMKT). También incluye la compra anticipada (AP) atributo como el número de días en los que se analiza la tarifa, así mismo incluye el mes y el día de la semana en que se aplica la tarifa. El padre 2 representa un conjunto de datos que incluye el origen (aeropuerto o ciudad donde se inicia el viaje), el medio por AP, y la moda por cada enrutamiento de cada DMKT. En este ejemplo se muestra, el punto de corte en el cual se elige aleatoriamente en el gen 2, devolviendo dos descendientes con información de ambos padres: descendencia 1 representa un conjunto de datos que incluye la media por temporada para cada uno de los mercados directos (DMKT), la el atributo compra anticipada (AP) el cual representa como el número de días en los que se analiza la tarifa, y la moda para cada enrutamiento de cada DMKT. El descendiente 2 representa un conjunto de datos que incluye el origen, la media por AP, y el mes en que se aplica la tarifa.

Además, se propone utilizar un operador genético de mutación para diversificar la población. Este operador genético elige de forma aleatoria un gen de los genes variables del cromosoma de uno de los padres, y este gen se reemplaza con un nuevo valor

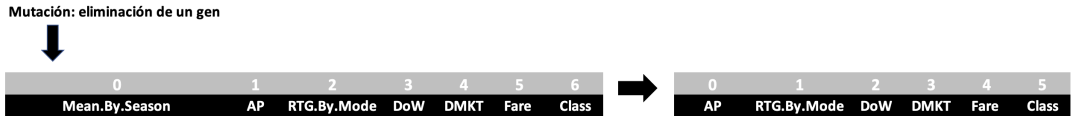

Figura 3.6 Ejemplo del operador de mutación

aleatorio (podría ser un espacio en blanco o eliminación). El nuevo individuo es similar al anterior (solamente se añade una pequeña variación). Como cuestión de aclaración, considérese un padre de muestra (consulte la Figura 3.6 codificada por el conjunto de datos de Air Canada que se ha considerado a lo largo de este capítulo para las explicaciones), que representa un conjunto de datos que incluye la media por temporada para cada de los mercados directos (DMKT), el atributo de compra anticipada (AP) como el número de días en los que se analiza la tarifa, la moda de cada enrutamiento y el día de la semana en que se aplica la tarifa. En este caso se elimina el atributo que representa la media por temporada, por lo que el nuevo individuo representa un conjunto de datos con el atributo de compra anticipada (AP) como el número de días en los que se analiza la tarifa, la moda de cada ruta, y el día de la semana en que se aplica la tarifa.

Por último, pero no menos importante, el flujo de trabajo general de la propuesta se describe en el algoritmo GEP-FL (ver Algoritmo 4). En el primero se necesitan tres pasos para producir el conjunto inicial de soluciones. Luego se lleva a cabo un proceso iterativo sobre una serie de iteraciones (generaciones) que devuelven las mejores soluciones encontradas después del bucle. No se requiere una explicación adicional para este algoritmo, ya que todos los procesos llevados a cabo por él fueron descritos previamente.

3.3. Estudio experimental

El objetivo primordial de esta sección es analizar, en primera instancia, si existen diferencias entre el uso o no la propuesta GEP-FL y, en segunda instancia, qué algoritmo de clasificación es mejor para el problema en cuestión. En el proceso de evaluación se han ejecutado un total de dieciocho algoritmos de clasificación que han sido propuestos previamente (ver Sección 3.2). El estudio experimental se ha llevado a cabo sobre un escenario que utiliza tarifas reales publicadas por la aerolínea Air Canada en el período de diciembre de 2019 a enero de 2020, correspondiente a un período de viajes entre los meses de diciembre de 2019 y abril de 2020. En esta sección se describen los experimentos que se llevaron a

Algoritmo 4 GEP-FL algorithm

Input n-iterations
Output best solutions

1: F = function set
2: T = terminal set
3: Generation of the initial population (see Algorithm 1)
4: Feature Learning (see Algorithm 2)
5: Evaluation of solutions (see Algorithm 3)
6: **for** i = 1 to n-iterations **do**
7: Roulette-wheel selection
8: Apply genetic operators
9: Evaluation of solutions
10: Update population
11: **end for**
12: **return** best solutions

cabo, para demostrar la utilidad del enfoque propuesto GEP-FL. Importante destacar que cada algoritmo fue ejecutado diez veces, y los resultados mostrados son la media de estas ejecuciones. En cuanto al análisis de la idoneidad del uso de Feature Learning para este problema, se han utilizado tres métricas: eficacia (Acc), *f-score* (F1) e interpretabilidad (Nr). A su vez, se compara el algoritmo de GEP-FL propuesto en contra de un enfoque clásico de búsqueda aleatoria (el cual no cuenta con los operadores de cruce y mutación), con la finalidad de demostrar que el enfoque evolutivo produce mejores resultados. Por lo tanto, primero se compara el enfoque GEP-FL con dos metodologías: a) considerando los datos originales; b) usando un método de búsqueda aleatoria. En este análisis todos los algoritmos de clasificación fueron ejecutados, empleando una validación cruzada de diez iteraciones en todos los datos disponibles después de que las fases de recopilación y de pre-procesado hayan sido completadas.

Comparando los resultados de Acc obtenidos por las tres metodologías (ver Tabla 3.2), se puede observar que no existen grandes diferencias, por lo que, se llevó a cabo un test de Friedman [44] para determinar si existen diferencias significativas entre estas tres metodologías (Org. Data, Rand-FL y GEP-FL). El valor p computado es $p = 0{,}000063$, de manera que, el test refuta la hipótesis nula en la cual todas las metodologías se comportan de forma similar en términos de Acc con un valor $\alpha = 0.01$. Por lo tanto, se aplica una prueba post-hoc para obtener diferencias significativas entre las metodologías. El test de Shaffer (ver Figura 3.7) demuestra que GEP-FL es la metodología que obtiene mejores resultados en términos de Acc con un valor α 0.01, existiendo diferencias significativas con respecto a los enfoques

	Accuracy (Acc)			F-score (F1)			Interpretability (Nr)		
Algorithm	Org. Data	Rand-FL	GEP-FL	Org. Data	Rand-FL	GEP-FL	Org. Data	Rand-FL	GEP-FL
BFTree	99.6	99.8	**99.9**	99.5	99.7	**99.9**	289	119	**64**
C50	99.7	99.7	**99.9**	99.6	99.8	**99.9**	166	102	**74**
CBA	97.9	98.7	**99.9**	97.9	98.6	**99.9**	238	60	**39**
CPAR	98.4	98.6	**99.6**	98.4	98.3	**99.6**	502	93	**84**
DecisionStump	45.7	**47.2**	45.1	60.9	**62.8**	61.7	**3**	**3**	**3**
FOIL	99.3	99.8	**99.9**	99.3	99.8	**99.9**	217	61	**37**
FOIL2	99.3	99.8	**99.9**	99.3	99.8	**99.9**	217	73	**45**
J48	99.7	99.8	**99.9**	99.6	99.7	**99.9**	273	114	**99**
J48Consolidated	99.6	99.7	**99.9**	99.6	99.7	**99.9**	298	**89**	119
J48graft	99.7	99.7	**99.9**	99.6	99.8	**99.9**	919	349	**232**
JCDT	99.6	**99.9**	99.6	99.6	**99.9**	99.6	238	167	**112**
JRip	99.7	99.8	**99.9**	99.6	99.8	**99.9**	89	80	**37**
LMT	97.8	98.9	**99.1**	97.9	98.8	**99.1**	113	**10**	**10**
OneR	86.4	**98.4**	94.5	85.3	**98.1**	94.0	**2**	**2**	**2**
PART	99.6	99.2	**99.9**	99.6	99.3	**99.9**	146	**48**	65
PRM	98.5	**99.8**	99.6	98.5	**99.8**	99.7	344	**64**	75
Ridor	99.6	97.4	**99.9**	99.6	97.2	**99.9**	**6**	85	142
simpleCART	99.7	**99.9**	**99.9**	99.6	99.8	**99.9**	146	91	**65**

Tabla 3.2 Resultados experimentales considerando diferentes algoritmos de clasificación en la fase de evaluación.

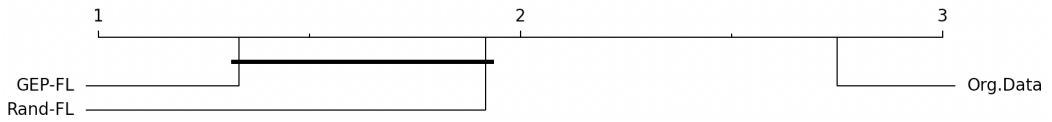

Figura 3.7 Diagrama de diferencia crítica que muestra una comparación estadística de exactitud (Acc) utilizando la prueba de Shaffer.

de Org. Data y Rand-FL. En términos de la métrica F1, los resultados del test de Friedman obtuvo un valor $p = 0{,}000015$, refutando la hipótesis nula en la cual las tres metodologías se comportan de manera similar con un valor α de 0.01. Así mismo se aplicó una prueba post-hoc para obtener diferencias significativas entre las metodologías. El test de Shaffer (ver Figura 3.8) demuestra que GEP-FL es la metodología que obtiene mejores resultados en términos de la métrica F1, existiendo diferencias significativas con respecto a los enfoques de Org. Data y Rand-FL. Finalmente, el test de Friedman aplicado al número de reglas, arrojó un valor $p = 0{,}000040$, refutando la hipótesis nula en la cual todas las metodologías se comportan de manera similar con un valor α de 0.01. La prueba post-hoc de Shaffer (ver Figura 3.9) demuestra que GEP-FL es la metodología que obtiene mejores resultados en términos del número de reglas obtenidas. Cabe mencionar que el enfoque Rand-FL también tiene un alto comportamiento.

En resumen, los resultados estadísticos muestran que el enfoque GEP-FL mejora el rendimiento de la clasificación y la interpretabilidad de los modelos. Por lo tanto, para poder determinar qué algoritmo de clasificación es el más adecuado para ser aplicado

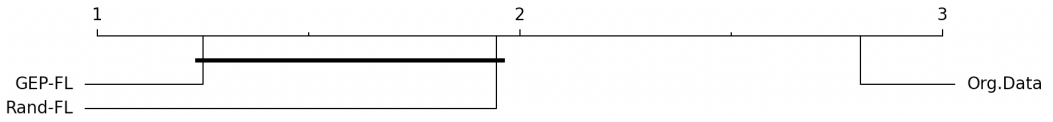

Figura 3.8 Diagrama de diferencia crítica que muestra una comparación estadística *F-score* (F1) utilizando la prueba de Shaffer.

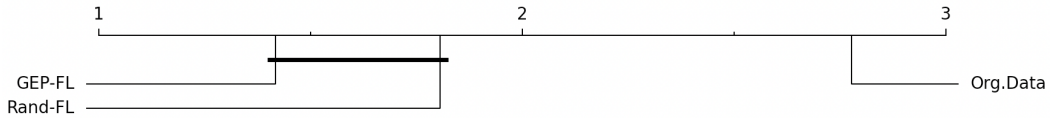

Figura 3.9 Diagrama de diferencia crítica que muestra una comparación estadística de interpretabilidad (Nr) utilizando la prueba de Shaffer.

Measure	Statistic	p-Value
Accuracy (Acc)	111.00	7.822e-16
F-score (F_1)	108.29	2.531e-15
Size (Nr)	163.81	2.2e-16

Tabla 3.3 Resultados del test de Friedman para diferentes algoritmos GEP-FL en la fase de evaluación.

a la metodología propuesta, se llevó a cabo un segundo análisis estadístico basado en los algoritmos que utilizaron el enfoque GEP-FL (ver Tabla 3.3). En primer lugar, los $valores-p$ calculados mediante la prueba estadística para Acc, F1 y Nr rechazaron la hipótesis nula en la que todos los algoritmos se comportan por igual, considerando $\alpha = 0.01$. Por lo que, se realizó la prueba post-hoc de Shaffer para identificar en dónde se encuentran estas diferencias, considerando un nivel significativo de $\alpha = 0.01$ y cuyos resultados se resumen a través de los diagramas de diferencias críticas que se muestran en las Figuras 3.10, 3.11 y 3.12. Como se muestra, JRip está en cuarta posición en Acc y F1, y en la quinta posición en Nr. En ninguno de estos análisis JRip es peor que los mejores algoritmos (ver Figuras 3.10, 3.11 y 3.12). En consideración de todo lo anterior, se define a JRip como el algoritmo que produce el mejor rendimiento.

3.4. Reglas descubiertas

El objetivo de este estudio es aplicar la metodología propuesta a un escenario real. En este sentido, se ha tomado a JRip como el algoritmo de clasificación utilizado en la fase de evaluación (tal y como se explicó en la sección anterior como resultado del estudio experimental). La Tabla 3.4 presenta los resultados obtenidos por la metodología propuesta,

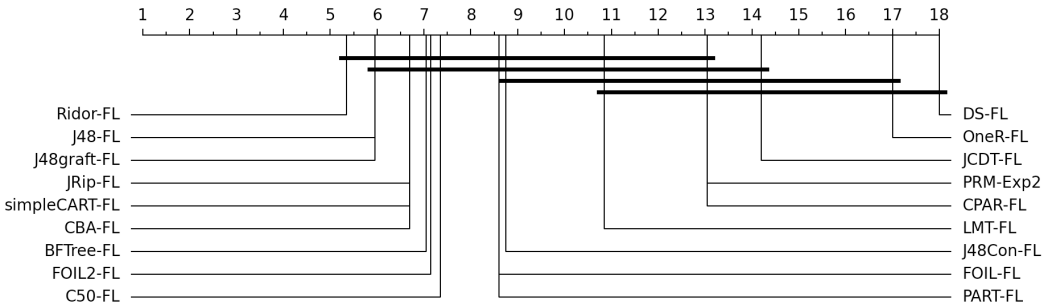

Figura 3.10 Diagrama de diferencia crítica que muestra una comparación estadística de la exactitud (Acc) utilizando la prueba de Shaffer.

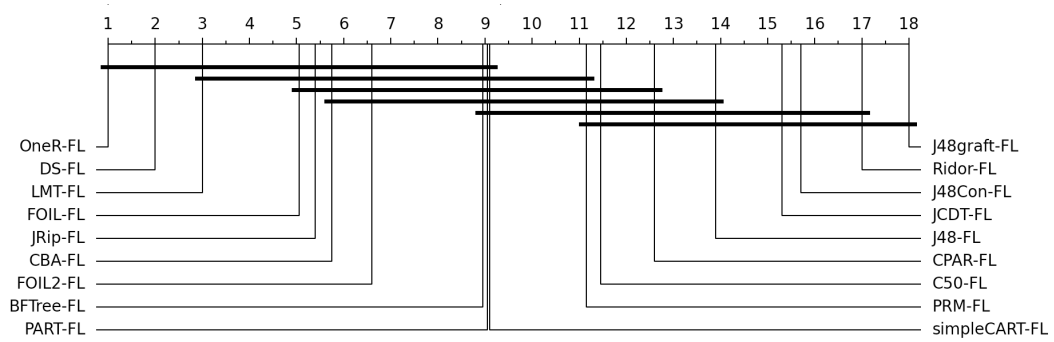

Figura 3.11 Diagrama de diferencia crítica que muestra una comparación estadística de *F-score* (F1) utilizando la prueba de Shaffer.

