UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN

"ESTIMACIONES DE ERROR Y GRADO DE TOLERANCIA PARA DISTINTAS TÉCNICAS DE MINERÍA DE DATOS"

T E S I S

QUE PARA OBTENER EL GRADO DE:

MAESTRO EN INGENIERÍA (COMPUTACIÓN)

P R E S E N T A:

RENÉ ALEJANDRO VILLEDA RUZ

DIRECTOR DE TESIS: M. en C. Javier García García

México, D.F. **2007.**

Agradecimientos

Nuevamente me encuentro terminando una etapa académica, misma que representa un paso más en este camino interminable en la adquisición de conocimiento dentro de nuestra Universidad. Sin embargo, otra vez este trabajo resulta ser el producto de un esfuerzo que ha estado respaldado por muchas y muy importantes personas quienes directa o indirectamente han contribuido en distinto grado y por lo tanto merecen una parte del reconocimiento en la obtención de esta nueva meta. Por lo anterior, quiero hacer presente mi agradecimiento...

A mi padre **Enrique Villeda Navarro**, por el apoyo incondicional que me ha brindado hasta esta etapa de mi vida. Sus palabras, consejos y ejemplos han sido una guía fundamental a lo largo de estos años. Sus atenciones constantes me han permitido enfocarme y superarme hasta las actuales instancias académicas y eso no lo puedo agradecer con medio alguno. También agradezco el soporte de mi hermano **Enrique Villeda Ruz** ya que aún con los altibajos que siempre tienen las relaciones entre hermanos, afortunadamente puedo decir que la mayoría son etapas buenas. A mi abuelo **Aquilino Villeda Andablo** dado que sin su visión difícilmente me hallaría en estas instancias, sus sabios consejos brindados desde la infancia han hecho gran parte de la persona que soy. Sin ellos y su paciencia para soportar mis "encierros" durante días y noches trabajando en proyectos, trabajos, prácticas y tareas (y el mal humor derivado de ellos) no me encontraría aquí. Los quiero mucho.

A la **UNAM**, por brindarme la oportunidad de pertenecer a ella desde hace más de 16 años cuando inicie el camino desde Iniciación Universitaria en la **ENP 2**, ser una institución extraordinaria en todos los sentidos pero sobretodo por brindarme los elementos necesarios para ser una mejor persona y el permitirme transmitir un poco de conocimiento dentro de sus recintos.

A mi familia paterna, particularmente a mi primo René Villeda Cruz. Afortunadamente hemos vivido muchas experiencias como primos, pero nos hemos acercado como hermanos. Gracias por tus "enseñanzas" ☺. A mis primas Elideth y Dánae Villeda

Contreras, Rosario Villeda Cruz, mis primos Christian y Salvador Mercado Villeda porque siempre han estado presente de una u otra forma.

A mis amigas y amigos, pero de manera muy, muy especial a Verónica Cruces Ramírez. Ella, siendo una persona excepcional en **todos** los sentidos, es quien realmente aguantó y soportó incondicionalmente todas mis quejas, molestias y problemas que atravesé durante este periodo tan difícil de mi vida. Sus palabras de apoyo, y en ocasiones de reclamo, han hecho de mi una mejor persona en todos los aspectos. Le agradezco enormemente haber compartido más detalles y experiencias de las que hubiese imaginado (y merecido) así como también demostrarme que hay cosas más importantes que el aspecto académico y material. Sin su apoyo el andar por este camino hubiese sido más complejo de lo que fue. Gracias, siempre serás importante.

A Pedro Abundes Jiménez, un gran amigo en todos los sentidos. Afortunadamente aún cuando hemos tomado caminos distintos, la amistad siempre estará presente; a Karla Jiménez Álvarez (Karliux) porque siempre se ha encontrado cerca y sus palabras llegan en el momento justo; a Elia A. Calderón Ríos, Ilitia A. Sauer Vera, Selene Calva Estrada, Carolina B. Chávez Cortés, Guadalupe Alvarado Arias, Lorena Montes de Oca Muñoz y Belén Reyes Alcaraz ya que cada una de ellas es una excelente persona y gracias a su impulso ha sido más fácil soportar tantas "desveladas simultáneas" durante la realización de este trabajo.

A Liliana M. González Suárez ("Liz") porque no tengo palabras para agradecer los comentarios y platicas que constantemente me han ayudado en momentos difíciles (aunque continúe esperando esa "achicalada"... ¡otros 2 años ☹!); a Julieta A. López Melendez ("Gina" ☺), simplemente porque es una niña genial y aún cuando nos hemos perdido la pista durante estos años, sé que siempre tendré un soporte en tus palabras y consejos. Tu dedicación y entereza siempre serán admirables. A Eudave (ok, ok... M. Lorena Eudave Loera ☺), todas esas noches con la mensajería instantánea no han hecho sino darme cuenta lo mucho que vales como persona, y ¡claro! pasar "conversaciones" tan divertidas como didácticas. A Minerva L. Luna Nava (Mins ☾), dado que tu empuje y coraje en todas las cosas que haces han servido de motivación para continuar a pesar de los problemas. Admiro tu tesón y fortaleza para afrontar cada día. A Karla Ramírez Pulido (Karlita), gracias a sus palabras, recomendaciones y sugerencias he podido afrontar con una perspectiva totalmente distinta todas las dificultades durante esta etapa. A Evelyn Gabriela Sánchez Olguin, porque es una persona extraordinaria y sensacional en todos los aspectos. Mostrarme tantas cosas en tan poco tiempo y ser ese pilar que mantenía mis esperanzas en momentos muy difíciles. Sin su presencia, el tiempo previo y durante la realización de la tesis no hubiera sido ni infinitamente lo genial y preciado que se manifestó. De corazón, gracias.

A mis compañeros y amigos de la maestría, en especial a Virginia Teodosio Procopio, simplemente por ser formidable y una persona que defiende sus ideales ante todo. Esas estancias en los jardines frente al IIMAS carecerían de sentido si no hubieses estado ahí. Eres una mujer muy creativa, inteligente, divertida y ¡ruda!; a Adriana Ramírez Vigueras porque desde siempre ha sido un apoyo muy importante, eres una estupenda

persona. Asistir a todas esas conferencias y mini-cursos forzados habría sido más difícil sin tu presencia; a Cano Gildardo Bautista García porque al final de cuentas terminamos conociéndonos más en seis meses de la maestría que en nueve semestres de la licenciatura y ahora se que eres una excelente persona y muy valiosa; a Valentina Muñoz Porras porque siempre te mostraste tal cual y aprendí que las personas brillantes pueden estar a tu lado y pasar desapercibidas hasta que te des oportunidad de conocerlas ¡Eres una de ellas! Ustedes formaron ese grupo especial de primer semestre que me permitió solventar y hacer mas "ligero" el golpe de tan ardua época. Las estancias en la biblioteca del IIMAS y desveladas en el Instituto de Matemáticas, bien han valido la pena.

A Christian Mena Ruvalcaba, porque ha sido un gran compañero y amigo durante los últimos semestres de la maestría, he aprendido de él muchas cosas importantes que han hecho posible la realización de este trabajo y solamente nosotros sabemos lo demandante que ha sido el trabajo bajo nuestro tutor (¡y de las recompensas de la investigación!) ☺; a Rafael Cruz Salas, Edgar R. Morales Contreras, Adidier M. Pérez Gómez, Ely Schoenfeld Liberman y Brenda Daniela Torres Castillo pues todos ellos han sido muy importantes para ser llevaderos estos semestres del posgrado. Compartiendo materias, haciendo tareas y lo más valioso, intercambiar conocimientos y formas de pensar sobre múltiples cuestiones extra académicas.

A todos mis compañeros de trabajo del Instituto de Geografía, aquí en la UNAM. Particularmente al Maestro José Quintero por el apoyo y facilidades prestadas durante la realización de mis estudios; a Ilia Jazmín Álvarez De Valle, una persona sensacional por su forma de ser y pensar, tan abierta y realmente preocupada por los demás. Tus comentarios, opiniones y consejos son y serán un referente en mi forma de ver la vida; a Josafat I. Guerrero Iñiguez una persona muy capaz que realmente se compromete con su trabajo y sobretodo su apoyo en múltiples ocasiones; a Luis Octavio Ramírez Fernández, Ana Rosa Rosales Tapia y Susana Cristina Almazán Colín,puesto que con ellos he aprendido más cosas aparte del aspecto laboral que en otros sitios. Los comentarios y "platicas" vertidas durante las horas de comida o bien en múltiples ocasiones mediante la mensajería instantánea han llegado en el momento adecuado. Recuerdo con mucho cariño mi estancia en esa institución con Ustedes.

A mis compañeros del grupo de trabajo "Refint", particularmente a César Martínez Gutiérrez, Eduardo Y. Ángel Hernández y Ann Margareth Meza Rodríguez. Excelentes alumnos con los cuales afortunadamente he podido trabajar e intercambiar muchos conocimientos y algunas anécdotas, empero realmente quedo a deber en ese intercambio.

A mis profesores, porque en mi formación siempre tendré presente algo de sus conocimientos, cada uno dentro de su área ha sido un gran aporte para lograr esta meta. Particularmente al Maestro Javier García García, por todos los valores y conocimientos que me ha transmitido y realmente ser amigos en las buenas y en las malas durante tanto tiempo. A la Doctora Hanna Oktaba, porque su forma de impartir cátedra ha valido para que mi interés en áreas antes desconocidas sea ahora mayor. A la Doctora Amparo López Gaona por sus consejos y sugerencias recibidas en momentos determinantes de la maestría. Al Doctor Héctor García Molina porque aún cuando lo he conocido muy

poco, es el ejemplo vivo de que una eminencia en computación no necesariamente es una persona engreída o soberbia, sino todo lo contrario. A Ustedes los admiro.

Al CONACyT y al "Macroproyecto: T. U. I. C." por el apoyo económico recibido en distintas etapas durante la realización de mis estudios de posgrado.

Por último, a quienes han sido mis alumnos (mi primer grupo de SMBD como ayudante y el segundo de SBD como titular siempre serán especiales), porque me han mostrado que la retroalimentación es fundamental en la enseñanza y en esa balanza he terminado por aprender más yo de ellos, que ellos de mi.

Y en general agradezco nuevamente a todos por su amistad y apoyo, me siento orgulloso y afortunado de conocer a personas tan valiosas (en muchos sentidos) como lo son ustedes GRACIAS.

René Alejandro Villeda Ruz

"Well, sure, the Frinkiac-7 looks impressive, don't touch it, but I predict that within 100 years, computers will be twice as powerful, 10,000 times larger, and so expensive that only the five richest kings of Europe will own them."

— Professor Frink

Índice general

Índice de tablas

Índice de figuras

Introducción

Durante los últimos años, el desarrollo de nuevas tecnologías que permiten la recuperación de grandes volúmenes de datos de manera ininterrumpida y en periodos de tiempo relativamente cortos, ha propiciado un aumento veloz y continuo en el tamaño de las bases de datos de infinidad de organizaciones. Todo ello ha generado grandes depósitos que concentran datos, generalmente de distintas fuentes y los cuales poseen en un estado pasivo información valiosa para las organizaciones. Gran parte de estos datos proviene de aplicaciones tan diversas como lo son la administración de recursos empresariales (*Enterprise Resource Management*), la administración de las relaciones con el cliente (*Customer Relationship Management*), las bitácoras de navegación por sitios Web o la representación de eventos metereológicos entre otros. Todo lo anterior ha ocasionado que las organizaciones se encuentren "**Ricas en datos pero pobres en conocimientos**"[56].

Las colecciones de datos se han vuelto complejas en contenido y continúan aumentando su tamaño tan rápidamente que el uso práctico de estos almacenes de datos (*Data Warehouse*) se vuelve limitando. Ya que si bien existen grupos de analistas dentro de cada organización dedicados a explotar estos datos, como humanos nos encontramos limitados en la capacidad de procesar de manera adecuada tal cantidad de datos y descubrir en ellos patrones o tendencias que representen información útil. Esta limitante ha demandado la creación de herramientas automáticas y semi-automáticas con la finalidad de auxiliar a los analistas y revelar esta información útil subyacente a los depósitos de datos.

Debido a lo anterior, durante la última década las organizaciones han implementado los sistemas de Inteligencia de Negocios (BI por sus siglas en inglés *Business Intelligence*). Dichos sistemas denotan al conjunto de estrategias y herramientas enfocadas a la administración y creación de conocimiento mediante el análisis de datos existentes en una organización o empresa. Este conjunto de herramientas y metodologías tienen en común las siguientes características:

- Facilitar el acceso a la información.
- Apoyar en la toma de decisiones.
- Orientación al usuario final.

De acuerdo a su nivel de complejidad se pueden clasificar las soluciones de Inteligencia de Negocios, en:

- Consultas e informes simples.
- Cubos OLAP de múltiples dimensiones.
- Minería de datos (*Data mining*).
- Sistemas de previsión empresarial.

La minería de datos constituye un área de investigación reciente y con un desarrollo muy importante, concibiendo numerosas técnicas y procedimientos que permiten la extracción de "conocimiento" implícito a partir de grandes volúmenes de datos. Los fundamentos teóricos de la minería de datos se encuentran en la inteligencia artificial y en el análisis estadístico.

Motivación

Particularmente, en años recientes se han ideado e implementado diversas técnicas como herramientas de apoyo en los procesos de minería de datos. Dichas técnicas se clasifican en dos grandes grupos: tareas descriptivas y tareas predictivas, ubicándose dentro de este último rubro las tareas de clasificación. Éstas constituyen el núcleo de los llamados modelos de clasificación, y se encargan de determinar la clase o tipo a la cual pertenece cada nuevo dato ingresado a la base de datos, todo ello con la finalidad de posteriormente predecir el estado de nuevos datos y a un nivel administrativo, brindar elementos que permitan orientar la toma de decisiones de las organizaciones.

Sin embargo una suposición fundamental de estas técnicas es asumir que el cúmulo de datos sobre los cuales trabajan se encuentra libre de errores o inconsistentes. Empero, debido a la arquitectura actual de los almacenes de datos, en la mayoría de los casos este conjunto de datos al proceder de distintas fuentes de datos se torna en un origen potencial de inconsistencias que deben ser resueltas mediante la aplicación de múltiples procesos de limpieza y transformación de datos. Dichos procesos de limpieza previos representan un gran porcentaje de tiempo y esfuerzo del proceso general de extracción de conocimiento.

Por ello, resulta indispensable contar con elementos que permitan a los analistas o encargados de la toma de decisiones, el determinar sobre la necesidad de ejecutar dichos procesos de limpieza y transformación de datos o bien suprimir dichas etapas del proceso general del extracción de conocimiento. Lo que permitirá utilizar ese tiempo en el desarrollo de otras actividades.

Objetivos

En el presente trabajo se estudia la opción de utilizar distintas técnicas de minado de datos, sobre conjuntos de datos con errores para determinar hasta qué punto se posee un modelo clasificador confiable. De manera puntual, los objetivos de la presente tesis son:

1. Estimar el porcentaje de error que se puede permitir en el conjunto de datos de entrenamiento para ser procesados posteriormente mediante diversas técnicas de minería de datos (particularmente clasificador ingenuo de Bayes, árboles de decisión, y k vecinos más próximos) de modo que este porcentaje de error no influya en la precisión del modelo clasificador y dicho modelo continúe siendo confiable.

2. Con base en lo anterior, determinar la necesidad de limpiar previamente los datos o dejarlos intactos, ahorrando así recursos para las organizaciones.

3. Comparar la precisión de cada clasificador utilizando diversos conjuntos de datos provenientes del repositorio "*UCI Machine Learning*"[1] .

4. Presentar elementos que permitan decidir a los analistas la necesidad de ejecutar procesos de limpieza sobre los datos que utilizarán o bien prescindir de estos procesos.

Resumen de capítulos

El presente trabajo se encuentra dividido en seis capítulos principales y un apéndice. En el capítulo 1 se presentan características del proceso de extracción de conocimiento de las bases de datos, sus etapas e importancia dentro de las organizaciones. Además se presentan de modo general las distintas tareas de minería de datos y su clasificación. En el capítulo 2 se analiza la primera técnica de clasificación utilizada: el clasificador ingenuo de Bayes y la teoría sobre la cual se fundamenta. En el capítulo 3 se desarrollan todos los elementos teóricos de los árboles de decisión para clasificación, en particular el basado en el algoritmo **C 4.5**, implementación que es utilizada durante el presente trabajo.

En el capítulo 4 se explican los detalles teóricos de la técnica de aprendizaje basada en instancias llamada "K vecinos más próximos" y su aplicación como método de clasificación. En el capítulo 5 se exponen todos los detalles relevantes de la experimentación llevada a cabo durante el presente trabajo, herramientas utilizadas para su realización, las características de los conjuntos de datos, peculiaridades de los errores introducidos y los resultados obtenidos utilizando los métodos anteriores de clasificación sobre 8 conjuntos de datos provenientes del UCI Machine Learning.

[1] http://mlearn.ics.uci.edu/MLRepository.html

El capítulo 6 presenta las conclusiones extraídas a partir de la investigación y experimentación realizadas y algunas líneas de investigación que pueden continuar el presente trabajo. Finalmente, el apéndice A describe las propiedades de las funciones de distribución que se utilizan durante la generación de errores para los conjuntos de datos de los experimentos.

Trabajo Relacionado

Actualmente existen muchos trabajos sobre los clasificadores. Desde mejoras estructurales para el manejo de datos relacionados y de conjuntos [58], pasando por la detección y la limpieza de errores tanto en los atributos clase como en el resto de los atributos [68] hasta la evaluación de estos durante la presencia de errores en los atributos clase [49].

El manejo de errores en los datos que se utilizan como conjunto inicial de datos por los clasificadores durante la fase de entrenamiento se ha manifestado de distintas formas, por ejemplo en [47] se presentan técnicas para la identificación de atributos que contienen errores que potencialmente pueden modificar el comportamiento de los modelos de datos y sus técnicas de explotación. Lo cual permite eliminar estos atributos donde se presentan los errores de modo que solamente se utilizan los atributos "buenos". Sin embargo, esto trae como consecuencia dos detalles muy importantes; el primero radica en que se elimina información que puede ser valiosa para el análisis posterior, y el segundo consiste en reconocer que dado que el porcentaje de error puede ser variable, la eliminación de un porcentaje elevado de tuplas puede evitar que se entrene de manera adecuada el clasificador.

Por otro lado, existen trabajos sobre clasificadores basados en patrones emergentes y su tolerancia al error [73], donde se enfatiza el hecho de que dado que los datos reales contienen errores, es necesario que un clasificador sea capaz de soportar inconsistencias en los datos. Presentando además técnicas que permiten evitar el sobre entrenamiento de los clasificadores.

Así mismo existen trabajos donde se aborda el comportamiento y eficacia que presentan los clasificadores bayesianos cuando se presentan errores en los datos [82]. Pero el enfoque del problema consiste en adoptar el modelo de errores dentro del modelo de aprendizaje de manera que se reconstruyen las probabilidades condicionales de distribución.

Por último, existen trabajos donde se estudia el comportamiento del clasificador ingenuo de Bayes en la presencia de errores con ciertas distribuciones [80] sin embargo no se contemplan otros tipos de clasificadores ni múltiples conjuntos de datos.

En el presente trabajo se hace hincapié sobre el porcentaje de error que puede ser aceptado en el conjunto de datos de entrenamiento para un modelo clasificador, en particular uno basado en el algoritmo ingenuo de Bayes, otro en el algoritmo C4.5 de árboles de decisión y el último basado en los k vecinos más cercanos, de modo que la precisión del modelo obtenido sea aceptable.

El trabajo resulta distinto por los siguientes puntos:

1. Se enfatiza el manejo de errores en los datos de entrada sobre los atributos característica para el entrenamiento de los modelos de clasificación. Asumiendo que los atributos clase no presentan errores.

2. En la presencia de errores sobre los datos, no se realiza un proceso de limpieza de datos sobre estos, sino que se manejan como información válida. Dado que el criterio para determinar la precisión del clasificador es el porcentaje de elementos correctamente etiquetados que brinda dicho modelo de clasificación como resultado del entrenamiento obtenido.

CAPÍTULO 1

Proceso de extracción de conocimiento

1.1. Introducción

En la actualidad la información de una organización se manifiesta no solamente como un elemento más dentro de la misma, sino también como un recurso muy importante y valioso, pues representa un factor de producción y desarrollo. Desde el punto de vista de las organizaciones se define al conocimiento como "la información que posee valor para ella, es decir, aquella información que permite generar acciones asociadas a satisfacer las demandas del mercado y apoyar las nuevas oportunidades a través de la explotación de las competencias centrales de la organización" [66].

Por lo anterior, es determinante para cualquier organización el contar con las herramientas que le permitan obtener la información requerida para satisfacer sus objetivos de manera rápida y confiable. Sin embargo, debido a la cantidad de datos que ingresan día a día a los depósitos de datos de las organizaciones, la tarea de extracción de información ha pasado de ser una práctica llevada a cabo enteramente por personas, a ser una tarea donde intervienen complejos procesos de minería de datos.

Es importante notar que el proceso de extracción de conocimiento no solamente incluye como elementos de trabajo aquellos datos que se encuentran dentro de una base de datos sino cualquier fuente de datos que sea útil para la organización. Sin embargo debido a la proliferación actual de los almacenes de datos, se vuelve importante enfocarse a este particular nicho de información.

1.2. El proceso de extracción de conocimiento en bases de datos

El proceso de extracción de conocimiento en bases de datos (KDD por sus siglas en inglés *Knowledge Discovery from Databases*), se define como el "proceso iterativo e interactivo no trivial de identificación de patrones válidos, previamente desconocidos, potencialmente útiles y en última instancia comprensibles a partir de los datos presentes en grandes bases de datos" [21] y constituye un elemento fundamental dentro de la familia de herramientas conceptuales y practicas que aglomera la **BI**.

Las fases principales que se identifican en el KDD, son [21]:

1. **Análisis**.- Consiste en determinar las fuentes de información que pueden ser útiles y el como conseguirlas.

2. **Diseño**.- Involucra el diseño general del esquema de un almacén de datos que consiga unificar de manera operativa todas las fuentes de información anteriormente identificadas.

3. **Implementación del almacén de datos**.- Generación y conformación del almacén de datos que permita la consulta y visualización previa de sus datos, esto para discernir cuales aspectos pueden interesar para ser objeto de estudio. Ésta fase incluye el proceso completo de integración.

4. **Preparación de datos**.- Fase que comprende diversas tareas como la selección, procesado previo, limpieza, reducción y transformación de los datos que se van a analizar.

5. **Minado de datos**.- Consiste en seleccionar y aplicar el método de minería de datos apropiado.

6. **Evaluación**.- Engloba tareas como la interpretación, transformación y representación de los patrones extraídos.

7. **Presentación**.- Incluye procesos de difusión y uso del nuevo conocimiento.

El proceso KDD involucra múltiples iteraciones y constantes ciclos entre las distintas fases. El flujo de datos básico sin presentar iteraciones o ciclos entre las fases se presenta en la Figura 1.1 (Basada en [22]). Es importante mencionar que la mayor parte de los estudios se han desarrollado dentro de la etapa de minería de datos, aún cuando las otras etapas también son igualmente importantes.

La minería de datos es una fase fundamental de todo el proceso, la cual se ve influenciada directamente por todas las fases anteriores, muy en especial por las fases de implementación del almacén de datos y de preparación de datos [60]. Congrega

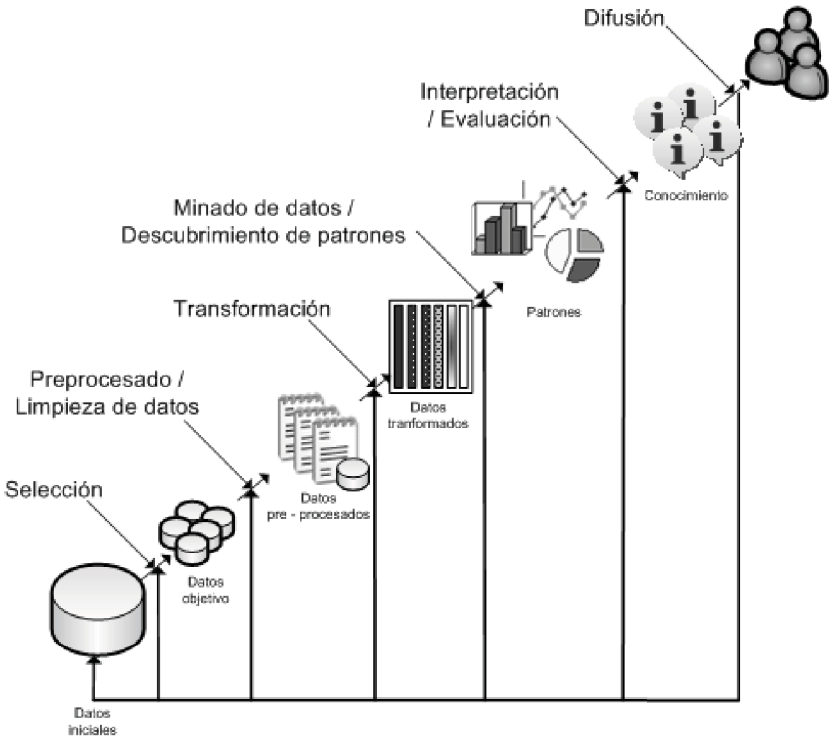


Figura 1.1: Flujo de datos durante las principales etapas del KDD.

una serie de técnicas que facilitan la extracción de conocimiento a partir de una gran cantidad de datos[1] .

Dicho conjunto incluye técnicas tan diversas como lo son: algoritmos genéticos, reconocimiento de patrones, aprendizaje automático, métodos estadísticos y herramientas de bases de datos en conjunción con los procesos analíticos en línea (OLAP por sus siglas en inglés *On-Line Analytical Processing*).

El término "Minería de Datos" es utilizado ambiguamente para referirse al proceso completo del KDD, sin embargo una interpretación que resulta más práctica consiste en tratar al KDD como el proceso general y a la minería de datos como las fases o sub-procesos con una aplicación práctica mayor.

Es importante observar que si bien el proceso general del KDD resulta ser costoso en términos de almacenamiento y tiempo de procesado, son las fases de **Implementación** y **Preparación de datos** aquellas que consumen una cantidad mayor de tiempo y recursos (aún más que las técnicas de minería de datos propiamente).

No obstante, en [12] se muestra que la etapa de **Preparación de datos** consume aproximadamente entre el 60 % y 80 % del tiempo total utilizado durante la extracción de conocimiento. Dicho fenómeno se presenta principalmente debido a los siguientes aspectos:

[1] Aunque no necesariamente pueden estar contenidos dentro de una base de datos.

- La enorme cantidad de datos por analizar.
  En la actualidad, los almacenes de datos condensan información de los cuales su tamaño oscila entre Gigabytes (GB) y Petabytes (PB)[2] .

- El manejo de datos con muchos atributos[3] .
  La cantidad de atributos por analizar aumenta considerablemente debido a la conjunción de datos a partir de las diversas fuentes de datos.

- Determinación de la semántica de los datos.
  El significado que posee cada dato dentro de un repositorio de datos, no necesariamente se preserva durante la integración de los datos. Dicho escenario vuelve necesaria esta verificación.

- La identificación de los elementos erróneos.
  La identificación de dichos elementos depende de la semántica que se da en determinado tiempo.

- La aplicación de algoritmos de selección, limpieza, transformación y reducción de datos.
  No todos los algoritmos utilizados presentan complejidad lineal, lo cual impacta de modo directo en los tiempos de ejecución.

Por esta razón, el disminuir o eliminar el tiempo utilizado en estas fases del proceso permitirá a las organizaciones enfocarse en otros elementos que les reditúen en una mayor eficiencia y eficacia dentro de sus propios procesos de **BI**.

Errores en los datos

Los almacenes de datos empresariales constituyen actualmente el modelo de almacenamiento más utilizado por diversas organizaciones debido a que proveen un ambiente para que éstas lleven a cabo una utilización eficiente de la información que está siendo generada y administrada por diversas aplicaciones operacionales simultáneamente.

Un almacén de datos se define según [41] como "una colección de datos homogénea, no volátil, variable en el tiempo y en la cual se encuentra integrada la información de la organización, utilizada como soporte para los procesos de toma de decisiones". El reunir los elementos de datos apropiados desde diversas fuentes de aplicación en un ambiente integral centralizado simplifica el problema de acceso a la información y en consecuencia, acelera los procesos de análisis y consultas así como también disminuye el tiempo de utilización de la información.

[2] Por ejemplo, el Centro Nacional de Investigación en Energía y Computación Científica (*NERSC*) en California, administra bases de datos de varios Petabytes: `http://www.nersc.gov`

[3] El volumen espacial ocupado por todos los atributos crece de modo exponencial con el número de atributos, y la densidad del entrenamiento disminuye proporcionalmente. Dicho fenómeno también se conoce como **la maldición de la dimensionalidad** [34].

Una característica primordial de los almacenes de datos, es que éstos se crean al extraer datos desde distintas fuentes heterogéneas como lo son:

- Bases de datos de aplicaciones operacionales.
- Archivos con formato de texto simple o complejo.
- Archivos con formato de hojas de cálculo.
- Medios de almacenamiento de aplicaciones heredadas.
- Equipos de adquisición de datos en tiempo real.
- Aplicaciones con registros de información continua en bitácoras.

La adquisición e integración de estos datos con los cuales se pretende crear una representación del mundo real, es una tarea difícil y muy propensa a errores, ya sea por eventos externos o internos a los medios de adquisición. Los distintos tipos de errores se pueden clasificar dependiendo de su origen, según [4] en:

- Errores de fuentes externas.- Son errores introducidos por eventos acontecidos en la realidad que se desea abstraer. Pueden ser causados por imperfecciones aleatorias en los datos, por ejemplo cambios en el voltaje de la línea de transmisión o instrumentos de medición mal calibrados.
- Errores de fuentes internas.- Errores introducidos por fallas internas de los sistemas, por ejemplo archivos recuperados de un disco duro en mal estado o escrituras a sectores de disco dañados.

En [52] se presenta un análisis con mayor detalle sobre el origen de datos ruidosos en los datos. En particular se plantea una subdivisión de datos ruidosos en **"datos no confiables"**, aquellos que se dan desde el mismo concepto que manejan (por ejemplo obtener una lectura de 1500 km/hrs en un dispositivo que mide la velocidad de un automóvil) y en **"datos erróneos"** (aquellos originados por errores humanos en la captura de datos). Además, se introduce una tercera categoría denominada **"información incompleta"** que denota los elementos que carecen de valores presentes para ciertas observaciones.

Para el presente trabajo, se consideran únicamente como errores aquellos datos que no pertenecen al rango de valores definido en determinado dominio, así como también la ausencia de algún dato.

Posterior a la adquisición de los datos, éstos son puestos a disposición de múltiples procesos de limpieza de datos para eliminar inconsistencias, resolver discrepancias semánticas y tratar la ausencia de elementos. Es preciso remarcar que esta fase de limpieza de datos se lleva antes de la fase de **Implementación** del KDD, o bien como sucede en la mayoría de los casos, es posterior a dicha fase. Los procesos de transformación,

reducción y combinación de datos ayudan a crear un ambiente idóneo para el acceso a la información. Este nuevo enfoque permite a las personas en todos los niveles de la organización efectuar su toma de decisiones con elementos mejor fundamentados.

Por otro lado una de las principales suposiciones de las técnicas de minería de datos, consiste en asumir que el conjunto de datos sobre los cuales se ejecutan se encuentran libres de errores y sin datos faltantes. Esto debido a que como se observa en la Figura 1.1, se identifica a la **Preparación de datos** como una tarea intermedia dentro del proceso general del KDD.

Dentro de los procesos de generación y adquisición de datos de las distintas organizaciones, es regla general que los datos erróneos recolectados constituyen una cantidad inferior que aquellos elementos que no necesitan limpieza alguna, ya que si esta situación no se presenta entonces el problema reside en las fuentes de datos y es ahí donde deberá realizarse un análisis detenido.

Por esta razón, cabe la posibilidad que debido a las características de los modelos clasificadores, no sea necesario llevar a cabo esta fase de limpieza de datos y con ello emplear este tiempo valioso en otras fases, obteniéndose resultados muy similares a aquellos en los cuales sí se llevará a cabo la fase de limpieza de datos.

1.3. Tareas de minería de datos

La minería de datos tiene como objetivo analizar los datos para extraer conocimiento. 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. Estas relaciones o resúmenes constituyen el modelo de los datos analizados. Existen muchas formas distintas de representar los modelos y cada una de ellas determina el tipo de técnica que puede utilizarse para inferirlos. En la práctica, los modelos pueden ser de dos tipos: predictivos y descriptivos.

- Modelos predictivos.- Estiman valores futuros o desconocidos de variables de interés, que se conocen como *variables objetivos* o *dependientes*, utilizando otras variables o atributos de la base de datos, a las que se conocen como *independientes* o *predictivas*.

- Modelos descriptivos.- Identifican patrones que explican o resumen los datos, es decir, sirven para explotar las propiedades de los datos examinados, no para predecir nuevos datos.

La minería de datos se ha utilizado de manera eficiente en múltiples tareas como detección de fraudes, administración y análisis de riesgos, segmentación y manejo de clientes, predicción de ventas, mercadotecnia, análisis de proteínas y diagnósticos médicos entre otras. Basados en la naturaleza de los problemas que se abordan, es posible catalogar a las tareas de minería de la siguiente forma:

- Clasificación.
- Agrupamiento.
- Regresión.
- Asociación.
- Predicción.
- Análisis de secuencias.
- Análisis de desviaciones

1.3.1. Clasificación

La clasificación es un problema de gran interés para los expertos del área de inteligencia artificial. Representa una de las principales tareas de minería de datos y se ubica dentro de los métodos de aprendizaje supervisado. Problemas tales como análisis de riesgos y mercadotecnia usualmente involucran algún tipo de clasificación. Dentro de este ámbito, los datos de entrada (denominados conjunto de entrenamiento) son instancias de las clases que se desean modelar e incluyen una serie de atributos o características. El objetivo de la clasificación es obtener una descripción precisa para cada clase utilizando los atributos de los datos de entrada, de modo que el modelo así obtenido puede servir para clasificar casos o nuevos elementos cuyas clases se desconozcan o, simplemente, para comprender mejor la información de la que dispone.

El modelo de clasificación puede construirse entrevistando a expertos. La mayor parte de los sistemas basados en conocimiento se han construido así a pesar de la dificultad intrínseca que la extracción manual del conocimiento posee. No obstante, si se dispone de suficiente información almacenada, el modelo de clasificación se puede construir generalizando a partir de ejemplos específicos mediante algún proceso inductivo. Los elementos de entrenamiento utilizados en la construcción del modelo de clasificación suelen expresarse en términos de un conjunto finito de propiedades o atributos con valores discretos o numéricos.

Al pertenecer a los métodos de aprendizaje supervisado, las categorías a las que han de asignarse los distintos elementos deben establecerse con anterioridad. En general, estas clases son disjuntas (aunque pueden establecerse jerarquías) y deberán ser discretas (para predecir atributos con valores continuos se suelen definir categorías discretas utilizando diversos términos, por ejemplo rangos representados por etiquetas).

Las técnicas inductivas de clasificación se basan en el descubrimiento de patrones en los datos de entrada, por lo que se recomienda disponer de suficientes casos de entrenamiento [33] para obtener un modelo de clasificación confiable. Además se necesitan suficientes datos para poder diferenciar patrones válidos de patrones debidos a

irregularidades o errores, esta diferenciación se suele realizar utilizando alguna técnica estadística.

La clasificación se refiere a la asignación de nuevos elementos dentro de categorías o clases basados en un atributo predecible. Cada elemento u observación contiene un conjunto de atributos y sus valores correspondientes donde a uno de ellos se le conoce como el atributo clase (atributo predecible). La tarea consiste en encontrar un modelo que describa el atributo clase en función de los valores de los atributos de entrada.

Este tipo de tarea consiste de varias etapas, primeramente se manifiesta la creación del modelo, luego un entrenamiento del mismo y para finalizar se considera una etapa de prueba. Para entrenar un modelo de clasificación es necesario conocer el valor del atributo clase de cada elemento de entrada del conjunto de entrenamiento, el cual es generalmente constituido por datos históricos.

Los algoritmos de minería de datos que requieren un objetivo sobre el cual compararse y evaluarse son considerados algoritmos supervisados, así pues, las tareas de clasificación constituyen un ejemplo de este último tipo de algoritmos.

Algoritmos de clasificación típicos se consideran los árboles de clasificación, las redes neuronales, los k-vecinos más próximos y el clasificador ingenuo de Bayes. Un árbol de decisión para cuatro clases posibles se presenta en la Figura 1.2.

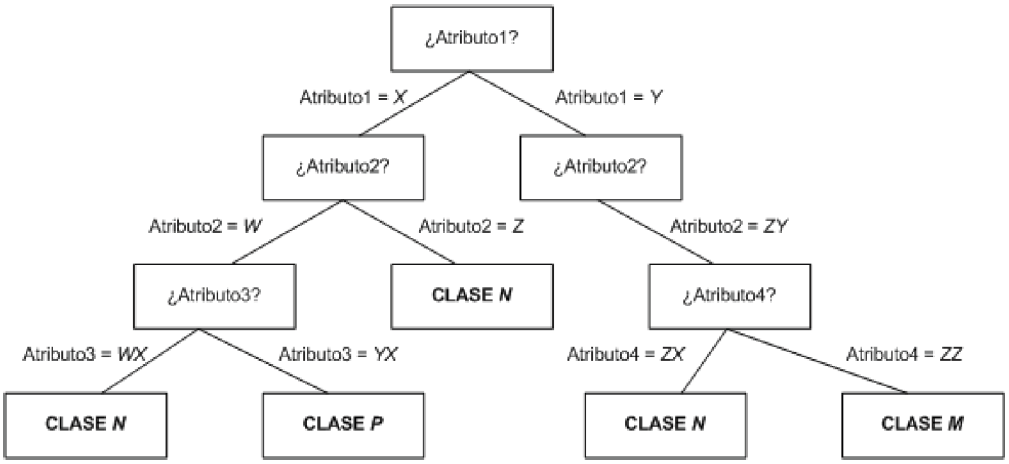


Figura 1.2: Un árbol de decisión.

1.3.2. Agrupamiento

El agrupamiento se utiliza para identificar grupos o conjuntos de elementos basados en los valores de sus atributos. Al finalizar el proceso es posible afirmar que los elementos de un mismo grupo tienen valores similares en el conjunto de atributos, es decir, poseen características semejantes y, al mismo tiempo son muy distintos a los objetos de otro grupo. El agrupamiento constituye una tarea no supervisada de minería de datos, ya que no se considera atributo alguno para guiar el proceso de entrenamiento y todos los

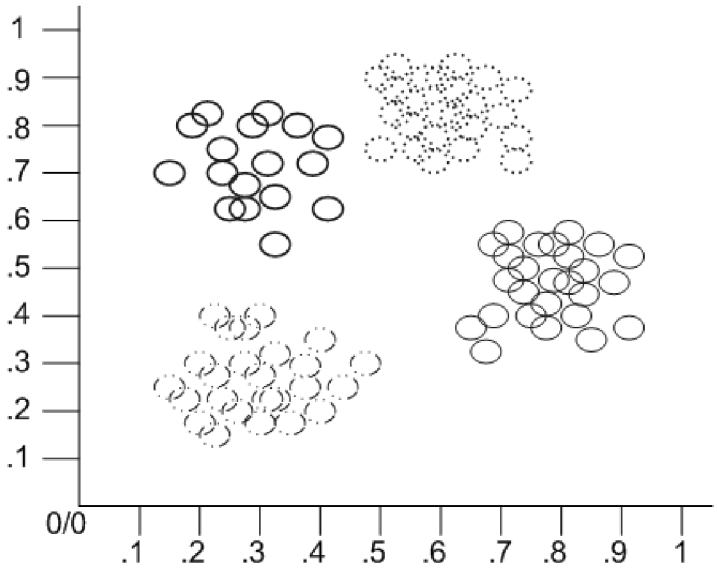


Figura 1.3: Agrupamiento sobre un conjunto de datos en dos dimensiones.

atributos son tratados de manera similar, siendo un problema potencial el encontrar una estructura dentro de una colección de datos no etiquetados. Al agrupamiento también se le conoce como segmentación, ya que parte o segmenta los datos en grupos que pueden ser o no disjuntos.

La mayoría de los algoritmos de agrupamiento construyen el modelo mediante varias iteraciones sobre el conjunto de datos y se detienen cuando se detectan convergencias dentro del modelo, es decir, cuando las fronteras de estos segmentos o grupos permanecen estables durante un número determinado de iteraciones.

Dentro de las principales áreas de aplicación se consideran la mercadotecnia, la biología y el análisis de riesgos naturales entre otras. Los algoritmos de agrupamiento se pueden clasificar en:

- Agrupamiento exclusivo.
- Agrupamiento traslapado.
- Agrupamiento jerárquico.
- Agrupamiento probabilístico.

Ejemplos de estos algoritmos son: algoritmo de K-medias, algoritmo de C-medias difusas, agrupamiento jerárquico y mezcla de Gaussianas.

1.3.3. Regresión

La regresión se refiere a una tarea muy similar a la clasificación. La diferencia principal radica en que el atributo predecible o atributo clase no es referente a un

dominio discreto sino continuo. El modo de trabajo de estos algoritmos consiste en generar una función real que asigna a cada elemento del conjunto de datos un valor real, de modo que la función se ajusta a dicho conjunto. Posteriormente se utilizan estos resultado para predecir el comportamiento de nuevos elementos al aplicarles dicha función.

El objetivo en este caso es minimizar el error (generalmente el error cuadrático medio) entre el valor predicho y el valor real. Las técnicas de regresión han constituido desde hace varias décadas un campo de estudio de los estadistas. Las áreas de aplicación incluyen la predicción del estado del tiempo, análisis de riesgos, administración de medicamentos o estudios químicos.

Los principales métodos de regresión incluyen a la regresión lineal, la regresión logística, árboles de regresión y redes neuronales.

1.3.4. Asociación

La tarea de asociación se ve ampliamente relacionada con el análisis de mercados. Del mismo modo, corresponde a una de las principales tareas de minería de datos. Los problemas que aborda consisten en analizar transacciones e identificar elementos que se repitan comúnmente entre distintas transacciones. La utilidad más práctica de la asociación se refiere a identificar conjuntos de elementos comunes y las reglas que les dan origen, y con ello generar propuestas para obtener nuevos elementos.

