Programa de Doctorado en Estadística, Optimización y Matemática Aplicada (EOMA)

Contribuciones al problema de clasificación en machine learning

Yolanda Orenes Casanova

Director de la tesis

Dr. D. Joaquín Sánchez Soriano

Codirector de la tesis

Dr. D. Alejandro Rabasa Dolado

Universidad Miguel Hernández de Elche

Octubre de 2022

La presente tesis doctoral se presenta en formato convencional y como indicios de calidad se presentan los siguientes resultados:

1. Yolanda Orenes, Alejandro Rabasa, Jesús Javier Rodriguez-Sala, Joaquín Sanchez-Soriano. Benchmarking Analysis of the Accuracy of Classification Methods Related to Entropy. Entropy vol. 23(7):850, 2021. https://doi.org/10.3390/e23070850
2. Yolanda Orenes, Alejandro Rabasa, Agustín Pérez-Martín, Jesús Javier Rodríguez-Sala, Joaquín Sanchez-Soriano. A computational experience for automatic feature selection on Big Data frameworks. International Journal of Design & Nature and Ecodynamics vol. 11:168-177, 2016. https://doi.org/10.2495/DNE-V11-N3-168-177

Universidad Miguel Hernández de Elche

D. Joaquín Sánchez Soriano, director de esta tesis doctoral, declaro que el presente trabajo, titulado

Contribuciones al problema de clasificación en machine learning

presentado por Doña Yolanda Orenes Casanova para obtener el título de doctor, fue llevado a cabo bajo mi supervisión en el Programa de Doctorado en Estadística, Optimización y Matemática Aplicada (EOMA) de la Universidad Miguel Hernández de Elche.

Elche, octubre de 2022

El director,

Dr. Joaquín Sánchez Soriano

Universidad Miguel Hernández de Elche

D. Alejandro Rabasa Dolado, co-director de esta tesis doctoral, declaro que el presente trabajo, titulado

Contribuciones al problema de clasificación en machine learning

presentado por Doña Yolanda Orenes Casanova para obtener el título de doctor, fue llevado a cabo bajo mi supervisión en el Programa de Doctorado en Estadística, Optimización y Matemática Aplicada (EOMA) de la Universidad Miguel Hernández de Elche.

Elche, octubre de 2022

El co-director,

Dr. Alejandro Rabasa Dolado

Universidad Miguel Hernández de Elche

D. Domingo Morales González, coordinador del Programa de Doctorado en Estadística, Optimización y Matemática Aplicada (EOMA) de la Universidad Miguel Hernández de Elche, declaro que el presente trabajo, titulado

Contribuciones al problema de clasificación en machine learning

presentado por Doña Yolanda Orenes Casanova para obtener el título de doctor, fue llevado a cabo en el Programa de Doctorado en Estadística, Optimización y Matemática Aplicada (EOMA) de la Universidad Miguel Hernández de Elche.

Elche, octubre de 2022

El coordinador del programa de doctorado,

Dr. Domingo Morales González

A la memoria de mi madre,

gracias mamá, por seguir con todos nosotros, aunque ahora de otra manera.

A mi padre y a mis hermanas, por su apoyo.

A mi marido, por su paciencia y comprensión.

A mi hija, por ser mi mayor motivación.

Agradecimientos

Quiero expresar mi gratitud a la Universidad Miguel Hernández (UMH) y al Centro de Investigación Operativa (CIO) por todos estos años, así como a mis profesores por su gran labor.

También agradecer al profesor Jesús Javier Rodríguez Sala del Centro de Investigación Operativa, por su constante ayuda en esta tesis.

Y sobre todo, dar las gracias a mis directores: Joaquín Sánchez Soriano y Alejandro Rabasa Dolado, por haberme conducido hasta aquí.

Tabla de contenido

Resumen

El problema de clasificación es un tema muy estudiado en la ciencia de datos, en concreto en el campo del aprendizaje automático o "*machine learning*". En la actualidad cada vez hay más información y los agentes económicos y sociales quieren extraer conclusiones relevantes de los datos que les ayuden a tomar mejores decisiones. El problema de clasificación es muy importante en la toma de decisiones en una gran variedad de campos, de hecho, en la literatura se puede encontrar un gran número de métodos que son capaces de realizar las tareas propias de la clasificación. La clasificación es una metodología de aprendizaje supervisado en la ciencia de datos, cuyo propósito es predecir la clase correcta, entre un conjunto de clases conocidas, de una nueva observación dada en base al conocimiento proporcionado por un conjunto de datos previo, también llamado datos de entrenamiento.

En esta tesis doctoral se trabaja el problema de la clasificación en los aspectos siguientes: Se hace una revisión bibliográfica exhaustiva del problema de clasificación. Se compara el análisis discriminante y el método de selección de características, RBS. Se estudia el desempeño de dos conceptos de la teoría de juegos, como técnicas para la selección de características, comparándolos con distintos métodos de selección de características implementados en Weka. Y se definen tres medidas de desempeño para evaluar el rendimiento de un clasificador. A continuación, se desarrolla cada uno de los aspectos anteriores.

En esta tesis se realiza una revisión bibliográfica muy amplia, que queda reflejada a lo largo de toda la memoria por estar estrechamente vinculada con la revisión de la literatura relacionada con el problema de clasificación y en particular, con la selección de características. Todo ello ha servido para elaborar un estado del arte del tema que ha sido muy útil como punto de partida para establecer diferentes problemas abiertos pendientes de estudiar.

Se sabe que una de las dificultades en el análisis de un conjunto de datos es su alta dimensionalidad, lo que puede implicar un peor rendimiento de los clasificadores utilizados. La respuesta más eficaz es reducir la dimensión transformando los datos o la otra alternativa puede ser la

selección de características. En esta tesis se lleva a cabo un estudio computacional en el que se comparan los resultados obtenidos mediante un método de reducción de la dimensión como es el análisis discriminante y un método de selección de características, incorporado en RBS. En dicho estudio se obtiene que en tiempo computacional el análisis discriminante es ligeramente mejor que el método RBS. Sin embargo, en términos de precisión para conjuntos de 1,000,000 de registros, el método de selección de características RBS ofrece mejores resultados.

Además, en esta memoria se lleva a cabo un estudio computacional comparando la selección de características mediante los valores de Shapley y Banzhaf con varios algoritmos de selección de características implementados en Weka. Lo que se hace es definir un juego cooperativo asociado a un problema de clasificación y se calculan los valores de Shapley y Banzhaf asociados a ese juego, seleccionando aquellas características con un mayor valor por considerarse que tienen una mayor influencia en la precisión de la predicción. Finalmente, se compara, para diversos conjuntos de datos, la selección de características obtenidas con los métodos basados en teoría de juegos y los métodos implementados en Weka. Resaltar que, dado el mismo conjunto de datos, no todos los clasificadores son igualmente precisos en sus predicciones. La precisión conseguida por un modelo de clasificación depende de varios factores. Por lo tanto, el análisis del desempeño de los clasificadores es relevante para determinar cuál funciona mejor.

Asimismo, en esta tesis se definen tres medidas de desempeño para evaluar el rendimiento de un clasificador. Se consideran tres clasificadores de referencia, en concreto, dos intuitivos y uno aleatorio. Para evaluar un clasificador se determina la reducción proporcional del error de clasificación cuando se utiliza el clasificador a evaluar con respecto a emplear uno de referencia. Este también es un enfoque interesante de la evaluación del desempeño de los clasificadores porque se puede medir lo ventajoso que es un nuevo clasificador con respecto a los tres de referencia simples, que pueden verse como las mejores opciones basadas en el sentido común. Además, también se analiza la relación entre los tres clasificadores de referencia y diferentes aspectos de la entropía del conjunto de datos. Se lleva a cabo un experimento intensivo para exponer cómo funcionan las medidas de rendimiento propuestas y cómo la

entropía puede afectar el rendimiento de un clasificador. Para validar lo observado en el experimento anterior, se realiza un experimento extensivo utilizando 11 conjuntos de datos y cuatro clasificadores implementados en Weka.

Capítulo 1. Introducción

La ciencia de datos es un campo de conocimiento que es eminentemente interdisciplinar, que integra metodologías y técnicas que proceden de las matemáticas, la estadística y la informática. Su objetivo es tratar de obtener información útil o nuevo conocimiento de enormes cantidades de datos que pueden ser complejos, incongruentes, dinámicos, de múltiples fuentes, estructurados o no estructurados. Dentro de la ciencia de datos, se encuentran el aprendizaje automático[1] (*machine learning*) y el aprendizaje profundo[2] (*deep learning*), los cuales también residen en el campo de la inteligencia artificial. En ambos casos, se busca la construcción de modelos que sean capaces de resolver problemas a partir de conjuntos de datos que sirven para entrenarlos, es decir, de alguna forma aprenden. Este aprendizaje puede ser supervisado o no supervisado. En el primer caso, se parte de conjuntos de datos previamente etiquetados, por tanto, se conocen los valores posibles de aquello que se quiere aprender a resolver, es decir, se sabe cuáles son las posibles soluciones al problema, mientras que en el segundo no. Uno de los subcampos del aprendizaje automático, y por tanto de la ciencia de datos, es el análisis predictivo de datos, que consiste en la construcción de modelos para hacer predicciones basadas en patrones que se extraen de los datos. Entre los problemas relevantes que se pueden encontrar en ciencia de datos está el problema de clasificación.

La clasificación es una metodología de aprendizaje supervisado en la ciencia de datos cuyo propósito es predecir la *clase* correcta, entre un conjunto de *clases* conocidas, de una nueva observación dada en base al

[1] Véase Kubat (2017) para una interesante introducción al aprendizaje automático.
[2] Véase Skansi (2018) para una introducción al aprendizaje profundo.

conocimiento proporcionado por un conjunto de datos previo, conocido como *datos de entrenamiento*.

El problema de clasificación es un tema relevante en la ciencia de datos, en particular en el ámbito del aprendizaje automático o más conocido por su denominación en ingles "*machine learning*" (Aggarwal, 2015; Kelleher et al., 2015; Kubat, 2017; Skiena, 2017). Además, el problema de la clasificación es muy importante en la toma de decisiones en muy diversos campos, por lo que no es difícil encontrar aplicaciones en campos como la medicina, la biotecnología, el marketing, la seguridad en las redes de comunicación, la robótica, el reconocimiento de imágenes y textos, etc. Tres cuestiones básicas del problema de clasificación son la reducción de la dimensión del conjunto de datos, el diseño y la implementación de clasificadores y la evaluación del rendimiento de estos clasificadores.

1.1. Objetivos

Los objetivos que se plantean en esta tesis doctoral están relacionados con diferentes aspectos del problema de clasificación, en particular, son los siguientes:

- Objetivo 1: Estudiar de forma detallada el método de selección de características.
- Objetivo 2: Elaborar una clasificación con las diferentes aplicaciones de uso del método de selección de características más significativas hasta la fecha.
- Objetivo 3: Estudiar la selección de características mediante un modelo matemático basado en la teoría de juegos.
- Objetivo 4: Comparar en distintas situaciones de carga el nuevo método RBS (Rabasa, 2009; Almiñana et al., 2012) con técnicas estadísticas clásicas como el análisis discriminante o similares.
- Objetivo 5: Introducir medidas del desempeño de los clasificadores basadas en benchmarking.

Los objetivos 3 a 5 se contextualizarán y detallarán con mayor profundidad en los capítulos correspondientes. De esta forma se trata de proporcionar al lector toda la información posible sobre el problema

estudiado dentro de cada uno de los capítulos. Obviamente, los objetivos 1 y 2 quedan distribuidos a lo largo de toda la memoria, por estar estos intrínsecamente relacionados con la revisión de la literatura relacionada con el problema de clasificación y, en particular, con la selección de características.

1.2. Materiales y métodos

Los materiales utilizados en el desarrollo de la investigación que se recoge en esta memoria son los habituales en ciencia de datos, a saber, ordenadores, software especializado, lenguajes de programación, conjuntos de datos públicos y, por supuesto, la literatura relacionada.

La investigación se planteó bajo una metodología clásica en el ámbito de la ciencia de datos, que evolucionó a lo largo del periodo de investigación desde la recopilación de información en el campo de estudio hasta la propuesta y testeo de las soluciones planteadas a los problemas estudiados, concluyendo con un análisis cualitativo y cuantitativo de sus características, frente a métodos ya existentes. De forma esquemática la metodología siguió los siguientes pasos:

1. Estudio de técnicas y soluciones existentes.
2. Planteamiento de modelos matemáticos y/o heurísticos de optimización.
3. Implementación de las soluciones.
4. Experimentos computacionales de métodos propios y ajenos.
5. Análisis de los resultados.
6. Comparación y documentación.
7. Conclusiones y propuestas.

1.3. Estructura y resultados

En el capítulo 2 se presenta el problema de clasificación y las cuestiones mencionadas haciendo una revisión sobre algunas de las principales soluciones aportadas en la literatura.

Una de las dificultades más frecuentes que se suele encontrar en el análisis de un conjunto de datos es su alta dimensionalidad, ya que cuando hay demasiadas variables el análisis es más difícil y computacionalmente

costoso, pueden existir, además, variables correlacionadas, variables redundantes o incluso variables que solo introducen ruido. Todos estos problemas pueden conducir a un peor rendimiento de los clasificadores que se utilicen. Por lo tanto, para resolver estas dificultades la solución suele pasar por reducir la dimensión y, para ello, generalmente se utiliza una de dos alternativas:

- reducir la dimensión transformando los datos, o
- seleccionar un subconjunto de características manteniendo la mayor parte de la información en el conjunto de datos, este enfoque se conoce como *selección de características*.

En el capítulo 3 de esta tesis se lleva a cabo un estudio computacional en el que se comparan una técnica de reducción de la dimensión mediante la transformación de los datos, la conocida técnica estadística del análisis discriminante lineal, y el método de selección de características incorporado en RBS (Almiñana et al., 2012). Se analizan los resultados en términos de tiempo y precisión (porcentaje de aciertos) en conjuntos de datos con 1,000, 10,000, 100,000 y 1,000,000 de registros, filas o tuplas, obteniendo en todos los casos que los tiempos de computación son mejores en el análisis discriminante que en el método RBS, aunque las diferencias son inferiores a un segundo para los conjuntos de 1,000, 10,000 y 100,000 registros, y de alrededor de 5 segundos para 1,000,000 de registros. Sin embargo, en términos de precisión, ambos métodos dan resultados similares para los conjuntos de datos de 1,000, 10,000 y 100,000 registros, y es para conjuntos de 1,000,000 de registros donde el método RBS ofrece mejores resultados que el análisis discriminante, superándolo en torno a 2 puntos porcentuales. Este estudio computacional se encuentra publicado en *Y. Orenes, A. Rabasa, A. Pérez-Martín, J.J. Rodríguez-Sala, J. Sánchez-Soriano. A computational experience for automatic feature selection on Big Data frameworks. International Journal of Design & Nature and Ecodynamics vol. 11:168-177, 2016*.

Una ventaja del enfoque de selección de características es que se mantiene el significado original de las variables. En problemas de clasificación, donde existe una variable objetivo nominal (el consecuente), la selección de las variables más relevantes no es un asunto trivial. El tema de la selección de características ya se ha abordado en muchos estudios en

el campo del aprendizaje automático. Entre los primeros artículos sobre la selección de características, en Fu y Cardillo (1967) se propone un método de selección de características y reconocimiento de patrones basado en la programación dinámica inversa probabilística. En Cardillo y Fu (1967) se comparan procedimientos de solución de características basados en funciones divergentes y discriminantes. En Chien (1969) se propone un método secuencial para seleccionar subconjuntos de características en el reconocimiento de patrones mediante el uso de estrategias adaptativas basadas en los resultados de retroalimentación proporcionados por el clasificador. Siguiendo estas ideas, en (Jurs et al., 1969; Jurs, 1970) se desarrollan nuevas estrategias para gestionar un mayor número de características para su aplicación en el análisis de espectros de masas. En Narendra y Fukunaga (1977) se propone un algoritmo de ramificación y poda basado en un esquema de enumeración eficiente con ecuaciones recursivas para la selección de características. Sin embargo, los métodos de búsqueda secuencial flotante se muestran computacionalmente más efectivos que los métodos de ramificación y poda para la selección de características en Pudil et al. (1994). Los algoritmos genéticos se proponen como procedimientos eficientes para la selección de características en problemas de clasificación y reconocimiento de patrones en Siedlecki y Sklansky (1989), Leardi et al. (1992) o Yang y Honovar (1998). En John et al. (1994) y Kohavi y John (1997) se proponen y se estudian las fortalezas y debilidades del enfoque "*wrapper*" para la selección de características en el que se determina el mejor subconjunto de características teniendo en cuenta el propio procedimiento de clasificación (*algoritmo de inducción*) a utilizar. La selección de subconjuntos de características utilizando diferentes criterios de similitud, redundancia y relevancia se estudia en Mitra et al. (2002), Yu y Liu (2004) o Peng et al. (2005). En Trabelsia et al. (2017) se introduce un nuevo método de selección de características de filtro para mejorar el rendimiento del clasificador de conceptos nominales (Meddouri et al., 2014). La selección de características basada en conceptos de la teoría de juegos se puede encontrar en Cohen et al. (2007) o Afghah et al. (2018).

En el capítulo 5 de esta tesis se presenta un estudio computacional comparando la selección de características mediante los valores de Shapley (Shapley, 1953) y Banzhaf (1965) con varios de los algoritmos de selección

de características implementados en Weka (Weka, 2020, 2021). Para ello se define un juego cooperativo asociado a un problema de clasificación y se calculan los valores de Shapley y Banzhaf asociados a ese juego, seleccionando aquellas características con un mayor valor, por considerarse que tienen una mayor influencia en la precisión de la predicción. Finalmente, se compara, para diversos conjuntos de datos, la selección de características obtenidas con los métodos basados en teoría de juegos y los métodos implementados en Weka. También hay varios artículos en los que los algoritmos de selección de características se basan en diferentes conceptos de entropía de la información (Duch et al., 2004; Aremu et al., 2020; Bai et al., 2020; Qu et al., 2020; Revanasiddappa y Harish, 2018; Zhao et al., 2020). En Liu y Yu (2005) se revisan los algoritmos de selección de características para la clasificación y el agrupamiento, y se categorizan para facilitar la elección del algoritmo más adecuado para el análisis de un conjunto de datos en particular. Finalmente, muchos de los procedimientos de selección de características incorporan el uso de su propio clasificador para medir la calidad de la selección, por lo que en muchas ocasiones es posible identificar el método de selección de características con el propio clasificador, como puede ocurrir en los métodos de selección de características tipo wrapper y embebido.

En cuanto a los métodos de clasificación, existen diferentes tipos de algoritmos de clasificación dependiendo de su estructura o principios matemáticos detrás de ellos. Así, se puede encontrar algoritmos de clasificación:

- basados en algoritmos de inducción de árbol de decisión como ID3 (Quinlan, 1986) y su extensión C4.5 (Quinlan, 1992), el algoritmo de clasificación y árbol de regresión CART (Breiman et al., 1984), y los algoritmos de random forest (Ho, 1995, 1998; Breiman, 2001);
- basados en similitudes como los algoritmos de K nearest neighbor (Cover y Hart, 1967; Dasarathy, 1991) y sus extensiones a algoritmos basados en instancias como IBL (Aha et al., 1991);

- basados en métodos de separación en espacios vectoriales tales como algoritmos de support vector machines (Cortes y Vapnik, 1995; Ben-Hur et al., 2001); o
- basados en conceptos y métodos probabilísticos o estadísticos como el análisis discriminante lineal (McLachlan, 2004), la regresión logística o los algoritmos de naïve Bayes (Langley y Thompson, 1994; John y Langley, 1995); entre otros.

Para obtener detalles sobre la clasificación y los problemas de aprendizaje y sus algoritmos véase, por ejemplo, Aggarwal (2015). Además, se puede encontrar en la literatura de machine learning muchos artículos en los que se utilizan diferentes conceptos y métodos de la entropía de la información junto con algoritmos de clasificación de aprendizaje para diseñar nuevos clasificadores que se aplican en diferentes contextos (Ramírez-Gallego et al., 2018; Rahman et al., 2020, Wang et al., 2019; Mannor et al., 2005; Lee et al., 2001; Cleary and Trigg, 1995; Holub et al., 2008; Fujino et al., 2008; Fan et al., 2017; Ramos et al., 2018; Berezinski et al., 2015).

Dado el mismo conjunto de datos, no todos los clasificadores son igualmente precisos en sus predicciones. La precisión conseguida por un modelo de clasificación depende de varios factores como la propia implementación del algoritmo, las heurísticas de poda y "*boosting*" incorporadas, el conjunto de datos utilizado, e incluso el conjunto de variables finalmente escogidas para la construcción del modelo. Por lo tanto, el análisis del desempeño de los clasificadores es relevante para determinar cuál funciona mejor. Existen diferentes medidas del desempeño de un clasificador y en la literatura de aprendizaje automático se puede encontrar varios trabajos que analizan el desempeño de diferentes clasificadores de acuerdo a esas medidas. En Costa et al. (2007) se muestra que las medidas de evaluación más usuales en la práctica son inadecuadas para los clasificadores jerárquicos y revisa las principales medidas de evaluación para los clasificadores jerárquicos. En Sokolova y Lapalme (2009) se analizan cómo los diferentes tipos de cambios en la matriz de confusión afectan las medidas de desempeño de los clasificadores. En Ferri et al. (2009) se lleva a cabo una experiencia computacional para analizar 18 medidas diferentes de rendimiento de clasificadores. En Parker (2011) se analizan las incoherencias de siete

medidas de rendimiento para clasificadores binarios desde un punto de vista tanto teórico como empírico para determinar qué medidas son mejores. En Labatut y Cherifi (2011) se estudian las propiedades y el comportamiento de 12 medidas de rendimiento para clasificadores multiclase planos. En Jiao y Du (2016) se revisan las medidas de rendimiento más comunes utilizadas en los predictores bioinformáticos para las clasificaciones.

El capítulo 4 de esta tesis se centra en la definición de medidas de rendimiento siguiendo las ideas de los coeficientes de concordancia en estadística, en particular, el coeficiente κ de Cohen (Cohen, 1960) y el coeficiente π de Scott (Scott, 1955). Estos coeficientes también se utilizan como medidas de rendimiento de los clasificadores (Witten y Frank, 2005) comparando las precisiones del clasificador a evaluar y un clasificador aleatorio. En este capítulo se analizan estas medidas de rendimiento desde otro punto de vista y se definen tres nuevas medidas de rendimiento basadas en el coeficiente π de Scott. En particular, se adopta la interpretación dada en Goodman y Kruskal (1954) para el estadístico o coeficiente λ. Se consideran tres clasificadores de referencia, el clasificador aleatorio y dos clasificadores intuitivos, cada uno de ellos utilizando diferentes niveles de información del conjunto de datos. Así, para evaluar un clasificador se determina la reducción proporcional del error de clasificación cuando se utiliza el clasificador a evaluar con respecto a emplear uno de los clasificadores de referencia. Este también es un enfoque interesante de la evaluación del desempeño de los clasificadores porque se puede medir cuanto de ventajoso es un nuevo clasificador con respecto a tres clasificadores de referencia simples que pueden verse como las mejores opciones basadas en el sentido común para los no expertos (pero suficientemente inteligentes) y cuyas tasas de error son más sencillas de determinar que el error de Bayes (Fukunaga, 1990). Por otro lado, se analiza la relación entre los tres clasificadores de referencia y diferentes aspectos de la entropía del conjunto de datos. Se lleva a cabo un experimento intensivo para ilustrar cómo funcionan las medidas de rendimiento propuestas y cómo la entropía puede afectar el rendimiento de un clasificador. Para validar lo observado en el experimento anterior, se lleva a cabo un experimento extensivo utilizando cuatro clasificadores implementados en Weka y 11 conjuntos de datos. Los resultados de este

capítulo han sido publicados en *Y. Orenes, A. Rabasa, J.J. Rodríguez-Sala, J. Sánchez-Soriano. Benchmarking Analysis of the Accuracy of Classification Methods Related to Entropy. Entropy vol. 23(7):850, 2021.*

En el capítulo 6 se presentan las principales conclusiones que se han obtenido en el desarrollo de la investigación, aquellos aspectos que quedarían pendientes de resolver, así como las futuras líneas de investigación que podrían desarrollarse a partir de los resultados ya obtenidos.

Finalmente, esta tesis concluye con la bibliografía utilizada y los enlaces donde se pueden encontrar los conjuntos de datos que se han utilizado en las experiencias computacionales.

Capítulo 2. El problema de clasificación en la ciencia de datos

La toma de decisiones es un hecho al que se tienen que enfrentar diariamente las personas para realizar una elección entre diferentes opciones y contextos. Antes se deben analizar las consecuencias de tomar una u otra opción. De igual manera ocurre con la toma de decisiones llevadas a cabo por la inteligencia artificial y los sistemas expertos. Hay sistemas que son capaces de clasificar y por tanto tomar decisiones, aunque previamente necesitan un entrenamiento.

En este capítulo se define el problema de clasificación automática. También se exponen algunos métodos de clasificación más frecuentes que se utilizan para la toma de decisiones en machine learning. Es importante saber el rendimiento de los distintos clasificadores, por lo que se ven algunas medidas de desempeño; así como varias técnicas de reducción de la dimensión de los datos que son las que evitan que los clasificadores manejen información irrelevante ayudándoles a tener un mejor rendimiento.

2.1. Introducción

Con el surgimiento de la Web, y más concretamente de la Web 2.0 y las redes sociales, se empezó a generar información de manera muy voluminosa y a gran velocidad que había que gestionar apareciendo así el concepto de Big Data.

Big Data es un término que hace referencia al problema de tener que administrar un gran volumen de datos, estructurados y no estructurados, generados a gran velocidad por usuarios, dispositivos loT (internet of things) o cualquier otro medio que haga que los datos sean variados y muy

abundantes. El concepto de Big Data incluye todas las fases del procesamiento y análisis de datos, es decir, recopilación, preparación, (o pre-procesamiento), análisis (o procesamiento) e interpretación de los resultados con el fin de generar patrones que ayuden a tomar decisiones en cualquier ámbito. Para analizar esos datos masivos surgen nuevas técnicas que complementan a las técnicas clásicas.

La minería de datos es una disciplina entre la estadística y la informática que se emplea en la fase de análisis de datos y sirve para extraer información implícita en grandes volúmenes de datos, que puede venir en diferentes formas como reglas, relaciones o patrones ocultos.

Alan Turing es considerado uno de los padres de la computación. Tuvo excelentes aportaciones en algunos conceptos fundamentales del aprendizaje automático y de la inteligencia artificial, desarrolló una máquina para descifrar mensajes secretos y en 1950 desarrolló el denominado test de Turing, que consiste en una prueba para evaluar la capacidad de una máquina para mostrar un comportamiento similar al de un ser humano.

Según Michell, (1997), el aprendizaje automático consiste en que "un programa de computadora aprende de la experiencia E con respecto a alguna clase de tareas T y una medida de desempeño P, si su desempeño en las tareas en T, medido por P, mejora con la experiencia".

Para Alloghani et al. (2019) el aprendizaje automático es una ramificación de la inteligencia artificial que se encarga de implementar e innovar algoritmos para hacer posible que los ordenadores sean capaces de aprender. El aprendizaje automático está creciendo fuertemente en muchas áreas, como la educación (Gong y Wang, 2011) y la medicina (Bax et al., 2018), entre otras.

El aprendizaje automático se encarga de construir algoritmos avanzados que simulan el aprendizaje de los humanos, de ahí que formen parte de la inteligencia artificial. Estas técnicas se basan en el entrenamiento de modelos que, tras aprender de los datos existentes, son capaces de analizar una gran cantidad de datos y poder tomar decisiones sin que ninguna regla (if then) preestablecida se las muestre. Existen

principalmente dos tipos de algoritmos de aprendizaje: para aprendizaje supervisado y para no supervisado.

En el aprendizaje no supervisado no hay ningún tipo de entrenamiento antes del aprendizaje. Por ejemplo, se pueden encontrar algoritmos de clustering (Lee, 2001; Liu, 2005) en los que se agrupan los datos por similaridad y de reducción de dimensión en los que se simplifican las variables manteniendo el significado original de los datos.

Los algoritmos de aprendizaje supervisado son aquellos que aprenden relacionando entradas con salidas de datos. A su vez hay dos tipos de problemas de aprendizaje supervisado, los de regresión en los que dada una entrada, su salida es un valor numérico continuo, y los de clasificación cuya salida es un valor categórico y son en los que nos vamos a centrar en esta tesis.

Los algoritmos de aprendizaje supervisado realizan su aprendizaje a través de la secuencia de pasos que se ilustra a continuación (figura 2.1):

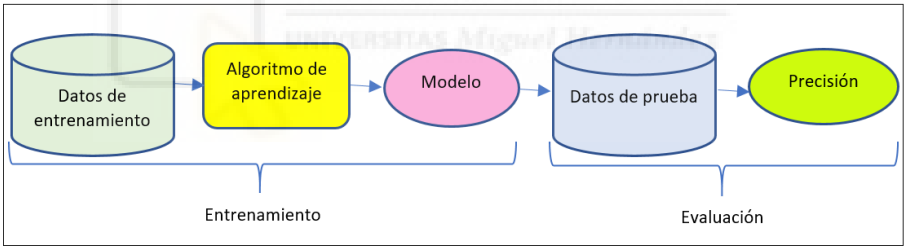

Figura 2.1: Aprendizaje supervisado. Fuente: Elaboración propia a partir de (Liu, 2011)

En la figura 2.1 se puede ver cómo, a partir de unos datos de entrenamiento, se utiliza un algoritmo de aprendizaje para entrenar un modelo, para el cual, se evalúa su nivel de precisión, utilizando una serie de datos de prueba diferentes a los que se utilizaron en la fase de entrenamiento.

2.2. El problema de clasificación

El problema de clasificación es un tema muy estudiado en minería de datos. En estos problemas, el objetivo consiste en predecir el valor de una característica especial de los datos que se denomina variable de clase o variable objetivo, de manera que se analizan los datos de entrenamiento para aprender las relaciones del resto de características con respecto a esta característica especial, la variable objetivo. El modelo aprendido, normalmente se utiliza para obtener las etiquetas de clase estimadas para nuevos registros, donde el valor de la variable de clase no es conocido (Aggarwal, 2015).

A continuación, se define formalmente el problema de clasificación:

"Dada una matriz D de datos de entrenamiento nxd y un valor de la etiqueta de clase en $\{1 \ldots . \mathrm{k}\}$ asociado con cada una de las n filas en D, crear un modelo de entrenamiento M, que se puede usar para predecir la etiqueta de clase de un registro d-dimensional $\bar{y} \notin D$" (Aggarwal, 2015).

Para Strumbelj y Kononenko (2010), la clasificación en el aprendizaje automático, es una modalidad de aprendizaje supervisado cuyo objetivo es predecir la etiqueta de clase para instancias de entrada sin etiquetar, siendo cada instancia descrita por valores de características pertenecientes a un espacio de características. Las predicciones se basan en conocimientos aprendidos de una muestra de instancias etiquetadas durante el entrenamiento.

"El espacio de características A es el producto cartesiano de n características (representadas con el conjunto $N = \{1,2,\ldots,\mathrm{n}\}$): $\mathrm{A} = \mathrm{A}_1 \times \mathrm{A}_2 \times \ldots \times \mathrm{An}$, donde cada característica A_i es un conjunto finito de valores de características."

En definitiva, según se ha visto en las definiciones anteriores, en la clasificación se emplea un conjunto de datos de ejemplo para aprender la estructura de los grupos, por lo que se considera aprendizaje supervisado, de manera que los datos de entrenamiento de ejemplo son imprescindibles para conseguir información sobre la composición de las clases. Por otro lado, para validar el modelo se utilizan datos no vistos anteriormente, también llamados, ejemplos o instancias de prueba, que deben ser

clasificadas por el modelo. Las clases o grupos formados por un modelo de clasificación en los ejemplos de prueba reflejan la estructura y el número de los grupos que se encuentran en el conjunto de datos de entrenamiento.

Entre las aplicaciones de las técnicas de clasificación se resaltan algunos autores que utilizan dichas técnicas aplicadas a distintos ámbitos, como por ejemplo, Wittekind y Tischoff (2004) en medicina con el fin de clasificar tumores para poder realizar un correcto tratamiento oncológico. Lin et al. (2011) en el ámbito de las finanzas con el objetivo de descubrir características financieras potencialmente útiles. Cheung y Li (2012) en marketing para averiguar oportunidades comerciales en las ventas y el marketing de nuevos productos. Cohen et al. (2005) en recursos hídricos con el fin de desarrollar modelos para la predicción categórica/continúa basada en datos de presencia/ausencia para especies y grupos de especies ecológicamente apropiados.

2.3. Métodos de clasificación

La clasificación automática consiste en utilizar modelos que permiten asignar una categoría o clase a diversas observaciones de entrada. En la literatura científica existen un gran número de métodos que son capaces de entrenar dichos modelos. Los métodos de clasificación más frecuentemente utilizados se pueden englobar en alguna de las siguientes cuatro categorías:

- Basados en algoritmos de inducción.
- Basados en similitudes.
- Basados en técnicas de separación en espacios vectoriales.
- Basados en conceptos y algoritmos probabilísticos o estadísticos.

En los siguientes subapartados se describen con mayor detalle varias de las técnicas de cada una de estas categorías.

2.3.1. Métodos basados en algoritmos de inducción

Se pueden destacar dos grupos de técnicas dentro de los algoritmos de inducción de árboles y reglas de decisión. Un grupo lo forman ID3

(Quinlan, 1986), su extensión C4.5 (Quinlan, 1992), además de su implementación en Java en el software Weka (Weka, 2020, 2021) J48. En el otro grupo está el algoritmo de clasificación y árbol de regresión CART (Breiman et al., 1984), y los algoritmos de random forest (Ho, 1995, 1998; Breiman, 2001).

Árboles de decisión

Uno de los modelos más conocido para la clasificación es el árbol de decisión. La selección del atributo utilizado en cada nodo del árbol para dividir los datos es fundamental para clasificar correctamente el conjunto de datos.

Estos modelos basados en árboles de decisión o frecuentemente llamados TDIDT (Top Dawn Induction of Decision Trees) se caracterizan por utilizar el algoritmo divide y vencerás (Kubat, 2017) siguiendo un criterio de división determinado.

Los árboles de decisión se llaman árboles de clasificación cuando el modelo es capaz de determinar la categoría o clase a la que pertenece un dato, y árboles de regresión cuando el modelo es capaz de predecir el valor de una variable continua.

Esta estructura se representa como un diagrama en forma de árbol y visto desde una perspectiva descendente, está formado por el nodo inicial (o nodo raíz) que despliega sus ramas hacia nodos inferiores llamados nodos internos. Cada nodo es un atributo en los árboles de decisión univariables o paralelos, y varios atributos en los multivariables (Brodley y Utgoff, 1992). En cada uno de los nodos internos, se establece un criterio de división o condición. Dicho criterio divide los datos de entrenamiento en dos o más ramas, cuya elección depende de si cumple o no la condición asociada a cada rama, hasta llegar a ser clasificados en el nodo hoja (Aggarwal, 2015). Es decir, cada rama es uno de los posibles valores de ese atributo y el nodo hoja contiene la etiqueta de clase con la predicción.

En definitiva, el entrenamiento del árbol consiste en ir generando divisiones para conseguir agrupaciones de datos lo más uniformes posible que pertenezcan a una única categoría. Esta fase del proceso de

clasificación con árboles de decisión tiene como punto de partida fijar una condición en el nodo raíz del que saldrá la primera partición binaria en dos subregiones, representadas por dos ramas que informan si el ejemplo cumple o no la condición. En dichas subregiones se pretende que en una de ellas haya un subgrupo de datos lo más homogéneo posible y cuando esto sea así ya no se divide más, mientras que los siguientes nodos que contienen el resto de condiciones continuarán de igual manera con la partición hasta llegar a los nodos hoja, a partir de los cuales, ya no se pueden generar más particiones.

Como se ha comentado anteriormente durante la construcción del árbol de decisión se llevan a cabo una serie de particiones que siguen un determinado criterio de división. Entre las diferentes medidas que permiten aplicar dicho criterio para elegir el atributo que llevará a cabo las particiones del árbol se encuentran las siguientes:

- Ganancia de Información
- Gain Ratio
- Índice de Gini

A continuación se describen estas tres medidas:

Ganancia de información, es una medida que utiliza el algoritmo ID3, está basada en la teoría de la información de Claude E. Shannon (Shannon, 1948).