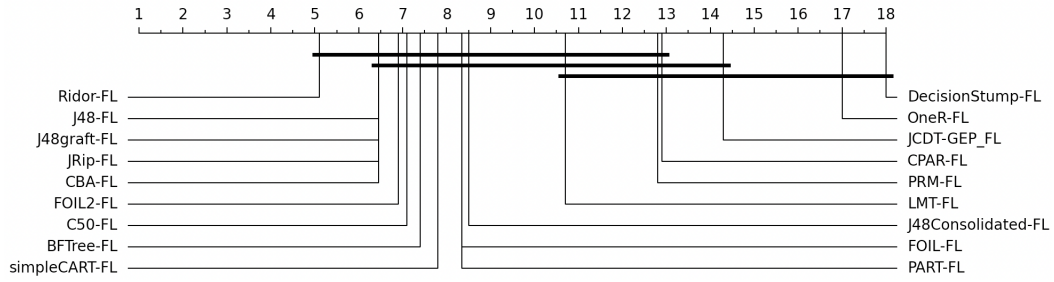

Figura 3.12 Diagrama de diferencia crítica que muestra una comparación estadística de interpretabilidad (Nr) utilizando la prueba de Shaffer.

denotando entre paréntesis el número de casos satisfechos y no satisfechos por cada regla. Los atributos que más aparecen con mayor frecuencia son *Season* y *AP.Mode*. El segundo, es una de los atributos aprendidos por el algoritmo GEP-FL, que proporciona reglas interesantes y de fácil interpretación. A continuación proporcionamos la interpretación de las doce reglas descubiertas más significativas:

No.	Reglas descubiertas:
1	$(Season = A) and (AP.Mode <= 63) => Class = K(512,0/0,0)$
2	$(AP <= 7) and (AP.Mode = 148,45) => Class = K(114,0/0,0)$
3	$(AP <= 7) and (AP.Mode <= 124,88) and (ORG = YWG) => Class = K(119,0/0,0)$
4	$(DMKT = YYCYYZ) and (AP.Mode <= 129,4) => Class = K(96,0/0,0)$
5	$(AP.Mode <= 145) and (DES = YYZ) and (ORG = YVR) => Class = A(641,0/0,0)$
6	$(AP.Mode <= 145) and (ORG = YWG) => Class = A(842,0/0,0)$
7	$(RTG = R-5) and (AP <= 7) => Class = A(451,0/0,0)$
8	$(AP <= 21) and (AP >= 18) and (AP.Mode <= 141) => Class = T(1294,0/0,0)$
9	$(Season = A) and (AP.Mode <= 188) and (AP.Mode >= 177) => Class = T(580,0/0,0)$
10	$(ORG = YUL) and (Season = H) and (AP.Mode <= 151) => Class = T(144,0/0,0)$
11	$(AP >= 21) => Class = L(7030,0/0,0)$
12	$(AP.Mode = 151) => Class = L(1298,0/0,0)$

Tabla 3.4 Reglas descubiertas cuando la metodología propuesta es aplicada a un escenario real.

- La regla número 1 demuestra el precio máximo que las tarifas de clase K, el cual es de $63 en cualquiera de los mercados y en todas las temporadas del año.

- La regla número 2 muestra una clara estrategia de precios para las tarifas publicadas con una compra anticipada de 7 días antes del vuelo y en la cual la moda son $148.45. Para las tarifas de clase K. Debido al número de casos en que esta regla aplica, se puede observar que están diseñadas específicamente para aplicarse a un número restringido de vuelos, en los cuales su desempeñó de ventas no es óptimo; por lo tanto, en este caso un bajo precio a solo 7 días de que se efectúe el vuelo, significa que la aerolínea está tratando de llenar el mayor número de asientos disponibles en estos vuelos. Este tipo de tácticas son difíciles de detectar en un proceso regular.

- La regla número 3 demuestra que para los vuelos que salen desde el aeropuerto de Winnipeg hacia cualquier otro destino y con 7 días de anticipación antes de la fecha de vuelo, la tarifa moda es de $124.88 para la clase K.

- La regla 4 demuestra una interesante estrategia de fijación de precios, en la que el precio más común para los viajes entre los aeropuertos de Calgary y Toronto tiene un precio de $129.4 para la clase K. Debido al número de casos en que se aplica esta regla se puede deducir que este es el precio estructural para las tarifas de clase K en esta ruta.

- Se puede observar en la Regla 5 que el precio estructural para la clase K en un vuelo entre Toronto y Vancouver con una compra anticipada de al menos 45 días antes de que se efectúe el vuelo es de $145.
- La regla 6 muestra la estrategia del precio base para vuelos que salen desde Winnipeg y con una compra anticipada de 45 antes de que la fecha de salida para la clase A es de $145.
- Se puede observar en la Regla 7 un interesante descubrimiento realizado por la metodología propuesta en el que las tarifas que pertenecen a la clase A y con la restricción de 7 días de compra anticipada en la mayoría de los casos son para vuelos directos. Este es un descubrimiento significativo porque se puede observar una clara utilización de la clase A como una tarifa táctica. Este tipo de estrategias son difíciles de descubrir por procesos manuales.
- La regla 8 demuestra que las tarifas con compra anticipada entre 18 y 21 días, precio más común para la clase T es de $141.
- Se puede observar en la Regla 9, que las tarifas pertenecientes a clase T y que aplican todo el año, los precios fluctúan entre $117 y $188, independientemente de cualquier compra anticipada. Este es un descubrimiento muy interesante porque la regla muestra la franja tarifaria para esta clase, básicamente descubre el precio más bajo y el precio más alto en el que se puede vender una tarifa de clase T.
- La Regla 10 muestra que los vuelos desde Montreal en temporada alta, la tarifa más común para la clase T es de $151.
- La regla 11 muestra que las tarifas con clase L están disponibles para pasajeros que intentan reservar vuelos con al menos 21 días de la fecha de salida.
- La regla 12 muestra que el punto de precio más común para la tarifa de clase L tiene un costo de $151.

En este punto, es clave demostrar si las reglas devueltas son lo suficientemente buenas, por lo que estas reglas también se comparan en contra de las reglas las obtenidas por el conocido algoritmo de reglas de asociación FP-Growth [164] las cuales fueron extraídas de las mejores soluciones candidatas obtenidas por el Enfoque GEP-FL. Esta comparación se efectúa basada en el número de reglas (ver Tabla 3.5) en la que se demuestra claramente que el enfoque GEP-FL devuelve un modelo reglas menor, el cual un usuario final puede entender de una manera más fácil que un conjunto de miles de reglas como los que genera

FP-Growth. Además, se ha analizado el pequeño conjunto de reglas devueltas por el enfoque GEP-FL en términos de la medida de calidad de $Lift$ [114] que calcula la importancia de la regla. Esta es una medida del rendimiento de un modelo de focalización de la clasificación de casos con una respuesta mejorada en comparación con un modelo de focalización de elección aleatoria. Las reglas obtenidas tienen un valor de $Lift$ superior a 4, lo que denota claramente la importancia de las reglas descubiertas. Adicionalmente, las reglas devueltas no son reglas generales que se puedan obtener fácilmente. Estas reglas aparecen en menos del 5 % de las transacciones, por lo que es computacionalmente complejo descubrirlas, y no se obtienen fácilmente analizando los datos.

3.5. Conclusiones del capítulo

Se ha demostrado que la extracción de reglas interpretables en las tarifas aéreas puede ser una tarea difícil, no solo porque es un problema multifactorial, sino también porque el número de tarifas publicadas por una aerolínea comercial puede contener varias restricciones y características similares entre sus clases. Debido a lo anterior, el encontrar información relevante es una tarea muy compleja de realizar. Para resolver este problema, se ha propuesto la implementación de una metodología para crear conjuntos de datos especiales transformando los atributos originales a través de un algoritmo de aprendizaje de atributos (FL). La metodología propuesta ha sido capaz de obtener un modelo de alta interpretación con un número suficiente de reglas y un menor número de antecedentes en cada regla sin afectar la eficacia de clasificación.

Data	GEP-FL	FP-growth
1	37	1,692
2	35	1,897
3	41	1,857
4	33	1,757
5	37	1,787
6	44	1,716
7	29	964
8	37	1,997
9	37	1,985
10	45	2,039

Tabla 3.5 Número de reglas obtenidas por el enfoque GEP-FL JRip and FP-growth.

La metodología propuesta se aplicó a un escenario real utilizando datos de la aerolínea más importante de Canadá (Air Canada). Con base en los resultados obtenidos, se presentan las siguientes conclusiones:

1. La aplicación de una metodología automatizada que permite producir información interesante para los equipos de analistas de precios es de gran utilidad, evitando la interacción humana y, por tanto, errores humanos.

2. La metodología propuesta es capaz de reducir el número de reglas extraídas y disminuir el número de condiciones en las reglas, lo cual es crucial para aumentar la interpretabilidad de los modelos.

3. La metodología propuesta es capaz de imitar el proceso de aprendizaje que generalmente llevan a cabo los equipos de analistas de precios.

Capítulo 4

Metodología evolutiva de descuentos dinámicos

DP se define como la tarea de ajustar continuamente los precios de un producto o un servicio, en el cual la meta principal de estos cambios de precios tiene dos objetivos: por un lado, las empresas quieren optimizar los márgenes y, por otro lado, quieren aumentar sus posibilidades de ventas [46]. Por lo tanto, en un entorno complejo de una industria como lo es la aviación comercial, en donde la competencia afecta de manera significativa el comportamiento de compras de los pasajeros, impactando el precio óptimo de las tarifas. Es fundamental la implementación de metodologías efectivas de DP para estimular la demanda, mitigar la pérdida de pasajeros y aumentar los ingresos de las aerolíneas. De acuerdo a la naturaleza de cada industria, DP puede tomar muchas formas [47]. La forma más sofisticada es casi instantánea, en la que las aerolíneas deben identificar a los pasajeros potenciales que buscan boletos en sus sitios web para crear ofertas específicas de descuentos en tiempo real como incentivo para aumentar la probabilidad de compra de boletos en determinadas rutas.

Históricamente, las aerolíneas han estado utilizando precios estáticos, por lo que, una aerolínea crea su estructura de tarifas utilizando un número limitado de puntos de precios y los lanza al público en general. Cada precio se desarrolla en función de varios factores, como son la demanda del mercado, la segmentación de pasajeros, las respuestas competitivas, entre otros. Desafortunadamente, para las aerolíneas estas estrategias de precios no son suficientes, porque existen pasajeros con diferente sensibilidad de precios que no se limitan la reserva de boletos a una fecha u horario e intentarán encontrar las tarifas más baratas independientemente de situaciones como son escalas largas. Otro tipo de pasajeros, que tienen una mayor disposición a pagar (*Willignes to Pay*, WTP) podría considerar tales factores como son la hora del vuelo, día de la semana (*Day of Week*, DOW) o la clase de asiento. Debido a

esto, los equipos de precios e ingresos de las aerolíneas tratan de crear precios según el tipo de pasajero y su WTP; desafortunadamente, los constantes cambios en los mercados debido a acciones competitivas de otras aerolíneas, como son los descuentos masivos, cambios de horario, lanzamiento de nuevas rutas, eliminación de vuelos o rutas, entre otros, hacen que esta segmentación sea limitada. Para hacer frente a estos problemas, hoy en día se puede ver un aumento del uso de técnicas de predictivas y descriptivas utilizando métodos de DM y de ML, que están siendo desarrollados por algunas aerolíneas para construcción de ofertas novedosas y metodologías para mejorar la segmentación del mercado, predecir el precio correcto en el momento correcto y extraer información relevante que les permitan comprender el comportamiento de compra de los pasajeros. Sin embargo, la implementación de un sistema de ofertas dinámicas dentro de la industria área no es una tarea fácil, de hecho, hoy en la actualidad, la construcción de ofertas es rudimentaria [125] y está limitada por las capacidades de los sistemas de distribución, los cuales cuentan con un conjunto finito de precios fijos para las tarifas [171].

Actualmente, la necesidad de metodologías automatizadas de DP en la industria aérea es enorme. Estas metodologías deben ser capaces de generar reglas interpretables mediante el uso de técnicas de DM capaces de explorar todos los atributos que permitan el descubrimiento de subgrupos ocultos e interesantes para identificar pasajeros potenciales y ofrecer las ofertas adecuadas a sus características. Dichas metodologías deben tener como finalidad aumentar la tasa de conversión, e integrar sistemas de recomendación basados en reglas que ayuden a los analistas a la rápida y precisa toma decisiones para determinar el tipo de pasajeros que requieren un descuento. Cabe mencionar que el análisis del comportamiento de compras y el cálculo de probabilidad de las mismas es una tarea difícil de realizar. Tomando todo lo anterior, en esta, apartado de la memoria de tesis, se propone una metodología de DM que permite la automatización de un proceso de selección de características y el uso del descubrimiento de SD para extraer reglas significativas que sirvan como base de un sistema de recomendación para la oferta de descuentos ofrecer ofertas dinámicos. En concreto, la novedad de esta metodología es la siguiente:

- Se propone una metodología automatizada que sirve como un sistema de recomendaciones basado en reglas para proporcionar descuentos dinámicos para incentivar las reservas de tarifas aéreas.
- También se propone un algoritmo de evolución gramatical para aplicar un proceso de selección de características y así obtener mejores datos que permitan a los algoritmos de SD la extracción de reglas significativas.

- Se aplicó esta metodología propuesta sobre un caso de estudio real mediante el uso de datos privados de una aerolínea comercial.

La Figura 4.1 muestra la metodología propuesta, la cual incluye cuatro pasos que se describen a profundidad: recopilación y preprocesamiento de datos, selección de características, extracción de reglas, modelo de probabilidades y selección de reglas. Todos los pasos anteriores se describen en este capítulo.

4.1. Recopilación y preprocesamiento de datos

En esta primera etapa, se descargan automáticamente de la base de datos para recopilar todas las búsquedas históricas disponibles de huéspedes realizadas en el sitio web de la aerolínea. Por lo cual para esta memoria de tesis se han considerado un total de diez rutas, las cuales tienen similitudes entre la longitud de su origen y destino y sus estructuras de precios. Esta información contiene la actividad efectuada en el sitio web por parte de los de los pasajeros potenciales entre las fechas del 1 de diciembre de 2021 y 15 de febrero de 2022. De esta descarga se obtiene un conjunto de datos con un total de 93,258 búsquedas realizadas, y un total de 49 atributos, de los cuales se seleccionan los siguientes 16 atributos automáticamente:

- Mercado directo (*Direct Market*, DMkt): atributo carácter que representa el origen y destino del viaje. Ejemplo: MADAMS (Madrid a Ámsterdam).
- Día de la semana de vuelo (*Flight Day of Week*, FlightDoW): atributo carácter que representa el día de la semana en que se realizará el vuelo. Ejemplo: TUE (martes).
- Día de reserva (*Hit Day of Week*, HitDoW): atributo carácter que representa el día de la semana en que la reserva de un vuelo se realiza. Ejemplo: FRI (viernes).
- Tiempo de reserva (*Booking time*): atributo numérico que representa la hora del día en que se ha realizado la reserva de un vuelo. Ejemplo: 13.

Figura 4.1 Propuesta de metodología para la creación de una tabla de descuentos dinámicos.