Dentro del contexto de la tarea de asociación, cada elemento o dicho de una manera general, cada par de datos atributo/valor es considerado como un solo ente (mejor conocidos como conjuntos de datos o "*datasets*"). La tarea de asociación comprende entonces dos objetivos: encontrar este conjunto de datos en otras transacciones y las reglas de asociación que les dan origen. La mayor parte de los algoritmos de asociación encuentran estos elementos luego de múltiples iteraciones sobre el conjunto de datos. Existe un concepto muy importante dentro de esta tarea de minería de datos llamado *soporte*, el cual representa el límite de frecuencia mínimo que se considera para cada conjuntos de datos por analizar, es decir un porcentaje mínimo de apariciones dentro del conjunto general.

Un elemento resultante de las tareas de asociación, son las reglas de asociación. Éstas, tienen la siguiente estructura genérica: $A, B \Rightarrow C$ con una probabilidad asociada, y donde A, B y C son todos conjuntos de elementos frecuentes. Esta probabilidad asociada se conoce como *confianza*. Sin embargo, las reglas de asociación no implican una relación causa-efecto, es decir, puede no existir una causa para que los datos estén asociados.

1.3.5. Predicción

Se considera una tarea de minería de datos complementaria de la clasificación. La predicción, como lo indica su nombre, brinda elementos congruentes y sustentados para

predecir el comportamiento de distintos fenómenos o determinar tendencias basadas en historiales. Usualmente utiliza como datos de entrada un conjunto de datos asociados con una o más variables de tiempo, por ejemplo una serie de números continuos (series de tiempo). Estas observaciones de datos típicamente contienen observaciones adyacentes, las cuales son dependientes del orden.

La técnica más conocida se denomina ARIMA por sus siglas en inglés *AutoRegressive Integrated Moving Average*.

1.3.6. Análisis de secuencias

Se utiliza para encontrar patrones en series de datos discretos. Una secuencia se compone de un conjunto de valores discretos, tales como etiquetas. Además, también se asocian con una componente representativa de tiempo,que al igual que la tarea de predicción, contienen observaciones adyacentes que son interdependientes, sin embargo la diferencia radica en que aquí las secuencias son representadas mediante valores discretos.

El análisis de secuencias y la asociación son similares en el sentido de que cada observación contiene una serie o conjunto de elementos o estados, empero, la diferencia entre las dos tareas se manifiesta en que el análisis de secuencias se enfoca a las transiciones que se presentan entre los estados o etiquetas, mientras que la asociación considera cada elemento de manera única e independiente. Por ejemplo, dentro de la tarea de análisis de secuencias, no es lo mismo adquirir el producto A y luego el producto B, que adquirir el producto B y posteriormente el A. Eventos que pueden ser tratados de manera similar dentro de la asociación como un único y equivalente conjunto de datos
.

Esta tarea de minería de datos es relativamente reciente pero ha adquirido relevancia debido a que las tareas donde se ocupa presentan un crecimiento importante, siendo dos de las más destacadas el análisis de registros de "navegación" en Internet y análisis de ADN. Dentro de las técnicas disponibles en esta tarea, destacan las cadenas de Markov.

1.3.7. Análisis de desviaciones

Dentro de nuestro contexto se considera una desviación como aquella observación que siendo atípica y/o errónea, presenta un comportamiento muy diferente con respecto de otros valores en una muestra escogida al azar de una población frente al análisis que se desea realizar sobre las observaciones experimentales.

El análisis de desviaciones se enfoca a encontrar aquellos eventos denominados como desviaciones del conjunto general de datos. También se le conoce como detección de excepciones (*outliers*), y de manera similar abarca la detección de cambios significativos a partir de observaciones anteriores del comportamiento o tendencias de los datos. Es decir, analizar los elementos que presentan un comportamiento distinto.

Este tipo de análisis se utiliza en aplicaciones de una diversidad de áreas. Siendo la

más común la detección de fraudes en transacciones bancarias; destacando también el análisis de redes y detección de intrusos. Estas tareas generalmente requieren el análisis de los datos en múltiples ocasiones, utilizando para ello versiones modificadas de árboles de decisión, redes neuronales, agrupamiento o algoritmos genéticos.

1.4. Mecanismos de aprendizaje

Las técnicas utilizadas en las tareas de clasificación se enmarcan de manera general dentro del contexto del aprendizaje de máquinas que a su vez constituyen un campo de estudio de la inteligencia artificial. Éste conjunto de técnicas se organizan en una taxonomía, basada en el resultado deseado del algoritmo. Dicha clasificación incluye los siguientes mecanismos de aprendizaje:

- Aprendizaje supervisado.- Incluye problemas donde se genera una función la cual asocia los elementos o parámetros de entrada a etiquetas o clases definidas con anterioridad. Es decir, se asocia un elemento representado por un vector $x = (x_1, \ldots, x_n) \in \Re^n$ a una de las r_k clases o etiquetas de la variable C. Dichas etiquetas ya están determinadas con anterioridad. La clase asociada se denota por c y adquiere algún valor entre $\{1, 2, \ldots, r_k\}$.

- Aprendizaje no supervisado.- Engloba aquellos problemas donde las etiquetas o clases existentes se desconocen así como también la pertenencia de cada elemento a la clase. Los algoritmos se encargan de determinar ambos problemas.

- Aprendizaje semi-supervisado.- Es una combinación de los dos anteriores. Existen tanto elementos con etiquetas asociadas como elementos sin etiqueta. Los algoritmos determinan a partir de los elementos con etiqueta, como se clasifican aquellos que carecen de ésta.

1.5. Modelos de clasificación

Aún cuando las tareas de minería de datos y en general el proceso de extracción de conocimiento de bases de datos son un área relativamente reciente, la mayoría de las técnicas que utilizan han existido desde años atrás. Las bases de los algoritmos que se utilizan derivan de estudios y trabajos de tres campos principalmente: la estadística, el aprendizaje de máquinas y las bases de datos.

La clasificación se refiere a la asignación de nuevos elementos dentro de categorías o clases basados en un atributo predecible, para ello se considera la abstracción de un conjunto de herramientas y elementos conceptuales que permiten llevar a cabo dicha tarea. Este conjunto se denomina de manera general **"Modelo de Clasificación"** o **"Modelo Clasificador"**.

1.6. Fases de los modelos de clasificación

Para la generación de un modelo de clasificación se necesita llevar a cabo una serie de etapas, desde el diseño conceptual hasta la aplicación del mismo. De manera general, el ciclo de desarrollo de un modelo de clasificación se compone de las siguientes fases:

1. Creación.
2. Entrenamiento.
3. Prueba.
4. Aplicación.

1.6.1. Fase de creación

La creación del modelo consiste en determinar dentro del contexto de la organización, aquellos medios que servirán como fuente de datos para el mismo, su semántica y estructura. Un modelo de minado de datos, y en particular para el presente trabajo un modelo de clasificación, se determina a partir de un conjunto inicial de datos (tablas relacionales o archivos con formatos predeterminados), elementos que los permiten identificar de manera única (llaves en el caso de tablas relacionales), elementos que representan observaciones de determinado fenómeno (atributos/valores, para tablas relacionales) y por lo menos un elemento o atributo que representa la clase por determinar. Además, cada modelo de clasificación se encuentra asociado a algún algoritmo clasificador, mismo que se encarga de tratar o analizar éstos últimos.

El modelo de clasificación es un contenedor similar a una tabla relacional, con la diferencia de que es utilizado para almacenar los patrones descubiertos por los algoritmos de clasificación de datos. Una representación gráfica de esto se presenta en la Figura 1.4.

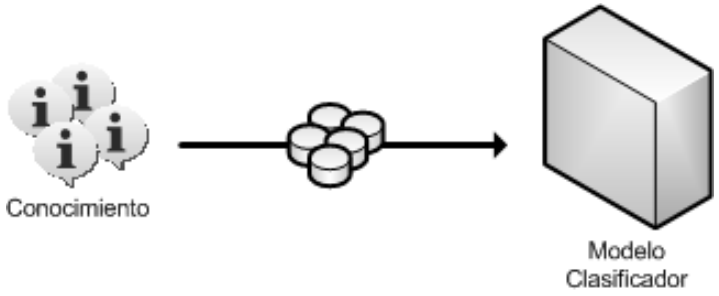


Figura 1.4: Fase de creación en el desarrollo de un modelo de clasificación.

1.6.2. Fase de entrenamiento

La fase de entrenamiento que también se conoce como **"Fase de Procesamiento"** o **"Fase de Aprendizaje"**, consiste en suministrar datos históricos al algoritmo de clasificación y encontrar patrones dentro de él, utilizando para ello los parámetros apropiados. A este conjunto de datos se le conoce como **"datos de entrenamiento"**, mismos que deben tener asignados previamente valores en el atributo clase. Para el conjunto de datos inicial es preciso utilizar únicamente una porción de los datos históricos como el conjunto de datos de entrenamiento, y el restante como el conjunto de datos de prueba.

En esta fase el algoritmo comienza por analizar los datos de entrada y dependiendo de su eficiencia, puede necesitar de una o más iteraciones sobre estos datos para encontrar las correlaciones entre los valores de los atributos.

Esta fase también puede ser considerada como aquella en la cual se encuentra la función $y = f(X)$, que predice la etiqueta o clase "y" asociada a cada elemento o tupla "X". Dentro de este contexto lo que se desea obtener es una función que separe los elementos en distintas clases. Típicamente esta asociación se representa en forma de reglas de clasificación, árboles de decisión o fórmulas matemáticas.

La fase de entrenamiento suele consumir bastante tiempo, pero generalmente se lleva a cabo dentro de las organizaciones de manera automática cada cierto tiempo, por ejemplo semanal o mensualmente. Como resultado de esta fase, el modelo de clasificación almacena los patrones que se descubrieron en forma de reglas. Una representación gráfica se muestra en la Figura 1.5.

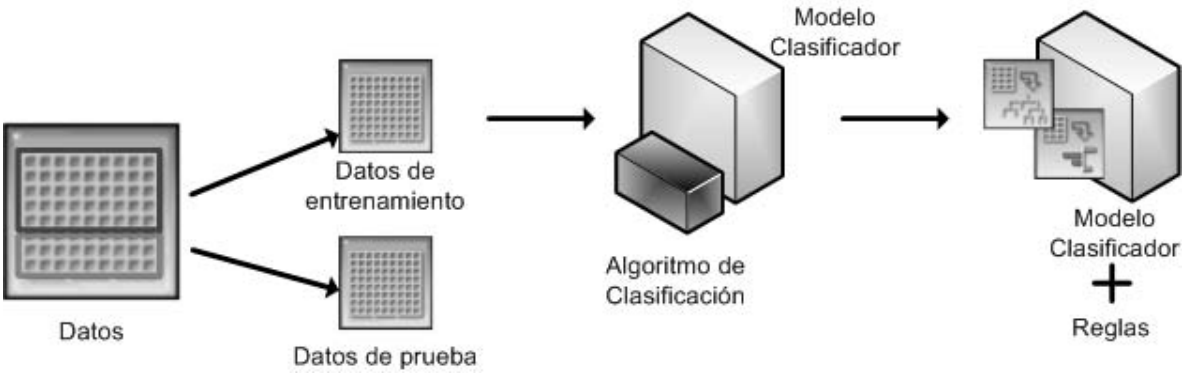


Figura 1.5: Fase de entrenamiento en el desarrollo de un modelo de clasificación.

1.6.3. Fase de prueba

La predicción generada por el modelo, se utiliza para aplicar sus patrones a un nuevo conjunto de datos y predecir el valor potencial de los atributos predecibles de cada nueva instancia. Así, es necesario contar con un conjunto de datos que no haya

sido analizado previamente por el algoritmo de clasificación y del cual se conozcan los valores del atributo clase, dado que si se utiliza el mismo conjunto de datos de la etapa de entrenamiento entonces los clasificadores presentan un fenómeno conocido como **"sobre entrenamiento"** ("*overfitting*"), el cual se refiere al hecho presente cuando el modelo clasificador reconoce únicamente los datos con los cuales fue creado y solamente estos elementos son a los cuales les asigna una etiqueta de manera correcta en la fase de prueba.

Una vez que se cuenta con este conjunto de datos de prueba, el modelo clasificador aplica las reglas o patrones que encontró durante la fase de entrenamiento sobre el conjunto de datos y dependiendo de sus valores asigna un resultado (su predicción) para cada instancia del conjunto de datos. Mismo que se debe comparar con el valor anterior y de este modo se verifica la precisión del modelo clasificador.

La fase de prueba en circunstancias normales no es tan compleja como las anteriores y para la mayoría de los modelos de clasificación suele ser muy rápida y como consecuencia, aplicarse en tiempo real.

Generalmente se considera como el objetivo final de una tarea de clasificación y además, la etapa que finaliza un ciclo dentro del análisis de datos. Esto con el propósito de volver a generar un nuevo modelo de clasificación tomando en cuenta nuevos datos. Por ello, el conjunto de reglas o patrones generados por un modelo de clasificación no son elementos inmutables durante el tiempo. La Figura 1.6 muestra gráficamente esta etapa.

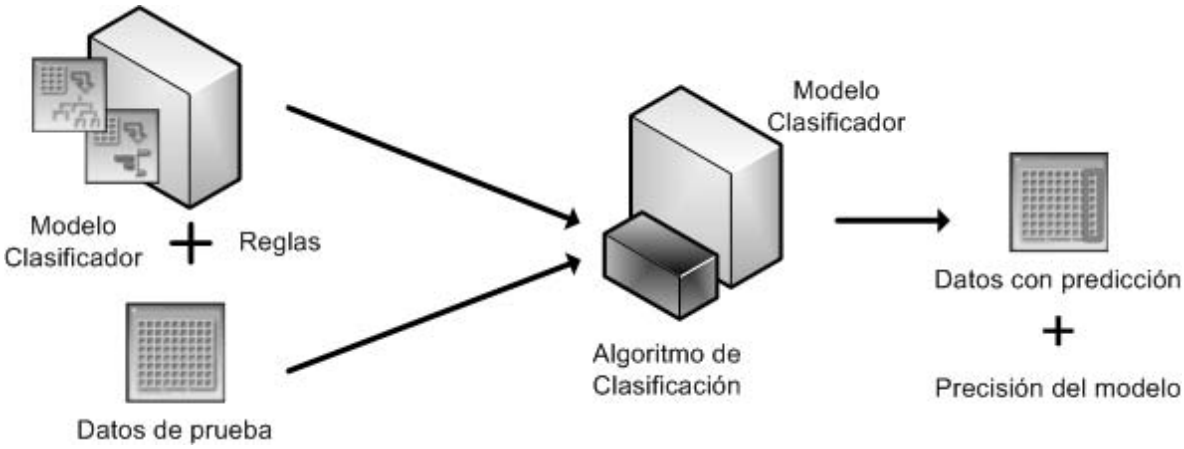


Figura 1.6: Fase de prueba en el desarrollo de un modelo de clasificación.

1.6.4. Fase de aplicación

Por último, la aplicación de este modelo de clasificación se lleva a cabo sobre un nuevo conjunto de datos, los cuales carecen de valor en su atributo clase y la obtienen durante esta etapa. Para posteriormente añadirse a la base de datos de conocimiento y

repetir el ciclo. Gráficamente, la última instancia del ciclo completo se presenta en la Figura 1.7.

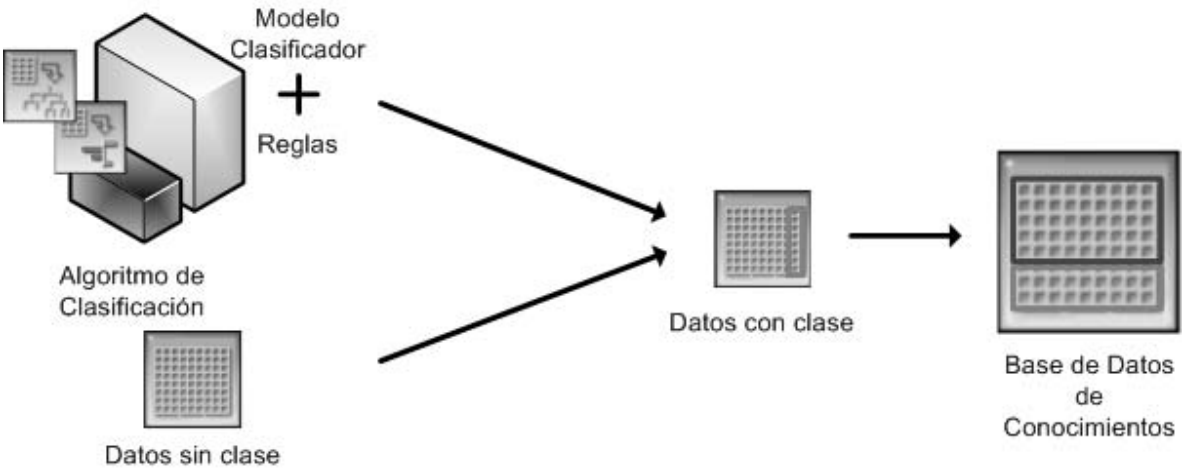


Figura 1.7: Ciclo completo del desarrollo de un modelo de clasificación. Los datos que han sido clasificados se agregan a la base de datos de conocimiento.

1.7. Precisión de un modelo de clasificación

La precisión asociada a un modelo de clasificación sobre un conjunto de datos en particular se define como "el porcentaje de elementos del conjunto de prueba que han sido correctamente etiquetados o catalogados por el clasificador tomando en consideración el conjunto de reglas asociadas al modelo en un tiempo dado" [33]. Por tanto, la precisión de un clasificador S para un conjunto de datos se obtendrá mediante la Ecuación 1.1.

$$precisión(S) = \frac{elementos\ bien\ clasificados}{elementos\ totales} \tag{1.1}$$

La veracidad de las reglas de clasificación obtenidas posteriormente de la ejecución del algoritmo de clasificación depende del estado de los datos en un tiempo dado, por lo cual es un concepto que se modifica durante el tiempo. Para conocer este porcentaje, es necesario comparar la etiqueta o el valor del atributo clase original con el valor asociado por el clasificador en el mismo atributo clase para un conjunto de datos distinto al conjunto de datos de entrenamiento, es decir, para elementos del conjunto de datos de prueba (Sección 1.6.3).

La precisión de un modelo clasificador permite determinar la utilidad práctica de este conjunto de reglas de clasificación, considerando el total de tuplas que se clasificaron efectivamente para cada una de las clases posibles. En ámbitos de reconocimiento de patrones, esta precisión se conoce como "**rango general de reconocimiento**".

Por otro lado, también es posible considerar el "**porcentaje de error**" o "**rango de clasificación errónea**" de un clasificador S, el cual consiste en evaluar $1-precisión(S)$ donde $precisión(S)$ corresponde a la precisión del clasificador S (Ecuación 1.1). En caso de que se llegue a utilizar el conjunto de entrenamiento como el conjunto de prueba para la estimación de esta precisión, entonces esta cantidad es conocida como el "**error de reemplazo**". No obstante esta estimación del error resulta ser una evaluación optimista del verdadero porcentaje de error, dado que el modelo no se prueba con muestras nuevas para él y constituye un factor importante para la aparición del fenómeno conocido como sobre entrenamiento (Sección 1.6.3).

Debido a que pueden existir múltiples clases asociadas a un modelo clasificador, es conveniente contar con herramientas que presenten de manera resumida la precisión de los clasificadores para cada una de estas posibles clases, una de estas herramientas se conoce como la **"Matriz de Confusión"** (C).

La matriz de confusión constituye una herramienta para analizar el desempeño de un modelo clasificador, ponderando qué tan bien ha reconocido tuplas o elementos para las distintas clases. Una matriz de confusión para un conjunto de datos con tres clases posibles se presenta en la Tabla 1.1.

Para un conjunto de datos con m clases distintas, una matriz de confusión es una tabla con al menos m por m entradas. La entrada $C_{i,j}$ en las primeras m filas y m columnas indican el número de tuplas o elementos de la clase i que fueron etiquetados por el clasificador como pertenecientes a la clase j. Para que se considere que un modelo de clasificación posee una precisión aceptable, idealmente la mayor parte de los elementos deben ser representados a lo largo de la diagonal de la matriz (es decir, han sido etiquetados con la clase i, cuando realmente pertenecen a esa clase i) desde la entrada $C_{1,1}$ hasta la entrada $C_{m,m}$; con el resto de las entradas con valores muy cercanos a cero. Existen modificaciones a esta configuración en donde además, la tabla puede contener más renglones o columnas conteniendo los totales o porcentajes de reconocimiento por cada una de las clases.

Predichas / **Actuales**	**clase1**	**clase2**	**clase3**	**Totales**
clase1	d	-	-	d
clase2	-	d	-	d
clase3	-	-	d	d
Totales	d	d	d	$3d$

Tabla 1.1: Matriz de Confusión.

Empero, en los casos reales ocurre que distintos errores tienen un costo muy distinto. El aprendizaje sensible a los costos puede considerarse como una generalización más realista del aprendizaje predictivo. En este contexto, la calidad de un determinado modelo se mide en términos de minimización de costos, en vez de minimización de errores. La manera más habitual de representar éstos costos en problemas de clasificación es

mediante la denominada Matriz de Costos (M) . En esta matriz se expresan los costos de todas las posibles combinaciones que se dan entre la clase provista por el modelo de clasificación y la clase real. Así, se tiene nuevamente que para un problema con m clases distintas, la matriz debe ser de tamaño $m \times m$. Donde la entrada $M_{i,j}$ denota el costo de clasificar un elemento de la clase i, como un elemento de la clase j. Un ejemplo se puede apreciar en la Tabla 1.2. Es importante notar que todos los elementos de la diagonal tienen el valor de 0 ($M_{i,i}$), debido a que los aciertos no poseen un costo[4] .

Predichas / **Actuales**	**clase1**	**clase2**	**clase3**
clase1	0	$	$$
clase2	$$	0	$$$
clase3	$$$	$$	0

Tabla 1.2: Matriz de Costos.

No obstante, conocer la matriz de costos de un determinado problema no es la situación habitual, por ello la estimación de costos supone en la mayoría de los casos un estudio previo que requiere de la inversión de tiempo y recursos, no siempre disponibles para la mayoría de las organizaciones. La situación más común es conocer los detalles de la matriz de costos en cuanto se entra en la fase de la aplicación del modelo. Cuando se desconoce la matriz durante la etapa de aprendizaje, se puede aplicar el análisis ROC (Sección 1.8) como otro método para evaluar el modelo de clasificación.

Así, para obtener una estimación del costo de un clasificador S cuando se conoce la matriz de costos y la matriz de confusión, se utiliza la siguiente fórmula:

$$Costo(S) = \sum_{1 \leqslant i \leqslant n, 1 \leqslant j \leqslant n} C_{i,j} \cdot M_{i,j}$$

En ella, la expresión $C_{i,j}$ refiere a la posición i, j de la matriz de costos y $M_{i,j}$ la posición i, j de la matriz de confusión. Por esto, se puede presentar la situación en donde un clasificador con una precisión mucho menor puede tener mayor costo.

1.7.1. Mejoras en la precisión de un modelo de clasificación

Dado que el cálculo de la precisión es solamente un estimado de qué tan bien se comportará el modelo clasificador en la presencia de nuevos elementos, es posible calcular límites de confianza para mejorar la calibración de este estimado.

Para la mejora en la precisión de un modelo clasificador se han ideado distintas técnicas. Éstas se enfocan en la modificación de la generación y suministro de los conjuntos de datos tanto de entrenamiento como de prueba. Algunas de estas técnicas son:

[4] Sin embargo, puede darse el caso de que se asocien valores negativos.

- Método de perdura.
  Método que consiste en dividir de manera aleatoria el conjunto original de datos en dos conjuntos independientes, el conjunto de datos de entrenamiento y el conjunto de datos de prueba. Normalmente dos terceras partes son elegidas como el conjunto de datos de entrenamiento y la restante tercera parte se utiliza como el conjunto de datos de prueba. Con esto se asegura que el estimado de la precisión será pesimista, dado que sólo una porción del conjunto de datos inicial se utiliza para derivar el modelo clasificador.

- Muestreo aleatorio.
  Método similar al de perdura dado que este se aplica k veces. Así, la precisión del modelo clasificador se determina mediante el promedio de las distintas precisiones obtenidas durante cada iteración.

- Validación cruzada.
  El método de Validación Cruzada consiste en dividir los datos de manera aleatoria en k subconjuntos mutuamente exclusivos. Generándose así los subconjuntos $A_1, A_2, \ldots, A_k$ de aproximadamente el mismo tamaño cada uno de ellos. Las fases de entrenamiento y prueba son llevadas a cabo k veces. En la iteración i, la partición A_i es considerada como el conjunto de datos de prueba, y el resto de los subconjuntos son utilizados colectivamente como el conjunto de datos de entrenamiento.
  En este método se garantiza, a diferencia de los anteriores, que cada subconjunto de datos es utilizado el mismo número de veces para funcionar como conjunto de entrenamiento y una vez para servir como conjunto de prueba.
  Dentro del contexto de las tareas de clasificación, la precisión estimada para el modelo es el número total de elementos correctamente etiquetados de las k iteraciones, dividido entre el número total de tuplas o elementos en el conjunto inicial de datos.

- Intercambio masivo
  El método de intercambio masivo consiste en asumir que la elección de una tupla o elemento para su adición al conjunto de datos de entrenamiento es distribuida uniformemente con reemplazo, es decir, cada que se elige un elemento es igualmente probable que sea reelegido y agregado nuevamente al conjunto de entrenamiento. Existen varias implementaciones de este método, cada una de ellas modifica el porcentaje de elementos que se consideran para cada conjunto de datos (entrenamiento y prueba). Una muy conocida es la **.632** que funciona de la siguiente manera: Dado un conjunto de datos inicial con tamaño d, este conjunto es muestreado d veces con reemplazo, generando una muestra *autosuficiente* o conjunto de entrenamiento con d elementos, de modo que existe una probabilidad muy alta de que varios elementos sean elegidos más de una vez, pero aquellos elementos que no se presentan en el conjunto de entrenamiento, serán elegidos para formar parte

del conjunto de prueba. Repitiendo el proceso varias veces, se tiene que un 63.2 % terminará dentro del conjunto de datos de entrenamiento y el restante 36.8 % en el conjunto de datos de prueba.

1.7.2. Métodos de creación de conjuntos de clasificadores para la mejora de precisión

Dado que existen múltiples algoritmos de clasificación y cada uno de ellos puede asociarse con distintas tareas, se pueden generar múltiples modelos de clasificación para un solo conjunto de datos. Por ello, también se identifican técnicas que permiten la combinación de dos o más modelos clasificadores a modo de generar soluciones más generales y precisas. Existen por tanto, muchas formas de generar y combinar conjuntos de modelos. En [14] se establece una taxonomía de métodos de construcción de conjuntos de clasificadores.

- Manipulación de los datos de entrenamiento.
- Manipulación de los atributos de entrada.
- Manipulación de los datos de salida.
- Métodos aleatorios.

Dentro de los primeros, se encuentran aquellos métodos que construyen un conjunto de modelos mediante la ejecución repetida del mismo algoritmo de clasificación. En cada iteración se parte de un conjunto diferente de datos de entrenamiento. La parte más importante es la definición de un mecanismo adecuado para la selección de subconjuntos a partir de los datos de entrenamiento original. Ejemplos de estos métodos son el Consenso y el Aumento, que a continuación se presentan con mayor detalle.

- Consenso
  La idea que subyace a este método, consiste en elegir una etiqueta para un elemento dado a partir de un consenso entre los resultados de múltiples clasificadores para el mismo elemento. Es decir, se genera un modelo clasificador general GC que contendrá los elementos por clasificar y la clase de dichos elementos, además de m distintos modelos clasificadores S_i [5] Para asignar la clase a un elemento, basta con que por lo menos la mitad más uno $((i/2) + 1)$ de los clasificadores asignen una etiqueta similar para dicho elemento. Así ese elemento sea clasificado dentro del clasificador general con etiqueta que se presentó en mayor número de ocasiones.

  En cuanto a la asignación de lo conjuntos de entrenamiento y prueba, se consideran k iteraciones donde en cada una de ellas se genera un conjunto de entrenamiento

[5] Posiblemente cada uno asociado a un algoritmo clasificador distinto.

(D_i) utilizando para ello un muestreo con reemplazo a partir del conjunto D original (son k aplicaciones del método de intercambio masivo). Debido a ello, algunos elementos del conjunto original D pueden no aparecer en algún conjunto de entrenamiento D_i, mientras otros elementos pueden aparecer en más de una ocasión.

Posteriormente, cada conjunto de entrenamiento generado (D_i) se asigna a un modelo clasificador S_i. Para clasificar un elemento, cada clasificador S_i devuelve su predicción y se toma como un voto. El clasificador más general, GC cuenta los votos para cada valor en el atributo clase y asigna la etiqueta que adquiera más votos al elemento en cuestión.

El clasificador creado con Consenso generalmente posee una precisión mayor que un único clasificador derivado del conjunto original de entrenamiento. Además no presenta un desempeño menor y es más robusto a los efectos de datos ruidosos. El incremento de la precisión se debe a que el modelo compuesto reduce la varianza de cada clasificador. Una representación gráfica de este método de mejora, se muestra en la Figura 1.8.

- Aumento.
  La técnica de Aumento se asemeja mucho al consenso debido a que nuevamente se toman en cuenta los resultados generados por k clasificadores para cada elemento por clasificar del conjunto de prueba. Sin embargo, para la elección de la clase a la que pertenece este elemento, se pondera cada resultado y la clase final consiste en una selección restringida de los clasificadores más precisos.

  En este caso, se asigna cierto peso o importancia a cada elemento del conjunto de entrenamiento. Nuevamente se crean una serie de k clasificadores[6] . Cuando el clasificador S_i ha terminado su etapa de entrenamiento, los pesos son actualizados para el subsecuente clasificador, es decir el S_{i+1}. Tomando atención especial a los elementos que no fueron correctamente etiquetadas por el clasificador S_i. El último clasificador, llamado clasificador Aumentado S_*, combina los votos llevados a cabo por cada clasificador S_i, donde el peso de cada voto de cada clasificador se encuentra en función de su precisión.

1.7.3. Aceptación de un modelo de clasificación

Una vez que se conoce la precisión de un clasificador, es necesario determinar cuando este modelo se convierte en un modelo valioso para la organización. Si bien a un mayor porcentaje general de precisión corresponde un número mayor de elementos correctamente etiquetados, también es cierto que dependiendo del contexto para el cual se ha construido el modelo será su aceptación o rechazo. Por ejemplo, si se considera un

[6] De igual forma, estos clasificadores pueden estar asociados a distintos algoritmos de clasificación.

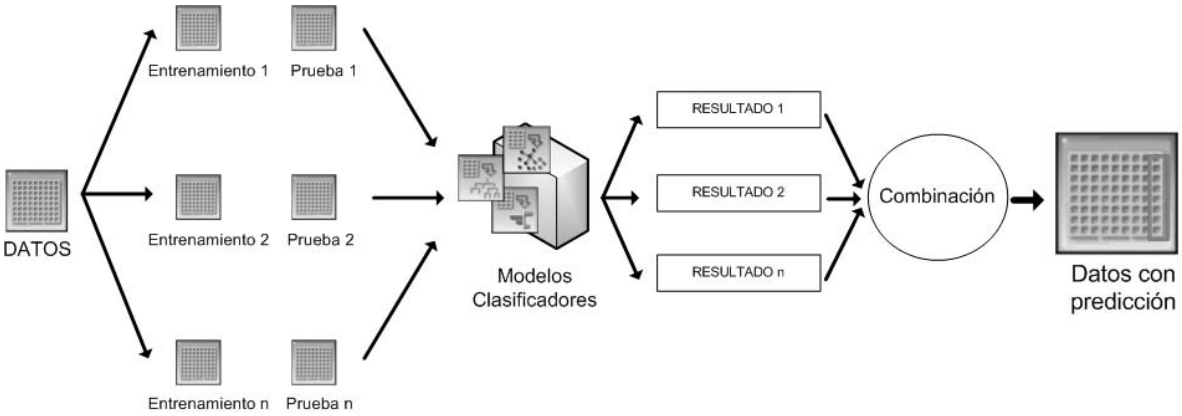


Figura 1.8: Combinación de modelos utilizando Consenso.

clasificador que se encarga de analizar datos médicos de pacientes y determinar si un paciente tiene o no cierta enfermedad, un porcentaje de precisión del 90 % puede crear la ilusión de que el clasificador realmente se está comportando bien, pero qué sucede si solamente entre el 3 % y 4 % de los elementos del conjunto de datos de entrenamiento poseen una etiqueta de "positivo" (asumiendo que únicamente se cuenta con las etiquetas "positivo" y "negativo", para manifestar que el paciente posee o no la enfermedad en cuestión). Claramente la precisión del 90 % no es aceptable para el clasificador - puede darse el caso de que este clasificador solamente se encuentra etiquetando correctamente los elementos con clase "negativo". En este caso es necesario determinar qué tanto el clasificador se encuentra reconociendo de manera correcta los elementos con etiqueta "positivo" y qué tan bien reconoce los elementos con la otra etiqueta ("negativo").

Por lo anterior, se afirma que los modelos de clasificación son sensibles a la distribución (no siempre todas las clases tienen la misma proporción o están balanceadas) y al costo ya que en muchas situaciones todos los errores producidos por un modelo predictivo no tienen las mismas consecuencias (Sección 1.7).

En casos donde se poseen datos por clasificar y únicamente se tienen 2 clases viables, es posible extender el análisis para la aceptación de un modelo de clasificación mediante el análisis ROC (Sección 1.8).

Como detalle importante es preciso considerar que en la mayoría de los problemas de clasificación, se asume que todas las tuplas o elementos son unívocamente clasificables, es decir, cada uno de ellos pertenece solamente a una clase. Sin embargo debido a lo vasto que resultan los problemas de clasificación, esta suposición no siempre es razonable. En estos casos la medida de precisión no es aplicable dado que un elemento puede ser correctamente etiquetado para varias clases, empero se puede asociar una probabilidad de pertenencia para cada una de las posibles clases.

Por último, si la precisión del clasificador se considera aceptable bajo los criterios anteriormente descritos, entonces este clasificador puede ser utilizado para evaluar y catalogar datos que ingresen posteriormente y de los cuales no sea conocida la etiqueta

o el valor del atributo clase (Sección 1.6.4).

1.8. Evaluación de un modelo de clasificación

En la actualidad existen múltiples algoritmos que permiten la construcción de distintos modelos de clasificación, mismos que divergen tanto en eficiencia como en eficacia. Por lo cual es preciso contar con parámetros que permitan determinar cuando un modelo M_1 es mejor que otro modelo M_2, puede parecer intuitivo elegir únicamente tomando en cuenta la precisión que posee cada uno de ellos, sin embargo se deben considerar más características, por ejemplo:

- Precisión.- La precisión del modelo de clasificación se refiere a la capacidad de predecir correctamente el valor del atributo clase para un conjunto de datos nuevo (datos que no tienen valor asociado en el atributo clase). La precisión puede estimarse utilizando uno o más conjuntos de datos de prueba que son independientes del conjunto de datos de entrenamiento. Existen distintos métodos, pero destacan la validación cruzada y el intercambio masivo vistos anteriormente.
- Velocidad.- Este rubro se refiere al costo computacional involucrado en generar y utilizar el modelo clasificador.
- Robustez.- Se refiere a la habilidad del modelo clasificador para llevar a cabo su tarea en la presencia de datos ruidosos o en la ausencia de estos.
- Capacidad en el manejo de datos.- Es la habilidad de construir y ejecutar el modelo clasificador en presencia de grandes cantidades de datos manteniendo un desempeño aceptable.
- Facilidad de interpretación.- Describe el nivel de comprensión y guía que brinda el modelo. Esta característica es un tanto subjetiva y por lo mismo, más difícil de conseguir.

Sin embargo, como se menciona en la Sección 1.7.3, los modelos de clasificación al ser sensibles a la distribución de las clases y sobretodo al costo de elementos clasificados erróneamente, presentan otras medidas de evaluación por ejemplo, el análisis ROC.

Análisis ROC

Considerando un problema donde únicamente se tienen dos clases para asignar. Sea p la clase positiva y n la clase negativa, existen 4 posibles resultados a partir de un modelo de clasificación. Si el resultado de la predicción es p y el valor actual es p, entonces se le conoce como un **"Verdadero Positivo"** (TP), por otro si el valor de la predicción del modelo es n entonces se trata de un **"Falso Positivo"** (FP). De modo similar, si el resultado de la predicción es n y el actual es n se denomina un

"Verdadero Negativo" (TN) y si la predicción es p el elemento se determina como un **"Falso Negativo"** (FN). La Tabla 1.3 resume lo anterior.

Predichas / **Actuales**	(p)	(n)
(p)	TP	FP
(n)	FN	TN

Tabla 1.3: Matriz de Confusión para un problema con 2 clases.

A partir de estos resultados, se define un conjunto de métricas que permiten evaluar dichos modelos [18], dicho conjunto aparece en la Tabla 1.4.

Por otro lado, la mala asignación de un elemento contiene intrínsecamente un costo asociado. Así, lo importante no es obtener un modelo de clasificación que falle lo menos posible sino uno que tenga un costo asociado menor. Para llevar a cabo esto se asocia a la matriz de confusión el costo en cada una de sus entradas y con ello se determina el costo total de cada modelo (Sección 1.7). Un problema que se presenta con respecto a la matriz de costos, es que no siempre se cuenta con esta información, además de que el costo final depende de dos elementos dependientes del contexto de cada problema:

1. El costo de los falsos positivos y falsos negativos: *FPcost* y *FNcost*.
2. El porcentaje de ejemplos de la clase negativa respecto de ejemplos de la clase positiva. (Neg / Pos).

Métrica	Siglas	Definición
Rango de TP (sensibilidad)	TPR	$TPR = TP/P = TP/(TP + FN)$
Rango de FP (falsa alarma)	FPR	$FPR = FP/N = FP/(FP + TN)$
Precisión	ACC	$ACC = (TP + TN)/(P + N)$
Especificación	SPC	$SPC = TN/(FP + TN) = 1 - FPR$
Valor de predicción positiva	PPV	$PPV = TP/(TP + FP)$
Valor de predicción negativa	NPV	$NPV = TN/(TN + FN)$
Rango de descubrimiento falso	FDR	$FDR = FP/(FP + TP)$

Tabla 1.4: Conceptos y terminología de una matriz de confusión para dos clases.

Para solventar estos inconvenientes, se utiliza el Análisis ROC (*Receiver Operating Characteristic*). Técnica utilizada por primera vez para evaluar radares en la segunda guerra mundial y posteriormente en el análisis de respuesta de transistores. Dicha técnica se desarrolló fundamentalmente para aplicaciones de diagnóstico médico a partir de 1970 y comienza a popularizarse a finales de los años 90 en minería de datos.

El análisis ROC se basa en generar una gráfica mediante un conjunto de puntos dados por los valores TPR y FPR, el primero determina un clasificador y su desempeño

en clasificar instancias positivas entre todos lo elementos disponibles durante la fase de prueba. Mientras el segundo determina cuantos resultados incorrectos positivos se dan, cuando realmente son negativos entre todos los elementos negativos disponibles durante la misma fase. En un espacio ROC se define el eje X con los valores de los FPR y el eje Y con los de TPR, dado que el TPR es equivalente a la sensibilidad, y FPR es equivalente a (1 - especificación), la gráfica ROC se conoce como la tabla de sensibilidad y (1 -especificación). Cada instancia de la matriz de confusión representa un elemento en la gráfica ROC. Una gráfica ROC se presenta en la Figura 1.9.

El método ideal se asocia mediante un punto en la esquina superior izquierda $(0, 1)$ del área ROC, mismo que representa una sensibilidad y especificación de 100 %. Una línea diagonal divide el área ROC en secciones de "buena" y "mala" clasificación. Los puntos sobre esta diagonal indican resultados aceptables y puntos debajo de esta línea reflejan resultados erroneos.

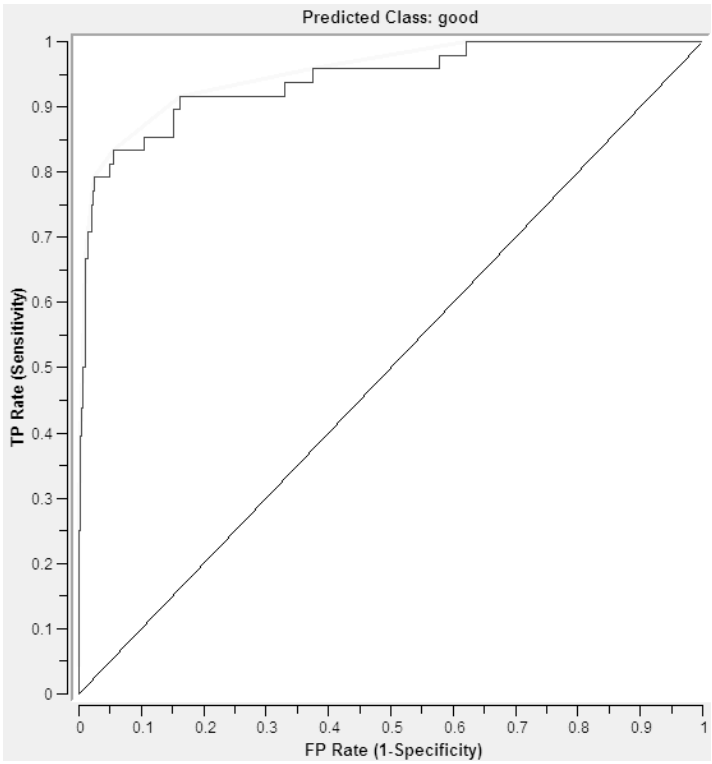


Figura 1.9: Gráfica de ROC.

Empero, el análisis ROC no es fácilmente extensible para más clases, ya que dadas n clases, se genera un espacio de n x $(n - 1)$ dimensiones, donde es necesario calcular la envolvente convexa para determinar los resultados aceptables y aquellos erróneos lo cual resulta muy complejo y difícil de visualizar. Aunque se consideran extensiones para el manejo de más clases [36], se ha observado que no existe una medida idónea para ponderar el desempeño de un modelo de clasificación [35].

1.9. Recapitulación

El proceso de extracción de conocimiento a partir de bases de datos, constituye un elemento fundamental en los procesos de BI dentro de las organizaciones. Sin embargo la etapa de **Preparación de datos** consume gran parte de los recursos del proceso en general. Por ello se plantea la creación de un criterio que permita determinar el ejecutar o no, los distintos procesos que comprenden esta etapa. Y en caso de no llevarse a cabo, ocupar estos recursos en otras actividades de la organización.