Los ítems de un conjunto de ejemplos E se clasifican como $\mathrm{Clase}_1, \mathrm{Clase}_2, \ldots, \mathrm{Clase}_\mathrm{n}$ de manera que las probabilidades de que se encuentren elementos de cada una de las clases son $p_1, p_2, \ldots, p_n$. El concepto de entropía de información $I(E)$ se define como

$$I(E) = -\sum_{i=1}^{n} p_i log_2(p_i). \quad (2.1)$$

Dado un atributo A del conjunto anterior cuyos valores que puede tomar son $A_1, A_2, \ldots, A_v$, se definen las particiones $E_1, E_2, \ldots, E_v$, del conjunto E de manera que cada partición E_i contiene los ejemplos de E con valor A_i para el atributo A. La ganancia de información del atributo A, $IG(A)$ viene dada por

$$\mathrm{IG(A)} = I(E) - \sum_{i=1}^{v} p(E_i) I(E_i). \quad (2.2)$$

Siendo $p(E_i)$ la probabilidad de hallar un ítem del conjunto E_i dentro de E. Para elegir el mejor atributo por el que extender cada uno de los nodos del árbol de decisión ID3 selecciona el que maximiza la ganancia de información. Puesto que la entropía de información permanece constante para un mismo nodo del árbol, maximizar la ganancia de información es lo mismo que minimizar el sumatorio de la expresión (2.3) (Rodríguez-Sala, 2014).

$$\sum_{i=1}^{v} p(E_i) I(E_i). \tag{2.3}$$

Para llevar a cabo la clasificación de un dato nuevo el algoritmo verifica las condiciones que se cumplen en cada nodo del árbol entrenado previamente. El proceso termina cuando el dato se etiqueta con la clase asignada.

Gain ratio, es una medida que utiliza C4.5 (Trabelsia et al., 2017). C4.5, con la finalidad de subsanar algunas dificultades de su antecesor, utiliza la razón de ganancia o gain ratio, ya que en ID3 la ganancia de información tiene tendencia a favorecer las características con más valores. Gain ratio es una métrica que aporta un criterio para ordenar las variables explicativas por orden de importancia frente a la variable objetivo. Dicha medida es importante cuando los datos se distribuyen uniformemente y es pequeña cuando los datos pertenecen a una rama. Tiene como objetivo penalizar la proliferación de nodos y se calcula usando la ecuación siguiente:

$$GR\ (att) = \frac{\text{IG (att)}}{\text{H (att)}}. \tag{2.4}$$

Siendo $IG(att)$ la ganancia de información de cada atributo att. Se presenta como la diferencia entre la entropía de la clase y la entropía del atributo. Es decir, es una medida para decidir la relevancia de un atributo, por lo tanto, evalúa atributos. Su fórmula viene representada por

$$IG(att) = H(Y) - H_{att}(Y). \tag{2.5}$$

Donde $H(Y)$ calcula la entropía de la clase y corresponde a la cantidad de información contenida en una fuente de información. Utiliza para ello la fórmula siguiente:

$$H(Y) = \sum_i -P(v_i) log_2 P(v_i). \tag{2.6}$$

En el caso de $P(v_i)$, para una clase i, representa la probabilidad de tener el valor v_i contribuyendo a los valores generales.

Lo que mide $H_{att}\ (Y)$ es la entropía del atributo att contribuyendo a la clase Y (Trabelsia et al., 2017). Para ello se vale de la expresión

$$H_{att}(Y) = \sum_{j=1}^{K} P_j H_{att}(Y_j). \quad (2.7)$$

De manera que K representa el número total de particiones por clase.

P_j es la probabilidad de tener el número de instancias de una partición de atributos.

Y_j presenta el número de una j-ésima partición por clase.

Índice de Gini, es una medida que utiliza CART. Dicho algoritmo emplea este índice como criterio de división. Este índice se considera una medida de pureza del nodo. Y se utiliza para medir el poder discriminativo de un atributo. Sea $\mathrm{v_1 \ldots v_r}$ los r valores posibles de un determinado atributo categórico, y sea p_j la fracción de puntos de datos que contienen el valor del atributo v_i que pertenecen a la clase $j \in \{1 \ldots \mathrm{k}\}$ para el valor del atributo v_i. Entonces, el índice de Gini $G(v_i)$ para el valor v_i de un atributo categórico se define (Aggarwal, 2015) como sigue:

$$G(v_i) = 1 - \sum_{j=1}^{k} p_j^2. \quad (2.8)$$

Entre las ventajas de los árboles de decisión se destaca que son muy utilizados por su velocidad de ejecución y porque gracias a su representación gráfica son fáciles de interpretar. Esto último se refiere a que tan solo se tiene que mirar el árbol para saber rápidamente cuáles son los atributos o variables explicativas más importantes y la dependiente. Además, al observar las condiciones establecidas en los nodos se puede entender la tendencia de los datos que son clasificados en una u otra categoría. En cuanto a sus desventajas resaltar el sobreajuste (overfitting) ya que clasifican muy bien durante el entrenamiento sobre todo si el árbol es muy frondoso (Kubat, 2017), pero no son capaces de generalizar y ajustarse tan bien a los datos nuevos siendo esto último lo que realmente se busca. Por lo que para evitar que se rebaje la precisión en relación al sobreajuste, se lleva a cabo la poda que reduce el tamaño del árbol

eliminando secciones o nodos que no aportan información importante para clasificar instancias mejorando la precisión predictiva mediante la disminución del sobreajuste.

En los árboles de decisión, la profundidad es la trayectoria más larga entre el nodo raíz y uno de los nodos hoja. A continuación, en el árbol de decisión sencillo de ejemplo (figura 2.2) se puede ver que su profundidad es dos. Se trata de un conjunto de atributos más importantes (en relación al clima) y sus dos clases P (positivo) y N (negativo).

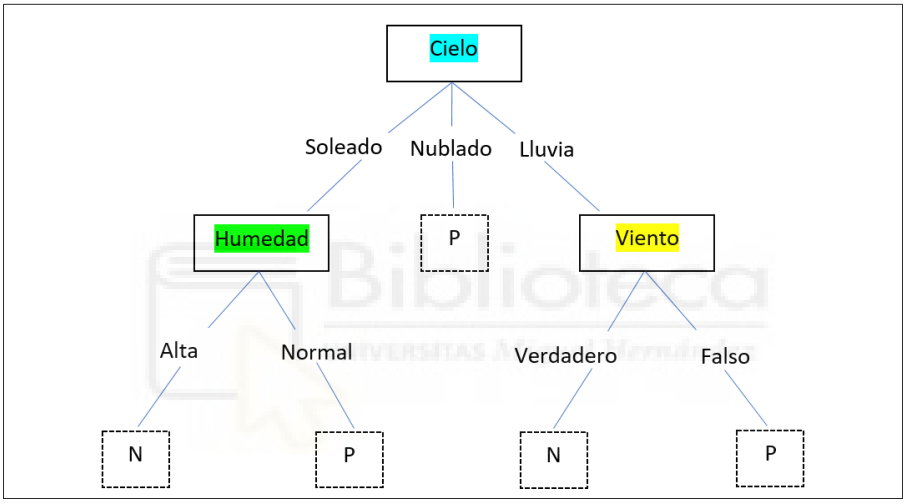

Figura 2.2: Árbol de decisión. Instancias positivas "P" e instancias negativas "N". Fuente: Elaboración propia a partir de Quinlan 1986

La figura 2.2 muestra un árbol de decisión aprendido. Este árbol de decisión clasifica las instancias como P (positivo) y N (negativo) en función de que se juegue o no al tenis.

Se pueden destacar los siguientes algoritmos de inducción de árboles de decisión:

- ID3
- C4.5
- J48
- CART

- Random forest

En cuanto al primer punto, el algoritmo de clasificación ID3 (induction of decision trees) fue desarrollado por J.R. Quinlan en 1986. Se le considera el punto de partida que propulsó estos sistemas llamados inducción mediante árboles de decisión o sencillamente árboles de decisión. Esta técnica emplea la estrategia divide y vencerás para realizar la clasificación. ID3 construye de forma recursiva un árbol de decisión paralelo, utilizando el criterio de división de la medida de información de Shannon. Para seleccionar el atributo se utiliza la medida de ganancia de información que está muy condicionada por la entropía (incertidumbre). Aquel atributo que tenga mayor ganancia será el elegido para llevar a cabo la división recursivamente. Cada uno de los atributos de los nodos internos se evalúan y de su valor depende la ruta a seguir para clasificar un ejemplo. Cuando todos los ejemplos pertenecen a la misma clase, el proceso finaliza y aparece el nodo hoja que contiene la etiqueta de clase. De manera que la clasificación de un ejemplo se lleva a cabo recorriendo el árbol desde la raíz hasta una de las hojas, lo que designará su clase. El algoritmo ID3 ha sido aplicado en diversos campos como por ejemplo en medicina para la predicción de enfermedades (Yang et al., 2018), en educación para analizar el aprendizaje independiente de la red de los estudiantes universitarios (Wang, 2017) y en seguridad de computadoras como los sistemas de detección de intrusos (Eesa et al., 2015).

En relación al segundo punto, C4.5 es un algoritmo desarrollado también por Quinlan (Quinlan, 1992). Se trata de una extensión del algoritmo ID3 por lo que igualmente es considerado un árbol de decisión. Algunas de las mejoras del algoritmo C4.5 respecto a su predecesor son las siguientes:

- Utiliza atributos discretos como ID3 y además continuos (discretización previamente). De manera que los nodos internos pueden establecer condiciones sobre un atributo A continuo cuya evaluación tendrá dos salidas $\mathrm{A} \leq \mathrm{Z}$ y $\mathrm{A} > \mathrm{Z}$, cotejando A y el umbral Z. Por el contrario, si el atributo es discreto solo hay una salida como en ID3.

- Usa los datos aún con valores de atributos faltantes. Lo que hace es que esos valores no disponibles de los atributos no los utiliza a la hora de calcular la ganancia de información. Solo usa los valores que existen.
- Modifica el criterio de división evitando la preferencia de seleccionar los atributos con muchos valores distintos para realizar la evaluación. Apostando por aquellos atributos que teniendo una ganancia de información menor tienen un menor número de valores para realizar la clasificación. Esto se soluciona usando el ratio de ganancia (gain ratio) como criterio de separación que tiene en cuenta la ganancia de información y las probabilidades del atributo (split information). Esta última es la entropía del conjunto de datos respecto a los valores del atributo en cuestión.

El algoritmo C4.5 ha sido aplicado en analizar la evaluación de la calidad de la enseñanza (Lu y Chen, 2016), estudiar el reconocimiento del color (Chen y He-Gen, 2016), prevenir y controlar los desastres causados por estallidos de rocas en las minas de carbón (Wang, 2021), aumentar la calidad del habla sintética de los sistemas de texto a voz, en presencia de siglas en el texto de entrada (Monzo et al., 2006) y en varios ámbitos de la medicina (Pradana et al., 2019), entre otras aplicaciones.

Referente al tercer punto, J48 es una implementación en Java (en el software Weka (Weka, 2020, 2021) de minería de datos) del algoritmo original de árbol de clasificación C4.5. J48 extiende algunas de las funcionalidades de C4.5 como permitir que el proceso de post-poda del árbol se realice por un método basado en la reducción de errores o que las divisiones sobre variables discretas sean siempre binarias, entre otras. Sobre todo, destacar uno de los parámetros de configuración de J48 por ser muy importante "confidenceFactor", el factor de confianza para la poda, puesto que influye mucho en el tamaño y capacidad de predicción del árbol creado. Consiste en lo siguiente: para cada operación de poda, define la probabilidad de error que se permite. Este factor tiene un valor establecido del 25%. Se pueden ir reduciendo los árboles bajando el valor de ese porcentaje.

Atendiendo al cuarto punto, CART o algoritmo de clasificación y regresión fue desarrollado por Breiman et al. (1984). Su metodología consiste en construit el árbol, elegir el tamaño correcto y clasificar nuevos datos a partir del árbol generado. Cuando la variable dependiente sea de tipo cualitativo son árboles de clasificación y cuando la variable dependiente sea continua son árboles de regresión. Estos árboles son capaces de conseguir agrupaciones homogéneas en cada una de las particiones que lleva a cabo para obtener el valor de la variable objetivo más probable para cada registro. El algoritmo CART en el caso de la clasificación mide con el índice de Gini la homogeneidad de cada una de las dos particiones de un nodo. Cuando el valor de dicho índice es cero significa que los nodos hoja tienen datos que pertenecen a una sola categoría y cuando son mayores que cero contienen impurezas por lo que se puede seguir con más particiones. En el caso de la regresión la métrica que se utiliza es el error cuadrático medio y la función de costo.

En cuanto al funcionamiento del algoritmo CART para realizar la clasificación consiste en generar particiones con agrupaciones muy homogéneas de la siguiente manera:

Se crea la partición del nodo raíz que es de donde se parte, utilizando todos los atributos y para cada uno de ellos se establece un punto intermedio entre dos valores consecutivos de cada atributo. Para cada uno de esos puntos intermedios se obtiene la partición que corresponde al nodo izquierdo y nodo derecho. Se calcula el índice Gini para cada nodo hijo. Se calcula la función de costo del nodo padre (es decir, el promedio ponderado de los índices Gini de los hijos). Se coge el menor valor en la función de costo del nodo padre que significa que la partición que resulta es la mejor, pues es la más homogénea. El proceso se repite iterativamente para todos los nodos obtenidos, menos los nodos hoja.

Respecto al funcionamiento del algoritmo CART para la regresión con árboles de decisión funciona de la siguiente manera:

Se calculan los puntos intermedios entre los dos valores consecutivos de cada atributo, para cada uno se obtiene las dos particiones calculando sus errores cuadráticos medios y su función de costo. Se selecciona el que

tenga menor costo. Se repite el proceso iterativamente hasta que se den las condiciones para que no se pueda dividir más (homogeneidad).

El algoritmo CART se ha utilizado para lograr un diagnóstico completo de fallas en línea para sistemas de rotor, como ubicación de fallas y tipos de fallas (Deng, 2020) y para métodos inteligentes de conducción segura para trenes de alta velocidad (Cheng, 2019), entre otras aplicaciones.

El último punto es random forest (o bosques aleatorios) es un algoritmo de clasificación y regresión que fue desarrollado por Breiman en 2001. Es importante resaltar que aunque en los árboles de decisión, la precisión de generalización de los datos nuevos puede verse afectada por la precisión de los datos de entrenamiento; sin embargo esta dificultad no es intrínseca a los árboles de clasificación. Random forest es un método basado en árboles de decisión que permite mejorar ambas precisiones (Ho, 1995).

Los clasificadores de bosques aleatorios (Breiman, 2001) constan de conjuntos de árboles de decisión que se construyen a partir de subconjuntos seleccionados aleatoriamente del conjunto de entrenamiento, y cada árbol emite un voto. De manera que la predicción final es el resultado con más votos. La aleatoriedad permite que cada árbol sea entrenado con un subconjunto de datos diferente. Para conseguir la predicción de un nuevo dato, con el bosque ya entrenado, los resultados de todos los árboles separados se agregan en una sola predicción como resultado final. En el caso de la clasificación cada árbol predice una clase de manera que se recogen todos los votos generados. Por lo que el resultado final de la predicción del bosque es la clase con frecuencia más alta, es decir, la más votada.

En la regresión cada árbol arroja una predicción. Se calcula el promedio de todos los valores de salida de los árboles del bosque para obtener la predicción. En estos modelos se crean numerosos árboles en paralelo sin prácticamente ninguna restricción para la construcción del bosque, por lo que no hay ajuste de un único árbol y no hay poda.

Algunas de las características más importantes del algoritmo random forest son las siguientes (Hultström, 2013):

- Se generan gran cantidad de árboles, no un único árbol. Estos se construyen a partir de muchos conjuntos de datos similares generados mediante bootstrap (remuestreo con reposición) de la muestra original.
- La aleatoriedad en este modelo es introducida para cada división de un nodo, es decir, se selecciona al azar un subconjunto de las p variables y se restringe la selección de la variable a este subconjunto. De manera que, no se selecciona la mejor variable de entre todas.

Algunas de las ventajas que tienen estos bosques es que no presentan el problema del sobreajuste, ya que se toma el promedio de todas las predicciones y son muy precisos, pues intervienen en el proceso muchos árboles. Una de las desventajas de los bosques aleatorios es que son difíciles de entender.

Este algoritmo se utiliza, entre otras aplicaciones, para mejorar las capacidades de procesamiento de datos del sistema de gestión de información financiera del negocio inmobiliario (Liu, 2018), para procesamiento de imágenes de detección remota en el océano (Sun, 2017) y para interpretación sísmica (Ao, 2019).

Reglas de decisión

Un árbol de decisión se puede ver a modo de gráfico en forma de árbol como ya se ha explicado, pero también es posible centrar la mirada en la información de cada una de sus ramas que puede interpretarse como una regla. Visto de esta manera, se puede decir que un árbol define un conjunto de reglas que es posible leer desde el nodo raíz hasta cada una de sus hojas. Una regla se denota como una secuencia analítica predictiva representada por el camino que se recorre desde la raíz de un árbol de decisión hasta una hoja (Kubat, 2017).

En definitiva, cualquier árbol de decisión se puede convertir en un conjunto de reglas de clasificación, es decir, una construcción del tipo Si {condición} Entonces {valor} . El algoritmo de generación de reglas consiste en que, por cada rama del árbol de decisión, las preguntas y sus

valores están situadas a la izquierda de las reglas (antecedentes) y la etiqueta del nodo hoja en la parte derecha (consecuente). En principio, este proceso genera un sistema de reglas complejo que se soluciona con la poda de esas reglas, por ejemplo C4.5 Rules, construye un conjunto de reglas de decisión a partir del árbol obtenido por C4.5 y a continuación se realiza un proceso de poda.

El conocimiento aprendido del árbol de decisión permite formar reglas (SI "antecedente" ENTONCES "consecuente") o conjuntos de reglas a partir de datos, por lo que es una técnica muy utilizada en la inferencia inductiva para la mejora de programas (Kubat, 2017). Las fases para crear reglas son las siguientes:

Se crea el árbol, se extraen las reglas de ese árbol (en base a los distintos caminos de la raíz a una hoja), se elimina aquellas reglas (o condición) que no contribuya a su precisión, se reúnen las reglas que son afines a la misma clase deduciendo una nueva regla de cada una de las clases que predice mejor, se ordenan las reglas en función de su error cometido y se examinan todas las reglas en el conjunto de entrenamiento y si alguna perjudica a la precisión se elimina. Este procedimiento permite obtener un conjunto de reglas de decisión igual de precisas que un árbol de decisión podado, con la ventaja de ser más fácil de interpretar para las personas.

Algunos de los algoritmos que forman reglas de decisión, utilizados en clasificación para la toma de decisiones en machine learning, coinciden con los ya vistos para los árboles de decisión y además existen otros, es decir, entre los algoritmos que generan estas reglas se destacan los siguientes: AQ, ID3, C4.5, J48, CART, random forest, RBS (Rabasa, 2009; Almiñana et al., 2012) y CREA (Rodríguez- Sala, 2014).

El algoritmo AQ (algorithm for quasi-optimal solutions) fue desarrollado por Michalski en 1973. Este algoritmo se considera el punto de partida del aprendizaje mediante inducción de reglas. AQ induce conjuntos de reglas Si P entonces $clase = C$. Siendo P un predicado booleano que tiene en cuenta los valores de los atributos en forma normal disyuntiva (Michalski, 1973). Aunque posteriormente fue Quinlan quien propuso un método para transformar un árbol de decisión en un conjunto de reglas de decisión.

El algoritmo RBS, fue desarrollado por Rabasa a partir del 2009 (Rabasa, 2009), es un algoritmo de ordenación de reglas de clasificación para datos discretos. RBS (Almiñana et al., 2012) es un algoritmo iterativo que considera el soporte de la regla como la probabilidad de que ocurra un antecedente de la regla, y la confianza como la probabilidad condicional de que ocurra un consecuente, (Rabasa et al., 2013) dado un antecedente específico. Cuenta con aplicaciones en medicina (Rabasa et al., 2013). Además RBS es un método de reducción de sistemas de reglas por regiones de significancia (post-poda). Asimismo, RBS ha ido evolucionando para generar reglas de clasificación (véase la sección 3.3.2).

CREA (classification rules extraction algorithm), es un algoritmo desarrollado por Rodríguez-Sala en 2014 (Rodríguez-Sala, 2014) para generación, reducción y ordenación de reglas de clasificación que en el mismo proceso se combina el método ID3 de generación de reglas de clasificación siguiendo el criterio de ganancia de información (Quinlan, 1979), y el método RBS (Rabasa, 2009) de ordenación de reglas de clasificación y reducción de sistemas de reglas por regiones de significancia (post-poda).

2.3.2. Métodos basados en similitudes

Dentro de este grupo se encuentran algoritmos como K nearest neighbor (Cover y Hart, 1967; Dasarathy, 1991) y sus extensiones a algoritmos basados en instancias como IBL (Aha et al., 1991).

K Nearest Neighbor

El algoritmo nearest neighbor (NN) o vecino más cercano fue desarrollado por T.M. Cover y P.E. Hart en 1967 como un método de clasificación de patrones. Se trata de un algoritmo de aprendizaje basado en instancias donde el entrenamiento es el último paso de la clasificación. A estas estructuras se les conoce también como algoritmos de aprendizaje perezosos (*lazy learners*). Hay un principio muy sencillo basado en instancias para describir el aprendizaje y es el siguiente: Las instancias

similares tienen etiquetas de clase similares. Para aprovechar este principio general se usa los clasificadores de vecinos más cercanos.

Se establecen los m ejemplos de entrenamiento más cercanos, para una instancia de prueba. La clase más importante tendrá la etiqueta dominante entre estos m ejemplos de entrenamiento. Lo más difícil en el uso del clasificador del vecino más cercano es la elección del parámetro m. Ya que un valor muy pequeño de m no es aconsejable porque no se obtendrán resultados robustos de clasificación debido a ruido en los datos (Aggarwal, 2015).

La elección de la métrica y la elección de un K apropiado que indica el número de vecinos que se tiene para la predicción son muy importantes. Ferrer et al. (2001) desarrollaron un algoritmo de clasificación NN no parametrizado que adapta localmente el valor K. La generalización de NN consiste en calcular los K vecinos más cercanos y la clase que sea mayoritaria entre esos vecinos será la que se asigne. Dicha generalización se llama K-NN.

Una de las ventajas del algoritmo K-NN es su buen comportamiento. Entre sus desventajas está el gran coste computacional, así como la elección de la métrica y el K más apropiados.

Los pasos seguidos para la implementación del algoritmo son:

1. Se almacena en un vector el conjunto de entrenamiento, junto a la clase asociada a cada muestra de este conjunto.
2. Se calcula la distancia euclídea de cada muestra de entrenamiento, a todas las demás que se tienen almacenadas en el vector del punto anterior y de las que se conoce la clase a la que corresponden, cogiendo solamente las K muestras más cercanas, y clasificando la nueva muestra de entrenamiento en la clase más frecuente a la que pertenecen los K vecinos obtenidos anteriormente.
3. El siguiente paso consiste en repetir el procedimiento con los datos de validación. Se hacen los cálculos para obtener el porcentaje de clasificación sobre los ejemplos de este conjunto (serán desconocidos en el aprendizaje) para conocer su generalización (García Cambronero y Gómez Moreno, 2006).

Instance-Based Learning

Los algoritmos instance-based learning (IBL) son descendientes del algoritmo vecino más cercano o nearest neighbor (Cover y Hart, 1967), que se basan en clasificar una nueva instancia considerando la clase de su vecino más cercano. La familia IBL se compone de versiones mejoradas del algoritmo NN, con el objeto de ir perfeccionándolo y restándole limitaciones.

IBL son algoritmos de aprendizaje basados en instancias que realizan predicciones de clasificación usando solo instancias específicas. Almacenar y usar instancias específicas aumenta el rendimiento de algunos algoritmos de aprendizaje supervisado. Entre ellos se destacan los árboles de decisión, reglas de clasificación y redes distribuidas. Se pueden reducir los requisitos de almacenamiento disminuyendo sensiblemente el aprendizaje y la precisión de la clasificación (Aha, 1991).

Componentes de los algoritmos IBL:

1. Función de similitud: Se encarga de calcular la similitud entre una instancia de entrenamiento i y la instancias en la descripción del concepto. Las similitudes son valores numéricos.
2. Función de clasificación: es la que clasifica la instancia i. A esta función llegan tanto los resultados de la función de similitud como los registros de desempeño de la clasificación de las instancias que describen el concepto.
3. Actualizador de la descripción de conceptos: registra el desempeño de la clasificación y toma decisiones sobre qué instancias incluir en la descripción del concepto. Sus entradas son la instancia i , los resultados de las funciones simillitud y clasificación, la descripción actual y modificada del concepto.

2.3.3. Métodos basados en técnicas de separación en espacios vectoriales

Dentro de este grupo se pueden encontrar los algoritmos de support vector machine (Cortes y Vapnik, 1995; Ben-Hur et al., 2001).

Support Vector Machine

Support vector machine (SVM) o máquinas de vectores de soporte fueron desarrolladas por Cortes y Vapnik en 1995. Son algoritmos que también se utilizan en la clasificación binaria para dividir un conjunto de datos en dos categorías. Se basan en métodos de separación en espacios vectoriales que es un conjunto de vectores, representados cada uno por una ecuación.

Los atributos son representados como vectores en un espacio vectorial con tantas dimensiones como características haya. Se calcula una línea recta o plano en función de las dimensiones, llamada hiperplano, que separa las categorías. Al sustituir cualquier punto que pertenezca a esta línea recta en la ecuación su resultado será cero. Si se trazan puntos a la derecha de la línea al sustituirlos en la ecuación el resultado es mayor que cero y con los puntos de la izquierda da menor que cero. El objetivo del algoritmo SVM es llevar a cabo la clasificación de la siguiente manera:

1. Se obtiene el hiperplano óptimo. De los diferentes hiperplanos que pueden separar el conjunto de datos hay que calcular el mejor de ellos, es decir, aquella línea que está justamente a la misma distancia de las dos categorías. Las máquinas de soporte vectorial serán las encargadas de obtener este hiperplano óptimo. Estas máquinas utilizan para la clasificación un algoritmo llamado hard magin o margen duro que consiste primeramente en obtener los puntos más cercanos entre una clase y otra (también llamados vectores de soporte), a continuación, encuentra la línea que los une y después marca una frontera perpendicular que separa esta línea en dos. La línea que se obtiene es el hiperplano óptimo.
2. Se clasifica el dato. En el caso de que los datos no estén claramente separados se obtendrán márgenes muy pequeños, ejemplo claro de overfitting. De manera que, ante un dato nuevo puede ser que no sea bien clasificado porque el margen es muy pequeño entre las clases. Por lo que el algoritmo anterior llamado hard margin no resulta demasiado útil. En este caso, las máquinas de soporte realizan la clasificación

empleando el algoritmo llamado soft margin o margen suave que es un algoritmo de máquina de vectores de soporte. Lo que hace es ensanchar el margen aceptando que haya un pequeño error al clasificar. Para que esto se produzca se incluye un parámetro C que se elige en el entrenamiento y es inversamente proporcional al ancho del margen. De esta manera, se cuenta con un clasificador más variable, pero a su vez se puede controlar la forma de obtener el margen en función de la separación de los datos de distinta categoría.

Por otro lado, en el caso de que no sea posible separar los datos empleando una línea recta para clasificarlos, lo que se hace es emplear el algoritmo de máquinas de vectores de soporte que añade alguna dimensión más a los datos utilizando una función no lineal. El método que se emplea en estas máquinas de soporte vectorial es el llamado Truco del Kernel (Vapnik et al., 1996). El reconocimiento y clasificación de imágenes urbanas aéreas (Arista-Jalife, 2017) es una de las diversas aplicaciones que tiene SVM.

Resaltar que en esta tesis se utiliza el algoritmo de optimización mínima secuencial (SMO) y es el algoritmo que usa Weka. Fue inventado por John Platt en 1998. Este algoritmo se utiliza para resolver el problema de optimización de la programación cuadrática surgido en el entrenamiento de máquinas de vectores de soporte (SVM). SMO divide este gran problema de programación cuadrática en una serie de problemas más pequeños que se resuelven analíticamente, evitando así el uso de una optimización numérica que emplea más tiempo (Platt, 1998). SMO reemplaza los valores faltantes, normaliza los atributos por defecto y transforma los atributos nominales en binarios.

2.3.4. Métodos basados en conceptos y algoritmos probabilísticos o estadísticos

Los algoritmos probabilísticos construyen un modelo que cuantifica la relación entre las variables de entrada y la variable de destino como una probabilidad (Aggarwal, 2015). Algunos de los modelos utilizados más frecuentemente son el análisis discriminante lineal (McLachlan, 2004), la

regresión logística o los algoritmos de naïve Bayes (Langley y Thompson, 1994; John y Langley, 1995) que se describen a continuación.

Análisis discriminante

El análisis discriminante, introducido por Fisher (1936), es una técnica que se utiliza para predecir la pertenencia a un grupo indicado por la variable dependiente, a partir de un conjunto de variables independientes (en el apartado 3.2.1 de esta tesis también se ha tratado el análisis discriminante). La finalidad del análisis discriminante es ver las diferencias de los grupos y predecir la clase a la que pertenece un objeto a partir de los valores que adquieren las variables independientes.

El análisis discriminante consigue a partir de las variables explicativas, unas funciones lineales de esas variables que clasifican a otros datos, siendo la función de mayor valor la que determina el grupo a que pertenecen. La información contenida en las variables independientes es empleada por el análisis discriminante para crear una función Z combinación lineal de las p variables independientes, que diferencia al máximo los dos grupos. La combinación lineal de variables independientes para el análisis discriminante también se llama función discriminante de Fisher y tiene la siguiente fórmula:

$$Z_{jk} = a + W_1 X_{1k} + W_2 X_{2k} + \cdots + W_p X_{pk}, \qquad (2.9)$$

donde:

$Z_{jk} =$ puntuación discriminante Z de la función discriminante para el objeto k.

$a =$ constante (si existe).

$W_i =$ peso discriminante para la vairable independiente i.

$X_{ik} =$ variable independiente i para el objeto k.

Regresión logística

Regresión logística simple, desarrollada por David Cox en 1958. Es una metodología de regresión que posibilita estimar la probabilidad de una variable cualitativa binaria en función de una variable cuantitativa. La

clasificación binaria es una aplicación de la regresión logística, en el que las observaciones se clasifican en uno de los dos grupos según el valor que tome la variable predictora.

Se trata de un clasificador probabilístico porque usa una suposición de modelado específica para conocer las variables características y la probabilidad de pertenencia a una clase. Los parámetros del modelo probabilístico subyacente deben estimarse en función de los datos.

En la regresión logística, la variable de clase es binaria y se extrae de $\{-1, +1\}$, también se puede trabajar con variables de clase no binarias.

"Sea $\Theta = (\theta 0, \theta 1 \ldots \theta d)$ un vector de $d + 1$ parámetros diferentes. El i-ésimo parámetro θi es un coeficiente relacionado con la i-ésima dimensión en los datos subyacentes, y $\theta 0$ es un parámetro de compensación.

Entonces, para un registro $X = (x_1 \ldots x_d)$, la probabilidad de que la variable de clase C tome los valores de $+1$ o -1, se modela con el uso de una función logística" (Aggarwal, 2015).

La regresión logística puede ser un clasificador probabilístico o clasificador lineal. Se usa un hiperplano lineal para separar las dos clases en los clasificadores lineales, como el discriminante de Fisher.

Naïve Bayes

Se trata de un algoritmo de clasificación predictivo y descriptivo basado en la teoría de la probabilidad del análisis de Bayes de 1763. Esta teoría presenta un tamaño de la muestra asintóticamente infinito e independencia estadística entre las variables independientes, refiriéndose a los atributos. Con estas condiciones, es posible calcular las distribuciones de probabilidad de cada clase con la finalidad de establecer la relación entre los atributos y la clase que son variables independientes y dependiente respectivamente (Giráldez, 2003).

Dado el ejemplo $e = (e_1, \ldots, e_m)$, siendo e_k el valor para el j-ésimo atributo. Los valores e_k siempre son discretos. La probabilidad de que ocurra la clase Ci se obtiene de la regla de Bayes,

$$P(C_i/e_1, \ldots\ldots, e_m) = \frac{P\,(C_i)\Pi_{k=1}^{m} P(e_k/C_i)}{P(e_1,\ldots,e_m)} \quad (2.10)$$

donde $P(e_k/C_i)$ se calcula de la proporción de ejemplos con valor e_k cuya clase es C_i y $P(C_i)$ es la proporción de la clase C_i.

En conclusión, en la tabla 2.1 se detalla un resumen sobre los métodos de clasificación más utilizados para la toma de decisiones en machine learning, vistos en esta sección.

TABLA 2.1. Algoritmos de clasificación por categorías.

Categoría	Algoritmo
Inducción de árbol de decisión (Árboles y reglas de decisión)	ID3 C4.5 J48 CART Random Forest
Similitudes	K Nearest Neighbors (KNN) Instance-Based Learning (IBL)
Espacios vectoriales	Support Vector Machines (SVM)
Conceptos y algoritmos probabilísticos o estadísticos	Análisis Discriminante Regresión Logística Naïve Bayes

2.4. Medidas de desempeño

Para evaluar el desempeño o rendimiento de un modelo generado por un sistema de aprendizaje automático se necesita conocer su precisión en la clasificación. Entre las medidas de desempeño existentes se destacan las siguientes: matriz de confusión, tasa de error, accuracy y curva ROC que se presentan a continuación.

2.4.1. Matriz de confusión

La matriz de confusión (Kohavi y Provost, 1998) es una herramienta que permite observar fácilmente el desempeño de un algoritmo de aprendizaje supervisado. Dicha matriz ordena en categorías (falso positivo, falso negativo, verdadero positivo y verdadero negativo) todos los casos precisando si el valor real coincide con el valor de predicción. Después se cuentan todos los casos de cada categoría mostrando en la

matriz los totales de cada una de esas categorías. Las filas de la matriz son los valores que se predicen y las columnas son los valores reales.

Esta matriz de confusión (o clasificación) muestra en una tabla los errores y los aciertos del clasificador. Lo que permite evaluar los resultados de la predicción.

TABLA 2.2: Matriz de confusión

		Clase verdadera		
		C1	C2	
Clase predicha	C1	a	b	Pc_1
	C2	c	d	Pc_2
		Pr_1	Pr_2	

Tipos de aciertos:

a = número de clasificaciones correctas en la clase C_1 = verdadero positivo (clases que corresponde predecir a través del algoritmo y fueron predichas).

d = número de clasificaciones correctas en la clase C_2 = verdadero negativo (se trata de las clases que no obtienen puntuación alta del algoritmo ni pertenecen al subconjunto de pruebas).

Tipos de errores:

b = número de clasificaciones incorrectas. Era C_2, sin embargo, se clasifica C1 = Error de tipo II o falso negativo (los enlaces que no tienen puntuación alta por parte del algoritmo y son parte del subconjunto de pruebas).

c = número de clasificaciones incorrectas. Era C_1, sin embargo, se clasifica C2 = Error de tipo I o falso positivo (clases que han sido predichas, pero no corresponde predecir. No corresponden al subconjunto de pruebas).

Proporciones:

Pc_1= proporción de casos que el clasificador asigna a la clase C_1.

Siendo $C_1 = a / (a + b)$.

$\mathrm{P}c_2$= proporción de casos que el clasificador asigna a la clase C_2.

Siendo $C_2 = \mathrm{d} / (\mathrm{c} + \mathrm{d})$.

Probabilidad:

$\mathrm{P}r_1$= probabilidad de la clase C_1.

$\mathrm{P}r_2$= probabilidad de la clase C_2.

Para generar una matriz de confusión hay que usar un conjunto de datos para el que ya se conoce los valores del atributo de resultados. Usualmente, se usan los datos de prueba que se guardaron precisamente para este momento.

2.4.2. Tasa de error

Una medida de precisión es la tasa de error E. Es la frecuencia de errores de un clasificador sobre el conjunto de datos. Se calcula dividiendo el número de errores de un clasificador entre el número total de casos. Por lo que el conjunto de datos se divide, a través, de unos métodos en un subconjunto de datos que se utiliza para el aprendizaje y el otro subconjunto para la evaluación. Algunos de los métodos que se emplean para separar los datos son la validación cruzada (Stone, 1974), split simple, etc.