- Número de adultos (*Number of adults*, NoOfAdults): atributo numérico que representa el número de adultos incluidos en la reserva. Ejemplo: 1.
- Hijos (*Children*): atributo numérico que representa el número de hijos incluidos en la reserva. Ejemplo: 1.
- Infantes (*Infants*): atributo numérico que representa el número de infantes incluidos en la reserva. Ejemplo: 0.
- Días anticipados de reserva (*Booking Peace*): atributo numérico que representa la diferencia de días entre la fecha en que la reserva se está realizando y la fecha de salida del vuelo. Ejemplo: 10.
- Duración del viaje (*Trip duration*): atributo numérico que representa la diferencia de días entre el vuelo de ida y el de vuelta. Las reservas que son para viajes *one-way* (solo ida) tienen un valor automático de 0. Ejemplo. 7 (7 días de diferencia entre el primer vuelo y el vuelo de vuelta).
- Tipo de Vuelo (*Type flight*): atributo carácter que representa si el vuelo es un vuelo directo o de conexión.
- OW.RT (*One Way Round Trip*, OW.RT): atributo carácter que indica si la reserva es un vuelo *one-way* (solo de ida) o *round trip* (de ida y vuelta).
- Tipo de usuario (*User type*): atributo carácter que indica si un invitado es nuevo o repetido.
- Estado de logueo (*Log Status*): atributo carácter que indica si el usuario está logueado en el sistema.
- Tipo de moneda (*Currency*): atributo carácter que representa el tipo de moneda que se muestra en el sitio web mientras se efectúa la búsqueda de vuelos. Ejemplo. Euros, Dólares, Pesos, entre otros.
- Precio (*Price*): atributo numérico que indica el precio total a pagar por pasajero.
- Reserva (*Booking*): atributo carácter que indica si el invitado reservó un vuelo o no. Ejemplo: “si” o “no”.

Después de recopilar los datos, se aplican técnicas de imputación de datos para usar medidas como la media y la moda para los atributos numéricos y categóricos y rellenar los valores nulos en el conjunto de datos. Durante esta etapa se crean los siguientes atributos:

- Origen Destino No Directo (*Non Direct Origin Destination*, NDOD): este atributo se crea utilizando el atributo DMkt; se extrae el origen y el destino de las rutas y se ordenan alfabéticamente. Por ejemplo, si el DMkt es MADAMS, el NDOD será AMSMAD.

- CharChildren: se crea un atributo carácter a partir del atributo NoOfChildren que transforman el número valores de 0 a "No", y valores de 1 o superiores a "Si".

- Número de días para la salida (NDO): se crea un atributo carácter a través de una serie de bins a partir del atributo *Booking Peace* usando los siguientes rangos 0-3, 4-7, 8-14, 15-21, 22-30, 31-45, 46-90 y las reservas superiores a 90 días.

- CharTripDuration: se crea un atributo carácter a través de una serie de bins a partir del atributo *Trip duration* usando los siguientes rangos 0-3, 4-7, 8-14, 15-21, 22-30 y viajes superiores a 30 días.

- Tipo Día de la Semana (*TypeDoW*): se crea un atributo carácter utilizando el atributo Flight DoW. Si el vuelo ocurre en días martes o miércoles se asigna el valor de OP (*off peak day*), de lo contrario si el vuelo ocurre en los otros días de la semana el valor asignado será P (*peak day*).

- Número de Pasajeros (NoOfPassengers): se crea un atributo númerico que es la suma de los atributos NoOFAdults, Children e Infants.

- Tiempo del día de búsqueda (*SearchDT*): se crea un atributo categórico que representa la hora del día en que se efectúa la buscando de un vuelo, por ejemplo: mañana, noche, tarde y medianoche.

Al final de la recopilación y preprocesamiento de datos, tenemos obtiene un conjunto de datos con 93,258 instancias y 25 atributos; este conjunto de datos se utilizará en las siguientes etapas de la metodología propuesta.

4.2. Método de extracción de reglas

En esta parte se propone algoritmo evolutivo gramatical, el cual puede desarrollar programas completos en un lenguaje arbitrario, mediante el uso de una cadena binaria de longitud variable. El genoma binario determina las reglas de producción que se utilizaran en un formato de *Backus-Naur*, utilizado un proceso de mapeo de genotipo a fenotipo. GE está configurado de tal manera que el algoritmo evolutivo es independiente de los programas de

```
G = (N, T, P, S) with:
N = {CharAdults, CharChildren,  NDO, CharTripDuration, TypeDoW,
NoOfPassengers, SearchDT, FlightDoW, HitDoW, OW.RT, Infants,
Type Flight, User Type, Log Status, Currency}
T = {=, !=}
P = { S = <e>;
    <e> = <a><o><v> | <a>;
    <a> = CharAdults | CharChildren |  NDO | CharTripDuration |
    TypeDoW | NoOfPassengers | SearchDT | FlightDoW | HitDoW |
    OW.RT | Infants | Type Flight | User Type | Log Status | Currency;
    <o> = "=" | "!=";
    <v> = CharAdults {One, Two, 3++} | CharChildren {One, Two, 3++} |
    NDO {0-3, 4-7, 8-14, 15-21, 22-30, 31-45, 46-90, 90++} |
    CharTripDuration {0, 0-3, 0-7, 0-14, 0-21, 22++} | TypeDoW{P, OP} |
    NoOfPassengers {One, Two, 3++} |
    SearchDT{morning, afternnon, night, mid-night} |
    FlightDoW {Mon, Tue, Wed, Thu, Fri, Sat, Sun} |
    HitDoW {Mon, Tue, Wed, Thu, Fri, Sat, Sun} | OW.RT {o, r} |
    Infants {0, 1, 2++} | Type Flight {non-stop, conection} |
    User Type {new, repeat} | Log Status {LogIn, noLogIn} |
    Currency {EUR, USD} }
```

Figura 4.2 Ejemplo del conjunto propuesto de reglas de producción.

salida en virtud del mapeo genotipo-fenotipo, lo cual es una característica que le permite tener éxito en muchas aplicaciones del mundo real [128], y versatilidad para explorar grandes espacios de búsqueda. El método propuesto selecciona y filtra algunos de los atributos para adaptar el conjunto de datos antes de que se introduzca a los algoritmos de SD. El objetivo es proporcionar las mejores características para que un algoritmo de SD pueda encontrar patrones interesantes e identificar los mejores individuos en los que la probabilidad de reservar un vuelo aumenta si se recibe una oferta de descuento. A continuación se muestran los pasos principales que sigue el algoritmo de GE:

- **Codificación**. El algoritmo de GE propuesto codifica a los individuos como cadenas simbólicas (longitud fija), que luego se expresan como entidades cromosómicas lineales de diferentes tamaños, considerando el conjunto de no terminales formados por quince atributos del conjunto de datos $N = \{$*CharAdults, CharChildren, NDO, CharTripDuration, TypeDoW, NoOfPassengers, SearchDT, FlightDoW, HitDoW, OW.RT, Infants, Type Flight, User Type, Log Status, Currency*$\}$ y el conjunto de terminales los cuales son las acciones a realizar para filtrar el conjunto de datos $T = \{=, !=\}$. El símbolo de inicio de la gramática es $S =< e >$ y P (el conjunto de reglas de producción) que se presenta en la Figura 4.2. En este punto, se debe comentar que los valores de los atributos que forman el conjunto de no terminales son solo atributos categóricos, la razón de esto es para facilitar la interpretación de las reglas. El atributo *Booking*

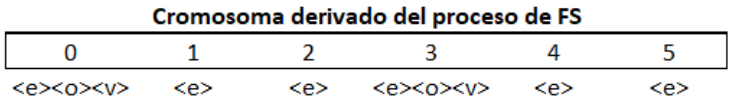

Figura 4.3 Ejemplo de un cromosoma derivado producido por la gramática propuesta..

es el atributo objetivo, el cual es utilizado por el algoritmo SD para la extracción de reglas, por lo tanto, no se integra al conjunto de no terminales. La Figura 4.3 representa un ejemplo de un cromosoma derivado que comprende un individuo de la población inicial, en la que los genes 0 y 3 son atributos que están seleccionados y en los que se aplica una condición de filtro usando uno de sus valores; el resto de los genes son los atributos seleccionados que serán parte del nuevo conjunto de datos que alimentara el algoritmo de SD para la extracción de reglas. La Figura 4.4 representa un ejemplo de un cromosoma derivado convertido en su equivalente cromosoma de análisis. La solución representa un cromosoma en el que el conjunto de datos será formado por los atributos *TypeDoW, NDO, CharTripDuration, CharAdults, CharChildren and HitDow* y filtrara el atributo *TypeDow* con valores iguales a "P" (*peak days*) y el atributo *CharAdults* con valores iguales a "*One*". Es importante resaltar que se requiere aplicar algunas restricciones, por ejemplo, los cromosomas deben contar con al menos cuatro atributos y tener un máximo de 8 atributos. Esto se aplica con la finalidad de controlar para controlar el tamaño de las soluciones, lo que representa un total de 409,705,619,895 combinaciones.

- **Conjunto inicial de soluciones.** Para obtener el conjunto inicial de soluciones, nuestro algoritmo genera aleatoriamente cromosomas lineales para codificar la información según las reglas de producción P, mapeando los elementos de N a T. El Algoritmo 5 muestra el pseudo-código para la generación del conjunto inicial de soluciones en su forma derivada. El algoritmo comienza con la creación aleatoria de un número entero al que se aplica una operación de módulo para determinar cuál de las dos opciones se establecerá en el gen de un cromosoma $< e >< o >< v >$ o $< e >$, si el resultado de la operación es igual a 0 entonces la primera opción es elegida, si el resultado es diferente de 0 entonces la segunda opción es elegida para ser establecida en el gen. Este proceso se ejecuta hasta que todos los genes de cada uno de los cromosomas de la población inicial sean completen.

Después de que se crean los cromosomas derivados, se ejecuta un proceso para crear los cromosomas de análisis y determinar que atributos del conjunto de datos formarán

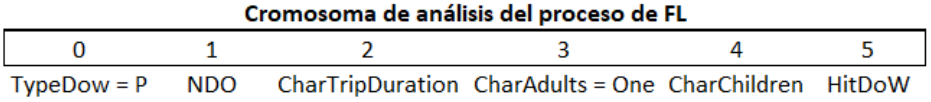

Figura 4.4 Ejemplo de un cromosoma de análisis producido por la gramática propuesta.

Algoritmo 5 Población inicial en forma derivada

```
Input pop-size, chromosome-size, S
Output solutions

 1: solutions = []
 2: for i = 1 to pop-size do
 3:     chromosome = [i]
 4:     for j = 1 to chromosome-size do
 5:         startElement = get.random.Integer mod 2
 6:         if startElement = 0 then
 7:             chromosome[j] = S[0]
 8:         else
                chromosome[j] = S[1]
 9:         end if
10:         chromosome[j] = length(chromosome-size)
11:     end for
12:     solutions[i] = chromosome
13: end for
14: return solutions
```

parte de las soluciones. El Algoritmo 6 define el pseudo-código de este proceso. El algoritmo recibe el conjunto inicial de soluciones en su forma derivada, la línea 2 representa el ciclo general para leer cada cromosoma, el cual se va a decodificar y transformar para obtener los conjuntos de datos. De la línea 3 a la línea 13 muestran dicha decodificación, donde se puede observar que cada cromosoma está formado por un proceso aleatorio de selección de elementos de N, T y V, los cuales son los atributos, las operaciones a filtrar (=, !=) y los valores de cada atributo respectivamente.

- **Evaluación.** Este procedimiento se encarga de asignar un valor de aptitud para cada solución (ver Ecuación 4.3). En la metodología propuesta se tiene que evaluar qué tan buenos son los conjuntos de datos obtenidos a través del proceso FS. Para realizar esto, se aplica un método de SD a cada uno de los conjuntos de datos obtenidos para obtener una función de aptitud que se basa en una combinación de la métrica conocida como precisión relativa ponderada de la medida (WRAcc) y el soporte o

Algoritmo 6 Población inicial en forma de cromosoma de análisis

Input population, N, T, O, V
Output solutions

```
1:  solutions = []
2:  for i = 1 to pop-size do
3:      chromosome = population[i]
4:      for j = 1 to length(chromosome) do
5:          if chromosome[j] = < e >< o >< v > then
6:              a = get.random.element(N)
7:              b = get.random.element(O)
8:              c = get.random.element(V)
9:              chromosome[j] = a+b+c
10:         else
11:             chromosome[j] = get.random.element(N)
12:         end if
13:         chromosome[j] = length(chromosome)
14:     end for
15:     solutions[i] = chromosome
16: end for
17: return solutions
```

probabilidad (p) que obtiene cada subgrupo. La tasa WRAcc es una medida híbrida que se encuentra en algún lugar entre generalidad, interés y precisión [163]. Se utiliza la siguiente notación: Sea $n(X)$ el número de ejemplos cubiertos por una regla $X \rightarrow Y$, $n(Y)$ representan el número de ejemplos de la clase Y, y $n(YX)$ representa el número de ejemplos correctamente clasificados (verdaderos positivos). Se utiliza $p(YX)$ etc. para las probabilidades correspondientes. La regla de precisión, o regla de confianza en la terminología de aprendizaje de reglas de asociación, se define como $Acc(X \rightarrow Y) = p(Y|X) = p(YX)p(X)$. WRAcc [163] se define como:

$$WRAcc(X \rightarrow Y) = p(X) \times (p(Y|X)p(Y)) \tag{4.1}$$

El soporte se considera como una de las medidas de calidad más utilizadas en la minería de patrones. Esta métrica calcula el número de registros (n) incluidos en el subconjunto representado por la regla P $\rightarrow$ t, se define como:

$$Support = \frac{n(P \rightarrow t)}{n} \tag{4.2}$$

El valor de aptitud o fitness se asigna mediante la siguiente fórmula:

Algoritmo 7 Evaluación de las soluciones

Input t-solutions, e-solutions
Output Evaluated-solutions

```
1: for i = 1 to size(t-solutions) do
2:    if n of features in solution[i] <4 then
3:       fitness = 0
4:    else
5:       create SD model
6:       set fitness
7:    end if
8:    Evaluated-Solutions = [e-solution[i], fitness]
9: end for
10: return Evaluated-solutions
```

$$Fitness = \frac{(WRAcc \times Support)}{2} \tag{4.3}$$

El algoritmo 7 muestra el pseudocódigo del proceso de evaluación. Este algoritmo recibe un conjunto de soluciones como conjuntos de datos (*t-solutions*), y el conjunto de soluciones en forma codificada (*e-solutions*). El Algoritmo funciona como el bucle principal para evaluar cada solución t creada en el Algoritmo 6. En cada iteración, la propuesta verifica si la solución contiene al menos cuatro atributos del conjunto de datos original, evaluando la solución a través de un algoritmo de SD. De lo contrario, si la solución no contiene al menos cuatro de los atributos originales, entonces se asigna un valor de aptitud de 0. La restricción de incluir al menos cuatro o más de los atributos originales tiene la finalidad de obtener reglas más largas. Finalmente, este procedimiento devuelve las soluciones en su nueva forma codificada junto con su respectivo valor de aptitud. Cabe mencionar que en este paso se desea obtener patrones de usuarios que realizaron reservas, por lo tanto, el método de SD se enfoca en obtener patrones que tiene como objetivo el atributo *Booking* y su valor igual a *"yes"*.

Para obtener las reglas y calcular el valor de aptitud, se propone el uso de tres métodos de SD, que están disponibles en el conocido paquete rsubgroup [11]:

- El método de búsqueda exhaustiva SD-map [12] depende de un valor mínimo de umbral de soporte para reducir el espacio de búsqueda. En situaciones donde el valor mínimo de soporte se establece en cero, luego SD-map realiza una búsqueda exhaustiva para cubrir todo el espacio de búsqueda sin podar. Su estructura

representa cada nodo mediante el uso de un elemento y la frecuencia del conjunto de elementos indicados desde la raíz hasta ese nodo. El conjunto resultante incluye solo subgrupos cuyo soporte es mayor al valor mínimo predefinido. Después de eso, una medida de calidad deseada se calcula para cada una de las soluciones, y aquellos subgrupos que no satisfacen el mínimo umbral de calidad no se consideran.