Una de las principales tareas de minería de datos, es la clasificación que consiste en asignar etiquetas o clases a los distintos elementos que ingresan a la base de conocimientos, generando así una predicción sobre nuevas instancias. La precisión de un modelo clasificador se define como el porcentaje total de elementos que son correctamente clasificados por dicho modelo clasificador. No obstante, no es la única medida de confiabilidad que se tiene para un modelo clasificador, también existen otras como el análisis ROC, diseñada inicialmente para problemas donde solamente existen dos posibles clases y cuya extensión a problemas con más clases resulta costosa computacionalmente y no representa una medida general del buen desempeño.

Debido a lo anterior, durante la realización del presente trabajo se ha optado por realizar un análisis basado únicamente en la precisión del modelo de clasificación.

CAPÍTULO 2

Clasificador Ingenuo de Bayes

2.1. Introducción

Las técnicas de minería de datos se caracterizan por funcionar sobre conjuntos de datos que representan una realidad dada, sin embargo uno de los principales problemas que presentan es el manejo de la incertidumbre. Hecho que no resulta para los métodos y técnicas bayesianas, dado que una de sus características es la utilización explícita de la teoría de la probabilidad para cuantificar la incertidumbre.

Los métodos bayesianos representan un modelo que permite un doble uso: descriptivo y predictivo. Dentro de la utilización como modelos descriptivos, sus algoritmos de aprendizaje se centran en el descubrimiento de las relaciones de independencia y/o relevancia entre sus variables, por lo que el modelo resultante del análisis refleja de forma explícita numerosas relaciones de interés. En cuanto al uso predictivo, la principal aplicación se enfoca a su utilización como clasificadores, enfoque fundamental que se trata en el presente trabajo.

El modelo clasificador ingenuo de Bayes (**"Naïve Bayes Classifier Model"**) se enmarca dentro del contexto de las tareas de clasificación supervisada. La teoría de la probabilidad y los métodos bayesianos que establecen sus bases teóricas, constituyen algunos de los elementos más utilizados en problemas de inteligencia artificial, por tanto, en problemas de aprendizaje automático. Como se menciona en [75], las 2 principales razones por las cuales los métodos bayesianos se consideran relevantes para su utilización son las siguientes:

1. Se identifican fácilmente como un método práctico para realizar inferencias a partir de los datos históricos

2. Facilitan un marco de trabajo útil para la comprensión y análisis de numerosas técnicas de aprendizaje y minería de datos que no trabajan de manera explícita con probabilidades.

Los métodos bayesianos se encuentran con el inconveniente del costo computacional que requiere llevar a cabo dicho enfoque. Por lo cual se considera común formular ciertas suposiciones que permiten restringir la complejidad de los modelos a utilizar.

Aunado a esto, estudios sobre el desempeño de los clasificadores Bayesianos analizados como técnica de aprendizaje automático han reportado un comportamiento muy similar a otras técnicas ampliamente utilizadas como lo son los árboles de decisión, máquinas de soporte vectorial y redes neuronales, tanto en términos de precisión del modelo [15] [50] y [45]; como con otras medidas de evaluación [39] (Análisis del Área Bajo el ROC[1]) y en su desempeño en presencia de grandes bases de datos. Todo ello aún en casos donde la suposición de independencia pareciese poco realista.

De manera gradual los investigadores de la comunidad de aprendizaje automático y adquisición de conocimiento se han dado cuenta de su potencialidad y robustez en problemas de clasificación supervisada. Razón por la cual es de particular interés analizar su comportamiento en presencia de datos ruidosos o incompletos.

En el presente capítulo se desarrolla la teoría que fundamenta al modelo clasificador ingenuo de Bayes, sus características y razones por las cuales ha sido elegido como uno de los modelos de clasificación por analizar.

2.2. Clasificadores bayesianos

El contexto en el cual se enmarcan los clasificadores bayesianos, tiene su fundamento teórico sobre el teorema de Bayes, así como la utilización de la hipótesis de independencia condicional de las variables predictoras, es decir, se asume que la influencia del valor en un atributo para una cierta clase es independiente de los valores de los restantes atributos. Ésta última suposición se lleva a cabo con el fin de simplificar los cálculos involucrados. Existen otras técnicas como las **redes Bayesianas**[2] , que permiten la representación de dependencias entre subconjuntos de atributos y también pueden utilizarse para tareas de clasificación.

Aún cuando existe una historia extensa en los procesos de reconocimiento de patrones, el clasificador ingenuo de Bayes (Naïve Bayes) es mencionado por primera vez dentro del área de aprendizaje no supervisado a finales de la década de los años ochenta [8] con la idea de llevar a cabo una comparación sobre su capacidad de predecir con respecto a otros métodos más sofisticados[3] .

[1] "*Area Under the Receiver Operating Characteristics*".

[2] Se les conoce también como **redes de confianza Bayesiana**, **redes probabilistas** o **redes causales**.

[3] Específicamente contra los árboles de decisión.

Esta técnica de clasificación y la hipótesis sobre la cual se fundamenta se identifica con varios nombres, algunos de ellos son: idiota Bayes [37] , simple Bayes [45], independiente Bayes [27] e ingenuo Bayes [76]. Durante el presente trabajo se utilizará ésta última denominación.

El teorema de Bayes constituye la regla básica para realizar inferencias, permite actualizar el conocimiento que se tiene sobre un suceso ó conjunto de sucesos en la presencia y análisis de nuevos datos u observaciones. Es decir, permite pasar de la *probabilidad a priori* ($P(suceso)$) a la *probabilidad a posteriori* ($P(suceso|observaciones)$). La probabilidad a priori puede referirse como la probabilidad inicial, la que se determina sin saber nada más. La probabilidad a posteriori es aquella que se obtiene tras conocer cierta información, por tanto, puede tomarse como un refinamiento del conocimiento. El teorema de Bayes se enuncia a continuación.

El teorema de Bayes

Teorema 1 (Bayes). Sean A y B dos sucesos aleatorios cuyas probabilidades se denotan por $P(A)$ y $P(B)$ respectivamente, y además se cumple que $P(B) > 0$. Supongamos conocidas las probabilidades a priori de los sucesos A y B, es decir, $P(A)$ y $P(B)$, así como la probabilidad condicionada del suceso B dado el suceso A, es decir $P(B|A)$. La probabilidad a posteriori del suceso A conocido dado que verifica el suceso B, es decir $P(A|B)$, puede calcularse a partir de la siguiente fórmula:

$$P(A|B) = \frac{P(A,B)}{P(B)} = \frac{P(A)P(B|A)}{P(B)} \tag{2.1}$$

Los elementos que enmarca la Ecuación 2.1 son la probabilidad a priori de la hipótesis $P(A)$ y de las observaciones $P(B)$ y las probabilidades condicionadas $P(A|B)$ y $P(B|A)$. A esta última se le conoce como la *verosimilitud* de que la hipótesis A haya producido el conjunto de observaciones B. Dentro del contexto de clasificación, donde se identifica una clase (C) y un conjunto de variables predictoras o atributos $\{A_1, ...A_n\}$, el teorema de Bayes se presenta así:

$$P(C|A_1, ...A_n) = \frac{P(C)P(A_1, ...A_n|C)}{P(A_1, ...A_n)} \tag{2.2}$$

Si C contiene k posibles valores $\{c_1, ..., c_k\}$, el valor que interesa identificar es el más plausible y entregarlo como resultado de la clasificación. En el contexto bayesiano, la hipótesis más plausible es la que posee mayor probabilidad a posteriori dados los atributos, y se conoce como *la hipótesis máxima a posteriori* o *hipótesis MAR*. De modo que la clase por asociar al elemento en cuestión, se obtiene mediante la Ecuación 2.3.

$$\begin{aligned} C_{MAP} &= argmax_{c\in\Omega_c} P(C|A_1,...,A_n) \\ &= argmax_{c\in\Omega_c} \frac{P(A_1,...,A_n|C)P(C)}{P(A_1,...,A_n)} \\ &= argmax_{c\in\Omega_c} P(A_1,...,A_n|C)P(C) \end{aligned} \tag{2.3}$$

En la Ecuación 2.3, el término Ω_C representa al conjunto de valores que puede tomar la variable C.

Por lo anterior, el teorema de Bayes facilita un método sencillo y con una semántica clara para resolver la tarea de clasificación, sin embargo presenta un problema muy importante, el cual es su alta complejidad computacional. Esto último se debe a que es necesario trabajar con distribuciones de probabilidad que involucran muchas variables, resultando en la mayoría de los casos intratables.

Estos problemas se minimizan haciendo uso de la suposición de independencia entre las variables que participan. Dicha suposición se considera el principio más importante del clasificador ingenuo de Bayes.

2.2.1. Clasificador ingenuo de Bayes

Se conoce al clasificador ingenuo de Bayes como el modelo más simple de clasificación bayesiano. En este caso la estructura es fija y solamente es necesario aprender los parámetros. Este clasificador se basa en dos premisas establecidas sobre las variables predictoras (atributos predictores) y la variable a predecir (atributo clase):

- Los posibles valores de los atributos clase son excluyentes, es decir la variable a predecir C toma sólo uno de sus posibles k valores $\{c_1, ..., c_k\}$.
- Los valores de los atributos predictores son condicionalmente independientes dado el valor del atributo clase, es decir, si se conoce el valor del atributo clase el conocimiento del valor de cualquiera de los atributos predictores es irrelevante para el resto de los atributos [16, 50].

  Esta condición se expresa matemáticamente por medio de la siguiente fórmula:

$$P(A_1, ...A_n|C) = \prod_{i=1}^{n} P(A_i|C) \tag{2.4}$$

La hipótesis de independencia asumida por el clasificador da lugar a un modelo gráfico probabilístico en el que existe un único nodo raíz representando la clase, y en la que todos los atributos son nodos hoja que tienen como único padre a la variable clase. Gráficamente se tiene una estructura semejante a la mostrada en la Figura 2.1.

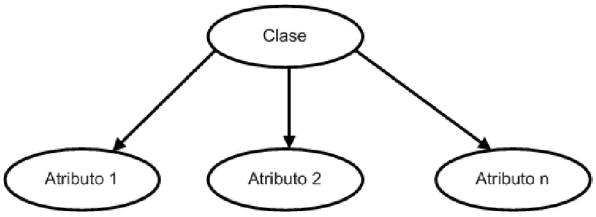


Figura 2.1: Topología de un clasificador ingenuo de bayes

Debido a la hipótesis de independencia utilizada en este clasificador, la expresión para obtener la hipótesis MAP se da en términos de la Ecuación 2.5.

$$\begin{aligned} C_{MAP} &= argmax_{c\in\Omega_c} P(A_1, ...A_n|c)P(c) \\ &= argmax_{c\in\Omega_c} P(c)\prod_{i=1}^{n} P(A_i|c) \end{aligned} \tag{2.5}$$

Por lo tanto, los parámetros que se deben estimar son $P(A_i|c)$ para cada atributo y la probabilidad a priori del atributo clase $P(c)$.

En el caso de que los k atributos predictores $\{c_1, ..., c_k\}$ sean continuos, el clasificador ingenuo de Bayes también puede aplicarse.En esta situación el problema se enfoca en encontrar el valor del atributo clase C que representa el de máxima probabilidad posteriori del atributo C dada la evidencia expresado como una instancia de los atributos $\{A_1, ...A_k\}$.

En este caso, la expresión para encontrar la hipótesis MAP se obtiene mediante la siguiente ecuación:

$$\begin{aligned} C_{MAP} &= argmax_{c\in\Omega_c} P(C|A_1, ...A_n) \\ &= argmax_{c\in\Omega_c} P(C)\prod_{i=1}^{n} f_{A_i|C}(A_i|C) \end{aligned} \tag{2.6}$$

En ella, el elemento $f_{A_i|C}(A_i|C)$ denota la función de densidad del atributo A_i, condicionada a que el valor del atributo clase sea C para todo elemento.

El modelo provisto por el clasificador bayesiano, hace que sea fácilmente aplicable a ciertos conjuntos de datos. Para encontrar la pertenencia a una clase específica se asume la independencia de los demás atributos y solamente hay que determinar las probabilidades sobre el número de elementos que pertenecen a la clase en cuestión.

Fases del clasificador ingenuo de Bayes

A grandes rasgos el núcleo del algoritmo del clasificador ingenuo de Bayes sobre un conjunto de datos, consiste de los siguientes pasos:

- Sea D el conjunto de elementos con sus respectivas etiquetas o valores de atributos clase. Cada elemento de n atributos es representado por un vector $X = (x_1, \ldots, x_n)$, representando n mediciones hechas al elemento de n atributos, respectivamente $A_1, \ldots, A_n$.

- Si existen m clases, $C_1, \ldots, C_m$. Dado un elemento o tupla, X, el clasificador va a predecir que esa tupla pertenece a la clase que posea una probabilidad posteriori mayor, condicionada con los valores de X para cada uno de sus atributos. Es decir, el clasificador determina que el elemento X pertenece a la clase C_i si y sólo si se cumple que

$$P(C_i|X) > P(C_j|X) \text{ para } i \leqslant j \leqslant m, j \neq i \tag{2.7}$$

  Es decir, la clase más probable al maximizar $P(C_i|X)$. Dicho valor esta dado por la ecuación del teorema de Bayes (Ecuación 2.1).

- Dado que $P(X)$ es constante para todas las clases, solamente es necesario maximizar el valor de $P(X|C_i)\ P(C_i)$. Si las probabilidades de cada clase no son conocidas, se asume que son equivalentes y solamente se busca maximizar el valor de $P(X|C_i)$. De otro modo, se maximiza el valor $P(X_i|C_i)\ P(C_i)$. Para estimar las probabilidades de las clases se utiliza la siguiente formula:

$$P(C_i) = \frac{|C_{i,D}|}{|D|} \tag{2.8}$$

  donde $|C_{i,D}|$ representa el número de elementos pertenecientes a la clase C_i en D (conjunto de entrenamiento).

- A continuación se hace utilización de la suposición de independencia condicional de la clase, ello con la finalidad de disminuir el costo computacional al calcular cada $P(X|C_i)$. Este cálculo se reduce a lo siguiente:

$$\begin{aligned} P(X|C_i) &= \prod_{k=1}^{n} P(x_k|C_i) \\ &= P(x_1|C_i)\ \cdot\ P(x_2|C_i)\ \cdot \ldots\ \cdot P(x_n|C_i) \end{aligned} \tag{2.9}$$

  De este modo, el cálculo de las probabilidades $P(x_1|C_i)\ \cdot\ P(x_2|C_i)\ \cdot \ldots \cdot P(x_n|C_i)$ se obtiene a partir de los elementos del conjunto de entrenamiento[4] . Este cálculo se lleva a cabo dependiendo del tipo de dominio del atributo en cuestión. Por lo cual se dan dos casos.

[4] Cada elemento x_i representa el valor del atributo A_k para el elemento X

1. Si el atributo es discreto, entonces $P(x_k|C_i)$ es el número de elementos de la clase C_i en D que poseen el valor de x_k para el atributo, dividido por $|C_{i,D}|$ (que representa el total de tuplas con la etiqueta C_i en D).
2. Por el contrario, si se trata de un atributo con dominio continuo, entonces el cálculo se basa en otra suposición. Dicha suposición consiste en asumir que los valores del atributo poseen una distribución normal con una media μ y una desviación estándar σ definida por la siguiente Ecuación.

$$g(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{2.10}$$

por lo tanto el cálculo de la probabilidad asociada se resuelve de la siguiente manera:

$$P(x_k|C_i) = g(x_k, \mu_{C_i}, \sigma_{C_i}) \tag{2.11}$$

Evidentemente, esta estimación tiene el inconveniente de que los datos no siempre siguen una distribución normal.

- El último paso consiste en determinar la etiqueta que será asignada al elemento X. Para ello se evalúa $P(x|C_i)\ P(C_i)$ para cada clase C_i. El clasificador predice que la clase del elemento es C_i solamente si se cumple

$$P(X|C_i)\ P(C_i) > P(X|C_j)\ P(C_j)\ para\ 1 \leqslant j \leqslant m, j \neq i \tag{2.12}$$

Es decir, se asocia la clase que posea el máximo valor en $P(X|C_i)\ P(C_i)$

Consideraciones especiales

Aún cuando los cálculos que se realizan con el clasificador ingenuo de Bayes para obtener los valores de los atributos clase son sencillos, en ocasiones se puede presentar dificultad en la obtención de éstos. Por ejemplo, la Ecuación 2.9 refiere que para la estimación de $P(X|C_i)$ se debe calcular el producto de las probabilidades[5] $P(x_1|C_1), P(x_2|C_2), \ldots, P(x_n|C_i)$. Dichas probabilidades se obtienen a partir de conjunto de datos de entrenamiento.

También es necesario obtener $P(X|C_i)$ para cada clase posible $(i = 1, 2, \ldots, m)$ con la idea de encontrar aquella clase C_i para la cual se cumple que su $P(X|C_i)\ P(C_i)$ es la mayor. Sin embargo, puede presentarse el caso de que por lo menos una $P(x_i|C_i)$ sea equivalente a 0. Lo cual, genera de manera directa que $P(X|C_i) = 0$.

Situación que no siempre refleja una realidad válida, ya que puede suscitarse el caso en el cual al no tomar en cuenta esa probabilidad de 0, se obtenga otra $P(X|C_i)$ indicando que el elemento X si pertenece a la clase C_i. El problema se reduce a observar

[5] Se considera la independencia condicional de la clase.

que una probabilidad de 0 cancela los efectos de las restantes probabilidades a posteriori involucradas en el producto (para la clase C_i).

La solución que se plantea [33] es asumir que el conjunto de datos de entrenamiento es potencialmente grande y agregar por lo menos un valor que pertenezca a cada clase C_i, lo cual no afecta la proporción que se mantiene entre todas las probabilidades para cada clase, de modo que todas éstas se calculan nuevamente y con ello se evita la aparición de una probabilidad equivalente a 0. Esta técnica para la estimación de probabilidades se conoce como **"Corrección Laplaciana"** o **"Estimador Laplaciano"**. Las estimaciones modificadas se comportan de modo muy similar a sus semejantes y evitan la aparición de probabilidades iguales a 0.

2.2.2. Clasificador semi-ingenuo de Bayes

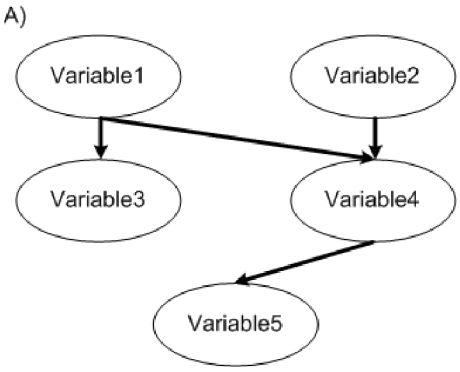
En [46] se presenta el clasificador semi-ingenuo de Bayes. En el cual se trata de evitar las estrictas premisas sobre las que se construye el paradigma ingenuo de Bayes por medio de la consideración de nuevas variables en las cuales no necesariamente tenga que aparecer el producto cartesiano de dos variables, sino tan sólo aquellos valores de dicho producto cartesiano que cumplan una determinada condición que surge al considerar el concepto de independencia junto con el de confiabilidad en la estimación de las probabilidades condicionadas.

Sin embargo existe el compromiso de sacrificar la "no ingenuidad" por la confiabilidad en la aproximación de las probabilidades. En [46] también se muestra un algoritmo que optimiza esta compensación entre confiabilidad e independencia, permitiendo detectar las dependencias entre los valores de los atributos. Este clasificador presenta beneficios al no considerar la independencia de atributos y conservar confiabilidad en las probabilidades obtenidas. Empero, el aprendizaje no es tan rápido como lo es en el caso del clasificador ingenuo de Bayes, además de no ser incremental.

2.2.3. Redes bayesianas

Las redes bayesianas o redes probabilísticas son una representación gráfica de dependencias para razonamiento probabilístico en sistemas expertos, en la cual los nodos representan variables proposicionales y los arcos la dependencia probabilística. Gracias a su elemento teórico de actualización de probabilidades, el Teorema de Bayes, las redes bayesianas son una herramienta extremadamente útil en la estimación de probabilidades ante nuevas evidencias, demostrando su potencialidad como modelos de representación del conocimiento con incertidumbre.

Formalmente se define una red bayesiana como una gráfica acíclica dirigida $G = (V, E)$ en la cual cada nodo (V_i) representa una variable y cada arco una dependencia probabilística $(E \subset V \times V)$, dicho arco especifica la probabilidad condicional de cada variable dados sus nodos antecesores, así cada nodo (V_i) tiene asociada una distribución de probabilidad condicionada $P(V_i|padres(V_i))$ que cuantifica el efecto o influencia de



B)

	Variable1 = 1 Variable2 = 1	Variable1 = 1 Variable2 = 0	Variable1 = 0 Variable2 = 1	Variable1 = 0 Variable2 = 0
Variable4 = 1	0.8	0.5	0.7	0.9
Variable4 = 0	0.2	0.5	0.3	0.1

Figura 2.2: Una red bayesiana sencilla: A) Representación de la gráfica acíclica dirigida. B) La tabla de probabilidades condicionadas para los valores de la variable `Variable4` mostrando cada combinación posible de los valores de sus nodos padre, `Variable1` y `Variable2`.

sus nodos padres. Esta información se almacena en varias tablas, cada una asociada a una variable y se conocen como tabla de probabilidades condicionadas. Un ejemplo de ello se presenta en la Figura 2.2, donde la tabla para la variable $Variable4$ especifica la distribución condicional $P(Variable4|padres(Variable4))$ (aquí $padres(Variable4)$ representan los nodos padres de $Variable4$, siendo estos: $Variable2$ y $Variable1$).

Dentro del contexto de minería de datos, se considera a $X = x_1, \ldots, x_n$ una tupla descrita por las variables o atributos $Y_1, \ldots, Y_n$, respectivamente. De modo que esta asociación permite que la red brinde una completa representación de las distribuciones de probabilidad conjuntas existentes. Dicha representación queda de manifiesto en la Ecuación 2.13.

$$P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i|padres(V_i)) \tag{2.13}$$

En la Ecuación 2.13, $P(x_1, \ldots, x_n)$ denota la probabilidad de que una combinación particular de los valores de X, y los valores de $P(x_i|padres(V_i))$ correspondan a las entradas en la tabla de probabilidades condicionadas para la variable V_i.

La estructura de la red especifica las relaciones de independencia condicional que se tienen en el dominio. El significado general de un arco del nodo V_i hacia el nodo V_j en la gráfica es que el nodo V_i tiene una influencia directa sobre el nodo V_j. Una vez que la topología de la red bayesiana está diseñada es necesario especificar una distribución de probabilidad condicional para cada variable dados sus padres. Las redes bayesianas no sólo modelan de forma cualitativa el conocimiento sino que además expresan de manera cuantitativa las relaciones entre las distintas variables que intervienen.

Con lo anterior, es posible interpretar a una red bayesiana de dos formas:

1. Distribución de probabilidad: Representa la distribución de la probabilidad conjunta de las variables representadas en la red.

2. Base de reglas: Cada arco representa un conjunto de reglas que asocian las variables involucradas[6] .

Los dos interpretaciones son equivalentes, sin embargo el primero punto resulta útil para entender como se construyen las redes, mientras que el segundo punto es más práctico para el diseño de procedimientos de inferencia.

2.2.4. Las redes bayesianas y minería de datos

Las redes bayesianas se consideran una alternativa para minería de datos, ya que presentan las siguientes ventajas:

- Permiten aprender sobre relaciones de dependencia y causalidad.
- Permiten combinar conocimiento con datos.
- Evitan el sobre entrenamiento de los datos.
- Pueden manejar bases de datos incompletas.

El obtener una red bayesiana a partir de datos es un proceso de aprendizaje, que se divide en dos aspectos:

1. Aprendizaje paramétrico: Donde dada una estructura, se deben obtener las probabilidades a priori y probabilidades condicionales requeridas.

2. Aprendizaje estructural: Donde es necesario obtener la estructura de la red Bayesiana, es decir, las relaciones de dependencia e independencia entre las variables involucradas.

Las técnicas de aprendizaje estructural dependen del tipo de estructura de red que se utilice ya sean árboles, poli-árboles o redes multi-conectadas. Otra alternativa es combinar conocimiento subjetivo del experto con aprendizaje. Para ello se parte de la estructura dada por el experto, la cual se valida y mejora utilizando datos estadísticos.

[6] Una codificación de un conjunto de afirmaciones de independencia condicional.

Aprendizaje paramétrico

El aprendizaje paramétrico consiste en encontrar los parámetros asociados a una estructura dada de una red bayesiana. Dichos parámetros consisten en las probabilidades a priori de los nodos raíz y las probabilidades condicionales de las demás variables, dados sus padres.

Si se conocen todas las variables, es fácil obtener las probabilidades requeridas. Las probabilidades previas corresponden a las marginales de los nodos raíz, y las condicionales se obtienen de las conjuntas de cada nodo con su padre. Para que se actualicen las probabilidades con cada caso observado, éstas se pueden representar como razones enteras, y actualizarse con cada observación.

Aprendizaje estructural

El clasificador ingenuo de Bayes asume independencia entre los atributos dada la clase y su estructura ya está dada, por lo que solamente se tienen que aprender las probabilidades de los valores de los atributos dada la clase. Una forma de mejorar la estructura de este clasificador consiste en añadir entre los nodos o atributos que tengan cierta dependencia. Existen dos estructuras básicas:

- TAN: clasificador bayesiano simple aumentado con un árbol.
- BAN: clasificador bayesiano simple aumentado con una red.

Otra forma es realizar alguna de las siguientes operaciones locales hasta que no mejore la precisión:

1. Eliminar un atributo
2. Unir dos atributos en una nueva variable combinada
3. Introducir un nuevo atributo que haga que dos atributos dependientes sean independientes (nodo oculto).

2.3. Recapitulación

Los clasificadores Bayesianos son clasificadores estadísticos, que pueden predecir tanto las probabilidades del número de miembros de clase, así como la probabilidad de que un nuevo elemento pertenezca a una clase en particular. La clasificación Bayesiana se fundamenta en el teorema de Bayes (Ecuación 2.1) y dichos clasificadores han demostrado una precisión y eficiencia comparables con otras técnicas de clasificación.

Una ventaja de este clasificador es su manejo de valores perdidos o desconocidos: en el clasificador ingenuo de Bayes si se intenta clasificar un elemento con un atributo sin valor, simplemente el atributo en cuestión no entra en el cálculo del producto que

sirve para obtener las probabilidades. Sin embargo, la proporción de elementos que se ven afectados por este tipo de error (ausencia de valor) y su influencia en la precisión de los modelos de clasificación basados en dicho clasificador, no ha sido ponderada.

Por otro lado, la presencia de errores fuera del rango de valores válidos constituye un problema presente en mayor proporción. En este caso, el clasificador ingenuo de Bayes se ve afectado al incluir nuevos elementos o valores dentro del cálculo de probabilidades y con ello, modificar su comportamiento dentro de la etapa de entrenamiento y modificar los criterios para asignar las clases de los nuevos elementos.

Razones por las cuales, se ha decidido analizar el comportamiento de este clasificador en presencia de distintos tipos, porcentajes y distribución de valores [7] en errores sobre los datos que se le presentan como elementos entrenamiento.

[7] En caso de que aplique.

CAPÍTULO 3

Árboles de decisión

3.1. Introducción

El presente capítulo trata una técnica para el aprendizaje de modelos comprensibles y proposicionales: los árboles de decisión. Mediante esta técnica se construye un modelo, hipótesis o representación de la regularidad presente en los datos. Además se hace referencia al hecho de que estos modelos son factibles de expresar de una forma simbólica, mediante un conjunto de condiciones y por tanto, presentan un modelo inteligible para los seres humanos. Debido a que los algoritmos desarrollados en estas técnicas restringen el aprendizaje sobre una única entrada de datos y no establecen relaciones entre más de un elemento a la vez (generalmente por la estructura de datos básica que utilizan, los árboles) ni sobre más de un atributo o característica de manera simultánea, se enmarcan como modelos proposicionales.

El uso de árboles de decisión tuvo su origen en las ciencias sociales con los trabajos de Sonquist y Morgan llevados a cabo en el Survey Research Center del "*Institute for Social Research*" de la Universidad de Michigan [42]. Una de las primeras implementaciones de métodos de ajuste de los datos basados en árboles de clasificación fue el programa AID (*Automatic Interaction Detection*), de Sonquist, Baker y Morgan [43]. Sin embargo, se conoce como el primer sistema que construía árboles de decisión al CLS (*Concept Learning System*) de Hunt [40], desarrollado en 1959 y depurado durante la década de los años sesenta. La principal aportación de este trabajo fue la propia metodología sin embargo no resultaba computacionalmente eficiente debido al método que empleaba en la extensión de los nodos[1] .

De manera casi simultánea, en el área de estadística se desarrolló un algoritmo recur-

[1] Se guiaba por una estrategia similar al *minimax* con una función que integraba diferentes costos.

sivo de clasificación no binario, denominado CHAID (*CHi-square Automatic Interaction Detection*) [78]. Posteriormente se introdujo un nuevo algoritmo para la construcción de árboles y su aplicación en problemas de regresión y clasificación [48], este método es conocido como CART (*Classification And Regression Trees*).

Casi al mismo tiempo el proceso de inducción mediante árboles de decisión comenzó a ser utilizado por la comunidad de aprendizaje automático [54, 61] y la comunidad de reconocimiento de patrones [38]. En 1979 se desarrolló el sistema ID3 ("*Iterative Dichotomizer (version) 3*") [65] que conceptualmente es fiel a la metodología de CLS pero le aventaja en el método de expansión de los nodos (basado en una función que utiliza la medida de la información de Shannon). Empero esta versión contenía diversos problemas (Sección 3.3.3), por ello se desarrolló una versión mejorada. Dicha versión se conoce como el sistema C4.5 [62].

La construcción de árboles de decisión, también denominados árboles de clasificación o de identificación, es el método de aprendizaje automático más utilizado. La tarea de aprendizaje para la cual los árboles de decisión se adecuan mejor, debido a su estructura, es la clasificación.

La tarea de clasificación posee diversas características, siendo una de las más importantes asumir que las clases son disjuntas, es decir, si para cierto problema se identifican 2 clases (C_1, C_2), un elemento es de la clase C_1 ó de la clase C_2, pero no puede pertenecer a ambas clases de manera simultánea. El dominio de aplicación de los árboles de decisión no está restringido a un ámbito concreto sino que pueden ser utilizados en diversas áreas. Algunas de las áreas donde han sido ampliamente utilizados son aplicaciones de diagnóstico médico, juegos como el ajedrez o sistemas de predicción meteorológica. Empero los árboles de decisión se adaptan especialmente bien a aquellos problemas que presentan las siguientes características:

- Los elementos por clasificar pueden ser descritos mediante vectores de pares atributo-valor.
- La clase objetivo toma valores discretos.
- Se considera la existencia de ruido en el conjunto de entrenamiento.
- Los valores de algunos atributos en los ejemplos del conjunto de entrenamiento pueden ser desconocidos.

El conocimiento obtenido en el proceso de aprendizaje se representa mediante una estructura arborescente. En ella cada nodo interior contiene una pregunta o evaluación sobre un atributo en particular. Sobre la respuesta a esta evaluación se basa la división del dicho nodo (con un hijo por cada posible respuesta). En el caso de los nodos hoja, cada una de ellas se refiere a una decisión o clasificación, siendo el valor de la variable de respuesta cuando se trata de árboles de regresión o el nombre de la clase a la cual pertenece en el caso de los árboles de clasificación.

Un árbol de decisión puede utilizarse para clasificar un nuevo elemento al comenzar un recorrido desde su raíz y siguiendo el camino determinado por las respuestas a la evaluación de cada uno de los nodos internos que se encuentren hasta que alcanzar una hoja del árbol. En ese momento, se le asigna la etiqueta correspondiente a la hoja alcanzada.

La construcción de los árboles de decisión se hace recursivamente de forma descendente (se parte de conceptos generales que se van especificando conforme se desciende en el árbol), por lo que se emplea el acrónimo TDIDT ("*Top-Down Induction on Decision Trees*") para referirse a la familia completa de algoritmos de este tipo. La familia de algoritmos TDIDT abarca desde algoritmos clásicos de la inteligencia artificial como CLS, ID3, C4.5 o CART hasta algoritmos optimizados como SLIQ o SPRINT[2] .

Una suposición inicial de los algoritmos TDIDT, es que no existe datos con errores en el conjunto de datos de entrada e intentan alcanzar una descripción perfecta de los mismos. Sin embargo, como se ha expuesto anteriormente, esto suele ser contraproducente en problemas reales donde se necesitan métodos capaces de manejar información con ruido y mecanismos que eviten el fenómeno del sobre entrenamiento. Para solventar estas deficiencias en el manejo de errores se han desarrollado técnicas de poda (como las empleadas en ASSISTANT o C4.5) que son útiles en este sentido. Ya que al obtener un árbol de decisión completo que se adapta perfectamente a los datos del conjunto de entrenamiento, se podan aquellas ramas del árbol con menor capacidad predictiva (Sección 3.2.2).

Por lo anterior es posible afirmar que la representación del conocimiento mediante árboles de decisión es bastante simple y, a pesar de carecer de la expresividad de las redes semánticas o de la lógica de primer orden, se utiliza muy a menudo para resolver problemas de clasificación de todo tipo.

3.1.1. Clasificación mediante inducción en árboles de decisión

La inducción en los árboles de decisión permite la representación de aprendizaje de éstos, a partir de un conjunto de elementos etiquetados en su clase. Un árbol de decisión es similar a un árbol con un flujo determinado, donde cada nodo interno (nodos que no son hoja) denota una evaluación sobre un atributo en particular, cada división representa un resultado posible de la evaluación y cada hoja o nodo terminal contiene la etiqueta de una clase. Un árbol de decisión sencillo se presenta en la Figura 3.1. Algunos algoritmos de árboles de decisión generan solamente árboles binarios, mientras que otros producen árboles con múltiples ramificaciones en sus nodos internos y hojas.

El funcionamiento general de un árbol de decisión resulta ser sencillo. Dentro del contexto de bases de datos, dada un elemento o tupla X para la cual no se tiene asociada una etiqueta en el atributo clase, cada uno de sus atributos es evaluado contra cada una de las ramificaciones del árbol de decisión en cuestión. De este modo, se traza una

[2] Dos algoritmos desarrollados en el IBM *Almaden Research Center* que se usan en minería de datos.

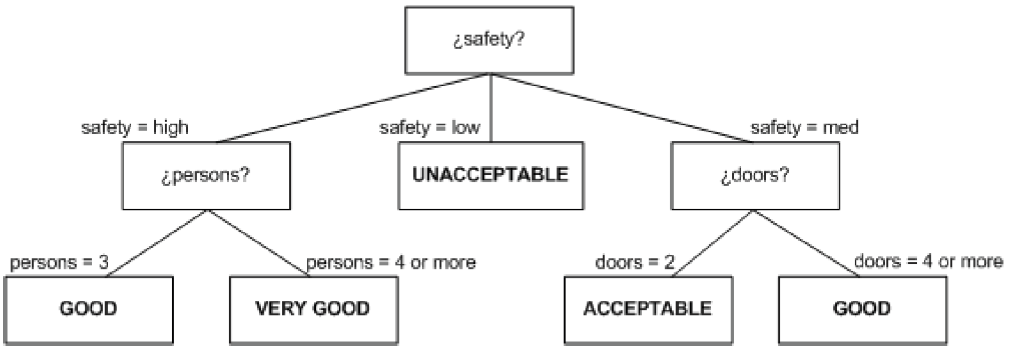


Figura 3.1: Árbol de decisión para el atributo clase "aceptación", indicando que tan aceptable es un automóvil de acuerdo a una estructura ideal. Cada nodo interno representa una evaluación sobre determinado atributo y cada hoja del árbol representa una clase.

ruta imaginaria a partir de la raíz hacia algún nodo hoja, mismo que posee la clase predicha para esa tupla X. Una facilidad que brindan los árboles de decisión es que sus resultados pueden convertirse de modo sencillo en reglas de clasificación.

La construcción de los árboles de decisión no requiere conocimiento del contexto o dominio, ni ajustes de parámetros por lo tanto es una técnica apropiada para el descubrimiento exploratorio de conocimiento. Así, las principales ventajas que presentan los árboles de decisión son las siguientes [33]:

- Manejo de datos con una alta dimensión (múltiples atributos).
- La representación del conocimiento adquirido en forma de árbol es intuitiva y generalmente fácil de asimilar. Se pueden desplegar de mayor a menor detalle (dependiendo del nivel de profundidad).
- Las fases de aprendizaje y clasificación de la inducción en ellos, son simples y rápidas.
- En general poseen una precisión aceptable.
- Son eficientes y existen variantes escalables a grandes volúmenes de datos.
- Tratan con atributos con dominios continuos o discretos.
- Son tolerantes al ruido, a atributos no significativos y a valores faltantes.

No obstante los árboles de decisión también presentan ciertas desventajas. En general no son tan precisos como otros métodos por ejemplo las redes neuronales o las máquinas de soporte vectorial (también conocidas como máquinas de vectores de soporte) [39]. Además se les enmarca dentro de la categoría de los método de aprendizaje

débil (*weak learners*[3]), ya que debido a su carácter voraz, dependen de la muestra de elementos. Es decir, para dos muestras distintas sobre el mismo conjunto de datos es factible que se generen dos árboles bastante diferentes. No obstante esta debilidad es la que permite que los árboles de decisión sean frecuentemente utilizados con técnicas de combinación como el aumento y el consenso (Sección 1.7.2). Los árboles de decisión han sido utilizados en múltiples áreas tales como la medicina [13], manofactura y producción [55], análisis financiero, astronomía y biología molecular [83]. Además, de ser la base de múltiples productos comerciales de inducción de reglas.

3.2. Generación de árboles de decisión

Los algoritmos para la generación de un árbol de decisión contemplan múltiples etapas, cada uno de ellos utiliza distintas técnicas para lograr una estructura que permita una mejor clasificación, sin embargo la estructura general del algoritmo básico, contempla los siguientes elementos y acciones.

- Sea D el conjunto de datos por utilizar. Inicialmente este representa al conjunto de datos de entrenamiento (incluyendo a sus etiquetas asociadas para cada elemento). Existen dos parámetros muy importantes, el primero se conoce como la "*lista_de_atributos*". Dicho parámetro denota una lista de atributos que describe a cada elemento. El segundo parámetro, "*método_de_selección_de_atributos*" especifica alguna heurística para seleccionar al atributo que mejor divide los elementos de acuerdo a las distintas clases. Dicho método emplea algún criterio de selección de atributos (Sección 3.2.1), tal como la ganancia de información o el índice Gini. Estos criterios permiten determinar, entre otras cosas, que el árbol generado sea binario o no.

- En el inicio de la construcción del árbol, el nodo raíz N contiene la totalidad de los elementos del conjunto de datos de entrenamiento (D).

- Si los elementos en D son de la misma clase, entonces el nodo N se convierte en una hoja y es etiquetado con esa clase.

- En caso contrario, se invoca la ejecución del método de selección de atributos para determinar el criterio de división. Dicho criterio indica que atributos se deben evaluar en el nodo N para determinar la mejor forma de separar o dividir los elementos contenidos en D dentro de clases individuales. Además, indica que

[3] Se les conoce a los métodos de aprendizaje débil a aquellos métodos de clasificación los cuales se encuentran ligeramente relacionados con una clasificación real. Por el contrario, los métodos de clasificación fuertes son métodos que se encuentran arbitrariamente bien correlacionados con la tarea de clasificación. Varios métodos de aprendizaje débil pueden generar un método de aprendizaje fuerte, utilizando elementos como el aumento o consenso (Sección 1.7.2).

ramificaciones se crearán a partir del nodo N con respecto a los resultados de las evaluaciones elegidas.

La finalidad de este criterio es determinar las particiones resultantes en cada ramificación y que éstas sean lo más puras posibles (es decir contengan el menor número de clases posibles). En otras palabras, si se dividen los elementos contenidos en D de acuerdo a resultados de evaluaciones mutuamente exclusivos, se espera que las divisiones sean puras. El hecho de que las divisiones sean puras permite que los elementos sean más fáciles de identificar en la fase de prueba, debido a que un elemento recorrerá un camino menor del nodo raíz a un nodo hoja donde obtendrá su etiqueta o clase.

- De este modo, el nodo en cuestión (N) es etiquetado con el criterio de división, que funcionará como una evaluación en ese nodo. Además, se crea una ramificación del nodo N hacia cada posible resultado de la evaluación del criterio de división. Los elementos en D son divididos acorde a esta evaluación. Se presentan 3 posibles escenarios. Sea A el atributo de división con v valores distintos[4] , $a_1, a_2, \ldots, a_v$:

  1. Si A posee un dominio discreto, entonces los resultados de las evaluaciones en el nodo N corresponden directamente a los valores de A. Una ramificación se crea para el valor conocido, a_j, de A y se etiqueta con ese valor. Se crean las divisiones D_j, que posee el subconjunto de elementos de D etiquetados con el valor de a_j en el atributo A. Dado que todos los elementos en una división tienen el mismo valor para el atributo A, entonces A no debe ser considerado en futuras divisiones de los elementos, por lo tanto se remueve de la lista de atributos ("*lista_de_atributos*") elegibles.
  2. En el caso de que el dominio del atributo A sea continuo, los dos resultados de las evaluaciones en el nodo N corresponden a las condiciones $A \leq$ *punto_division* y $A >$ *punto_division*, donde "*punto_division*" corresponde al valor obtenido del método[5] "*método_de_selección_de_atributos*". Así, dos ramificaciones crecen a partir del nodo N y se etiquetan de acuerdo a los resultados anteriores. Los elementos quedan divididos de tal forma que D_1 contiene aquellos elementos de D tales que para el atributo A se cumple que $A \leq$ *punto_division*, mientras que D_2 contiene el resto de los elementos.
  3. Si el atributo A posee un dominio que es discreto y además se estipula producir un árbol binario[6] , entonces se realiza una evaluación de pertenencia en el nodo N (tipo "$A \in S_A$"). Así, S_A se considera el subconjunto de división para A, cuyos valores son un subconjunto de los valores de A. Si un elemento posee el valor de a_j en el atributo A y además $a_j \in S_A$, entonces la evaluación

[4] Basados en los valores de los datos de entrenamiento.