La tasa de error empírica del modelo se mide sobre los datos de evaluación con la fórmula siguiente:

$$E = \frac{Nº\ de\ errores}{Nº\ total\ de\ casos}. \qquad (2.11)$$

O lo que es lo mismo:

NTP es el número de verdaderos positivos

NTN es el número de verdaderos negativos

NFP es el número de falsos positivos

NFN es el número de falsos negativos

El número de errores es la suma de $\mathrm{NFP} + \mathrm{NTP}$. Además, T es el número total de casos, es decir,

$$\mathrm{T} = \mathrm{NFP} + \mathrm{NFN} + \mathrm{NTP} + \mathrm{NTN}.$$

De lo anterior se deduce la tasa de error E, es decir la fórmula

$$E = \frac{NFP+NFN}{NFP+NFN+NTP+NTN}. \tag{2.12}$$

2.4.3. Accuracy

Otra medida de precisión es accuracy (Acc) que evalúa el desempeño en el aprendizaje automático. Es la frecuencia de clasificaciones correctas del clasificador sobre el conjunto de datos. Se calcula dividiendo el número de clasificaciones correctas, $\mathrm{NTP} + \mathrm{NTN}$, por el número total de casos. Devuelve el porcentaje de aciertos de la predicción resultante (Kubat, 2017) por lo que

$$Acc = \frac{NTP+NTN}{NFP+\ NFN+NTP+NTN}. \tag{2.13}$$

Hay que considerar que accuracy también puede venir expresada en función de la tasa de error, dándose la siguiente expresión:

$$\mathrm{Acc} = 1 - \mathrm{E}. \tag{2.14}$$

2.4.4. Curva ROC

La curva ROC (receiver operating characteristic) es una representación gráfica (figura 2.3) que se usa frecuentemente en clasificación binaria para expresar resultados especialmente cuando la distribución de clases es altamente desbalanceada, utilizándose cada vez más en la evaluación de la predicción. Puede utilizarse para generar estadísticas que resumen el rendimiento de un clasificador. En este gráfico se puede visualizar la distribución de las fracciones de falsos positivos y verdaderos positivos. La fracción de verdaderos positivos se denomina sensibilidad, supone la probabilidad de clasificar correctamente a un individuo cuyo estado real sea positivo. La especificidad es la probabilidad de clasificar correctamente a un individuo cuyo estado real sea clasificado como negativo. Esto significa restar uno de la fracción de falsos positivos.

La curva ROC se construye en base a la unión de distintos puntos de corte, correspondiendo el eje Y a la sensibilidad y el eje X a $\{1 - especificidad\}$ de cada uno de ellos. Ambos ejes incluyen valores entre 0 y 1 (0% a 100%). En los gráficos de curva ROC se traza una línea desde el punto 0,0 al punto 1,1, llamada diagonal de referencia o línea de no-discriminación (Cerda y Cifuentes, 2012).

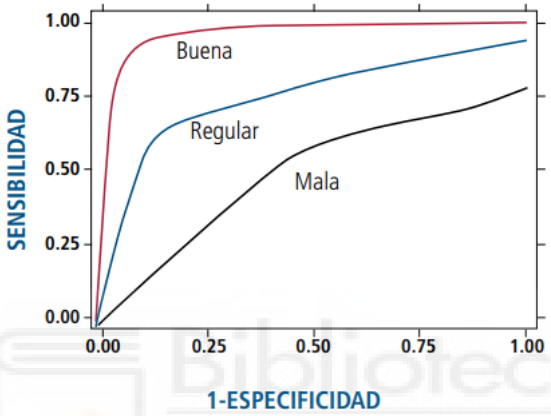

Figura 2.3: Distintas posibilidades de Curvas Roc (Manterola, 2009).

2.5. Técnicas de reducción de la dimensión

Muy a menudo es conveniente disminuir el volumen de datos para evitar información irrelevante, por lo que han proliferado técnicas que tienen esa función.

Para reducir la dimensión de los datos se utiliza una de estas dos alternativas:

- Reducir la dimensión transformando los datos.
- Seleccionar un subconjunto de características manteniendo la mayor parte de la información en el conjunto de datos, este enfoque se conoce como selección de características.

2.5.1. Técnicas estadísticas de reducción de la dimensión

Entre las técnicas de reducción de la dimensión de los datos se pretende resaltar el análisis discriminante que es un método de clasificación y

también de reducción de dimensión, esto último es debido a la combinación lineal de variables independientes (el análisis discriminante ya se ha introducido en el apartado 2.3.4 y se verá un poco más en detalle en el apartado 3.2.1).

2.5.2. Selección de características o atributos

La selección de características es la primera etapa del proceso de clasificación. El objetivo de los algoritmos de selección de características es seleccionar los atributos más informativos (eliminar rasgos redundantes) con respecto a la etiqueta de la clase.

Para evaluar la calidad del subconjunto seleccionado los métodos de selección de características buscan sobre el espacio de características, aplicando una función criterio.

En la clasificación, de acuerdo con la estrategia de búsqueda, se utilizan tres tipos principales de métodos para la selección de características: filter (Liu y Setiono, 1996), wrapper (Kohavi y John, 1997) y embedded (Chandrashekar y Sahin, 2014; Miao y Niu, 2016):

1. Modelos de filtro (filter): Tienen un criterio matemático muy preciso para evaluar la calidad de una característica o un subconjunto de características. Este criterio se utiliza para filtrar las características irrelevantes. A este modelo pertenecen algunos métodos de selección de características, como el análisis discriminante lineal, que crea una combinación lineal de las características originales como un nuevo conjunto de características. La función criterio es independiente del algoritmo de aprendizaje. Los modelos de filtro son los más utilizados en la práctica porque son muy rápidos, ya que para evaluar la calidad de los subconjuntos de características previamente seleccionados no necesitan hacer llamadas al algoritmo de aprendizaje.
2. Modelos envolventes (wrapper): Utilizan un algoritmo de clasificación para evaluar el rendimiento del algoritmo que ha seleccionado un subconjunto de características. Es decir, en este modelo el mismo algoritmo de aprendizaje que

después se empleará para la clasificación es el que genera un conjunto de reglas que utilizará la función criterio para evaluar la calidad del subconjunto de características seleccionado. En el modelo envolvente la función criterio no es independiente del algoritmo de aprendizaje (Giráldez, 2003).

3. Modelos integrados (embedded): Algunos algoritmos de aprendizaje, tienen inmerso como una parte no separable la selección de características. Por lo que esta se diseña de forma exclusiva para este aprendizaje, de manera que su rendimiento debe ser mayor. Se encuentran incluidos en estos modelos los clasificadores de máquinas de vectores de soporte (SVM), los métodos de regresión logística y las redes neuronales (Aggarwal, 2015).

Capítulo 3. Una experiencia computacional para la selección automática de características en entornos Big Data

Este capítulo de la tesis doctoral está basado en el artículo *Y. Orenes, A. Rabasa, A. Pérez-Martín, J.J. Rodríguez-Sala, J. Sanchez-Soriano. A computational experience for automatic feature selection on Big Data frameworks. International Journal of Design & Nature and Ecodynamics vol. 11:168-177, 2016.*

El sistema de reglas de clasificación es una de las técnicas analíticas predictivas utilizadas en los problemas de Big Data, donde es habitual encontrar conjuntos de datos con millones de filas, pero también con decenas de variables (atributos). Los sistemas de reglas de clasificación consisten en conjuntos de reglas que tienen un llamado antecedente (variable o conjunto de variables que pueden ser numéricas o nominales) y un consecuente (variable objetivo, siempre nominal). Si las variables antecedentes son numéricas, muchos algoritmos generadores de reglas de clasificación emplean métodos tradicionales de selección automática de características, basados en técnicas ya establecidas en el ámbito científico, como el análisis discriminante o el análisis de clústeres. En este capítulo, se propone la comparación del método de selección y clasificación de características, basado en la métrica Waci que aporta el método RBS (originalmente diseñado para manejar solo variables nominales) y los métodos clásicos de selección de características. Tras la definición formal del método RBS, se presenta el diseño de una experiencia informática que permite comparar cualitativa y cuantitativamente el RBS-CREA (o RBS adaptado) y otros métodos de selección de características. Por último, se discuten las condiciones óptimas de aplicación de cada método y se

identifican futuras áreas de investigación en el campo de la selección automática de características.

3.1. Introducción

Los sistemas decisionales integran cada vez más fuentes de datos, y estas son grandes y heterogéneas en el tiempo. Por ello, los modelos predictivos basados en reglas de clasificación, por ejemplo, del tipo ID3 (Quinlan, 1986), necesitan incorporar mecanismos que permitan elegir las variables más incidentes en la variable objetivo que se pretende predecir. Así, los sistemas de selección de características también están evolucionando y adaptándose a estos cambios que están relacionados principalmente con problemas de alta dimensionalidad.

En este capítulo se propone un método para la selección automática de características en marcos de Big Data. El método propuesto evoluciona a partir de un método de generación y gestión de reglas de clasificación que fue inicialmente diseñado para gestionar únicamente variables nominales.

El problema de la selección de características en problemas de Big Data y el enfoque general se exponen en la sección 3.2 donde se introduce una técnica clásica como es el análisis discriminante para comparar posteriormente.

El proceso de simulación para la generación de conjuntos de datos sintéticos se presenta en la sección 3.3. Como se espera que el método de selección de características se pruebe en varios escenarios de sobrecarga, se generarán varios conjuntos de datos. Además, se muestra una visión general del experimento, en la que se explica detalladamente cómo se procesarán los conjuntos de datos con diferentes técnicas de selección de características y cómo se utilizarán los diferentes conjuntos de atributos generados para la generación del correspondiente conjunto de reglas. Los conjuntos de reglas finales (procedentes de los diferentes conjuntos de atributos reducidos) se compararán bajo varios criterios que también se presentan.

Ese experimento computacional se muestra en la sección 3.4, donde se presenta una comparación empírica de la precisión de los sistemas de

reglas. Por último, en la sección 3.5, se presentan y discuten las conclusiones sobre el potencial del método y las condiciones en las que parece ser más apropiado. Además, se señalan las futuras líneas de investigación relativas a las metodologías de selección de características.

3.2. Definición del problema y objetivo principal

En esta sección se presenta el análisis discriminante con una breve descripción formal, incluyendo sus principales objetivos y las restricciones en el manejo de las variables. Esta técnica se utiliza habitualmente en los métodos de selección de características (Lê Cao et al., 2011) (especialmente con variables numéricas). Esta técnica estadística se incluye en este estudio para establecer un marco comparativo para medir el método de selección de atributos que ofrece la métrica Waci de RBS.

A continuación, se introduce el método de generación de reglas, RBS , para que en el siguiente apartado se adapte para poder gestionar con variables numéricas.

3.2.1. Análisis discriminante

El análisis discriminante es una técnica estadística multivariante para analizar si existen diferencias entre grupos de objetos (categorías) sobre un conjunto de variables (independientes) medidas sobre ellos. Así, el mecanismo permite reducir el número de variables independientes y clasificar un futuro ítem cuyos valores de las variables se conocen, pero se desconoce el grupo al que pertenece el ítem (Fisher, 1936).

Una combinación lineal de variables independientes para el análisis discriminante, también llamada función discriminante, tiene la siguiente forma:

$$Z_{jk} = a + W_1 X_{1k} + W_2 X_{2k} + \cdots + W_p X_{pk}, \quad (3.1)$$

donde:

$Z_{jk} =$ puntuación discriminante Z de la función discriminante para el objeto k.

$a =$ constante (si existe).

$W_i =$ peso discriminante para la variable independiente i.

$X_{ik} =$ variable independiente i para el objeto k.

Tipos de variables en el análisis discriminante:

- Variables dependientes: una variable cualitativa (nominal) con tantos valores discretos como grupos.
- Variables independientes (variables de clasificación o discriminantes): variables con algún tipo de relación con los grupos de la variable dependiente. Estas variables pueden ser numéricas o nominales.

Objetivos del análisis discriminante:

- Determinar las reglas de asignación de los individuos de las poblaciones sobre una clasificación conocida.
- Encontrar las diferencias entre los grupos respecto a las variables consideradas.
- Clasificación de los individuos de origen desconocido en uno de los grupos.
- Determinar las mejores combinaciones lineales (funciones discriminantes) de las variables independientes para diferenciar los grupos y así clasificar los nuevos casos.

Etapas del análisis discriminante:

- Definición del problema
- Selección de variables independientes y dependientes
- Selección del tamaño de la muestra
- Comprobación de la hipótesis
- Modelo de estimación
- Validación de las funciones discriminantes
- Contribución de la variable al poder discriminatorio
- Evaluación de la función de predicción

3.2.2. Reglas de clasificación y selección de características mediante RBS

El RBS es un algoritmo de ordenación de reglas de clasificación para datos discretos, que incorpora un conjunto de mejoras respecto a otros

algoritmos del mismo tipo, haciéndolo considerablemente más rápido. RBS (Almiñana et al., 2012) es un algoritmo iterativo que considera el soporte de la regla como la probabilidad de que ocurra un antecedente de la regla, y la confianza como la probabilidad condicional de que ocurra un consecuente, (Rabasa et al., 2013) dado un antecedente específico. Así, para una regla $r = [A \rightarrow Q]$, el soporte y la confianza pueden expresarse como sigue:

$$Soporte(r = [A \rightarrow Q]) = \frac{N_A}{N}, \quad Confianza(r = [A \rightarrow Q]) = \frac{N_{A \rightarrow Q}}{N_A} \tag{3.2}$$

donde N es el número total de registros en el conjunto de datos, N_A es el número de tuplas en las que aparece el antecedente de la regla, y $N_{A \rightarrow Q}$ es el número de tuplas en las que aparecen el antecedente y el consecuente de la regla. La figura 3.1 representa el soporte y la confianza de un hipotético conjunto de reglas. Así, se definen varias fronteras para dividir el plano en cuatro regiones, denominadas REG-1, REG-2, REG-3 y REG-0. RBS encuentra las fronteras mínimas y máximas del soporte y la confianza. La figura 3.1 muestra un ejemplo de posicionamiento de fronteras y distribución de reglas.

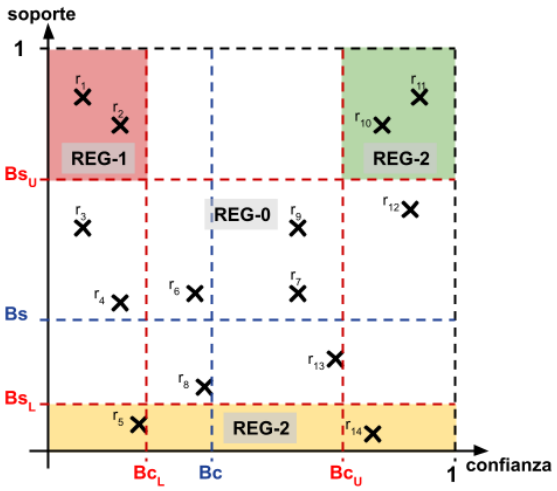

Figura 3.1: Regiones RBS

Donde,

- Bs: Límite de soporte que se calcula a partir del número de antecedentes posibles en un conjunto de reglas dado.
- Bs_U: Límite superior de soporte, correspondiente a la media de soporte de las reglas sobre Bs.
- Bs_L: Límite inferior de soporte, correspondiente a la media de soporte de las reglas bajo Bs.
- Bc: Límite de confianza que se calcula a partir del número de posibles consecuentes en un determinado conjunto de reglas.
- Bc_U: Límite superior de confianza, correspondiente a la media de confianza de las reglas sobre Bc.
- Bc_L: Límite inferior de confianza, correspondiente a la media de confianza de las reglas bajo Bc.

Además, RBS define y calcula el $waci$ (índice de correlación de atributos ponderado), una medida de la importancia del grupo de atributos seleccionado para cada sistema de conjuntos de reglas Rr.

$$waci(Rr) = \frac{w_1|REG-1|+w_2|REG-2|+w_3|REG-3|}{|R|(w_1 a(REG-1)+w_2 a(REG-2)+w_3 a(REG-3))}, \quad (3.3)$$

donde $|REG - i|$ representa el total de reglas en la región i, w_i es el peso asignado (parámetro) para la región i, y $a(REG - i)$ es la superficie de la región i.

Así, $waci$ puede considerarse como una medida de correlación de variables para modelar una determinada variable de clase. Así, RBS proporciona tanto una lista de la combinación de variables más significativas (para clasificar la variable objetivo) como los conjuntos de reglas de clasificación para cada una de esas combinaciones.

Aunque RBS se ha aplicado con éxito a varios problemas de clasificación, por ejemplo, en Medicina (Rabasa et al., 2013) tiene una restricción muy importante: solo maneja variables nominales.

3.3. Experimento computacional

3.3.1. Definición de los conjuntos de datos y proceso de generación de conjuntos de datos semisintéticos

Para comparar los métodos de análisis discriminante y RBS, se ha simulado un conjunto de ocho conjuntos de datos sintéticos. Estos conjuntos de datos son similares a los datos reales de gestión de reservas hoteleras en la costa este de España. Cada conjunto de datos tiene las siguientes ocho variables:

Variables antecedentes:

- Season: off season, Christmas, Easter y summer.
- Advance in reserve: número de días antes de la reserva.
- Advance in reserve (discreta): <7, 8-15, >15 (días).
- Accommodation days: número de días en el hotel.
- Accommodation days (discreta): <2, 3-7, 8-15, >15 (días).
- Country: Spain (SP), Germany (GE) y United Kingdom (UK).
- Room type: individual (ind), double (dob), triple (tri) y suite (sui).

Variable consecuente:

- Daily spending (discreta): <70, 70-100, 101-140, >140 (euros).

Algunas características de la simulación:

- Casi el 60% de las habitaciones están reservadas en verano.
- La temporada de menos reservas es la de Navidad.
- Alrededor del 50% de los alojamientos se reservan con más de 15 días de antelación.
- Alrededor del 55% de las habitaciones están reservadas para más de 15 días de alojamiento y están en verano.
- Los países están distribuidos uniformemente.
- Las habitaciones individuales se reservan menos que el resto.
- Se introducen algunos ruidos aleatorios para obtener diferentes conjuntos de datos.

Se han simulado conjuntos de datos de diferentes tamaños para realizar pruebas de cálculo. Se han generado conjuntos de 1,000, 10,000, 100,000 y 1,000,000 de registros (tuplas) con las características mencionadas.

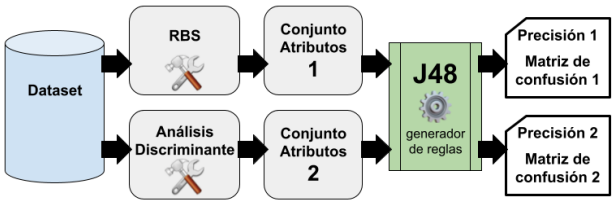

Figura 3.2: Método de investigación

3.3.2. Resumen del experimento

Cada uno de los conjuntos de datos se procesa con el método de análisis discriminante y el método RBS, proporcionando sus correspondientes conjuntos de características seleccionadas que se utilizan para la generación de dos conjuntos de reglas de clasificación diferentes. Cada uno de estos conjuntos de reglas alcanza unos ratios de precisión específicos que se comparan a través de sus correspondientes matrices de confusión proporcionadas por el árbol de clasificación J48. J48 es una implementación en Java (en el software Weka (Weka, 2020, 2021)) del algoritmo original de árbol de clasificación C4.5 que es una versión actualizada del clásico ID3 (Quinlan, 1986). C4.5 (también su correspondiente implementación J48) maneja tanto antecedentes nominativos (categóricos) como numéricos.

El análisis discriminante se ha implementado utilizando el software estadístico R versión 3.2.3 (Team R) con el paquete MASS (Venables y Ripley, 2002), mientras que el RBS se ha implementado utilizando lenguaje C. Ambos se han desarrollado en un servidor virtual con sistema operativo Linux Centos v6 (64 bits), 2 CPUs para 2.8 GHz y 2 GBytes de RAM.

3.4. Comparación empírica

3.4.1. Comparación cuantitativa

Al calcular los conjuntos de datos de 100,000 registros, tanto el análisis discriminante de R como la selección de características de RBS tardan entre 1 y 2 segundos. Por lo tanto, no hay diferencias sustanciales en el tiempo de cálculo sobre conjuntos de datos de 100,000 registros. Tampoco se aprecian diferencias sustanciales entre las precisiones de estos métodos a lo largo de 100,000 tuplas de archivos (véase tabla 3.1).

Al calcular el conjunto de datos de 1,000,000 de registros, el análisis discriminante de R tarda una media de 8.25 segundos, mientras que la selección de características de RBS tarda una media de 12.47 segundos.

Sin embargo, aunque los tiempos del método de análisis discriminante realizado con R son ligeramente mejores que los obtenidos por la selección de características de RBS, el primero solo proporciona la combinación óptima de variables, mientras que el segundo proporciona las 20 mejores combinaciones y también sus correspondientes conjuntos de reglas de clasificación.

TABLA 3.1. Precisión de la clasificación (%) para el análisis discriminante y el método RBS. Comparación empírica con conjuntos de datos de 100,000 registros.

Conjuntos de datos	1	2	3	4	5	6	7	8
Análisis discriminante	84.09	84.21	84.05	84.37	84.24	84.13	83.97	83.81
Método RBS	84.18	84.03	83.86	84.53	84.33	83.85	84.02	83.78

A continuación, las tablas 3.2 y 3.3 contienen el tiempo requerido por ambos métodos para calcular los 8 conjuntos de datos de 100,000 registros y 1,000,000, respectivamente.

TABLA 3.2. Tiempo de cálculo (segundos) para los 8 conjuntos de datos de 100,000 registros.

Conjunto de datos	1	2	3	4	5	6	7	8
Análisis discriminante	0.5	0.6	0.7	0.6	0.7	0.5	0.6	0.5
Método RBS	1.5	1.4	2.0	1.5	1.6	1.5	1.7	1.4

TABLA 3.3. Tiempo de cálculo (segundos) para los 8 conjuntos de datos de 1,000,000 de registros.

Conjunto de datos	1	2	3	4	5	6	7	8
Análisis discriminante	7.2	7.4	7.9	8.1	7.5	7.4	10.3	10.2
Método RBS	12.2	12.1	12.2	12.1	12.6	12.1	13.3	13.1

3.4.2. Comparación cualitativa

Como se ha descrito anteriormente, las comparaciones de ambos métodos (análisis discriminante y RBS) en términos de tiempo de cálculo y precisión de los sistemas de reglas generados, conducen a resultados muy similares con conjuntos de datos de 1,000; 10,000; incluso 100,000 tuplas (tabla 3.1). Por lo tanto, para llevar a cabo una comparación cualitativa, es necesario aumentar el tamaño de los conjuntos de datos hasta 1,000,000 de tuplas (donde el tiempo de cálculo empieza a ser diferente en cada método). Así, centrando la atención en archivos de 1,000,000 de registros, las variables más significativas proporcionadas por ambos métodos son esencialmente las mismas.

En el experimento de análisis discriminante, todas las variables se incluyen en el modelo discriminante. Este análisis proporciona más o menos la misma conclusión sobre las funciones discriminantes. Cerca del 88% de la varianza entre grupos se explica mediante la primera función discriminante.

RBS proporciona 8 tablas (una por cada conjunto de datos) como la que se muestra en la tabla 3.4, donde la primera columna contiene el número de la combinación de variables (de 1 a 20) ordenada por Waci (véase la expresión (3.3)). Además, entre paréntesis, se muestra la cantidad de variables que forman dicha combinación. La última columna muestra el Waci para cada combinación de variables.

TABLA 3.4. Mejores 20 combinaciones de variables para el conjunto de datos con 100.000 registros.

#(nº)	season	advan.reserve.D	accommod.days.D	country	Room	Waci
#1 (1)	–	–	accommod.days.D	–		0.510
#2 (1)	season	–	–	–		0.502
#3 (2)	season	–	accommod.days.D	–		0.502
#4 (1)	–	advan.reserve.D	–	–		0.388
#5 (1)	–	–	–	country		0.209
#6 (1)	–	–	–	–	–	0.183
#7 (2)	season	advan.reserve.D	–	–	–	0.175
#8 (2)	–	–	accommod.days.D	country	–	0.170
#9 (2)	season	–	–	country	–	0.167
#10 (2)	–	advan.reserve.D	accommod.days.D	–	–	0.148
#11 (2)	–	advan.reserve.D	–	country	room	0.129
#12 (2)	season	–	–	–	–	0.125
#13 (3)	season	advan.reserve.D	accommod.days.D	–	–	0.125
#14 (3)	–	advan.reserve.D	accommod.days.D	country	–	0.107
#15 (3)	season	advan.reserve.D	–	–	–	0.101
#16 (3)	season	–	accommod.days.D	country	–	0.099
#17 (2)	–	–	–	country	room	0.096
#18 (3)	season	advan.reserve.D	–	country	–	0.092
#19 (2)	–	–	accommod.days.D	–	–	0.091
#20 (3)	season	–	accommod.days.D	–	room	0.084

Por lo tanto, la mejor variable para predecir la variable objetivo es $\{accommod.days.D\}$. Si hay que considerar dos variables, la mejor combinación es $\{season, accommod.days.D\}$ con un índice de correlación muy similar. La mejor combinación de tres variables es la formada por $\{accommod.days.D, season, advan.reserve.D\}$.

De forma análoga, para cada conjunto de datos (desde el conjunto de datos 2 hasta el 8) se calculan las mejores combinaciones de variables, proporcionando resultados muy similares, con las siguientes excepciones: Para el conjunto de datos 5, la mejor cuya combinación de tres variables incluye la variable $country$, en lugar de $advan.reserve.D$ $\{accommod.days.D, season, country\}$. También, para el conjunto de datos 6, cuya combinación de tres variables incluye $room$, en lugar de $accommod.days.D$ $\{room, season, advan.reserve.D\}$.

Dada la gran similitud entre los resultados obtenidos y con el fin de simplificar la interpretación de los resultados, se ha elegido un archivo particular de 1,000,000 de registros Dataset 2. Para este conjunto de datos, las variables más significativas proporcionadas por cada método son las siguientes.

(i) Árbol de clasificación J48

En este método, se utiliza el conjunto completo de variables $\{ season, advan.reserve.D, accommod.days.D, country, room \}$ y mediante el uso de Weka (Weka, 2020, 2021) para la generación de un árbol de clasificación J48, el modelo final se caracteriza como sigue: Instancias clasificadas correctamente: 826,127 (82.6127%)

TABLA 3.5. Matriz de confusión del método J48 utilizando todas las variables originales del conjunto de datos.

Clasificado como →	"< 70"	"70 – 100"	"101 – 140"	"> 140"
"< 70"	75,346	5,511	0	428
"70 – 100"	40,320	27,430	0	2,103
"101 – 140"	20,941	144	171,362	29,320
"> 140"	5,582	0	69,524	551,989

Así, por ejemplo, la primera fila de la tabla 3.5 significa que 75,346 instancias del total de reglas que deben clasificarse como "<70", se clasificaron correctamente, mientras que 5,511 de ellas se clasificaron incorrectamente como "70-100"; ninguna se clasificó incorrectamente como "101-140" y 428 se clasificaron incorrectamente como ">140".

(ii) Análisis discriminante

Este método no elimina ninguna variable y proporciona el conjunto completo de variables {season, advan.reserve.D, accommod.days.D, country, room} con el peso correspondiente calculado para este modelo. Con el modelo para cada conjunto de datos, este procedimiento puede clasificar las instancias. A continuación, la tabla 3.6 muestra la matriz de

confusión después de aplicar el análisis discriminante para la selección de características sobre el conjunto de datos. Instancias correctamente clasificadas: 841,756 (84.1756%)

TABLA 3.6. Matriz de confusión del método de análisis discriminante para el conjunto de datos.

Clasificado como →	"< 70"	"70 – 100"	"101 – 140"	"> 140"
"< 70"	61,793	3,584	13,913	1,995
"70 – 100"	27,058	17,798	17,851	7,146
"101 – 140"	28,162	252,000	162,546	30,807
"> 140"	6,531	279,000	20,666	599,619

Así, por ejemplo, la primera fila de la tabla 3.6 significa que del total de instancias que deben clasificarse como "<70", 61,793 instancias se clasificaron correctamente, mientras que 3,584 de ellas se clasificaron incorrectamente como "70-100"; 13,913 se clasificaron incorrectamente como "101-140" y 1,995 se clasificaron incorrectamente como ">140".

(iii) Selección de características basada en la métrica Waci de RBS (RBS adaptado)

La mejor combinación de dos variables consiste en $\{accommod.days.D, season\}$ porque, con un índice de correlación muy similar, considerar dos variables antecedentes es mejor que considerar solo una. Esta es la combinación seleccionada para generar el modelo de clasificación y medir su precisión. Utilizando Weka para la generación de un árbol de clasificación J48, el modelo final se caracteriza como sigue: Instancias clasificadas correctamente: 863,027 (86.3027%).

TABLA 3.7. Matriz de confusión del método RBS de selección de características para el conjunto de datos.

Clasificado como →	"< 70"	"70 – 100"	"101 – 140"	"> 140"
"< 70"	25,699	5,939	48,754	893
"70 – 100"	0	29,533	38,559	1,761
"101 – 140"	0	155	190,591	31,021
"> 140"	0	0	9,891	617,204

De forma análoga a las tablas 3.5 y 3.6, en la tabla 3.7 la primera fila significa que del total de reglas que deben ser clasificadas como "<70", 25,699 instancias fueron clasificadas correctamente, mientras que 5,939 de ellas fueron clasificadas incorrectamente como "70-100"; 48,754 fueron clasificadas incorrectamente como "101-140" y 893 fueron clasificadas incorrectamente como ">140".

A continuación, la tabla 3.8 resume la precisión alcanzada con el análisis discriminante y la selección de características de RBS adaptado para los 8 conjuntos de datos de 100,000 y 1,000,000 de registros, respectivamente.

TABLA 3.8. Precisión de la clasificación (%) para el análisis discriminante y el método RBS. Comparación empírica con conjuntos de datos de 1.000.000 de tuplas.

Dataset	1	2	3	4	5	6	7	8
Análisis discriminante	84.18	84.16	84.22	84.14	84.20	84.18	84.18	84.28
Método RBS	86.30	85.71	86.89	85.89	86.77	85.87	86.66	86.22

3.5. Conclusiones y futuras líneas de investigación, a partir de RBS

El RBS adaptado proporciona un buen tiempo de cálculo (aunque ligeramente superior al proporcionado por el análisis discriminante) para la selección automática de características.

Además este método, con respecto al RBS original (Rabasa, 2009), cuenta con la ventaja de generar conjuntos de reglas de clasificación.

Se trata de una ventaja cualitativa muy importante respecto a los métodos que solo reducen el conjunto de atributos, pero no proporcionan sus correspondientes conjuntos de reglas de clasificación.

Las precisiones alcanzadas en la clasificación utilizando las variables proporcionadas por el RBS son mejores que las proporcionadas por el análisis discriminante.

En futuras investigaciones, se aplicará la selección de características a nuevos problemas, con muchos más atributos, y se comparará la precisión conseguida con la alcanzada al aplicar la reducción de dimensión con métodos estadísticos clásicos. Tal vez, un análisis de componentes

principales basado en el análisis factorial pueda reducir la dimensión del conjunto de datos para volver a intentarlo con el análisis discriminante. Esto se notará especialmente en presencia de un mayor número de variables.

El RBS se muestra como un método muy preciso de selección de características; sin embargo, será necesario incorporar un método de discretización de variables numéricas que permita el tratamiento de este tipo de variables desde el propio algoritmo, en lugar de como un paso en la etapa de preprocesamiento. También es necesario reducir el tiempo de ejecución cuando esta técnica se aplique en conjuntos de datos extremadamente grandes.

Capítulo 4. Análisis comparativo de la precisión de los métodos de clasificación relacionados con la entropía

Este capítulo de la tesis doctoral está basado en el artículo *Y. Orenes, A. Rabasa, J.J. Rodríguez-Sala, J. Sánchez-Soriano. Benchmarking Analysis of the Accuracy of Classification Methods Related to Entropy. Entropy vol. 23(7):850, 2021.*

En la literatura de aprendizaje automático se pueden encontrar numerosos métodos para resolver problemas de clasificación. Se proponen tres nuevas medidas de rendimiento para analizar dichos métodos. Estas medidas se definen utilizando el concepto de reducción proporcional del error de clasificación con respecto a tres clasificadores de referencia, el aleatorio y dos clasificadores intuitivos que se basan en cómo una persona no experta podría realizar la clasificación simplemente aplicando un enfoque frecuentista. Se demuestra que estos tres métodos simples están estrechamente relacionados con diferentes aspectos de la entropía del conjunto de datos. Por lo tanto, estas medidas tienen en cuenta la entropía del conjunto de datos a la hora de evaluar el rendimiento de los clasificadores. Esto permite medir la mejora de los resultados de clasificación en comparación con los métodos simples y, al mismo tiempo, cómo afecta la entropía a la capacidad de clasificación. Para ilustrar cómo se pueden utilizar estas nuevas medidas de rendimiento para analizar los clasificadores teniendo en cuenta la entropía del conjunto de datos, se realiza un experimento intensivo en el que se utiliza el conocido algoritmo J48, y un conjunto de datos del repositorio UCI sobre el que se ha seleccionado previamente un subconjunto de los atributos más relevantes.

A continuación, se realiza un experimento extensivo en el que se consideran cuatro clasificadores heurísticos, y 11 conjuntos de datos.

4.1. Introducción

Como se ha expuesto en el capítulo 1, la clasificación es uno de los temas más relevantes del aprendizaje automático (Aggarwal, 2015; Kelleher et al., 2015; Kubat, 2017; Skiena, 2017). En general, el objetivo de la clasificación supervisada es predecir la clase correcta, entre un conjunto de clases conocidas, de una nueva observación dada, basándose en el conocimiento proporcionado por un conjunto de datos, conocido como "datos de entrenamiento". Además, el problema de la clasificación es muy importante en la toma de decisiones en muchos campos diferentes, por lo que no es difícil encontrar aplicaciones en campos como la medicina, la biotecnología, el marketing, la seguridad en las redes de comunicación, la robótica, el reconocimiento de imágenes y textos, etc. Tres cuestiones en los problemas de clasificación son la selección de subconjuntos de atributos, el diseño e implementación de clasificadores, y la evaluación del rendimiento de los clasificadores (Aggarwal, 2015; Kelleher et al., 2015; Kubat, 2017; Skiena, 2017). En este capítulo, nos centraremos principalmente en este último aspecto.

Por otro lado, la entropía aparece en la estadística o en la teoría de la información como una medida de diversidad, incertidumbre, aleatoriedad o incluso complejidad. Por ello, se puede encontrar el uso de la entropía en el problema de la selección de características y el diseño de clasificadores. Shannon (1948) introdujo la entropía en el contexto de la teoría de la comunicación y la información. Este concepto se ha utilizado con frecuencia en los modelos de aprendizaje basados en la información (Kelleher et al., 2015). Dos extensiones de la medida de entropía de Shannon, que también se utilizan con frecuencia, son la entropía de Renyi (Renyi, 1961) y la entropía de Tsallis (Tsallis, 1988). En Amigó et al. (2018) se puede encontrar una revisión sobre las entropías generalizadas.

Como ya se ha adelantado en el capítulo 1, existen diferentes tipos de algoritmos de clasificación en función de su estructura o de los principios que los sustentan. Así, se puede encontrar algoritmos de clasificación (1) basados en la inducción de algoritmos de árboles de decisión como ID3

(Quilan, 1986) y su extensión C4.5 (Quilan, 1992), el algoritmo de árbol de clasificación y regresión CART (Breiman et al., 1984), y los algoritmos de random forest (Ho, 1995, 1998; Breiman, 2001)); (2) basados en similitudes como los algoritmos de K nearest neighbor (Cover y Hart, 1967; Dasarathy, 1991) y sus extensiones a algoritmos basados en instancias como IBL (Aha et al., 1991); (3) basados en métodos de separación en espacios vectoriales como los algoritmos de support vector machines (SVM) (Cortes y Vapnik, 1995; Ben-Hur et al., 2001); o (4) basados en conceptos y métodos probabilísticos o estadísticos como el análisis discriminante lineal (McLachlan, 2004), la regresión logística o los algoritmos de naïve Bayes (Langley y Thompson, 1994; John y Langley, 1995); entre otros. Para más detalles sobre los problemas de clasificación y aprendizaje y sus algoritmos, véase Aggarwal (2015). Además, se puede encontrar en la literatura de aprendizaje automático muchos trabajos en los que se utilizan diferentes conceptos y métodos de la entropía de la información junto con algoritmos de clasificación de aprendizaje para diseñar nuevos clasificadores que se apliquen en diferentes contextos (Ramírez-Gallego et al., 2018; Rahman et al., 2020; Wang et al., 2019; Mannor et al., 2005; Lee et al., 2001; Cleary et al., 1995; Holub et al., 2008; Fujino et al., 2008; Fan et al., 2017; Ramos et al., 2018; Berezinski et al., 2015).