- El método *Beam-Search* [105], es una adaptación de CN2-SD el cual es un algoritmo de iteración que busca una conjunción de atributos que se comporta de acuerdo con la medida de entropía. Por lo tanto, las reglas que cubren un gran cantidad de registros de datos para un solo valor objetivo son preferibles a los que cubren pocos registros de otros valores objetivos. Este método funciona de forma iterativa, manteniendo reglas que cubren las transacciones que no han sido previamente cubiertas. Este proceso iterativo se lleva a cabo hasta que no hay más reglas que cubrir.
- BSD [107] (el método de descubrimiento de grupos basado en bisset) utiliza una estrategia de ramificación y vinculación, donde un espacio de búsqueda condicionado se extrae recursivamente, similar a SD-Map, donde en su fase de inicialización construye dos conjuntos de bits para cada selector involucrado en el descubrimiento del subgrupo. El primer conjunto de bits representa los casos de la base de datos con un valor objetivo positivo, y después los casos con un valor objetivo negativo. La construcción de estos conjuntos de bits se puede lograr en un solo paso a través de la base de datos. Los pasos finales del algoritmo pueden operar sobre las estructuras de datos generadas y no necesita más pases a la base de datos, lo que lo hace un algoritmo rápido mientras descubre subgrupos correspondientes.

- **Operadores genéticos.** Aquí se describen los operadores genéticos utilizados por el algoritmo propuesto. Primero, para la selección se aplica una selección de ruleta [183], para lo cual a cada uno de los individuos de la población se le asigna una parte proporcional a su ajuste de una ruleta, de tal forma que la suma de todos los porcentajes sea la unidad. Los mejores individuos reciben una porción de la ruleta mayor que la recibida por los peores, la población está ordenada basándose en el ajuste, por lo que las porciones más grandes se encuentran al inicio de la ruleta. La ruleta se gira tantas veces como soluciones haya en la población para mantener el tamaño de la población constante. La selección de la ruleta, se seleccionan las soluciones según la aptitud y la suerte, lo que significa que algunas veces las mejores soluciones se pueden perder.

Sin embargo, al combinar la selección de la ruleta con la clonación de las mejores soluciones de cada generación, se garantiza que al menos las mejores soluciones no se pierdan, por lo que aplica una clonación de las mejores soluciones. A esta técnica de clonar las mejores soluciones en cada generación se conoce como elitismo y se utiliza en la mayoría de los esquemas de selección estocástica. A continuación, en cuanto al operador de cruce, el enfoque propuesto sigue un típico punto único de cruce basado en la distancia de hamming [28] el cual simplemente re-combina individuos cuya información genética es diferente, si esta condición se cumple entonces el cruce actúa en un solo punto de la solución. El operador de cruce elige un gen al azar para las soluciones padres, después intercambia las subcadenas, creando dos descendientes. Además, se aplica un método de mutación típico basado en un solamente en un punto. Finalmente, después de un número de iteraciones del algoritmo, se introducen nuevos individuos aplicando un proceso de reinicialización, el cual genera una nueva población usando los pasos descritos para generar la población inicial. Esta nueva población se integra junto con las mejores soluciones. Todo proceso continúa hasta que se cumple el número máximo de iteraciones a realizar.

Finalmente, se selecciona solo la mejor regla o subgrupo obtenido de cada solución del proceso FS, las cuales se almacenan en un repositorio de reglas para ser analizadas en la siguiente etapa. Se selecciona solamente un subgrupo (el mejor subgrupo) de los generados para cada solución por una razón: para evitar reglas o subgrupos redundantes. Se entiende por reglas redundantes, aquellas reglas que tienen atributos, longitud y aptitud similares, por lo tanto, la regla con mejor aptitud es la que se selecciona para formar parte del repositorio de reglas.

4.3. Modelo de probabilidades

Esta etapa de la metodología se divide en dos fases, (1) el cálculo de la probabilidad de reservar un vuelo y (2) el proceso de análisis de incremento de probabilidad. Para calcular las probabilidades de reserva se utiliza un algoritmo de clasificación (XGBoost [141]), el cual predice la probabilidad de que un pasajero potencial vaya a reservar un vuelo sin ningún tipo de oferta o descuento, se utiliza el atributo *Booking* como la clase a predecir. Cabe mencionar que, en este paso lo que interesa es obtener un valor numérico (las probabilidades) no la predicción actual de la clase; por lo que, se utiliza el algoritmo de clasificación como un clasificador suave.

Después de obtener las probabilidades, el segundo paso en esta etapa es probar cada regla almacenada en el repositorio, filtrando el conjunto de datos original para obtener cada subgrupo, por ejemplo, si la regla es la siguiente: "$NDO = 0-7, CharAdults = 1, Infants = None, FlightDoW = Thu$"; entonces, el conjunto de datos se filtra utilizando los parámetros de la regla. Después de filtrar el conjunto de datos, se simula un descuento del 20 por ciento aplicando una reducción en el atributo *PricePerPassenger*, que representa el precio a pagar por una reserva de vuelo. Cada uno de estos subgrupos, se introducen en el modelo de clasificación que calcula las nuevas probabilidades. Finalmente, se comparan estas nuevas probabilidades en contra las probabilidades originales para obtener la medida del incremento de probabilidad a la que le denominamos *boost*: la cual es básicamente resultado de sustraer la probabilidad de un subgrupo con descuento menos la probabilidad del subgrupo sin descuento. Los subgrupos que obtienen el mejor boost son los que se seleccionan para ser parte de la siguiente etapa. Esta etapa es necesaria para identificar aquellos subgrupos de pasajeros cuya probabilidad de reservar un vuelo aumenta si reciben un descuento.

4.4. Estudio experimental y descubrimiento de reglas

El objetivo inicial de esta sección es determinar cuál de los tres algoritmos de SD que se propusieron en la Sección 4.2 de este capítulo de memoria de tesis, obtiene mejores resultados para el problema en cuestión. Por lo tanto, el proceso de evaluación se lleva a cabo aplicando los algoritmos antes mencionados. De manera que, se aplicó la metodología propuesta a un escenario real que comprende una serie de datos privados de una aerolínea comercial con actividad de reservas de vuelo durante el período 1 de diciembre de 2021 al 15 de febrero de 2022.

En esta sección se llevan a cabo varios experimentos para demostrar la utilidad del enfoque GE-FS propuesto. En este sentido, se utilizan tres algoritmos de SD en el proceso procedimiento de evaluación para crear un repositorio de reglas. Cada una de las reglas que son parte del repositorio han sido probadas por el modelo de clasificación suave obtenido en la etapa anterior de nuestra metodología. Por lo tanto, se obtuvieron las probabilidades de reservar un vuelo si se recibe un descuento para cada miembro de cada subgrupo que forma parte del repositorio de reglas. El objetivo final es demostrar que el uso de una estrategia de FS evolutiva es apropiada para este problema, por lo tanto, se comparó el comportamiento de los algoritmos con y sin el uso de la estrategia de FS utilizando tres métricas: (1) el tamaño de la regla (Rs) que es el número de antecedentes que forman parte de la regla, (2) el *boost* (ver ecuación 4.4) que es la diferencia entre la probabilidad de reservar un vuelo con descuento por subgrupo (SubP) menos la probabilidad de reservar un vuelo sin un descuento

Algoritmo	Org. Data (Rs)	GE-FS (Rs)	Org. Data (Boost)	GE-FS (Boost)	Org. Data (p)	GE-FS (p)
Beam	2.2	4.9	0.030	0.036	0.208	0.229
BSD	2.2	4.6	0.030	0.031	0.208	0.205
SD-Map	2.2	4.9	0.030	0.036	0.208	0.229

Tabla 4.1 Resultados de la fase experimental.

por subgrupo (IndP) y (3) la probabilidad de que un pasajero pertenezca a un subgrupo y tenga un boost positivo (SgPosBoost) ver ecuación 4.5.

$$Boost = SubP - IndP \tag{4.4}$$

$$p = \frac{SgPosBoost}{n} \tag{4.5}$$

Todos los algoritmos SD se ejecutaron utilizando toda la información disponible, después de que las fases de recopilación de datos y de preprocesamiento se completaron. Comparando el tamaño de las reglas obtenidas en ambos experimentos (ver Tabla 4.1, que reporta los mejores resultados de la fase experimental), se puede observar que el enfoque GE-FS produce reglas más largas, que en este caso son mejores. Mientras la reglas sean más largas, estas restringen en mayor medida los subgrupos, lo cual ayuda a controlar el riesgo de la perdida de dinero a gran escala. En promedio, el enfoque GE-FS produce reglas con una longitud de entre 4.6 a 4.9, mientras que cuando los algoritmos de SD extraen las reglas directamente del conjunto de datos originales produce reglas con una longitud de solo 2.2 en promedio. Además, el enfoque GE-FS obtiene una ganancia mínima de *boost*, lo que significa que los subgrupos descubiertos producen resultados similares en la predicción de pasajeros potenciales, demostrando que la probabilidad de reservar un vuelo aumentará si este tipo de pasajeros recibe una oferta. En comparación con el primer experimento, cuando los algoritmos *Beam* y *SD-Map* son alimentados por los conjuntos de datos que produce el enfoque GE-FS, estos extraen subgrupos en los cuales la probabilidad de obtener un valor positivo de *boost* aumenta al aplicarse un descuento, este incremento es de 0.02 puntos aproximadamente.

En cuanto a las reglas descubiertas, se ha realizado una comparación del repositorio de reglas creado por el enfoque propuesto en contra del repositorio creado por los algoritmos de SD utilizando solo los datos originales. Para demostrar estas diferencias, se toma como muestra el algoritmo SD-Map del cual se analizan las mejores reglas obtenidas en ambos experimentos. La Tabla 4.2 muestra las mejores reglas en el repositorio que fue generado utilizando el conjunto de datos original. De estas reglas se puede observar que el atributo

RuleNo	Description	RuleSize	Boost	p
1	$Bins_T rip_D uration = 0, Children = NO, Infants = NO$	3	0.033	0.217
2	$Bins_T rip_D uration = 0, Children = NO, OW.RT = o$	3	0.033	0.216
3	$Bins_T rip_D uration = 0, Children = NO, OW.RT = o, Infants = NO$	4	0.033	0.217
4	$Bins_T rip_D uration = 0, Infants = NO, OW.RT = o$	3	0.033	0.221
5	$LogStatus = LoggedIn, Children = NO, Infants = NO$	3	0.015	0.143
6	$OW.RT = o, Children = NO, Infants = NO$	3	0.033	0.217
7	$TypeFlight = non-stop, Children = NO, Infants = NO$	3	0.035	0.239

Tabla 4.2 Reglas descubiertas por SD-Map

RuleNo	Description	RuleSize	Boost	p
1	$TypeFlight = non-stop, LogStatus = LoggedIn, FlightDoW = Wed, Children = NO, Infants = NO$	5	0.052	0.282
2	$LogStatus = LoggedIn, OW.RT = o, HitWeekDay = WeekDay, UserType = New, TypeDoW = P, TypeFlight = non-stop$	6	0.050	0.264
3	$LogStatus = LoggedIn, Children = NO, TypeFlight = non-stop, OW.RT = o, TypeDoW = P, Infants = NO$	6	0.047	0.257
4	$TypeFlight = non-stop, FlightDoW = Wed, HitWeekDay = WeekDay, Children = NO, Infants = NO$	5	0.031	0.213
5	$TypeFlight = non-stop, FlightDoW = Wed, Children = NO, Infants = NO$	4	0.031	0.212
6	$TypeFlight = non-stop, LogStatus = NotLoggedIn, HitWeekDay = WeekDay, Children = NO, TypeDoW = OP$	5	0.020	0.238
7	$Bins_T rip_D uration = 0, TypeFlight = non-stop, FlightDoW = Wed, Children = NO, Infants = NO$	5	0.031	0.190
8	$OW.RT = o, TypeFlight = non-stop, FlightDoW = Wed, Children = NO, Infants = NO$	5	0.031	0.190
9	$Bins_T rip_D uration = 0, LogStatus = NotLoggedIn, Children = NO, TypeDoW = OP, TypeFlight = non-stop$	5	0.019	0.170

Tabla 4.3 Reglas descubiertas por GE-FS y SD-Map.

Children = 0 es el que se encuentra con mayor frecuencia dentro de los subgrupos extraídos. A continuación se describen las características más interesantes:

- Se puede observar que las reglas del 1 al 4 básicamente representan el mismo subgrupo, los tamaños de las reglas son los mismos a excepción de la regla número 3 que es la única que tiene una longitud de 4. Estas reglas muestran un subgrupo en que si los pasajeros viajan sin ningún acompañante, de forma directa (viaje sin escalas), la probabilidad de obtener un incremento de *boost* al recibir un descuento es de aproximadamente 0.033 y con una probabilidad de aplicar una reserva de 0.217.

- La regla número 7 es otro subgrupo redundante, el cual es muy similar a las primeras cuatro reglas. Esta regla, también muestra pasajeros que viajan solos en un vuelo sin escalas; la principal diferencia es que este subgrupo obtiene una mayor *boost* (0.035) y una mayor probabilidad de reservar un vuelo (0.249).

- Al aplicar SD-Map directamente el conjunto de datos original, el descubrimiento de subgrupos significativos es muy limitado. De manera que, la generación de una tabla dinámica de ofertas en un entorno real no es una opción viable.

La Tabla 4.3 muestras las mejores reglas en el repositorio obtenido por el enfoque GE-FS, en el cual se puede observar reglas de mayor longitud y con menor redundancia entre los subgrupos extraídos. A continuación se describen las características más importantes:

- La regla número 2 muestra una serie de individuos que realizan su búsqueda en el sitio Web entre semana (de lunes a viernes), los cuales intentan viajar entre los días jueves a lunes (*Peak Days*) en un vuelo sin escalas; quienes incluso se han logueado en el sistema. El incremento de *boost* obtenido por este tipo de personas es de 0.05, por lo tanto, la probabilidad de realizar una reserva si este su tipo de pasajeros potenciales reciben una oferta es de 0.26.

- La regla número 6 es otro subgrupo interesante, el cual muestra pasajeros potenciales, los cuales no se han logueado en el sistema y quienes tienen una preferencia por viajar en los que se conoce como *off-peak days* (martes o miércoles), y quienes realizan la búsqueda de vuelos en el sitio web de lunes a viernes, para realizar el viaje solos. Debido a estos patrones encontrados, se puede observar una cierta sensibilidad de precios, porque están buscando viajar en días en los que los precios regularmente son más bajos. El incremento de *boost* que este subgrupo obtiene es de 0.02, con una probabilidad total de 0.24 para realizar una reserva si ellos reciben una oferta.

- En general, el enfoque de GE-FS es capaz de encontrar un diversificado número de subgrupos, evitando la redundancia, potenciando el incremento de probabilidad en los pasajeros potenciales para reservar un vuelo si ellos reciben una oferta.

4.5. Conclusiones del capítulo

El descubrimiento de subgrupos interesantes en los que las probabilidades para reservar un vuelo aumentan si los individuos que pertenecen a estos subgrupos reciben un descuento es una tarea complicada de llevar a cabo. De manera que, para solucionar este problema, se propuso una metodología para crear un repositorio de ofertas dinámicas a través de la implementación de un algoritmo GE-FS. Debido a la experimentación realizada con la implementación de esta metodología, se puede concluir lo siguiente:

- El enfoque GE-FS es capaz de descubrir subgrupos interesantes en los cuales los individuos que pertenecen a estos, su probabilidad de reservar un vuelo incrementa si ellos reciben una oferta.

- El enfoque GE-FS es capaz de crear un repositorio de ofertas dinamicas, el cual puede ser utilizado en un entorno real, debido a que evita la redundancia de las reglas, obteniendo un modelo de alta interpretacion.

- Finalmente, la metodología también es capa de encontrar reglas de mayor longitud, lo cual es importante en un entorno comercial porque con esto se puede restringir de

manera adecuada quien recibe una oferta, evitando patrones globales en los cualquier tipo de pasajero pueda recibir una oferta, minimizando el riego de perdida de capital.