[5] Generalmente y debido a los criterios de selección, este valor no se encuentra dentro de los valores existentes del atributo A.

[6] Condición dictada ya sea por el algoritmo utilizado o el criterio de selección de atributos.

dada en el nodo N se cumple. Dos ramificaciones se crean a partir del nodo N. Por convención, la ramificación izquierda se etiqueta con la leyenda *si*, de modo que el subconjunto D_1 corresponde a los elementos etiquetados de D tales que cumplen con la evaluación. Por su parte la ramificación derecha del nodo N se etiqueta con la leyenda *no*, y similarmente D_2 contiene el subconjunto de elementos etiquetados de D para los cuales no se satisface la evaluación.

- El algoritmo utiliza el mismo proceso recursivamente para generar un árbol de decisión para los elementos resultantes de cada división de D (D_j). Dicha división recursiva se detiene únicamente cuando se presenta alguna de las siguientes condiciones:

  1. Todos los elementos en la división D (representados en el nodo N) pertenecen a la misma clase.
  2. No existen atributos mediante los cuales pueda llevarse a cabo alguna división sucesiva. En este caso, se lleva a cabo una votación en la cual se determina convertir un nodo N en una hoja y etiquetarlo con la clase más común para los elementos de D. Alternativamente se pueden almacenar las distribuciones de cada clase.
  3. No existen elementos para alguna ramificación (alguna división D_j) tal que sea vacía. Aquí, una hoja es creada con la etiqueta correspondiente a la clase mayoritaria en D.

- Al finalizar estas etapas se obtiene un árbol de decisión listo para ser utilizado directamente con el conjunto de datos de prueba, o bien ser objeto de un procesamiento posterior denominado poda[7] .

La complejidad computacional del algoritmo, dado un conjunto de datos de entrenamiento D es $O(n \cdot |D| \cdot log(|D|))$, donde n es el numero de atributos que describen los elementos en D y $|D|$ es el total de elementos de D (la cardinalidad). En este caso el término $n \cdot |D|$ refiere a una lectura sobre los elementos del conjunto de datos de entrenamiento en D para todos los atributos y para cada nivel del árbol (término $log(|D|)$). Lo cual puede significar grandes tiempos en la fase de entrenamiento y falta de memoria cuando se manejan grandes bases de datos.

Distintas versiones incrementales de inducción para árboles de decisión han sido propuestas [11, 29, 69, 67]. Algunas de estas mejoras incluyen acciones que reconstruyen la estructura creada inicialmente al obtener nuevos conjuntos de entrenamiento al volver a evaluar el árbol creado anteriormente [77].

[7] Este procesamiento de poda también puede ser llevado a cabo durante la construcción del árbol (Sección 3.2.2)

Como se mencionó anteriormente, los distintos algoritmos implementan mejoras utilizando ya sea con estructuras de datos auxiliares o bien, variando los criterios de elección de los atributos al crear el árbol (Sección 3.2.1) y los mecanismos utilizados para la poda (Sección 3.2.2).

3.2.1. Elección de atributos de división

Una de las etapas que realizan los algoritmos utilizados para la construcción de árboles de decisión es determinar el atributo que servirá como elemento de división de datos, es decir, aquel atributo contra el cual serán evaluados los demás elementos del conjunto de datos de entrenamiento en primera instancia. Y posteriormente los elementos del conjunto de datos de prueba. Los criterios para la selección de atributos consisten en heurísticas que permiten elegir el atributo más adecuado de separación que "mejor divide" una partición de datos de un conjunto de elementos de entrenamiento etiquetados, en clases individuales. Un criterio efectivo ayuda al algoritmo en cuestión a construir hipótesis con una precisión mayor.

La idea general que manejan estos criterios de selección de atributos es determinar para cada nodo interno, aquel atributo tal que dependiendo de la evaluación de los elementos del conjunto de datos sobre él, todos los elementos que caen dentro de una partición dada pertenecerán a la misma clase. En caso de que se presente este resultado, se habla de una división pura, es decir una división que genera nuevos nodos con distribuciones homogéneas para las distintas clases. Por ejemplo en la Figura 3.2 para un conjunto de datos con 2 clases posibles se prefiere la segunda división de datos al separar de manera más precisa el conjunto de datos.

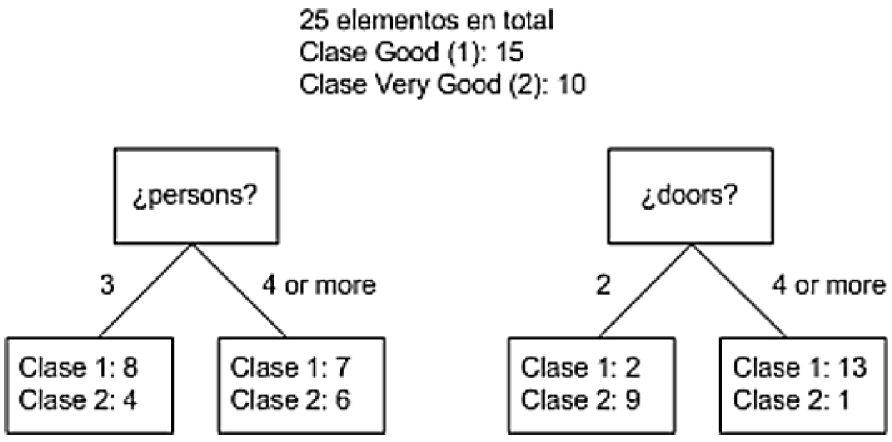


Figura 3.2: Ejemplo de distintas particiones con diferentes grados de pureza. Se prefiere la división con la evaluación sobre el atributo "doors", debido a que existe una distribución más homogénea.

Conceptualmente, el "mejor criterio de división" es aquel que se acerca más a este escenario. Los criterios para la selección de atributos también se les conoce como "reglas

de partición" o "reglas de división", dado que determinan como es que los elementos son divididos en determinado nodo del árbol. Además, la elección del mejor criterio de división resulta en un árbol de decisión más eficiente, dado que se minimiza el número de evaluaciones necesarias para etiquetar a un nuevo elemento.

Los distintos criterios empleados en la elección de atributos para la segmentación de los elementos se consideran voraces[8] debido a que realizan una optimización local al asignar una calificación para cada uno de los atributos analizados, dicha puntuación se genera a partir del conjunto de datos de entrenamiento. Así, el atributo que posee la mejor calificación para una determinada evaluación se escoge como el atributo de división para los elementos dados. Es decir si cada elemento de D esta compuesto por n atributos ($\{A_1, A_2, \ldots, A_n\}$) y se cumple que $cal(A_i) > cal(A_j)$ para $i \neq j$ y $j = 1, \ldots n$ (donde $cal(A_i)$ denota la calificación asignada por un criterio de selección) entonces el nodo analizado en cuestión es etiquetado con la evaluación sobre el atributo A_i[9] .

Posteriormente, se generan las ramificaciones para cada valor resultante de la evaluación del criterio y los elementos son divididos acorde a las particiones determinadas.

A continuación se presentan tres criterios para la selección de atributos y sus principales características, los cuales conforman un conjunto básico. Sin embargo aún cuando presentan ciertas deficiencias han mostrado resultados considerablemente buenos en la práctica [33].

1. Ganancia de información[10] .
2. Rango de ganancia[11] .
3. Índice Gini[12] .

La notación utilizada en las secciones que a continuación se presentan, son las siguientes. Sea D el conjunto de los datos por dividir, es decir el conjunto de datos de entrenamiento, mismo que poseen una etiqueta o clase asociada para cada uno de ellos. Suponiendo que el atributo por el cual se va a identificar la clase tiene m valores distintos, designando así m clases diferentes $C_i (i = 1, \ldots, m)$. Sea $C_{i,D}$ el conjunto de elementos de la clase C_i en D, y sea $|D|$ y $|C_{i,D}|$ el número de elementos en D y $|C_{i,D}|$ respectivamente.

Ganancia de información

El criterio de ganancia de información es el más popular, fue definido inicialmente por Quinlan para el algoritmo ID3 [65]. El criterio se basa en el trabajo de Claude

[8] *Greedy.*
[9] Esta evaluación que se genera depende del dominio del atributo (Sección 3.2) y del tipo de división que se requiera.
[10] Information gain.
[11] Gain ratio.
[12] Gini index.

Shannon sobre la teoría de la información [70], la cual estudia el valor o "contenido de información" de los mensajes. La idea general es elegir al atributo con la mayor ganancia de información. La ganancia de información se define como "la diferencia entre el requerimiento original de información y el nuevo requerimiento" [70]. Dentro del contexto de los árboles de decisión y particularmente para este criterio, se entiende como la capacidad de representar información con un número menor de elementos a cada sucesión de evaluaciones, es decir, distinguir una mayor cantidad de elementos para cada una de las clases a las cuales pertenecen los elementos en un nodo dado.

Así, dado un conjunto de elementos D, la calidad de cada atributo es el promedio de las entropías de los nodos que se producen por esta división multiplicado por -1 (se define: $informacion = -entropia$). Como se mencionó en la Sección 3.2 para cada atributo se calcula este criterio y se elige el mejor. Es decir, para seleccionar un atributo, éste debe brindar la mayor ganancia de información posible para producir una situación en donde la entropía definida para los siguientes nodos sea lo más pequeña posible (ya que si se obtiene una entropía igual a 0, entonces se puede clasificar efectivamente a un elemento). Tal aproximación minimiza el número de evaluaciones necesarias para clasificar un nuevo elemento y además, garantiza que se encuentre un árbol sencillo, aunque al ser una heurística, no sea siempre el más sencillo.

La información esperada que se necesita para clasificar un elemento del conjunto D está dada por la siguiente ecuación:

$$Info(D) = -\sum_{i=1}^{m} P_i log_2(P_i) \tag{3.1}$$

Esta es la ecuación que define al elemento $Info(D)$ conocido también como la **entropía** de D. Dentro del contexto de clasificación el elemento P_i denota la probabilidad de que un elemento arbitrario en D pertenezca a la clase C_i y se calcula mediante $|C_{i,D}|/|D|$. El logaritmo en base 2 se utiliza dado que la información se codifica en bits. Así, la entropía es el promedio de la información que se necesita en determinado nodo para identificar la etiqueta del atributo clase de un elemento de D.

Para dividir los elementos de D tomando en cuenta algún atributo A_i cuya ganancia de información ha sido máxima entre todos los restantes atributos y que tiene v valores distintos $a_i, a_2, \ldots, a_v$, se prosigue de la siguiente manera. Si el atributo A_i posee un dominio discreto, estos valores corresponden directamente a las v ramificaciones resultantes de la evaluación en el atributo A. Así, este atributo puede utilizarse para dividir D dentro de v particiones o subconjuntos, $D_1, D_2, \ldots, D_v$, donde D_j contiene aquellos elementos de D que tienen el valor de a_j en el atributo A_i. Estas particiones o subdivisiones corresponderán a las ramificaciones que crecen a partir del nodo N.

Sin embargo, estas particiones generalmente contendrán una colección de elementos de distintas clases, en lugar de pertenecer a una sola clase. La cantidad de información necesaria para obtener una clasificación exacta, se estima mediante la siguiente fórmula:

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \cdot Info(D_j) \tag{3.2}$$

El término $|D_j|/|D|$ actúa como el peso de la partición $j - esima$. $Info_A(D)$ es la información que se espera requerir para clasificar un elemento de D en la partición mediante A. A menor cantidad de información requerida, se obtendrán particiones más simples.

Por lo anterior, dentro del contexto de clasificación se define a la ganancia de información de un atributo A_i como el requerimiento original de información para representar a los elementos de D, menos el requerimiento necesario que se genera con la entropía del atributo A_i. Es decir $Gain(A_i) = Info(D) - Info_{A_i}(D)$.

En otras palabras, $Gain(A_i)$, indica qué tanto se ha ganado al dividir mediante el atributo A_i. El atributo A_i con la ganancia de información mayor ($Gain(A_i)$) se elige cómo el atributo de partición o división en el nodo N. Esto es equivalente a afirmar que se crea una partición en el atributo A_i que genera la mejor clasificación, de modo que la cantidad de información aún requerida para finalizar la clasificación de los elementos es mínima.

En caso de que se maneje un atributo cuyo dominio de valores es continuo se elige el mejor punto de división y éste sirve como un umbral para el atributo A_i. Para determinar cómo se elige este punto se procede de la siguiente forma. Primeramente se ordenan los valores del atributo en forma incremental. De este modo, el punto medio entre cada par de valores adyacentes se puede considerar como un posible punto de división. Por lo tanto, dados v valores de A_i, se consideran $v - 1$ posibles puntos de división. Por cada uno de los posibles puntos de división para el atributo A_i, se evalúa $Info_{A_i}(D)$

Al finalizar estas evaluaciones, el punto con el menor requerimiento de información esperada para el atributo A_i, es elegido como el punto de división para A_i. Después de esto, el conjunto de D se divide en dos subconjuntos. D_1 que contiene aquellos elementos donde $A_i \leq valor\_punto\_division$ y D_2 es el subconjunto que contiene los elementos de D tales que $A_i > valor\_punto\_division$

El criterio "ganancia de información" presenta el problema de desviación ante conjuntos de prueba con múltiples resultados para un atributo determinado, es decir en la presencia de atributos con un dominio muy grande. Ya que por su misma construcción, se tiende a seleccionar atributos que poseen una gran cantidad de valores y esto afecta significativamente su desempeño. La razón de ello es que los atributos con muchos valores poseen tantos que tienden a limitar separadamente los elementos del conjunto de entrenamiento utilizados en cuestión dentro de subconjuntos muy pequeños. Debido a esto, se tendrá una ganancia de información muy alta relativa a los elementos del conjunto de entrenamiento, a pesar de poseer una capacidad de predicción muy baja con respecto a los elementos del conjunto de prueba.

Un ejemplo resulta de considerar conjuntos de datos donde se mantiene la llave primaria. Así si se selecciona como atributo de división a una llave primaria PK entonces se generarán tantas particiones como valores distintos, conteniendo cada una de ellas un único elemento. En este tipo de partición se obtienen conjuntos puros, ya que la información necesaria para clasificar el conjunto de datos (D) basados en esta partición sería $Info_{PK}(D) = 0$.

Por lo cual, la información obtenida al llevar a cabo esta partición en el atributo es máxima. Empero una distribución con estas características carece de utilidad para llevar a cabo una clasificación.

Rango de ganancia

Debido al problema que presenta el anterior criterio, Quinlan introdujo el término de **"rango de ganancia"** [63] el cual intenta atenuar este comportamiento. La forma de resolverlo consiste en aplicar un tipo de normalización a la ganancia de información utilizando un valor conocido como **"información de división"** ($SplitInfo(A_i)$). Este valor, penaliza estos atributos que poseen muchos valores. La información de división de define como:

$$SplitInfo_{A_i}(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \cdot log_2\left(\frac{|D_j|}{|D|}\right) \tag{3.3}$$

Este valor representa la información potencial generada al dividir el conjunto de entrenamiento, D, en v particiones (cada una corresponde a uno de los resultados de las evaluaciones sobre el atributo A). Es importante observar que para cada salida, se considera el número de elementos que tienen esa salida con respecto al número total de elementos en D. Es decir, este método toma en consideración el número y tamaño de las posibles ramificaciones cuando se elige un atributo. De tal manera que el rango de ganancia para un atributo A_i se define en términos de la ganancia de información que se obtiene ($Gain(A_i)$) entre el valor de la información de división ($SplitInfo(A_i)$). Esto se presenta en la Ecuación 3.4.

$$GainRatio(A_i) = \frac{Gain(A_i)}{SplitInfo(A_i)} \tag{3.4}$$

La Ecuación 3.4 expresa la proporción de información que es útil y generada por la división, es decir, aquella que resulta ser útil para la clasificación. Así luego de calcular este valor para todos los atributos, el atributo con el rango de ganancia mayor se elige como el atributo de división para ese nodo en particular. No obstante de la Ecuación 3.4 es posible observar que mientras el valor de la información de división ($SplitInfo(A_i)$) se acerca a 0, el rango de ganancia se vuelve inestable. Es decir, si el rango de ganancia llega a seleccionar atributos cuya información de división puede ser cero o muy pequeña

(lo cual sucede cuando algunos atributos llegan a tener el mismo valor para casi todas los elementos del conjunto de entrenamiento), este valor queda indefinido. Para soslayar este fenómeno, se han definido algunas heurísticas. Por ejemplo en [62] se indica que primeramente hay que calcular la ganancia de información para cada atributo, y después se aplica el método de rango de ganancia solamente considerando aquellos atributos para los cuales su ganancia de información es mayor que el promedio de todos los atributos.

Índice Gini

El índice Gini es una técnica cuyas bases se introducen en el algoritmo utilizado en CART [48] y es el criterio utilizado en SPRINT [69]. Este índice pondera la impureza de una partición de datos o conjunto datos de entrenamiento, es decir mide la diversidad de las clases presentes dentro de un conjunto de datos, la peor situación produce un valor de 0.5 (cuando se trata de conjuntos con dos clases posibles). Este valor disminuye mientras una división favorezca alguna clase. El criterio se define como:

$$Gini(D) = 1 - \sum_{i=1}^{m} P_i^2 \tag{3.5}$$

En esta Ecuación, P_i representa la probabilidad de que un elemento del conjunto D pertenezca a la clase C_i y es estimado por $|C_{i,D}|/|D|$. Es máximo cuando los elementos se encuentran distribuidos de manera equitativa entre todas las clases y representa la información menos interesante, similarmente es mínimo (0) cuando todos los elementos pertenecen a una clase, implicando información más importante dentro del contexto.

El índice Gini considera una división binaria para cada atributo A_i, por lo cual, debido al dominio del atributo se consideran dos procesos distintos.

- Si A_i posee dominio discreto de v valores distintos, $a_1, a_2, \ldots, a_v$, dentro del conjunto de datos D. Se examinan todos los posibles subconjuntos que pueden integrarse con los distintos valores conocidos de A_i. Esto con la finalidad de determinar la mejor división binaria basada en el atributo A_i, dado que A_i cuenta con v valores distintos entonces existen 2^v posibles subconjuntos[13] por analizar. Cada subconjunto, denotado por S_{A_i}, representa una posible evaluación binaria de pertenencia (por ejemplo "$A \in S_{A_i}$") para el atributo A_i. De este modo, dado un elemento esta evaluación se satisface si el valor del atributo A_i para el elemento en cuestión se encuentra dentro de los valores listados en S_{A_i}.

Cuando se consideran divisiones binarias, se calcula una suma ponderada del nivel de impureza de cada conjunto resultante de la división. Por ejemplo, una división binaria

[13] Técnicamente solamente se evalúan $2^v - 2$ subconjuntos, sin embargo el conjunto potencia y el conjunto vacío no representan conceptualmente una división.

del atributo A del conjunto D en D_1 y D_2, genera un índice Gini representado por la siguiente ecuación:

$$Gini_{A_i}(D) = \frac{|D_1|}{|D|}Gini(D_1) + \frac{|D_2|}{|D|}Gini(D_2) \tag{3.6}$$

Así, para cada atributo se considera cada una de las divisiones binarias.

- Si A_i posee dominio continuo entonces cada posible punto de división debe ser tomado en cuenta. Para ello se utiliza una estrategia similar a la descrita para la ganancia de información, donde el punto medio entre cada par de valores adyacentes ordenados es tratado como un posible punto de división. Posteriormente se calculan todos los valores de índice Gini para cada uno de estos puntos intermedios y aquel cuyo valor sea menor es elegido como el punto de división. Por último, el conjunto D queda dividido en dos subconjuntos D_1 que contiene aquellos elementos donde $A_i \leq valor\_punto\_division$ y D_2 es el subconjunto que contiene los elementos de D tales que $A_i > valor\_punto\_division$.

En cualquier caso, la disminución de la impureza que se presenta mediante la realización de una división binaria sobre un atributo A_i se define como:

$$\Delta Gini(A_i) = Gini(D) - Gini_{A_i}(D) \tag{3.7}$$

El atributo que maximiza esta reducción de impureza (o equivalentemente, aquel que posee el índice Gini menor) es seleccionado como el atributo de división. Este atributo y, ya sea el subconjunto de división (para atributos discretos) o el punto de división (para atributos continuos), conforma en conjunción el criterio de división.

Otros criterios

Cada uno de estos criterios poseen distintos niveles de desviación o tendencias. Por un lado la ganancia de información presenta este fenómeno respecto a atributos con múltiples valores, por otro lado aunque el rango de ganancia se ajusta para aminorar esta tendencia, se observa cierta predisposición a generar divisiones no balanceadas donde alguna de ellas tiende a ser considerablemente de menor tamaño que las restantes [62]. Por último, la medida del índice Gini se comporta de manera similar a la ganancia de información, exhibe dificultades cuando el número de clases es grande y además tiende a favorecer evaluaciones que resultan en divisiones del mismo tamaño y pureza.

Se han desarrollado otros criterios de selección de atributos, el algoritmo de decisión CHAID utiliza un criterio basado en la prueba estadística χ^2 de independencia. Otros criterios incluyen a C-SEP[14] , que presenta mejores resultados que la ganancia de

[14] Class SEParation.

información y el índice Gini en presencia de ciertas condiciones [20] y la estadística-G [72](criterio que asemeja a una distribución χ).

Además se definen criterios que consideran divisiones de múltiples variables (cuando la división de elementos se basa en *combinaciones de atributos* en lugar de un solo atributo). El sistema CART [48] permite encontrar divisiones de este tipo basándose en combinaciones lineales de atributos. Estas divisiones de múltiples atributos constituyen una forma de construcción de atributos, donde nuevos atributos se crean basados en los existentes inicialmente[15] .

Por último, es importante considerar que aún cuando todos estos criterios presentan ciertas tendencias o desviaciones, la complejidad de la inducción del árbol decisión incrementa exponencialmente con respecto a la altura del mismo [53]. Por lo cual, los criterios tienden a producir árboles más anchos (se prefieren árboles con múltiples ramificaciones que los árboles binarios y eso crea árboles más balanceados). Sin embargo, otros estudios muestran que los árboles anchos al considerar más hojas [19], tienden a poseer un índice de error mayor. Aún cuando existen múltiples trabajos que analizan la eficiencia de estos criterios [20, 51, 6, 71] ninguno de ellos resulta ser idóneo para aplicar en todos los problemas.

3.2.2. Podado de árboles

Cuando se construye un árbol de decisión, múltiples ramificaciones reflejarán anomalías del conjunto de datos de entrenamientos debido al ruido o valores fuera de límites. Ajustarse demasiado a los datos de entrenamiento suele tener como consecuencia que el modelo se comporte mal con nuevos elementos, ya que el modelo solamente es una aproximación del concepto objetivo del aprendizaje. Lo anterior se hace patente cuando los datos de entrenamiento contienen ruido (errores en los atributos o incluso en las clases) ya que el modelo intentará ajustarse a los errores y esto perjudicará el comportamiento global del modelo aprendido, lo cual degenera en un sobre entrenamiento de los árboles de decisión.

El problema anterior es abordado por distintos métodos de podado de árboles, éstos típicamente utilizan medidas estadísticas para eliminar ramificaciones poco confiables. Un árbol y su equivalente podado se presenta en la Figura 3.3. Los árboles podados tienden a ser menores, menos complejos y por lo mismo más fáciles de comprender. Además generalmente son más rápidos y mejores al clasificar los datos de prueba, que los árboles no podados.

Se distinguen dos métodos de podado:

1. Podado previo. Un árbol de decisión es podado cuando se detiene su construcción, al decidir que no es necesario llevar a cabo más divisiones o particiones del subconjunto de elementos del conjunto de datos de entrenamiento en un nodo determinado. En ese momento el nodo en cuestión es marcado como una hoja. Así,

[15] La creación de atributos es una tarea de la fase de transformación de datos del KDD.

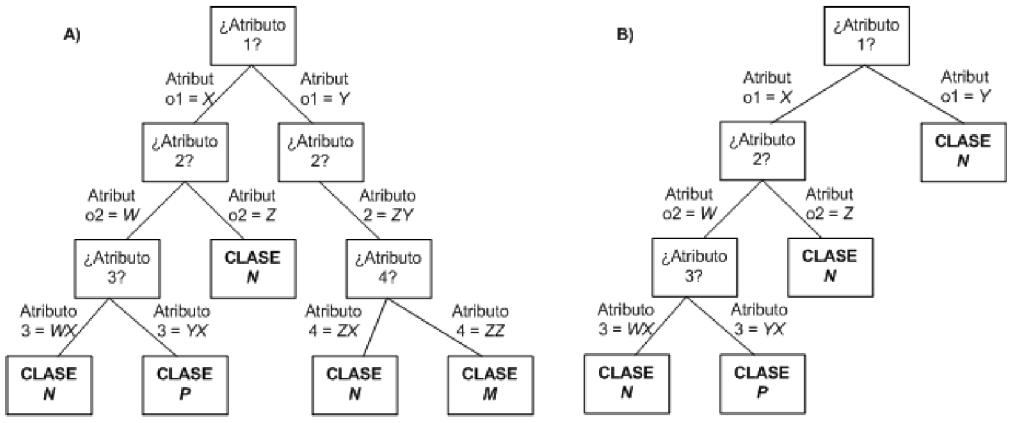


Figura 3.3: Un árbol de decisión **A)** y su homólogo podado **B)**.

dicha hoja contendría la etiqueta de la clase más frecuente en el subconjunto de datos o bien, las distintas distribuciones de probabilidad para dichos elementos. La determinación de llevar a cabo o no una división para un nodo determinado depende de múltiples elementos, por ejemplo estadísticas de importancia o criterios de selección de atributos. Si la división de los elementos en un nodo resulta en una división que cae fuera de diversos umbrales, entonces la división posterior de dicho subconjunto se detiene.

2. Podado posterior. El segundo enfoque y más común, se conoce como podado posterior, el cual remueve sub-árboles a partir de un árbol totalmente construido. Un sub-árbol en un nodo dado es podado al remover sus ramificaciones y reemplazarlas con una hoja. La hoja es etiquetada con la clase más frecuente en el sub-árbol que está reemplazando. Este enfoque se utiliza en múltiples algoritmos [48].

La forma de abordar este último enfoque varía entre distintos algoritmos, por ejemplo [48] considera la complejidad del costo de podado, como una función del número de hojas en el árbol y el rango de error del árbol[16] . Inicia analizando las hojas del árbol continuando hacia la raíz, y por cada nodo interno, N, calcula el costo del sub-árbol en N y el costo del sub-árbol en N si se poda en ese lugar (es decir, se reemplaza mediante una hoja). Dichos valores se comparan, si el costo del sub-árbol en el nodo N resulta en un costo menor[17] , entonces este sub-árbol es podado. En caso contrario, se mantiene. De este modo se generan progresivamente distintos árboles podados, y en general el árbol de decisión más pequeño que minimiza este costo, es el preferido.

Además, existen métodos denominados "podados pesimistas", en ellos se caracterizan dos elementos importantes, el primero consiste en utilizar el rango de error para decidir

[16] Donde este rango de error es el porcentaje de elementos clasificados de forma errónea por el árbol en cuestión.

[17] Para estimar este costo, se utiliza un conjunto de podado que contiene elementos con etiquetas en el atributo clase.

llevar a cabo o no la poda sobre los distintos sub-árboles identificados, el segundo se refiere a la utilización del conjunto de entrenamiento para estimar los distintos rangos de error. Se conoce por pesimista debido a que ajusta los rangos de error a partir del conjunto de datos de entrenamiento al agregar una penalización, cada que se incurre en una desviación. A diferencia de una estimación de la precisión basada en el conjunto de datos de entrenamiento que es una estimación optimista y por lo tanto posee ciertas tendencias o desviaciones.

Otro método para la elección de podado [64], se basa en analizar el número de bits necesarios para codificar dichos árboles podados. El mejor árbol podado es aquel que minimiza el número de bits de codificación. A diferencia de los métodos anteriores, no es necesario tomar en cuenta un conjunto independiente de elementos.

De este modo, dichos enfoques pueden llevar a cabo de manera alternada. No obstante una consecuencia de utilizar estos métodos, ya sea alguno de ellos o de manera simultánea, es que los nodos hoja posiblemente ya no serán puros, es decir, es probable que contengan elementos de varias clases. El podado posterior requiere más cálculos que el podado previo, sin embargo generalmente genera un árbol más confiable. De los métodos de podado, ninguno ha demostrado ser superior a los demás en todos los ámbitos [9, 17]. Por último, es importante considerar que si bien los árboles podados suelen ser más compactos que sus homólogos sin poda alguna, aún así pueden ser grandes y complejos. Los árboles de decisión pueden sufrir de dos problemas que aumentan la complejidad de interpretación, los cuales son:

- Repetición .- Se presenta cuando un atributo es repetidamente evaluado a lo largo de una ramificación del árbol. Este fenómeno se ilustra en la Figura 3.4.
- Replicación .- Consiste en la aparición de sub-árboles duplicados dentro de la estructura general del árbol. Dicha condición se ilustra en la Figura 3.5.

Estas situaciones pueden disminuir la precisión y comprensión del árbol de decisión. Una forma de evitar la aparición de estos eventos, consiste en utiliza divisiones de múltiples variables (divisiones basadas en combinaciones de atributos) o bien, utilizar otras formas de representación de conocimiento como lo son las reglas de asociación.

3.2.3. Escalamiento

La eficiencia de los algoritmos de los árboles de decisión se encuentra determinada para conjuntos de datos relativamente pequeños, sin embargo en entornos de grandes bases de datos se convierte en un problema de consideración debido generalmente a que es necesario un análisis sobre todos los elementos y sus atributos para decidir por aquel que mejor cumpla con el criterio en cuestión y debido al tamaño de las grandes bases de datos no es posible asumir este requerimiento.

Los primeros algoritmos de los árboles de decisión asumen que los elementos de los conjuntos de entrenamiento residen totalmente en memoria. Esta suposición será

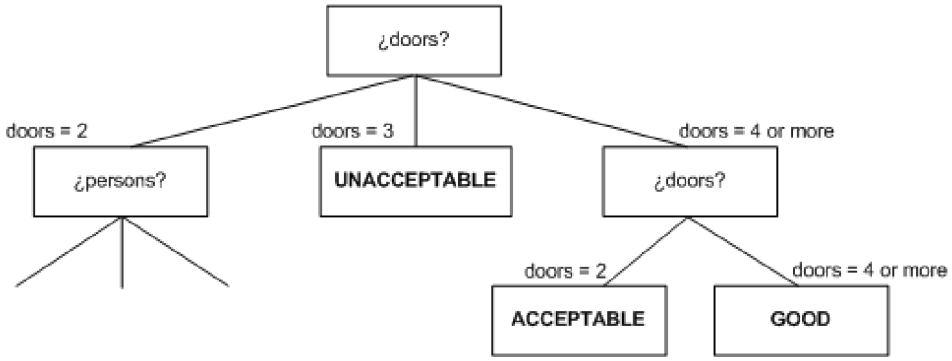


Figura 3.4: Árbol de decisión donde se aprecia la repetición de evaluación sobre un atributo. Un atributo es evaluado en múltiples ocasiones sobre la misma ramificación.

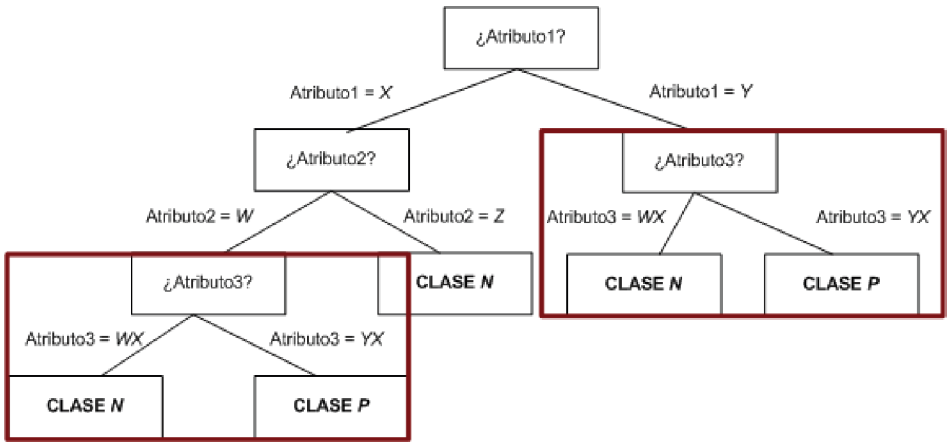


Figura 3.5: Árbol de decisión donde se aprecia el fenómeno de replicación de un subárbol dentro del árbol principal.

tomada en cuenta para todos los conjuntos de datos que se emplean en el presente trabajo.

3.3. Árboles de decisión y aprendizaje automático

Desde el punto de vista del aprendizaje automático, el problema de asociar a los nuevos elementos una clase dada se describe de manera similar como una búsqueda de un árbol que clasifique correctamente los elementos. De esta manera, en los árboles de decisión podemos distinguir las siguientes características:

- Espacio de hipótesis.- Se genera un espacio que contiene a todos los posibles árboles de decisión. Es decir, cualquier función finita que tome valores discretos puede ser representada como un árbol de decisión. Además, en cada elección se considera una única hipótesis, a diferencia de otros algoritmos del aprendizaje automático en los que se consideran simultáneamente todas las hipótesis consistentes con los ejemplos. El algoritmo considera en cada paso gran cantidad de elementos, mientras que otros consideran los elementos uno a uno. Esta es la razón de la robustez frente al ruido que presentan los árboles de decisión. Si los elementos fueran considerados uno a uno y alguno de ellos fuese erróneo, el efecto sobre la corrección de los resultados sería mayor que si el dato erróneo se considera simultáneamente a otro conjunto de datos en los que no hay ruido.

- Método de búsqueda.- Se manifiesta una escalada[18] , a partir del árbol vacío. El algoritmo no permite retroceso; una vez se ha elegido un atributo para un nodo, los ejemplos de este se clasifican según ese atributo, y más adelante no hay posibilidad de volver a este punto y considerar un atributo distinto, por lo cual es posible obtener óptimos locales en lugar de óptimos globales.

- Utilización de heurística.- Se presenta la utilización de una heurística que guía la búsqueda, generalmente se emplea la ganancia de información (Sección 3.2.1).

3.3.1. Algoritmo genérico de clasificación

Existen diversos algoritmos relacionados con los árboles de decisión (ID3, C4.5, etc.). Sin embargo todos ellos parten de un ciclo principal que tiene la siguiente estructura general:

- Se asigna el mejor atributo A al siguiente nodo. El mejor atributo será aquel que ofrezca mayor ganancia de información.

- Para cada valor de A se crea una nueva arista descendente.

[18] *Hill-climbing.*

- Se clasifican los ejemplos del conjunto de entrenamiento de ese nodo entre sus descendientes.
- Si los ejemplos del conjunto de entrenamiento quedan perfectamente clasificados (todos pertenecen a la misma clase), entonces se detiene. Si no, itera el proceso sobre los descendientes de ese nodo, no volviendo a usar el atributo A.

Como se ha presentado con anterioridad, el concepto de ganancia de información (Sección 3.2.1) es utilizado, y sirve para decidir qué atributo se debe utilizar para cada nodo del árbol.

3.3.2. Algoritmo ID3

El objetivo principal del algoritmo consiste en construir un árbol de decisión que explique cada instancia de la secuencia de entrada de la manera más compacta posible.

Las características básicas del algoritmo ID3 son:

- Desarrollado por Quinlan [62].
- Pertenece a la familia TDIDT.
- Elección del mejor atributo dependiendo de una determinada heurística.

Sin embargo, tiene un inconveniente muy importante que consiste en favorecer indirectamente a aquellos atributos con muchos valores, los cuales no necesariamente resultan ser los más útiles. Además no considera el manejo de atributos con dominios continuos, así como tampoco el manejo de poda de árboles.

Dentro de algoritmo ID3, se define al sesgo inductivo de manera general, como la mínima información adicional que hay que suministrar para que un algoritmo clasifique correctamente los nuevos elementos que se le presenten. Si esta información no existiese el algoritmo no serviría para clasificar elementos, aparte de los del conjunto de entrenamiento. En el caso del ID3, la necesidad del sesgo inductivo surge debido a que dado un conjunto de entrenamiento, en general, existen muchos árboles de decisión consistentes con los ejemplos y no hay forma directa de elegir alguno de ellos.

Una primera heurística consiste en preferir la generación de los árboles cortos a los largos. Es decir, si se emplea una búsqueda a lo ancho (BFS) en el espacio de hipótesis, iniciando por la raíz del árbol, se recorren todos los árboles de profundidad 1, después los de profundidad 2 y así sucesivamente hasta encontrar un árbol que fuera consistente con el conjunto de entrenamiento. En otras palabras, se elige el primer árbol aceptable que encuentre en su búsqueda en escalada a través del espacio de árboles posibles. Sin embargo, ID3 no siempre devuelve el árbol más corto. Se eligen árboles en los cuales los atributos con mayor ganancia de información estén más cerca de la raíz[19] .

[19] Es preferible elegir la hipótesis más simple de entre todas aquellas que resuelven el problema.

Listado 3.1: Pseudo-código del algoritmo ID3

```
Si todos los ejemplos de E pertenecen a una misma clase C, entonces
   arbol1 <--- nodo etiquetado con C
SiNo
   Si a = NULL , entonces
      C <--- clase mayoritaria de elementos de E
      arbol1 <--- nodo etiquetado con C
   SiNo
      A <--- mejor atributo de a
      arbol1 <--- nodo etiquetado con A
      Para cada v perteneciente a los valores de A, hacer
         EAv <--- los elementos de E que tienen el valor v para el
             atributo A
         Si EAv = NULL, entonces
            arbol2 <--- nodo etiquetado con la clase mayoritaria en E
         SiNo
            arbol2 <--- ID3(EAv, a-{A})
         arbol1 <--- añadir a arbol1 el arbol2, a través de una rama
             etiquetada con v
Devolver arbol1
```

Si, dado un espacio de hipótesis H, se consideran dos hipótesis $h, h' \in H$, se dice que h presenta un sobre entrenamiento al conjunto de datos C si clasifica mejor que h' los elementos del conjunto de entrenamiento, pero h' clasifica mejor que h el conjunto de datos completo de posibles instancias. Las principales causas de que esto ocurra son:

- Exceso de ruido (lo que se traduce en nodos adicionales).
- Un conjunto de entrenamiento demasiado pequeño como para ser una muestra representativa de la verdadera función objetivo.

Para evitar este sobre entrenamiento, se consideran dos tipos de estrategias:

1. Estrategias que limitan el crecimiento del árbol antes de que llegue a clasificar perfectamente los elementos del conjunto de entrenamiento.
2. Estrategias que permiten que el árbol crezca completamente, y después realizan una poda.

Éstas últimas han demostrado ser más eficaces que las primeras. El pseudo-código del algoritmo ID3 se presenta en el Listado 3.1.

3.3.3. Algoritmo C4.5

C4.5 es un algoritmo para la construcción de árboles de decisión basado en el algoritmo ID3, perteneciente a la familia TDIDT. También ha sido desarrollado por Quinlan. El núcleo del algoritmo es el mismo que en el ID3. Sin embargo, incorpora muchas de las técnicas que abordan y tratan los inconvenientes siguientes:

- Incorporación de atributos tanto discretos como continuos.
  Inicialmente el algoritmo ID3 se planteó para atributos que presentaban un dominio discreto. Para incorporar el manejo de dominios continuos, se dividen estos valores en intervalos discretos, de forma que el atributo tendrá siempre valores comprendidos en uno de estos intervalos. Por ejemplo, si se tiene un atributo A con valores continuos, el algoritmo crea dinámicamente un nuevo atributo booleano Ac, que es cierto si $A \leq c$, y falso en otro caso. La cuestión más importante es decidir qué valor c se elige, siendo aquel que proporciona una mayor ganancia de información. En la práctica no se suele escoger un solo valor, sino que se divide el rango de valores en varios intervalos. Y sólo se escogen aquellos valores umbrales que dejen al menos 2 casos con valor conocido a su izquierda y a su derecha.

- Utilización del rango de ganancia.
  En caso de que la ganancia de información no sea conveniente como heurística en la tarea concreta que se esté tratando (Sección 3.2.1).

- Método de poda posterior.
  Para evitar el sobre entrenamiento (Sección 3.2.2).

- Método probabilístico para solucionar el problema de los atributos con valor desconocido.
  En ciertos casos existen atributos de los cuales se conoce su valor para algunos ejemplos, y para otros no. En estos casos lo más común es estimar el valor basándose en otros elementos de los que sí se conoce el valor. Así para el caso de atributos con dominio discreto, al elemento de valor desconocido se le da el valor que más aparezca en los demás elementos. En el caso de atributos con dominio continuo se puede asignar la media a los elementos sin valor. Otros métodos más complejos se basan en la probabilidad [33], no en el valor más común. Por ejemplo, si en un nodo se evalúa un atributo booleano del que se tiene 6 elementos iguales a 1 y cuatro ejemplos iguales a 0, la probabilidad de que un elemento del que no se conoce su valor sea 1 es de 0.6, y 0.4 de que sea 0. El algoritmo C4.5 usa este método probabilístico.

Al inicio solamente se encuentra el nodo raíz, en cada nodo se utiliza una estrategia de divide y vencerás (aplicando el algoritmo a conjuntos de datos cada vez más pequeños), llevando a cabo una optimización local sin regreso. Se calcula la frecuencia de los elementos para cada clase. Si todos pertenecen a alguna clase C_j entonces el nodo

será etiquetado como una hoja con la clase C_j. En el caso de que existan más clases, se calcula la ganancia de información para cada atributo. Aquel que posea una mayor ganancia de información se elige como el atributo de división. Las evaluaciones se hacen dependiendo del tipo de dominio del atributo. Se consideran 2 casos:

- Para discretos, se crean n particiones una por cada valor
- Para continuos se manejan dos particiones a partir de un punto de división o umbral

A partir del nodo, se crean n nuevos nodos y el conjunto de datos se divide entre ellos. En el caso de que alguno de ellos sea vacío para los elementos, el nodo recién creado se convierte en una hoja y se asocia por etiqueta la clase más frecuente. En caso contrario se aplica recursivamente lo anterior.

3.4. Recapitulación

El aprendizaje de árboles de decisión está englobado como una metodología del aprendizaje supervisado. La representación que se utiliza para las descripciones del concepto adquirido es una estructura arborescente conocida como árbol de decisión. Los árboles de decisión suele ser más robustos frente al ruido y conceptualmente sencillos. Para el manejo del ruido los árboles de decisión utilizan técnicas de poda, las cuales les permiten eliminar ramificaciones que representan la clasificación para aquellos elementos que presentan este ruido.