Dado el mismo conjunto de datos, no todos los clasificadores son igual de precisos en sus predicciones. La precisión alcanzada por un modelo de clasificación depende de varios factores, como la propia implementación del algoritmo, la heurística de poda y boosting incorporadas, el conjunto de datos utilizado e incluso el conjunto de variables finalmente elegido para la construcción del modelo. Por lo tanto, el análisis del rendimiento de los clasificadores es relevante para determinar cuál funciona mejor. Se sabe que existe un límite inferior en la tasa de error que pueden alcanzar los clasificadores: el error de Bayes (Fukunaga, 1990). Este error está asociado al clasificador de Bayes, que asigna una observación a la clase con mayor probabilidad posterior (Fukunaga, 1990). De manera que, este clasificador y su error asociado pueden considerarse como puntos de referencia para evaluar el rendimiento de un clasificador determinado. Sin embargo, el error de Bayes solo puede calcularse para un número reducido de problemas. Por lo tanto, se pueden encontrar diferentes

aproximaciones y límites de este error en la literatura (véase, por ejemplo, Tumer y Ghosh (1996) y las referencias en él incluidas). Como se ha comentado en el capítulo 1, en la literatura de aprendizaje automático, existen diferentes medidas del rendimiento de un clasificador y se puede encontrar varios trabajos que analizan el rendimiento de diferentes clasificadores en función de ellas. Costa et al. (2007) mostraron que las medidas de evaluación más habituales en la práctica eran inadecuadas para los clasificadores jerárquicos y revisaron las principales medidas de evaluación para los clasificadores jerárquicos. Sokolova y Lapalme (2009) analizaron cómo diferentes tipos de cambios en la matriz de confusión afectaban a las medidas de rendimiento de los clasificadores. En particular, estudiaron las propiedades de invariabilidad de 24 medidas de rendimiento para clasificadores binarios, multiclase, multietiquetados y jerárquicos. Ferri et al. (2009) llevaron a cabo un experimento para analizar 18 medidas de rendimiento diferentes de clasificadores. También estudiaron las relaciones entre las medidas y su sensibilidad desde diferentes enfoques. Parker (2011) analizó las incoherencias de siete medidas de rendimiento para clasificadores binarios tanto desde un punto de vista teórico como empírico para determinar qué medidas eran mejores. Labatut y Cherifi (2011) estudiaron propiedades y el comportamiento de 12 medidas de rendimiento para clasificadores planos multiclase. Jiao y Du (2016) revisaron las medidas de rendimiento más comunes utilizadas en predictores bioinformáticos para clasificaciones. Valverde-Albacete y Peláez-Moreno (2010, 2014, 2017, 2020) analizaron el rendimiento de la clasificación con métodos teóricos de la información. En particular, propusieron analizar los clasificadores mediante medidas entrópicas sobre sus matrices de confusión. Para ello, utilizaron el diagrama de entropía de Finetti o triángulo de entropía y una descomposición adecuada de una entropía de tipo Shannon, y luego definieron dos medidas de rendimiento para los clasificadores: la precisión modificada por la entropía (EMA) y el factor de transferencia de información normalizado (NIT). El EMA es la proporción esperada de veces que el clasificador adivina correctamente la clase de salida, y el factor NIT es la proporción de información disponible transferida de la entrada a la salida. El cociente de estas dos medidas proporciona información sobre la cantidad de información disponible para el aprendizaje.

En este capítulo, nos centramos en la definición de las medidas de rendimiento. En particular, siguiendo las ideas sobre los coeficientes de concordancia de la estadística, el $\varkappa$ de Cohen (1960) y el π de Scott (1955), que también se han utilizado como medidas de rendimiento de los clasificadores (Witten y Frank, 2005), se consideran tres medidas de rendimiento estrechamente relacionadas con ellas. Estos estadísticos se definieron originalmente para medir el nivel de concordancia entre las clasificaciones realizadas por dos evaluadores. La fórmula matemática es la siguiente:

$$Nivel\ de\ concordancia = \frac{P_0 - P_e}{1 - P_e}, \tag{4.1}$$

donde P_0 representa la proporción observada de clasificaciones en las que coinciden los dos evaluadores al clasificar los mismos datos de forma independiente; y P_e es la proporción de acuerdo que cabe esperar sobre la base del azar. Dependiendo de cómo se defina P_e, se obtiene la $\varkappa$ de Cohen o la π de Scott. En el aprendizaje automático, estos estadísticos se utilizan como medidas de rendimiento considerando el clasificador a evaluar y un clasificador aleatorio, donde P_0 es la precisión del clasificador. En este capítulo, se consideran estas medidas de rendimiento desde otro punto de vista y se definen tres nuevas medidas de rendimiento basadas en la π de Scott. En particular, se utiliza la interpretación dada en Goodman y Kruskal (1954) para los estadísticos λ. Así, se consideran tres clasificadores de referencia, el clasificador aleatorio y dos clasificadores intuitivos. Los tres clasificadores asignan clases a las nuevas observaciones utilizando la información de la distribución de frecuencias de todos los atributos en los datos de entrenamiento. Para ser más específicos, el clasificador aleatorio, X, predice al azar con la distribución de frecuencias de las clases a la mano, mientras que el primer clasificador intuitivo, V, predice el resultado más probable para cada posible observación con la distribución de frecuencias de las clases en los datos de entrenamiento, y el segundo clasificador intuitivo, I, predice el resultado más probable para cada posible observación con la distribución de frecuencias conjunta de todos los atributos en los datos de entrenamiento. Los dos clasificadores intuitivos descritos se postularon, construyeron y analizaron, pero se rechazaron en favor de tecnologías de clasificación más modernas antes del año 2000. Sin embargo, podrían seguir siendo útiles para definir otras medidas de

rendimiento al estilo de la $\varkappa$ de Cohen o la π de Scott. Así, para evaluar un clasificador se determina la reducción proporcional del error de clasificación cuando se utiliza el clasificador a evaluar respecto a utilizar uno de los clasificadores de referencia. En este sentido, P_0 es la precisión del clasificador a evaluar y P_e es la precisión (esperada) del clasificador de referencia. En el caso de que el clasificador de referencia sea el clasificador aleatorio se obtiene una medida de rendimiento como la π de Scott, pero la interpretación que se da es diferente a la habitual en la literatura de aprendizaje automático. Como ya adelantaba el capítlo 1, este es también un enfoque interesante de la evaluación del rendimiento de los clasificadores porque se puede medir lo ventajoso que es un nuevo clasificador con respecto a tres clasificadores de referencia simples que pueden considerarse como las mejores opciones de sentido común para personas no expertas, pero suficientemente inteligentes, y cuyas tasas de error son más sencillas de determinar que el error de Bayes.

Por otro lado, se analiza la relación entre los tres clasificadores de referencia y diferentes aspectos de la entropía del conjunto de datos. Así, el clasificador aleatorio X y el clasificador intuitivo V están directamente relacionados con la entropía del atributo objetivo, mientras que el clasificador intuitivo I está estrechamente relacionado con la entropía del atributo objetivo cuando se considera todo el conjunto de datos, es decir, con la entropía condicional del atributo objetivo dadas las restantes variables del conjunto de datos. Atendiendo a estas relaciones, se puede analizar el rendimiento de los clasificadores teniendo en cuenta la entropía del conjunto de datos (Williams y Beer, 2010). Este es un enfoque interesante porque permite identificar bajo qué condiciones de incertidumbre informativa (medida mediante la entropía) funciona mejor un clasificador.

Hasta donde se sabe, las principales contribuciones de este capítulo a la literatura de aprendizaje automático son las siguientes:

1. Se consideran el clasificador aleatorio y dos clasificadores intuitivos como clasificadores de referencia. Estos clasificadores pueden considerarse sencillos, intuitivos y naturales para los decisores no expertos con sentido común.

2. Se definen tres nuevas medidas de rendimiento de los clasificadores basadas en la π de Scott, la precisión de los clasificadores y los clasificadores de referencia.
3. Se interpretan nuestras medidas de rendimiento de los clasificadores en términos de reducción proporcional del error de clasificación. Por lo tanto, se mide cuánto mejora un clasificador la clasificación realizada por los clasificadores de referencia. Esta interpretación es interesante porque es fácil de entender y, al mismo tiempo, se determina la ganancia de precisión relacionada con tres clasificadores simples. En cierto sentido, proporcionan información sobre si el diseño del clasificador ha merecido la pena.
4. Las tres medidas de rendimiento de los clasificadores se sitúan en el intervalo [−1, 1], donde −1 significa que el clasificador en evaluación empeora en un 100% la clasificación correcta realizada por el clasificador de referencia correspondiente, esto corresponde a que el clasificador asigna incorrectamente todas las observaciones, y 1 significa que el clasificador reduce en un 100% la clasificación incorrecta realizada por el clasificador de referencia correspondiente, esto corresponde a que el clasificador asigna correctamente todas las observaciones.
5. Los clasificadores de referencia captan la entropía del conjunto de datos. El clasificador aleatorio X y el clasificador intuitivo V miden la entropía del atributo objetivo, y el clasificador intuitivo I refleja la entropía condicional del atributo objetivo dadas las restantes variables del conjunto de datos. Por lo tanto, permiten analizar el rendimiento de un clasificador teniendo en cuenta la entropía del conjunto de datos. Estas medidas, en particular la basada en los clasificadores intuitivos, ofrecen una información diferente a otras medidas de rendimiento de los clasificadores, que se consideran interesantes. El objetivo, por tanto, no es sustituir ninguna medida de rendimiento conocida, sino proporcionar una medida de un aspecto diferente del rendimiento de un clasificador.
6. Se lleva a cabo un experimento intensivo para ilustrar cómo funcionan las medidas de rendimiento propuestas y cómo la entropía puede afectar al rendimiento de un clasificador. Para ello, se consideran un conjunto de datos concreto y el algoritmo de

clasificación J48 (Yadav y Chandel, 2015; Alloghani et al., 2019; Romeo et al., 2020), una implementación proporcionada por Weka (Witten y Frank, 2005; Frank et al., 2016; Weka, 2020, 2021), del clásico algoritmo C4.5 presentado por Quinlan (1986, 1992).

7. Para validar lo observado en el experimento anterior, se realiza un amplio experimento utilizando cuatro clasificadores implementados en Weka y 11 conjuntos de datos.

El resto del capítulo se organiza como sigue. En la sección 4.2, se presenta la metodología y los materiales utilizados en el capítulo. En particular, el método de selección de características, el algoritmo del clasificador intuitivo, la descripción de varios clasificadores heurísticos implementados en Weka (Witten y Frank, 2005; Frank et al., 2016; Weka, 2020, 2021), y la definición y el análisis teórico de las medidas de rendimiento introducidas en este capítulo. En la sección 4.3, se lleva a cabo el experimento para ilustrar cómo funcionan las medidas de rendimiento y cómo pueden utilizarse para analizar el rendimiento de los clasificadores en términos de entropía. En la sección 4.4, se discuten los resultados obtenidos y se concluye. Las tablas se incluyen en la sección 4.5.

4.2. Materiales y métodos

4.2.1. Método y software utilizados para la selección de características

El método utilizado para realizar la selección y la clasificación de las variables más influyentes es la Evaluación de Atributos de la Relación de Ganancia (Trabelsia et al., 2017) (implementada en Weka (Witten y Frank, 2005; Frank et al., 2016; Weka, 2020, 2021)). Esta medida, $GR(att)$ en la ecuación (4.2), proporciona un criterio objetivo para ordenar las variables explicativas por importancia frente a la variable objetivo. El *gain ratio*, por su propio diseño, penaliza la proliferación de nodos y mejora las variables que se distribuyen de forma uniforme. El ratio de ganancia de cada atributo se calcula mediante la siguiente fórmula:

$$GR(att) = \frac{IG(att)}{H(att)}, \tag{4.2}$$

donde $IG(att)$ es una medida para evaluar la ganancia informativa proporcionada por cada atributo, que se considera una medida popular

para evaluar los atributos. En concreto, es la diferencia entre la entropía del atributo consecuente y la entropía cuando se conoce att, $H(att)$. Así, el método de selección de características calcula la ganancia informativa para cada atributo att (Trabelsia et al., 2017).

4.2.2. Metodología y software para el método de clasificación intuitivo I

La idea básica del clasificador intuitivo es generar reglas de clasificación a partir de un conjunto de datos donde todos los valores son discretos (etiquetas de texto). Los datos del conjunto de datos tendrán C columnas o atributos $(A_1,\ldots,A_C)$. Uno de los atributos (A_C en la figura 4.1) es la variable objetivo, utilizada para clasificar las instancias. Los demás atributos $(A_1,\ldots,A_{C-1})$ son las variables explicativas del problema o antecedentes.

$$rule: \langle A_1 = V_1\rangle,\cdots,\langle A_{C-1} = V_{C-1}\rangle \rightarrow \langle A_C = V_C\rangle \tag{4.3}$$

Una regla de clasificación constará de un antecedente (lado izquierdo de la regla) y un consecuente (lado derecho de la regla), como se ilustra en la ecuación (4.3). El antecedente estará compuesto por $C-1$ pares atributo/valor ($\langle A_i = V_i\rangle$), donde los atributos son las variables explicativas. El consecuente consistirá en un par de atributos (variable/valor objetivo) de la forma $\langle A_C = V_C\rangle$.

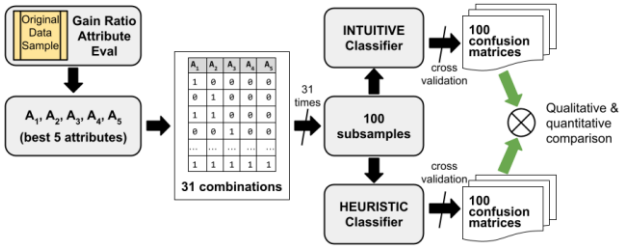

Figura 4.1: Experimento para cada escenario de datos

El clasificador intuitivo I cuenta los valores más repetidos dentro de la muestra de datos. Esto podría ser lo que haría cualquier persona no experta para intentar identificar los patrones más probables de una

muestra de datos aplicando el sentido común. El algoritmo del clasificador intuitivo I (véase el algoritmo 4.1) realiza un barrido exhaustivo por todos los registros del conjunto de datos y cuenta cuántas veces se da cada combinación de valores en el lado izquierdo de la regla (antecedente), hasta llegar a lo que se llamará soporte de la regla (R.supp). Análogamente, dado un antecedente, para cada regla de clasificación, el algoritmo cuenta el número de veces que se da cada una de sus posibles consecuencias o parte derecha de la regla. Se le llama confianza de la regla (R.conf). (véase el algoritmo 4.1).

Algoritmo 4.1. Pseudocódigo del algoritmo del clasificador intuitivo I.

```
1: INPUT:
2: S: muestra de datos de entrenamiento con C columnas y N filas
3: Los atributos C – 1 son el antecedente
4: 1 variable de clase es el consecuente
5: INICIO DEL ALGORITMO
6: CRS←∅ /*inicializado como void set*/
7: for cada fila de S do
8:     if existe una regla Rj en CRS tal que Antecedent(Rj) = Antecedent(fila)
       and Consequent(Rj) = Consequent(fila) then
9:         for all Ri en CRS tal que Antecedent(Ri) = Antecedent(fila) do
10:            Ri.supp←Ri.supp + 1
11:        end for
12:        Rj.conf←Rj.conf + 1
13:    else
14:        R←Nueva regla
15:        R.antecedent←Antecedent(fila)
16:        R.consequent←Consequent(fila)
17:        R.supp←1
18:        R.conf←1
19:        for all Ri en CRS tal que Antecedent(Ri) = Antecedent(fila) do
20:            Ri.supp←Ri.supp + 1
21:        end for
22:        CRS←CRS + R /*añadir R a CRS*/
23:    end if
24: end for
25: return CRS: Conjunto de reglas de clasificación /*OUTPUT*/
26: FIN DEL ALGORITMO
```

Obsérvese que cada regla (R) del conjunto de reglas (CRS), generada según el algoritmo 4.1, tiene asociados valores de soporte y confianza

(R.supp, R.conf). Estos valores son, como se ha indicado anteriormente, el número de veces que se repite el antecedente en la muestra de datos y, el número de veces que, dado un determinado antecedente, se repite su clase del consecuente en la muestra de datos. Estos dos contadores permiten determinar qué patrones son los más repetidos. Este modelo, formado por el conjunto de reglas CRS, predice la variable de clase de una instancia "s" aplicando el algoritmo 4.2.

El algoritmo 4.2 infiere el valor de la clase de instancia "s", utilizando la regla de conjunto CRS cuyo antecedente se asemeja más al antecedente de "s" (coincidiendo con un mayor número de atributos). En el caso de que haya varias reglas con el mismo número de coincidencias, se selecciona la que tiene un mayor soporte. Si hay varias reglas con igual soporte, se elige la de mayor confianza. Una vez identificada esa regla, la clase predicha es el valor del consecuente de la regla seleccionada.

Algoritmo 4.2. Pseudocódigo del algoritmo para predecir con un modelo CRS.

```
INPUT:
s: fila de prueba con atributos antecedentes C − 1
CRS: Conjunto de reglas de clasificación
USE: RSS: Subconjunto de reglas
INICIAR EL ALGORITMO
for c = C − 1 to 1 do
    RSS←{Ri∈CRS / c atributos de s son iguales a c atributos de Ri}
    if RSS ≠ ∅ then
        R←R1 /* R1 es la primera regla de RSS */
        for j = 2 to |RSS| do
            if R.supp < Rj.supp then
                R←Rj
            else if R.supp = Rj.supp and R.conf < Rj.conf then
                R←Rj
            end if
        end for
        return R.consequent /*OUTPUT*/
    end if
end for
FIN DEL ALGORITMO
```

4.2.3. Metodología y software para los clasificadores heurísticos

Para la generación de modelos predictivos a partir del enfoque heurístico, se consideran varios clasificadores heurísticos: J48, naïve Bayes, SMO y random forest.

El modelo de árbol de decisión J48 (Yadav y Chandel, 2015; Alloghani et al., 2019; Romeo et al., 2020) es una implementación proporcionada por Weka del algoritmo clásico C4.5 (Quinlan, 1986, 1992). J48 extiende algunas de las funcionalidades de C4.5 como permitir que el proceso de post-poda del árbol se realice por un método basado en la reducción de errores o que las divisiones sobre variables discretas sean siempre binarias, entre otras (Witten y Frank, 2005). Estos árboles de decisión se consideran métodos de clasificación supervisada. Existe una variable dependiente o de clase (variable de naturaleza discreta), y el clasificador, a partir de una muestra de entrenamiento, determina el valor de esa clase para los nuevos casos. El proceso de construcción del árbol comienza con el nodo raíz, que tiene asociados todos los ejemplos o casos de entrenamiento. Primero se elige la variable o atributo a partir del cual dividir la muestra de entrenamiento original (nodo raíz), buscando que en los subconjuntos generados haya una mínima variabilidad respecto a la clase. Este proceso es recursivo, es decir, una vez obtenida la variable con mayor homogeneidad respecto a la clase en los nodos hijos, se vuelve a realizar el análisis para cada uno de los nodos hijos. Este proceso recursivo se detiene cuando todos los nodos hoja contienen casos de la misma clase, y entonces debe evitarse el sobreajuste, para lo cual se implementan los métodos de pre-poda y post-poda de los árboles.

También se considera el algoritmo naïve Bayes implementado en Weka (Witten y Frank, 2005; Frank et al., 2016; Weka, 2020, 2021), que es un clasificador muy conocido (Langley y Thompson, 1994; John y Langley, 1995) basado en el Teorema de Bayes. Los detalles sobre los clasificadores naïve Bayes se pueden encontrar prácticamente en cualquier libro de ciencia de datos o aprendizaje automático. Por otro lado, Frank et al. (2016) es una excelente referencia para el software Weka.

El SMO es una implementación en Weka (Witten y Frank, 2005; Frank et al., 2016; Weka, 2020, 2021) del algoritmo de optimización mínima secuencial de Platt (Platt, 1998; Keerthi et al., 2001; Hastie y Tibshirani,

1998) para el entrenamiento de un clasificador de máquina de vectores de soporte (Cortes y Vapnik, 1995). El SMO es un algoritmo sencillo para resolver rápidamente los problemas cuadráticos de la máquina de vectores de soporte mediante la descomposición del problema cuadrático global en subproblemas cuadráticos más pequeños que son más fáciles y rápidos de resolver.

Por último, también se utilizará el clasificador de random forest implementado en el software Weka (Witten y Frank, 2005; Frank et al., 2016; Weka, 2020, 2021). Los clasificadores de random forest (Breiman, 2001) consisten en conjuntos de árboles de decisión que se construyen a partir de subconjuntos seleccionados aleatoriamente del conjunto de entrenamiento, y la clasificación final es el resultado de la agregación de la clasificación proporcionada por cada árbol.

4.2.4. Medidas de evaluación

La evaluación de los clasificadores o modelos de predicción es muy importante porque permite (1) comparar diferentes clasificadores o modelos para hacer la mejor elección, (2) estimar cómo se comportará el clasificador o modelo en la práctica, y (3) convencer al responsable de la toma de decisiones de que el clasificador o modelo será adecuado para su propósito (véase Aggarwal, 2015; Kelleher et al., 2015). La forma más sencilla de evaluar un clasificador para un problema concreto dado por un conjunto de datos es considerar la proporción de clasificación correcta. Si se denota por Z el clasificador y por D el conjunto de datos, el rendimiento de Z al clasificar un atributo concreto (el consecuente) en D viene dado por

$$precisión\big(Z(D)\big) = \frac{número\ de\ predicciones\ correctas}{total\ de\ predicciones} \tag{4.4}$$

Esta medida se conoce como precisión. Existen otras medidas de evaluación (véase Aggarwal, 2015; Kelleher et al., 2015), pero en este capítulo nos centramos en definir nuevas medidas basadas de alguna manera en los conceptos de reducción proporcional del error de clasificación (Goodman y Kruskal, 1954) y de entropía (Shannon, 1948).

Nuestro enfoque para definir las medidas de evaluación basadas en la entropía consiste en considerar clasificadores simples que capturan la

entropía del problema. Estos clasificadores desempeñan el papel de referencia a la hora de evaluar otros clasificadores.

Considérese un conjunto de datos D con N instancias (filas) y C atributos (columnas) de forma que los atributos $A_1,\ldots,A_{C-1}$ se consideran las variables explicativas (antecedentes) y A_C es el atributo que debe explicarse o predecirse (consecuente). Sean $a_{C1}, a_{C2}, \ldots, a_{CK}$ las categorías o clases de la variable A_C, y que $p_{C1}, p_{C2}, \ldots, p_{CK}$ son las frecuencias relativas de esas categorías en D. Asociado a este problema, se puede considerar una variable aleatoria X del espacio muestral $\Omega = \{a_{C1}, a_{C2}, \ldots, a_{CK}\}$ a $\mathbb{R}$, tal que $X(a_{Cj}) = j$, y $Prob(X = j) = p_{Cj}$. Por tanto, X tiene la distribución discreta no uniforme $D(p_{C1}, p_{C2}, \ldots, p_{CK})$, es decir, $X \sim D(p_{C1}, p_{C2}, \ldots, p_{CK})$. Esta X puede considerarse el clasificador aleatorio para el consecuente A_C en el conjunto de datos D, definido como

$$X(A_C, D)(i) = X(i), \tag{4.5}$$

donde i es una observación o instancia. Además, se puede definir otro clasificador simple e intuitivo para el consecuente A_C en el conjunto de datos D como sigue

$$V(A_C, D)\ (i) = arg\ max\ \{p_{C1}, p_{C2}, \ldots, p_{CK}\}, \tag{4.6}$$

donde i es una observación o instancia, es decir, este clasificador intuitivo predice el resultado más probable para cada posible observación con la distribución de frecuencia del consecuente A_C.

Si se toman las N instancias del conjunto de datos, entonces la clasificación de cada instancia i por el clasificador aleatorio X tiene una distribución categórica, generalizada de Bernoulli o multinoulli con el parámetro p_i, donde p_i es la frecuencia asociada a la categoría que toma el atributo A_C para la instancia i, es decir, $X(i) \sim B(p_i)$. Por tanto, el número esperado de aciertos en la clasificación de las N instancias viene dado por

$$E\left(\sum_{i=1}^{N} X(i)\right) = \sum_{i=1}^{N} E\big(X(i)\big) = \sum_{i=1}^{N} p_i = \sum_{j=1}^{K} p_{Cj} N p_{Cj} = N \sum_{j=1}^{K} p_{Cj}^2. \tag{4.7}$$

Suponiendo que la clasificación de cada instancia se realiza de forma independiente, la varianza del número de aciertos en la clasificación de las N instancias viene dada por

$$V\left(\sum_{i=1}^{N} X(i)\right) = \sum_{i=1}^{N} V\big(X(i)\big) = \sum_{i=1}^{N} p_i(1-p_i) = \sum_{j=1}^{K} p_{Cj} N p_{Cj}\big(1-p_{Cj}\big) = N\sum_{j=1}^{K} p_{Cj}^2\big(1-p_{Cj}\big). \quad (4.8)$$

Obsérvese que si se considera un conjunto de instancias diferente al conjunto de datos D, entonces las ecuaciones (4.7) y (4.8) vendrían dadas por

$$E\left(\sum_{i=1}^{N'} X(i)\right) = \sum_{j=1}^{k} N'_{Cj} p_{Cj} \quad y \quad V\left(\sum_{i=1}^{N'} X(i)\right) = \sum_{j=1}^{k} N'_{Cj} p_{Cj}\left(1-p_{Cj}\right), \quad (4.9)$$

donde N'_{Cj} es el número de instancias para las que el atributo A_C toma el valor a_{Cj}. Asimismo, si se está interesado en la proporción de éxito en la clasificación, entonces la ecuación (4.7) simplemente se convierte en

$$E\left(\sum_{i=1}^{N} X(i)\right) = \sum_{j=1}^{K} p_{Cj}^2. \quad (4.10)$$

Así, la ecuación (4.10) proporciona la precisión esperada del clasificador aleatorio X, es

$$E\left(\sum_{i=1}^{N} X(i)\right) = E\left(precisión\big(X(A_C, D)\big)\right). \quad (4.11)$$

De la misma manera, se puede llegar a que la precisión del clasificador V es

$$precisión\big(V(A_C, D)\big) = max\{p_{C1}, p_{C2}, \dots, p_{CK}\}. \quad (4.12)$$

Por otro lado, la entropía de Shannon (Shannon, 1948) del atributo A_C en el conjunto de datos D viene dada por

$$H^S(A_C, D) = -\sum_{j=1}^{K} p_{Cj} \log_2 p_{Cj}. \quad (4.13)$$

La entropía de Shannon puede verse como una medida de entropía de Renyi (Renyi, 1961) o una medida de entropía de Tsallis (Tsallis, 1988), que tienen las siguientes expresiones matemáticas para el atributo A_C en el conjunto de datos D,

$$H^{R,\alpha}(A_C, D) = \frac{1}{1-\alpha}\log_2\left(\sum_{j=1}^{K} p_{Cj}^{\alpha}\right), y \quad (4.14)$$

$$H^{T,\alpha}(A_C, D) = \frac{1}{\alpha-1}\left(1-\sum_{j=1}^{K} p_{Cj}^{\alpha}\right), \quad (4.15)$$

respectivamente.

Las medidas de entropía de Renyi y de Tsallis coinciden con la entropía de Shannon cuando α llega a 1, por lo que la medida de entropía de Shannon se considera una medida de entropía de Renyi o una medida de

entropía de Tsallis de orden α = 1. Si se considera la medida de entropía de Renyi y la medida de entropía de Tsallis de orden α = 2, se obtiene

$$H^{R,2}(A_C, D) = -\log_2\left(\sum_{j=1}^{K} p_{Cj}^{\alpha}\right), y \tag{4.16}$$

$$H^{T,\alpha}(A_C, D) = \left(1 - \sum_{j=1}^{K} p_{Cj}^{\alpha}\right). \tag{4.17}$$

Las medidas de entropía dadas en las ecuaciones (4.16) y (4.17) están muy relacionadas con la ecuación (4.10), que mide el ratio de éxito esperado en la clasificación del clasificador aleatorio X.

Ahora, se tiene el siguiente resultado que relaciona el ratio de éxito esperado del clasificador aleatorio y las diferentes medidas de entropía anteriores del consecuente A_C cuando es binario.

Teorema 4.1. *Sean* D, *y* D^* *dos conjuntos de datos con los mismos atributos y* A_C *un atributo binario que se considera el consecuente. Entonces, se cumplen los siguientes enunciados:*

1. $H^S(A_C, D) > H^S(A_C, D^*) \Leftrightarrow H^{R,2}(A_C, D) > H^{R,2}(A_C, D^*)$.
2. $H^S(A_C, D) > H^S(A_C, D^*) \Leftrightarrow H^{T,2}(A_C, D) > H^{T,2}(A_C, D^*)$.
3. $H^S(A_C, D) > H^S(A_C, D^*) \Leftrightarrow p_{C1}^2 + p_{C2}^2 < p_{C1}^{*2} + p_{C2}^{*2}$

Demostración. Para demostrar el teorema basta con demostrar el enunciado 3, ya que los otros dos enunciados se deducen de las expresiones matemáticas de $H^{R,2}$ y $H^{T,2}$ y del enunciado 3. Sean p_{C1}, p_{C2} y p_{C1}^*, p_{C2}^* dos distribuciones de frecuencia de A_C tales que la entropía asociada con la primera es mayor que la entropía asociada con la segunda. Considérese que $p_{C1} \neq p_{C1}^*$, entonces $p_{C2} \neq p_{C2}^*$. En caso contrario, el resultado es inmediato. Como la entropía de la primera distribución de frecuencias es mayor que la entropía de la segunda, se sabe que $|p_{C1} - p_{C2}| < |p_{C1}^* - p_{C2}^*|$. Supóngase, sin pérdida de generalidad, que $p_{C1} > p_{C1}^*$. Como $p_{C1} + p_{C2} = p_{C1}^* + p_{C2}^* = 1, p_{C2} < p_{C2}^*$.

Por otro lado, se tiene que

$$p_{C1}^2 + p_{C2}^2 - (p_{C1}^{*2} + p_{C2}^{*2}) = p_{C1}^2 + (1 - p_{C1})^2 - (p_{C1}^{*2} + (1 - p_{C1}^*)^2). \tag{4.18}$$

Después de algunos cálculos, se tiene que

$$p_{C1}^2 + p_{C2}^2 - (p_{C1}^{*2} + p_{C2}^{*2}) = -2p_{C1}(1 - p_{C1}) + 2p_{C1}^*(1 - p_{C1}^*) < 0 \tag{4.19}$$

donde la última desigualdad viene del hecho de que $|p_{C1} - p_{C2}| < |p^*_{C1} - p^*_{C2}|$, y la función $x(1-x)$ es mayor cuanto menor sea la diferencia entre x y $(1-x)$. Por lo tanto, $p^2_{C1} + p^2_{C2} < p^{*2}_{C1} + p^{*2}_{C2}$.
La prueba de la relación inversa es similar.

El teorema 4.1 no puede extenderse a atributos con más de 2 valores posibles, como muestra el siguiente ejemplo.

Ejemplo 4.1. Considérese dos conjuntos de datos D y D', y un atributo común A para ambos con tres valores posibles $\{a, b, c\}$, tales que $p_a = 0.54$, $p_b = 0.01$, $p_c = 0.45$, y $p'_a = 0.25$, $p'_b = 0.05$, $p'_c = 0.70$. En esta situación, se tiene que $H^S(A, D) = 1.065 < 1.076 = H^S(A, D')$, pero $H^{T,2}(A, D) = 0.506 > 0.445 = H^{T,2}(A, D')$.

Por otro lado, si se considera la medida de entropía de Renyi cuando α va a ∞, se obtiene

$$H^{R,\infty}(A_C, D) = -\log_2(max\{p_{C1}, p_{C2}, \dots, p_{CK}\}), \tag{4.20}$$

y se pueden demostrar resultados similares a los anteriores. Sin embargo, todas las medidas de entropía de Renyi están correlacionadas, por lo que H^S, $H^{R,2}$, y $H^{R,\infty}$, también están correlacionados.

A la vista del análisis anterior, la entropía del atributo A_C es captada de alguna manera por el clasificador aleatorio X y el clasificador intuitivo V, en el sentido de que cuanto mayor sea la entropía, menor será el número (esperado) de aciertos en la clasificación, y a la inversa. Por lo tanto, el clasificador aleatorio X y el clasificador intuitivo V pueden utilizarse como puntos de referencia a la hora de evaluar otros clasificadores, teniendo en cuenta la entropía del consecuente. A continuación, se define una medida de evaluación basada en el análisis anterior.

Definición 4.1. *Sea Z un clasificador. Dado un conjunto de datos D, y un consecuente A_C, el rendimiento de Z con respecto al clasificador aleatorio X viene dado por*

$$\gamma^X(Z(D)) = \begin{cases} \frac{\mu(Z,D)-\mu(X,D)}{1-\mu(X,D)} & si\ \mu(Z,D) - \mu(X,D) \geq 0, \\ \frac{\mu(Z,D)-\mu(X,D)}{\mu(X,D)} & si\ \mu(Z,D) - \mu(X,D) < 0, \end{cases} \tag{4.21}$$

donde $\mu(X, D) = \frac{E(\sum_{i=1}^{M} X(i))}{M}$, tal que M es el número total de predicciones y $\mu(Z, D)$ es la proporción de clasificaciones correctas utilizando el clasificador (Z).

Obsérvese que el primer caso de la definición de la medida de rendimiento γ^X coincide con la π de Scott. Si se utiliza el clasificador intuitivo V en lugar del clasificador de referencia X, se obtiene la medida de rendimiento γ^V. La medida de evaluación γ^X (respectivamente γ^V) oscila entre -1 y 1, donde -1 es el peor caso, y se consigue cuando el clasificador no predice correctamente ninguna instancia; 0 significa que el rendimiento es como el clasificador aleatorio X (resp. γ^V); y 1 es el mejor caso, y se consigue cuando el clasificador clasifica correctamente todas las instancias. Los valores intermedios miden en qué proporción el clasificador rinde mejor (valores positivos) o peor (valores negativos) que el clasificador aleatorio (resp. V).

Por otro lado, se puede interpretar la medida de rendimiento γ^X (resp. γ^V) en términos de reducción proporcional del error de clasificación con respecto al clasificador aleatorio (resp. V). En efecto, si se predicen M instancias, se puede escribir la ecuación (4.21) como sigue:

$$\gamma^X(Z(D)) = \begin{cases} \frac{M\mu(Z,D)-M\mu(X,D)}{M-M\mu(X,D)} & si\ M\mu(Z,D) - M\mu(X,D) \geq 0, \\ \frac{M\mu(Z,D)-M\mu(X,D)}{M\mu(X,D)} & si\ M\mu(Z,D) - M\mu(X,D) < 0. \end{cases} \quad (4.22)$$

Ahora, se puede escribir la ecuación (4.22) de la siguiente manera:

$$\gamma^X(Z(D)) = \begin{cases} \frac{(M-M\mu(X,D))-(M-M\mu(Z,D))}{M-M\mu(X,D)} & si\ M\mu(Z,D) - M\mu(X,D) \geq 0, \\ \frac{M\mu(Z,D)-M\mu(X,D)}{M\mu(X,D)} & si\ M\mu(Z,D) - M\mu(X,D) < 0. \end{cases} \quad (4.23)$$

Por último, la ecuación (4.23) puede interpretarse como sigue:

$$\gamma^X(Z(D)) = \begin{cases} \frac{núm.esperado\ de\ errores\ con\ X - núm.errores\ con\ Z}{núm.esperado\ de\ errores\ con\ X} & si\ M\mu(Z,D) - M\mu(X,D) \geq 0, \\ \frac{núm.aciertos\ con\ Z - núm.esperado\ de\ aciertos\ con\ X}{núm.esperado\ de\ aciertos\ con\ X} & si\ M\mu(Z,D) - M\mu(X,D) < 0. \end{cases} \quad (4.24)$$

Así, el primer caso de γ^X mide la reducción proporcional del error de clasificación cuando se utiliza el clasificador Z con respecto al uso del clasificador aleatorio X. El segundo caso de γ^X mide la reducción proporcional del éxito de la clasificación cuando se utiliza el clasificador Z

con respecto al uso del clasificador aleatorio X. Lo mismo puede decirse cuando se utiliza el clasificador intuitivo V como clasificador de referencia.

Por lo tanto, γ^X da información sobre cuánto mejora o empeora la clasificación un clasificador Z con respecto a un clasificador que decide la clase aleatoriamente teniendo en cuenta la distribución de frecuencias de las clases. Además, γ^V da información sobre cuánto mejora o empeora la clasificación un clasificador Z con respecto a un clasificador que simplemente predice la clase más probable según la distribución de frecuencias de las clases. Dado que los dos clasificadores anteriores solo utilizan información relacionada con las clases, estas dos medidas proporcionan información sobre si es pertinente utilizar clasificadores más sofisticados que incorporen información de otros atributos.