Capítulo 5

Conclusiones y trabajo futuro

En este capítulo se presentan los comentarios finales en forma de conclusiones del trabajo realizado, enumeración de las publicaciones tanto presentadas en congresos como las publicadas en revistas internacionales, así como la descripción de las futuras líneas de investigación que se proponen como continuación del trabajo presentado en esta memoria de tesis.

5.1. Conclusiones

Las principales conclusiones obtenidas tras el desarrollo del trabajo realizado en esta tesis son las siguientes:

1. Tras hacer una búsqueda y revisión bibliográfica de la literatura relacionada acerca de los métodos que se utilizan para enfrentar la guerra de precios y los métodos de precios dinámicos dentro de la industria aérea, se ha encontrado una serie de trabajos interesantes. Esto indica que ambos problemas que se tratan de resolver en esta tesis es de actualidad y de gran importancia para las aerolíneas comerciales. De hecho, la implementación de metodologías automatizadas capaces de producir modelos de alta interpretación es de vital importancia para los equipos de analistas de precios y ganancias, debido a que se pueden reducir el número de errores humanos, aumentar la capacidad y velocidad de análisis, y la extracción de patrones o reglas interesantes.

2. La tarea de predecir y extraer conocimiento en las tarifas aéreas es una tarea muy difícil de conseguir, principalmente por dos causas: la alta cantidad de tarifas a analizar y los cambios constantes que suceden diariamente. Para solventar estas dos dificultades, en esta tesis se ha propuesto la una metodología en la cual se integra un algoritmo

de programación de expresión genética (GEP); el cual es capaz de imitar las tareas de limpieza y transformación de datos que a menudo realizan los analistas de precios dentro de una aerolínea. Por lo tanto, este algoritmo crea conjuntos de datos transformados, los cuales alimentan a un algoritmo de clasificación para predecir la clase de la tarifa a la que pertenece y extraer una serie de reglas, creando un modelo de fácil interpretación que puede ser utilizado por los analistas de precios. La metodología demostró una mejora tanto en las métricas de clasificación, como en la métrica de interpretación, siendo capaz de generar un modelo de alta interpretación. Esta metodología fue probada con un conjunto de datos reales que han sido publicados por la aerolínea bandera de Canadá (Air Canada).

3. La identificación de individuos cuya probabilidad de realizar una reserva de vuelo aumente si recibe un descuento mientras hace una búsqueda de tarifas en una página web es una tarea de enorme interés para las aerolíneas. En esta tesis doctoral, se ha propuesto una metodología que tiene como finalidad la creación de un repositorio de reglas que permitan identificar subgrupos de pasajeros, los cuales la probabilidad de efectuar una reserva de vuelo aumenta si estos reciben una oferta. Dentro de esta metodología, se ha propuesto un algoritmo de gramáticas evolutivas (GE) el cual funciona como un selector de características, creando diversos conjuntos de datos que alimentan a un algoritmo de SD para la extracción de reglas, generando un repositorio con los mejores subgrupos el cual funciona como una tabla de ofertas dinámicas y como un modelo de interpretación a la vez. La metodología demostró ser capaz de generar un repositorio de reglas únicas, evitando la redundancia de las mismas; así mismo, estos subgrupos descubiertos muestran pasajeros cuya probabilidad de efectuar una reserva aumenta si estos reciben una oferta. La metodología fue probada en un escenario real utilizando datos privados de una aerolínea comercial.

4. Las dos metodologías propuestas demostraron que la utilización de técnicas de DM, como son los algoritmos de clasificación basados en reglas y los métodos de SD, en conjunto con algún tipo de algoritmo evolutivo, pueden ser muy eficientes para resolver problemas a los cuales se enfrentan las aerolíneas actualmente.

5. Este trabajo explica como se pueden crear modelos de alta interpretación, utilizando métodos de clasificación y métodos de SD, los cuales pueden ayudar a la generación de ganancias dentro de un entorno comercial altamente competitivo.

5.2. Trabajo futuro

Como líneas de trabajo futuro y que sirven de continuación y mejora de esta tesis, se plantean las siguientes tareas:

1. Aplicar las dos metodologías propuestas de predicción de tarifas aéreas para enfrentar la guerra de precios y la creación de una tabla dinámica de ofertas en un entorno con múltiples aerolíneas comerciales. El objetivo es realizar más pruebas con un volumen de datos mayor, donde se integren más rutas, ya que para esta tesis solo se ha experimentado con un número limitado de mercados.

2. Adaptar ambas metodologías dentro de un proceso de *data streams* que permita identificar cambios significativos en las reglas o subgrupos que forman parte de los modelos de interpretación, y los cuales tengan la capacidad de retroalimentación para adaptar/eliminar reglas o subgrupos obsoletos e integran los nuevos patrones que se muestren en los datos. Actualmente, la información fluye y cambia continuamente segundo a segundo, por lo que, la industria aérea no es la excepción. Debido a esto, si se aplica un paso o proceso dentro ambas metodologías propuestas para identificar cambios significativos en las reglas encontradas, sería de gran utilidad para una aerolínea, ya que le ayudaría a generar mayores ingresos y proporcionaría información relevante y actual de los cambios que están sucediendo en el mercado y a quien poder realizar algún tipo de oferta o descuento para incentivar el incremento de reservas de forma dinámica y totalmente automatizada.

3. Para la metodología que es propuesta para enfrentar la guerra de precios se plantea desarrollar una herramienta software específica que esté orientada para ser usada por un equipo de analistas de precios, los cuales no son expertos en DM. El objetivo de la herramienta sería integrar y facilitar todo el proceso de descubrimiento de conocimiento, desde la descarga y el pre-procesado de los datos, la aplicación del algoritmo evolutivo de aprendizaje de características (FL), la ejecución de un método de clasificación, hasta la visualización del modelo de interpretación. De esta forma se evitaría tener que hacer el pre-procesado, transformación de datos y análisis e interpretación de tarifas de forma manual.

4. Para la metodología que tiene como finalidad la creación de una tabla de ofertas dinámicas, desarrollar un proceso de backend que tenga como objetivo ejecutar cada uno de los pasos de la metodología para hacer la actualización de la tabla de ofertas y que pueda ser implementada en el sitio web de una aerolínea y que se realicen

los descuentos de forma dinámica. De esta manera, la tabla de ofertas reflejaría los cambios más recientes en el comportamiento mostrado por los pasajeros que buscan vuelos en el sitio web.

5.3. Publicaciones asociadas a la tesis

Con el propósito de divulgar los distintos resultados obtenidos de la investigación realizada en esta tesis, se han generado dos publicaciones, las cuales han sido presentadas, una como artículos de revista (1) y ponencia en un congreso internacional (2). A continuación se detallan ambos trabajos:

1. Facing up fare war: generating competitive price models with gene expression programming. Barron, Marco Antonio, Jose Maria Luna, and Sebastian Ventura. IEEE Access, 2022, DOI: 10.1109/ACCESS.2022.3225435.

2. Dynamic Airline Discounts using an Evolutionary Subgroup Discovery Methodology. Barron, Marco Antonio, Jose Maria Luna, and Sebastian Ventura. 2022 IEEE International Conference on Omni-layer Intelligent Systems (COINS). IEEE Computer Society, 2022. DOI: 10.1109/COINS54846.2022.9854942.

Bibliografía

[1] Abellán, J. and Moral, S. (2003). Building classification trees using the total uncertainty criterion. *International Journal of Intelligent Systems*, 18(12):1215–1225.

[2] Aggarwal, C. C. (2015). *Data mining: the textbook*. Springer.

[3] Aggarwal, C. C., Bhuiyan, M. A., and Hasan, M. A. (2014). Frequent pattern mining algorithms: A survey. In *Frequent pattern mining*, pages 19–64. Springer.

[4] Aha, D. W., Kibler, D., and Albert, M. K. (1991). Instance-based learning algorithms. *Machine learning*, 6(1):37–66.

[5] Alshammari, M. and Mezher, M. (2020). A comparative analysis of data mining techniques on breast cancer diagnosis data using weka toolbox. *(IJACSA) International Journal of Advanced Computer Science and Applications*, 8:224–229.

[6] Angeline, P. J. (1994). Genetic programming: On the programming of computers by means of natural selection: John r. koza, a bradford book, mit press, cambridge ma, 1992, isbn 0-262-11170-5, xiv+ 819pp., us 55,00.

[7] Archak, N., Ghose, A., and Ipeirotis, P. G. (2011). Deriving the pricing power of product features by mining consumer reviews. *Management science*, 57(8):1485–1509.

[8] Arnoud, V. (2015). den boer. *Dynamic pricing and learning: Historical origins, current research, and new directions. Surveys in Operations Research and Management Science*, 20(1):1–18.

[9] Arrieta, A. B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., García, S., Gil-López, S., Molina, D., Benjamins, R., et al. (2020). Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information fusion*, 58:82–115.

[10] Atzmueller, M. (2015). Subgroup discovery. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 5(1):35–49.

[11] Atzmueller, M., Atzmueller, M. M., and Java, S. (2021). Package 'rsubgroup'.

[12] Atzmueller, M. and Puppe, F. (2006). Sd-map–a fast algorithm for exhaustive subgroup discovery. In *European Conference on Principles of Data Mining and Knowledge Discovery*, pages 6–17. Springer.

[13] Bay, S. D. and Pazzani, M. J. (2001). Detecting group differences: Mining contrast sets. *Data mining and knowledge discovery*, 5(3):213–246.

[14] Begum, S. A. and Devi, O. M. (2011). Fuzzy algorithms for pattern recognition in medical diagnosis. *Assam University Journal of Science and Technology*, 7(2):1–12.

[15] Bel, L., Allard, D., Laurent, J.-M., Cheddadi, R., and Bar-Hen, A. (2009). Cart algorithm for spatial data: Application to environmental and ecological data. *Computational Statistics & Data Analysis*, 53(8):3082–3093.

[16] Bernardo, J., Bayarri, M., Berger, J., Dawid, A., Heckerman, D., Smith, A., West, M., et al. (2003). The variational bayesian em algorithm for incomplete data: with application to scoring graphical model structures. *Bayesian statistics*, 7(453-464):210.

[17] Bertsimas, D. and Tsitsiklis, J. (1993). Simulated annealing. *Statistical science*, 8(1):10–15.

[18] Beyer, H.-G. and Schwefel, H.-P. (2002). Evolution strategies–a comprehensive introduction. *Natural computing*, 1(1):3–52.

[19] Bhargava, N., Jain, A., Kumar, A., and Le, D.-N. (2017). Detection of malicious executables using rule based classification algorithms. In *ICITKM*, pages 35–38.

[20] Bousquet, O., von Luxburg, U., and Rätsch, G. (2004). Advanced lectures on machine learning. vol. 3176 of lecture notes in computer science.

[21] Boutorh, A. and Guessoum, A. (2014). Grammatical evolution association rule mining to detect gene-gene interaction. In *BIOINFORMATICS*, pages 253–258.

[22] Brabazon, A. (2018). Grammatical evolution in finance and economics: A survey. *Handbook of Grammatical Evolution*, pages 263–288.

[23] Brabazon, A., Matthews, R., O'Neill, M., and Ryan, C. (2002). Grammatical evolution and corporate failure prediction. In *Proceedings of the 4th Annual Conference on Genetic and Evolutionary Computation*, pages 1011–1018.

[24] Breiman, L., Friedman, J., Stone, C., and Olshen, R. (1984). Classification and regression trees chapman & hall. *New York*.

[25] Bremermann, H. J. (1958). *The evolution of intelligence: The nervous system as a model of its environment*. University of Washington, Department of Mathematics.

[26] Bremermann, H. J. et al. (1962). Optimization through evolution and recombination. *Self-organizing systems*, 93:106.

[27] Bremermann, H. J. and Rogson, M. (1964). An evolution-type search method for convex sets. Technical report, CALIFORNIA UNIV BERKELEY.

[28] Cannon, W. et al. (1932). The wisdom of the body norton. *New York, NY*.

[29] Chaudhari, A. and Mulay, P. (2019). A bibliometric survey on incremental clustering algorithm for electricity smart meter data analysis. *Iran Journal of Computer Science*, 2(4):197–206.

[30] Chen, L., Mislove, A., and Wilson, C. (2016). An empirical analysis of algorithmic pricing on amazon marketplace. In *Proceedings of the 25th international conference on World Wide Web*, pages 1339–1349.

[31] Chen, M. K. and Sheldon, M. (2016). Dynamic pricing in a labor market: Surge pricing and flexible work on the uber platform. *Ec*, 16:455.

[32] Chen, Y., Li, F., and Fan, J. (2015). Mining association rules in big data with ngep. *Cluster Computing*, 18(2):577–585.

[33] Chokkalingam, S. P. and Komathy, K. (2013). Comparison of different classifier in weka for rheumatoid arthritis. In *2013 International Conference on Human Computer Interactions (ICHCI)*, pages 1–6.

[34] Clark, P. and Niblett, T. (1989). The cn2 induction algorithm. *Machine learning*, 3(4):261–283.

[35] Cohen, W. W. (1995). Fast effective rule induction. *Machine learning proceedings 1995*, pages 115–123.

[36] Cramer, N. L. (1985). A representation for the adaptive generation of simple sequential programs. In *proceedings of an International Conference on Genetic Algorithms and the Applications*, pages 183–187.

[37] Damanik, I. S., Windarto, A. P., Wanto, A., Andani, S. R., Saputra, W., et al. (2019). Decision tree optimization in c4. 5 algorithm using genetic algorithm. In *Journal of Physics: Conference Series*, volume 1255, page 012012. IOP Publishing.

[38] Dehuri, S. and Cho, S.-B. (2008). Multi-objective classification rule mining using gene expression programming. In *2008 Third International Conference on Convergence and Hybrid Information Technology*, volume 2, pages 754–760. IEEE.

[39] Den Boer, A. V. (2015). Dynamic pricing and learning: historical origins, current research, and new directions. *Surveys in operations research and management science*, 20(1):1–18.

[40] Dong, G. and Bailey, J. (2012). *Contrast data mining: concepts, algorithms, and applications*. CRC Press.

[41] Došilović, F. K., Brčić, M., and Hlupić, N. (2018). Explainable artificial intelligence: A survey. In *2018 41st International convention on information and communication technology, electronics and microelectronics (MIPRO)*, pages 0210–0215. IEEE.

[42] Duan, L., Tang, C., Tang, L., Zuo, J., and Zhang, T. (2009). An effective microarray data classifier based on gene expression programming. In *2009 Fifth International Conference on Natural Computation*, volume 4, pages 523–527. IEEE.

[43] Duffy, J. and Engle-Warnick, J. (2002). Using symbolic regression to infer strategies from experimental data. In *Evolutionary computation in Economics and Finance*, pages 61–82. Springer.

[44] Eftimov, T. and Korošec, P. (2019). A novel statistical approach for comparing meta-heuristic stochastic optimization algorithms according to the distribution of solutions in the search space. *Information Sciences*, 489:255–273.

[45] Elmaghraby, W. and Keskinocak, P. (2003). Dynamic pricing in the presence of inventory considerations: Research overview, current practices, and future directions. *Management science*, 49(10):1287–1309.

[46] Farris, P. W., Bendle, N., Pfeifer, P. E., and Reibstein, D. (2010). *Marketing metrics: The definitive guide to measuring marketing performance*. Pearson Education.

[47] Faruqui, A. and Palmer, J. (2011). Dynamic pricing and its discontents. *Regulation*, 34:16.

[48] Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., et al. (1996). Knowledge discovery and data mining: Towards a unifying framework. In *KDD*, volume 96, pages 82–88.

[49] Ferreira, C. (2002a). Discovery of the boolean functions to the best density-classification rules using gene expression programming. In *European Conference on Genetic Programming*, pages 50–59. Springer.