Un árbol de decisión puede interpretarse esencialmente como una serie de reglas compactadas para su representación en forma de árbol. Cada nodo está etiquetado con un par atributo-valor y las hojas con una clase, de forma que la trayectoria que se determina desde la raíz hasta alcanzar una hoja etiquetada con la clase del ejemplo, determina la pertenencia para cada nuevo elemento.

Una de las características importantes de los árboles de decisión consiste en la forma en que manejan la ausencia de valores y observaciones con valores fuera de rangos válidos. Para los primeros se determina la mejor partición del nodo basado en una atributo con los datos que se tiene disponible. Si cuando se desea clasificar un elemento del conjunto de entrenamiento (o de prueba posteriormente) no existe el valor correspondiente del atributo entonces se utiliza una partición sustituta para evaluar a ese elemento. Este proceso se repite hasta que exista una partición donde se tenga definido un atributo con un valor para la división. Para el manejo de los valores fuera de rangos, los elementos son asignados durante la evaluación del nodo en cuestión, hacia aquella ramificación que contenga ese rango de valor. Recordando que las ramificaciones de los árboles de decisión definen rangos que contienen todos los posibles valores para atributos con dominios continuos y las etiquetas para los atributos con dominios discretos. Sin embargo, la susceptibilidad que presentan ante los distintos grados de error en los datos no ha sido analizada a nuestro conocimiento.

Las propiedades generales de los árboles de decisión, así como su amplia utilización dentro de las áreas de aprendizaje de máquinas y reconocimiento de patrones, hace que esta técnica sea elegida (en particular el árbol de decisión C4.5) para analizar su comportamiento como clasificador en presencia de distintos tipos, porcentajes y distribución de errores en los datos que se le presentan como elementos entrenamiento.

CAPÍTULO 4

K vecinos más próximos

4.1. Introducción

Los métodos de clasificación presentados anteriormente se catalogan como métodos de aprendizaje no retardados. Este tipo de métodos se caracterizan por que al obtener un conjunto de datos de entrenamiento empiezan a desarrollar una generalización del modelo (la clasificación propiamente) antes de obtener nuevos elementos para clasificar, es decir el conjunto de datos de prueba. Se puede visualizar al modelo de aprendizaje listo y dispuesto a clasificar elementos que no conoce.

Por otro lado, se encuentran los métodos de aprendizaje retardados. En los cuales el modelo retarda la aplicación de sus distintas etapas lo más que sea posible antes de iniciar con la construcción propiamente del modelo, es decir para clasificar un elemento del conjunto de datos de prueba. Así, cuando se le suministra un elemento de entrenamiento, un método de aprendizaje retardado simplemente almacena (o realiza el menor procesamiento posible sobre los datos) y espera hasta que obtenga un elemento del conjunto de datos de prueba. En esos momentos es cuando lleva a cabo una generalización para clasificar el elemento basándose en su similitud con respecto a los elementos de entrenamiento previamente almacenados. A diferencia de los métodos de aprendizaje no retardados, los métodos retardados realizan menos trabajo cuando un elemento de entrenamiento les es suministrado y más trabajo cuando llevan a cabo el proceso de clasificación.

Cuando se utilizan métodos de aprendizaje retardados para la clasificación, éstos suelen ser computacionalmente muy costosos ya que requieren técnicas de almacenamiento muy eficientes y diseñadas específicamente para ello. Ya que es necesario manejar estructuras de datos sobre las cuales se mantienen las instancias actuales, mismas sobre las que se harán las comparaciones pertinentes.Además ofrecen una panorama

muy pobre sobre la estructura de los datos. Sin embargo, un punto a su favor es que brindan soporte incremental de manera directa, son capaces de modelar espacios de decisión complejos que pudieran no ser tan fácilmente descritos por otros algoritmos de aprendizaje.

Cuando se aprende a partir de ejemplos o datos conocidos, generalmente lo que se intenta es adquirir la capacidad de tomar una decisión sobre nuevos datos. En teoría, el repetir un comportamiento ante la presencia de elementos muy parecidos a los analizados con anterioridad resulta una acción idónea. Esta acción resulta ser clave en varias tareas de minería de datos. Así, la clasificación posterior se realiza por medio de una función que mide la proximidad o parecido de cada nuevo elemento con respecto a los elementos analizados anteriormente y observando las clases de éstos elementos.

De lo anterior se esbozan dos nociones importantes. La primera consiste determinar que se entiende por similitud, dando así lugar al concepto matemático equivalente de distancia. Y la segunda radica en especificar cuando se aprovecha dicha similitud: si se llevará a cabo con un pre-procesamiento no retardado (anticipado o ansioso) o bien mediante un pre-procesamiento retardado (perezoso). Los métodos que buscan instancias similares se ajustan muy bien a esta segunda perspectiva: esperar hasta que se plantea una cuestión sobre un nuevo elemento. En dicho momento, se busca entre los elementos anteriormente almacenados al más similar y se asigna la etiqueta o clase al nuevo elemento de manera idéntica al elemento almacenado. Esto supone que no se crea un modelo general para la tarea en cuestión, sino que se predice al momento y en función de cada caso concreto. Lo anterior hace que estos métodos sean conocidos como: aprendizaje basado en instancias, métodos de aprendizaje retardados o perezosos o métodos basados en similitud o en distancias.

Algunas de las ventajas que presentan estos clasificadores son la siguientes:

- Poseen una implementación simple y son analíticamente tratables, es decir puede construirse un modelo matemático que permite predecir el comportamiento.
- Resultan ser óptimos en la precisión con el manejo de grandes bases de datos. Se ha demostrado [44] que el error que presentan se encuentra acotado por el error presentado por el clasificador ingenuo de Bayes[1] .
- Utilizan información local, lo cual les permite ser altamente adaptables.
- El costo del aprendizaje es casi nulo.
- No es necesario hacer suposiciones sobre los elementos[2] .
- Facilita la extensión del mecanismo para predecir valores continuos.

[1] El nivel de error se acota de la siguiente forma para utilizando $k = 1$.
$P(error)_{Bayes} > P(error)_{1-NN} < 2P(error)_{Bayes}$

[2] A diferencia, por ejemplo, del clasificador ingenuo de Bayes.

- Debido a su estructura, permiten una implementación paralela de manera directa.
- Dependiendo de la función de medida de distancia, son tolerantes al ruido.

Se han enumerado desventajas e inconvenientes del aprendizaje basado en ejemplares [5], pero se suelen considerar como las más importantes a las siguientes:

- Requieren de gran almacenamiento para almacenar todas las instancias.
- Requieren de trabajo computacional intenso para el cálculo de las distancias entre los elementos de los conjuntos de datos, es decir el cómputo de los k-vecinos más próximos.
- No existe un mecanismo para decidir el valor óptimo de k, este depende del conjunto de datos.
- Su rendimiento disminuye, son sensibles a la presencia de múltiples atributos[3] .
- Las mismas medidas de proximidad sobre atributos simbólicos suelen proporcionar resultados muy dispares en problemas diferentes.

Los clasificadores basados en proximidad o similitud, se presentaron por primera vez en la década de los años 50 [24] y hasta décadas más recientes se han implementado de modo eficiente.

Las áreas de aplicación donde se ha utilizado esta técnica incluyen campos como identificación de imágenes de satélites [7], aplicaciones GIS [32, 79], clasificación de textos, reconocimiento de imágenes entre otras. Múltiples pruebas experimentales han demostrado su eficacia, por ejemplo en el estudio de diversas colecciones de documentales [81] o resúmenes de literatura médica [59]. El método perece eficaz cuando existen múltiples categorías y cuando los documentos son heterogéneos y difusos. Esto también se ha confirmado en trabajos de categorización de páginas web expuestos [28].

Por otro lado estos clasificadores han sido ampliamente estudiados con múltiples trabajos sobre el incremento de su precisión, por ejemplo la combinación de k vecinos más próximos con el clasificador ingenuo de Bayes [25, 84] o su implementación mediante árboles binarios de búsqueda [26]. Sin embargo su desempeño en la presencia de distintos tipos de errores es desconocida.

4.2. Medidas de distancia

Para formalizar el concepto de similitud se recurre a la utilización de medidas de distancia. Así, si se desea conocer la similitud entre dos instancias o elementos, es necesario elegir una función de distancia y calcular con ella la distancia entre los dos

[3] La maldición de la dimensionalidad.

elementos. Empero, el método utilizado para definir esta función de distancia impacta directamente en la relación que se obtenga para los elementos que se comparan, de modo que para cada función de distancia utilizada, la medida de distancia varia, por lo tanto la elección de esta función resulta ser fundamental. Lo anterior también se considera una ventaja, puesto que permite adaptar los distintos métodos de aprendizaje para trabajar con múltiples clases de problemas.

Existen un conjunto muy amplio de medidas de distancia, desde la distancia euclidiana o clásica hasta otras que cumplen los requisitos de una función de distancia (llamada entonces métrica) y pueden funcionar mejor dependiendo del contexto en el cual se de. Más aún, dependiendo de la aplicación se puede definir funciones a la medida que presenten un comportamiento similar a una métrica de distancia.

Definición de una métrica Sean x, y y z elementos, se define como una métrica o función de distancia $d(\cdot, \cdot)$, si se cumplen las siguientes propiedades:

- No negatividad: $d(x, y) \geq 0$
- Reflexibilidad: $d(x, y) = 0$ si y solo si $x = y$
- Simetría: $d(x, y) = d(y, x)$
- Desigualdad del triángulo: $d(x, y) + d(y, z) \geq d(x, z)$

Si la segunda propiedad no se cumple, entonces $d(\cdot, \cdot)$ se conoce como una pseudométrica.

4.2.1. Medidas de distancia númericas

Las medidas de distancia más comunes son aquellas que se aplican sobre dos elementos, tales que todos los atributos son númericos. Por ejemplo, sean x y y dos elementos de dimensión n, se definen las siguientes distancias entre ellos:

- Distancia Euclídeana.- Es la distancia clásica, definida como la longitud de la recta que une dos puntos en el espacio euclídeano:

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

- Distancia de Manhattan[4] .- Se hace referencia al recorrido de un camino no en diagonal entre dos puntos:

$$d(x, y) = \sum_{i=1}^{n} |x_i - y_i|$$

[4] También conocida como "*Distancia por cuadras*".

- Distancia de Chebychev.- Calcula la discrepancia más grande en alguna de las dimensiones:

$$d(x,y) = max_{i=1\ldots n}|x_i - y_i|$$

- Distancia del coseno.- Considerando a cada elemento como un vector, la distancia es el ángulo que forma estos vectores:

$$d(x,y) = arccos\left(\frac{x^T y}{||x|| \cdot ||y||}\right)$$

- Distancia de Mahalanobis.- Distancia que utiliza una matriz de covarianza S :

$$d(x,y) = \sqrt{(x-y)^T S^{-1} (x-y)}$$

Existen aplicaciones donde cada elemento se encuentra compuesto por atributos númericos que constituyen una secuencia de eventos relacionados, por ejemplo $x = (x_1, x_2, \ldots, x_n)$, para este tipo de datos se consideran distancias especiales como la distancia de Spearman [30] o análisis de coeficientes de correlación. Para la utilización de todas las métricas anteriores, es necesario llevar a cabo un procesamiento de normalización sobre los atributos númericos, ya que algún valor muy elevado para alguno de los atributos tendría una magnitud media mucho mayor que las restantes y por lo mismo tendrá mayor peso al momento de calcular las distancias.

4.2.2. Medidas de distancia no númericas

El concepto de distancia no es exclusivo de los atributos númericos. También se lleva el concepto hacia atributos nominales. Para ello se utiliza la función delta (δ). Los valores que genera esta función son $\delta(x,y) = 0$ únicamente cuando $x = y$, y $\delta(x,y) = 1$ en otro caso. Con esta función se define la distancia entre dos elementos nominales x y y como:

$$d(x,y) = \omega \sum_{i=1}^{n} \delta(x_i, y_i)$$

Donde ω es un factor de reducción. Dicho factor se elige convenientemente cuando los atributos son combinación de nominales y númericos, se puede utilizar esta función de distancias para el subconjunto de atributos nominales, y alguna de las anteriores para el subconjunto de los atributos númericos y al finalizar combinar ambos valores utilizando un factor determinado.

De modo similar, se pueden definir distancias para otros tipos de datos más complejos. Cuando se trabaja con caracteres, una distancia común es la distancia de Levenshtein o distancia de Edición [57], en la que se ponderan inserciones, borrados y

sustituciones para obtener el número mínimo de operaciones requeridas para transformar una cadena de caracteres en otra. Así, entre menor sea el número de operaciones se considera que las cadenas son más similares. Otra distancia muy común utilizada en codificaciones binarias de los datos, es la distancia de Hamming que se define como "el número de posiciones de bits en los cuales dichos vectores toman valores diferentes", y esta dada por la siguiente ecuación:

$$d(x, y) = \sum_i x_i \ \oplus \ y_i$$

Por otro lado, si los datos son conjuntos más que vectores, se define la distancia entre conjuntos de la siguiente forma:

$$d(x, y) = \frac{|x \cup y| - |x \cap y|}{|x \cup y|}$$

La cercanía se define, como se ha mencionado anteriormente, mediante alguna distancia. Sin embargo no sólo es importante el conocer que un elemento tenga otro cercano, sino qué cantidad de ellos se encuentran cerca. Para ello se introduce el concepto de densidad que complementa al de distancia para evaluar la similitud entre elementos.

4.3. K vecinos más próximos

El método de clasificación del vecino más próximo asigna a cada nuevo elemento la clase del elemento que se encuentre más cercano utilizando para ello una función de distancia. Esta regla se conoce como 1-NN (*One Nearest Neighbor*), pero presenta problemas al ignorar la densidad o la región donde se encuentra el elemento. Ya que ciertamente puede ser que un elemento con clase C_i sea el elemento más cercano, sin embargo los siguientes x elementos más cercanos pueden pertenecer a otra clase C_j. Un ejemplo de esta clasificación se presenta en la Figura 4.1 para un modelo con dos atributos, representándose por ello en un plano.

La variante más conocida de este método se conoce como los k vecinos más próximos (k-NN o *k-Nearest Neighbor*), el cual se describe a inicios de la década de los años 50 [24], en sus inicios el método no fue ampliamente utilizado debido en parte a los grandes requerimientos de equipo que necesitaba, no fue sino hasta la década de los años 60 cuando se retomaron los trabajos alrededor de esta idea. Actualmente se encuentra enmarcado dentro del área de reconocimiento de patrones y clasificación.

Los clasificadores por vecinos más próximos basan su aprendizaje en la similitud. Comparan cada elemento del conjunto de datos de prueba con los elementos del conjunto de datos de entrenamiento y lo asocian con aquellos similares. Usualmente los elementos de entrenamiento se describen por medio de su n atributos y cada uno se representa como un punto en un espacio de n dimensiones, así todos los elementos del conjunto de datos de entrenamiento se almacenan en una representación de espacio n dimensional.

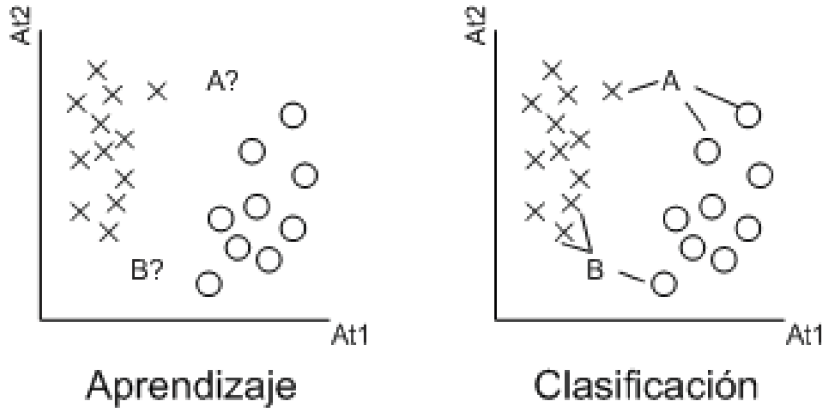


Figura 4.1: Ejemplo de clasificación mediante algoritmo k-NN. Se aprecia cómo el proceso de aprendizaje consiste en el almacenamiento de todos los elementos del conjunto de entrenamiento. Los elementos se representan de acuerdo a los valores de sus dos atributos y la clase a la que pertenecen (las clases son + y -). La clasificación consiste en la búsqueda de los k elementos (en este caso 3) más cercanos al elemento a clasificar.

Cuando se suministra un elemento nuevo, el clasificador busca este patrón espacial por los k elementos de entrenamiento que se acercan (asemejan) al elemento desconocido. Estos k elementos son los vecinos más próximos o cercanos del nuevo elemento y así, se asigna la clase mayoritaria entre éstos elementos. En caso de existir un empate entre el número de elementos con sus clases, la clase para el nuevo elemento se asigna de forma aleatoria entre estas clases. Por lo anterior, este valor k resulta determinante y no siempre es sencillo obtener un valor ideal.

El cálculo de la similitud dependerá de la función de distancia que se ocupe, generalmente la euclidiana (Sección 4.2). Sin embargo, dependiendo de la elección de la medida de distancia[5] , y en el caso de los datos númericos, puede ser necesario aplicar un proceso de normalización. Este proceso evita que atributos con valores muy grandes, obtengan un peso muy superior en comparación de atributos con valores bajos.

De este modo cuando se presentan atributos con dominios númericos se pueden aplicar distintos métodos de normalización empero, generalmente se utiliza el método *min-max*, para transformar un valor v de un atributo númerico A al valor v' dentro del rango $[0, 1]$ mediante la siguiente ecuación:

$$v' = \frac{v \ - \ min_A}{max_A \ - \ min_A} \tag{4.1}$$

En la Ecuación 4.1, min_A y max_A representan los valores mínimo y máximo del atributo A.

Para el algoritmo de los k vecinos más próximos, el elemento desconocido es asignado al elemento más común dentro de todos sus k vecinos. Cuando $k = 1$, para el elemento

[5] Recordando que la elección de la medida será en función del tipo de dato que se trate.

sin clase se asocia la clase del elemento del conjunto de entrenamiento que mas sea parecido a él. Como se ha presentando, este es un modelo basado en una función de distancia que mide o pondera la similitud entre elementos. En el caso de presentarse atributos con dominios discretos, dados n elementos de un conjunto de entrenamiento, $C = \{c_1, c_2, \ldots, c_l\}$ el conjunto de clases posibles y un elemento de prueba x, se encuentran sus k vecinos más próximos $x_1, x_2, \ldots, x_k$ y se realiza un consenso para asignar a x la clase más común (la moda). Esto es, la clase del elemento x, denotada por $clase(x)$ se determina por la Ecuación 4.2.

$$clase(x) = argmax_{c \in C} \sum_{i=1}^{k} \delta(f(x_i), c) \tag{4.2}$$

En la Ecuación 4.2, δ es una función tal que $\delta(f(x_j), c) = 1$ si $f(x_j) = c$ y $\delta(f(x_j), c) = 0$ en cualquier otro caso. Con $k = 1$, $clase(x) = f(x_j)$ con x_j siendo el elemento que minimiza la distancia $d(x, x_j)$.

Es decir, simplemente se asigna la clase del elemento más similar a él. Para el caso de atributos con dominios continuos, el modo de obtención de la clase se modifica. Ya que se consideran el valor medio de entre los k vecinos. Siendo $x_1, x_2, \ldots, x_k$ los k elementos más similares a x, se obtiene la clase mediante la Ecuación 4.3.

$$clase(x) = \frac{\sum_{i=1}^{k} f(x_i)}{k} \tag{4.3}$$

Sin embargo, puede presentarse el hecho de que los elementos más cercanos para un determinado k no necesariamente representen a la clase del nuevo elemento, dado que para un $k + 1$ la clase del elemento se modifica. Para ello se asocia un peso al voto de cada elemento que participa en la elección de la clase. Normalmente se pondera la contribución de cada vecino en base a su distancia al nuevo elemento. Así, los elementos más cercanos tendrán un mayor peso. Esto se logra mediante la utilización del inverso del cuadrado de la distancia ($w_i = 1/d(y, x_i)^2$).

Con esta nueva modificación, la clasificación para un elemento x con atributos discretos se da mediante la Ecuación 4.4 y para atributos continuos por la Ecuación 4.5.

$$clase(x) = argmax_{c \in C} \sum_{i=1}^{k} w_i \delta(f(x_i), c) \tag{4.4}$$

$$clase(x) = \frac{\sum_{i=1}^{k} w_i f(x_i)}{\sum_{i=1}^{k} w_i} \tag{4.5}$$

Usando la ponderación según la distancia inversa no es necesario restringir la votación a los k elementos más cercanos, puede utilizarse la totalidad de los ejemplos almacenados. Se garantiza que los elementos lejanos influyen poco. En cualquier caso, siempre se recomienda que la implementación no utilice alguna comparación lineal debido a que resulta costosa computacionalmente.

Manejo de valores faltantes

En el caso de valores faltantes, es decir si un valor de una atributo A para el elemento X_1 no se conoce (o bien para otro elemento X_2 con el cual se compara), se asume que entre ambos elementos existe la diferencia máxima posible. Por ejemplo si se ha llevado a cabo una normalización al rango $[0, 1]$ y se trata de un dato discreto, se toma el valor de 1 si por lo menos alguno de los valores esta ausente. En el caso de un dato númerico si ambos valores son desconocidos entonces nuevamente se asigna 1 como la diferencia entre ellos (el valor máximo), en el caso de que solamente uno de los valores se desconoce entonces se toma la diferencia como $|1 - v'|$ o $|0 - v'|$, cualquiera que sea la mayor entre éstas.

Un parámetro muy importante dentro de este algoritmo, es el número de vecinos a considerar para la similitud (k). Dicho parámetro se determina mediante prueba y error. Iniciando con $k = 1$ se puede ir aumentando el número de vecinos a considerar. En general, a mayor cantidad de elementos de entrenamiento, mayor será el valor de k (intuitivamente esto significa que la clasificación se basará en una mayor cantidad de elementos almacenados). Así, mientras el número de elementos de entrenamiento aumenta y $k = 1$, el rango de error no puede ser mayor que el doble del rango de error reportado por el clasificador bayesiano. Por el contrario, si k aumenta, entonces el rango de error se acerca al rango del clasificador bayesiano [10].

Este método utiliza comparaciones basadas en distancia para, intrínsecamente, asignar un peso equivalente a cada atributo. Por lo tanto, puede padecer de una baja precisión en la presencia de datos con ruido o atributos irrelevantes.

Por otro lado, este clasificador puede ser extremadamente lento cuando se clasifican los elementos del conjunto de prueba, Si D es el conjunto de prueba con $|D|$ elementos y $k = 1$, entonces se requieren por lo menos de $O(|D|)$ comparaciones para clasificar un nuevo elemento[6] .

Dado que el algoritmo k-NN permite que los atributos de los elementos sean simbólicos y numéricos, así como que haya atributos sin valor, el algoritmo para el cálculo de la distancia entre ejemplares se complica ligeramente. En el Listado 4.1 se muestra el pseudo código del algoritmo k vecinos más próximos.

4.4. Recapitulación

El método de los k vecinos más próximos está considerado como un buen representante de los clasificadores basados en instancias y es de gran sencillez conceptual. Se suele denominar método porque es el esqueleto de un algoritmo que admite el intercambio de la función de proximidad dando lugar a múltiples variantes, adaptándolo según a conveniencia. La función de proximidad puede decidir la clasificación de un nuevo elemento atendiendo a la clasificación del mismo o de la mayoría de los k elementos

[6] Existen mejoras donde llevando a cabo un ordenamiento de los datos, este número de comparaciones puede reducirse a $O(log(|D|))$.

Listado 4.1: Pseudo-código del algoritmo k vecinos más próximos

```
Entrada:
        D       // Datos de entrenamiento
        K       // Cantidad de vecinos a considerar
        t       // Elemento por clasificar
Salida:
        c       // Clase asignada a t
Algoritmo:
        N = 0;
        Para cada elemento d en D hacer
                si |N| < o = K entonces
                        N = N U d;
                sino
                        if Existe un elemento en N tal que sim(t,u) > o =
                            sim(t,d) entonces
                                N = N - u;
                                N = N U d;
        c = clase a la cual la mayoría de los u en N son clasificados
```

más cercanos (o semejantes a él). Admite también funciones de proximidad que consideran el costo de los atributos que intervienen, lo que permite, eliminar los atributos irrelevantes. Este método presenta un soporte importante a la presencia de elementos carentes de valor, sin embargo el estudio sobre su tolerancia a errores fuera de rangos válidos, aún no ha sido analizado a profundidad.

Debido a estas características, también ha sido elegido para analizar su comportamiento como clasificador en presencia de distintos tipos, porcentajes y distribución de errores en los datos que se le presentan como elementos entrenamiento.

CAPÍTULO 5

Experimentación

5.1. Introducción

La experimentación junto con los estudios basados en observaciones, son algunos de los principales métodos de investigación. En los experimentos se miden y comparan los efectos de distintos parámetros y generalmente se modifica el estado de las muestras y se observa el efecto de estas modificaciones. Es decir, la experimentación es un método científico de investigación que consiste en hacer operaciones y prácticas destinadas a demostrar, comprobar o descubrir fenómenos o principios.

La minería de datos y los algoritmos de aprendizaje utilizados para las distintas tareas de clasificación, se consideran aún ciencias experimentales dado que mucho del conocimiento obtenido de los algoritmos es mediante su implementación y análisis de su comportamiento sobre conjuntos de datos específicos. Es decir, pueden ejecutarse distintos algoritmos y cotejar los resultados obtenidos a fin de hallar el mejor para un conjunto de datos particular o bien, modificar continuamente los parámetros hasta encontrar aquellos que logran la mejor ejecución.

Una consecuencia de lo anterior, es interpretar que las conclusiones obtenidas son un tanto carentes de validez, sin embargo, existen argumentos que justifican la ejecución de este tipo de experimentación. El más importante consiste en que aún cuando se han desarrollado trabajos para fundamentar la idea de una "*base de datos general de experimentación*" [2], las ideas presentadas todavía no se concretan por lo cual los experimentos se deben realizar sobre cada conjunto de datos, además, la variedad de las aplicaciones y utilización de los distintos sistemas de Inteligencia de Negocios hacen necesario un ajuste continuo en los distintos parámetros que ocupa cada algoritmo de minería de datos.

En este capítulo se muestra un panorama general de los experimentos elaborados

para corroborar la propuesta de la presente tesis:

- *"Estimar el rango o porcentaje de error que se puede permitir en el conjunto de datos de entrenamiento para ser procesados posteriormente mediante diversas técnicas de minería de datos*[1] *de modo que este rango de error no influya en la precisión del modelo clasificador, y con ello, generar un criterio que advierta la necesidad de llevar a cabo un proceso de limpieza de datos".*

Posteriormente se desarrolla una descripción del ambiente y aplicaciones utilizadas para la realización y reporte de dichos experimentos. El capítulo continua con una breve descripción de los conjuntos de datos utilizados, así como una explicación de cada uno de los parámetros y condiciones dispuestas para cada algoritmo utilizado. Finalizando con una presentación, análisis y discusión de los resultados obtenidos.

5.1.1. Consideraciones generales

El objetivo general de los experimentos llevados a cabo es mostrar el impacto que tiene la existencia de errores sobre los distintos algoritmos de aprendizaje para los distintos métodos de clasificación presentados con anterioridad. Para ello se consideran las siguientes características de los errores[2] :

- Tipo.- Un dato se considera como un error al referir a valores fuera del dominio definido para el atributo en particular o bien a la presencia de datos faltantes[3] .
- Influencia.- La presencia de errores sobre distintos atributos influye de manera desigual debido a la importancia que posee cada uno de los atributos para los distintos algoritmos.
- Distribución.- En el caso de que los errores se refieran a valores fuera del dominio, estos valores exhiben cierta función de distribución. Para el caso de los errores debidos a la ausencia de valor, esta distribución no se presenta ya que la ausencia de valor siempre se representa por una marca indistinta, conocida como marca η o `NULL`.
- Grado de afectación.- Los errores pueden afectar a más de un atributo de manera simultánea y dependiendo del número de atributos afectados, será el grado de afectación que se observe de manera general.

Para determinar la precisión de cada uno de los modelos de clasificación y analizar su comportamiento, se determinó utilizar únicamente como medida de evaluación al rango general de reconocimiento de cada modelo, las razones de esta decisión se basan en los detalles presentados en la Sección 1.8.

[1] En particular: clasificador ingenuo de Bayes, árboles de decisión (C4.5), y k vecinos más próximos.

[2] Esta clasificación se basa parcialmente en [52] y ha sido adaptada para el contexto del presente trabajo.

[3] Etiquetas o marcas `NULL` dentro del ambiente de bases de datos.

5.2. Herramientas utilizadas

Los experimentos dependen de la ejecución e implementación de los distintos algoritmos de aprendizaje para tareas de clasificación, por lo cual resultó fundamental la elección de una herramienta que contuviera la implementación de todos ellos a fin de ser consistentes y no presentar desviaciones en los resultados debidos a problemas de programación. Durante la realización del presente trabajo, se llevó a cabo un estudio detallado de distintas opciones de código abierto (*OpenSource*), siendo las más relevantes:

- Knime: The Konstanz Information Miner[4]
- WEKA: Waikato Environment for Knowledge Analysis[5]
- Orange Canvas[6]

Los principales criterios tomados en cuenta para la elección de la herramienta consideraron cuatro características importantes: la eficiencia, la implementación de todos los algoritmos analizados, desarrollo bajo licencia de código libre y la disponibilidad de un ambiente de trabajo de fácil manejo. Por lo cual, para la realización de los experimentos se optó por la herramienta **Orange Canvas**. Por último, para la manipulación[7] de los datos se seleccionó el sistema manejador de bases de datos **PostgreSQL**.

Orange Canvas

Orange Canvas es una aplicación definida bajo un marco de trabajo basado en componentes. Lo cual brinda a los usuarios la facilidad de desarrollar sus propios componentes basándose en el lenguaje de programación **Python** (aunque algunos componentes se encuentran implementados con el lenguaje de programación C) y posteriormente utilizar estos nuevos componentes dentro del entorno de trabajo de la misma herramienta. Además, define un conjunto amplio de componentes elementales y permite la extensión de los mismos. Las principales características de Orange Canvas incluyen:

- Formatos de entrada/salida: Permite la lectura y escritura de archivos en múltiples formatos.
- Procesamiento previo: Integra herramientas para la selección de atributos, filtrado de atributos según su relevancia, selección de subconjuntos de datos, transformación de valores continuos en discretos y viceversa, entre otras.

[4] `www.knime.org`
Department of Computer and Information Science - Konstanz University, Germany.

[5] `www.cs.waikato.ac.nz/ml/weka`
Department of Computer Science - University of Waikato, New Zealand.

[6] `http://www.ailab.si/orange`
Faculty of Computer and Information Science - University of Ljubljana, Slovenia.

[7] Inserción de errores.

- Modelos predictivos: Implementación de algoritmos de tareas de clasificación y predicción, por ejemplo árboles de clasificación[8] , clasificador ingenuo de Bayes, k-NN, clasificador por mayoría, máquinas de soporte vectorial para clasificación, regresión logística y clasificadores basados en reglas.

- Métodos de combinación de modelos: Incluye la implementación de las técnicas de Aumento (1.7.2), Consenso (1.7.2), y Bosque de árboles.

- Métodos de descripción de datos: Integra distintos métodos de visualización, análisis de distribución de datos, proyección linear, agrupamiento jerárquico, agrupamiento por k-medias y mapas auto-organizados.

- Técnicas de validación de modelos: Incluye distintas técnicas como validación cruzada, medición de la precisión de modelos de clasificación, análisis AUC ("*Area under Receiver Operating Characteristic*"), análisis de "*Lift Chart*" y análisis de "*Calibration Plot*".

PostgreSQL

"PostgreSQL" es un sistema manejador de bases de datos objeto-relacional que fue desarrollado por la Universidad de California en Berkley. Es de libre distribución y se encuentra bajo licencia "Berkeley Software Distribution"(BSD). Para el presente trabajo se utilizó la versión 8.2.4 que se ejecuta sobre múltiples distribuciones del sistema operativo **Linux** y **Windows XP™ Professional**.

Actualmente se ha posicionado como una elección contra opciones comerciales ya que contiene características importantes y cumple con gran parte del estándar **SQL2003** [31], es distribuido bajo licencia BSD la cual permite trabajar sin tener que preocuparse por pagar licencias y además permite la extensión de sus funcionalidades mediante la programación de distintas funciones definidas por el usuario. En particular con el lenguaje de programación interno llamado *plpgpsql*.

La elección del sistema manejador de bases de datos se hizo basándose principalmente en los siguientes puntos:

- Gratuidad para el desarrollo de aplicaciones en el ámbito académico.

- Gran cantidad de documentación disponible.

- Múltiples características presentes para el desarrollo.

Para la generación de los distintos reportes de los experimentos, se utilizó la herramienta **OpenCalc** contenida en el paquete **OpenOffice**[9] .

[8] Incluye creación de árboles de modo interactivo, implementación del algoritmo C4.5 y uno propio.
[9] `http://es.openoffice.org`

5.3. Descripción de los conjuntos de datos

Los conjuntos de datos utilizados durante la experimentación del presente trabajo, han sido obtenidos del "**Irving Repository of Machine Learning Databases**" de la Universidad de California (UCI Machine Learning) [1]. Se han seleccionado las siguientes bases de datos: "*Car*", "*Iris*", "*Shuttle*", "*Tennis*", "*Balance Scale*", "*Vehicle*". Adicionalmente, dos conjuntos de datos se obtuvieron del repositorio de datos "**Wiley**" [74]: "*Voting Records*" y "*CollegePlans*". Estos repositorios han sido ampliamente utilizados para la experimentación dentro del área de aprendizaje de máquinas, siendo citados en más de 1000 artículos.

Dado que los tipos de problemas en que se enfoca el presente trabajo son de clasificación, todas las bases de datos utilizadas se componen de diversas variables o atributos de entrada y uno solo de salida (atributo clase); este último puede tomar sólo uno de sus valores por cada instancia (clases mutuamente excluyentes). Por otro lado, el dominio de valores de cada una de los atributos que componen las bases de datos son discretos y en aquellos casos en los cuales son continuos, se han hecho discretos mediante la utilización de las herramientas de Orange Canvas (por medio de un proceso de discretización basado en entropía) y han quedado representados mediante rangos.

En la Tabla 5.1 se presenta un resumen de las principales propiedades de cada conjunto de datos.

No.	Nombre	Atributos	Clases	Instancias entrenamiento	Instancias prueba	Instancias totales
1	Car	6	4	1210	518	1728
2	Iris	4	3	100	50	150
3	Shuttle	9	7	5600	2400	8000
4	Diabetes	8	2	537	321	768
5	Balance Scale	4	3	438	187	625
6	Credit	15	2	483	207	690
7	Voting Records	20	2	306	130	436
8	CollegePlans	6	2	5600	2400	8000

Tabla 5.1: Resumen de las características de los conjuntos de datos utilizados.

Donde la columna:

- *Nombre* indica el nombre mediante el cual es referenciado dicho conjunto de datos en el presente trabajo.
- *Atributos* indica la cantidad de atributos que posee el conjunto de datos original.
- *Clases* indica cuantos valores puede tomar el atributo clase.

- *Instancias entrenamiento (IE)* representa la cantidad de registros que abarcan el conjunto datos de entrenamiento.
- *Instancias prueba (IP)* representa la cantidad de registros que abarcan el conjunto datos de prueba.
- *Instancias totales (IT)* representa la cantidad de registros que abarcan el conjunto datos total ($IT = IE + IP$[10]).

A continuación se describe con mayor nivel de detalle cada una de los conjuntos de datos utilizadas. Para cada uno de ellos se describe su dominio, los atributos que los componen con sus correspondientes rangos de valores posibles o dominios y la distribución de las clases tanto en el conjunto de casos de entrenamiento así como en el de prueba.

5.3.1. Descripción detallada

"Car"

Este conjunto de datos se obtuvo de un sencillo modelo jerárquico de decisión, originalmente desarrollado para la demostración de DEX [3]. El modelo evalúa los automóviles de acuerdo a una estructura conceptual predeterminada y determina si el automóvil es aceptable o no. Dicha estructura se muestra en el Listado 5.1.

Como se observa, dicha evaluación contiene tres conceptos intermedios: PRICE, TECH y COMFORT. Para la evaluación final, dichos conceptos han sido eliminados del conjunto de datos final. El conjunto de datos original no presenta valores faltantes. Detalles de los atributos se presentan en la Tabla 5.2 y la distribución de los valores o etiquetas del atributo clase se encuentran en la Tabla 5.3.

Atributo	Dominio de valores	Valores
buying	Discreto	v-high, high, med, low
maint	Discreto	v-high, high, med, low
doors	Discreto	2, 3, 4, 5-more
persons	Discreto	2, 4, more
lug_boot	Discreto	small, med, big
safety	Discreto	low, med, high
Clase	Discreto	unacc, acc, good, v-good

Tabla 5.2: Detalles de los atributos para el conjunto de datos "Car".

[10] La división de los datos se llevó con la intención de evitar el fenómeno de sobre entrenamiento (Sección 1.6.3), utilizando para ello un 70 % del conjunto original como datos de entrenamiento y el restante 30 % como datos de prueba.

Listado 5.1: Estructura conceptual de evaluación para el conjunto de datos “Car”

```
CAR                     aceptación del automóvil
. PRICE                 precio final total
. . buying              precio de compra
. . maint               precio de mantenimiento
. TECH                  características técnicas
. . COMFORT             comodidad
. . . doors             número de puertas
. . . persons           número de pasajeros
. . . lug_boot          tamaño de la cajuela de equipaje
. . safety              seguridad estimada del automóvil
```

Clase	Elementos	%
unacc	1210	70.02
acc	384	22.22
good	69	3.99
v-good	65	3.76

Tabla 5.3: Distribución de las clases para el conjunto de datos “Car”.

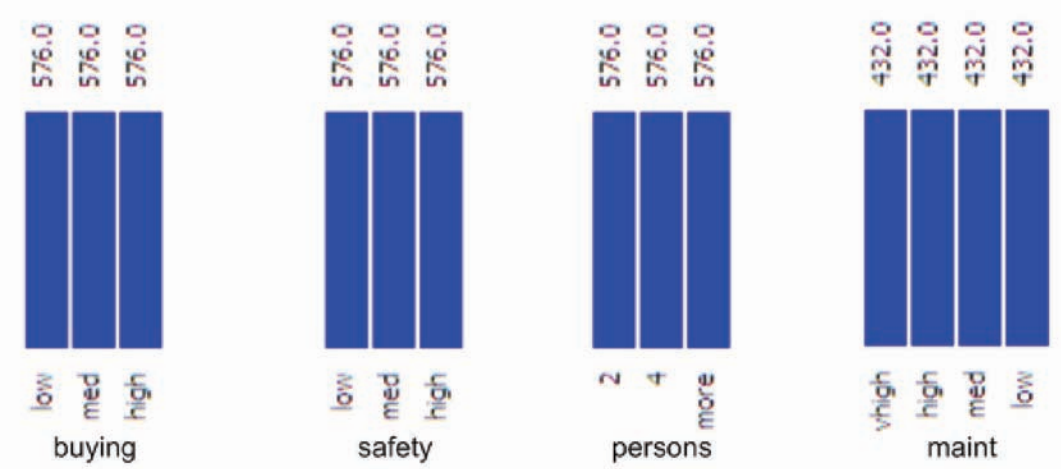


Figura 5.1: Distribución de los valores de los atributos “buying”, “safety”, “persons” y “maint”.

"Iris"

Es el conjunto de datos más conocido dentro del ámbito del reconocimiento de patrones. Presentado en el artículo [23]. Contiene tres clases de 50 instancias cada una, cada clase hace referencia a un tipo de planta.

Una clase no es linealmente separable de las otras dos. Los datos obtenidos difieren del conjunto original debido a correcciones que se realizaron sobre estos datos. El conjunto no presenta valores ausentes. Detalles de los atributos se presentan en la Tabla 5.4. La distribución de los valores o etiquetas del atributo clase se encuentran detallados en la Tabla 5.5.

Atributo	Dominio de valores	Valores
sepal length	Continuo	{4.3, . . . , 7.9}
sepal width	Continuo	{2.0, . . . , 4.4}
petal length	Continuo	{1.0, . . . , 6.9}
petal width	Continuo	{0.1, . . . , 2.5}
Clase	Discreto	Iris Setosa, Iris Versicolour, Iris Virginica

Tabla 5.4: Detalles de los atributos para el conjunto de datos "Iris".

Clase	Elementos	%
Iris Setosa	50	33.33
Iris Versicolour	50	33.33
Iris Virginica	50	33.33

Tabla 5.5: Distribución de las clases para el conjunto de datos "Iris".

"Shuttle"

Conjunto de datos donado por el Departamento de Ciencias de la Computación de la Universidad de Sydney, N.S.W., Australia. Esta base de datos se utilizó en el proyecto Europeo "**StatLog**", que involucraba la comparación de desempeño entre algoritmos de aprendizaje de máquina, estadísticos, de redes neuronales y otros, al aplicarlos sobre conjuntos de datos reales.

El conjunto no presenta valores ausentes. Detalles de los 9 atributos se presentan en la Tabla 5.6. La distribución de los valores o etiquetas del atributo clase se encuentran detallados en la Tabla 5.7, donde se aprecia que aproximadamente 80 % de los datos pertenece a la clase 1.

En los conjuntos de datos que se distribuyen, el atributo clase se haya codificado de la siguiente forma para cada una de las clases posibles: class0 – *Rad Flow*, class1 – *Fpv Close*, class2 – *Fpv Open*, class3 – *High*, class4 – *Bypass*, class5 – *Bpv Close*, y class6 – *Bpv Open*.