Por otro lado, las medidas γ^X y γ^V incorporan de alguna manera la información sobre la entropía del consecuente a la evaluación de un clasificador, pero no tienen en cuenta el resto de los atributos (los antecedentes). No obstante, se puede realizar un análisis similar considerando todas las posibles cadenas de atributos diferentes, obteniendo resultados análogos. Por otro lado, el método de clasificación intuitiva descrito en la sección 4.2.2 puede ser otra forma de tener en cuenta todos los atributos y la entropía del conjunto de datos, ya que su definición se basa en la repetición de instancias que está relacionada con la entropía del conjunto de datos. En particular, está relacionada con la entropía condicional del atributo A_C dadas las restantes variables del conjunto de datos. Así, otra medida de evaluación de los clasificadores relacionada con la entropía podría ser utilizar este método de clasificación intuitivo como punto de referencia, siendo su definición análoga a las anteriores. A continuación, se expone formalmente la definición de esta medida.

Definition 4.2. *Sea Z un clasificador. Dado un conjunto de datos D, y un consecuente A_C, el rendimiento de Z con respecto al clasificador intuitivo I viene dado por*

$$\Gamma(Z(D)) = \begin{cases} \frac{\mu(Z,D)-\mu(I,D)}{1-\mu(I,D)} & si\ \mu(Z,D) - \mu(I,D) \geq 0, \\ \frac{\mu(Z,D)-\mu(I,D)}{\mu(I,D)} & si\ \mu(Z,D) - \mu(I,D) < 0, \end{cases} \tag{4.25}$$

donde $\mu(I, D)$ *es la proporción de clasificaciones correctas utilizando el clasificador* (I)*, y* $\mu(Z, D)$ *es la proporción de clasificaciones correctas utilizando el clasificador* (Z).

La interpretación de Γ es completamente análoga a la de γ anterior, solo cambiando el clasificador aleatorio X y el clasificador intuitivo V por el clasificador intuitivo I. Sin embargo, da alguna información extra sobre los clasificadores, en el sentido de que, como utiliza toda la información del conjunto de datos, proporciona información sobre cuánto de relevante es utilizar clasificadores más sofisticados.

4.3. Experimentos computacionales: diseño y resultados

En esta sección, se ilustra cómo funcionan las medidas de evaluación introducidas en la sección 4.2. Para ello, se diseña un experimento en el que se considera cinco escenarios de entropía para un atributo binario (el consecuente), y para cada uno de esos escenarios se estudian 31 combinaciones de atributos explicativos (los antecedentes). De este modo, se puede dar una mejor idea de cómo funcionan estas medidas de evaluación y cómo miden el rendimiento de los clasificadores en diferentes situaciones de entropía. A continuación, se va más allá y se realiza una amplia comparación para cuatro clasificadores utilizando 11 conjuntos de datos diferentes cuyos resultados se presentan de forma concisa.

4.3.1. Conjuntos de datos y escenarios

Se parte de la hipótesis de trabajar en un contexto de clasificación en el que el objetivo a predecir es discreto y más concretamente binario, pero podría considerarse otra variable objetivo multiclase. Para el experimento más intensivo se ha elegido un conocido conjunto de datos del UCI *Machine Learning Repository* (Dua y Graff, 2019) denominado "thyroid0387.data".

Este conjunto de datos ha sido ampliamente utilizado en la literatura en problemas relacionados con el campo de la clasificación. Dado que en este capítulo solo se utiliza como ejemplo y no interesa el tema clínico en sí que recogen los datos, para facilitar el experimento de este estudio y hacerlo exhaustivo, dicho conjunto de datos ha sido mínimamente preprocesado como sigue:

- Se han añadido y renombrado las cabeceras.
- Se han eliminado los atributos numéricos y se han dejado solo los nominales.
- La variable de clase se ha recodificado en casos positivos y negativos (la muestra original tiene varios tipos de casos positivos).

Por último, el conjunto de datos utilizado para realizar el experimento tiene las siguientes características:

- Número de filas: 9,173
- Número de atributos/columnas: 23 (todos nominales)
  - 22 variables explicativas (antecedentes)
  - 1 variable objetivo (consecuente)
    - 2,401 casos positivos
    - 6,772 casos negativos

La variable objetivo utilizada para clasificar, que corresponde a un diagnóstico clínico, está desequilibrada, ya que tiene un valor positivo en 2,401 tuplas y un valor negativo en 6,772. A partir de estos datos, se consideran cinco tipos de escenarios posibles con diferentes proporciones entre valores positivos y negativos (véase la tabla 4.1).

TABLA 4.1. Los cinco escenarios de datos.

Escenario	Positivo	Negativo	Total	Ratio positivo/negativo	Entropía del consecuente
S1	2,400	800	32,000	3:1	0.811
S2	2,400	1,200	3,600	2:1	0.918
S3	2,400	2,400	4,800	1:1	1.000
S4	2,000	4,000	6,000	1:2	0.918
S5	2,000	6,000	8,000	1:3	0.811

Los 10 conjuntos de datos restantes utilizados en el experimento más extenso también proceden del UCI Machine Learning Repository (Dua y Graff, 2019). Se han realizado las siguientes modificaciones, comunes a todos ellos.

1. En todos los conjuntos de datos que no tenían una fila con la cabecera, se ha añadido, teniendo en cuenta las especificaciones de

la sección "Información de atributos" de cada uno de estos conjuntos de datos del repositorio UCI.

2. La configuración en Weka para discretizar ha sido con el parámetro "bins" = 5 (para obtener 5 grupos) y el parámetro "UseEqualFrecuency" = true (para que los grupos de datos obtenidos fueran equitativos).
3. Al discretizar en Weka (filtro discretizado no supervisado) los resultados obtenidos eran intervalos numéricos, por lo que posteriormente se les cambió el nombre.

En particular, además del conjunto de datos ya mencionado, se ha utilizado los siguientes conjuntos de datos:

(1) "Healthy_Older.data" (Shinmoto Torres et al., 2013);

(2) "Avila.data" (de Stefano et al., 2018);

(3) "Adult.data";

(4) "nursery.data";

(5) "Bank marketing" (Moro et al., 2014);

(6) "HTRU2.data" (Lyon et al., 2016; Lyon 2021);

(7) "connect-4.data";

(8) "tic-tac-toe.data";

(9) "Credit approval.data";

(10) "Mushroom.data".

Las principales características de estos conjuntos de datos se resumen en la tabla 4.2.

TABLA 4.2. Características de los conjuntos de datos. **Filas** es el número de filas del conjunto de datos; **atributos** es el número de atributos incluyendo el consecuente, y debajo el tipo de variables; **clases** es el número de clases del consecuente; y la **distribución de las clases** es el número de casos de cada clase en el consecuente.

Conjunto de datos	Filas	Atributos	Clases	Distribución de las clases
Thyroid	9,173	23 2 categóricos 21 binarios	2	2401, 6772
Healthy	75,128	10 8 reales 1 binario 1 categórico	4	16406, 4911, 51520, 2291
Avila	20,867	11 10 reales 1 categórico	12	8572, 10, 206, 705, 2190, 3923, 893, 1039, 1663, 89, 1044, 533
Adult	32,561	12 3 reales 1 entero 6 categóricos 2 binarios	2	7841, 24720
Nursery	12,960	9 8 categóricos 1 binario	5	4320, 4266, 24044, 328
Bank	45,211	11 1 real 1 entero 5 categóricos 4 binarios	2	39922, 5289
HTRU2	17,898	9 8 reales 1 binario	2	16259, 1639
Connect-4	67,557	43 43 categóricos	3	6449, 16635, 44473
Tic-tac-toe	958	10 9 categóricos 1 binario	2	332, 626
Credit	690	10 5 categóricos 5 binarios	2	383, 307
Mushroom	8,124	23 17 categóricos 6 binarios	2	4208, 3916

Además, se realizaron algunos preprocesamientos específicos de los datos en los conjuntos de datos "Adult.data" y "Bank marketing" (Moro et al., 2014). En "Adult.data", se eliminaron las filas con valores erróneos y se descartaron tres atributos (capital-gain, capital-loss, native-country); y

en "Bank marketing", el conjunto de datos seleccionado fue "bank-full.csv", y se descartaron 6 atributos (balance, day, duration, campaign, pdays y previous).

4.3.2. Diseño experimental

El experimento consiste en determinar la precisión de un clasificador heurístico, el ya mencionado J48, en comparación con tres clasificadores de referencia: el clasificador aleatorio y dos clasificadores intuitivos. Estos tres clasificadores contienen, en cierta medida, información sobre la entropía presente en el conjunto de datos, como se ha explicado en la sección anterior. Por lo tanto, se proporcionan medidas de evaluación de ese clasificador teniendo en cuenta la entropía del sistema. En este sentido, se trata de evaluar cómo se comporta este clasificador en términos de la mejora (o deterioro) obtenida con respecto a tres clasificadores que pueden considerarse como puntos de referencia y que se basan en la distribución simple de los datos del conjunto de datos, y luego en la entropía de los mismos.

Por otro lado, también interesa observar las diferencias entre las tres medidas de evaluación de los clasificadores introducidas en la sección anterior, y qué efecto tiene, considerando más o menos información del conjunto de datos, a la hora de realizar clasificaciones de instancias. Para ello, se considera los cinco escenarios descritos en la tabla 4.1, que tienen diferente nivel de entropía de Shannon en el consecuente. Para cada uno de estos escenarios, se sigue el proceso representado en la figura 4.1.

En primer lugar, partiendo de la muestra original de datos y fijando la variable consecuente (o variable objetivo) A_C que se va a estudiar, se seleccionan las cinco variables (atributos) más correlacionadas con la variable objetivo. A continuación, se ordenan $(A_1, A_2, A_3, A_4, A_5)$, es decir, se determina cuál está más correlacionada con el consecuente y cuál menos, para lo cual se utiliza el método de la relación de ganancia de atributos descrito en la sección 4.2.1. En la tabla 4.3, se muestra las puntuaciones de la relación de ganancia observadas para cada uno de los cinco escenarios $(S1, S2, S3, S4, S5)$ considerados.

TABLA 4.3. Resultados de la evaluación de los atributos de la relación de ganancia en los cinco escenarios.

Atributos	$S1$	$S2$	$S3$	$S4$	$S5$
A_1	0.036	0.050	0.083	0.122	0.102
A_2	0.037	0.037	0.082	0.076	0.134
A_3	0.033	0.034	0.028	0.020	0.016
A_4	0.034	0.032	0.028	0.015	0.013
A_5	0.029	0.022	0.026	0.013	0.010

En este punto, se enfatiza una vez más que el propósito no es analizar un problema particular, sino solo utilizar un conjunto de datos para analizar las medidas de evaluación introducidas en este capítulo y también mostrar un análisis de los clasificadores heurísticos cuando se consideran las características de entropía del conjunto de datos. Por esta razón, los atributos A_1, A_2, A_3, A_4, A_5 no son necesariamente los mismos ni están en el mismo orden en los cinco escenarios. Simplemente se llama genéricamente A_1 al atributo mejor correlacionado con la variable objetivo en cada escenario, aunque no sea la misma variable en cada uno de ellos. En consecuencia, los demás atributos ocupan de la segunda a la quinta posición en el ranking de correlación con el atributo consecutivo en cada escenario, siempre según la evaluación del atributo de la relación de ganancia. En cada uno de los escenarios, estos cinco atributos se utilizarán como variables predictoras o explicativas (antecedentes) para generar los modelos de clasificación. No es objetivo de este capítulo profundizar en los diferentes métodos de selección de subconjuntos de características (atributos), sino que simplemente se utiliza uno de ellos, siempre el mismo (atributo de ratio de ganancia), para trabajar solo con aquellos atributos que en cada caso sean realmente significativos. Reducir el tamaño del problema de 22 a 5 variables explicativas permitirá realizar un experimento completo con el que ilustrar y analizar las dos medidas de evaluación introducidas, y mostrar una forma de analizar el rendimiento de un clasificador heurístico cuando se consideran diferentes grados de entropía en el conjunto de datos. Para seleccionar los cinco mejores atributos, se utiliza el software Weka (Witten y Frank, 2005; Weka, 2020, 2021), en particular, su función *Select attributes*, con *GainRatioAttributeEval* como evaluador de atributos, *ranker* como método de búsqueda, y validación cruzada como modo de selección de atributos. Obsérvese que Weka ofrece dos medidas de la relevancia de los atributos (antecedentes).

El mérito medio y su desviación estándar, y el rango medio y su desviación estándar. La primera se refiere a la media de las correlaciones medidas con *GainRatioAttributeEval* en 10 ciclos (aunque con 5 ciclos hubiera sido suficiente, ya que solo se quieren los 5 primeros atributos) de pliegue de validación. El rango medio se refiere al orden medio en que quedó cada atributo en cada uno de los diez ciclos. Véase (Witten y Frank, 2005; Weka, 2020) para más detalles sobre Weka.

Una vez elegidas las cinco mejores atribuciones, el siguiente paso es establecer las 31 posibles combinaciones del conjunto de variables predictoras. Estas 31 combinaciones serán los antecedentes a considerar en un conjunto de reglas de clasificación o en un árbol de decisión. Es decir, se llevarán a cabo 31 estudios de clasificación para predecir el atributo consecuente A_C en función de cada una de estas combinaciones de variables explicativas (véase la tabla 4.4).

TABLA 4.4. Las 31 combinaciones de los cinco mejores atributos A_1, A_2, A_3, A_4 y A_5 para predecir el consecuente A_C .

Comb.	Antecedentes	Comb.	Antecedentes	Comb.	Antecedentes
#1	A_5	#12	A_2, A_3	#23	A_1, A_3, A_4, A_5
#2	A_4	#13	A_2, A_3, A_5	#24	A_1, A_2
#3	A_4, A_5	#14	A_2, A_3, A_4	#25	A_1, A_2, A_5
#4	A_3	#15	A_2, A_3, A_4, A_5	#26	A_1, A_2, A_4
#5	A_3, A_5	#16	A_1	#27	A_1, A_2, A_4, A_5
#6	A_3, A_4	#17	A_1, A_5	#28	A_1, A_2, A_3
#7	A_3, A_4, A_5	#18	A_1, A_4	#29	A_1, A_2, A_3, A_5
#8	A_2	#19	A_1, A_4, A_5	#30	A_1, A_2, A_3, A_4
#9	A_2, A_5	#20	A_1, A_3	#31	A_1, A_2, A_3, A_4, A_5
#10	A_2, A_4	#21	A_1, A_3, A_5		
#11	A_2, A_4, A_5	#22	A_1, A_3, A_4		

Para cada una de estas combinaciones de atributos se generan 100 submuestras para evitar posibles sesgos en la selección de registros.

En tercer lugar, para cada uno de los escenarios descritos (tabla 4.1), para cada una de las 31 combinaciones de atributos antecedentes (tabla 4.4), y para cada una de las 100 submuestras aleatorias, se generan modelos de clasificación, tanto con los dos clasificadores intuitivos como con el

método heurístico J48. Así, se ha realizado 15,500 modelos de clasificación heurística con el método J48, así como con nuestra propia implementación del clasificador I.

Por último, para ambos clasificadores se calcula sus precisiones, a partir de sus correspondientes matrices de confusión, utilizando la validación cruzada. Así, para calcular el ratio de éxito $\mu(X, D)$ del clasificador aleatorio X, se utiliza directamente el resultado teórico dado por la ecuación (4.7), y lo mismo para el clasificador intuitivo V utilizando la ecuación (4.12), mientras que para calcular el ratio de éxito $\mu(I, D)$ del clasificador intuitivo I, se utiliza la matriz de confusión obtenida por validación cruzada. Asimismo, el ratio de éxito $\mu(Z, D)$ del clasificador heurístico, en nuestro caso J48, se calcula también mediante la matriz de confusión obtenida por validación cruzada. A partir de estos resultados, ya se pueden calcular las medidas de evaluación introducidas en la sección 4.2.4.

Por lo tanto, se tiene un diseño experimental con dos factores (escenarios de entropía y combinaciones de atributos) con 100 réplicas para cada combinación cruzada de factores. Esto permite analizar en profundidad cómo se comporta un clasificador heurístico cuando se considera tanto la entropía de la variable consecuente como el número de atributos utilizados como antecedentes. Por tanto, el experimento ilustra tanto el funcionamiento de las medidas de evaluación como el análisis de los efectos de la entropía y el número de atributos seleccionados para predecir la variable consecuente en el rendimiento de un clasificador heurístico.

4.3.3. Resultados

Después de realizar todos los modelos de clasificación descritos en la sección anterior para cada uno de los cinco escenarios, cada modelo se somete a una prueba de validación cruzada y se determinan las matrices de confusión. Con esta información se pueden calcular algunas medidas de rendimiento para el clasificador heurístico J48. La medida de rendimiento más sencilla es la precisión, que mide la tasa de éxito en la predicción. La tabla 4.5 muestra la precisión de J48 y del clasificador intuitivo I para cada uno de los cinco escenarios considerados.

TABLA 4.5. Medidas de precisión para el clasificador aleatorio, el clasificador intuitivo V, J48 y el clasificador intuitivo I cuando se utiliza la combinación de atributos A_{31} para cada escenario. La precisión y el error absoluto medio se calculan como la precisión media y el error absoluto medio de las 100 submuestras. Los resultados se presentan como precisión ± error absoluto medio.

Escenario	E(acc(X(D)))	acc(V(D))	acc(J48(D))	acc(I(D))
S1	0.6250	0.7500 ± 0.2500	0.7489 ± 0.3739	0.7481 ± 0.2519
S2	0.5556	0.6667 ± 0.3333	0.6724 ± 0.4358	0.6729 ± 0.3271
S3	0.5000	0.5000 ± 0.5000	0.5241 ± 0.4856	0.4835 ± 0.5165
S4	0.5556	0.6667 ± 0.3333	0.6751 ± 0.4366	0.6766 ± 0.3234
S5	0.6250	0.7465 ± 0.2535	0.7543 ± 0.3734	0.7537 ± 0.2487

En la tabla 4.5, se observa que, para este conjunto de datos, el rendimiento de J48 es en promedio ligeramente mejor que el rendimiento del clasificador intuitivo I, pero la media absoluta para J48 son peores que los errores absolutos medios del clasificador intuitivo I, excepto para $S5$. Sin embargo, esta comparación podría analizarse con más detalle considerando otros aspectos como el número de veces que un método supera al otro o la entropía. Asimismo, las mejoras respecto al clasificador intuitivo V no son demasiado grandes, lo que significaría que o bien el modelo no es muy bueno, o bien que en este caso concreto el uso de información de otros atributos y/o clasificadores más sofisticados no aportan mejoras notables respecto al clasificador intuitivo V.

Considérese ahora que un clasificador vence a otro clasificador cada vez que el primero clasifica correctamente un número de elementos del conjunto de prueba superior a los elementos clasificados correctamente por el segundo. Cuando ocurra lo contrario, se dirá que el segundo clasificador supera al primero. Cuando la diferencia entre los elementos bien clasificados por ambos métodos sea 0, se dirá que se ha producido un empate. El número de veces que J48 y el clasificador intuitivo ganan para cada escenario y cada combinación de los cinco mejores atributos se muestra en las tablas 4.A1-4.A5 de la sección 4.5. La tabla 4.6 resume el porcentaje de veces que cada método gana para cada escenario.

TABLA 4.6. Porcentaje de veces que cada método gana en cada una de las 3,100 instancias consideradas (100 submuestras para cada una de las 31 combinaciones de los cinco mejores atributos) para cada escenario dado en la tabla 4.1.

Escenario	**J48 gana**	***V* gana**	**J48 gana**	***I* gana**	***I gana***	***V gana***
S1	10.42	41.71	46.03	23.94	18.39	58.13
S2	67.65	15.65	24.48	37.39	74.90	13.29
S3	97.55	0.16	73.48	4.45	32.52	66.90
S4	78.13	0.26	16.52	42.19	84.52	0.00
S5	98.03	1.23	76.90	15.16	97.29	2.00
Porcentaje %	70.36	11.80	47.48	24.63	61.52	28.06

En la tabla 4.6, se observa que J48 clasifica mejor que el método intuitivo I en el 47.48% de los casos, mientras que el método intuitivo I clasifica mejor que J48 en el 24.63% de los casos. J48 clasifica especialmente mejor en los escenarios $S5$ y $S3$, mientras que el método intuitivo I clasifica mejor en los escenarios $S2$ y $S4$. Además, J48 supera claramente al clasificador intuitivo V en todos los escenarios excepto en $S1$, mientras que el método intuitivo I clasifica mejor que el clasificador intuitivo V en los escenarios $S2$, $S4$ y $S5$. Por lo tanto, en términos absolutos se puede decir que J48 se comporta razonablemente bien con respecto al conjunto de datos utilizado. Sin embargo, además de saber si un método clasifica mejor que otro, es aún más relevante saber cuánto mejor clasifica en términos relativos como se ha mencionado anteriormente. En este sentido, disponer de un punto de referencia es importante para evaluar cuánta mejora hay cuando se compara con él. En las tablas 4.A1-4.A5 de la sección 4.5, se puede encontrar las medidas de evaluación introducidas en la sección 4.2.4 aplicadas a la media de los resultados obtenidos para las 100 submuestras para cada combinación de los mejores atributos cuando se utiliza J48 y el clasificador intuitivo. La tabla 4.7 resume estas medidas para cada uno de los cinco escenarios considerados.

TABLA 4.7. Intervalos de valores de la medida de evaluación γ^X para J48 y el método intuitivo *I*, e intervalos de valores de la medida de evaluación Γ para J48 para cada escenario.

Escenario	γ^X ($J48$)	γ^X (I)	γ^V ($J48$)	γ^V (I)	Γ($J48$)
S1	0.3303-0.3333	0.3282-0.3333	-0.0015-0.0000	-0.0025-0.0000	0.0000-0.0032
S2	0.2499-0.2636	0.2498-0.2650	-0.0001-0.0182	-0.0001-0.0200	-0.0009-0.0001
S3	0.0004-0.0492	-0.0903-0.0419	0.0004-0.0492	-0.0903-0.0419	0.0049-0.0844
S4	0.2500-0.2693	0.2500-0.2729	0.0000-0.0257	0.0000-0.0306	-0.0026-0.0004
S5	0.3134-0.3635	0.3125-0.3627	-0.0087-0.0524	-0.0091-0.0512	0.0010-0.0027

En primer lugar, nótese que en este caso la medida γ^X coincide en todos los escenarios con la π de Scott. Por otro lado, más allá de lo analizado cuando se evalúa qué método clasifica mejor simplemente en función del número de aciertos, en la tabla 4.7 se observa que el rendimiento de J48 y del clasificador intuitivo I son muy similares cuando se comparan con el clasificador aleatorio X y el clasificador intuitivo V para cada uno de los escenarios (columnas correspondientes a las medidas de evaluación γ^X y γ^V). Esto se refleja claramente en la medida de evaluación Γ de J48, que es el resultado de la comparación con el método intuitivo I (véase la definición 4.2). También se observa que, para el conjunto de datos utilizado en el experimento, el rendimiento de los clasificadores mejora con la disminución de la entropía del consecuente, es decir, cuanto menor es la entropía, mayor es el rendimiento de ambos clasificadores con respecto al clasificador aleatorio X.

Además, si se observa, por ejemplo, el escenario $S3$, $\gamma^V(J48)$ se ve que J48 mejora el rendimiento del clasificador intuitivo V, que solo utiliza la información proporcionada por la distribución de frecuencias del atributo objetivo, hasta un 5% utilizando la información proporcionada por otros atributos distintos del atributo objetivo. Por tanto, este porcentaje puede interpretarse como el aprovechamiento que J48 hace de esta información adicional. Si se mira ahora en $\Gamma(J48)$, se ve que esta mejora alcanza casi el 8.5% con respecto al clasificador intuitivo I. Este porcentaje puede interpretarse como el mejor aprovechamiento que hace J48 de la información en relación al clasificador intuitivo I. Llegados a este punto, ya se podría valorar, teniendo en cuenta las implicaciones prácticas de un

mejor rendimiento, si merece la pena el uso de un clasificador más sofisticado que los dos clasificadores intuitivos.

Por lo tanto, la comparación con un punto de referencia es importante porque las medidas de rendimiento a menudo no reflejan lo que realmente se gana con respecto a una forma simple, incluso aleatoria, de clasificar. Por lo tanto, el uso de medidas basadas en clasificadores simples de referencia que capturen de alguna manera la entropía del conjunto de datos parece apropiado y proporciona información relevante sobre el rendimiento de los clasificadores. En particular, el uso de ambos clasificadores intuitivos como referencias (*benchmarks*) parece razonable, porque aunque como clasificadores han sido descartados en favor de otros que utilizan tecnologías más modernas y elaboradas, siguen siendo lo suficientemente fáciles de entender e intuitivos como para al menos considerarlos de referencia a la hora de medir el rendimiento de los clasificadores, ya que el aleatorio se utiliza habitualmente en el aprendizaje automático.

Las tablas 4.5, 4.6 y 4.7 no reflejan qué efecto tiene la combinación de atributos en el desempeño de los métodos de clasificación. Por lo tanto, se puede dar un paso más en el análisis incorporando este aspecto. El resultado del análisis de varianza para diferencias en los efectos de los escenarios de entropía y la combinación de atributos en la medida de evaluación γ^X de J48 se muestra en la tabla 4.7bis.

TABLA 4.7bis. Tabla ANOVA de dos factores para la medida de evaluación γ^X de J48.

Fuente de variación	Suma de cuadrados	df	Cuadrado medio	F-ratio	p-valor
Escenarios	174.8877	4	43.72191	1674039.44	0.0000
Combinaciones	0.2954	30	0.00985	377.05	0.0000
Interacción	0.7244	120	0.00603	231.13	0.0000
Error	0.4008	15345	0.00003		
Total	176.3083	15499			

En vista de la tabla 4.7bis, se puede concluir que tanto el escenario de entropía como la combinación de atributos tienen efectos estadísticamente

significativos sobre el desempeño del clasificador heurístico J48 medido por la medida de evaluación γ^X. En el caso del escenario de entropía, este efecto estadísticamente significativo también es significativo desde un punto de vista práctico, ya que la diferencia en el éxito de la clasificación es muy diferente de un escenario a otro (véase la tabla 4.7). Así, se puede concluir que la capacidad predictiva de J48 con respecto al clasificador aleatorio es mejor cuanto menor es la entropía de la variable consecuente. Sin embargo, no se puede decir lo mismo respecto a la combinación de atributos. Si bien del análisis de varianza se puede concluir que la combinación de atributos tiene un efecto estadísticamente significativo en el desempeño del clasificador heurístico J48, estas diferencias no son realmente relevantes desde un punto de vista práctico en el escenario $S1$, porque la diferencia entre los más pequeños y la media más grande es solo 0.003. En el caso del escenario $S2$, la diferencia más alta es 0.0137, correspondiendo la media más grande a las combinaciones de atributos A_6 y A_{22}. Para el escenario $S3$, la diferencia más alta es 0.0488 y la media de desempeño más grande corresponde a A_7 y A_{23}. En el caso del escenario $S4$, la diferencia es 0.0193 y la media más grande corresponde a la combinación A_{30}. Finalmente, en el escenario $S5$, la diferencia es 0.0136, y el mejor desempeño medio corresponde a A_{28}. Consúltese la tabla 4.7 para obtener detalles sobre los intervalos de variación de las medias de las 100 submuestras de cada combinación de atributos para cada escenario. Además, las desviaciones estándar para cada submuestra varían entre casi 0 y 0,0080.

El análisis estadístico anterior se puede realizar para el clasificador intuitivo I y nuevamente para el clasificador heurístico J48, pero considerando la medida de evaluación Γ. En ambos casos, los resultados obtenidos son similares.

4.3.4. Experimento extensivo

En esta subsección se presenta los resultados de un extenso experimento en el que se considera cuatro clasificadores heurísticos además del clasificador intuitivo I, y 11 conjuntos de datos. En particular, se considera cuatro algoritmos de clasificación implementados en Weka (Witten y Frank, 2005; Frank et al., 2016; Weka, 2020, 2021), J48, naïve

Bayes, SMO y random forest, que se han descrito brevemente en la sección 2.3; y 11 conjuntos de datos del UCI Machine Learning Repository (Dua and Graff, 2019) que se han descrito en la sección 4.3.1.

El objetivo de este extenso análisis es comprobar si los resultados obtenidos en el experimento anterior se repiten para otros clasificadores y otros conjuntos de datos. El primer paso en todos los casos es seleccionar los 5 atributos más relevantes mediante el método de selección de características descrito en la sección 4.2.1. Los resultados se muestran en la tabla 4.8.

Tabla 4.8. Los cinco atributos más relevantes de cada conjunto de datos según la evaluación de atributos de la relación de ganancia (véase la sección 4.2.1).

#	Dataset	1º	2º	3º	4º	5º
1	Thyroid	hypopit.	pregnant	Psych	goitre	referral_ source
2	Healthy	C4	C3	C6	C5	C7
3	Avila	F5	F1	F9	F3	F7
4	Adult	Mar.Sta.	Relat.	Sex	Age	Educ
5	Nursery	F2	F1	F7	F5	F4
6	Bank	poutcome	contact	Housing	month	Loan
7	HTRU2	A3	A1	A4	A6	A5
8	Connect-4	g6	d3	f6	d2	b6
9	Tic-tac-toe	m-m-s	b-l-s	t-l-s	t-r-s	b-r-s
10	Credit	A9	A10	A4	A5	A6
11	Mushroom	odor	gill-size	stalk-surface-above-ring	spore-print-color	ring-type

A continuación, se aplican los cinco clasificadores con la selección de atributos de la tabla 4.8. Se calculan sus precisiones a partir de sus correspondientes matrices de confusión mediante validación cruzada. Las precisiones resultantes para cada clasificador y conjunto de datos se muestran en la tabla 4.9.

TABLA 4.9. Precisiones y errores absolutos medios de los cinco clasificadores y los 11 conjuntos de datos. Los resultados se presentan como precisión ± error medio absoluto.

#	Dataset	I	J48	SMO	Naïve Bayes	Random Forest
1	Thyroid	0.743 ± 0.257	0.744 ± 0.381	0.743 ± 0.257	0.741 ± 0.373	0.743 ± 0.374
2	Healthy	0.953 ± 0.024	0.963 ± 0.030	0.949 ± 0.255	0.935 ± 0.042	0.963 ± 0.028
3	Avila	0.653 ± 0.058	0.666 ± 0.074	0.600 ± 0.141	0.610 ± 0.087	0.657 ± 0.069
4	Adult	0.825 ± 0.175	0.824 ± 0.250	0.818 ± 0.182	0.763 ± 0.240	0.824 ± 0.236
5	Nursery	0.508 ± 0.197	0.548 ± 0.224	0.508 ± 0.265	0.531 ± 0.233	0.508 ± 0.224
6	Bank	0.885 ± 0.115	0.894 ± 0.186	0.893 ± 0.107	0.890 ± 0.175	0.893 ± 0.167
7	HTRU2	0.971 ± 0.029	0.971 ± 0.049	0.969 ± 0.031	0.969 ± 0.050	0.971 ± 0.048
8	Connect-4	0.658 ± 0.228	0.665 ± 0.313	0.665 ± 0.318	0.663 ± 0.318	0.665 ± 0.311
9	Tic-tac-toe	0.801 ± 0.193	0.794 ± 0.258	0.753 ± 0.247	0.753 ± 0.374	0.840 ± 0.209
10	Credit	0.859 ± 0.141	0.862 ± 0.220	0.858 ± 0.142	0.861 ± 0.193	0.848 ± 0.199
11	Mushroom	1.000 ± 0.000	1.000 ± 0.000	0.999 ± 0.001	0.999 ± 0.020	1.000 ± 0.000

En las tablas 4.10 y 4.11, se presentan los resultados obtenidos cuando se utilizan γ^X y γ^V como medida de rendimiento de la evaluación.

TABLA 4.10. Medida de evaluación γ^X para los cinco clasificadores y los 11 conjuntos de datos, y entropía de Shannon normalizada [0, 1] del atributo consecuente para cada conjunto de datos.

#	Dataset	Entropía	γ^X(*I*)	γ^X(*J48*)	γ^X *SMO*)	γ^X *NB*)	γ^X(*RF*)
1	Thyroid	0.829	0.335	0.338	0.335	0.330	0.335
2	Healthy	0.632	0.901	0.922	0.893	0.864	0.922
3	Avila	0.737	0.549	0.566	0.480	0.493	0.554
4	Adult	0.796	0.521	0.519	0.502	0.352	0.519
5	Nursery	0.739	0.279	0.338	0.279	0.313	0.279
6	Bank	0.521	0.443	0.487	0.482	0.468	0.482
7	HTRU2	0.442	0.826	0.826	0.814	0.814	0.826
8	Connect-4	0.769	0.312	0.326	0.326	0.322	0.326
9	Tic-tac-toe	0.931	0.561	0.545	0.455	0.455	0.647
10	Credit	0.991	0.715	0.721	0.713	0.719	0.692
11	Mushroom	0.999	1.000	1.000	0.998	0.998	1.000

TABLA 4.11. Medida de evaluación γ^V para los cinco clasificadores y los 11 conjuntos de datos, y la precisión del clasificador intuitivo V.

#	Dataset	acc(V)	$\gamma^V(I)$	$\gamma^V(J48)$	$\gamma^V(SMO)$	$\gamma^V(NB)$	$\gamma^V(RF)$
1	Thyroid	0.738	0.018	0.022	0.018	0.011	0.018
2	Healthy	0.686	0.850	0.882	0.838	0.793	0.882
3	Avila	0.411	0.411	0.433	0.321	0.338	0.418
4	Adult	0.759	0.273	0.269	0.244	0.016	0.269
5	Nursery	0.333	0.262	0.322	0.262	0.297	0.262
6	Bank	0.883	0.017	0.094	0.085	0.060	0.085
7	HTRU2	0.908	0.683	0.683	0.661	0.661	0.683
8	Connect-4	0.658	0.000	0.020	0.020	0.014	0.020
9	Tic-tac-toe	0.653	0.426	0.406	0.287	0.287	0.538
10	Credit	0.555	0.683	0.690	0.681	0.688	0.658
11	Mushroom	0.518	1.000	1.000	0.998	0.998	1.000

Como se ha mencionado antes, se sabe que la medida γ^X está estrechamente relacionada con las medidas $\varkappa$ y π. En las tablas 4.10 y 4.11, se observa que una mayor entropía en el atributo consecuente no significa un peor rendimiento de los clasificadores (Valverde-Albacete y Peláez-Moreno, 2014). Esto no es sorprendente, ya que todos los clasificadores utilizan no solo la información de la distribución de frecuencias del atributo consecuente, sino también la información proporcionada sobre él por los restantes atributos del conjunto de datos. Por lo tanto, parece adecuado utilizar la entropía de todo el conjunto de datos como referencia a la hora de evaluar el rendimiento de los clasificadores. Esta entropía es captada de alguna manera por el clasificador intuitivo I, como se ha explicado anteriormente. En la tabla 4.12, se presentan los resultados obtenidos cuando se utiliza Γ como medida de rendimiento de la evaluación.

TABLA 4.12. Medida de evaluación Γ para los cuatro clasificadores heurísticos y los 11 conjuntos de datos.

#	Dataset	Γ(*J*48)	Γ(*SMO*)	Γ(*NB*)	Γ(*RF*)
1	Thyroid	0.004	0.000	-0.003	0.000
2	Healthy	0.213	-0.004	-0.019	0.213
3	Avila	0.037	-0.081	-0.066	0.012
4	Adult	-0.001	-0.008	-0.075	-0.001
5	Nursery	0.081	0.000	0.047	0.000
6	Bank	0.078	0.070	0.043	0.070
7	HTRU2	0.000	-0.002	-0.002	0.000
8	Connect-4	0.020	0.020	0.015	0.020
9	Tic-tac-toe	-0.009	-0.060	-0.060	0.196
10	Credit	0.021	-0.001	0.014	-0.013
11	Mushroom	0.000	-0.001	-0.001	0.000

El clasificador intuitivo I tendrá mayor precisión cuanto menor sea la entropía condicional del atributo objetivo dado el conjunto de datos (o el subconjunto de atributos seleccionados si se realiza previamente una función de selección), por lo que será más difícil que un clasificador mejore significativamente los resultados de clasificación de este clasificador intuitivo. Por otro lado, es necesario destacar que la selección del mejor subconjunto de atributos ha sido relevante a lo largo del proceso de clasificación, ya que el método utilizado se basa en la reducción de la entropía. En este sentido, Γ mediría cuánto contribuye un clasificador al procedimiento completo de clasificación con respecto a lo que aporta el proceso de selección de atributos. Por tanto, Γ ofrece una información diferente a otras medidas de rendimiento de los clasificadores, que se consideran interesantes. El objetivo, por tanto, no es sustituir a cualquier medida de rendimiento conocida, sino proporcionar una medida de un aspecto diferente del rendimiento de un clasificador.