[50] Ferreira, C. (2002b). Gene expression programming in problem solving. In *Soft computing and industry*, pages 635–653. Springer.

[51] Ferreira, C. (2006). *Gene expression programming: mathematical modeling by an artificial intelligence*, volume 21. Springer.

[52] Fiig, T., Le Guen, R., and Gauchet, M. (2018). Dynamic pricing of airline offers. *Journal of Revenue and Pricing Management*, 17(6):381–393.

[53] Fogel, D. B. (1995). Evolutionary computation: Toward a new philosophy of machine intelligence, institute of electrical and electronics engineers. *Inc, New York*, pages 155–171.

[54] Fogel, D. B. (1998a). *Artificial intelligence through simulated evolution*. Wiley-IEEE Press.

[55] Fogel, D. B. (1998b). Evolutionary computation. the fossil record. selected readings on the history of evolutionary computation. *Classifier Systems*.

[56] Fogel, D. B. (2006). Foundations of evolutionary computation. In *Modeling and Simulation for Military Applications*, volume 6228, page 622801. International Society for Optics and Photonics.

[57] Fogel, D. B., Fogel, L. J., and Atmar, J. W. (1991). Meta-evolutionary programming. In *Conference record of the twenty-fifth asilomar conference on signals, systems & computers*, pages 540–541. IEEE computer Society.

[58] Fogel, G. B. and Corne, D. W. (2003). An introduction to evolutionary computation for biologists. In *Evolutionary Computation in Bioinformatics*, pages 19–38. Elsevier.

[59] Fournier-Viger, P., Lin, J. C.-W., Kiran, R. U., Koh, Y. S., and Thomas, R. (2017). A survey of sequential pattern mining. *Data Science and Pattern Recognition*, 1(1):54–77.

[60] Fournier-Viger, P., Yang, P., Kiran, R. U., Ventura, S., and Luna, J. M. (2021). Mining local periodic patterns in a discrete sequence. *Information Sciences*, 544:519–548.

[61] Frank, E. and Witten, I. H. (1998). Generating accurate rule sets without global optimization.

[62] Fraser, A. (1968). The evolution of purposive behavior. purposive systems.

[63] Friedman, J., Hastie, T., Tibshirani, R., et al. (2000). Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *Annals of statistics*, 28(2):337–407.

[64] Fürnkranz, J. and Kliegr, T. (2015). A brief overview of rule learning. In *International symposium on rules and rule markup languages for the semantic web*, pages 54–69. Springer.

[65] Fürnkranz, J., Gamberger, D., and Lavrac, N. (2012). *Foundations of Rule Learning*. Cognitive Technologies. Springer.

[66] García, S., Fernández, A., Luengo, J., and Herrera, F. (2009). A study of statistical techniques and performance measures for genetics-based machine learning: accuracy and interpretability. *Soft Computing*, 13(10):959.

[67] García, S., Luengo, J., and Herrera, F. (2015). *Data Preprocessing in Data Mining*, volume 72 of *Intelligent Systems Reference Library*. Springer.

[68] Georgoulas, G., Gavrilis, D., Tsoulos, I. G., Stylios, C., Bernardes, J., and Groumpos, P. P. (2007). Novel approach for fetal heart rate classification introducing grammatical evolution. *Biomedical Signal Processing and Control*, 2(2):69–79.

[69] Goldberg, D. E. and Holland, J. H. (1988). Genetic algorithms and machine learning.

[70] Goodman, B. and Flaxman, S. (2017). European union regulations on algorithmic decision-making and a "right to explanation". *AI magazine*, 38(3):50–57.

[71] Gross, J. and Groß, J. (2003). *Linear regression*, volume 175. Springer Science & Business Media.

[72] Gunning, D. (2017). Explainable artificial intelligence (xai). *Defense advanced research projects agency (DARPA), nd Web*, 2(2):1.

[73] Hall, P. (2018). On the art and science of machine learning explanations. *arXiv preprint arXiv:1810.02909*.

[74] Haris NA, Abdullah M, O. A. R. F. (2014). Optimization and data mining for decision making. In *In 2014 World Congress on Computer Applications and Information Systems (WCCAIS)*, volume 10, pages 1–4. IEEE.

[75] Harsoor, A. S. and Patil, A. (2015). Forecast of sales of walmart store using big data applications. *International Journal of Research in Engineering and Technology*, 4(6):51–59.

[76] Hegland, M. (2007). The apriori algorithm–a tutorial. *Mathematics and computation in imaging science and information processing*, pages 209–262.

[77] Heil, O. P. and Helsen, K. (2001). Toward an understanding of price wars: Their nature and how they erupt. *International Journal of Research in Marketing*, 18(1-2):83–98.

[78] Hernández Orallo, J., Ramírez Quintana, M. J., and Ferri Ramírez, C. (2004). *Introducción a la Minería de Datos*. Pearson Educación.

[79] Herrera, F., Carmona, C. J., González, P., and Del Jesus, M. J. (2011). An overview on subgroup discovery: foundations and applications. *Knowledge and information systems*, 29(3):495–525.

[80] Hofer, C., Windle, R. J., and Dresner, M. E. (2008). Price premiums and low cost carrier competition. *Transportation Research Part E: Logistics and Transportation Review*, 44(5):864–882.

[81] Holland, J. H. (1992). Genetic algorithms. *Scientific american*, 267(1):66–73.

[82] Holland, J. H. et al. (1992). *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press.

[83] Holte, R. C. (1993). Very simple classification rules perform well on most commonly used datasets. *Machine learning*, 11(1):63–90.

[84] Hornik, K., B. C. . Z. A. (2009). Open-source machine learning: R meets weka. *Computational Statistics*, pages 225–232.

[85] Hossin, M. and Sulaiman, M. N. (2015). A review on evaluation metrics for data classification evaluations. *International journal of data mining & knowledge management process*, 5(2):1.

[86] Hu, J. and Guo, W. (2019). Flexibility analysis in waste-to-energy systems based on decision rules and gene expression programming. In *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, pages 988–993. IEEE.

[87] Hu, J. and Mojsilovic, A. (2007). High-utility pattern mining: A method for discovery of high-utility item sets. *Pattern Recognition*, 40(11):3317–3324.

[88] Huang, J. and Deng, C. (2009). A novel multiclass classification method with gene expression programming. In *2009 International Conference on Web Information Systems and Mining*, pages 139–143. IEEE.

[89] Iba, W. and Langley, P. (1992). Induction of one-level decision trees. In *Machine Learning Proceedings 1992*, pages 233–240. Elsevier.

[90] Iqbal, M., Azam, M., Naeem, M., Khwaja, A., and Anpalagan, A. (2014). Optimization classification, algorithms and tools for renewable energy: A review. *Renewable and Sustainable Energy Reviews*, 39:640–654.

[91] Jędrzejowicz, J. and Jędrzejowicz, P. (2010). Cellular gep-induced classifiers. In *International Conference on Computational Collective Intelligence*, pages 343–352. Springer.

[92] John, R. (1992). Koza. genetic programming: On the programming of computers by means of natural selection.

[93] Jdrzejowicz, J. and Jdrzejowicz, P. (2008). Gep-induced expression trees as weak classifiers. In *Industrial Conference on Data Mining*, pages 129–141. Springer.

[94] Kapania, N. R., Subosits, J., and Christian Gerdes, J. (2016). A sequential two-step algorithm for fast generation of vehicle racing trajectories. *Journal of Dynamic Systems, Measurement, and Control*, 138(9).

[95] Karakasis, V. K. and Stafylopatis, A. (2008). Efficient evolution of accurate classification rules using a combination of gene expression programming and clonal selection. *IEEE transactions on evolutionary computation*, 12(6):662–678.

[96] Kavšek, B. and Lavrač, N. (2006). Apriori-sd: Adapting association rule learning to subgroup discovery. *Applied Artificial Intelligence*, 20(7):543–583.

[97] Kavurucu, Y., Senkul, P., and Toroslu, I. H. (2009). Ilp-based concept discovery in multi-relational data mining. *Expert Systems with Applications*, 36(9):11418–11428.

[98] Khan, R. A., Suleman, T., Farooq, M. S., Rafiq, M. H., and Tariq, M. A. (2017). Data mining algorithms for classification of diagnostic cancer using genetic optimization algorithms. *Ijcsns*, 17(12):207.

[99] Kim, B., Khanna, R., and Koyejo, O. O. (2016). Examples are not enough, learn to criticize! criticism for interpretability. *Advances in neural information processing systems*, 29.

[100] Komarek, P. (2004). *Logistic regression for data mining and high-dimensional classification*. Carnegie Mellon University.

[101] Koza, J. R. et al. (1994). *Genetic programming II*, volume 17. MIT press Cambridge, MA.

[102] Kralj, P., Lavrac, N., Gamberger, D., and Krstacic, A. (2007). Supporting factors to improve the explanatory potential of contrast set mining: Analyzing brain ischaemia data. In *11th Mediterranean Conference on Medical and Biomedical Engineering and Computing 2007*, pages 157–161. Springer.

[103] Krämer, A., Friesen, M., and Shelton, T. (2018). Are airline passengers ready for personalized dynamic pricing? a study of german consumers. *Journal of Revenue and Pricing Management*, 17(2):115–120.

[104] Lantseva, A., Mukhina, K., Nikishova, A., Ivanov, S., and Knyazkov, K. (2015). Data-driven modeling of airlines pricing. *Procedia Computer Science*, 66:267–276.

[105] Lavrac, N., Kavsek, B., Flach, P., and Todorovski, L. (2004). Subgroup discovery with cn2-sd. *J. Mach. Learn. Res.*, 5(2):153–188.

[106] Leibold, A. O. S. and López, Y. A. (2016). Low cost carriers in mexico. In *The Low Cost Carrier Worldwide*, pages 119–132. Routledge.

[107] Lemmerich, F., Rohlfs, M., and Atzmueller, M. (2010). Fast discovery of relevant subgroup patterns. In *Twenty-Third International FLAIRS Conference*.

[108] Li, J., Liu, H., Downing, J. R., Yeoh, A. E.-J., and Wong, L. (2003). Simple rules underlying gene expression profiles of more than six subtypes of acute lymphoblastic leukemia (all) patients. *Bioinformatics*, 19(1):71–78.

[109] Li, J. and Wong, L. (2002). Identifying good diagnostic gene groups from gene expression profiles using the concept of emerging patterns. *Bioinformatics*, 18(5):725–734.

[110] Links, A. F. and BookDust, I. M. (2002). Alex s. fraser. *IEEE Transactions on Evolutionary Computation*, 6(5).

[111] Liu, B., Hsu, W., Ma, Y., et al. (1998). Integrating classification and association rule mining. In *Kdd*, volume 98, pages 80–86.

[112] Liu, X., Cai, Z., and Gong, W. (2008). An improved gene expression programming for fuzzy classification. In *International Symposium on Intelligence Computation and Applications*, pages 520–529. Springer.

[113] Liu, Z., Wynter, L., and Xia, C. (2002). *Pricing information services in a competitive market: avoiding price wars*. PhD thesis, INRIA.

[114] Luna, J. M., Ondra, M., Fardoun, H. M., and Ventura, S. (2018). Optimization of quality measures in association rule mining: an empirical study. *Int. J. Comput. Intell. Syst.*, 12(1):59–78.

[115] Luna, J. M., Pechenizkiy, M., Duivesteijn, W., and Ventura, S. (2020). Exceptional in so many ways - discovering descriptors that display exceptional behavior on contrasting scenarios. *IEEE Access*, 8:200982–200994.

[116] Marghny, M. and El-Semman, I. (2005). Extracting fuzzy classification rules with gene expression programming. In *ALML 2005 Conference*. Citeseer.

[117] Maroco, J., Silva, D., Rodrigues, A., Guerreiro, M., Santana, I., and de Mendonça, A. (2011). Data mining methods in the prediction of dementia: A real-data comparison of the accuracy, sensitivity and specificity of linear discriminant analysis, logistic regression, neural networks, support vector machines, classification trees and random forests. *BMC research notes*, 4(1):1–14.

[118] Márquez-Vera, C., Cano, A., Romero, C., and Ventura, S. (2013). Predicting student failure at school using genetic programming and different data mining approaches with high dimensional and imbalanced data. *Applied intelligence*, 38(3):315–330.

[119] Mauceri, S., Sweeney, J., and McDermott, J. (2021). One-class subject authentication using feature extraction by grammatical evolution on accelerometer data. In *Heuristics for Optimization and Learning*, pages 393–407. Springer.

[120] Mazouni, R. and Rahmoun, A. (2015). Agge: A novel method to automatically generate rule induction classifiers using grammatical evolution. In *Intelligent Distributed Computing VIII*, pages 279–288. Springer.

[121] McAfee, R. P. and Te Velde, V. (2006). Dynamic pricing in the airline industry. *Handbook on economics and information systems*, 1:527–67.

[122] Miller, T. (2018). Explanation in artificial intelligence: Insights from the social sciences.

[123] Mitchell, M. (1998). *An introduction to genetic algorithms*. MIT press.

[124] Molnar, C., Casalicchio, G., and Bischl, B. (2020). Interpretable machine learning–a brief history, state-of-the-art and challenges. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 417–431. Springer.

[125] Niinistö, J. (2020). Utilizing machine learning in data-driven pricing.

[126] Noaman, A. Y., Luna, J. M., Ragab, A. H., and Ventura, S. (2016). Recommending degree studies according to students' attitudes in high school by means of subgroup discovery. *International Journal of Computational Intelligence Systems*, 9(6):1101–1117.

[127] Novak, P. K., Lavrač, N., and Webb, G. I. (2009). Supervised descriptive rule discovery: A unifying survey of contrast set, emerging pattern and subgroup mining. *Journal of Machine Learning Research*, 10(2).

[128] O'Neill, M. and Ryan, C. (2001). Grammatical evolution. *IEEE Transactions on Evolutionary Computation*, 5(4):349–358.

[129] Otero, D. F. and Akhavan-Tabatabaei, R. (2015). A stochastic dynamic pricing model for the multiclass problems in the airline industry. *European Journal of Operational Research*, 242(1):188–200.

[130] O'Neil, M. and Ryan, C. (2003). Grammatical evolution. In *Grammatical evolution*, pages 33–47. Springer.

[131] P. Belobaba, A. O. and Barnhart, C. (2016). The global airline industry. In *Soft computing and industry*, page 82. 2nd ed. John Wiley Sons, Ltd.

[132] Padillo, F., Luna, J. M., and Ventura, S. (2020a). LAC: library for associative classification. *Knowl. Based Syst.*, 193:105432.

[133] Padillo, F., Luna, J. M., and Ventura, S. (2020b). Lac: Library for associative classification. *Knowledge-Based Systems*, 193:105432.

[134] Pels, E. and Rietveld, P. (2004). Airline pricing behaviour in the london–paris market. *Journal of Air Transport Management*, 10(4):277–281.

[135] Pérez, J. M., Muguerza, J., Arbelaitz, O., Gurrutxaga, I., and Martín, J. I. (2007). Combining multiple class distribution modified subsamples in a single tree. *Pattern Recognition Letters*, 28(4):414–422.

[136] Pitfield, D. (2005). A time series analysis of the pricing behaviour of directly competitive'low-cost'airlines. *International Journal of Transport Economics/Rivista internazionale di economia dei trasporti*, pages 15–39.

[137] Quinlan, J. R. and Cameron-Jones, R. M. (1995). Induction of logic programs: Foil and related systems. *New Generation Computing*, 13(3):287–312.

[138] Rai, D., Thoke, A., and Verma, K. (2012). Enhancement of associative rule based foil and prm algorithms. In *2012 Students Conference on Engineering and Systems*, pages 1–4. IEEE.