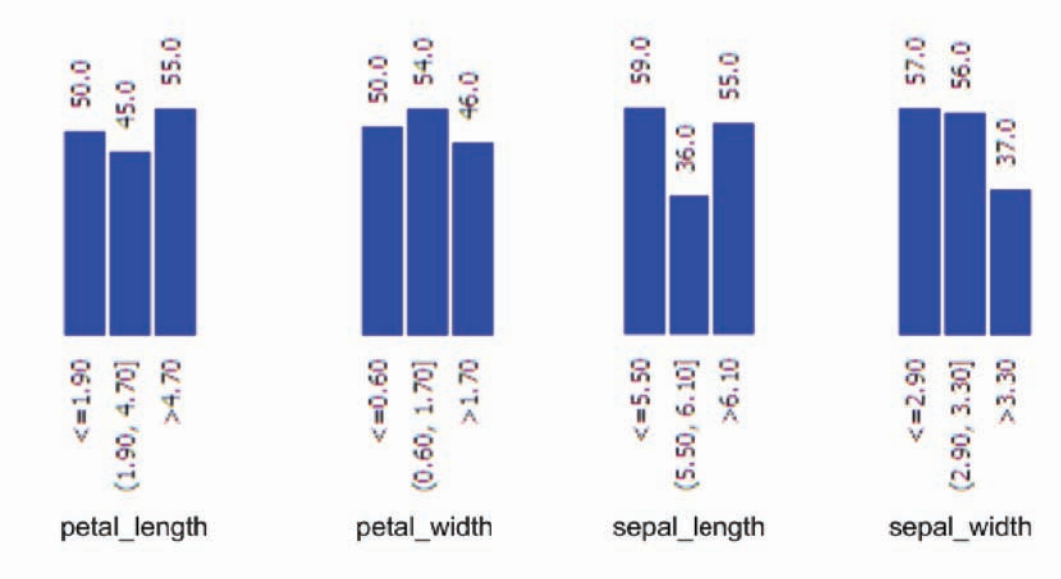


Figura 5.2: Distribución de los valores de los atributos "petal length", "petal width", "sepal length" y "sepal width".

Atributo	Dominio de valores	Valores
A1	Continuo	{-12,500, . . . , 12,500}
A2	Continuo	{-12,500, . . . , 12,500}
A3	Continuo	{-12,500, . . . , 12,500}
A4	Continuo	{-12,500, . . . , 12,500}
A5	Continuo	{-12,500, . . . , 12,500}
A6	Continuo	{-12,500, . . . , 12,500}
A7	Continuo	{-12,500, . . . , 12,500}
A8	Continuo	{-12,500, . . . , 12,500}
A9	Continuo	{-12,500, . . . , 12,500}
dtype	Discreto	class0, class1, class2, class3 class4, class5, class6

Tabla 5.6: Detalles de los atributos para el conjunto de datos "Shuttle".

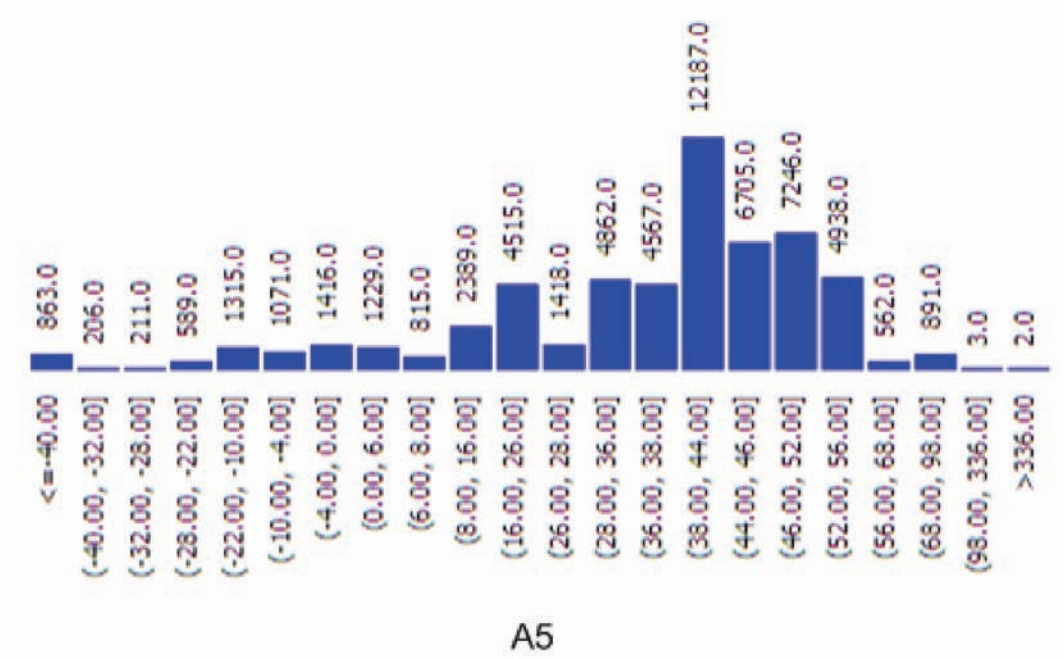


Figura 5.3: Distribución de los valores del atributo "A5".

Por último, es importante señalar que este conjunto de datos se distribuye con los conjuntos de datos de entrenamiento y de prueba de manera determinada.

Clase	Descripción	Elementos Entrenamiento	%	Elementos Prueba	%
1	Rad Flow	34108	78.41	11478	79.16
2	Fpv Close	37	0.09	13	0.09
3	Fpv Open	132	0.30	39	0.27
4	High	6748	15.51	2155	14.86
5	Bypass	2458	5.65	809	5.58
6	Bpv Close	6	0.01	4	0.03
7	Bpv Open	11	0.03	2	0.01

Tabla 5.7: Distribución de las clases para el conjunto de datos "Shuttle".

"Diabetes"

La base de datos refleja el resultado de un estudio llevado a cabo sobre una población de nativos del estado de Arizona en Estados Unidos de Norteamérica. Es una base de datos generada por el *National Institute of Diabetes and Digestive and Kidney Diseases*, y es muy utilizada en los trabajos sobre aprendizaje de máquinas. El diagnóstico se lleva a cabo sobre un atributo clase de dos valores, para determinar según criterios de la organización mundial de salud si el paciente presenta signos de diabetes. Las

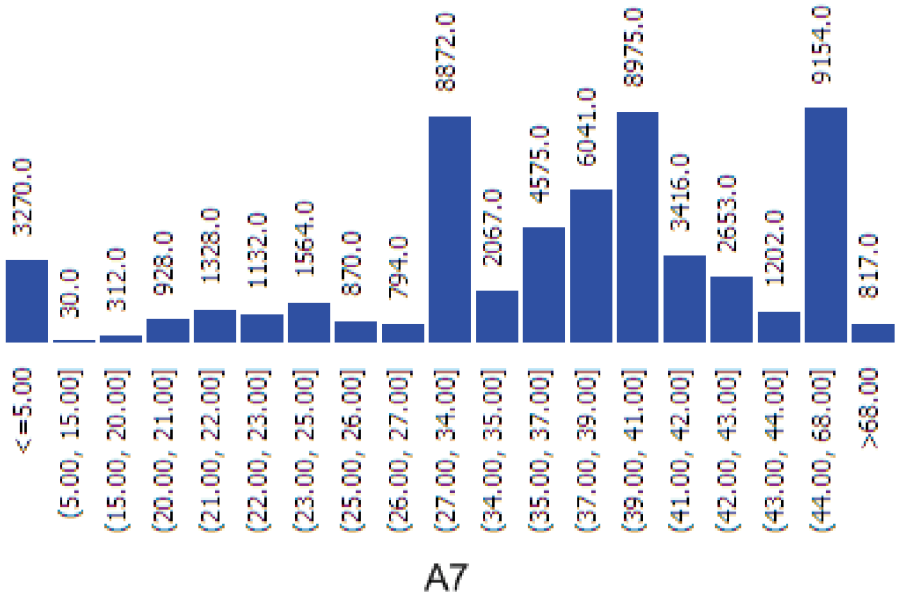


Figura 5.4: Distribución de los valores del atributo "A7".

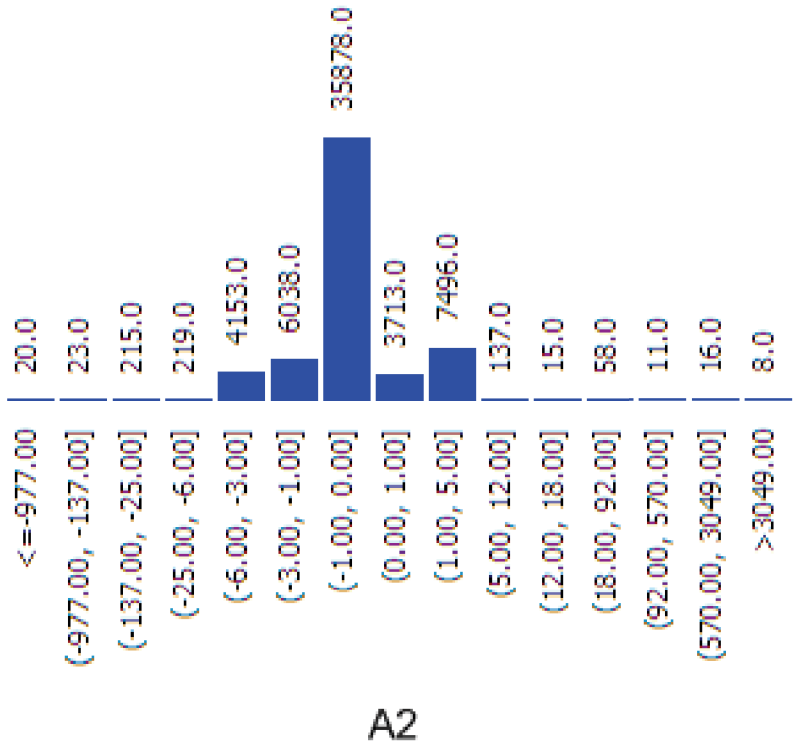


Figura 5.5: Distribución de los valores del atributo "A2".

características que se analizaron fueron las siguientes: número de veces que ha tenido un embarazo, concentración de glucosa, presión diastólica, grueso de la piel debajo del Triceps, nivel de insulina, índice de masa corporal (IMC), *Diabetes pedigree function* y la edad.

Dentro del conjunto de datos se aplicaron ciertas restricciones para formar parte de la muestra que se analizó. En particular se manejan las siguientes:

- Todos los pacientes son mujeres.
- La edad mínima fue de 21 años.
- Tener ascendencia directa de los indígenas Pima.

La clase 1 determina como positivo para diabetes (padece la enfermedad). El conjunto de datos que distribuye, presenta un proceso de normalización para los atributos característica. El conjunto no presenta valores ausentes. Detalles de los atributos se presentan en la Tabla 5.9 y la distribución de los valores o etiquetas del atributo clase se encuentran detallados en la Tabla 5.8.

Clase	Elementos	%
Class 0	500	65.10
Class 1	268	34.9

Tabla 5.8: Distribución de las clases para el conjunto de datos “Diabetes”.

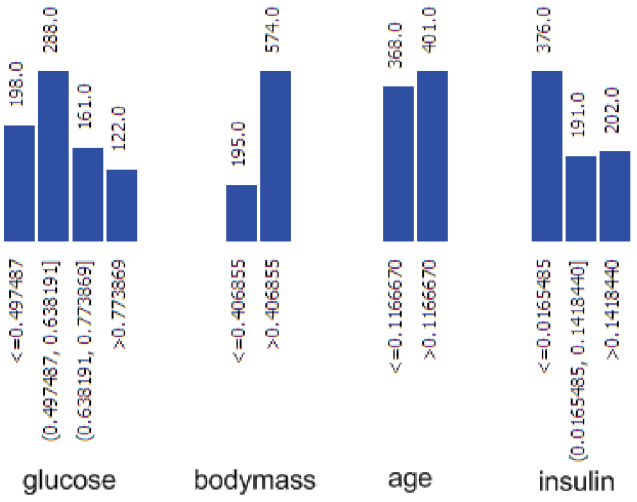


Figura 5.6: Distribución de los valores de los atributos “glucose”, “BodyMass”, “Age” e “Insulin”.

Atributo	Dominio de valores	Valores
Pregnant	Continuo	{0.0, ..., 1.0}
Glucose	Continuo	{0.0, ..., 1.0}
Pressure	Continuo	{0.0, ..., 1.0}
Thickness	Continuo	{0.0, ..., 1.0}
Insulin	Continuo	{0.0, ..., 1.0}
BodyMass	Continuo	{0.0, ..., 1.0}
DiabetesPF	Continuo	{0.0, ..., 1.0}
Age	Continuo	{0.0, ..., 1.0}
Diabetes?	Discreto	class0, class1

Tabla 5.9: Detalles de los atributos para el conjunto de datos "Diabetes".

"Balance Scale"

Este conjunto de datos se generó para modelar resultados experimentales psicológicos. Cada elemento es clasificado dependiendo si presenta una tendencia en los brazos de la balanza, ya sea con una tendencia hacia la derecha, a la izquierda o se encuentra balanceado. Los atributos son el peso izquierdo, la distancia izquierda, el peso derecho y la distancia derecha. La forma correcta de encontrar la clase es obteniendo el valor máximo entre ($distancia{-}izquierda * peso{-}izquierdo$) y ($distancia{-}derecha * peso{-}derecho$). Si son equivalentes, se determina que es balanceado.

El conjunto de datos consta de 4 atributos características mas uno clase, con 625 elementos y sin ningún valor faltante. Detalles de los atributos se presentan en la Tabla 5.10. La distribución de los valores o etiquetas del atributo clase se encuentran detallados en la Tabla 5.11.

Atributo	Dominio de valores	Valores
Clase	Discreto	L, B, R
Peso izquierdo	Discreto	1, 2, 3, 4, 5
Distancia izquierda	Discreto	1, 2, 3, 4, 5
Peso derecho	Discreto	1, 2, 3, 4, 5
Distancia derecha	Discreto	1, 2, 3, 4, 5

Tabla 5.10: Detalles de los atributos para el conjunto de datos "Balance Scale".

Clase	Elementos	%
L	288	46.08
B	49	7.84
R	288	46.08

Tabla 5.11: Distribución de las clases para el conjunto de datos "Balance Scale".

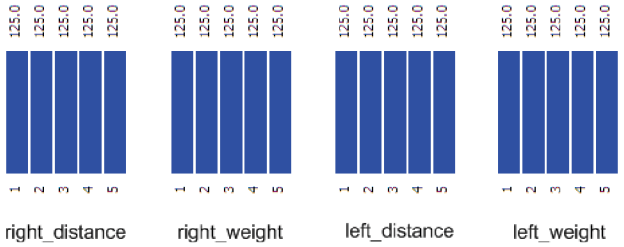


Figura 5.7: Distribución de los valores de los atributos “right_distance”, “right_weight”, “left_distance” e “left_weight”.

“Credit”

El conjunto de datos que mantiene la información de solicitudes de tarjetas de crédito de ciertos habitantes de Japón. Las instancias representan aquellas personas que obtuvieron o no la aprobación de la tarjeta en cuestión. Todos los nombres y valores han sido modificados en el conjunto de datos original cambiándolos por símbolos que carecen de significado alguno, lo anterior con la idea de mantener la confidencia de la información personal.

El conjunto de datos posee una mezcla variada de atributos discretos (9) y continuos (6), éstos últimos con algunos valores nominales grandes y otros con valores pequeños. También existen 35 instancias con datos faltantes que representan un 5 % del total, con los datos faltantes en los siguientes atributos: A1 con 12, A2 con 12, A4 con 6, A5 con 6, A6 con 9, A7 con 9 y A14 con 13.

El conjunto de datos consta de 15 atributos características mas uno clase y con 690 elementos. Detalles de los atributos se presentan en la Tabla 5.12. La distribución de los valores o etiquetas del atributo clase se encuentran detallados en la Tabla 5.13.

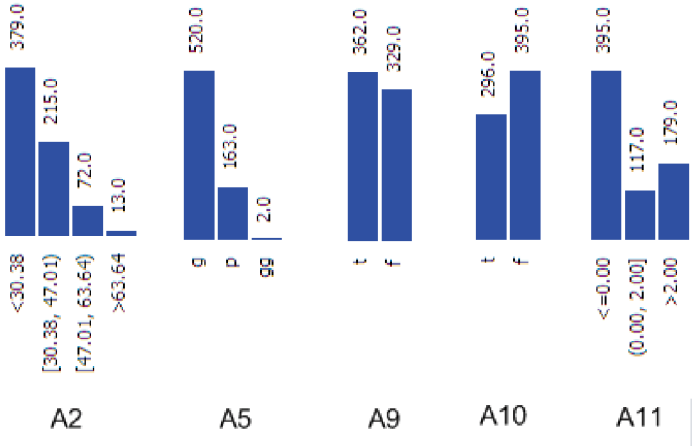


Figura 5.8: Distribución de los valores de los atributos “A2”, “A5”, “A9”,“A10” e “A11”.

Atributo	Dominio de valores	Valores
A1	Discreto	b, a
A2	Continuo	{13.75, . . . , 80.25}
A3	Continuo	{0.0, . . . , 9.96}
A4	Discreto	u, y, l, t
A5	Discreto	g, p, gg
A6	Discreto	c, d, cc, i, j, k, m, r, q, w, x, e, aa, ff
A7	Discreto	v, h, bb, j, n, z, dd, ff, o
A8	Continuo	{0.0, . . . , 9.46}
A9	Discreto	t, f
A10	Discreto	t, f
A11	Continuo	{9, . . . , 11}
A12	Discreto	t, f
A13	Discreto	g, p, s
A14	Continuo	{0, . . . , 980}
A15	Continuo	{0, . . . , 9800}
A16	Discreto	+, -

Tabla 5.12: Detalles de los atributos para el conjunto de datos "Credit".

Clase	Elementos	%
+	307	44.5
-	383	55.5

Tabla 5.13: Distribución de las clases para el conjunto de datos "Credit".

"Voting Records"

Conjunto de datos que contiene una selección de votos en la cámara de Representantes de los Estados Unidos de Norteamérica en 2002, sobre distintas problemáticas y las propuestas para cada una de ellas. Adicionalmente a los resultados de cada problemática (con tres posibles valores: "Y", "N" o "NULL" para las abstenciones), se incluyen los nombres y el partido político al cual representa cada uno de los integrantes.

Detalles de los atributos se presentan en la Tabla 5.15. La distribución de los valores o etiquetas del atributo clase se encuentran detallados en la Tabla 5.14.

Clase	Elementos	%
D(emocrat)	211	49
R(epublic)	223	51

Tabla 5.14: Distribución de las clases para el conjunto de datos "Voting Records".

Atributo	Dominio de valores	Valores
Reacondicionamiento de las finanzas de la campaña	Discreto	Y, N
Ventajas del desempleo y de impuesto	Discreto	Y, N
Resolución fiscal 2003	Discreto	Y, N
Reducciones de impuestos permanentes	Discreto	Y, N
Estampillas del alimento	Discreto	Y, N
Desperdicios nucleares	Discreto	Y, N
Defensa fiscal 2003	Discreto	Y, N
Abortos indiscriminados	Discreto	Y, N
Defense Authorization Recommitment	Discreto	Y, N
Renovación del bienestar	Discreto	Y, N
Abrogación del impuesto de estado	Discreto	Y, N
Impuesto a parejas	Discreto	Y, N
Prohibición del aborto en últimas etapas	Discreto	Y, N
Seguridad para miembros de la Unión	Discreto	Y, N
Seguridad para miembros de la función pública	Discreto	Y, N
Protecciones de Whistleblower	Discreto	Y, N
Comercio andino	Discreto	Y, N
Denegaciones del servicio del aborto	Discreto	Y, N
Concesiones médicas de negligencia	Discreto	Y, N
Apoyo militar para resolución de UN	Discreto	Y, N
Partido	Discreto	D, R

Tabla 5.15: Detalles de los atributos para el conjunto de datos “Voting Records”.

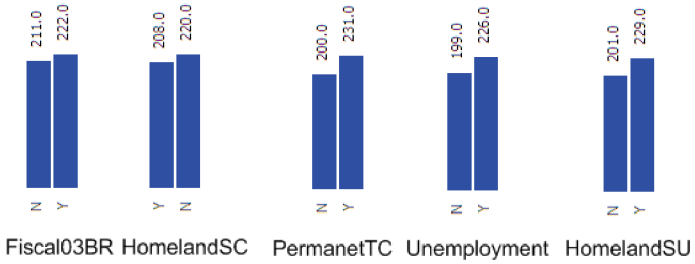


Figura 5.9: Distribución de los valores de los atributos "Fiscal03BR", "HomelandSC", "PermanetTC","Unemployment" y "HomelandSU".

"College Plan"

Este conjunto de datos condensa datos reales del colegio Midwest, y trata sobre las intenciones de los alumnos para ingresar a una Universidad basándose en un número de factores. Estos factores se detallan en la Tabla 5.17. El conjunto original contiene todos los atributos como valores discretos, con los atributos "*ingreso*" e "*IQ*" discretizados en rangos. La distribución de los valores o etiquetas del atributo clase se encuentran detallados en la Tabla 5.16.

Clase	Elementos	%
Plans to attend	4400	55
Does not plan to attend	3600	45

Tabla 5.16: Distribución de las clases para el conjunto de datos "College Plan".

Atributo	Dominio de valores	Valores
ID	Discreto	{1, . . . , 9000}
Género	Discreto	Male, Female
Ingresos de los padres	Discreto	{4,500, . . . , 82,390}
IQ	Discreto	{0, . . . , 140}
Animo de los padres	Discreto	Encouraged, Not Encouraged
Planes de asistencia	Discreto	Plans to attend, Doesn't plan to attend

Tabla 5.17: Detalles de los atributos para el conjunto de datos "College Plan".

5.3.2. Metodología utilizada

El proceso que se realizó durante los experimentos con los conjuntos de datos consistió en un ciclo interactivo el cual se detalla a continuación.

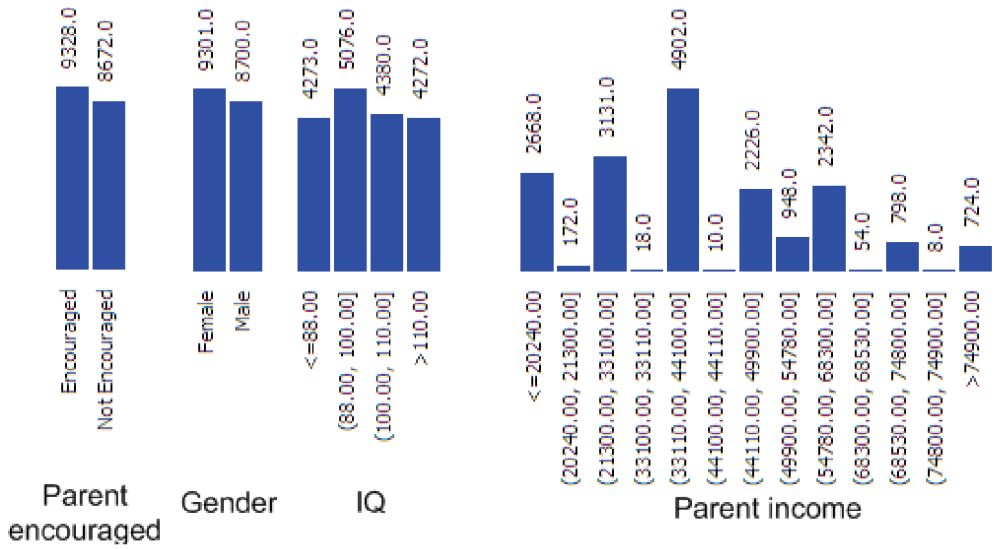


Figura 5.10: Distribución de los valores de los atributos "ParentEncouragement", "ParentIncome", "IQ" y "Gender".

1. Cargar cada uno de los conjuntos de datos dentro de una tabla relacional en una base de datos en el sistema manejador de bases de datos (**SMBD**) PostgreSQL v 8.2.4 [11] . Para lograr esto algunos de los conjuntos de datos han tenido que ser modificados en su formato original a fin de ser reconocidos por las instrucciones de carga del SMBD.

2. Utilizar el lenguaje de programación *plpgpsql* del SMBD para programar las distintas funciones de distribución de probabilidad[12] (**fdp**) necesarias para introducir posteriormente los errores en cada uno de los conjuntos de datos.

3. Mediante el software Orange Canvas, determinar cuales son los atributos más significativos de cada conjunto de datos utilizando la métrica de Ganancia de Información (Sección 3.2.1).

4. Para cada conjunto de datos se generó una nueva tabla donde se introdujeron errores mediante la combinación de los siguientes parámetros:

   a) Tipo de error.- Se definen dos tipos de errores: Valores fuera del conjunto válido de valores o la ausencia de valor (`NULL`) (Sección 5.1.1). La presencia de cada tipo de error sobre un conjunto de datos es excluyente.

[11] `http://www.postgresql.org`
[12] Esto se logró por medio de funciones definidas por el usuario (**UDF´s**).

b) Distribución de error.- Se definieron las funciones de distribución de probabilidad: constante, uniforme y geométrica. Dichas distribuciones se han elegido debido a que representan el comportamiento más general de los valores en los errores. Es importante remarcar que en el caso de los errores por ausencia de valor (marcas `NULL`), la distribución de dichos valores carece de sentido.

c) Porcentaje de error.- Se determina introducir errores entre el 1 % y el 10 %, en intervalos de 5 %. Debido a las caracteristicas de los experimentos, se consideran las combinaciones de errores entre más de un atributo de modo aditivo, es decir el total del porcentaje de error es independiente del número de atributos afectados.

Esta generación e introducción de errores se llevo a cabo de manera aleatoria, garantizando así que no se presenta un sesgo en la forma en la cual se afectan los datos.

5. Para cada nueva tabla (que representa un nuevo conjunto de datos con determinado error), se extraen los datos del SMBD a un archivo separado por tabuladores.

   a) Cargar estos archivos dentro de Orange Canvas.

   b) Dividir cada conjunto de datos en dos subconjuntos. Uno de entrenamiento (aproximadamente 2/3 del total) y otra de validación o prueba (con los datos restantes).

   c) Procesar conjunto de datos de entrenamiento mediante el respectivo algoritmo de clasificación (árbol C4.5, clasificador ingenuo de Bayes y los k vecinos más cercanos) con los parámetros por omisión que sugiere el software Orange Canvas.

   d) Analizar los cambios de la precisión para cada algoritmo de clasificación mediante el conjunto de prueba.

   e) Registrar la precisión de cada ejecución y la disminución que se observó en ésta.

   f) Plasmar el resultado en una gráfica para cada nueva medición de la precisión obtenida.

6. Registrar la precisión de cada conjunto de datos sin errores y comparar con cada uno de los registros de los conjuntos de datos con errores.

Los pasos 1 y 2, son necesarios debido a que los conjuntos de datos originales no contienen datos con errores o bien, la ausencia de estos.

El paso 3 es importante debido a que el número de atributos suceptibles de la introducción de errores es amplio y algunos de los algoritmos de clasificación utilizados, los contemplan dependiendo de la información que aportan. Por lo cual se han utilizado

dichos criterios para la selección de los atributos más importantes de cada conjunto de datos.

Dependiendo de la combinación de factores o características, se ha creado una nueva tabla para cada conjunto de datos afectado por los errores. Por ejemplo para el conjunto de datos "Iris" se creó una tabla que almacena los datos afectados en el atributo "*petal_width*" con errores distribuidos uniformemente en 1 % del total de datos. Distintos ejemplos de estos conjuntos de datos con errores se presentan en la Tabla 5.18.

Conjunto de datos	Atributos afectados	Tipo	Distribución	%
Car	buying	Fuera de rango	Normal	1
Car	{buying, persons}	Ausencia valor	N/A	5
Diabetes	{GL, BM, PG}	Fuera de rango	Geométrica	10
Voting Records	Comercio andino	Fuera de rango	Constante	10
Voting Records	Desperdicios nucleares	Ausencia valor	N/A	1

Tabla 5.18: Ejemplos de las tablas creadas y características de los errores introducidos.

La aplicación de cada uno de los algoritmos de clasificación, hace necesario el paso número 5. Durante este paso se extraen los datos del SMBD y se procesan mediante Orange Canvas. Una imagen del ambiente de trabajo creado para analizar los datos se presenta en la Imagen 5.11. Para el presente trabajo se han utilizado los siguientes componentes (se presentan también los valores de los distintos parámetros que cada componente ofrece):

- Data-File: Lee datos de un archivo cuyos datos se encuentran separados por tabuladores.
- Data-Discretize: Utilizado únicamente para conjuntos de datos con atributos de valores continuos. Transforma los datos continuos en discretos mediante rangos identificados. Los principales parámetros y que se han utilizado por sus valores por omisión son los siguientes:
  - Proceso de transformación en valores discretos: Por entropía.
  - Tratamiento de atributos individualmente: Utilizar valor por omisión.
- Data-Sampler: Permite seleccionar subconjuntos de datos de un conjunto original, de este modo se ha seleccionado 2/3 como conjunto de datos de entrenamiento y el resto como conjunto de datos de prueba. Los parámetros por defecto que se utilizaron son:
  - Opciones: Por estratos (seleccionado).
  - Tipo de muestreo: Aleatorio
  - Tamaño de muestra: Valor ajustado al 2/3 para cada conjunto de datos.

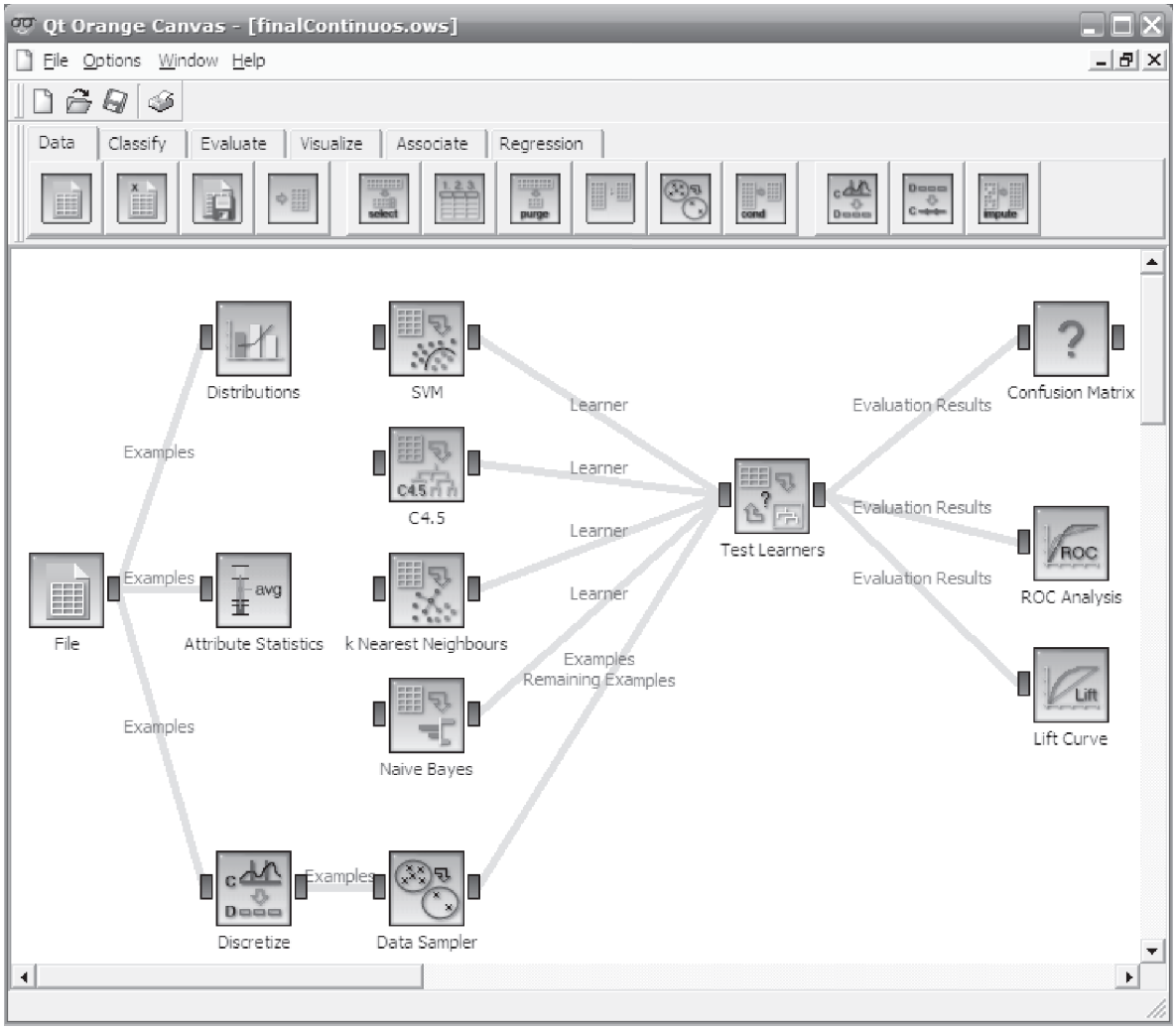


Figura 5.11: Presentación del ambiente de trabajo en Orange Canvas.

- Classify-C4.5: Implementación del algoritmo de clasificación C4.5. Los parámetros por defecto que se utilizaron son:
  - Medida de división de atributos: Ganancia de información (seleccionado).
  - Podado: Número mínimo de ejemplares en nodos hora: 2.
  - Nivel de confianza en podado posterior: 25.
- Classify-kNN: Implementación del algoritmo de clasificación k vecinos más cercanos. Los parámetros por defecto que se utilizaron son:
  - Número de vecinos a considerar: 5.
  - Métrica: Euclidiana.
  - Normalizar valores continuos (seleccionado).
- Classify-NaiveBayes: Implementación del algoritmo de clasificación ingenuo de Bayes. Los parámetros por defecto que se utilizaron son:
  - Estimación de probabilidad: Laplaciana[13] y Frecuencia Relativa [14]
  - Ajuste de umbrales (seleccionado para conjuntos de datos con dos valores en el atributo clase).
- Evaluate-TestLearners: Ejecuta los distintos algoritmos con el conjunto de datos de entrada. Presenta distintas estadísticas como la sensitividad, valor del área bajo el ROC, valor de información, etc. Estos datos también se analizan.
- Evaluate-ConfusionMatriz: Presenta los resultados del anterior componente de manera gráfica en una matriz de confusión. A partir de estos datos se aprecia con mayor precisión el comportamiento de cada algoritmo de clasificación en la presencia de los errores. Se genera una matriz para cada uno de ellos.

[13] Para conjuntos de datos con más de dos valores en el atributo clase.
[14] Para conjuntos de datos con dos valores en el atributo clase.

5.4. Resultados y análisis

Los resultados obtenidos de la experimentación son bastante amplios debido a la gran cantidad de pruebas que se llevaron a cabo (un aproximado de 2016 experimentos). A continuación se muestran estos resultados que surgen de aplicar la metodología anteriormente descrita a cada uno de los conjuntos de datos.

5.4.1. Resultados sobre el conjunto de datos "Car"

Los resultados obtenidos mediante la utilización de los algoritmos de clasificación ingenuo de Bayes, árbol de decisión C4.5 y k vecinos más próximos se resumen en las Tablas 5.20, 5.21 y 5.22 respectivamente.

En ellas es posible observar que la precisión del modelo clasificador presenta una disminución, sin embargo ésta resulta no ser mayor del 3.6 % para el clasificador ingenuo de Bayes (en la combinación de los atributos "safety" y "persons" afectados con 10 % de errores compartidos en rangos inválidos con una distribución constante), del 5 % para el árbol de decisión C4.5 (para el atributo "persons" afectado con 10 % de errores en rangos inválidos con una distribución uniforme) y del 2.4 % para los k vecinos más próximos (con el atributo "safety" afectado en 10 % de errores totales por ausencia de valor y sin alguna distribución en particular).

Para el caso del clasificador ingenuo de Bayes, se observa en la Tabla 5.20 que para este conjunto de datos la disminución de la precisión se encuentra relacionada de modo directo con el porcentaje de error que existe en el conjunto de datos, no así con el número de atributos afectados o la distribución de los valores de los errores (para esto último, el promedio de la disminución fue de 2.04 % con distribución uniforme, 2.14 % con distribución geométrica y 1.5 % con distribución constante).

Respecto al árbol de decisión C4.5, se determina nuevamente que el factor más importante para que se presente una disminución en la precisión del clasificador es el porcentaje total de error que contiene el conjunto de datos (Tabla 5.21). En el caso del k vecinos más próximos, se presenta en mismo patrón: el porcentaje de error resulta ser el elemento más importante de la clasificación. Por último, en la Tabla 5.19 se presenta una comparativa entre los 3 algoritmos de clasificación. En ella se observa que el clasificador ingenuo de Bayes posee una tolerancia mayor al error.

Algoritmo	Precisión inicial	Precisión mínima	Diferencia máxima
Ingenuo Bayes	0.8477	0.8111	3.6 %
Árbol C4.5	0.8382	0.7841	5.4 %
k - NN	0.8247	0.8002	2.4 %

Tabla 5.19: Comparativa de la precisión presentada por cada algoritmo sobre el conjunto de datos "Car".

5.4.2. Resultados sobre el conjunto de datos "Iris"

Los resultados obtenidos mediante la utilización de los algoritmos de clasificación ingenuo de Bayes, árbol de decisión C4.5 y k vecinos más próximos para el conjunto de datos "Iris", se resumen en las Tablas 5.24, 5.25 y 5.26 respectivamente.

En esta caso para el modelo clasificador donde se utilizó ingenuo Bayes, la disminución de la precisión fue mayor en comparación al anterior conjunto de datos, siendo de 7.3 % en la presencia de 10 % de error en el conjunto de datos (en la combinación de los atributos "Petal_Width"(PW), "Petal_Length"(PL) y "Sepal_Lenght"(SL) con error de rangos inválidos). Analizando el comportamiento general de los errores, se puede apreciar que para los errores fuera de rango la distribución de los valores no afecta significativamente a la disminución de la precisión (para una distribución uniforme el promedio de disminución fue de 5 %, para la distribución geométrica fue de 8 % y con la distribución constante fué nuevamente de 5 %) y en el caso de los errores por ausencia, el comportamiento es muy similar a la distribución constante (el promedio de disminución se ubicó en 7 %).

Para el modelo que utilizó el árbol de decisión C4.5, la mayor disminución observable de la precisión que se registró fue de 8.8 % (con el atributo "Petal_Width"(PW) afectado en 10 % de errores totales por ausencia de valor y sin alguna distribución en particular). En este caso, es importante observar que el atributo "Petal_Width"(PW) posee la ganancia de información mayor entre los atributos del conjunto de datos, por ello al afectar este atributo con un porcentaje de error elevado la precisión disminuye considerablemente[15] . Sin embargo, aún cuando dicho atributo participa en otras combinaciones donde se introducieron errores, al ser un error distribuido entre todos los atributos, éste no tuvo sesgo alguno. Por ejemplo en la combinación de 4 atributos afectados "Petal_Width"(PW), "Sepal_Length"(SL), "Petal_Length"(PL) y "Sepal_Width"(SW), cada uno presenta a lo más un 2.5 % de error. En este modelo otra vez se observa que en el caso de errores fuera de rango, la distribución de los valores no es un factor determinante para la reducción de la precisión del modelo (con una merma de 3 %, 4 % y 2 % para las distribuciones normal, geométrica y constante), siendo un factor más importante la ganancia de información que presenta cada atributo.

En el caso del modelo con el algoritmo de los k vecinos más próximos, la mayor disminución en la precisión fue de un 11 % (en la combinación de los atributos "Petal_Width"(PW), "Petal_Length"(PL) y "Sepal_Lenght"(SL) afectados con 10 % de errores de rangos inválidos compartidos y con una distribución geométrica).

Por último, en la Tabla 5.23 se presenta una comparativa entre los 3 algoritmos de clasificación. Se observa de nueva cuenta que el clasificador ingenuo de Bayes posee una tolerancia mayor al error que los otros dos métodos. Sin embargo, es preciso observar que aún cuando los errores se han introducido de manera porcentual, la influencia que tienen es mayor debido a lo pequeño que resulta el conjunto de datos analizados.

[15] Recordando que este criterio se maneja como el criterio de división por omisión dentro del algoritmo C4.5 (Sección 3.3.3).

Precisión original	0.8477		Atributos afectados:					
			safety (S)	buying (B)	persons (P)	SP	SPB	SPBM
Error por valor	Distribución Uniforme	1 %	0.8343	0.8441	0.8188	0.8285	0.8500	0.8540
		5 %	0.8285	0.8324	0.8478	0.8362	0.8170	0.8230
		10 %	0.8208	0.8420	0.8285	0.8285	0.8230	0.8210
	Distribución Geométrica	1 %	0.8478	0.8420	0.8208	0.8304	0.8400	0.8340
		5 %	0.8265	0.8475	0.8348	0.8344	0.8460	0.8420
		10 %	0.8209	0.8247	0.8227	0.8363	0.8310	0.8230
	Distribución Constante	1 %	0.8474	0.8397	0.8381	0.8458	0.8250	0.8460
		5 %	0.8478	0.8479	0.8285	0.8455	0.8380	0.8320
		10 %	0.8324	0.8477	0.8285	0.8111	0.8380	0.8340
Error por ausencia	N/A	1 %	0.8343	0.8382	0.8321	0.8382	0.8380	0.8440
		5 %	0.8317	0.8270	0.8263	0.8304	0.8230	0.8360
		10 %	0.8115	0.8211	0.8190	0.8212	0.8210	0.8130

Tabla 5.20: Resultados de la degradación de la precisión del clasificador ingenuo de Bayes para el conjunto de datos "Car". Los errores por valor presentan distintas funciones de probabilidad en la distribución de los mismos. Para errores por ausencia no se considera distribución alguna. M denota al atributo "maint".

Precisión original	0.8382		Atributos afectados:					
			safety (S)	buying (B)	persons (P)	SP	SPB	SPBM
Error por valor	Distribución Uniforme	1 %	0.8401	0.8265	0.8169	0.8439	0.8230	0.8650
		5 %	0.7977	0.8381	0.8362	0.8189	0.7820	0.8090
		10 %	0.8169	0.7841	0.7880	0.7957	0.8170	0.8250
	Distribución Geométrica	1 %	0.8381	0.8285	0.8074	0.8189	0.8320	0.8250
		5 %	0.8343	0.8361	0.8131	0.8119	0.8210	0.8230
		10 %	0.8188	0.8324	0.8285	0.8073	0.8110	0.8130
	Distribución Constante	1 %	0.8285	0.8301	0.8382	0.8304	0.8220	0.8220
		5 %	0.8271	0.8363	0.8134	0.8247	0.8360	0.8170
		10 %	0.8016	0.8324	0.8247	0.8204	0.8250	0.8050
Error por ausencia	N/A	1 %	0.8285	0.8150	0.8189	0.8054	0.8360	0.8230
		5 %	0.8135	0.8175	0.8210	0.8035	0.8120	0.8230
		10 %	0.8110	0.8129	0.8005	0.7980	0.8090	0.7990

Tabla 5.21: Resultados de la degradación de la precisión del clasificador basado en árboles de decisión C4.5 para el conjunto de datos "Car". Los errores por valor presentan distintas funciones de probabilidad en la distribución de los mismos. Para errores por ausencia no se considera distribución alguna. M denota al atributo "maint".