Finalmente, en las tablas 4.11 y 4.12, se observa que las medidas de rendimiento γ^V y Γ proporcionan información complementaria sobre los clasificadores. En la tabla 4.11, se puede observar cómo cada clasificador aprovecha la información proporcionada por los atributos del conjunto de datos para clasificar mejor el atributo objetivo, mientras que en la tabla 4.12 se puede observar cuánto mejor que el clasificador intuitivo I son los clasificadores capaces de utilizar la información del conjunto de datos para predecir correctamente las clases del atributo objetivo.

4.4. Discusión y Conclusiones

En el experimento se ha demostrado que tanto la selección de características como la entropía del atributo consecuente pueden ser relevantes para el resultado del rendimiento de un algoritmo de clasificación. Por lo tanto, parece interesante tener en cuenta la diversidad de la variable de respuesta o del conjunto de datos a la hora de evaluar un clasificador. Además, se observa el efecto de la entropía, en el sentido de que a menor entropía, mayor es la tasa de éxito en las clasificaciones, lo que parece intuitivamente razonable. Por otro lado, se observa en el experimento que la elección de un mayor número de características no siempre proporciona un mejor rendimiento del algoritmo de clasificación, por lo que este tipo de análisis es relevante a la hora de seleccionar un número adecuado de características, sobre todo cuando el algoritmo de selección de características no ha utilizado el algoritmo clasificador para la selección óptima. Un análisis riguroso de esto último puede encontrarse en Brown et al. (2012).

Las medidas de rendimiento de los clasificadores que solo utilizan los resultados del propio algoritmo de clasificación, como el ratio de aciertos (exactitud), no aportan realmente información sobre su capacidad real de clasificar correctamente respecto a los métodos no sofisticados. Por esta razón, es importante el uso de medidas relativas cuando se comparan con clasificadores simples de referencia, porque dan información sobre la relación entre la ganancia en la clasificación correcta de las instancias y el esfuerzo realizado en el diseño de nuevos clasificadores con respecto al uso de clasificadores simples e intuitivos, es decir, se puede evaluar mejor la mejora real proporcionada por el algoritmo de clasificación. Además, si el clasificador de referencia incorpora algún tipo de información adicional, como diferentes aspectos de la entropía de todo el conjunto de datos o el atributo consecuente, la información proporcionada por la medida de rendimiento será aún más relevante.

En este capítulo se han utilizado tres clasificadores simples, el clasificador aleatorio X, el clasificador intuitivo V y el clasificador intuitivo I. Los dos primeros utilizan simplemente la distribución del atributo consecuente para clasificar y se ha demostrado que están estrechamente relacionados con la entropía de ese atributo, mientras que el tercero utiliza

toda la distribución del conjunto de datos para clasificar y su rendimiento se aproxima a la entropía condicional del atributo consecuente dados los atributos restantes (o un subconjunto de atributos si se aplica previamente la selección de características) en el conjunto de datos. Estos tres clasificadores se han utilizado como referencia para introducir tres medidas del rendimiento de los clasificadores. Estas miden cuánto mejora (o empeora) un clasificador respecto a estos clasificadores simples que están relacionados con ciertos aspectos de la entropía del atributo consecuente dentro del conjunto de datos. Por lo tanto, son medidas que reflejan el rendimiento de los clasificadores heurísticos, teniendo en cuenta la entropía de alguna manera, y esto es importante, porque cuanto mayor es la entropía, mayor es la dificultad para clasificar correctamente, como se ha visto en el experimento, lo que da una mejor idea del verdadero rendimiento de un clasificador. Asimismo, las tres medidas de rendimiento de los clasificadores pueden interpretarse en términos de reducción proporcional del error de clasificación, lo que hace que estas medidas sean fácilmente comprensibles. En particular, γ^X está estrechamente relacionada con las conocidas medidas $\varkappa$ y π, y proporciona información sobre cuánto mejora un clasificador los resultados de clasificación en relación con un clasificador aleatorio que solo tiene en cuenta la información contenida en la distribución de frecuencias de las clases de atributos objetivo. γ^V proporciona información sobre cómo un clasificador es capaz de utilizar la información contenida en todo el conjunto de datos (o en un subconjunto del conjunto de datos) para mejorar los resultados de la clasificación en relación con un clasificador que solo utiliza la información de la distribución de frecuencias de las clases de atributos objetivo y siempre predice la clase más probable. Por último, Γ proporciona información sobre cuánto mejora un clasificador los resultados de la clasificación cuando utiliza una tecnología más elaborada de gestión de datos que el clasificador intuitivo I que simplemente predice la clase más probable dado un perfil particular de atributos en el conjunto de datos.

Para concluir, aunque los dos clasificadores intuitivos utilizados en este capítulo ya fueron descartados en favor de otros más modernos y sofisticados, se cree que siguen siendo útiles como clasificadores de referencia, ya que el aleatorio es comúnmente utilizado en el aprendizaje

automático, para luego diseñar medidas de rendimiento basadas en ellos que se ha mostrado a lo largo de este capítulo que proporcionan información relevante sobre el rendimiento de los clasificadores diferente a otras medidas de rendimiento.

4.5. Tablas

A continuación se muestra en las tablas 4A1 a 4A5 el número de veces que J48 y el clasificador intuitivo ganan para cada escenario.

Tabla 4.Al. Escenario SI, 3.200 filas, relación 3: 1 de valores positivos/negativos para la variable objetivo, 100 submuestras por combinación y las evaluaciones de atributos de relación de ganancia de las cinco mejores variables son 0.036, 0.037, 0.033, 0.034 y 0.029 (de más a menos relevante).

Comb.	Ant.	J48 gana	*I* gana	γ^{X} (*J*48)	Γ (J48)	γ^{X} (*I*)	γ^{V} (*J*48)	γ^{V} (*I*)
#1	5	45	10	0.3328	0.0019	0.3316	-0.0003	-0.0009
#2	4	39	33	0.3326	0.0003	0.3324	-0.0004	-0.0005
#3	45	56	28	0.3320	0.0022	0.3306	-0.0007	-0.0014
#4	3	39	16	0.3318	0.0024	0.3301	-0.0008	-0.0016
#5	35	55	33	0.3314	0.0031	0.3294	-0.0010	-0.0020
#6	34	57	30	0.3311	0.0029	0.3291	-0.0011	-0.0021
#7	345	56	34	0.3305	0.0032	0.3284	-0.0014	-0.0025
#8	2	0	0	0.3333	0.0000	0.3333	0.0000	0.0000
#9	25	45	10	0.3328	0.0019	0.3316	-0.0003	-0.0009
#10	24	44	28	0.3326	0.0005	0.3322	-0.0004	-0.0006
#11	245	61	26	0.3321	0.0025	0.3304	-0.0006	-0.0015
#12	23	47	24	0.3316	0.0021	0.3302	-0.0009	-0.0016
#13	235	55	36	0.3312	0.0027	0.3294	-0.0011	-0.0020
#14	231	56	32	0.3307	0.0024	0.3291	-0.0013	-0.0021
#15	2345	58	33	0.3303	0.0030	0.3282	-0.0015	-0.0025
#16	1	0	0	0.3333	0.0000	0.3333	0.0000	0.0000
#17	15	45	10	0.3328	0.0019	0.3316	-0.0002	-0.0009
#18	14	40	32	0.3326	0.0004	0.3324	-0.0004	-0.0005
#19	145	57	27	0.3321	0.0023	0.3306	-0.0006	-0.0014
#20	13	39	16	0.3318	0.0024	0.3301	-0.0008	-0.0016
#21	135	53	33	0.3312	0.0028	0.3294	-0.0011	-0.0020
#22	134	55	31	0.3310	0.0028	0.3291	-0.0012	-0.0021
#23	1345	58	33	0.3305	0.0032	0.3284	-0.0014	-0.0025
#24	12	0	0	0.3333	0.0000	0.3333	0.0000	0.0000
#25	125	45	10	0.3328	0.0019	0.3316	-0.0003	-0.0009
#26	124	44	28	0.3327	0.0007	0.3322	-0.0003	-0.0006
#27	1245	62	25	0.3321	0.0025	0.3304	-0.0006	-0.0015
#28	123	47	24	0.3316	0.0022	0.3302	-0.0009	-0.0016
#29	1235	55	35	0.3311	0.0026	0.3294	-0.0011	-0.0020
#30	1234	57	31	0.3308	0.0026	0.3291	-0.0013	-0.0021
#31	12345	57	34	0.3303	0.0031	0.3282	-0.0015	-0.0025
	Total	1427	742					
	%	46.03	23.94					

Como se observa en la tabla 4.A1 se trata del primer escenario en el que hay un total de 3,200 filas, de las cuales 2,400 filas son positivas y 800 filas son negativas, es decir la relación es 3:1 para la variable objetivo.

TABLA 4.A2. Escenario *S*2, 3,600 filas, relación 2:1 de valores positivos/negativos para la variable objetivo, 100 submuestras por combinación, y las evaluaciones de atributos de la relación de ganancia de las cinco mejores variables son 0.050, 0.037, 0.034, 0.032 y 0.022 (de más a menos relevante).

Comb.	**Ant.**	**J48 gana**	***I* gana**	γ^X (*J*48)	**Γ (J48)**	γ^X (*I*)	γ^V (*J*48)	γ^V (*I*)
#1	5	38	35	0.2499	0.0000	0.2498	-0.0001	-0.0001
#2	4	7	34	0.2526	-0.0004	0.2532	0.0034	0.0043
#3	45	35	45	0.2525	-0.0003	0.2530	0.0034	0.0041
#4	3	1	12	0.2611	-0.0004	0.2617	0.0148	0.0156
#5	35	43	34	0.2604	-0.0002	0.2607	0.0139	0.0143
#6	34	6	41	0.2636	-0.0008	0.2649	0.0182	0.0198
#7	345	33	54	0.2625	-0.0008	0.2638	0.0167	0.0184
#8	2	1	0	0.2500	0.0000	0.2500	0.0000	0.0000
#9	25	38	35	0.2499	0.0000	0.2498	-0.0001	-0.0001
#10	24	31	34	0.2524	-0.0003	0.2529	0.0032	0.0038
#11	245	41	46	0.2523	-0.0003	0.2527	0.0030	0.0036
#12	23	9	46	0.2612	-0.0007	0.2623	0.0150	0.0163
#13	235	37	48	0.2605	-0.0005	0.2612	0.0140	0.0150
#14	231	26	59	0.2636	-0.0009	0.2650	0.0181	0.0200
#15	2345	34	57	0.2628	-0.0008	0.2640	0.0171	0.0187
#16	1	0	0	0.2500	0.0000	0.2500	0.0000	0.0000
#17	15	38	35	0.2499	0.0000	0.2498	-0.0001	-0.0001
#18	14	7	34	0.2525	-0.0004	0.2532	0.0034	0.0043
#19	145	33	47	0.2524	-0.0004	0.2530	0.0032	0.0041
#20	13	1	12	0.2612	-0.0004	0.2617	0.0149	0.0156
#21	135	43	34	0.2605	-0.0002	0.2607	0.0140	0.0143
#22	134	6	41	0.2636	-0.0008	0.2649	0.0182	0.0198
#23	1345	36	51	0.2628	-0.0007	0.2638	0.0170	0.0184
#24	12	0	0	0.2500	0.0000	0.2500	0.0000	0.0000
#25	125	38	35	0.2499	0.0001	0.2498	-0.0001	-0.0001
#26	124	31	34	0.2524	-0.0003	0.2529	0.0032	0.0038
#27	1245	41	46	0.2523	-0.0002	0.2527	0.0031	0.0036
#28	123	9	45	0.2612	-0.0007	0.2623	0.0150	0.0163
#29	1235	37	48	0.2605	-0.0005	0.2612	0.0140	0.0150
#30	1234	25	60	0.2636	-0.0009	0.2650	0.0182	0.0200
#31	12345	34	57	0.2628	-0.0008	0.2640	0.0170	0.0187
	Total	759	1159					
	%	24.48	37.39					

Como se observa en la tabla 4.A2 se trata del segundo escenario en el que hay un total de 3,600 filas, de las cuales 2,400 filas son positivas y 1,200 filas son negativas, es decir la relación es 2:1 para la variable objetivo.

TABLA 4.A3. Escenario *S*3, 4,800 filas, relación 1:1 de valores positivos/negativos para la variable objetivo, 100 submuestras por combinación, y las evaluaciones de atributos de la relación de ganancia de las cinco mejores variables son 0.083, 0.082, 0.028, 0.028 y 0.026 (de más a menos relevante).

Comb.	Anteced.	J48 gana	*I* gana	γ^X (*J*48)	Γ (*J*48)	γ^X (*I*)	γ^V (*J*48)	γ^V (*I*)
#1	5	72	1	0.0365	0.0390	-0.0026	0.0365	-0.0026
#2	4	100	0	0.0076	0.0839	-0.0833	0.0076	-0.0833
#3	45	33	0	0.0448	0.0173	0.0279	0.0448	0.0279
#4	3	100	0	0.0067	0.0842	-0.0846	0.0067	-0.0846
#5	35	49	4	0.0417	0.0259	0.0161	0.0417	0.0161
#6	34	100	0	0.0140	0.0833	-0.0756	0.0140	-0.0756
#7	345	18	5	0.0492	0.0076	0.0419	0.0492	0.0419
#8	2	18	0	0.0323	0.0070	0.0255	0.0323	0.0255
#9	25	100	0	0.0343	0.0797	-0.0493	0.0343	-0.0493
#10	24	60	11	0.0324	0.0253	0.0074	0.0324	0.0074
#11	245	100	0	0.0436	0.0806	-0.0402	0.0436	-0.0402
#12	23	57	0	0.0323	0.0235	0.0090	0.0323	0.0090
#13	235	100	0	0.0399	0.0805	-0.0441	0.0399	-0.0441
#14	231	86	2	0.0324	0.0470	-0.0153	0.0324	-0.0153
#15	2345	99	0	0.0487	0.0781	-0.0319	0.0487	-0.0319
#16	1	100	0	0.0004	0.0832	-0.0903	0.0004	-0.0903
#17	15	77	0	0.0372	0.0433	-0.0064	0.0372	-0.0064
#18	14	100	0	0.0075	0.0838	-0.0832	0.0075	-0.0832
#19	145	37	0	0.0448	0.0190	0.0262	0.0448	0.0262
#20	13	100	0	0.0071	0.0844	-0.0845	0.0071	-0.0845
#21	135	50	4	0.0417	0.0274	0.0147	0.0417	0.0147
#22	134	100	0	0.0140	0.0832	-0.0755	0.0140	-0.0755
#23	1345	19	5	0.0492	0.0081	0.0414	0.0492	0.0414
#24	12	15	45	0.0325	0.0049	0.0277	0.0325	0.0277
#25	125	100	0	0.0345	0.0795	-0.0489	0.0345	-0.0489
#26	124	55	28	0.0328	0.0216	0.0115	0.0328	0.0115
#27	1245	100	0	0.0436	0.0811	-0.0407	0.0436	-0.0407
#28	123	49	23	0.0327	0.0192	0.0138	0.0327	0.0138
#29	1235	100	0	0.0399	0.0807	-0.0443	0.0399	-0.0443
#30	1234	84	10	0.0325	0.0437	-0.0117	0.0325	-0.0117
#31	12345	100	0	0.0482	0.0786	-0.0331	0.0482	-0.0331
	Total	2278	138					
	%	73.48	4.45					

En relación a la tabla 4.A3, esta muestra el tercer escenario en el que hay un total de 4,800 filas, de las cuales 2,400 filas son positivas y 2,400 filas son negativas, es decir la relación es 1:1 (escenario balanceado) para la variable objetivo.

TABLA 4.A4. Escenario $S4$, 6,000 filas, relación 1:2 de valores positivos/negativos para la variable objetivo, 100 submuestras por combinación, y las evaluaciones de atributos de la relación de ganancia de las cinco mejores variables son 0.122, 0.076, 0.02, 0.015 y 0.013 (de más a menos relevante).

Comb.	Anteced.	J48 gana	*I* gana	γ^X (J48)	Γ (J48)	γ^X (*I*)	γ^V (*J*48)	γ^V (*I*)
#1	5	10	19	0.2652	-0.0004	0.2659	0.0203	0.0212
#2	4	0	0	0.2500	0.0000	0.2500	0.0000	0.0000
#3	45	13	20	0.2653	-0.0004	0.2658	0.0204	0.0211
#4	3	0	0	0.2500	0.0000	0.2500	0.0000	0.0000
#5	35	28	18	0.2651	-0.0003	0.2655	0.0202	0.0207
#6	34	0	0	0.2500	0.0000	0.2500	0.0000	0.0000
#7	345	30	20	0.2651	-0.0002	0.2655	0.0201	0.0206
#8	2	0	0	0.2686	0.0000	0.2686	0.0248	0.0248
#9	25	23	76	0.2686	-0.0022	0.2720	0.0248	0.0293
#10	24	0	38	0.2692	-0.0002	0.2695	0.0256	0.0260
#11	245	21	78	0.2691	-0.0025	0.2728	0.0254	0.0305
#12	23	76	8	0.2686	0.0004	0.2684	0.0248	0.0245
#13	235	23	76	0.2685	-0.0020	0.2715	0.0247	0.0287
#14	231	46	40	0.2692	0.0000	0.2692	0.0256	0.0256
#15	2345	24	76	0.2690	-0.0022	0.2723	0.0254	0.0298
#16	1	0	47	0.2501	-0.0002	0.2505	0.0001	0.0006
#17	15	8	28	0.2653	-0.0004	0.2660	0.0204	0.0213
#18	14	0	47	0.2501	-0.0002	0.2505	0.0001	0.0006
#19	145	11	30	0.2653	-0.0004	0.2659	0.0204	0.0212
#20	13	0	47	0.2501	-0.0002	0.2505	0.0001	0.0006
#21	135	24	25	0.2651	-0.0003	0.2656	0.0202	0.0208
#22	134	0	47	0.2501	-0.0002	0.2505	0.0001	0.0006
#23	1345	27	27	0.2651	-0.0003	0.2656	0.0202	0.0207
#24	12	0	47	0.2687	-0.0002	0.2691	0.0250	0.0254
#25	125	21	76	0.2686	-0.0023	0.2721	0.0247	0.0295
#26	124	0	69	0.2693	-0.0004	0.2699	0.0257	0.0266
#27	1245	18	80	0.2689	-0.0026	0.2729	0.0253	0.0306
#28	123	41	47	0.2687	-0.0001	0.2688	0.0250	0.0251
#29	1235	22	76	0.2685	-0.0020	0.2716	0.0247	0.0288
#30	1234	24	69	0.2693	-0.0002	0.2697	0.0257	0.0262
#31	12345	22	77	0.2691	-0.0022	0.2724	0.0254	0.0299
	Total	512	1308					
	%	16.52	42.19					

En vista de la tabla 4.A4 se puede observar el cuarto escenario en el que hay un total de 6,000 filas de las cuales 2,000 filas son positivas y 4,000 filas son negativas, es decir la relación es 1:2 para la variable objetivo.

TABLA 4.A5. Escenario *S*5, entre 7,820 y 7,940 filas, relación 1:3 de valores positivos/negativos para la variable objetivo, 100 submuestras por combinación, y las evaluaciones de atributos de la relación de ganancia de las cinco mejores variables son 0.102, 0.134, 0.016, 0.013 y 0.010 (de más a menos relevante).

Comb.	Ant.	J48 gana	*I* gana	γ^X (*J*48)	Γ(*J*48)	γ^X(*I*)	γ^V(*J*48)	γ^V(*I*)
#1	5	59	41	0.3336	0.0013	0.3327	0.0191	0.0179
#2	4	81	0	0.3339	0.0016	0.3328	0.0181	0.0165
#3	45	59	41	0.3234	0.0012	0.3226	-0.0003	-0.0007
#4	3	81	0	0.3288	0.0016	0.3277	0.0090	0.0074
#5	35	59	41	0.3209	0.0012	0.3201	-0.0022	-0.0026
#6	34	81	0	0.3187	0.0016	0.3176	-0.0037	-0.0043
#7	345	59	41	0.3310	0.0012	0.3302	0.0141	0.0129
#8	2	73	10	0.3187	0.0012	0.3179	-0.0039	-0.0043
#9	25	59	41	0.3234	0.0012	0.3226	-0.0003	-0.0007
#10	24	73	10	0.3338	0.0012	0.3330	0.0190	0.0178
#11	245	59	41	0.3209	0.0012	0.3201	-0.0020	-0.0024
#12	23	74	9	0.3263	0.0012	0.3254	0.0041	0.0028
#13	235	59	41	0.3109	0.0012	0.3101	-0.0087	-0.0091
#14	231	74	9	0.3313	0.0013	0.3305	0.0146	0.0133
#15	2345	59	41	0.3234	0.0012	0.3226	-0.0006	-0.0010
#16	1	80	1	0.3462	0.0015	0.3452	0.0377	0.0363
#17	15	89	10	0.3331	0.0023	0.3316	0.0112	0.0089
#18	14	93	1	0.3438	0.0020	0.3425	0.0319	0.0300
#19	145	89	9	0.3384	0.0026	0.3366	0.0220	0.0194
#20	13	75	4	0.3424	0.0014	0.3414	0.0291	0.0277
#21	135	93	5	0.3397	0.0026	0.3380	0.0239	0.0214
#22	134	89	4	0.3246	0.0016	0.3235	-0.0018	-0.0024
#23	1345	95	3	0.3398	0.0027	0.3380	0.0240	0.0214
#24	12	74	9	0.3541	0.0013	0.3532	0.0524	0.0512
#25	125	89	8	0.3232	0.0024	0.3215	-0.0026	-0.0034
#26	124	84	9	0.3336	0.0015	0.3326	0.0124	0.0109
#27	1245	89	8	0.3308	0.0026	0.3290	0.0070	0.0044
#28	123	70	13	0.3347	0.0010	0.3341	0.0141	0.0131
#29	1235	91	5	0.3346	0.0025	0.3330	0.0144	0.0119
#30	1234	81	11	0.3398	0.0013	0.3390	0.0236	0.0223
#31	12345	94	4	0.3447	0.0025	0.3431	0.0343	0.0319
	Total	2384	470					
	%	76.90	15.16					

Como se observa en la tabla 4.A5 se trata del quinto y último escenario en el que hay un total ligeramente inferior a 8,000 filas, de las cuales

aproximadamente 2,000 filas son positivas y en torno a 6,000 filas son negativas, es decir la relación es 1:3 para la variable objetivo.

Capítulo 5. Análisis comparado de los métodos de selección de características basados en teoría de juegos

En este capítulo se realiza una revisión del estado del arte de la aplicación de la teoría de juegos, en particular del conocido valor de Shapley, en el problema de clasificación en aprendizaje automático. Asimismo, estudia el desempeño de dos conceptos de la teoría de juegos, el valor de Shapley y el valor de Banzhaf, como técnicas para la selección de características comparándolos con distintos métodos de selección de características implementados en Weka sobre una serie de varios conjuntos de datos.

5.1. Introducción

Como ya se ha expuesto en el capítulo 1, una de las dificultades más frecuentes que se encuentran en el análisis de un conjunto de datos es la alta dimensionalidad, ya que cuando hay demasiadas variables el análisis es más difícil y costoso computacionalmente, puede haber variables correlacionadas, variables redundantes o incluso variables ruidosas. Todos estos problemas pueden conducir a un peor rendimiento de los

clasificadores. Por ello, para resolver estas dificultades, se suele utilizar una de las dos alternativas siguientes (1) reducir la dimensión transformando los datos, o (2) seleccionar un subconjunto de características manteniendo la mayor parte de la información del conjunto de datos; este enfoque se conoce como selección de características. Por ejemplo, en el capítulo 3 se comparan el análisis discriminante lineal y el método de selección de características RBS. Una ventaja del enfoque de selección de características es que se mantiene el significado original de las variables. En los problemas de clasificación, en los que hay una variable objetivo nominal (el consecuente), la selección de las variables más relevantes no es una cuestión trivial. La cuestión de la selección de características ya se ha abordado en muchos estudios en el campo del aprendizaje automático utilizando diferentes enfoques (Fu y Cardillo, 1967; Cardillo y Fu, 1967; Chien, 1969; Jurs et al., 1969; Jurs, 1970; Narendra y Fukunaga, 1977; Pudil et al., 1994; Siedlecki y Sklansky, 1989; Leardi et al., 1992; Yang y Honavar, 1998; John et al., 1994; Kohavi y John, 1997; Mitra et al., 2002; Yu y Liu, 2004; Peng et al., 2005; Trabelsia et al., 2017; Meddouri et al., 2014; Cohen et al., 2007; Afghah et al., 2018; Duch et al., 2004; Aremu et al., 2020; Bai et al., 2020; Qu et al., 2020; Revanasiddappa y Harish, 2018; Zhao et al., 2020). Liu y Yu (2005) revisaron los algoritmos de selección de características para la clasificación y la agrupación, y los clasificaron para facilitar la elección del algoritmo más adecuado para el análisis de un conjunto de datos concreto.

Chandrashekar y Sahin (2014), y Miao y Niu (2016) clasifican los métodos de selección de características de acuerdo con la estrategia de búsqueda en tres grupos:

1. Los *métodos de filtrado* ("*filter methods*") utilizan técnicas de ranking de variables como criterio principal para la selección de atributos. Los métodos de ranking se utilizan debido a su simplicidad y se obtienen buenos resultados en aplicaciones prácticas. Un criterio de ranking adecuado se utiliza para puntuar los atributos y se fija un umbral para eliminar los atributos por debajo del mismo. Los métodos de ranking son filtros ya que se aplican antes de la clasificación para filtrar los atributos menos relevantes.
2. Los *métodos de envoltura* ("*wrapper methods*") usan el predictor como una caja negra y el desempeño del predictor como la función

objetivo para evaluar el subconjunto de atributos. Dado que evaluar 2^{C-1} subconjuntos se convierte en un problema NP-duro, los subconjuntos subóptimos se encuentran empleando algoritmos heurísticos de búsqueda. Se pueden utilizar varios algoritmos de búsqueda para encontrar un subconjunto de atributos que maximice la función objetivo que es el rendimiento de la clasificación.

3. Los *métodos embebidos* ("*embedded methods*") buscan reducir el tiempo de cálculo necesario para reclasificar diferentes subconjuntos, que es lo que se hace en los métodos de envoltura. El enfoque principal es incorporar la selección de características como parte del proceso de entrenamiento.

Un número importante de los procedimientos de selección de características incorporan el uso de su propio clasificador para medir la calidad de la selección, de manera que en bastantes ocasiones se puede identificar el método de selección de características con el propio clasificador, como en los métodos de selección de características de envoltura y embebidos.

En los últimos 20 años han aparecido múltiples aplicaciones de la teoría de juegos al problema de clasificación, en las siguientes líneas se hace una breve revisión que permite hacerse una mejor idea sobre el estado del arte en esta temática. Keinan et al. (2004, 2006) introducen el *análisis del valor de Shapley de múltiples perturbaciones* (MSA) y estudian el interés de utilizar el valor de Shapley (Shapley, 1953) para medir la contribución de distintos elementos en el desempeño de redes artificiales y biológicas, en comparación con el análisis funcional de contribuciones (Aharonov et al., 2003). Dada la complejidad computacional del valor de Shapley, utilizan muestreo con reemplazamiento sobre todos los órdenes posibles para su cálculo aproximado cuando es inabordable el cálculo exacto. Asimismo, también sugieren que otros valores de la teoría de juegos como el valor de Banzhaf (1965) podrían ser utilizados. Cohen et al. (2005, 2007) introducen y estudian un algoritmo para la selección de características en problemas de clasificación basado en el MSA. Strumbelj y Kononenko (2010, 2014) proponen utilizar el valor de Shapley para explicar la contribución del conocimiento del valor de los atributos en el resultado de las predicciones individuales en los modelos de clasificación y predicción.

Como en los trabajos antes mencionados, también utilizan la aproximación por muestreo para el cálculo del valor de Shapley. Lundberg y Lee (2017) presentan un enfoque que unifica seis métodos de interpretación de la relevancia de las características en una determinada predicción haciendo uso del valor de Shapley de unos juegos cooperativos adecuados asociados a los problemas de predicción. Zaeri-Amirani et al. (2018) y Afghah et al. (2018) proponen utilizar la teoría de juegos para la selección de características en el problema de detección de falsas alarmas en las UCI, el primero haciendo uso del valor de Shapley y el segundo utilizando el valor de Banzhaf (Banzhaf, 1965).

Más recientemente, Aas et al. (2020) extienden el análisis de la influencia de las características en predicciones individuales utilizando el valor de Shapley cuando las características son dependientes. Chu y Chan (2020) proponen el uso del valor de Shapley para la selección de características pero descomponiéndolo en un mayor número de interacciones para medir mejor la contribución de una combinación específica de características. Tripathi et al. (2020) plantean analizar la interacción entre las características en problemas binarios de clasificación mediante el valor de Shapley. Tripathi et al. (2021) proponen un método para la selección de un subconjunto de características basado en el valor de Shapley que es transparente el proceso, tiene relación con la tarea final y justifica la relevancia de las características. Guha et al. (2021) presentan un algoritmo genético cooperativo que haciendo uso de su valor de Shapley selecciona las mejores características para diseñar modelos de reconocimiento visual humano. Jothi et al. (2021) utilizan el valor de Shapley para la selección de características para construir modelos de clasificación de datos de salud mental. Davila-Pena et al. (2022) proponen una medida de la influencia de las características en los problemas de clasificación basada en el valor de Shapley, la caracterizan axiomáticamente y la aplican a un problema de la COVID-19.

En contraste a todos los trabajos que muestran el interés del valor de Shapley en el análisis de las características en los modelos de clasificación y predicción, Ma y Tourani (2020) muestran que eliminar una variable con un valor de Shapley alto de un modelo no siempre perjudica el desempeño, mientras que eliminar una variable con un valor de Shapley bajo de un modelo podría afectar negativamente el desempeño. Por lo tanto, el uso

del valor de Shapley para la selección de características no siempre da como resultado el modelo más parsimonioso y óptimamente predictivo en general. Así, el valor de Shapley de una variable no siempre refleja su relación causal con el objetivo de interés. Kumar et al. (2020) también muestran que el uso del valor de Shapley no siempre responde a lo que se esperaría que sea una explicación ni su interpretación siempre es clara. Por tanto, es necesario seguir trabajando en esta línea para encontrar en qué casos el valor de Shapley sí que es una medida relevante del impacto de las características en el desempeño de los modelos de clasificación y predicción y su interpretación es razonable.

Finalmente, Fryer et al. (2021) realizan una buena revisión de la literatura sobre el uso del valor de Shapley en la selección de características en los problemas de clasificación e interpretan en el contexto de estos problemas el significado de los axiomas que caracterizan el valor de Shapley.

5.2. Un poco de teoría de juegos. El valor de Shapley y el valor de Banzhaf

Un juego cooperativo de utilidad transferible es un par (N, v), donde $N = \{1,2, \dots, n\}$ es el conjunto de jugadores y v es la función característica definida de 2^N (el conjunto de todos los subconjuntos de N) en $\mathbb{R}$ con $v(\emptyset) = 0$. El conjunto de todos los juegos cooperativos de utilidad transferible con conjunto de jugadores N se denota por G^N. En este caso, el conjunto de jugadores serán los atributos antecedentes en el conjunto de datos y la función característica será alguna medida que evalúa el interés o la capacidad predictiva del modelo que se obtiene para cada subconjunto de atributos.

Dado un juego (N, v), una *distribución entre los jugadores* es un vector $z \in \mathbb{R}^N$ tal que $\sum_{i \in N} z_i \leq v(N)$. Una distribución se dice *eficiente* si satisface que $\sum_{i \in N} z_i = v(N)$. Un *valor* para G^N es una función $\rho: G^N \to \mathbb{R}^N$ que a cada juego (N, v) le asigna una distribución entre los jugadores $\rho(N, v)$.

Un concepto interesante en teoría de juegos cooperativos es el de *contribución marginal*. La contribución marginal de un jugador $i \in N$ a una

coalición $S \subset N$ es el valor que este jugador aporta a la coalición, es decir, en términos matemáticos:

$$M_i(S) = v(S \cup \{i\}) - v(S). \quad (5.1)$$

Este concepto también resulta interesante en términos de la selección de características puesto que mide la mejora (o eventualmente el empeoramiento) en la evaluación del modelo de utilizar el atributo i a no utilizarlo cuando ya se tienen en el modelo los atributos en S.

Dos valores basados en las contribuciones marginales son el *valor de Shapley* (Shapley, 1953) y el *valor de Banzhaf* (1965). El valor de Shapley de un juego $(N, v) \in G^N$ se define como sigue:

$$\Phi_i(N, v) = \sum_{S \subset N \setminus \{i\}} \frac{|S|!(|N|-|S|-1)!}{|N|!} M_i(S), \forall i \in N. \quad (5.2)$$

El valor de Shapley puede interpretarse como la contribución esperada de un jugador cuando la probabilidad de que se forme una coalición de cualquier tamaño es la misma y todas las coaliciones del mismo tamaño son equiprobables. Si se considera que todas las coaliciones son equiprobables, independientemente de su tamaño, entonces se tiene el valor de Banzhaf de un juego $(N, v) \in G^N$ que se define de la siguiente manera:

$$\beta_i(N, v) = \sum_{S \subset N \setminus \{i\}} \frac{1}{2^{|N|-1}} M_i(S), \forall i \in N. \quad (5.3)$$

Una forma diferente de interpretar el valor de Shapley es considerando que los jugadores son aleatoriamente ordenados y cada uno de ellos obtienen su contribución marginal de acuerdo a dicho orden. Entonces el valor esperado para todos los posibles órdenes es el valor de Shapley. Formalmente, dado un juego $(N, v) \in G^N$, si Σ^N es el conjunto de todos los posibles ordenamientos de los jugadores, entonces el valor de Shapley está definido como

$$\Phi_i(N, v) = \frac{1}{|N|!} \sum_{\sigma \in \Sigma} [v(P(\sigma, i) \cup \{i\}) - v(P(\sigma, i))], \forall i \in N, \quad (5.4)$$

donde $P(\sigma, i)$ es el conjunto de predecesores de i en el orden σ.

Se puede observar que el valor de Shapley y el valor de Banzhaf están basados en el mismo concepto, las *contribuciones marginales*, y la única diferencia se encuentra en la evaluación de la probabilidad con la que se puede formar una u otra coalición, por tanto, pueden considerarse otros

valores con la misma estructura pero con diferentes probabilidades y que se denominan valores probabilísticos (Weber, 1988) y que se definen para cada juego $(N, v) \in G^N$ como

$$\varphi_i(N, v) = \sum_{S \subset N \setminus \{i\}} p^i(S)\, M_i(S), \forall i \in N, \tag{5.5}$$

donde cada $p^i, i \in N,$ es una distribución de probabilidad definida sobre las coaliciones que no contienen al jugador i. Obviamente, el valor de Shapley y el valor de Banzhaf son dos casos particulares de valores probabilísticos.

El valor de Shapley ha sido muy estudiado en la literatura por las buenas propiedades matemáticas que satisface, una de ellas es que es siempre una distribución eficiente. Además, todas estas propiedades tienen relación con diversos principios de equidad, justicia, imparcialidad, consistencia o monotonía, entre otras. En Roth (1988) y Algaba et al. (2019a), se pueden encontrar numerosos ejemplos de las propiedades que el valor de Shapley satisface y sus aplicaciones a campos muy diferentes. Algaba et al. (2019b) hace una breve revisión sobre el valor de Shapley, sus propiedades y sus extensiones, intentando dar una visión de por qué el valor de Shapley es considerado como un paradigma de equidad. Por otra parte, el valor de Banzhaf no ha recibido tanta atención desde que fue introducido en detalle por Banzhaf en 1965. No obstante, en los últimos años se puede encontrar un número creciente de trabajos sobre este valor.

Quizás, el mayor inconveniente de las fórmulas presentadas anteriormente para el cálculo del valor de Shapley y el valor de Banzhaf, y para los valores probabilísticos en general, se puede observar en sus propias expresiones matemáticas, ya que es necesario conocer el valor de cada coalición y este puede ser un problema inabordable computacionalmente cuando el número de jugadores es grande. Por ello, una alternativa es recurrir a su cálculo aproximado por técnicas de muestreo (véanse, por ejemplo, Mann y Shapley, 1960; Keinan et al., 2004, 2006; Castro et al., 2009, 2017; Maleki et al., 2014). En este capítulo se apuesta por recurrir a técnicas de muestreo para el cálculo aproximado de los valores de Shapley y Banzhaf como se detallará en las secciones siguientes.