[139] Rajab, K. D. (2019). New associative classification method based on rule pruning for classification of datasets. *IEEE Access*, 7:157783–157795.

[140] Raju, J. and Zhang, Z. (2010). *Smart Pricing: How Google, Priceline, and Leading Businesses Use Pricing Innovation for Profitabilit (paperback)*. Pearson Prentice Hall.

[141] Ramraj, S., Uzir, N., Sunil, R., and Banerjee, S. (2016). Experimenting xgboost algorithm for prediction and classification of different datasets. *International Journal of Control Theory and Applications*, 9:651–662.

[142] Rish, I. et al. (2001). An empirical study of the naive bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, volume 3, pages 41–46.

[143] Rojas, R. (1996). The backpropagation algorithm. In *Neural networks*, pages 149–182. Springer.

[144] Rokach, L. and Maimon, O. (2005). Decision trees. In *Data mining and knowledge discovery handbook*, pages 165–192. Springer.

[145] Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215.

[146] Rusell, S. and Norvig, P. (2003). Artificial intelligence: A modern approach. *Pretice Hall Series in Artificial Intelligence*, 1:649–789.

[147] Ryan, C., Collins, J. J., and Neill, M. O. (1998). Grammatical evolution: Evolving programs for an arbitrary language. In *European Conference on Genetic Programming*, pages 83–96. Springer.

[148] Salzberg, S. L. (1994). C4. 5: Programs for machine learning by j. ross quinlan. morgan kaufmann publishers, inc., 1993. *Machine learning*.

[149] Santos, F. A. d. N., Mayer, V. F., and Marques, O. R. B. (2020). Dynamic pricing and price fairness perceptions: a study of the use of the uber app in travels. *Turismo: Visão e Ação*, 21:239–264.

[150] Selcuk, A. M. and Avşar, Z. M. (2019). Dynamic pricing in airline revenue management. *Journal of mathematical analysis and applications*, 478(2):1191–1217.

[151] Sengpoh, L. (2015). The competitive pricing behaviour of low cost airlines in the perspective of sun tzu art of war. *Procedia-Social and Behavioral Sciences*, 172:741–748.

[152] Shiri, J., Sadraddini, A. A., Nazemi, A. H., Kisi, O., Landeras, G., Fard, A. F., and Marti, P. (2014). Generalizability of gene expression programming-based approaches for estimating daily reference evapotranspiration in coastal stations of iran. *Journal of hydrology*, 508:1–11.

[153] Shukla, N., Kolbeinsson, A., Otwell, K., Marla, L., and Yellepeddi, K. (2019). Dynamic pricing for airline ancillaries with customer context. In *Proceedings of the 25th ACM SIGKDD International Conference on knowledge discovery & data mining*, pages 2174–2182.

[154] Simovici, D. A. (2012). *Linear algebra tools for data mining*. World Scientific.

[155] Siu, K. K., Butler, S. M., Beveridge, T., Gillam, J., Hall, C., Kaye, A. H., Lewis, R. A., Mannan, K., McLoughlin, G., Pearson, S., et al. (2005). Identifying markers of pathology in saxs data of malignant tissues of the brain. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 548(1-2):140–146.

[156] Song, H. S., kyeong Kim, J., and Kim, S. H. (2001). Mining the change of customer behavior in an internet shopping mall. *Expert systems with applications*, 21(3):157–168.

[157] Spears, W. M., De Jong, K. A., Bäck, T., Fogel, D. B., and De Garis, H. (1993). An overview of evolutionary computation. In *European Conference on Machine Learning*, pages 442–459. Springer.

[158] Sumner, M., Frank, E., and Hall, M. (2005). Speeding up logistic model tree induction. In *European conference on principles of data mining and knowledge discovery*, pages 675–683. Springer.

[159] Swinburne, R. (2004). Bayes' theorem. *Revue Philosophique de la France Et de l*, 194(2).

[160] Taunk, K., De, S., Verma, S., and Swetapadma, A. (2019). A brief review of nearest neighbor algorithm for learning and classification. In *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*, pages 1255–1260. IEEE.

[161] Turing, A. M. (2009). Computing machinery and intelligence. In *Parsing the turing test*, pages 23–65. Springer.

[162] Vamathevan, J., Clark, D., Czodrowski, P., Dunham, I., Ferran, E., Lee, G., Li, B., Madabhushi, A., Shah, P., Spitzer, M., et al. (2019). Applications of machine learning in drug discovery and development. *Nature Reviews Drug Discovery*, 18(6):463–477.

[163] Ventura, S., Luna, J. M., et al. (2018). *Supervised descriptive pattern mining*. Springer.

[164] Wang, K., Tang, L., Han, J., and Liu, J. (2002). Top down fp-growth for association rule mining. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 334–340. Springer.

[165] Webb, G. I. (1999). Decision tree grafting from the all-tests-but-one partition. In *Ijcai*, volume 2, pages 702–707.

[166] Webb, G. I. (2010). Association discovery. In *ACM SIGKDD Workshop on Useful Patterns*, page 7.

[167] Weiss, S. M. and Indurkhya, N. (1998). *Predictive data mining: a practical guide*. Morgan Kaufmann.

[168] West, D. M. (2018). *The future of work: Robots, AI, and automation*. Brookings Institution Press.

[169] Whigham, P. A. (1996). Search bias, language bias, and genetic programming. *Genetic Programming*, 1996:230–237.

[170] Witten, I. H., Frank, E., and Hall, M. A. (2011). *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann, Amsterdam, 3 edition.

[171] Wittman, M. D. and Belobaba, P. P. (2018). Customized dynamic pricing of airline fare products. *Journal of Revenue and Pricing Management*, 17(2):78–90.

[172] Wohlfarth, T., Clémençon, S., Roueff, F., and Casellato, X. (2011). A data-mining approach to travel price forecasting. In *2011 10th International Conference on Machine Learning and Applications and Workshops*, volume 1, pages 84–89. IEEE.

[173] Wong, T.-T. and Tseng, K.-L. (2005). Mining negative contrast sets from data with discrete attributes. *Expert Systems with Applications*, 29(2):401–407.

[174] Wrobel, S. (1997). An algorithm for multi-relational discovery of subgroups. In *European symposium on principles of data mining and knowledge discovery*, pages 78–87. Springer.

[175] Wu, X., Kumar, V., Quinlan, J. R., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G. J., Ng, A., Liu, B., Philip, S. Y., et al. (2008). Top 10 algorithms in data mining. *Knowledge and information systems*, 14(1):1–37.

[176] Wu, Z. and Yao, M. (2009). A new gep algorithm based on multi-phenotype chromosomes. In *2009 Second International Workshop on Computer Science and Engineering*, volume 1, pages 204–209. IEEE.

[177] Xiao, R., Li, M.-J., and Zhang, H.-J. (2004). Robust multipose face detection in images. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(1):31–41.

[178] Yadav, J. and Sharma, M. (2013). A review of k-mean algorithm. *Int. J. Eng. Trends Technol*, 4(7):2972–2976.

[179] Yasodha P, A. N. (2014). Comparative study of diabetic patient data using classification algorithm in weka tool. *Int. J. Comput. Appl. Technol. Res.*, pages 554–558.

[180] Yin, X. and Han, J. (2003). Cpar: Classification based on predictive association rules. In *Proceedings of the 2003 SIAM international conference on data mining*, pages 331–335. SIAM.

[181] Yu, C.-S., Lin, Y.-J., Lin, C.-H., Wang, S.-T., Lin, S.-Y., Lin, S. H., Wu, J. L., and Chang, S.-S. (2020). Predicting metabolic syndrome with machine learning models using a decision tree algorithm: Retrospective cohort study. *JMIR Med Inform*, 8(3):e17110.

[182] Zhong, J., Feng, L., and Ong, Y.-S. (2017a). Gene expression programming: A survey. *IEEE Computational Intelligence Magazine*, 12(3):54–72.

[183] Zhong, J., Feng, L., and Ong, Y.-S. (2017b). Gene expression programming: A survey. *IEEE Computational Intelligence Magazine*, 12(3):54–72.

[184] Zhou, C., Xiao, W., Tirpak, T. M., and Nelson, P. C. (2003). Evolving accurate and compact classification rules with gene expression programming. *IEEE Transactions on Evolutionary Computation*, 7(6):519–531.

[185] Zhu, J., Liapis, A., Risi, S., Bidarra, R., and Youngblood, G. M. (2018). Explainable ai for designers: A human-centered perspective on mixed-initiative co-creation. In *2018 IEEE Conference on Computational Intelligence and Games (CIG)*, pages 1–8. IEEE.

[186] Zuo, J., Tang, C., and Zhang, T. (2002). Mining predicate association rule by gene expression programming. In *International Conference on Web-Age Information Management*, pages 92–103. Springer.

Apéndice A

Datos utilizados en la metodología GEP-FL

Los datos que se utilizaron para la propuesta del Capítulo 3 de esta memoria de tesis provienen de *Sabre Global Distribution System* también conocido simplemente como *Sabre*, la cual es una empresa líder en software y tecnología en la industria aérea a nivel mundial. *Sabre* nació de una iniciativa conjunta entre American Airlines e IBM para crear el primer sistema computarizado de reservas de aerolíneas del mundo en la década de los años 60. Desde entonces, este software ha evolucionado hasta convertirse en un ecosistema tecnológico que toca casi todas las etapas de la experiencia de un pasajero. La matriz de este sistema está organizada en tres unidades de negocio:

- *Sabre Travel Network*: sistema de distribución global.
- *Sabre Airline Solutions*: tecnología para aerolíneas.
- *Sabre Hospitality Solutions*: soluciones tecnológicas para hoteles.

Un dato importante que cabe mencionar, es que *Sabre* es una de las tecnologías más duraderas de la industria, la cual ha estado presente desde el nacimiento de la automatización de las aerolíneas hasta siendo nombrada en el puesto número 7 en la lista del "mejor software jamás escrito"de InformationWeek. *Sabre* se puede considerar como un *legacy software* dentro de la industria aérea.

Los equipos de analistas de precios que tiene acceso a *Sabre*, lo utilizan cotidianamente de forma manual para realizar consultas de tarifas que reflejen el precio histórico o actual de una ruta. Un ejemplo de la información que se obtiene a través de una consulta en *Sabre*, es el que se muestra en la Figura A.1. A continuación se describe la información que los

```
Línea 1  | FQYULYOW05JAN20-AC«
Línea 2  | YMQ-YOW       CXR-AC       SUN 05JAN20                    CAD
Línea 3  | THE FOLLOWING CARRIERS ALSO PUBLISH FARES YMQ-YOW:
Línea 4  | 2R 5T 7F 9B AF JV PB PD TS UA
Línea 5  | //SEE FQHELP FOR INFORMATION ABOUT THE NEW FARE DISPLAYS//
Línea 6  | ALL FEES/TAXES/SVC CHARGES INCLUDED WHEN ITINERARY PRICED
Línea 8  | SURCHARGE FOR PAPER TICKET MAY BE ADDED WHEN ITIN PRICED
Línea 9  | AC     YMQYOW          05JAN20
Línea 10 |     V FARE BASIS     BK    FARE   TRAVEL-TICKET AP  MINMAX  RTG
Línea 11 |   1   LZ6PZATG       L X   477.00      ----      30/1  -/  -    5
Línea 12 |   2   TZ2PZATG       T X   492.00      ----      21/1  -/  -    5
Línea 13 |   3   SZ8PZATG       S X   505.00      ----      18/1  -/  -    5
Línea 14 |   4   LZ6PZAFL       L X   507.00      ----      30/1  -/  -    5
Línea 15 |   5   TZ2PZAFL       T X   522.00      ----      21/1  -/  -    5
Línea 16 |   6   WZ4PZATG       W X   530.00      ----      14/1  -/  -    5
Línea 17 |   7   SZ8PZAFL       S X   535.00      ----      18/1  -/  -    5
Línea 18 |   8   LZ6PZACO       L X   547.00      ----      30/1  -/  -    5
Línea 19 |   9   TZ2PZACO       T X   562.00      ----      21/1  -/  -    5
Línea 20 |  10   VZDPZATG       V X   569.00      ----      10/1  -/  -    5
Línea 21 |  11   WZ4PZAFL       W X   570.00      ----      14/1  -/  -    5‡
```

Figura A.1 Ejemplo de tarifas extraídas en archivo de texto.

analistas de precios obtienen al realizar una consulta como la que se muestra en el ejemplo propuesto:

- La Línea 1 muestra lo que se conoce como *Fare Quote* que es básicamente una consulta para obtener las tarifas que ha publicado la aerolínea *Air Canada*. Como se puede observar después de los caracteres FQ se indican el origen y el destino del mercado o ruta de la cual se quieren obtener las tarifas disponibles, utilizando lo que se conoce como los códigos de aeropuerto, que en este caso es YULYOW (una ruta de la ciudad de Montreal a Ottawa). Después de indicar la ruta, se agrega la fecha en que el vuelo tendrá efecto y así obtener todas las tarifas que están disponibles para esa fecha. En este ejemplo, es un vuelo para el día 5 de enero de 2020. La Línea 1 termina con los caracteres AC que es el código para representar a la aerolínea *Air Canada*.

- La línea 2 muestra el resultado del FQ donde se puede observar la ruta, la aerolínea que ha publicado una tarifa en esta ruta y el tipo de cambio. En esta caso, la ruta es YMQ-YOW (De Montreal a Ottawa),la aerolínea que oferta las tarifas es Air Canada (AC), y el tipo de cambio es en dólares canadienses (CAD).

- En las líneas 3 y 4 se nombran otras aerolíneas que también publican tarifas en esta ruta, en este ejemplo se puede observar que existen otros transportadores aéreos o de otro tipo como son algún servicio de tren que pueden publicar tarifas en esta ruta como son: Aero Condor (2R), Germania (ST), Accesrail 9B, Air France (AF), Bearskin (JV), Provincial (PB), Porter Airlines (PD).Air Transat (TS) and United Airlines (UA)también publican tarifas en esta ruta. Esta información no es de mucho

uso, ya que aerolíneas como AF o UA son aerolíneas fuera de Canadá que solo puede ofertas estas tarifas a través de acuerdos comerciales con la aerolínea Air Canada, por lo que estas dos líneas no son de mucho interés para un analista de precios desde el aspecto competitivo de análisis de tarifas.

- De la línea 5 a la 8, solo se muestra información para utilizar la ayuda del sistema y obtener más información acerca de otros factores que forman parte de los precios de las tarifas como son los impuestos y otro tipo de cargos.
- La línea 9 muestra el operador que ofrece las tarifas, la ruta y la fecha en que aplica estas tarifas.
- De la línea 10 hasta el final se muestran las tarifas que han sido expuestas al público, por lo que se puede observar los códigos de la tarifa base, la clase de cada tarifa, el monto de la tarifa, los días de compra anticipada para que la tarifa aplique y el código de ruta.

Es importante mencionar que para que un analista de precios pueda hacer bien su trabajo tiene que conocer a detalles los *fare basis codes* (códigos de las tarifas base) de sus competidores para poder interpretar y extraer patrones de los cambios que se efectúan constantemente en el mercado. Un *fare basis code* es utilizado por las aerolíneas para identificar un tipo de tarifa y permitir que el personal de ventas o los agentes de viajes encuentren las reglas aplicables a una tarifa. Cada aerolínea establece sus propios fare basis codes , por lo que, se puede crear cualquier cantidad de reservas o clases de tarifas, a las cuales se pueden aplicar diferentes precios y condiciones de reserva. Describir las clases de tarifas es complicado porque varían de una aerolínea a otra.