Precisión original	0.8247		Atributos afectados:					
			safety (S)	buying (B)	persons (P)	SP	SPB	SPBM
Error por valor	Distribución Uniforme	1 %	0.8381	0.8265	0.8169	0.8151	0.8536	0.8381
		5 %	0.8324	0.8478	0.8381	0.8246	0.8110	0.8228
		10 %	0.8305	0.8208	0.8112	0.8208	0.8247	0.8074
	Distribución Geométrica	1 %	0.8285	0.8420	0.8112	0.8189	0.8286	0.8304
		5 %	0.8207	0.8189	0.8266	0.8266	0.8285	0.8343
		10 %	0.8169	0.7881	0.8132	0.8169	0.8247	0.817
	Distribución Constante	1 %	0.8189	0.8205	0.8150	0.8208	0.8247	0.8149
		5 %	0.8227	0.8248	0.8266	0.8246	0.8266	0.8149
		10 %	0.8093	0.8247	0.8151	0.8247	0.8169	0.8073
Error por ausencia	N/A	1 %	0.8132	0.8242	0.8266	0.8113	0.8208	0.8153
		5 %	0.8078	0.8234	0.8220	0.8150	0.8095	0.8143
		10 %	0.8002	0.8122	0.8110	0.8030	0.8069	0.8043

Tabla 5.22: Resultados de la degradación de la precisión del clasificador basado en k vecinos más próximos para el conjunto de datos "Car". Los errores por valor presentan distintas funciones de probabilidad en la distribución de los mismos. Para errores por ausencia no se considera distribución alguna. M denota al atributo "maint".

Algoritmo	Precisión inicial	Precisión mínima	Diferencia máxima
Ingenuo Bayes	0.9778	0.9048	7.3 %
Árbol C4.5	0.9556	0.8667	8.8 %
k - NN	0.9778	0.8667	11 %

Tabla 5.23: Comparativa de la precisión presentada por cada algoritmo sobre el conjunto de datos "Iris".

5.4.3. Resultados sobre el conjunto de datos "Balance Scale"

Los resultados obtenidos mediante la utilización de los algoritmos de clasificación ingenuo de Bayes, árbol de decisión C4.5 y k vecinos más próximos para el conjunto de datos "Balance Scale", se resumen en las Tablas 5.28, 5.29 y 5.30 respectivamente.

En esta caso para el modelo clasificador donde se utilizó ingenuo Bayes, la disminución de la precisión presentó un ligero aumento en comparación al anterior conjunto de datos, ubicándose 7.9 % en la presencia de 10 % de error en el conjunto de datos, en la combinación de los atributos "Right_Distance"(RD) y "Left_Weight"(LW) con error de rangos inválidos y distribución constante. Del análisis general de los errores, se observa que para los errores fuera de rango la distribución de los valores no afecta significativamente a la disminución de la precisión (para una distribución uniforme el promedio de disminución fue de 5 %, para la distribución geométrica fue de 2 % y con la distribución constante fué nuevamente de 6 %) y en el caso de los errores por ausencia, el comportamiento fue mejor ya que el promedio en la reducción de la precisión se ubicó en 1 %.

Por otro lado, también es notorio que si bien existen conjuntos de datos en los cuales los errores afectan a más de un atributo, puede asumirse que este conjunto de datos resultase en un detrimento mayor de la precisión, empero no sucede así ya que el error al distribuirse entre los atributos genera, para el caso de ingenuo Bayes y con errores fuera de rango, conjuntos que poseen una probabilidad menor de pertenencia con respecto a los conjuntos creados a partir de los datos sin errores y los elementos del conjunto de prueba siguen obteniendo etiquetas de estos conjuntos válidos. En el caso de errores por ausencia, esta disminución de la precisión no se da en tal proporción debido a que el algoritmo evita el procesado de aquellos elementos con valores faltantes (Sección 2.3).

Para el modelo clasificador con el árbol de decisión C4.5 el comportamiento general es muy similar al ingenuo Bayes, empero aquí se aprecia que en la presencia de errores sobre más atributos, el árbol generado posee mayores nodos y hojas. Razón por la cual la precisión del modelo se ve más afectada[16] ya que los elementos del conjunto de prueba deben ajustarse a un conjunto mayor de posibles evaluaciones (Sección 3.2.1).

Otra consideración importante dentro de este conjunto de datos es con respecto al modelo basado en los k vecinos más próximos donde la mayor disminución en la

[16] Este fenómeno no solamente se da para este conjunto de datos.

Precisión original	0.9778		Atributos afectados:					
			PW	SL	PW_SL	PW_PL_SW	PW_PL_SL	PW_PL_SW_SL
Error por valor	Distribución Uniforme	1 %	0.9778	0.9778	0.9778	0.9778	0.9524	0.9556
		5 %	0.9556	0.9556	0.9556	0.9333	0.9333	0.9429
		10 %	0.9111	0.9333	0.9333	0.9143	0.9238	0.9333
	Distribución Geométrica	1 %	0.9778	0.9778	0.9333	0.9778	0.9778	0.9333
		5 %	0.9778	0.9778	0.9556	0.9333	0.9333	0.8889
		10 %	0.9333	0.9111	0.9333	0.8857	0.8667	0.8444
	Distribución Constante	1 %	0.9524	0.9524	0.9619	0.9619	0.9524	0.9429
		5 %	0.9333	0.9619	0.9238	0.9524	0.9333	0.9524
		10 %	0.9143	0.9143	0.9238	0.9524	0.9238	0.9333
Error por ausencia	N/A	1 %	0.9714	0.9619	0.9429	0.9619	0.9429	0.9143
		5 %	0.9333	0.9238	0.9238	0.9333	0.9143	0.8952
		10 %	0.9238	0.9143	0.9238	0.9048	0.9048	0.8381

Tabla 5.24: Resultados de la degradación de la precisión del clasificador ingenuo de Bayes para el conjunto de datos "Iris". Los errores por valor presentan distintas funciones de probabilidad en la distribución de los mismos. Para errores por ausencia no se considera distribución alguna (PW: Petal_Width, SL: Sepal_Length, PL: Petal_Length y SW: Sepal_Width; las combinaciones de atributos se denotan mediante "_").

Precisión original	0.9556		Atributos afectados:					
			PW	SL	PW_SL	PW_PL_SW	PW_PL_SL	PW_PL_SW_SL
Error por valor	Distribución Uniforme	1 %	0.9556	0.9556	0.9333	0.9556	0.9524	0.9556
		5 %	0.9556	0.9556	0.9778	0.9333	0.9238	0.9333
		10 %	0.9333	0.9556	0.9556	0.8952	0.9143	0.9048
	Distribución Geométrica	1 %	0.9778	0.9778	0.9556	0.9778	0.9778	0.9333
		5 %	0.9556	0.9333	0.9333	0.9778	0.9778	0.8222
		10 %	0.9333	0.9556	0.9143	0.9238	0.9333	0.8222
	Distribución Constante	1 %	0.9429	0.9524	0.9238	0.9333	0.9333	0.9333
		5 %	0.9238	0.9238	0.9238	0.9143	0.9143	0.9333
		10 %	0.9333	0.9524	0.9238	0.9238	0.9524	0.9143
Error por ausencia	N/A	1 %	0.9429	0.9333	0.9238	0.9333	0.9333	0.9048
		5 %	0.9048	0.9143	0.8571	0.9143	0.9238	0.8952
		10 %	0.8667	0.8952	0.8952	0.9238	0.8381	0.8762

Tabla 5.25: Resultados de la degradación de la precisión del clasificador basado en árboles de decisión C4.5 para el conjunto de datos "Iris". Los errores por valor presentan distintas funciones de probabilidad en la distribución de los mismos. Para errores por ausencia no se considera distribución alguna (PW: Petal_Width, SL: Sepal_Length, PL: Petal_Length y SW: Sepal_Width; las combinaciones de atributos se denotan mediante "_").

Precisión original	0.9778		Atributos afectados:					
			PW	SL	PW_SL	PW_PL_SW	PW_PL_SL	PW_PL_SW_SL
Error por valor	Distribución Uniforme	1 %	0.9556	0.9778	0.9333	0.9778	0.9429	0.9556
		5 %	0.9556	0.9778	0.9778	0.9333	0.9333	0.8952
		10 %	0.9778	0.9778	0.9556	0.9143	0.9048	0.9143
	Distribución Geométrica	1 %	0.9778	0.9778	0.9556	0.9778	0.9778	0.9333
		5 %	0.9556	0.9778	0.9778	0.9778	0.9778	0.8889
		10 %	0.9556	0.9333	0.8889	0.9048	0.8667	0.9048
	Distribución Constante	1 %	0.9524	0.9714	0.9429	0.9429	0.9429	0.9619
		5 %	0.9333	0.9714	0.9524	0.9238	0.9429	0.9429
		10 %	0.9429	0.9143	0.9143	0.9333	0.9429	0.9143
Error por ausencia	N/A	1 %	0.9524	0.9238	0.9238	0.9429	0.9619	0.9429
		5 %	0.9143	0.9333	0.9143	0.9048	0.9333	0.9238
		10 %	0.9333	0.9238	0.9143	0.9238	0.9048	0.9048

Tabla 5.26: Resultados de la degradación de la precisión del clasificador basado en k vecinos más próximos para el conjunto de datos "Iris". Los errores por valor presentan distintas funciones de probabilidad en la distribución de los mismos. Para errores por ausencia no se considera distribución alguna (PW: Petal_Width, SL: Sepal_Length, PL: Petal_Length y SW: Sepal_Width; las combinaciones de atributos se denotan mediante "_").

precisión fue de un 11.6 % (para el atributo "Right_Distance"(RD) afectado con 10 % de errores de rangos inválidos con una distribución constante). Por último, en la Tabla 5.27 se presenta una comparativa entre los 3 algoritmos de clasificación. En ella se observa el modelo basado en el árbol de decisión C4.5 posee una tolerancia mayor al error que los otros dos métodos. Sin embargo, no es, por mucho, el mejor modelo (los otros dos lo superan ampliamente en cuanto a la precisión inicial).

Algoritmo	Precisión inicial	Precisión mínima	Diferencia máxima
Ingenuo Bayes	0.8138	0.7340	7.9 %
Árbol C4.5	0.6225	0.6013	2.1 %
k - NN	0.7606	0.6437	11.6 %

Tabla 5.27: Comparativa de la precisión presentada por cada algoritmo sobre el conjunto de datos "Balance Scale".

5.4.4. Resultados sobre el conjunto de datos "College Plan"

Los resultados obtenidos mediante la utilización de los algoritmos de clasificación ingenuo de Bayes, árbol de decisión C4.5 y k vecinos más próximos para el conjunto de datos "College Plan", se resumen en las Tablas 5.32, 5.33 y 5.34 respectivamente.

Para el modelo con el clasificador ingenuo Bayes, fueron pocos los conjuntos de datos donde se presentó alguna disminución de la precisión, siendo la mayor de un 1.5 % (para el atributo "Animo de los padres"(PE) afectado con 10 % de errores de rangos inválidos con una distribución uniforme), dándose casos en los cuales la precisión mejoró. Del análisis general de los errores, se observa nuevamente que para los errores fuera de rango la distribución de los valores no afecta significativamente a la disminución de la precisión. Para una distribución uniforme el promedio de disminución fue de 0.4 %, para la distribución geométrica fue de -0.6 % y con la distribución constante permaneció idéntica y en el caso de los errores por ausencia, el comportamiento fue igual a la distribución geométrica, ubicándose en un -0.6 %.

Como en los anteriores casos, aún cuando se analizaron conjuntos de datos en los cuales se afectaron a más de un atributo, la disminución de la precisión no presentó un patrón distinto.

En el modelo clasificador con el árbol de decisión C4.5 el comportamiento general es muy parecido al ingenuo Bayes, empero aquí se aprecia que en la presencia de errores sobre más atributos, el árbol generado posee mayores nodos y hojas. Razón por la cual la precisión del modelo se ve más afectada ya que los elementos del conjunto de prueba deben ajustarse a un conjunto mayor de posibles evaluaciones (Sección 3.2.1). La mayor disminución en la precisión fue de un 12.2 % (en la combinación de los atributos "Animo de los padres"(PE), "Ingreso de los padres"(PI) y "Género" (GE) con error de ausencia de valor). Además se observó al igual que en otros conjuntos de datos, que al afectarse

Precisión original 0.8138 Atributos afectados:

			RD	LW	RD_LW	LD_RW	RD_LD_RW	LD_RD_LW
Error por valor	Distribución Uniforme	1 %	0.8191	0.8292	0.8084	0.7661	0.8031	0.8191
		5 %	0.8088	0.8033	0.8137	0.8192	0.8193	0.8296
		10 %	0.7664	0.7765	0.8461	0.8084	0.8297	0.8243
	Distribución Geométrica	1 %	0.7340	0.8040	0.7977	0.7980	0.8090	0.8192
		5 %	0.8087	0.8085	0.7707	0.7770	0.7919	0.7917
		10 %	0.7925	0.7909	0.7824	0.7875	0.7977	0.7979
	Distribución Constante	1 %	0.7656	0.7390	0.7824	0.8455	0.8090	0.8179
		5 %	0.7606	0.8135	0.7768	0.7822	0.7862	0.7713
		10 %	0.7340	0.7714	0.7340	0.7447	0.7341	0.7550
Error por ausencia	N/A	1 %	0.8088	0.7875	0.7556	0.7923	0.7767	0.8033
		5 %	0.8088	0.8084	0.8138	0.8142	0.7872	0.7828
		10 %	0.7603	0.8188	0.8033	0.8028	0.8033	0.7980

Tabla 5.28: Resultados de la degradación de la precisión del clasificador ingenuo de Bayes para el conjunto de datos "Balance Scale". Los errores por valor presentan distintas funciones de probabilidad en la distribución de los mismos. Para errores por ausencia no se considera distribución alguna (RD: Right_Distance, LD: Left_Distance, RW: Right_Weight y LW: Left_Weight; las combinaciones de atributos se denotan mediante "_").

Precisión original 0.6225 Atributos afectados:

			RD	LW	RD_LW	LD_RW	RD_LD_RW	LD_RD_LW
Error por valor	Distribución Uniforme	1 %	0.6752	0.6175	0.5858	0.6593	0.6750	0.5912
		5 %	0.6649	0.5855	0.6435	0.6065	0.6319	0.6596
		10 %	0.6491	0.6013	0.6494	0.6060	0.6013	0.6092
	Distribución Geométrica	1 %	0.6438	0.6703	0.6176	0.7131	0.6871	0.6541
		5 %	0.6757	0.6538	0.6593	0.6593	0/6168	0.6374
		10 %	0.6065	0.6488	0.6330	0.6437	0.6377	0.6808
	Distribución Constante	1 %	0.6597	0.6906	0.6974	0.6661	0.6171	0.6541
		5 %	0.5903	0.6545	0.6277	0.6219	0.6330	0.5797
		10 %	0.6546	0.6437	0.6178	0.6593	0.6276	0.5853
Error por ausencia	N/A	1 %	0.6755	0.6013	0.6967	0.6011	0.6226	0.7127
		5 %	0.7027	0.6377	0.6277	0.6972	0.6972	0.6070
		10 %	0.6283	0.5748	0.6324	0.6539	0.6760	0.6595

Tabla 5.29: Resultados de la degradación de la precisión del clasificador basado en árboles de decisión C4.5 para el conjunto de datos "Balance Scale". Los errores por valor presentan distintas funciones de probabilidad en la distribución de los mismos. Para errores por ausencia no se considera distribución alguna (RD: Right_Distance, LD: Left_Distance, RW: Right_Weight y LW: Left_Weight; las combinaciones de atributos se denotan mediante "_").

Precisión original 0.7606 Atributos afectados:

			RD	LW	RD_LW	LD_RW	RD_LD_RW	LD_RD_LW
Error por valor	Distribución Uniforme	1 %	0.6755	0.6913	0.6969	0.7128	0.6808	0.6546
		5 %	0.7016	0.6967	0.7499	0.7289	0.7595	0.7450
		10 %	0.6437	0.7071	0.7395	0.6909	0.7656	0.7300
	Distribución Geométrica	1 %	0.7714	0.7128	0.7125	0.7767	0.7452	0.7287
		5 %	0.7394	0.7174	0.6858	0.7122	0.7229	0.7552
		10 %	0.7026	0.7437	0.7128	0.7178	0.7603	0.7336
	Distribución Constante	1 %	0.7499	0.7393	0.7660	0.7602	0.7344	0.7819
		5 %	0.7607	0.7711	0.6913	0.6868	0.6910	0.6700
		10 %	0.6916	0.7282	0.6807	0.7552	0.7287	0.7021
Error por ausencia	N/A	1 %	0.7292	0.7558	0.6856	0.7394	0.7504	0.7660
		5 %	0.7772	0.7441	0.7605	0.7182	0.7018	0.7293
		10 %	0.6755	0.7660	0.7340	0.7235	0.7450	0.7293

Tabla 5.30: Resultados de la degradación de la precisión del clasificador basado en k vecinos más próximos para el conjunto de datos "Balance Scale". Los errores por valor presentan distintas funciones de probabilidad en la distribución de los mismos. Para errores por ausencia no se considera distribución alguna (RD: Right_Distance, LD: Left_Distance, RW: Right_Weight y LW: Left_Weight; las combinaciones de atributos se denotan mediante "_").

los atributos que poseen una ganancia de información, la disminución de la precisión es mayor. Las columnas donde aparece el atributo "Ingreso de los padres"(PI) muestran este detalle.

En este conjunto de datos, el modelo clasificador basado en los k vecinos más cercanos mostró un patrón casi idéntico al basado en el ingenuo Bayes, siendo de 2.7 % el mayor detrimento de la precisión.

Por último, en la Tabla 5.31 se presenta una comparativa entre los 3 algoritmos de clasificación. En ella se observa el modelo basado en el árbol de decisión C4.5 posee una tolerancia menor al error que los otros dos métodos.

Algoritmo	Precisión inicial	Precisión mínima	Diferencia máxima
Ingenuo Bayes	0.8287	0.8120	1.6 %
Árbol C4.5	0.8437	0.7211	12.2 %
k - NN	0.8295	0.8022	2.7 %

Tabla 5.31: Comparativa de la precisión presentada por cada algoritmo sobre el conjunto de datos "College Plan".

5.4.5. Resultados sobre el conjunto de datos "Voting Record"

Los resultados obtenidos mediante la utilización de los algoritmos de clasificación ingenuo de Bayes, árbol de decisión C4.5 y k vecinos más próximos para el conjunto de datos "Voting Record", se resumen en las Tablas 5.36, 5.37 y 5.38 respectivamente.

Para el modelo clasificador que utilizó ingenuo Bayes, la precisión mayor que se registró fue de 11 % para con la presencia de errores en los atributos "HomelandSC"(HSC), "PermanentTC"(PTC), "Unemployment"(UNP) y "HomelandSU"(HSU) con errores de rangos inválidos y una distribución constante. De nueva cuenta se aprecia que la distribución de los valores no presenta un factor determinante para la degradación de la precisión, tanto para conjuntos de datos en los cuales solamente se afectó un atributo (siendo la diferencia de un 3 %), como para varios atributos (5 % la diferencia mayor para 3 atributos). En cuanto a los errores por ausencia, la reducción más importante fue de 5 %, mientras que para errores por valores fuera de rango es de 7 %, lo cual tampoco representa un facto decisivo en la reducción de la precisión.

En el caso del modelo que utilizó árbol de clasificación C4.5, la reducción más significativa, para cualquier combinación de tipo de error, porcentaje y distribución de valores (cuando aplica), únicamente fue de 2.2 % (para con la presencia de errores en los atributos "HomelandSC"(HSC), "PermanentTC"(PTC), "Unemployment"(UNP) y "HomelandSU"(HSU) con errores de rangos inválidos y una distribución uniforme). Por su parte, con el modelo de k vecinos más próximos, el patrón se conserva, es decir aún cuando la degradación existe, esta no resulta ser mayor al 3 % (evento que se

Precisión original 0.8287 Atributos afectados:

			PE	PI	PE_PI	IQ_GE	PE_PI_GE	PE_PI_IQ_GE
Error por valor	Distribución Uniforme	1 %	0.8280	0.8315	0.8233	0.8315	0.8337	0.8269
		5 %	0.8257	0.8257	0.8276	0.8248	0.8241	0.8224
		10 %	0.8120	0.8263	0.8285	0.8281	0.8241	0.8181
	Distribución Geométrica	1 %	0.8300	0.9172	0.9117	0.8312	0.9290	0.9231
		5 %	0.8231	0.9094	0.9137	0.8248	0.9096	0.9139
		10 %	0.8231	0.9028	0.9057	0.8261	0.9126	0.9022
	Distribución Constante	1 %	0.8263	0.8228	0.8281	0.8300	0.8272	0.8265
		5 %	0.8257	0.8248	0.8246	0.8272	0.8200	0.8289
		10 %	0.8267	0.8231	0.8250	0.8333	0.8250	0.8339
Error por ausencia	N/A	1 %	0.8333	0.8333	0.9254	0.8270	0.9178	0.9233
		5 %	0.8274	0.9091	0.9159	0.8302	0.9131	0.9119
		10 %	0.8196	0.9022	0.9039	0.8326	0.9122	0.9128

Tabla 5.32: Resultados de la degradación de la precisión del clasificador ingenuo de Bayes para el conjunto de datos "College Plan". Los errores por valor presentan distintas funciones de probabilidad en la distribución de los mismos. Para errores por ausencia no se considera distribución alguna (PE: Animo de los padres, PI: Ingreso de los padres, IQ: IQ y GE: Género; las combinaciones de atributos se denotan mediante "_").

Precisión original	0.8437		Atributos afectados:					
			PE	PI	PE_PI	IQ_GE	PE_PI_GE	PE_PI_IQ_GE
Error por valor	Distribución Uniforme	1 %	0.8354	0.8441	0.8317	0.8372	0.8372	0.8415
		5 %	0.8326	0.8372	0.8387	0.8367	0.8387	0.8378
		10 %	0.7989	0.8311	0.8302	0.8381	0.8341	0.8193
	Distribución Geométrica	1 %	0.8328	0.7746	0.7402	0.8344	0.9197	0.6733
		5 %	0.8337	0.7409	0.7704	0.8322	0.7465	0.6733
		10 %	0.8300	0.7819	0.7824	0.8367	0.7774	0.7313
	Distribución Constante	1 %	0.8391	0.8352	0.8370	0.8472	0.8433	0.8328
		5 %	0.8367	0.8337	0.8400	0.8348	0.8237	0.8391
		10 %	0.8280	0.8296	0.8309	0.8465	0.8306	0.8400
Error por ausencia	N/A	1 %	0.8404	0.7694	0.7537	0.8337	0.7076	0.7211
		5 %	0.8313	0.7763	0.7763	0.8370	0.7337	0.7180
		10 %	0.8296	0.7746	0.7337	0.8296	0.7211	0.7211

Tabla 5.33: Resultados de la degradación de la precisión del clasificador basado en árboles de decisión C4.5 para el conjunto de datos "College Plan". Los errores por valor presentan distintas funciones de probabilidad en la distribución de los mismos. Para errores por ausencia no se considera distribución alguna (PE: Animo de los padres, PI: Ingreso de los padres, IQ: IQ y GE: Género; las combinaciones de atributos se denotan mediante "_").

Precisión original 0.8295 Atributos afectados:

			PE	PI	PE_PI	IQ_GE	PE_PI_GE	PE_PI_IQ_GE
Error por valor	Distribución Uniforme	1 %	0.8263	0.8341	0.8174	0.8289	0.8328	0.8330
		5 %	0.8237	0.8283	0.8328	0.8194	0.8241	0.8252
		10 %	0.8100	0.8209	0.8191	0.8213	0.8241	0.8043
	Distribución Geométrica	1 %	0.8281	0.8985	0.8800	0.8312	0.8733	0.8344
		5 %	0.8215	0.8822	0.8898	0.8193	0.8752	0.8226
		10 %	0.8178	0.8733	0.8531	0.8050	0.8761	0.8193
	Distribución Constante	1 %	0.8244	0.8241	0.8311	0.8302	0.8263	0.8233
		5 %	0.8228	0.8244	0.8280	0.8209	0.8183	0.8263
		10 %	0.8200	0.8187	0.8187	0.8187	0.8220	0.8213
Error por ausencia	N/A	1 %	0.8306	0.8939	0.8961	0.8185	0.8835	0.8346
		5 %	0.8241	0.8733	0.8800	0.8265	0.8874	0.8226
		10 %	0.8022	0.8683	0.8622	0.8063	0.8693	0.8244

Tabla 5.34: Resultados de la degradación de la precisión del clasificador basado en k vecinos más próximos para el conjunto de datos "College Plan". Los errores por valor presentan distintas funciones de probabilidad en la distribución de los mismos. Para errores por ausencia no se considera distribución alguna (PE: Animo de los padres, PI: Ingreso de los padres, IQ: IQ y GE: Género; las combinaciones de atributos se denotan mediante "_").

presentó con la presencia de errores por ausencia de valor en 10 % sobre los atributos "Fiscal03BR"(FBR), "HomelandSC"(HSC) y "PermanentTC"(PTC)).

Por último, en la Tabla 5.35 se presenta una comparativa entre los 3 algoritmos de clasificación. Para este conjunto de datos, el modelo que presenta una precisión menor resulta ser el que utiliza ingenuo Bayes, mientras que aquellos con árbol C4.5 y k vecinos más próximos poseen una eficiencia muy similar, además de ser más tolerantes a los errores. Un factor a considerar sobre esta diferencia representativa, es la cantidad de atributos que tiene el conjunto de datos (20).

Algoritmo	Precisión inicial	Precisión mínima	Diferencia máxima
Ingenuo Bayes	0.9339	0.8271	11 %
Árbol C4.5	0.9846	0.9692	2.2 %
k - NN	0.9855	0.9538	3.3 %

Tabla 5.35: Comparativa de la precisión presentada por cada algoritmo sobre el conjunto de datos "Voting Record".

5.4.6. Resultados sobre el conjunto de datos "Credit"

Los resultados obtenidos mediante la utilización de los algoritmos de clasificación ingenuo de Bayes, árbol de decisión C4.5 y k vecinos más próximos para el conjunto de datos "Credit", se resumen en las Tablas 5.41, 5.37 y 5.38 respectivamente.

Para este conjunto de datos, el modelo clasificador basado en ingenuo Bayes presenta el comportamiento similar a los anteriores conjuntos de datos. La diferencia más amplia que se presentó fue de 11 % para el error fuera de rango de 10 % sobre la combinación de los atributos "A9", "A10", "A5" y "A2" con una distribución constante. Mientras que la diferencia promedio en los restantes conjuntos de datos no superó el 7 %.

Por su parte, los resultados el modelo clasificador basado en el árbol de decisión C4.5 para este conjunto de datos muestran dos aspectos interesantes, el primero consiste en que su precisión ha sido la más baja entre todos los modelos y el segundo radica en que aún su baja precisión el modelo presenta la mejor tolerancia los errores, ubicándose la mayor reducción en un 4.4 % (Tabla 5.39), dicha reducción se presentó para el error fuera de rango al 10 % sobre la combinación de los atributos "A9" y "A10" con una distribución constante.

El último modelo de clasificación, basado en k vecinos más próximos, no muestra una influencia determinante a causa del tipo de error (la mayor reducción para ausencia de valor es de 3.1 % y para error por valor fuera de rango es de 8.6, aunque la precisión sobre este conjunto de datos resultó una excepción ya que en promedio la diferencia solamente fue de 3.9 %), porcentaje o distribución del valor de los errores. Por último, en la Tabla 5.39 se presenta una comparativa entre los 3 algoritmos de clasificación, para este conjunto de datos.

Precisión original	0.9339		Atributos afectados:					
			FBR	HSC	FBR_HSC	UNP_HSU	FBR_HSC_PTC	HSC_PTC_UNP_HSU
Error por valor	Distribución Uniforme	1 %	0.8416	0.9103	0.9265	0.8920	0.8826	0.8776
		5 %	0.8652	0.8350	0.8725	0.8573	0.8652	0.8348
		10 %	0.8436	0.8416	0.8493	0.8342	0.8345	0.8342
	Distribución Geométrica	1 %	0.8548	0.8875	0.9179	0.8262	0.8256	0.8641
		5 %	0.8503	0.8573	0.8866	0.9179	0.8652	0.8650
		10 %	0.8582	0.8795	0.8345	0.9036	0.8650	0.8342
	Distribución Constante	1 %	0.8963	0.8652	0.8715	0.8869	0.8573	0.8499
		5 %	0.8581	0.8729	0.8658	0.8806	0.8103	0.8271
		10 %	0.8726	0.8729	0.8658	0.8806	0.8271	0.8271
Error por ausencia	N/A	1 %	0.9026	0.9111	0.9182	0.8875	0.9342	0.9040
		5 %	0.9575	0.8647	0.9026	0.8655	0.9182	0.8581
		10 %	0.8581	0.8960	0.9023	0.8667	0.9182	0.8575

Tabla 5.36: Resultados de la degradación de la precisión del clasificador ingenuo de Bayes para el conjunto de datos "Voting Record". Los errores por valor presentan distintas funciones de probabilidad en la distribución de los mismos. Para errores por ausencia no se considera distribución alguna (FBR: Fiscal03BR, HSC: HomelandSC, PTC: PermanentTC, UNP: Unemployment, HSU HomelandSU; las combinaciones de atributos se denotan mediante "_").

Precisión original	0.9849		Atributos afectados:					
			FBR	HSC	FBR_HSC	UNP_HSU	FBR_HSC_PTC	HSC_PTC_UNP_HSU
Error por valor	Distribución Uniforme	1 %	0.9923	0.9846	0.9846	0.9823	0.9826	0.9923
		5 %	0.9923	0.9923	0.9826	0.9849	0.9849	0.9849
		10 %	0.9923	0.9923	0.9769	0.9846	0.9846	0.9692
	Distribución Geométrica	1 %	0.9849	0.9846	0.9846	0.9849	0.9846	0.9769
		5 %	0.9823	0.9775	0.9849	0.9849	0.9823	0.9846
		10 %	0.9823	0.9826	0.9923	0.9772	0.9769	0.9846
	Distribución Constante	1 %	0.9772	0.9852	0.9923	0.9846	0.9621	0.9846
		5 %	0.9846	0.9849	0.9849	0.9852	0.9849	0.9846
		10 %	0.9846	0.9772	0.9846	0.9772	0.9772	0.9769
Error por ausencia	N/A	1 %	0.9849	0.9849	0.9849	0.9769	0.9769	0.9695
		5 %	0.9769	0.9849	0.9846	0.9849	0.9772	0.9769
		10 %	0.9695	0.9695	0.9772	0.9772	0.9775	0.9775

Tabla 5.37: Resultados de la degradación de la precisión del clasificador basado en árboles de decisión C4.5 para el conjunto de datos "Voting Record". Los errores por valor presentan distintas funciones de probabilidad en la distribución de los mismos. Para errores por ausencia no se considera distribución alguna. (FBR: Fiscal03BR, HSC: HomelandSC, PTC: PermanentTC, UNP: Unemployment, HSU HomelandSU; las combinaciones de atributos se denotan mediante "_").

Precisión original	0.9855		Atributos afectados:					
			FBR	HSC	FBR_HSC	UNP_HSU	FBR_HSC_PTC	HSC_PTC_UNP_HSU
Error por valor	Distribución Uniforme	1 %	0.9923	0.9792	0.9846	0.9692	0.9772	0.9692
		5 %	0.9923	0.9772	0.9775	0.9698	0.9772	0.9775
		10 %	0.9621	0.9769	0.9538	0.9695	0.9772	0.9775
	Distribución Geométrica	1 %	0.9769	0.9846	0.9849	0.9849	0.9695	0.9618
		5 %	0.9695	0.9769	0.9849	0.9823	0.9641	0.9692
		10 %	0.9646	0.9519	0.9695	0.9638	0.9544	0.9618
	Distribución Constante	1 %	0.9846	0.9923	0.9769	0.9695	0.9621	0.9769
		5 %	0.9618	0.9695	0.9772	0.9695	0.9621	0.9849
		10 %	0.9462	0.9618	0.9692	0.9772	0.9772	0.9772
Error por ausencia	N/A	1 %	0.9467	0.9695	0.9849	0.9695	0.9769	0.9621
		5 %	0.9538	0.9769	0.9846	0.9621	0.9541	0.9618
		10 %	0.9618	0.9621	0.9769	0.9541	0.9538	0.9695

Tabla 5.38: Resultados de la degradación de la precisión del clasificador basado en k vecinos más próximos para el conjunto de datos "Voting Record". Los errores por valor presentan distintas funciones de probabilidad en la distribución de los mismos. Para errores por ausencia no se considera distribución alguna. (FBR: Fiscal03BR, HSC: HomelandSC, PTC: PermanentTC, UNP: Unemployment, HSU HomelandSU; las combinaciones de atributos se denotan mediante "_").

Algoritmo	Precisión inicial	Precisión mínima	Diferencia máxima
Ingenuo Bayes	0.7985	0.6812	11 %
Árbol C4.5	0.5530	0.5197	4.4 %
k - NN	0.8511	0.7646	8.6 % (3.9 % prom)

Tabla 5.39: Comparativa de la precisión presentada por cada algoritmo sobre el conjunto de datos "Credit".

5.4.7. Resultados sobre el conjunto de datos "Diabetes"

Los resultados obtenidos mediante la utilización de los algoritmos de clasificación ingenuo de Bayes, árbol de decisión C4.5 y k vecinos más próximos para el conjunto de datos "Diabetes",

El análisis sobre el último conjunto de datos, "Diabetes", presenta un patrón similar a los conjuntos de datos anteriores. Básicamente existe una reducción sobre la precisión de los modelos de clasificación (Tabla 5.40) sin embargo dicha reducción no es mayor que 2.2 % para el modelo con ingenuo Bayes (con error por valores de fuera de rango en un 10 % afectando los atributos "Glucose"(GL) y "Age"(AG)), 7.7 % para árbol de decisión C4.5 (con error por valores de fuera de rango en un 10 % afectando los atributos "Glucose"(GL), "BodyMass"(BM) y "Pregnant"(PG)) y 10 % para k vecinos más próximos (con error por valores de fuera de rango en un 10 % afectando los atributos "Glucose"(GL), "BodyMass"(BM), "Age"(AG) e "Insulin"(IN)).

Del mismo modo, la diferencia entre errores por ausencia y errores por valores fuera de rango, impactan de manera similar (la mayor variante entre ellos es de un deterioro del 1.5 %, 2.5 % y 2.4 % para los modelos basados en ingenuo Bayes, C4.5 y k-NN respectivamente) a la precisión general del modelo.

Una comparativa de la precisión de estos modelos sobre el conjunto de datos "Diabetes" se resume en la Tabla 5.40, donde se aprecia de modo más evidente que el clasificador ingenuo de Bayes presenta la mejor tolerancia y precisión general.

Algoritmo	Precisión inicial	Precisión mínima	Diferencia máxima
Ingenuo Bayes	0.7565	0.7286	2.2 %
Árbol C4.5	0.7454	0.6729	7.7 %
k - NN	0.7100	0.6152	10 %

Tabla 5.40: Comparativa de la precisión presentada por cada algoritmo sobre el conjunto de datos "Diabetes".

Precisión original	0.7985		Atributos afectados:					
			A9	A2	A9_A10	A9_A10_A2	A10_A5_A2	A9_A10_A5_A2
Error por valor	Distribución Uniforme	1 %	0.7738	0.7592	0.7453	0.7308	0.7329	0.7433
		5 %	0.7597	0.7697	0.7371	0.7060	0.7205	0.7246
		10 %	0.7405	0.7267	0.7308	0.7308	0.7060	0.7101
	Distribución Geométrica	1 %	0.7101	0.7592	0.7081	0.7536	0.7308	0.7308
		5 %	0.7453	0.7371	0.7164	0.7205	0.7288	0.7101
		10 %	0.7329	0.7619	0.7329	0.7288	0.7619	0.7143
	Distribución Constante	1 %	0.7205	0.7474	0.7536	0.7329	0.7557	0.7433
		5 %	0.7350	0.7329	0.7536	0.7453	0.7267	0.7433
		10 %	0.7184	0.7205	0.7122	0.7205	0.7122	0.7205
Error por ausencia	N/A	1 %	0.7371	0.7598	0.7619	0.7288	0.7288	0.7391
		5 %	0.7350	0.7495	0.7350	0.7329	0.7391	0.7412
		10 %	0.7329	0.7350	0.7640	0.7267	0.7205	0.7226

Tabla 5.41: Resultados de la degradación de la precisión del clasificador ingenuo de Bayes para el conjunto de datos "Credit". Los errores por valor presentan distintas funciones de probabilidad en la distribución de los mismos. Para errores por ausencia no se considera distribución alguna (las combinaciones de atributos se denotan mediante "_", para este conjunto de datos no hay mas descripción de los atributos).

Precisión original	0.8612		Atributos afectados:					
			A9	A2	A9_A10	A9_A10_A2	A10_A5_A2	A9_A10_A5_A2
Error por valor	Distribución Uniforme	1 %	0.8433	0.8482	0.8342	0.8559	0.8611	0.8217
		5 %	0.8384	0.8481	0.8342	0.8500	0.8424	0.8300
		10 %	0.8244	0.8193	0.8238	0.8135	0.8383	0.8217
	Distribución Geométrica	1 %	0.8549	0.8611	0.8611	0.8652	0.8445	0.8652
		5 %	0.8259	0.8528	0.8549	0.8404	0.8673	0.8321
		10 %	0.8362	0.8528	0.8549	0.8528	0.8487	0.8445
	Distribución Constante	1 %	0.8549	0.8569	0.8457	0.8549	0.8611	.08549
		5 %	0.8549	0.8549	0.8404	0.8528	0.8549	0.8445
		10 %	0.8238	0.8487	0.8197	0.8259	0.8445	0.8321
Error por ausencia	N/A	1 %	0.8300	0.8362	0.8487	0.8487	0.5466	0.8559
		5 %	0.8342	0.8507	0.8404	0.8528	0.8466	0.8549
		10 %	0.8383	0.8383	0.8362	0.8466	0.8542	0.8549

Tabla 5.42: Resultados de la degradación de la precisión del clasificador basado en árboles de decisión C4.5 para el conjunto de datos "Credit". Los errores por valor presentan distintas funciones de probabilidad en la distribución de los mismos. Para errores por ausencia no se considera distribución alguna (las combinaciones de atributos se denotan mediante "_", para este conjunto de datos no hay mas descripción de los atributos).

Precisión original	0.8511		Atributos	afectados:				
			A9	A2	A9_A10	A9_A10_A2	A10_A5_A2	A9_A10_A5_A2
Error por valor	Distribución Uniforme	1 %	0.8749	0.8702	0.8489	0.8571	0.8406	0.8385
		5 %	0.8177	0.8173	0.8178	0.8509	0.8344	0.8282
		10 %	0.7646	0.8130	0.7867	0.8344	0.8282	0.8323
	Distribución Geométrica	1 %	0.8364	0.8199	0.8261	0.8385	0.8178	0.8364
		5 %	0.8406	0.8178	0.8199	0.8344	0.8364	0.8406
		10 %	0.8344	0.8095	0.8261	0.8219	0.8240	0.8199
	Distribución Constante	1 %	0.8344	0.8489	0.8385	0.8364	0.8530	0.8364
		5 %	0.8282	0.8261	0.8427	0.8489	0.8282	0.8157
		10 %	0.8137	0.8178	0.8364	0.8302	0.8240	0.8137
Error por ausencia	N/A	1 %	0.8323	0.8406	0.8240	0.8509	0.8551	0.8509
		5 %	0.8385	0.8219	0.8323	0.8509	0.8323	0.8364
		10 %	0.8178	0.8406	0.8199	0.8219	0.8530	0.8282

Tabla 5.43: Resultados de la degradación de la precisión del clasificador basado en k vecinos más próximos para el conjunto de datos "Credit". Los errores por valor presentan distintas funciones de probabilidad en la distribución de los mismos. Para errores por ausencia no se considera distribución alguna (las combinaciones de atributos se denotan mediante "_", para este conjunto de datos no hay mas descripción de los atributos).

5.5. Recapitulación

La experimentación resulta ser un elemento importante en el desarrollo de nuevos conocimientos debido a que permite sustentar de forma práctica aquellos elementos teóricos que se presumen verdaderos. Dentro de la minería de datos, no se cuenta con una base de datos general de experimentación que permita el análisis y ponderación de la eficiencia y eficacia para los distintos algoritmos que se emplean en las tareas de clasificación.

En el capítulo se presentó el resultado de llevar a cabo experimentos sobre distintos conjuntos de datos y con ello determinar la tolerancia a los distintos errores.

Al finalizar, se ha observa que las ideas iniciales sobre el distinto grado de tolerancia para los algoritmos utilizados dentro de los modelos de clasificación es acertada, siendo en algunos casos independiente de las características de los datos y solamente teniendo importancia el porcentaje total de errores que se presentan.

CAPÍTULO 6

Conclusiones

Conclusiones

La principal aportación del presente trabajo es la evaluación y comparación experimental de los modelos de clasificación basados en ingenuo Bayes, árbol de clasificación C4.5 y k vecinos más próximos[1] en cuanto a su precisión y tolerancia a los distintos tipos de errores identificados durante la investigación.

En general, los tres modelos presentan una tolerancia aceptable considerando el grado de error presente en todos los conjuntos de datos analizados. Sin embargo, como resultado directo del análisis de los resultados de la experimentación se han podido identificar los siguientes elementos que resultan ser de importancia para que las personas encargadas del área de BI tomen en cuenta.

- La importancia y sensibilidad de los datos e información que se estudian, deben ser analizadas previamente y en caso de ser factible asociar una matriz de costos para mejorar la precisión del modelo (Sección 1.7).