5.3. Algoritmos de selección de características basados en el valor de Shapley y en el valor de Banzhaf

Dado un conjunto de datos con C atributos $(A_1,\ldots,A_C)$, para el que se quiere construir un modelo que sea capaz de predecir el resultado del atributo A_C (consecuente) a partir del conocimiento del resto de atributos $A_1,\ldots,A_{C-1}$ (antecedentes), se desea conocer qué impacto tiene cada uno de los atributos antecedentes en la capacidad predictiva del valor del atributo consecuente, de forma que se pueda seleccionar un número reducido de atributos para formular un modelo con un buen desempeño. Si solo se considera la posibilidad de que todos los atributos formen parte del modelo de clasificación, se está ante un problema de atribución (véase, por ejemplo, Algaba et al., 2019c) en el que se desea evaluar cuánto de la capacidad predictiva es atribuible a cada atributo. La atribución de relevancia a través de los valores de Shapley proporciona información valiosa sobre el resultado de modelos complejos que, de otro modo, serían difíciles de interpretar. Y en el contexto del aprendizaje automático, se interpretarían como el valor explicativo de cada atributo. En este sentido, este problema podría ser útil para establecer un ranking entre los atributos que indicará qué atributos son más relevantes o más explicativos para la construcción de un buen modelo de clasificación. Una forma de abordar este tipo de problemas de atribución es desde la perspectiva de la teoría de juegos, pero para ello, el primer paso es definir un juego asociado al problema, después calcular un valor del juego y, finalmente, seleccionar aquellos atributos con un mayor valor. Estos pasos son estándar y se recogen en Fryer et al. (2021) como la estructura básica de un algoritmo de selección de atributos utilizando, en particular, el valor de Shapley.

Siguiendo el esquema anterior, el primer paso es definir la función característica del juego de atribución de relevancia de las características en el problema de clasificación. En este caso, parece lógico que la función característica refleje el desempeño del modelo de clasificación con ese subconjunto de características. Para ello, es necesario elegir dos elementos: una medida de desempeño y un clasificador. En este capítulo se elige como medida de desempeño la precisión (véase la ecuación (4.4)) y como clasificador el algoritmo intuitivo I (véanse los algoritmos 4.1 y 4.2), definido en el capítulo 4. Formalmente,

Definición 5.1. *Dado un problema de clasificación con conjunto de atributos antecedentes* F *y consecuente* A_C*, y dado conjunto de datos* D*, se define el* ***juego cooperativo de atribución de relevancia de las características en el problema de clasificación****,* (F, v)*, como*

$$v(S) = precisión\big(I(D)\big), \forall S \subset F. \tag{5.6}$$

Obsérvese que para calcular la precisión de un modelo de clasificación se necesita distinguir dos conjuntos de datos o dividir el conjunto de datos en dos bloques, uno de entrenamiento para construir el modelo de clasificación con el clasificador elegido y otro de validación con el que se calcula la precisión del modelo construido. Una alternativa a lo anterior para evitar posibles sesgos en la división del conjunto de datos es utilizar la validación cruzada. En este capítulo se utilizará este último procedimiento de validación. En Cohen et al. (2005, 2007) se introducen juegos análogos utilizando también como medida de desempeño la precisión y como método para su cálculo la validación cruzada, pero como clasificadores se utilizan el C4.5 (Quilan, 1992), naïve Bayes y 1-NN (Cover y Hart, 1967; Dasarathy, 1991).

Una vez definido el juego para evaluar la relevancia de cada uno de los atributos en la capacidad predictiva del modelo de clasificación se podría utilizar cualquier valor para juegos cooperativos. En la literatura casi todos los trabajos utilizan el valor de Shapley, pero en este capítulo se utilizará tanto el valor de Shapley como el valor de Banzhaf. Por supuesto, si se considera que el concepto de contribuciones marginales es adecuado para medir el impacto de un atributo en la capacidad predictiva de un modelo de clasificación, otros valores probabilísticos podrían ser utilizados. En este punto, cabe plantearse qué distribuciones de probabilidad sobre las coaliciones podrían ser más adecuadas, quizás, esto sea una cuestión relevante a tener en cuenta para futuras investigaciones. En este sentido, el hecho de utilizar tanto el valor de Shapley como el valor de Banzhaf es un primer paso en esta dirección.

Como se ha apuntado en la sección anterior, el mayor problema de los valores probabilísticos es su cálculo, puesto que en general es un problema NP-duro (Deng y Papadimitriou, 1994), por ello es necesario recurrir a aproximaciones basadas en muestreo (véanse, por ejemplo, Mann y Shapley, 1960; Keinan et al., 2004, 2006; Castro et al., 2009, 2017; Maleki

et al., 2014). En casi toda la literatura relacionada con la aplicación del valor de Shapley para la selección de características o la atribución de relevancia en problemas de aprendizaje automático, se recurre a estas aproximaciones, quedando las diferencias entre ellos reducidas a la técnica de muestreo utilizada y/o al tamaño de los subconjuntos de atributos que se seleccionan. En este capítulo se utilizará el muestro aleatorio simple tanto para seleccionar permutaciones, en el caso del cálculo del valor de Shapley, como para seleccionar coaliciones, en el caso del cálculo del valor de Banzhaf. Una vez calculada la relevancia de cada atributo de acuerdo con el juego definido, el último paso consiste en seleccionar los atributos que formaran parte del modelo de clasificación, la opción más sencilla es quedarse con los k mejores, que es la alternativa que se ha utilizado en este capítulo y que es habitual en la literatura sobre este tema.

A continuación, se describen los algoritmos implementados para determinar la función característica del juego cooperativo introducido en la definición 5.1, los valores exactos y aproximados por muestreo del valor de Shapley y el valor de Banzhaf.

Algoritmo 5.1. Pseudocódigo del algoritmo para la determinación de la función característica del juego descrito en la definición 5.1 mediante validación cruzada.

```
INPUT:
    data: conjunto de datos con N + 1 columnas (atributos)
          - N atributos antecedentes (jugadores)
          - 1 atributo consecuente
OUTPUT:
    acc: resultados de la validación cruzada para cada combinación de
         antecedentes (valor de la función característica para cada
         combinación de atributos)
INICIO DEL ALGORITMO:
for numCoalition = 1 to 2^N-1
    subset = BuildCoalition(data, numCoalition)
    for fold = 1 to 10
        testSet = the fold^th segment of subset
        trainingSet = subset - testset
        predictiveModel = NaiveClassificationAlgorithm(trainingSet)
        acc[numCoalition][fold] = Test(testSet, predictiveModel)
    end for
    acc[numCoalition][11] = Average(acc[numCoalition][1...10])
end for
FIN DEL ALGORITMO
```

El algoritmo 5.1 calcula la precisión de predicción para cada posible combinación de variables antecedentes (es decir, para cada posible coalición de jugadores) utilizando para ello validación cruzada (cross-validation) diez veces. De esta forma se evitan posibles sesgos y se puede dar una medida del error.

Algunas explicaciones sobre las principales funciones utilizadas en el algoritmo 5.1 son las siguientes. *BuildCoalition* devuelve un subconjunto de datos que contiene solo las columnas de los datos que se corresponden con el número de coalición indicada (más la columna consecuente). *NaiveClassificationAlgorithm* es el algoritmo 4.1 para el método de clasificación intuitivo *I*. Recuérdese que este algoritmo genera un modelo predictivo basado en reglas de clasificación a partir de un conjunto de datos de entrenamiento. *Test* valida el modelo predictivo con el conjunto de test proporcionado, y devuelve el porcentaje de acierto de predicción (accuracy) resultante, para ello, internamente, utiliza el algoritmo 4.2.

Obsérvese también que la salida del algoritmo 5.1., *acc*, es una matriz que contiene en cada fila los resultados de una coalición, donde en las 10 primeras posiciones están los resultados de predicción de las 10 validaciones cruzadas, y en la última posición (la 11) el promedio de esas primeras 10 posiciones, es decir, el resultado de haber aplicado validación cruzada 10 veces. Por tanto, cada una de las diez primeras columnas de la matriz se corresponde con la función característica del juego dado en la definición 5.1 para la correspondiente validación cruzada, y la última columna es su promedio.

El algoritmo 5.2 calcula de forma exacta el valor de Shapley y el valor de Banzhaf para el juego introducido en la definición 5.1. Este algoritmo, entre sus entradas tiene la salida del algoritmo 5.1, la matriz *acc*, por tanto, se calcula el valor de Shapley y el valor de Banzhaf para cada uno de los 11 juegos que resultan de la validación cruzada.

Algoritmo 5.2. Pseudocódigo del algoritmo para el cálculo exacto de los valores de Shapley y de Banzhaf.

```
INPUT:
    acc: resultados de la validación cruzada para cada coalición
OUTPUT:
    shap: valores de Shapley
    banz: valores de Banzhaf
INICIO DEL ALGORITMO:
    coalitions = tamaño de acc
    N = log2(coalitions + 1)
for fold = 1 to 11
    for i = 1 to N
        shap[i][fold] = banz[i][fold] = 0
    end for
    for c=1 to coalitions
        players = GetNumPlayers(c)
        for i = 1 to N
            if i ∈ c then // si el jugador i está en la coalición "c"
                shap[i][fold] += Factor(players-1, N) * acc[c][fold]
                banz[i][fold] += acc[c][fold] / coalitions
            else
                shap[i][fold] -= Factor(players, N) * acc[c][fold]
                banz[i][fold] -= acc[c][fold] / coalitions
            end if
        end for
    end for
end for
FIN DEL ALGORITMO
```

Algunas de las funciones más relevantes que se utilizan en el algoritmo 5.2 son las siguientes. *GetNumPlayers* devuelve el número de jugadores de la coalición indicada. $Factor(p,N) = \frac{(p!\ (N-p-1)!)}{N!}$.

Obsérvese que las matrices de salida *shap* y *banz* contienen, respectivamente, los valores de Shapley y Banzhaf, en cada fila están los datos de cada jugador, en las posiciones 1 a 10 están los valores de las pruebas de validación cruzada 1 a 10 de la prueba cross validation, y en la posición 11 el valor global.

Por último, el algoritmo 5.3 calcula los valores de Shapley y Banzhaf estimados mediante muestreo aleatorio simple (MAS). Como se ha indicado con anterioridad, para el caso del valor de Shapley el muestro se

realiza sobre el conjunto de permutaciones, mientras que en el caso del valor de Banzhaf se lleva a cabo sobre el conjunto de posibles coaliciones.

Algoritmo 5.3. Pseudocódigo del algoritmo para el cálculo aproximado por muestro de los valores de Shapley y Banzhaf

```
INPUT:
    data: conjunto de datos con N + 1 columnas (atributos)
          - N atributos antecedente (jugadores)
          - 1 atributo consecuente
    samplesize: tamaño muestral
OUTPUT:
    shap: Shapley values
    banz: Banzhaf values
INICIO DEL ALGORITMO:
    C1 = {1, 2, 3, ... , N} // coalición de todos los atributos antecedentes
for f = 1 to 11
    for i = 1 to N
        shap[1][i][f] = banz[1][i][f] = 0
    end for
end for
for M = 1 to samplesize
    C1 = RandomPermutation(C1)
    n = Random(1, N)
    C2 = RandomCoalition(N, n)
    for i = 1 to N
        C1' = {p ∈ C1 / p is before i in C1}
        accShap1 = AccuracyModel(data, C1' ∪ {i})
        accShap2 = AccuracyModel(data, C1')
        C2' = {p ∈ C2 / p ≠ i}
        accBanz1 = AccuracyModel(data, C2' ∪ {i})
        accBanz2 = AccuracyModel(data, C2')
        for f = 1 to 11
        shap[M][i][f] += accShap1[f] - accShap2[f]
        banz[M][i][f] += accBanz1[f] - accBanz2[f]
        if M < samplesize then
            shap[M+1][i][f] = shap[M][i][f]
            banz[M+1][i][f] = banz[M][i][f]
        end if
        shap[M][i][f] = shap[M][i][f] / M
        banz[M][i][f] = banz[M][i][f] / M
        end for
    end for
end for
FIN DEL ALGORITMO
```

Las funciones que se pueden destacar del algoritmo 5.3 son las siguientes. *RandomPermutation* devuelve una permutación aleatoria de la coalición indicada, en este caso de la coalición de todos los atributos. Esta función es la que sirve para seleccionar la muestra de permutaciones para el cálculo del valor de Shapley. *Random*(*min*, *max*) devuelve un número aleatorio comprendido entre los valores *min* y *max*, y *RandomCoalition*(N, n) devuelve una coalición aleatoria de jugadores (atributos) comprendidos entre 1 y N de tamaño n. Estas dos funciones sirven para seleccionar la muestra de coaliciones para el cálculo del valor de Banzhaf. La primera función selecciona el tamaño de la coalición y la segunda los atributos que la forman. *AccuracyModel* calcula la precisión del modelo predictivo basado en reglas de clasificación del algoritmo 4.1 para el conjunto de datos y la coalición indicados aplicando validación cruzada. En concreto, devuelve un vector con los valores de precisión (*accuracy*) de cada repetición de la validación cruzada (ítems de 1 a 10) además de la precisión global del modelo (ítem 11).

Obsérvese que las matrices de salida *shap* y *banz* ahora son de tres dimensiones. La nueva dimensión M representa las muestras consideradas para la estimación de los valores de Shapley y Banzhaf, dicha estimación se va acumulando de forma que para sucesivas muestras $(M = 1, 2, \ldots, samplesize)$ la matriz contiene la estimación acumulada del valor de Shapley o Banzhaf hasta dicho valor de M.

5.4. Comparación de diferentes métodos de selección de características

En la literatura existen diferentes métodos de ordenación y selección de características de acuerdo a distintos criterios y medidas. En esta sección se utilizarán algunos de los implementados en Weka (Weka, 2020, 2021) para comparar los resultados con los obtenidos de la aplicación de los algoritmos de selección de características basados en el valor de Shapley y el valor de Banzhaf que se describieron en la sección 5.3. En particular, se utilizarán los siguientes siete *rankers*:

- *ClassifierAttributeEval* (*Class*) que evalúa el valor de un atributo mediante el uso de un clasificador especificado por el usuario. En esta sección se ha utilizado el clasificador J48.

- *CorrelationAttributeEval* (*Corr*) que evalúa el valor de un atributo midiendo la correlación (de Pearson) entre este y la clase.
- *GainRatioAttributeEval* (*Gain*) que evalúa los atributos midiendo su relación de ganancia con respecto a la clase.
- *InfoGainAttributeEval* (*Info*) que evalúa los atributos midiendo su ganancia de información con respecto a la clase.
- *OneRAttributeEval* (*OneR*) que utiliza la medida de precisión simple adoptada por el clasificador OneR. Puede usar los datos de entrenamiento para la evaluación o puede aplicar una validación cruzada interna: el número de pliegues es un parámetro.
- *ReliefFAttributeEval* (*RelF*) que está basado en instancias. Muestrea instancias aleatoriamente y verifica instancias vecinas de la misma clase y de diferentes clases.
- *SymmetricalUncertAttributeEval* (*Symm*) que evalúa un atributo midiendo su incertidumbre simétrica con respecto a la clase.

Véase Frank et al. (2016) o Weka (2020, 2021), para detalles sobre el funcionamiento de estos *rankers*.

La base de comparación en la ordenación y selección de características de los rankers anteriores y los basados en teoría de juegos se ha realizado sobre un total de 11 conjuntos de datos, los cuales ya fueron descritos en la subsección 4.3.1.

Todos los resultados han sido obtenidos con 10 repeticiones en validación cruzada. Puesto que lo relevante en esta comparación es la ordenación de las características de acuerdo a su importancia, no se muestran los valores concretos obtenidos en las funciones de mérito utilizadas en cada uno de los rankers o los valores de Shapley y Banzhaf, sino las ordenaciones obtenidas de más relevante a menos, mostrando únicamente las 5 características más relevantes.

Para calcular la concordancia entre los órdenes obtenidos por cada clasificador se utilizará una adaptación del coeficiente de correlación de Spearman, lo que mostrará el nivel de concordancia entre los órdenes obtenidos entre dos clasificadores. Esta medida puede aplicarse al total de las características ordenadas, pero en esta sección solo se aplicará a las cinco mejores características en el orden y, además, a las características que

sean comunes. Obviamente, cuando el número de características comunes sea 0 ó 1 no se puede hablar de concordancia por lo que no se calculará y se indicará por (--). La expresión matemática del coeficiente de correlación de Spearman viene dada por

$$S = 1 - \frac{6\sum_{i=1}^{n} d_i^2}{n(n^2-1)}, \tag{5.7}$$

donde n es el número de características y d_i es la diferencia en los órdenes para la característica i con cada uno de los rankers. El coeficiente de Spearman S toma valores entre -1 y $+1$, donde -1 indicará que las ordenaciones son completamente discordantes y $+1$ que las ordenaciones son completamente concordantes. El valor 0 indicará que no hay concordancia, y los valores intermedios indicarán diferentes niveles de concordancia o discordancia.

Dado que las ordenaciones no son apareadas, lo que se hace es lo siguiente para calcular el coeficiente de correlación de Spearman. Dadas dos ordenaciones de sendos rankers, se considera que una de las ordenaciones es la natural, es decir, $1, 2, 3, \dots, n$, y la otra se ajusta con la primera respondiendo a la pregunta siguiente: ¿qué posición ocupa la característica que ocupa la posición i-ésima en la primera ordenación en la ordenación del segundo ranker? Con estas dos ordenaciones se calcula ya el coeficiente de Spearman S. Como ejemplo considérese que tres características han sido ordenadas por dos rankers de la siguiente forma:

C_1, C_2, C_3 y C_3, C_1, C_2

Entonces las posiciones correspondientes de las características en los dos órdenes serían: $1,2,3$ y $2,3,1$, respectivamente. Ahora haciendo las diferencias y aplicando la expresión (5.7) se obtendría:

$$S = 1 - \frac{6 \cdot (1+1+4)}{3(9-1)} = -0.5.$$

Por tanto, las dos ordenaciones son discordantes. Esto quiere decir que, aunque los dos clasificadores han seleccionado estas tres características entre las cinco mejores, sus ordenaciones han sido diferentes dentro de esa selección de características.

5.5. Resultados y discusión

En esta sección se presentan los resultados obtenidos de la aplicación de los rankers escritos en la sección anterior, el valor de Shapley y el valor de Banzhaf para cada uno de los once conjuntos de datos. Asimismo, para cada uno de esos conjuntos se analiza el grado de coincidencia en la selección de las cinco mejores características, es decir, las cinco primeras características en el raking correspondiente, entre todos los rankers utilizados y los valores de Shapley y Banzhaf. Con ello se trata de mostrar las discrepancias y similitudes entre todos ellos en la selección de características. También se analizará si aquellas características comunes han sido seleccionadas manteniendo el orden, es decir, si dos rankers tienen en su selección dos o más características en común, si estas mantienen la relación relativa de orden en ambos. En todos estos análisis el objeto central será estudiar el comportamiento de los valores de Shapley y Banzhaf en comparación con el resto.

A continuación, para cada conjunto de datos se presentan en tablas los resultados obtenidos en relación a lo comentado en el párrafo anterior. Los resultados para el valor de Shapley y el valor de Banzhaf en las tablas 5.1, 5.8 y 5.11, que se corresponden con los conjuntos de datos Thyroid, Connect4 y Mushroom, han sido obtenidos utilizando los algoritmos de cálculo aproximado con un tamaño muestral de 100. Obviamente para otros tamaños muestrales los resultados podrían ser algo diferentes. El hecho de no calcularlos de forma exacta se debe a que por el tamaño del conjunto de datos en términos de número de características el problema es inabordable computacionalmente.

TABLA 5.1. Para el conjunto de datos **Thyroid**, mejores cinco características seleccionadas por cada uno de los rankers y por los valores de Shapley y Banzhaf. Se indica también el ranking promedio y su desviación media.

Ranker	1º	2º	3º	4º	5º
Class	Pregnant (1±0)	131_treatment (2±0)	Query_hypoth. (3±0)	Thyroid_surgery (4±0)	Sick (5±0)
Corr	Pregnant (1±0)	Sex (2±0)	Psych (3.5±0.5)	On_thyroxine (3.5±0.5)	TSH_measured (5.2±0.4)
Gain	Hypopituitary (1±0)	Pregnant (2±0)	Psych (3±0)	Goitre (4.4±0.49)	Referral_source (4.6±0.49)
Info	Referral_source (1±0)	Pregnant (2±0)	Sex (3±0)	Psych (4±0)	On_thyroxine (5±0)
OneR	Pregnant (1±0)	Query_hypoth. (3.8±0.4)	On_thyroxine (5±2)	Query_on_thyr. (5.3±0.9)	Hypopituitary (5.4±6.1)
RelF	TSH_measured (1±0)	T3_measured (2±0)	TBG_measured (4.2±1.08)	Goitre (4.2±0.87)	Sex (5±0)
Symm	Pregnant (1±0)	Referral_source (2±0)	Psych (3±0)	Sex (4±0)	On_thyroxine (5.1±0.3)
Shapley	Sex (1±0)	Psych (2.5±0.5)	On_antithyroid (2.5±0.5)	131_treatment (4.42±0.49)	Thyroid_surgery (4.58±0.49)
Banzhaf	Pregnant (2.50±2.58)	Goitre (5.75±2.67)	TT4_measured (6.25±2.50)	FTI_measured (7.42±2.08)	T4U_measured (7.75±4.71)

TABLA 5.1BIS. Para el conjunto de datos **Thyroid**, coincidencia en la selección de las cinco mejores características entre todos los rankers utilizados. Entre paréntesis se muestra el coeficiente de correlación de Spearman entre las características comunes.

Ranker	Class	Corr	Gain	Info	OneR	RelF	Symm	Shapley	Banzhaf
Class	--	1(--)	1(--)	1(--)	2(1.0)	0(--)	1(--)	2(1.0)	1(--)
Corr		--	2(1.0)	4(1.0)	2(1.0)	2(–1.0)	4(0.8)	2(1.0)	1(--)
Gain			--	3(–0.5)	1(--)	1(--)	3(0.5)	1(--)	2(1.0)
Info				--	2(1.0)	1(--)	5(0.8)	2(1.0)	1(--)
OneR					--	0(--)	1(--)	0(--)	1(--)
RelF						--	1(--)	1(--)	1(--)
Symm							--	2(–1.0)	1(--)
Shapley								--	0(--)
Banzhaf									--

Para el conjunto de datos Thyroid (véanse las tablas 5.1 y 5.1BIS) el grado de coincidencia en la selección de las cinco mejores características de todos los rankers, incluyendo el valor de Shapley y el valor de Banzhaf,

es muy bajo, existiendo solo un grado de coincidencia alto, tanto en las características seleccionadas como en su orden, entre los rankers *Corr*, *Info* y *Symm*. El resto de rankers, incluidos los dos basados en teoría de juegos, tienen un grado de coincidencia muy bajo. Obsérvese que este conjunto de datos tiene 22 variables explicativas y la variable respuesta solo dos clases, por tanto, es posible, que el resultado observado se deba a que los rankers tienen criterios de selección muy dispares y la cantidad de variables explicativas hace que haya mucha variabilidad. En cualquier caso, hay tres características que han sido las más seleccionadas: Pregnant(7/9), Sex(5/9) y Psych(5/9). Estas características también fueron seleccionadas por el valor de Banzhaf la primera y por el valor de Shapley las otras dos. Esto indica que el valor de Shapley y Banzhaf coinciden en detectar cuáles pueden ser las características más relevantes.

TABLA 5.2. Para el conjunto de datos **Healthy**, mejores cinco características seleccionadas por cada uno de los rankers y por los valores de Shapley y Banzhaf. Se indica también el ranking promedio y su desviación media.

Ranker	1º	2º	3º	4º	5º
Class	C4 (1±0)	C5 (2±0)	C2 (3±0)	C3 (4±0)	C9 (5±0)
Corr	C4 (1±0)	C3 (2±0)	C5 (3±0)	C6 (4±0)	C2 (5±0)
Gain	C4 (1±0)	C3 (2±0)	C6 (3±0)	C5 (4±0)	C7 (5±0)
Info	C4 (1±0)	C3 (2±0)	C5 (3±0)	C6 (4±0)	C7 (5±0)
OneR	C4 (1±0)	C5 (2±0)	C2 (3±0)	C3 (4±0)	C9 (5±0)
RelF	C4 (1±0)	C3 (2±0)	C5 (3±0)	C6 (4±0)	C7 (5±0)
Symm	C4 (1±0)	C3 (2±0)	C6 (3±0)	C5 (4±0)	C7 (5±0)
Shapley	C4 (1±0)	C3 (2.3±0.42)	C6 (2.7±0.42)	C5 (4±0)	C7 (5±0)
Banzhaf	C4 (1±0)	C3 (2.5±0.5)	C6 (2.5±0.5)	C5 (4±0)	C7 (5±0)

TABLA 5.2BIS. Para el conjunto de datos **Healthy**, coincidencia en la selección de las cinco mejores características entre todos los rankers utilizados. Entre paréntesis se muestra el coeficiente de correlación de Spearman entre las características comunes.

Ranker	Class	Corr	Gain	Info	OneR	RelF	Symm	Shapley	Banzhaf
Class	--	4(0.4)	3(0.5)	3(0.5)	5(1.0)	3(0.5)	3(0.5)	3(0.5)	3(0.5)
Corr		--	4(0.8)	4(1.0)	4(0.4)	4(1.0)	4(0.8)	4(0.8)	4(0.8)
Gain			--	5(0.9)	3(0.5)	5(0.9)	5(1.0)	5(1.0)	5(1.0)
Info				--	3(0.5)	5(1.0)	5(1.0)	5(1.0)	5(1.0)
OneR					--	3(0.5)	3(0.5)	3(0.5)	3(0.5)
RelF						--	5(0.9)	5(0.9)	5(0.9)
Symm							--	5(1.0)	5(1.0)
Shapley								--	5(1.0)
Banzhaf									--

Para el conjunto de datos Healthy (véanse las tablas 5.2 y 5.2BIS más arriba) el grado de coincidencia en la selección de características de todos los rankers es muy alto, y ello también se observa para los valores de Shapley y Banzhaf, por tanto, el funcionamiento de todos ellos es muy similar. En particular, hay tres caracterísitcas seleccionadas por todos los rankers, también por *Shapley* y *Banzhaf*, que son C4, C5 y C3. Por lo que, en este caso, el valor de Shapley y el valor de Banzhaf muestra un funcionamiento completamente análogo al resto de rankers utilizados como referencias.

Los resultados obtenidos para el conjunto de datos Avila (véanse las tablas 5.3 y 5.3BIS en la página siguiente) muestran dos grupos diferenciados de rankers con un alto grado de coincidencia tanto en la selección como en la ordenación dentro de cada grupo. El primer grupo estaría formado por los rankers *Gain*, *Info*, y *Symm*, y el segundo grupo por *RelF*, *Shapley* y *Banzhaf*. Los otros tres rankers tienen comportamientos dispares en sus comparaciones con el resto. Asimismo, las características F1 y F5 son seleccionadas por todos los rankers. Por tanto, una vez más, el valor de Shapley y el valor de Banzhaf muestran capacidad para seleccionar características relevantes para la predicción de la clase de la variable respuesta.

TABLA 5.3. Para el conjunto de datos **Avila**, mejores cinco características seleccionadas por cada uno de los rankers y por los valores de Shapley y Banzhaf. Se indica también el ranking promedio y su desviación media.

Ranker	1º	2º	3º	4º	5º
Class	F5 (1±0)	F9 (2±0)	F7 (3±0)	F1 (4±0)	F8 (6±0)
Corr	F1 (1±0)	F9 (2±0)	F5 (3±0)	F6 (4.4±0.49)	F7 (4.6±0.49)
Gain	F5 (1±0)	F1 (2±0)	F9 (3±0)	F3 (4±0)	F7 (5±0)
Info	F5 (1±0)	F1 (2±0)	F9 (3±0)	F3 (4±0)	F7 (5±0)
OneR	F5 (1±0)	F9 (2±0)	F7 (3±0)	F1 (4±0)	F6 (5.5±1.5)
RelF	F1 (1±0)	F5 (2±0)	F3 (3±0)	F2 (4±0)	F4 (5±0)
Symm	F5 (1±0)	F1 (2±0)	F9 (3±0)	F3 (4±0)	F7 (5±0)
Shapley	F1 (1±0)	F5 (2±0)	F3 (3±0)	F2 (4±0)	F4 (5±0)
Banzhaf	F1 (1±0)	F5 (2.4±0.48)	F3 (2.6±0.48)	F2 (4±0)	F4 (5±0)

TABLA 5.3BIS. Para el conjunto de datos **Avila**, coincidencia en la selección de las cinco mejores características entre todos los rankers utilizados. Entre paréntesis se muestra el coeficiente de correlación de Spearman entre las características comunes.

Ranker	Class	Corr	Gain	Info	OneR	RelF	Symm	Shapley	Banzhaf
Class	--	3(−1.0)	4(0.4)	4(0.4)	4(1.0)	2(−1.0)	4(0.4)	2(−1.0)	2(−1.0)
Corr		--	4(0.4)	4(0.4)	5(0.1)	2(1.0)	4(0.4)	2(1.0)	2(1.0)
Gain			--	5(1.0)	4(0.4)	3(0.5)	5(1.0)	3(0.5)	3(0.5)
Info				--	4(0.4)	3(0.5)	5(1.0)	3(0.5)	3(0.5)
OneR					--	2(−1.0)	4(0.4)	2(−1.0)	2(−1.0)
RelF						--	3(0.5)	5(1.0)	5(1.0)
Symm							--	3(0.5)	3(0.5)
Shapley								--	5(1.0)
Banzhaf									--

TABLA 5.4. Para el conjunto de datos **Adult**, mejores cinco características seleccionadas por cada uno de los rankers y por los valores de Shapley y Banzhaf. Se indica también el ranking promedio y su desviación media.

Ranker	1º	2º	3º	4º	5º
Class	Educ (1.1±0.3)	EducNum (1.9±0.3)	WorkClass (3.8±2.4)	FNLWGT (4.1±0.3)	HoursWeek (4.8±0.6)
Corr	MaritalStatus (1±0)	Relationship (2±0)	Sex (3±0)	Age (4±0)	Educ (6±1)
Gain	MaritalStatus (1±0)	Relationship (2±0)	Sex (3±0)	Age (4±0)	Educ (5±01)
Info	RelationShip (1±0)	MaritalStatus (2±0)	EducNum (3±0)	Educ (4±0)	Age (5±0)
OneR	Educ (1±0)	EducNum (2±0)	WorkClass (3±0)	FNLWGT (4±0)	HoursWeek (5±0)
RelF	RelationShip (1±0)	MaritalStatus (2±0)	Age (3±0)	Occupation (4±0)	HoursWeek (5±0)
Symm	MaritalStatus (1±0)	Relationship (2±0)	Age (3±0)	Educ (4±0)	EducNum (5±0)
Shapley	Educ (1.8±0.96)	EducNum (1.8±0.96)	MaritalStatus (2.5±1)	Relationship (2.9±0.92)	Occupation (5.3±0.42)
Banzhaf	Relationship (1.6±0.48)	MaritalStatus (1.6±0.72)	Educ (3±0.2)	EducNum (3±0.2)	Occupation (4.9±0.38)

TABLA 5.4BIS. Para el conjunto de datos **Adult**, coincidencia en la selección de las cinco mejores características entre todos los rankers utilizados. Entre paréntesis se muestra el coeficiente de correlación de Spearman entre las características comunes.

Ranker	Class	Corr	Gain	Info	OneR	RelF	Symm	Shapley	Banzhaf
Class	--	1(--)	1(--)	2(–1.0)	5(1.0)	1(--)	2(1.0)	2(1.0)	2(1.0)
Corr		--	5(1.0)	4(0.6)	1(--)	3(0.5)	4(1.0)	3(–0.5)	3(0.5)
Gain			--	4(0.6)	1(--)	3(0.5)	4(1.0)	3(–0.5)	3(0.5)
Info				--	2(–1.0)	3(1.0)	5(0.5)	4(–1.0)	4(0.8)
OneR					--	1(--)	2(1.0)	2(1.0)	2(1.0)
RelF						--	3(0.5)	2(–1.0)	3(1.0)
Symm							--	4(–0.6)	4(0.8)
Shapley								--	5(0.1)
Banzhaf									--

En el conjunto de datos Adult (véanse las tablas 5.4 y 5.4BIS) se observan tres grupos de clasificadores en cuanto a las características que seleccionan, aunque las concordancias en las ordenaciones son dispares. El primer grupo lo forman *Gain, Corr, Info y Symm* que coinciden en al

menos 4 de las características seleccionadas, el segundo grupo los forman *Symm, Shapley y Banzhaf* que también coinciden en al menos 4 de las características seleccionadas, y el último grupo lo forman *Class y OneR* que coinciden en las cinco características seleccionadas y en la ordenación de las mismas. En este caso queda solo, *RelF*. Globalmente, la característica Educ es seleccionada por todos los rankers menos *RelF*, y las características MaritalStatus y Relationship son seleccionadas por todos los rankers excepto *Class y OneR*. Finalmente, EducNum es seleccionada por 6 rankers, todos excepto *Corr, Gain y RelF*. Por tanto, se observa que *Shapley y Banzhaf* han seleccionado las 4 características más relevantes a la vista de todos los resultados obtenidos por los rankers.

TABLA 5.5. Para el conjunto de datos **Nursery**, mejores cinco características seleccionadas por cada uno de los rankers y por los valores de Shapley y Banzhaf. Se indica también el ranking promedio y su desviación media.

Ranker	1º	2º	3º	4º	5º
Class	Class1 (1±0)	F2 (2±0)	F1 (3±0)	F7 (4.3±0.46)	F5 (4.7±0.46)
Corr	Class1 (1±0)	F2 (2±0)	F1 (3±0)	F5 (4.4±0.49)	F7 (4.6±0.49)
Gain	Class1 (1±0)	F2 (2±0)	F1 (3±0)	F7 (4±0)	F5 (5±0)
Info	Class1 (1±0)	F2 (2±0)	F1 (3±0)	F7 (4±0)	F5 (5±0)
OneR	Class1 (1±0)	F2 (2±0)	F1 (3±0)	F7 (4.4±0.49)	F5 (4.6±0.49)
RelF	Class1 (1±0)	F2 (2±0)	F1 (3±0)	F5 (4±0)	F7 (5±0)
Symm	Class1 (1±0)	F2 (2±0)	F1 (3±0)	F7 (4±0)	F5 (5±0)
Shapley	Class1 (1±0)	F2 (2±0)	F1 (3±0)	F5 (4.1±0.18)	F7 (5.4±0.76)
Banzhaf	Class1 (1±0)	F2 (2±0)	F1 (3±0)	F5 (4.3±0.48)	F7 (5.4±0.88)

Para el conjunto de datos Nursery (véanse las tablas 5.5 y 5.5BIS) no hay mucho que comentar, puesto que todos los rankers seleccionan las mismas cinco características con pequeñas variaciones en la ordenación de las dos últimas. En este sentido, *Shapley y Banzhaf* no se desvían de los resultados obtenidos con el resto de rankers, lo cual indica que cuando hay

unas características claramente más relevantes que otras *Shapley y Banzhaf* las detectan correctamente.

TABLA 5.5BIS. Para el conjunto de datos **Nursery**, coincidencia en la selección de las cinco mejores características entre todos los rankers utilizados. Entre paréntesis se muestra el coeficiente de correlación de Spearman entre las características comunes.

Ranker	Class	Corr	Gain	Info	OneR	RelF	Symm	Shapley	Banzhaf
Class	--	5(0.9)	5(1.0)	5(1.0)	5(1.0)	5(0.9)	5(1.0)	5(0.9)	5(0.9)
Corr		--	5(0.9)	5(0.9)	5(0.9)	5(1.0)	5(0.9)	5(1.0)	5(1.0)
Gain			--	5(1.0)	5(1.0)	5(0.9)	5(1.0)	5(0.9)	5(0.9)
Info				--	5(1.0)	5(0.9)	5(1.0)	5(0.9)	5(0.9)
OneR					--	5(0.9)	5(1.0)	5(0.9)	5(0.9)
RelF						--	5(0.9)	5(1.0)	5(1.0)
Symm							--	5(0.9)	5(0.9)
Shapley								--	5(1.0)
Banzhaf									--

TABLA 5.6. Para el conjunto de datos **Bank**, mejores cinco características seleccionadas por cada uno de los rankers y por los valores de Shapley y Banzhaf. Se indica también el ranking promedio y su desviación media.