Generalemente, los *fare basis codes* comienzan con una letra, la cual es la clase de la tarifa, lo que comúnmente se conoce como la *fare class* y la cual siempre es representada por una de las 26 letras del alfabeto en inglés, a este primer caracter le siguen otras letras o caracteres numéricos. Un *fare basis code* se compone de 2 a 8 caracteres. En la actualidad, se puede observar que diversas aerolíneas están adoptando un estándar de 8 caracteres, pero esto varía en base a la estructura y necesidades de cada operador aéreo. El primer carácter del *fare basis code* es siempre una letra y casi siempre coincidirá con la *fare class*. Estas *fare classes* son los identificadores utilizados por el departamento de gestión de ingresos de una aerolínea para controlar cuántos asientos se pueden vender en un nivel de tarifa en particular. Por ejemplo, un avión puede tener 30 asientos disponibles para la cabina de Clase Económica, suponiendo que solo se existen 3 clases A, B y C, el equipo de gestión de ingresos puede

asignar 5 asientos para la clase A, 10 para la clase B y 15 para la clase C, como una forma de optimización para maximizar las ganancias de una aerolínea.

A las *fare classes* también se le conoce como códigos de reserva, los cuales en principio fueron definidos por IATA. Hoy en día, las aerolíneas se han desviado de este estándar y los códigos de reserva actuales son específicos de cada aerolínea, los cuales muchas veces no tienen patrones definidos. Como en el caso de la metodología propuesta en el capítulo 3 de esta memoria de tesis, en el cual se analizan tarifas de la aerolínea Air Canada, la cual utiliza las letras *fare classes* **K, A, L, T, S**; siendo la de más bajo precio la clase **K** y la más cara clase **S**. Por lo tanto, el mismo *faris basis code* puede tener diferente significado para otra aerolínea. Muchos operadores aéreos utilizan la mayoría de las letras del alfabeto para gestionar el ingreso de sus tarifas. Sin embargo, algunos códigos de reserva han conservado el mismo significado en la mayoría de las aerolíneas, como son los que se muestran en la Tabla A.1, en la cual se pueden observar códigos de reserva comunes como son F - Primera Clase, J - Clase Ejecutiva, W - Económica Premium, Y - Económica. La Tabla A.2 muestra algunos ejemplos de otras letras y números en posiciones alternas del *fare basis code*.

En la Tabla A.1 se muestran las cuatro cabinas que existen dentro de la industria de las aerolíneas. No todos los operadores aéreos ofrecen las cuatro, algunos de ellos de hecho solo ofrecen la Clase Económica; por lo que, en un asiento de clase económica en una aerolínea puede ser muy diferente de otro asiento económico en otra aerolínea. Otro factor importante a considerar es que muchas aerolíneas tienen los mismos modelos de aviones en sus flotas, pero configuran los interiores de manera diferente, por lo que de acuerdo a la ubicación del

Código de Reserva	Descripción
F	Tarifa completa Primera Clase.
J	Tarifa completa Clase Ejecutiva
W	Tarifa completa Clase Económica Premium
Y	Tarifa completa Clase Económica

Tabla A.1 Ejemplo de códigos de reserva comunes

Código de Reserva	Posición Estándar	Descripción
E	Segunda posición	Indicador de tarifas de excursión
Números	Posiciones alternas	Indicadores comodines
H o L	Posiciones alternas	Indicador de temporadas
OW	Posiciones alternas	*One Way*
RT	Posiciones alternas	*Round Trip*

Tabla A.2 Ejemplo de otros caracteres del código de reserva

asiento puede variar el precio y el servicio del mismo. A continuación se describe lo que un pasajero puede esperar de cada una de estas cabinas:

- **Clase Económica**: es la clase más básica, los asientos son los más estrechos, van desde alrededor de 41 centímetros hasta poco más de 48 centímetros de ancho y la distancia entre asientos, varía entre 76 y 86 centímetros. En la actualidad, la Clase Económica ofrece poco más que un asiento que llevará a un pasajero del punto A al punto B.

- **Clase Económica Premium**: Esta clase es ligeramente más cómoda que la Clase Económica, ofrece asientos más anchos y más espaciosos para las piernas a un precio más bajo que la Clase Ejecutiva o la Primera Clase en la mayoría de las aerolíneas. Algunas aerolíneas ponen la Clase Económica y la Clase Económica Premium en la cabina principal, ubicando los asientos Premium en la parte delantera de la cabina. La Clase Económica Premium no solo puede estar físicamente separada de la Clase Económica, sino que también puede ofrecer beneficios adicionales como comida de cortesía, equipaje adicional y kits de amenidades.

- **Clase Ejecutiva**: La Clase Ejecutiva es una clase completamente diferente de la Clase Económica, el servcio abarca desde un asiento más ancho completamente reclinable, servicio completo de comidas, entretenimiento a bordo y un kit de amenidades, minibar personal, multicomidas servidas en porcelana fina y un área de bar completa con barman y canapés.

- **Primera Clase**: La Primera Clase es una experiencia de lujo que varía según la aerolínea. Este servicio puede variar desde asiento reclinable de 6 pies y 8 pulgadas con firmeza ajustable y función de masaje, una puerta para privacidad, entretenimiento a bordo y servicio completo de comidas hasta una suite de 3 habitaciones con sala de estar, dormitorio con cama doble y baño privado con ducha.

La Tabla A.3 muestra algunos de los caracteres principales que utiliza la aerolinea Air Canada para estructurar sus clases de tarifas. Los códigos de reserva son a su vez indicadores de las restricciones que existen para que una tarifa se pueda aplicar, como es el caso de la temporada o por su término en inglés *seasonality*. Un ejemplo son las letras **L** y **H** que son utilizadas por Air Canada y muchas otras aerolíneas en el mundo para identificar la temporada alta y baja respectivamente. Estas temporadas varían en fechas de acuerdo a la demanda del mercado, ya que varían de acuerdo al comportamiento que pueda mostrar en una ciudad, una región, estado o país. Por ejemplo, en Canadá la temporada baja (L) puede ser del 16 de enero al 10 de febrero, del 20 de febrero al 20 de marzo y así durante diferentes

Caracter	Descripción
J, C, D	Clase Ejecutiva
Z, P	Clase Ejecutiva
O	Clase Económica Premium
K, A, L...	Clase Económica

Tabla A.3 Ejemplo de códigos de clases de tarifas

etapas del año. Por lo general, la temporada baja es sinónimo de baja demanda y precios más bajos. Un ejemplo de la temporada alta (H) puede ser del 15 de diciembre al 15 de enero, del 11 de febrero al 19 de febrero y así sucesivamente. Esta temporada alta es sinónimo de alta demanda y precios más altos. Existen otras letras que se utilizan específicamente por cada aerolínea, como es el ejemplo de Air Canada que utiliza la letra **A** lo cual significa que esta letra puede aplicar en fechas muy específicas por diversas razones, como puede ser un evento que surja en una ciudad en especial y se prevé una demanda más alta de normal, por lo tanto, se utiliza esa letra para identificar esas fechas y agregar precios más altos, ya que el WTP de los pasajeros se incrementará durante esas fechas.

Para concluir, los equipos conformados por analistas de precios deben de poder tener la capacidad de decodificar los códigos de reserva de sus competidores, sin esta información o este conocimiento no les sería posible interpretar las acciones que suceden día a día en el mercado. El análisis de servicios, productos y restricciones que aplican a cada tarifa es una fundamental para el éxito de una aerolínea.

Apéndice B

Datos utilizados en el enfoque GE-FS

Los datos utilizados para la propuesta del Capítulo 4 de esta memoria de tesis provienen de la plataforma *Adobe Analytics* la cual proporciona a las empresas una amplia gama de información sobre su sitio web y campañas de marketing, incluidos los siguientes:

- **Fuentes de tráfico**: información sobre cómo los visitantes encuentran el sitio web, incluidos los motores de búsqueda, las redes sociales y los sitios de referencia.
- **Datos de comportamiento**: detalles sobre qué páginas ven los visitantes, cuánto tiempo permanecen en el sitio y qué acciones realizan, como puede ser realizar una compra o completar un formulario.
- **Datos de comercio electrónico**: métricas sobre ingresos, tasas de conversión y rendimiento del producto.
- **Seguimiento de campañas**: datos sobre el rendimiento de las diferentes campañas de marketing y cuáles generan la mayor cantidad de conversiones.
- **Segmentación**: la capacidad de segmentar los datos por diferentes criterios, como la ubicación geográfica o la fuente de referencia.
- **Informes y visualización**: varios informes y visualizaciones, como paneles en tiempo real e informes personalizables, para ayudar a las empresas a dar sentido a los datos e identificar tendencias y patrones.

A continuación se mencionan algunos ejemplos de como las aerolíneas están utilizando los datos que proporciona *Adobe Analytics*:

- **Cálculo del número de páginas visitadas**: qué páginas del sitio web reciben más tráfico y el tiempo que pasan los visitantes en cada página, como pueden ser las páginas para reservar los vuelos o las páginas donde se promocionan ciertos destinos u otro tipo de ofertas.
- **Calcular la tasa de rebote**: el porcentaje de visitantes que abandonan el sitio después de ver solo una página.
- **Calcular la tasa de salida**: el porcentaje de visitantes que abandonan el sitio desde una página que muestra una ruta en específico.
- **Calcular de la tasa de clics**: el porcentaje de visitantes que hacen clic en un enlace o botón específico.
- **Calcular la tasa de conversión**: el porcentaje de visitantes que completan una reserva al mirar cierta o ciertas rutas.
- **Análisis de rastreo**: la capacidad de rastrear a los visitantes a medida que se mueven a través del sitio web e identificar dónde lo están dejando.
- **Análisis de ingreso al sitio web**: la capacidad de identificar si el visitante proviene un buscador como *Google, Google Fligths, Sky Scanner*, entre otros.

Una importante característica de los datos que se obtienen de *Adobe Analytics* es que se pueden utilizar para identificar áreas del sitio web que no funcionan bien y realizar mejoras para aumentar las tasas de participación y conversión. Además, estos datos se pueden segmentar según diferentes criterios, como la ubicación geográfica, la fuente de referencia o el tipo de dispositivo, para proporcionar una comprensión más detallada del comportamiento de los visitantes. Sin embargo, se requiere de la implantación de técnicas de análisis avanzadas como pueden ser la DM para poder lograr este grado de segmentación o identificación de patrones.

Aunque *Adobe Analytics* es una plataforma de análisis web que brinda a las empresas una gran cantidad de datos e información, también existen algunas desventajas potenciales que se deben considerar:

- **El Costo**: *Adobe Analytics* es un servicio de pago, por lo que, su estructura de precios puede ser compleja, lo que dificulta el presupuesto para algunas aerolíneas, sobre todo para aquellas en el que su modelo de negocios son bajo costo (*Low Cost*, LC) o ultra bajo costo (*Ultra Low Cost*, ULCC).

- **Complejidad**: *Adobe Analytics* es una plataforma amplia en funciones, por lo que, puede ser difícil para algunos usuarios navegar y comprender todas sus funciones y capacidades. Esta plataforma requiere de cierta experiencia técnica para configurarla y usarla de manera efectiva, especialmente para funciones más avanzadas.

- **La calidad de los datos**: los datos recopilados por *Adobe Analytics* son recopilados por esta herramienta, son a través de códigos difíciles de comprender por un humano. De manera que, se requiere de un índice de códigos para su interpretación. Esto dificulta la creación de reportes para diferentes audiencias, pues estos datos en principio se deben humanizar al transformar estos códigos en atributos legibles para los humanos. Debido a esto, se corre el riesgode generar datos inexactos o incompletos.

- **La privacidad de datos**: *Adobe Analytics* puede recopilar datos confidenciales sobre los visitantes, como su historial de navegación e información personal, lo que puede generar inquietudes sobre la privacidad de los datos que las aerolíneas están recopilando.

- **Integración:** como la mayoría de las plataformas propietarias *Adobe Analytics* está diseñada para integrarse con otros productos de *Adobe*, tales como puede ser *Adobe Experience Cloud*, por lo que, es posible que no se integre bien con otras herramientas de marketing creadas por terceros.

Como se ha mencionado anteriormente los atributos de los datos que recolecta *Adobe Analytics* los nombres aparecen en códigos a los cuales se les denomina eVar. A continuación se describen los elementos que conforman un eVar:

- **Elemento**: El elemento es el nombre descriptivo de la variable de conversión. Este nombre es la forma en que se hace referencia a la eVar en los informes generales.

- **Tipo de eVar**: existen dos tipos de eVar. Por un lado, las cadenas de texto; las cuales son las capturas de texto utilizado en el sitio web, estas son el tipo más común de eVar. Un ejemplo de estas variables puede ser si se están rastreando cosas como campañas internas o palabras clave de búsqueda interna, se configura la herramienta para la captura de texto. Por otro lado, las variables tipo contador, que como su nombre lo indica, cuentan el número de veces que ocurre una acción antes del evento de éxito, como puede ser la ejecución de un reserva de un vuelo en un sitio web de alguna aerolínea.

- **Asignación**: Determina cómo *Adobe Analytics* asigna el crédito por un evento de éxito si una variable recibe varios valores antes del evento. Los valores admitidos incluyen los siguientes:

    1. El valor más reciente: el cual es el último valor de eVar y el que recibe el crédito por eventos exitosos.
    2. Valor original: la primera eVar que recibe crédito por eventos exitosos.
    3. Lineal: asigna eventos de éxito por igual en todos los valores de eVar.

- **Tiempo de expiración:** Especifica un período de tiempo, o evento, después del cual caduca el valor de eVar (ya no recibe crédito por eventos exitosos). Si se produce un evento de éxito después de la expiración de la eVar, el valor Ninguno recibe crédito por el evento (ninguna eVar estaba activa).

- **El estado del eVar:** El estado del eVar se define en tres estados diferentes:

    1. Desactivado: Desactiva e elimina la eVar de la lista de variables de conversión.
    2. Sin subrelaciones: le impide desglosar la eVar por una dimensión.
    3. Subrelaciones básicas: le permite desglosar una eVar por cualquier dimensión completa (por ejemplo, servicios, productos o campañas).

- **Reinicio:** Restablece cualquier valor existente en la eVar.

En el sitio web de *Adobe Analytics* también se menciona que ofrece servicios de análisis avanzados, incluyendo técnicas de optimización predictivas basadas en aprendizaje automático. Sin embargo, esta plataforma no ofrece servicios de DM como pueden ser la extracción de reglas o patrones, por lo que, para integrar estos datos a un proceso de descuentos dinámicos, se tuvo que realizar tareas de ingeniería de datos fuera del enfoque de la metodología propuesta en el Capítulo 4 y formar una base de datos en la cual se humanizor los eVar a nombres que pudieran ser legibles de manera natural por un humano. Esta base de datos se conformo de un total de 49 atributos, de los cuales dentro de la metodología propuesta sólo se consideran 16.

Para describir brevemente como se obtiene esta base de datos, el conjunto de datos original se obtiene de las búsquedas que se realizan en el sitio Web, el cual está conformado por un total de 180 eVar, cada búsqueda por visitante genera en promedio 16 líneas, las cuales a través del identificador que genera cada visitante al sitio web se reducen a una sola línea para resumir toda la actividad que realiza mientras navega en el sitio web. De los 180 eVar

originales se reducen a un total de 49, que son los que proveen diferente información acerca de las búsquedas. Esta reducción significativa en el número de eVar se debe a que algunos de estos repiten cierta información o no tiene relevancia. De estos 49 atributos en la base de datos se utilizaron solo 16 para la metodología propuesta en el Capítulo 4 de esta memoria de tesis, y los cuales se describieron ampliamente.

Para concluir se debe mencionar que el proceso de conversión de las eVar a un base de datos estructurada, es proceso lento debido al elevado monto de información cruda que genera esta herramienta y a los pasos que se deben de seguir para poder elaborar esta base datos y de la cual se pueda extraer conocimiento.