- El número de atributos afectados resultó ser únicamente determinante para los modelos de clasificación que utilizan los k vecinos más próximos, debido a que mientras se cuente con un mayor número de atributos y se distribuya el error en ellos, la similitud entre los elementos se mantiene mediante aquellos atributos que no son afectados. Por lo tanto, poseer un conjunto de datos con pocos atributos y que presenten errores en ellos, tiene un impacto en la precisión del modelo, como puede verse en las Tablas 5.35 y 5.39 de las páginas 112 y 116 respectivamente.

[1] Con $k = 5$ debido al parámetro por omisión se manejó.

- La distribución de los errores dentro de múltiples atributos para un conjunto de datos no es significativa para los algoritmos utilizados en los modelos de clasificación con los cuales se experimentaron a excepción del modelo donde se utilizó el árbol de decisión C.4.5. Ya que uno de los criterios para la introducción de errores fue precisamente la afectación sobre atributos que poseen una Ganancia de Información mayor, y por ende la elección de estos atributos como atributos de división se modificó. Sin embargo, dependiendo de las características del conjunto de datos, es posible que se presente el caso de que el error distribuido sobre varios atributos y en una proporción de 10 % tenga un menor impacto que un error que afecta un único atributo en un 5 %.

- Si los errores se encuentran aleatoriamente dentro del conjunto de datos y son errores por valores fuera de rango, la distribución de los valores de estos errores no son un factor determinante en la reducción de la precisión del modelo.

- La reducción de la precisión presente en el modelo clasificador es independiente del tipo de error que se encuentre en el conjunto de datos, ya sea por valores fuera del rango o bien con elementos ausentes.

Por lo anterior, se puede enunciar para el conjunto de datos analizados el *criterio de supresión de los procesos de limpieza de datos* de la siguiente forma:

Criterio de supresión de los procesos de limpieza de datos

Para un conjunto de datos que sea analizado por alguno de los clasificadores estudiados en el presente trabajo, en caso de no contar con alguna otra métrica para los clasificadores[2] , se puede afirmar que:

- Para un modelo de clasificación que utilice el árbol de decisión C4.5, si el error que se elige como el atributo de división por el algoritmo es menor que la proporción de error en los demás atributos entonces el conjunto de datos puede prescindir de los procesos de limpieza y transformación de datos, siempre y cuando sea aceptable una disminución de la precisión del modelo[3] . La razón de esto radica en que al acumular errores sobre los atributos que son elegidos como atributos de división, progresivamente la ganancia de información que poseen va disminuyendo al grado de que posteriormente el atributo en cuestión deja de ser elegido y otro atributo pasa a ser el nuevo atributo de división durante la generación del árbol. Como se ve en las Secciones 5.4.2 y 5.4.4 páginas 96 y 104.

- Para un modelo de clasificación que utilice los k vecinos más próximos, si el número de atributos por analizar es relativamente grande y los errores afectan

[2] Como puede ser la utilización de matrices de costos.

[3] En el caso particular de los conjuntos analizados la degradación de la precisión fue de menos de 10 %.

una proporción mínima de los atributos entonces el conjunto de datos puede prescindir de los procesos de limpieza y transformación de datos, siempre y cuando sea aceptable una disminución la precisión del modelo[4] . En este caso, la razón recae en que al acumular errores sobre un solo atributo, la similitud que guardan los elementos se conserva debido a los restantes atributos. Lo anterior se ve en las Secciones 5.4.6 y 5.4.5 páginas 112 y 108.

- Para un modelo de clasificación que utilice ingenuo Bayes, si el número de atributos es pequeño y la proporción de errores es menor que el número de elementos de la clase menos numerosa entonces el conjunto de datos puede prescindir de los procesos de limpieza y transformación de datos, siempre y cuando sea aceptable una disminución la precisión del modelo[5] . Para este modelo clasificador, el fundamento de lo anterior se basa en el hecho de que al modificar los datos con errores introducidos aleatoriamente éstos se ven afectados en proporciones similares para los elementos de las distintas clases, así los valores de las probabilidades o "*scores*" que se calculan para la asignación de las etiquetas o clases, se ve afectada y por ende, un mismo elemento será asignado a una clase distinta.

Sin embargo, el criterio no debe ser generalizado a discreción, debido a la sensibilidad de la información y el costo que puedan tener los distintos datos.

Por último, y como criterios adicionales que se encontraron durante la presente investigación, es importante considerar lo siguiente:

- Los modelos clasificadores: árboles de decisión C4.5, clasificador ingenuo de Bayes y los k vecinos más próximos son tolerantes al ruido o presencia de errores en los datos de entrenamiento. Sin embargo, la eficiencia entre estos modelos y por consiguiente la elección para su aplicación dentro de algún proceso de extracción de conocimiento tendrá que tomar en cuenta el tipo y las características de los datos por analizar. Así, por ejemplo, si se tratase únicamente de datos númericos y se dispone de una gran capacidad de procesamiento, el modelo ideal por utilizar es el k vecinos más próximos.

- El trabajo computacional debe ser un factor decisivo en la elección del algoritmo a utilizar dentro del modelo de clasificación, ya que el algoritmo de k vecinos más próximos (y en general los algoritmos retardados para clasificación, dentro del área de minería de datos) resultan ser más lentos y la velocidad de procesamiento se reduce significativamente a medida que aumenta el tamaño del conjunto de datos por analizar.

[4] En el caso particular de los conjuntos analizados la degradación de la precisión fue de menos de 5 %.

[5] En el caso particular de los conjuntos analizados la degradación de la precisión fue de menos de 7 %.

Trabajo futuro

Los distintos algoritmos de clasificación que se han analizado en el presente trabajo cuentan con múltiples extensiones y características que pueden expandir el trabajo, además otros conceptos que se encuentran de trasfondo como lo son por ejemplo almacenes de datos o bases de datos distribuidas, permiten extender el análisis. Dentro de las principales líneas de investigación se vislumbran las siguientes:

1. Analizar y cuantificar la tolerancia al error de otros algoritmos de clasificación, por ejemplo redes neuronales o maquinas de soporte vectorial.

2. En el caso de la utilización de algoritmo de árboles de inducción, considerar dentro de un ambiente distribuido la conjunción de distintas fuentes de datos e identificar posibles conjuntos de datos que sean factibles de recibir un pre-procesado debido a que serán atributos no tomados en cuenta como atributos de división.

3. Analizar el grado de tolerancia utilizando otras estructuras auxiliares como la matriz de costos o el área ROC.

APÉNDICE A

Funciones de distribución de probabilidad

La probabilidad es un mecanismo por medio del cual pueden estudiarse sucesos aleatorios cuando éstos se comparan con los fenómenos determinísticos. La probabilidad tiene un papel determinante en la aplicación de la inferencia estadística porque una decisión cuyo fundamento se encuentra en la información contenida en una muestra aleatoria, puede estar equivocada. Sin una adecuada comprensión de las leyes básicas de la probabilidad, es difícil utilizar la metodología estadística de manera efectiva.

A.1. Desarrollo axiomático de la probabilidad

Para formalizar la definición de probabilidad, a través de un conjunto de axiomas se presentan brevemente los conceptos básicos de la teoría de conjuntos sobre los cuales se fundamente la definición formal de probabilidad. Esta definición es general y permite incorporar las distintas interpretaciones de las probabilidad. La colección de todos los posibles resultados de un experimento aleatorio es importante en la definición de la probabilidad.

Definición A.1.1. El conjunto de todos los posibles resultados de un experimento aleatorio recibe el nombre de *espacio muestral*.

El conjunto de todos los posibles resultados puede ser finito, infinito numerable o infinito no numerable. Por ello, se añaden las siguientes definiciones.

Definición A.1.2. Se dice que un espacio muestral es *discreto* si su resultado puede ponerse en una correspondencia uno a uno con el conjunto de los números enteros positivos.

Definición A.1.3. Se dice que un espacio muestral es *continuo* si sus resultados consisten de un intervalo de números reales.

Con respecto a los resultados de un espacio muestral, se puede estar particularmente interesado en un subconjunto de éstos. De esta manera, se tiene la siguiente definición:

Definición A.1.4. Un *evento* del espacio muestral es un grupo de resultados contenidos en éste, cuyos miembros tienen una característica común.

Por característica común se debe entender que únicamente un grupo de resultados en particular satisface la característica y que los restantes, contenidos en el espacio muestral, no. Se dice que ha ocurrido un evento si los resultados del experimento aleatorio incluyen a algunos de los que definen al evento. En este contexto, el espacio muestral, evento en sí mismo, puede entenderse como un *evento seguro*, puesto que se tiene un 100 % de certidumbre de que ocurrirá un resultado del espacio muestral cuando el experimento se lleve a cabo. Para completar el estudio, se tienen las siguientes definiciones:

Definición A.1.5. El evento que contiene a ningún resultado del espacio muestral recibe el nombre de *evento nulo* o *vacío*.

Retomando algunas definiciones de la teoría de eventos. Sean E_1 y E_2 cualesquiera dos eventos que se encuentren en un espacio muestral dado, denotado por S, se tienen las siguientes definiciones:

Definición A.1.6. El evento formado por todos los posibles resultados en E_1 o E_2 o en ambos, recibe el nombre de la *unión* de E_1 y E_2 y se denota por $E_1 \cup E_2$.

Definición A.1.7. El evento formado por todos los resultados comunes tanto a E_1 como a E_2 recibe el nombre de *intersección* de E_1 y E_2 y se denota por $E_1 \cap E_2$.

Definición A.1.8. Se dice que los eventos E_1 y E_2 son *mutuamente excluyentes* o *disjuntos* si no tienen resultados en común; es decir que $E_1 \cap E_2 = \emptyset \equiv$ *evento vacío*.

Definición A.1.9. Si cualquier resultado de E_2 también es un resultado de E_1, se dice que el evento E_2 está *contenido* en E_1, y se denota por $E_2 \subset E_1$.

Definición A.1.10. El *complemento* de un evento E con respecto al espacio muestral S, es aquel que contiene a todos los resultados de S que no se encuentran en E, y se denota por $\overline{E}$.

La probabilidad es un número real que mide la posibilidad de que ocurra un resultado del espacio muestral, cuando el experimento se lleve a cabo. Por lo tanto, la probabilidad de un evento también es un número real que mide la posibilidad colectiva, de ocurrencia, de los resultados del evento cuando se lleve a efecto el experimento. A continuación se da la definición axiomática de la probabilidad.

Definición A.1.11. Sean S cualquier espacio muestral y E cualquier evento de éste. Se llamará *función de probabilidad* sobre el espacio muestral S a $P(E)$ si satisface los siguientes axiomas:

1. $P(E) \geqslant 0$
2. $P(S) = 1$

3. Si, para los eventos $E_1, E_2, E_3, \ldots,$

$$E_i \cap E_j = \emptyset \text{ para toda } i \neq j, \text{ entonces}$$
$$P(E_1 \cup E_2 \cup \cdots) = P(E_1) + P(E_2) + \cdots$$

La razón de estos tres axiomas se convierte en aparente cuando se recuerda la interpretación de la probabilidad como una frecuencia relativa. Es decir, la probabilidad de un evento refleja la proporción de veces en que ocurrirá cuando el experimento se repita. A continuación se presentan algunas consecuencias de estos tres axiomas.

Teorema 2. La probabilidad de un evento vacío es nula. O bien, $P(\emptyset) = 0$.

Demostración. Se tiene que

$$S \cup \emptyset = S \text{ y } S \cap \emptyset = \emptyset.$$

Por el axioma 3,

$$P(S \cup \emptyset) = P(S) + P(\emptyset).$$

Pero por el axioma 2, $P(S) = 1$ y de esta forma $P(\emptyset) = 0$ □

Teorema 3. Para cualquier evento $E \subset S, 0 \leqslant P(E) \leqslant 1$

Demostración. Por el axioma 1, $P(E) \geqslant 0$; de aquí que sólo es necesario probar que $P(E) \leqslant 1$.

$$E \cup \overline{E} = S \text{ y } E \cap \overline{E} = \emptyset.$$

Por los axiomas 2 y 3,

$$P(E \cup \overline{E}) = P(E) + P(\overline{E}) = P(S) = 1.$$

Dado que $P(E) \geqslant 0$, se tiene que $P(E) \leqslant 1$ □

El axioma 3 da la probabilidad de la unión de dos eventos disjuntos. El siguiente resultado general se conoce como "Regla de adición de probabilidades":

Teorema 4. Sea S un espacio muestral que contiene a cualesquiera dos eventos A y B, entonces se tiene:

$$P(A \cup B) = P(A) + P(B) - P(A \cap B).$$

Generalmente hablando, la probabilidad condicional de un evento A dado otro evento B, denotada $P(A|B)$ es la probabilidad de que el evento A ocurra cuando sabemos que el evento B ocurrió. Esta es la razón por la cual se llama condicional a esta probabilidad. La probabilidad de que el evento A ocurra está condicionada por la ocurrencia de B. Este información adicional sobre A se incluye en el cómputo de su probabilidad condicional cuando analizamos los resultados posibles que se pueden observar cuando sabemos que B ha ocurrido. Con base en lo anterior, se define la probabilidad condicional de la siguiente manera:

Definición A.1.12. Sean A y B cualesquiera dos eventos que se encuentran en un espacio muestral S de manera tal que $P(B) > 0$. La probabilidad condicional de A al ocurrir el evento B, es el cociente de la probabilidad conjunta de A y B con respecto a la probabilidad marginal de B; de esta manera se tiene

$$P(A|B) = \frac{P(A \cap B)}{P(B)}, P(B) > 0 \tag{A.1}$$

A.1.1. Eventos estadísticamente independientes

Al considerar la probabilidad condicional de algún evento A, dada la ocurrencia de otro evento B, siempre implica que las probabilidades de A y B son de alguna manera dependientes entre ellas. Es decir, la información con respecto a la ocurrencia de B afectará la probabilidad de A. Supóngase que la ocurrencia de B no tiene algún efecto sobre la probabilidad de A, en el sentido de que la probabilidad condicional $P(A|B)$ es igual a la probabilidad marginal $P(A)$, aún a pesar de que haya ocurrido el evento B. Esta situación origina un concepto muy importante que se conoce como independencia estadística.

Definición A.1.13. Sean A y B dos eventos cualesquiera de un espacio muestral S. Se dice que el evento A es *estadísticamente independiente* del evento B si se cumple que $P(A|B) = P(A)$.

Recordando la definición de probabilidad condicional (Ecuación A.1), algunas consecuencias de la anterior definición son las siguientes. Dado que

$$P(A|B) = \frac{P(A \cap B)}{P(B)}.$$

Si A es independiente de B, entonces

$$P(A|B) = P(A) = \frac{P(A \cap B)}{P(B)}.$$

o

$$\frac{P(A \cap B)}{P(B)} = P(A)P(B).$$

Además, puesto que $P(A \cap B) = P(A)P(B|A)$, se tiene que:

$$P(A)P(B) = P(A)P(B|A).$$

o equivalentemente

$$P(B) = P(B|A).$$

Por lo tanto, puede concluirse que si un evento A es estadísticamente independiente de B, entonces el evento B es independiente de A y se verifican las tres relaciones siguientes:

1. $P(A|B) = P(A)$,
2. $P(B|A) = P(B)$ y
3. $P(A \cap B) = P(A)P(B)$.

A.1.2. El teorema de Bayes

Teorema 5 (Bayes). Sean A y B dos sucesos aleatorios cuyas probabilidades se denotan por $P(A)$ y $P(B)$ respectivamente, y además se cumple que $P(B) > 0$. Supongamos conocidas las probabilidades *a priori* de los sucesos A y B, es decir, $P(A)$ y $P(B)$, así como la probabilidad condicionada del suceso B dado el suceso A, es decir $P(B|A)$. La probabilidad *a posteriori* del suceso A conocido dado que verifica el suceso B, es decir $P(A|B)$, puede calcularse a partir de la siguiente fórmula:

$$P(A|B) = \frac{P(A,B)}{P(B)} = \frac{P(A)P(B|A)}{P(B)} \tag{A.2}$$

Los elementos que enmarca la Ecuación A.2 son la probabilidad a priori de la hipótesis $P(A)$ y de las observaciones $P(B)$ y las probabilidades condicionadas $P(A|B)$ y $P(B|A)$. A esta última se le conoce como la *verosimilitud* de que la hipótesis A haya producido el conjunto de observaciones B.

La expresión dada en la Ecuación A.2 fue desarrollada por el reverendo Thomas Bayes (1702-1761). A primera vista no es más que una aplicación de las probabilidades condicionales. Empero, ha sido clave en el desarrollo de la inferencia estadística bayesiana que se emplea en la interpretación subjetiva de la probabilidad

A.2. Variables aleatorias y distribuciones de probabilidad

Anteriormente se presentaron los conceptos básicos de probabilidad con respecto a eventos que se encuentran en un espacio muestral. Los experimentos se conciben de manera que los resultados del espacio muestral son cualitativos o cuantitativos. Puede ser útil la cuantificación de los resultados cualitativos de un espacio muestral y, mediante el empleo de medidas númericas, estudiar su comportamiento aleatorio. El concepto de variable aleatoria proporciona un medio para relacionar cualquier resultado con una medida cuantitativa.

Definición A.2.1. Sea S un espacio muestral sobre el que se encuentra definida una función de probabilidad. Sea X una función de valor real definida sobre S, de manera que transforme los resultados de S en puntos sobre la recta de los reales. Se dice entonces que X es una *variable aleatoria*

Se dice que X es "*aleatoria*" porque involucra la probabilidad de los resultados del espacio muestral, y X es una función definida sobre el espacio muestral, de modo que transforma todos los posibles resultados del espacio muestral en cantidades númericas. Es posible definir variables aleatorias cuyos valores sean contables o no. Además, ya que una variable aleatoria es una caracterización cuantitativa de los resultados de un espacio muestral, ésta posee intrínsecamente la naturaleza discreta o continua de este espacio.

Definición A.2.2. Se dice que una variable aleatoria X es *discreta* si el número de valores que puede tomar es contable (ya sea finito o infinito), y si éstos pueden arreglarse en una secuencia que corresponde con los enteros positivos.

Definición A.2.3. Se dice que una variable aleatoria X es *continua* si sus valores consisten en uno o más intervalos de la recta de los reales.

A.2.1. Distribuciones de probabilidad de variables aleatorias discretas

Una variable aleatoria discreta X representa los resultados de un espacio muestral en forma tal que por $P(X = x)$ se representa la probabilidad de que X tome el valor de x. De esta forma, al considerar los valores de una variable aleatoria es posible desarrollar una función matemática que asigne una probabilidad a cada *realización* x de la variable aleatoria X. Esta función recibe el nombre de *función de probabilidad* de la variable aleatoria X. El término más general, *distribución de probabilidad*, se refiere a la colección de valores de la variable aleatoria y a la distribución de probabilidades entre éstos. Sin embargo, hacer referencia a la distribución de probabilidad de X no solamente implica la existencia de la función de probabilidad, sino también la existencia de la *función de distribución acumulativa* de X.

Definición A.2.4. Sea X una variable aleatoria discreta. Se llamará a $P(x) \equiv P(X = x)$ función de probabilidad de la variable aleatoria X, si satisface las siguiente propiedades:

1. $P(x) \geqslant 0$ para todos los valores x de X.

2. $\sum_x P(x) = 1$

Definición A.2.5. La función de distribución acumulativa de la variable aleatoria X es la probabilidad de que X *sea menor o igual a un valor específico de* x y está dada por:

$$F(x) \equiv P(X \leqslant x) = \sum_{x_i \leqslant x} P(x_i) \tag{A.3}$$

A.2.2. Distribuciones de probabilidad de variables aleatorias continuas

La distribución de probabilidad de una variable continua X está caracterizada por una función $f(x)$ que recibe el nombre de *función de densidad de probabilidad*. Esta función $f(x)$ no es la misma función de probabilidad que para el caso discreto. Como existe la probabilidad de que X tome el valor específico donde x es cero, la función de densidad de probabilidad no representa la probabilidad de que $X = x$, más bien, proporciona un medio para determinar la probabilidad de un intervalo $a \leqslant X \leqslant b$. Su definición formal es la siguiente:

Definición A.2.6. Si existe una función $f(x)$ tal que cumpla:

1. $f(x) \geqslant 0$, $-\infty < x < \infty$
2. $\int_{-\infty}^{\infty} f(x)dx = 1$, y
3. $P(a \leqslant X \leqslant b) = \int_a^b f(x)dx$

para cualesquiera a y b, entonces $f(x)$ es la función de densidad de probabilidad de la variable aleatoria continua X

A.3. Distribuciones discretas de probabilidad

A continuación se presentan con detalle algunas distribuciones específicas de probabilidad que han demostrado, empíricamente, ser modelos útiles para diversos problemas prácticos y por lo cual han sido seleccionadas para introducir errores en los conjuntos de datos. Esto mediante su programación en el lenguaje *plpgsql* dentro del SMBD PostgreSQL.

A pesar de ello tales distribuciones presentan un carácter teórico en el sentido en que sus funciones de probabilidad o de densidad de probabilidad se deducen matemáticamente con base en ciertas hipótesis que se suponen válidas para los fenómenos aleatorios. La elección de una distribución de probabilidad para representar un fenómeno de interés práctico debe estar motivada tanto por la comprensión de la naturaleza del fenómeno en sí, como por la posible verificación de la distribución seleccionada a través de evidencia empírica.

En cada caso se expondrán las principales características distintivas de las distribuciones y otras medidas descriptivas.

A.3.1. Distribución binomial

Constituye una de las distribuciones de probabilidad discretas más útiles. Sus áreas de aplicación incluyen inspección de calidad, ventas, medicina y otras. Se puede considerar un experimento donde el resultado es la ocurrencia ("*éxito*") o la no ocurrencia

de un evento ("*fracaso*"), y sea la p la probabilidad de éxito cada vez que se lleva a cabo el experimento y $1-p$ la probabilidad de fracaso. Suponiendo que el experimento se realiza n veces, y cada uno de éstos es independiente de todos los demás, y sea X la variable aleatoria que representa el número de éxitos en los n ensayos. El interés es determinar la probabilidad de obtener exactamente $X = x$ éxitos durante los n ensayos. Las dos suposiciones de la distribución binomial son:

1. La probabilidad de éxito p permanece constante para cada ensayo.

2. Los n ensayos son independientes entre sí.

Para obtener la función de probabilidad, primero se determina la probabilidad de tener en n ensayos x éxitos consecutivos seguidos de $n-x$ fracasos consecutivos. Lo cual se representa mediante:

$$p^x(1-p)^{n-x}.$$

La probabilidad de obtener exactamente x éxitos y $n-x$ fracasos en cualquier otro orden es el producto de $p^x(1-p)^{n-x}$ por el número de órdenes distintos. Este último es el número de combinaciones de n objetos tomando x a la vez. De acuerdo a lo anterior, se tiene la siguiente definición:

Definición A.3.1. Sea X una variable aleatoria que representa el número de éxitos en n ensayos y p la probabilidad de éxito en cualquiera de éstos. Se dice entonces que X tiene una distribución binomial con función de probabilidad:

$$P(x;n,p) = \begin{cases} \frac{n!}{(n-x)!x!}p^x(1-p)^{n-x} & x = 0,1,2,\ldots,n. \\ 0 \text{ para cualquier otro valor.} & 0 \leqslant p \leqslant 1 \text{ para } n \text{ entero} \end{cases} \tag{A.4}$$

Los parámetros de la distribución binomial son n y p. Éstos definen una familia de distribuciones binomiales en donde cada miembro tiene la función de probabilidad determinada por la Ecuación A.4. El término "distribución binomial" proviene del hecho de que los valores de $P(x;n,p)$ para $x = 0,1,2,\ldots,n$ son los términos sucesivos de la expansión binomial $[(1-p)+p]^n$, es decir:

$$\begin{aligned}[(1-p)+p]^n &= \sum_{x=0}^{n} \frac{n!}{(n-x)!x!}p^x(1-p)^{n-x} \\ &= \sum_{x=0}^{n} P(x;n,p)\end{aligned}$$

Pero dado que $[(1-p)+p]^n = 1$ y $P(x;n,p) \geqslant 0$ para $x = 0,1,2,\ldots,n$ este hecho también verifica que $P(x;n,p)$ es una función de probabilidad. Las propiedades básicas de la distribución binomial se encuentran resumidas en la Tabla A.1.

Función de probabilidad			**Parámetros**
$P(x;n,p) = \begin{cases} \frac{n!}{(n-x)!x!}p^x(1-p)^{n-x} \end{cases}$			n entero positivo
$x = 0, 1, 2, \ldots, n$			$p, 0 \leqslant p \leqslant 1$
Media	**Varianza**	**Coeficiente de sesgo**	**Curtosis relativa**
np	$np(1-p)$	$\frac{1-2p}{[np(1-p)]^{1/2}}$	$3 + \frac{[1-6p(1-p)]}{np(1-p)}$

Tabla A.1: Propiedades básicas de la distribución binomial.

A.3.2. Distribución binomial negativa

Sea un escenario binomial en el cual se observa una secuencia de ensayos independientes; la probabilidad de éxito en cada ensayo es constante e igual a p. En lugar de fijar el número de ensayos en n y observar el número de éxitos, sea el hecho de continuar los ensayos hasta obtener exactamente k éxitos. En este caso, la variable aleatoria es el número de ensayos necesarios para observar k éxitos. Estas situación lleva a lo que se conoce como la distribución binomial negativa.

Se desea obtener la probabilidad de que en el n-ésimo ensayo ocurra el k-ésimo éxito. Si se continúan los ensayos independientes hasta que ocurre el k-ésimo éxito, entonces el resultado del último ensayo fue éxito. Antes de´este, habrán ocurrido $k-1$ éxitos en $n-1$ ensayos. El número de maneras distintas en las que pueden observarse lo anterior, es: $\binom{n-1}{k-1}$. Por lo tanto, la probabilidad que interesa esta determinada por:

$$p(n;k,p) = \binom{n-1}{k-1} p^k(1-p)^{n-k} \qquad n = k, k+1, k+2, \ldots \tag{A.5}$$

La expresión anterior es la función de probabilidad conocida como *distribución Pascal*. Mediante el empleo de una sustitución de $n = x+k$, puede obtenerse la distribución binomial negativa, en donde x es el valor de una variable aleatoria que representa el número de fracasos hasta que se observan, de manera exacta, k éxitos.

Definición A.3.2. Sea $X + k$ el número de ensayos independientes necesarios para alcanzar, de manera exacta, k éxitos en un experimento binomial en donde la probabilidad de éxito en cada ensayo es p. Se dice entonces que X es una variable binomial negativa con función de probabilidad:

$$P(x;k,p) = \begin{cases} \binom{k+x-1}{k-1} p^k(1-p)^x & \begin{array}{c} x = 0, 1, 2, \ldots \\ k = 1, 2, \ldots \\ 0 \leqslant p \leqslant 1 \end{array} \\ 0 & \text{para cualquier otro valor.} \end{cases} \tag{A.6}$$

El nombre se debe a que las probabilidades dadas por la Ecuación A.6 corresponden a los términos sucesivos de la expansión binomial de:

$$\left(\frac{1}{p} - \frac{1-p}{p}\right)^{-k} \tag{A.7}$$

Distribución geométrica

Debe notarse que si $k = 1$ en la Ecuación A.6, surge un caso especial de la distribución binomial negativa, que se conoce con el nombre de *distribución geométrica* y cuya función de probabilidad está dada por:

$$P(x;p) = p(1-p)^x, \quad x = 0, 1, 2, \ldots, \quad 0 \leqslant p \leqslant 1 \tag{A.8}$$

La variable aleatoria geométrica representa el número de fallas que ocurren antes de que se presente el primer éxito. Las propiedades básicas de la distribución geométrica se encuentran resumidas en la Tabla A.2.

Función de probabilidad			**Parámetros**
$P(x;k,p) = p(1-p)^x$			$k, k > 0$
$x = 0, 1, 2, \ldots,$			$p, 0 \leqslant p \leqslant 1$
Media	**Varianza**	**Coeficiente de asimetría**	**Curtosis relativa**
$\frac{(1-p)}{p}$	$\frac{(1-p)}{p^2}$	$\frac{2-p}{[(1-p)]^{1/2}}$	$3 + \frac{(p^2-6p+6)}{(1-p)}$

Tabla A.2: Propiedades básicas de la distribución geométrica.

A.4. Distribuciones continuas de probabilidad

A.4.1. Distribución uniforme

Sea un evento en el cual una variable aleatoria toma valores de in intervalo finito, de modo que éstos se encuentran distribuidos igualmente sobre el intervalo. Esto es, la probabilidad de que la variable aleatoria tome un valor en cada sub-intervalo de igual longitud es la misma. Se dice entonces que la variable aleatoria se encuentra *distribuida uniformemente* sobre el intervalo.

Definición A.4.1. Se dice que una variable aleatoria X está distribuida uniformemente sobre el intervalo (a, b) si su función de densidad de probabilidad está dada por:

$$f(x;a,b) = \begin{cases} \frac{1}{(b-a)} & a \leqslant x \leqslant b \\ 0 & \text{para cualquier otro valor} \end{cases} \tag{A.9}$$

La función de densidad de probabilidad de una distribución uniforme es constante en el intervalo (a, b). La distribución uniforme proporciona una representación adecuada para redondear las diferencias que surgen al medir cantidades físicas entre los valores observados y los reales. Las propiedades básicas de la distribución uniforme se encuentran resumidas en la Tabla A.3.

Función de probabilidad			**Parámetros**
$f(x; a, b) = \frac{1}{(b-a)}, \quad a \leqslant x \leqslant b$			$a, \quad -\infty < a < \infty$ $b, \quad -\infty < b < \infty$
Media	**Varianza**	**Coeficiente de asimetría**	**Desviación media**
$\frac{(a+b)}{2}$	$\frac{(b-a)^2}{12}$	0	$\frac{(b-a)}{4}$

Tabla A.3: Propiedades básicas de la distribución uniforme.

Bibliografía

[1] D.J.Ñewman A. Asuncion. UCI machine learning repository, 2007.

[2] Hendrik Blockeel. Experiment databases: A novel methodology for experimental research. In Francesco Bonchi and Jean-François Boulicaut, editors, *KDID*, volume 3933 of *Lecture Notes in Computer Science*, pages 72–85. Springer, 2005.

[3] M. Bohanec and V. Rajkovic. Expert system for decision making, 1990.

[4] Pavel Brazdil and Peter Clark. Learning from imperfect data. *Machine Learning, Meta-reasoning and Logics*, pages 207–232, 1990.

[5] Garry Briscoe and Terry Caelli. *A compendium of machine learning: volume 1: symbolic machine learning*. Ablex Publishing Corp., Norwood, NJ, USA, 1996.

[6] W. L. Buntine and T.Ñiblett. A further comparison of splitting rules for decision-tree induction. *Machine Learning*, 1(8):75–85, 1992.

[7] Goddard Space Flight Center. Discover earth science data and services. `http://gcmd.nasa.gov/records/k_Nearest_Neighbor.html`.

[8] B. Cestnik, I. Kononenko, and I. Bratko. Assistant-86: A knowledge elicitation tool for sophisticated users. *Progress in Machine Learning / S. Press*, pages 31–45, 1987.

[9] W. W. Cohen. Efficient pruning methods for separate-and-conquer rule learning systems. *Proceedings of the 13th International Joint Conference on Artificial Intelligence (IJCAI-93)*, pages 988–994, 1993.

[10] T. Cover and P. Hart. Nearest neighbor pattern classification. *Information Theory, IEEE Transactions on*, 13(1):21–27, 1967.

[11] S. L. Crawford. Extensions to the cart algorithm. *Intl. J. of Man-Machine Studies*, 1(31):197–217, 1989.

[12] Tamraparni Dasu and Theodore Johnson. *Exploratory Data Mining and Data Cleaning*. John Wiley & Sons, Inc., New York, NY, USA, 2003.

[13] Dursun Delen, Glenn Walker, and Amit Kadam. Predicting breast cancer survivability: a comparison of three data mining methods. *Artificial Intelligence in Medicine*, 34(2):113–127, 2005.

[14] Thomas G. Dietterich. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine Learning*, 40(2):139–157, 2000.

[15] Pedro Domingos and Michael J. Pazzani. Beyond independence: Conditions for the optimality of the simple bayesian classifier. In *International Conference on Machine Learning*, pages 105–112, 1996.

[16] R.O. Duda and Hart P.E. *Pattern classification and scene analysis*. John Wiley & Sons, Inc., 1973.

[17] Floriana Esposito, Donato Malerba, and Giovanni Semeraro. A comparative analysis of methods for pruning decision trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(5):476–491, 1997.

[18] T. Fawcett. Roc graphs: Notes and practical considerations for researchers, 2004.

[19] U. M. Fayyad and K. B. Irani. What should be minimized in a decision tree? *National Conference Artificial Intelligence (AAAI'90)*, pages 749–754, 1990.

[20] U. M. Fayyad and K. B. Irani. The attribute selection problem in decision tree generation. *National Conference Artificial Intelligence (AAAI'92)*, pages 104–110, 1992.

[21] U.M. Fayyad, G. Piatetskiy-Shapiro, P. Smith, and U. Ramasasmy. Advances in knowledge discovery and data mining. *AAAI Press/MIT Press*, 1996.

[22] Usama M. Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. From data mining to knowledge discovery: an overview. *American Association for Artificial Intelligence*, pages 1–34, 1996.

[23] R.A. Fisher. The use of multiple measurements in taxonomic problems. *Anals of Eugenics*, 1(7):36–179, 1936.

[24] E. Fix and J. L. Hodges Jr. Discriminatory analysis, non-parametric discrimination: consistency properties. in technical report 21-49-004(4), usaf school of aviation medicine, randolph field, texas, 1951.

[25] Eibe Frank, Mark Hall, and Bernhard Pfahringer. Locally weighted naive bayes. In *Proceedings of the 19th Annual Conference on Uncertainty in Artificial Intelligence (UAI-03)*, pages 249–25, San Francisco, CA, 2003. Morgan Kaufmann.

[26] Jerome H. Freidman, Jon Louis Bentley, and Raphael Ari Finkel. An algorithm for finding best matches in logarithmic expected time. *ACM Trans. Math. Softw.*, 3(3):209–226, 1977.

[27] A. Gammerman and A. R. Thatcher. Bayesian diagnostic probabilities without assuming independence of symptoms. In *Methods of Information in Medicine*, volume 30, pages 15–22, 1991.

[28] Susan Gauch and Il-Yeol Soong, editors. *Proceedings of the Eighth International Conference on Information and Knowledge Management*, New York, 1999. ACM.

[29] J. Gehrke, V. Ganti, R. Ramakrishnan, and W.-Y. Loh. Boat-optimistic decision tree construction. *ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'99)*, pages 169–180, 1999.

[30] Jean Dickinson Gibbons and Subhabrata Chakraborti. *Nonparametric Statistical Inference*. CRC Press, 3th edition, 2003.

[31] PostgreSQL Global Development Group. Postgresql 8.0: Sql conformance. `http://www.postgresql.org/docs/8.0/static/features.html`.

[32] Reija Haapanen and Alan R. Ek. Software and instructions for knn applications in forest resources. *Department of Forest Resources*, 2001.

[33] Jiawei Han and Micheline Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Series, 2000.

[34] D. Hand, H. Mannila, and P. Smyth. *Principles of Data Mining*. MIT Press, first edition, 2001.

[35] D. J. Hand. Construction and assessment of classification rules., 1997. Chichester: Wiley.

[36] David J. Hand and Robert J. Till. A simple generalisation of the area under the roc curve for multiple class classification problems. *Mach. Learn.*, 45(2):171–186, 2001.

[37] David J. Hand and Keming Yu. Idiot's bayes: Not so stupid after all? In *International Statistical Review / Revue Internationale de Statistique*, volume 69, pages 385–398, 2001.

[38] J. Henrichon and K. Fu. A nonparametric partitioning procedure for pattern classification. *IEEE Transactions on Computers*, C-18:604–624, 1969.

[39] J. Huang, J. Lu, and C. X. Ling. Comparing naive bayes, decision trees, and svm with auc and accuracy, 2003.

[40] E. Hunt, J. Martin, and P. Stone. Experiments in induction, 1966.

[41] William H. Inmon. Building the data warehouse. *Wiley-QED*, 1992.

[42] Sonquist J.A. and Morgan J.N. The detection of interaction effects. *Institute for Social Research - University of Michigan*, 1(1), 1964.

[43] Sonquist J.A., Morgan J.N., and Baker E.L. Searching for structure (aid). *Institute for Social Research - University of Michigan*, 1(1), 1971.

[44] Yu Jiangsheng. Method of k-nearest neighbors. *Institute of Computational Linguistics*, 2002.

[45] I. Kononenko. Comparison of inductive and naive bayesian learning approaches to automatic knowledge acquisition. In *B. Wielinga: Current Trends in Knowledge Acquisition*, 1990.

[46] I. Kononenko. Semi-naïve bayesian classifiers. *Proceedings of the 6th European Working Session on Learning*, pages 206–219, 1991.

[47] J. Kubica and A. Moore. Probabilistic noise identification and data cleaning, 2002.

[48] Breiman L., Friedman J.H, Olshen R.A., and Stone C.J. *Classification and Regression Trees*. Chapman & Hall, 1984.

[49] Chuck P. Lam and David G. Stork. Evaluating classifiers by means of test data with noisy labels. In *IJCAI*, pages 513–518, 2003.

[50] P. Langley, W. Iba, and K. Thompson. An analysis of bayesian classifiers. In *Proceedings of the 10th National Conference on Artificial Intelligence*, pages 223–223, 1992.

[51] W. Y. Loh and Y. S. Shih. Split selection methods for classification trees. *Statistica Sinica*, 1(7):815–840, 1997.

[52] Michel M. Manago and Yves Kodratoff. Noise and knowledge acquisition. *IJCAI-87*, pages 348–354, 1987.

[53] J. K. Martin and D. S. Hirschberg. The time complexity of decision tree induction., 1995.

[54] R. S. Michalski. Aqval/1 – computer implementation of a variable-value3d logic system vl and examples of its application to pattern recognition. *Proceedings of the First International Joint Conference on Pattern Recognition*, pages 3–17, 1973.

[55] Sreerama K. Murthy. Automatic construction of decision trees from data: A multi-disciplinary survey. *Data Mining and Knowledge Discovery*, 2(4):345–389, 1998.

[56] John Naisbitt. *Megatrends 2000.* Avon Books, first edition, 1991.

[57] Gonzalo Navarro. A guided tour to approximate string matching. *ACM Computing Surveys*, 33(1):31–88, 2001.

[58] Jennifer Neville, David Jensen, and Brian Gallagher. Simple estimators for relational bayesian classifiers. In *ICDM '03: Proceedings of the Third IEEE International Conference on Data Mining*, page 609, Washington, DC, USA, 2003. IEEE Computer Society.

[59] W. Hersh. Oshumed. An interactive retrieval evaluation and large test collection for research. *ACM SIGIR Conference, Information Retrieval*, pages 192–201, 1994.

[60] D. Pyle. *Data Preparation for Data Mining.* Morgan Kaufmann, Harcourt Intl., 1999.

[61] J. R. Quinlan. Learning efficient classification procedures and their application to chess end games. *Machine learning: An artificial intelligence approach.*, 1983.

[62] J. R. Quinlan. *C4.5: Programs for Machine Learning.* Morgan Kaufmann, 1993.

[63] J. R. Quinlan. Improved use of continuous attributes in C4.5. *Journal of Artificial Intelligence Research*, 4:77–90, 1996.

[64] J. R. Quinlan and R. L. Rivest. Inferring decision trees using the minimum description length principle. *Information and Computation*, pages 227–248, 1989.

[65] J.R. Quinlan. Discovering rules from large collections of examples. *Expert Systems in the Microelectronic Age - Edimburgo University Press*, 1979.

[66] Luis F. Muñoz Roldán. Estrategias de mercadeo en internet. `http://www.slideshare.net/israelrusso/km-23173/`.

[67] J. C. Schlimmer and D. Fisher. A case study of incremental concept induction. *Nat. Conf. Artificial Intelligence (AAAI'86)*, pages 496–501, 1986.

[68] S. Schwarm and S. Wolfman. Cleaning data with bayesian methods, 2000.

[69] J. Shafer, R. Agrawal, and M. Mehta. Sprint: A scalable parallel classiffier for data mining. *Int. Conf. Very Large Data Bases (VLDB'96)*, pages 544–555, 1996.

[70] C. E. Shannon and W. Weaver. *The Mathematical Theory of Communication.* University of Illinois Press, Urbana, Illinois, 1949.

[71] Y.-S. Shih. Families of splitting criteria for classification trees. *Statistics and Computing*, 2000.

[72] R. Sokal and F. Rohlf. Biometry. freeman, 1981.

[73] Qun Sun, Xiuzhen Zhang, and Kotagiri Ramamohanarao. Noise tolerance of ep-based classifiers. In *Australian Conference on Artificial Intelligence*, pages 796–806, 2003.

[74] ZhaoHui Tang and Jamie MacLennan. Datasets for data mining with sql server 2005 book.
`http://wiley.com/tang/datasets`.

[75] Mitchell T.M. *Machine Learning*. McGraw-Hill, 1997.

[76] B. S. Todd and R. Stamper. The relative accuracy of a variety of medical diagnostic programs. In *Methods of Information in Medicine*, volume 33, pages 402–416, 1994.

[77] P. E. Utgoff. An incremental id3. *Fifth Int. Conf. Machine Learning*, pages 107–120, 1988.

[78] Kass V. An explanatory technique for investigating large quantities of categorical data. *Appl. Statist.*, 29(1):119–127, 1980.

[79] Ranga Raju Vatsavai. Olgp: An experimentation with umn-mapserver. *Department CS and Engineering*, 2003.

[80] Rene Villeda-Ruz and Javier Garcia-Garcia. Meaningful error estimations for data analysis. *Conf. Data Mining Workshop ENC'07*, 2007.

[81] Yiming Yang. An evaluation of statistical approaches to text categorization. *Inf. Retr.*, 1(1-2):69–90, 1999.

[82] Yirong Yang, Yi Xia, Yun Chi, and Richard R. Muntz. Ucla computer science department technical report csd-tr no. 030056-1 learning naive bayes classifier from noisy data, 2004.

[83] Zheng Rong Yang. Mining sars-cov protease cleavage data using non-orthogonal decision trees: a novel method for decisive template selection. *Bioinformatics*, 21(11):2644–2650, 2005.

[84] Xie Zhipeng, Wynne Hsu, Liu Zongtian, and Mong-Li Lee. Snnb: A selective neighborhood based naive bayes for lazy learning. In *PAKDD '02: Proceedings of the 6th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, pages 104–114, London, UK, 2002. Springer-Verlag.