Ranker	1º	2º	3º	4º	5º
Class	Poutcome (1±0)	Marital (2±0)	Job (3±0)	Month (4±0)	Education (5±0)
Corr	Poutcome (1±0)	Housing (2±0)	Contact (3±0)	Loan (4±0)	Marital (5.2±0.4)
Gain	Poutcome (1±0)	Contact (2±0)	Housing (3±0)	Month (4±0)	Loan (5±0)
Info	Poutcome (1±0)	Month (2±0)	Contact (3±0)	Housing (4±0)	Job (5±0)
OneR	Poutcome (1±0)	Marital (3.2±0.4)	Month (3.6±3.2)	Job (3.8±0.4)	Education (4.4±1.2)
RelF	Job (1±0)	Age (2±0)	Month (3±0)	Education (4.3±0.46)	Marital (4.7±0.46)
Symm	Poutcome (1±0)	Contact (2±0)	Month (3±0)	Housing (4±0)	Job (5.4±0.49)
Shapley	Poutcome (1±0)	Default (2.3±0.42)	Loan (3.8±0.84)	Housing (4.2±1.28)	Contact (4.3±0.7)
Banzhaf	Poutcome (1±0)	Default (2.3±0.42)	Loan (3.7±0.76)	Contact (4.1±0.74)	Housing (4.6±1.6)

TABLA 5.6BIS. Para el conjunto de datos **Bank**, coincidencia en la selección de las cinco mejores características entre todos los rankers utilizados. Entre paréntesis se muestra el coeficiente de correlación de Spearman entre las características comunes.

Ranker	Class	Corr	Gain	Info	OneR	RelF	Symm	Shapley	Banzhaf
Class	--	2(1.0)	2(1.0)	3(0.5)	5(0.9)	4(-0.2)	3(0.5)	1(--)	1(--)
Corr		--	4(0.8)	3(0.5)	2(1.0)	1(--)	3(0.5)	4(0.4)	4(0.2)
Gain			--	4(0.4)	2(1.0)	1(--)	4(0.8)	4(0.2)	4(0.4)
Info				--	3(1.0)	2(-1.0)	5(0.9)	3(0.5)	3(1.0)
OneR					--	4(-0.4)	3(1.0)	1(--)	1(--)
RelF						--	2(-1.0)	0(--)	0(--)
Symm							--	3(0.5)	3(1.0)
Shapley								--	5(0.9)
Banzhaf									--

El análisis del conjunto de datos Bank (véanse las tablas 5.6 y 5.6BIS) muestra tres grupos de rankers con concordancias en la ordenación dispares. Por una parte, se tendría el grupo formado por *Class, OneR y RelF* que coinciden en la selección de al menos 4 características. Por otra parte, estaría el grupo formado por *Corr, Shapley y Banzhaf* que también coinciden en la selección de al menos 4 características. Y, por último, estaría el grupo *Gain, Info, Symm, Shapley y Banzhaf* que también coinciden en la selección de al menos 4 características. En cualquier caso, Shapley y Banzhaf coinciden en la selección de las cinco características, pero con diferente ordenación. La característica más seleccionada es Poutcome, que es seleccionada por todos los rankers excepto *RelF*. Mientras que Month, Housing y Contact son selecciondas por 6 rankers cada una. *Shapley y Banzhaf* seleccionan Poutcome, Contact y Housing, por tanto, son capaces de detectar a tres de los cuatro considerados más relevantes por todos los rankers.

En el caso del conjunto de datos HTRU2 (véanse las tablas 5.7 y 5.7BIS en la página siguiente) se está casi en la misma situación que en el conjunto de datos Nursery, pero se pueden distinguir dos grupos con concordancias tanto en la selección de características como en las ordenaciones muy altas. El primer grupo lo formarían *Class y OneR*, y el segundo grupo *Corr, Gain, Info, RelF, Symm, Shapley y Banzhaf*. Hay dos características seleccionadas

por todos los rankers, A3 y A4, y otras dos, A1 y A6, seleccionadas por todos los rankers del segundo grupo. Una vez más se muestra que *Shapley y Banzhaf* se comportan tan bien como otros rankers.

TABLA 5.7. Para el conjunto de datos **HTRU2**, mejores cinco características seleccionadas por cada uno de los rankers y por los valores de Shapley y Banzhaf. Se indica también el ranking promedio y su desviación media.

Ranker	1º	2º	3º	4º	5º
Class	A8 (1±0)	A7 (2±0)	A2 (3±0)	A3 (4±0)	A4 (5±0)
Corr	A3 (1±0)	A1 (2±0)	A4 (3±0)	A6 (4±0)	A5 (5±0)
Gain	A3 (1±0)	A1 (2±0)	A4 (3±0)	A6 (4±0)	A5 (5±0)
Info	A3 (1±0)	A1 (2±0)	A4 (3±0)	A6 (4±0)	A5 (5±0)
OneR	A8 (1±0)	A7 (2±0)	A2 (3±0)	A3 (4±0)	A4 (5±0)
RelF	A3 (1±0)	A1 (2±0)	A4 (3±0)	A2 (4±0)	A6 (5±0)
Symm	A3 (1±0)	A1 (2±0)	A4 (3±0)	A6 (4±0)	A5 (5±0)
Shapley	A3 (2.1±0.94)	A4 (2.6±1.44)	A1 (2.8±0.68)	A6 (3.6±1.36)	A5 (4.8±1.2)
Banzhaf	A3 (1.9±0.9)	A1 (2.4±0.6)	A4 (2.5±1.2)	A6 (3.9±0.96)	A5 (4.9±0.92)

TABLA 5.7BIS. Para el conjunto de datos **HTRU2**, coincidencia en la selección de las cinco mejores características entre todos los rankers utilizados. Entre paréntesis se muestra el coeficiente de correlación de Spearman entre las características comunes.

Ranker	Class	Corr	Gain	Info	OneR	RelF	Symm	Shapley	Banzhaf
Class	--	2(1.0)	2(1.0)	2(1.0)	5(1.0)	3(−0.5)	2(1.0)	2(1.0)	2(1.0)
Corr		--	5(1.0)	5(1.0)	2(1.0)	4(1.0)	5(1.0)	5(0.9)	5(1.0)
Gain			--	5(1.0)	2(1.0)	4(1.0)	5(1.0)	5(0.9)	5(1.0)
Info				--	2(1.0)	4(1.0)	5(1.0)	5(0.9)	5(1.0)
OneR					--	3(−0.5)	2(1.0)	2(1.0)	2(1.0)
RelF						--	4(1.0)	4(0.8)	4(1.0)
Symm							--	5(0.9)	5(1.0)
Shapley								--	5(0.9)
Banzhaf									--

TABLA 5.8. Para el conjunto de datos **Connect4**, mejores cinco características seleccionadas por cada uno de los rankers y por los valores de Shapley y Banzhaf. Se indica también el ranking promedio y su desviación media.

Ranker	1º	2º	3º	4º	5º
Class	D3 (1±0)	C1 (4.2±0.75)	C2 (5.6±1.85)	C3 (5.8±1.83)	C4 (7.6±1.2)
Corr	A1 (1±0)	D1 (2±0)	G1 (3±0)	B1 (4±0)	D3 (5.2±0.4)
Gain	G6 (1±0)	D3 (2±0)	F6 (3.2±0.6)	D2 (3.9±0.3)	B6 (5.7±0.78)
Info	A1 (1±0)	D1 (2±0)	G1 (3.1±0.3)	D2 (3.9±0.3)	D3 (5±0)
OneR	D3 (1±0)	G6 (2.1±0.3)	C2 (4.7±0.78)	G4 (4.9±1.14)	C3 (6.4±1.2)
RelF	D1 (1±0)	D2 (2±0)	C2 (3±0)	A1 (4±0)	D3 (5±0)
Symm	A1 (1.4±0.49)	D3 (1.6±0.49)	D2 (3±0)	D1 (4±0)	G1 (5±0)
Shapley	E4 (1.33±0.44)	D2 (1.67±0.44)	F2 (3±0)	B2 (4.33±0.44)	F1 (4.67±0.44)
Banzhaf	D3 (1.67±0.56)	D1 (2.25±1.08)	D2 (2.67±0.89)	C3 (3.50±0.67)	C2 (4.92±0.15)

TABLA 5.8BIS. Para el conjunto de datos **Connect4**, coincidencia en la selección de las cinco mejores características entre todos los rankers utilizados. Entre paréntesis se muestra el coeficiente de correlación de Spearman entre las características comunes.

Ranker	Class	Corr	Gain	Info	OneR	RelF	Symm	Shapley	Banzhaf
Class	--	1(--)	1(--)	1(--)	3(1.0)	2(−1.0)	1(--)	0(--)	3(0.5)
Corr		--	1(--)	4(1.0)	1(1.0)	3(0.5)	4(0.4)	0(--)	2(−1.0)
Gain			--	2(−1.0)	2(−1.0)	2(−1.0)	2(1.0)	1(--)	2(1.0)
Info				--	1(--)	4(0.4)	5(0.1)	1(--)	3(−0.5)
OneR					--	2(−1.0)	1(--)	0(--)	3(0.5)
RelF						--	4(−0.8)	1(--)	4(−0.2)
Symm							--	1(--)	3(0.5)
Shapley								--	1(--)
Banzhaf									--

El conjunto de datos Connect4 (véanse las tablas 5.8 y 5.8BIS anteriores) es el que presenta mayor variabilidad en la selección de características obtenidas por los rankers, no pudiendo establecer ninguna agrupación clara. La característica más seleccionada es D3, que es

propuesta por todos los rankers excepto *Shapley*, y la segunda más seleccionada es D2, que es ordenada entre las cinco más relevantes por 6 de los rankers, todos excepto *Class, Corr y One*R. En este caso, *Banzhaf* ha coincidido en la selección de las dos características más seleccionadas, e incluso la tercera, D1, lo cual es un resultado positivo. En este caso concreto, *Banzhaf* parece que se desempeña mejor que *Shapley*. La razón de estos resultados puede deberse al gran número de características que tiene este conjunto de datos.

TABLA 5.9. Para el conjunto de datos **Tic-tac-toe**, mejores cinco características seleccionadas por cada uno de los rankers y por los valores de Shapley y Banzhaf. Se indica también el ranking promedio y su desviación media.

Ranker	1º	2º	3º	4º	5º
Class	m-m-s (1±0)	m-l-s (2±0)	t-m-s (3±0)	t-r-s (4±0)	b-r-s (5±0)
Corr	m-m-s (1±0)	b-l-s (3.2±1.33)	t-l-s (3.4±0.92)	t-r-s (3.5±0.92)	b-r-s (3.9±1.14)
Gain	m-m-s (1±0)	b-l-s (3.2±1.33)	t-l-s (3.5±0.92)	t-r-s (3.6±1.02)	b-r-s (3.7±1.1)
Info	m-m-s (1±0)	b-l-s (3.3±1.35)	t-l-s (3.4±0.92)	t-r-s (3.6±1.02)	b-r-s (3.7±1.1)
OneR	m-m-s (1±0)	m-l-s (2±0)	t-m-s (3±0)	t-r-s (4±0)	b-r-s (5±0)
RelF	m-m-s (1±0)	b-r-s (3.2±1.17)	t-l-s (3.5±0.92)	t-r-s (3.6±1.11)	b-l-s (3.7±1.19)
Symm	m-m-s (1±0)	b-l-s (3.3±1.35)	t-l-s (3.4±0.92)	t-r-s (3.6±1.02)	b-r-s (3.7±1.1)
Shapley	m-m-s (1.2±0.32)	b-l-s (3.3±1.16)	b-r-s (3.4±1.28)	t-r-s (3.8±0.68)	t-l-s (4±1.6)
Banzhaf	m-m-s (1.1±0.18)	b-l-s (3.6±1.32)	b-r-s (3.7±1.1)	t-r-s (4±0.8)	t-l-s (4.3±1.7)

Como en Nursery y HTRU2, para el conjunto de datos Tic-tac-toe (véanse las tablas 5.9 y 5.9BIS que están arriba y abajo, respectivamente) existe un alto grado de coincidencia tanto en la selección de características como en su ordenación. No obstante, se pueden distinguir dos grupos, por un lado, *Class y One*R que seleccionan las mismas cinco características y con la misma ordenación, y por otro lado, *Corr, Gain, Info, RelF, Symm, Shapley y Banzhaf* que también eligen las mismas cinco características pero con pequeñas variaciones en el orden. Hay tres características que son seleccionadas por todos los rankers que son m-m-s, t-r-s y b-r-s. Una vez

más es evidente que el desempeño de *Shapley y Banzhaf* es comparable al del resto de rankers evaluados.

TABLA 5.9BIS. Para el conjunto de datos **Tic-tac-toe**, coincidencia en la selección de las cinco mejores características entre todos los rankers utilizados. Entre paréntesis se muestra el coeficiente de correlación de Spearman entre las características comunes.

Ranker	Class	Corr	Gain	Info	OneR	RelF	Symm	Shapley	Banzhaf
Class	--	3(1.0)	3(1.0)	3(1.0)	5(1.0)	3(0.5)	3(1.0)	3(0.5)	3(0.5)
Corr		--	5(1.0)	5(1.0)	3(1.0)	5(0.1)	5(1.0)	5(0.6)	5(0.6)
Gain			--	5(1.0)	3(1.0)	5(0.1)	5(1.0)	5(0.6)	5(0.6)
Info				--	3(1.0)	5(0.1)	5(1.0)	5(0.6)	5(0.6)
OneR					--	3(0.5)	3(1.0)	3(0.5)	3(0.5)
RelF						--	5(0.1)	5(0.3)	5(0.3)
Symm							--	5(0.6)	5(0.6)
Shapley								--	5(1.0)
Banzhaf									--

TABLA 5.10. Para el conjunto de datos **Credit**, mejores cinco características seleccionadas por cada uno de los rankers y por los valores de Shapley y Banzhaf. Se indica también el ranking promedio y su desviación media.

Ranker	1º	2º	3º	4º	5º
Class	A9 (1±0)	A10 (2±0)	A6 (3±0)	A7 (4±0)	A13 (6.2±1.33)
Corr	A9 (1±0)	A10 (2±0)	A4 (3±0)	A5 (4±0)	A7 (5.9±0.83)
Gain	A9 (1±0)	A10 (2±0)	A4 (3.3±0.9)	A5 (4.1±0.3)	A6 (4.8±0.6)
Info	A9 (1±0)	A10 (2±0)	A6 (3±0)	A7 (4±0)	A5 (5±0)
OneR	A9 (1±0)	A10 (2±0)	A6 (3±0)	A7 (4±0)	A13 (6.2±1.33)
RelF	A9 (1±0)	A6 (2±0)	A7 (3±0)	A10 (4±0)	A12 (5±0)
Symm	A9 (1±0)	A10 (2±0)	A6 (3±0)	A7 (4.4±0.8)	A5 (4.8±0.4)
Shapley	A9 (1.1±0.18)	A10 (1.9±0.18)	A6 (4.8±1.96)	A7 (5.1±1.54)	A13 (5.2±1.84)
Banzhaf	A9 (1.1±0.18)	A10 (1.9±0.18)	A4 (5.3±1.02)	A5 (5.3±1.02)	A13 (5.5±2.3)

TABLA 5.10BIS. Para el conjunto de datos **Credit**, coincidencia en la selección de las cinco mejores características entre todos los rankers utilizados. Entre paréntesis se muestra el coeficiente de correlación de Spearman entre las características comunes.

Ranker	Class	Corr	Gain	Info	OneR	RelF	Symm	Shapley	Banzhaf
Class	--	3(1.0)	3(1.0)	4(1.0)	5(1.0)	4(0.4)	4(1.0)	5(1.0)	3(1.0)
Corr		--	4(1.0)	4(0.8)	3(1.0)	3(0.5)	4(0.8)	3(1.0)	4(1.0)
Gain			--	4(0.8)	3(1.0)	3(0.5)	4(0.8)	3(1.0)	4(1.0)
Info				--	4(1.0)	4(0.4)	5(1.0)	4(1.0)	3(1.0)
OneR					--	4(0.4)	4(1.0)	5(1.0)	3(1.0)
RelF						--	4(0.4)	4(0.4)	2(1.0)
Symm							--	4(0.4)	3(1.0)
Shapley								--	3(1.0)
Banzhaf									--

Para el conjunto de datos Credit (véanse las tablas 10 y 10BIS más arriba) se distinguen dos grupos de rankers con alta coincidencia tanto en la selección como en la ordenación. El primer grupo estaría formado por los rankers *Class, Info, OneR, RelF, Symm y Shapley*, mientras que el segundo grupo por *Corr, Gain, Info, Symm y Banzhaf*. En ambos casos con una coincidencia en la selección de al menos 4 caracterísitcas y una alta concordancia en la ordenación. Todos los rankers seleccionan las características A9 y A10. Y siete rankers, entre ellos *Shapley*, seleccionan las caracterísitcas A6 y A7. Por tanto, en este caso, *Shapley* selecciona las 4 características más seleccionadas. Así, para este conjunto de datos se desempeña mejor que *Banzhaf*.

En el último conjunto de datos estudiado, Mushroom (véanse las tablas 5.11 y 5.11BIS en la página siguiente), hay, como en otros casos, gran variabilidad en la selección de características y solo pueden identificarse dos pequeños grupos de rankers. Por un lado, *Gain, Info, OneR* y *Symm* que tienen una coincidencia en la selección de al menos cuatro características y una concordancia en la ordenación moderada. Y, por otro lado, *RelF y Shapley* que coinciden en la selección de cuatro características. La característica más seleccionada es Odor, ordenada en primer o segundo lugar por 8 rankers, todos excepto *Class*. Las otras dos características más seleccionadas son Spore-print-col. y Ring-type, seleccionadas por 7 y 6

rankers, respectivamente. En este caso, *Shapley* selecciona estas tres carácterística, por lo que se puede decir que se desempeña tan bien como otros rankers y mejor que *Banzhaf*.

TABLA 5.11. Para el conjunto de datos **Mushroom**, mejores 5 características seleccionadas por cada uno de los rankers y por los valores de Shapley y Banzhaf. Se indica también el ranking promedio y su desviación media.

Ranker	1º	2º	3º	4º	5º
Class	Populat. (1±0)	Stalk-root (2±0)	Gill-spacing (3.1±0.3)	Gill-color (3.9±0.3)	Stalk-col.-bel. (5.1±0.3)
Corr	Odor (1±0)	Gill-size (2±0)	Bruises (3±0)	Stalk-surf.-ab. (4±0)	Stalk-surf.-bel. (5±0)
Gain	Odor (1±0)	Gill-size (2±0)	Stalk-surf.-ab. (3±0)	Spore-print-col. (4±0)	Ring-type (5±0)
Info	Odor (1±0)	Spore-print-col. (2±0)	Gill-color (3±0)	Ring-type (4±0)	Stalk-surf.-ab. (5±0)
OneR	Odor (1±0)	Spore-print-col. (2±0)	Gill-color (3±0)	Ring-type (4.4±0.49)	Stalk-surf.-ab. (4.6±0.49)
RelF	Odor (1±0)	Spore-print-col. (2±0)	Habitat (3±0)	Ring-type (4±0)	Bruises (5±0)
Symm	Odor (1±0)	Spore-print-col. (2±0)	Stalk-surf.-ab. (3±0)	Ring-type (4±0)	Gill-size (5±0)
Shapley	Spore-print-col. (1.3±0.56)	Odor (2.3±0.5)	Ring-type (2.4±0.58)	Stalk-col.-bel. (4±0)	Habitat (5.2±0.28)
Banzhaf	Odor (1±0)	Gill-size (2.3±0.5)	Spore-print-col. (2.9±0.46)	Gill-color (4.1±0.64)	Stalk-root (5.3±0.71)

TABLA 5.11BIS. Para el conjunto de datos **Mushroom**, coincidencia en la selección de las cinco mejores características entre todos los rankers utilizados. Entre paréntesis se muestra el coeficiente de correlación de Spearman entre las características comunes.

Ranker	Class	Corr	Gain	Info	OneR	RelF	Symm	Shapley	Banzhaf
Class	--	0(--)	0(--)	1(--)	1(--)	0(--)	0(--)	1(--)	2(–1.0)
Corr		--	3(1.0)	2(1.0)	2(1.0)	1(--)	3(0.5)	1(--)	2(1.0)
Gain			--	4(0.4)	4(0.4)	3(1.0)	5(0.3)	3(0.5)	3(1.0)
Info				--	5(1.0)	3(1.0)	4(0.8)	3(0.5)	3(1.0)
OneR					--	3(1.0)	4(0.8)	3(0.5)	3(1.0)
RelF						--	3(1.0)	4(0.6)	2(1.0)
Symm							--	3(0.5)	3(0.5)
Shapley								--	2(–1.0)
Banzhaf									--

Una vez analizadas las diferencias en la selección de características entre distintos rankers implementados en Weka, *Shapley* y *Banzhaf*, el siguiente paso consistirá en analizar qué precisión se consigue con las características seleccionadas por los distintos clasificadores. En particular, se han utilizado 4 clasificadores, J48, SMO, naïve Bayes y random forest, también implementados en Weka, para evaluar las precisiones que se consiguen con las características seleccionadas por cada uno de los rankers, además, se ha incluido la precisión cuando se utilizan todas las características (variables) de los diferentes conjuntos de datos utilizados en este capítulo, para comprobar qué pérdida de precisión se obtiene cuando solo se eligen 5 características del total que tiene el conjunto de datos para hacer la predicción de las clases de la variable consecuente.

A continuación, se muestran los resultados de las precisiones obtenidas para cada uno de los conjuntos de datos. En concreto, se presenta una tabla por cada uno de los conjuntos de datos, que permite visualizar las diferencias que se obtienen en términos de precisión y, por tanto, la bondad de la selección de características para cada ranker. Las precisiones más bajas para cada clasificador están sombreadas en gris, mientras que las más altas se destacan en letra negrita. En el caso de que la precisión más alta se obtenga para el uso de todas las características de la base de datos, también se destacará en negrita la más alta de entre las precisiones obtenidas para los rankers.

TABLA 5.12. Para el conjunto de datos **Thyroid**, desempeño de 4 clasificadores para las características seleccionadas por los rankers y para todas las características. Precisión ± error absoluto medio.

Ranker	**Clasificadores**			
	J48	**SMO**	**Naïve Bayes**	**Random Forest**
Todas	**74.38±0.38**	**74.36±0.26**	73.82±0.37	73.39±0.36
Class	74.33±0.38	74.33±0.26	74.33±0.38	74.33±0.38
Corr	74.33±0.38	74.33±0.26	74.33±0.37	74.28±0.37
Gain	**74.38±0.38**	**74.35±0.26**	74.15±0.37	**74.35±0.37**
Info	74.33±0.38	74.33±0.26	74.08±0.37	74.13±0.37
OneR	74.33±0.38	**74.35±0.26**	**74.36±0.38**	74.33±0.38
RelF	73.82±0.39	73.82±0.26	73.82±0.38	73.77±0.38
Symm	74.33±0.38	74.33±0.26	74.08±0.37	74.13±0.37
Shapley	73.82±0.39	73.82±0.26	73.82±0.38	73.82±0.38
Banzhaf	**74.38±0.38**	74.33±0.26	74.33±0.38	**74.35±0.38**

Para el conjunto de datos Thyroid (véase la tabla 5.12), las diferencias entre las precisiones obtenidas por los clasificadores para cada una de las selecciones de 5 características obtenidas por los rankers son todas ellas inferiores a 0.6 puntos porcentuales, lo que significa que no se puede concluir que una selección haya sido claramente superior al resto. En este sentido, las selecciones realizadas por *Shapley* y *Banzhaf* son tan buenas como las realizadas por el resto, al menos para los cinco clasificadores utilizados. Además, para Banzhaf se obtienen las precisiones más altas para dos de los cuatro clasificadores utilizados. Finalmente, se pueden distinguir dos grupos de rankers atendiendo a las precisiones obtenidas, *Class, Corr, Gain, Info, OneR, Symm* y *Banzhaf* con precisiones algo mayores al 74% en todos los casos, y *RelF* y *Shapley* con precisiones superiores al 73% y próximas al 74%.

TABLA 5.13. Para el conjunto de datos **Healthy**, desempeño de 4 clasificadores para las características seleccionadas por los rankers y para todas las características. Precisión ± error absoluto medio.

Ranker	Clasificadores			
	J48	**SMO**	**Naïve Bayes**	**Random Forest**
Todas	**97.76±0.02**	**95.81±0.25**	**93.88±0.04**	**98.19±0.01**
Class	91.69±0.06	89.68±0.26	89.11±0.07	91.73±0.06
Corr	95.36±0.04	94.34±0.26	92.78±0.05	95.40±0.03
Gain	**96.28±0.03**	**94.85±0.26**	**93.53±0.04**	**96.30±0.03**
Info	**96.28±0.03**	**94.85±0.26**	**93.53±0.04**	**96.30±0.03**
OneR	91.69±0.06	89.68±0.26	89.11±0.07	91.74±0.06
RelF	**96.28±0.03**	**94.85±0.26**	**93.53±0.04**	**96.30±0.03**
Symm	**96.28±0.03**	**94.85±0.26**	**93.53±0.04**	**96.30±0.03**
Shapley	**96.28±0.03**	**94.85±0.26**	**93.53±0.04**	**96.30±0.03**
Banzhaf	**96.28±0.03**	**94.85±0.26**	**93.53±0.04**	**96.30±0.03**

Para el conjunto de datos Healthy (véase la tabla 5.13) todas las precisiones son muy altas, superiores al 89% en todos los casos, sin embargo, se pueden diferenciar tres grupos. Por una parte, estaría el grupo formado por *Gain, Info, RelF, Symm, Shapley* y *Banzhaf* con precisiones obtenidas con sus selecciones entre el 93.5% y más del 96%; un segundo grupo formado únicamente por Corr con precisiones entre el 92.5% y el más del 95%, y finalmente, el grupo formado por Class y OneR con precisiones entre el 89% y algo más del 91.5%. En este caso, tanto *Shapley*

como *Banzhaf* están entre los que obtienen las precisiones más altas para cada uno de los clasificadores.

TABLA 5.14. Para el conjunto de datos **Avila**, desempeño de 4 clasificadores para las características seleccionadas por los rankers y para todas las características. Precisión ± error absoluto medio.

Ranker	Clasificadores			
	J48	SMO	Naïve Bayes	Random Forest
Todas	**93.36±0.01**	**71.37±0.14**	**63.89±0.07**	**93.77±0.03**
Class	59.12±0.09	57.73±0.14	57.04±0.09	58.79±0.08
Corr	62.42±0.08	61.90±0.14	60.67±0.09	60.64±0.08
Gain	66.56±0.07	60.00±0.14	**61.00±0.09**	65.68±0.07
Info	66.56±0.07	60.00±0.14	**61.00±0.09**	65.68±0.07
OneR	62.42±0.08	61.90±0.14	60.67±0.09	60.64±0.08
RelF	**87.43±0.02**	**62.74±0.14**	59.48±0.09	**87.24±0.02**
Symm	66.56±0.07	60.00±0.14	**61.00±0.09**	65.68±0.07
Shapley	**87.43±0.02**	**62.74±0.14**	59.47±0.09	**87.24±0.02**
Banzhaf	**87.43±0.02**	**62.74±0.14**	59.47±0.09	**87.24±0.02**

En el análisis de los resultados de las precisiones obtenidas para el conjunto de datos Avila (véase la tabla 5.14) se observa que hay una mayor variabilidad en las precisiones obtenidas que en los conjuntos de datos anteriores. Es fácil observar que hay cuatro grupos, el primero formado solo por el ranker *Class*, el segundo formado por los rankers *Corr* y *OneR*, el tercero por *Gain, Info* y *Symm*, y el cuarto y último formado por *RelF*, *Shapley* y *Banzhaf*. Ninguno de estos grupos es mejor que el resto para los cuatro clasificadores utilizados. Por tanto, para ordenar estos grupos se utilizará la precisión media obtenida por los clasificadores para cada uno de los rankers. El grupo que ofrece mejor precisión media es el formado por *RelF*, *Shapley* y *Banzhaf* con un 74.22% y, además, es el que ofrece mejor desempeño para tres de los cuatro clasificadores. El segundo grupo mejor en términos de precisión sería el formado por *Gain, Info* y *Symm* con un promedio de 63.31%, y siendo el que ofrece mayor precisión para uno de los cuatro clasificadores. El tercer grupo sería el formado por *Corr* y *OneR* con una precisión media del 61.41%, y el peor grupo sería el formado por el ranker *Class* con una precisión media del 58.17%. Por tanto, *Shapley* y *Banzhaf* están entre los rankers que hacen una selección que luego ofrece mejores precisiones para los clasificadores utilizados.

TABLA 5.15. Para el conjunto de datos **Adult**, desempeño de 4 clasificadores para las características seleccionadas por los rankers y para todas las características. Precisión ± error absoluto medio.

Ranker	Clasificadores			
	J48	**SMO**	**Naïve Bayes**	**Random Forest**
Todas	**83.08±0.24**	**83.92±0.16**	80.84±0.21	82.21±0.22
Class	79.16±0.31	77.96±0.22	76.68±0.27	78.48±0.29
Corr	82.36±0.25	81.84±0.18	76.33±0.24	**82.40±0.24**
Gain	82.36±0.25	81.84±0.18	76.33±0.24	**82.40±0.24**
Info	82.36±0.25	81.84±0.18	78.89±0.22	**82.40±0.24**
OneR	79.16±0.31	77.96±0.22	76.68±0.27	78.48±0.29
RelF	82.67±0.25	**81.92±0.18**	77.72±0.24	82.21±0.24
Symm	82.37±0.18	81.85±0.18	78.89±0.22	**82.40±0.24**
Shapley	**82.39±0.25**	81.87±0.18	**81.71±0.21**	82.36±0.24
Banzhaf	**82.39±0.25**	81.87±0.18	**81.71±0.21**	82.36±0.24

Para el conjunto de datos Adult (véase la tabla 5.15), tal y como sucedía para el conjunto de datos Avila, se observa una gran variabilidad en las precisiones obtenidas por los clasificadores para cada una de las seleccciones de características de los rankers. En este caso, se podrían hacer dos agrupaciones diferentes de rankers, una para los clasificadores J48, SMO y random forest y otra para el clasificador naïve Bayes. En el primer caso, el grupo con mejores precisiones estaría formado por *Corr, Gain, Info, RelF, Symm, Shapley* y *Banzhaf*, y el grupo con peores precisiones sería el formado por *Class* y *One*R. En el segundo caso, se pueden distinguir hasta cuatro grupos, siendo el mejor de ellos, en términos de precisión, el formado por *Shapley* y *Banzhaf*, el segundo mejor el formado por *Info* y *Symm*, el tercero el formado solo por *RelF*, y el último estaría formado por *Class*, *Corr*, *Gain* y *One*R. Si se toman en consideración las precisiones medias obtenidas para los cuatro clasificadores, también se distinguen cuatro grupos, el primer grupo formado por *Shapley* y *Banzhaf* con una precisión media de alrededor del 82%, el segundo formado por *Info, RelF* y *Symm* con precisiones medias sobre el 81%, el tercero con Corr y Gain y una precisión media del 80.73% y, por último, Class y OneR con una precisión media del 78.07%. Por tanto, *Shapley* y *Banzhaf* se encuentran entre los rankers que han seleccionado las características que luego han dado mayor precisión en la aplicación de los clasificadores elegidos.

TABLA 5.16. Para el conjunto de datos **Nursery**, desempeño de 4 clasificadores para las características seleccionadas por los rankers y para todas las características. Precisión ± error absoluto medio.

Ranker	Clasificadores			
	J48	**SMO**	**Naïve Bayes**	**Random Forest**
Todas	**97.05±0.02**	**90.32±0.08**	**93.13±0.24**	**99.00±0.03**
Class	92.99±0.04	90.32±0.24	89.03±0.08	93.06±0.04
Corr	92.99±0.04	90.32±0.24	89.03±0.08	93.06±0.04
Gain	92.99±0.04	90.32±0.24	89.03±0.08	93.06±0.04
Info	92.99±0.04	90.32±0.24	89.03±0.08	93.06±0.04
OneR	92.99±0.04	90.32±0.24	89.03±0.08	93.06±0.04
RelF	92.99±0.04	90.32±0.24	89.03±0.08	93.06±0.04
Symm	92.99±0.04	90.32±0.24	89.03±0.08	93.06±0.04
Shapley	92.99±0.04	90.32±0.24	89.03±0.08	93.06±0.04
Banzhaf	92.99±0.04	90.32±0.24	89.03±0.08	93.06±0.04

El análisis del conjunto de datos Nursery (véase la tabla 5.16) no aporta gran información en cuanto a la precisión obtenida por los clasificadores, puesto que todos los rankers hicieron la misma selección de características, por lo que los resultados son completamente idénticos para todos ellos.

TABLA 5.17. Para el conjunto de datos **Bank**, desempeño de 4 clasificadores para las características seleccionadas por los rankers y para todas las características. Precisión ± error absoluto medio.

Ranker	Clasificadores			
	J48	**SMO**	**Naïve Bayes**	**Random Forest**
Todas	89.34±0.19	**89.29±0.11**	88.52±0.18	88.02±0.17
Class	89.29±0.19	**89.29±0.11**	88.87±0.17	88.82±0.17
Corr	89.26±0.19	**89.29±0.11**	89.30±0.18	89.29±0.18
Gain	**89.35±0.19**	**89.29±0.11**	88.95±0.17	89.28±0.17
Info	89.33±0.19	**89.29±0.11**	88.85±0.18	88.89±0.17
OneR	89.29±0.19	**89.29±0.11**	88.87±0.17	88.82±0.17
RelF	88.30±0.21	88.34±0.12	88.11±0.19	87.94±0.19
Symm	89.33±0.19	**89.29±0.11**	88.85±0.18	88.89±0.17
Shapley	89.27±0.19	**89.29±0.11**	**89.31±0.18**	**89.30±0.18**
Banzhaf	89.27±0.19	**89.29±0.11**	**89.31±0.18**	**89.30±0.18**

Para el conjunto de datos Bank (véase la tabla 5.17) las precisiones obtenidas por los clasificadores para cada una de las selecciones de características realizadas por los rankers son pequeñas. En concreto, todas

ellas están entre 87.94% y 89.35% por lo que se puede decir que todas las selecciones son casi igualmente buenas. No obstante, si se consideran los promedios de las precisiones, se pueden determinar dos grupos de rankers. Uno de ellos formado solo por *RelF* que es el que arroja peores resultados con un promedio de precisión del 88.17%, y otro grupo formado por el resto de rankers con unos promedios superiores al 89%. Además, *Shapley* y *Banzhaf* en particular, ofrecen las precisiones más altas para tres de los cuatro clasificadores.

TABLA 5.18. Para el conjunto de datos **HTRU2**, desempeño de 4 clasificadores para las características seleccionadas por los rankers y para todas las características. Precisión ± error absoluto medio.

Ranker	Clasificadores			
	J48	**SMO**	**Naïve Bayes**	**Random Forest**
Todas	97.09±0.05	**96.91±0.03**	95.54±0.05	96.87±0.05
Class	96.91±0.05	96.70±0.03	95.71±0.05	96.86±0.05
Corr	**97.10±0.05**	**96.91±0.03**	**95.96±0.05**	**97.08±0.05**
Gain	**97.10±0.05**	**96.91±0.03**	**95.96±0.05**	**97.08±0.05**
Info	**97.10±0.05**	**96.91±0.03**	**95.96±0.05**	**97.08±0.05**
OneR	96.91±0.05	96.70±0.03	95.71±0.05	96.86±0.05
RelF	**97.10±0.05**	**96.91±0.03**	91.74±0.08	96.92±0.05
Symm	**97.10±0.05**	**96.91±0.03**	**95.96±0.05**	**97.08±0.05**
Shapley	**97.10±0.05**	**96.91±0.03**	**95.96±0.05**	**97.08±0.05**
Banzhaf	**97.10±0.05**	**96.91±0.03**	**95.96±0.05**	**97.08±0.05**

El ánalisis del conjunto de datos HTRU2 (véase la tabla 5.18) es muy similar al que se realizó para el conjunto de datos Bank. Las diferencias entre las precisiones son muy pequeñas dentro de cada clasificador, excepto en el caso del naïve Bayes en el que destaca su peor desempeño para la selección de caracerísticas obtenidas con el ranker *RelF* con un 91.74% de precisión. Además, si se toman las precisiones promedio, se obtienen dos grupos, uno que consiste de solo el ranker *RelF* con un promedio de 95.67%, y otro con el resto de rankers con unos promedios de precisión superiores al 96.50%. Una vez más, *Shapley* y *Banzhaf* se encuentran entre los rankers para los que los clasificadores dan los mejores resultados de precisión.

TABLA 5.19. Para el conjunto de datos **Connect4**, desempeño de 4 clasificadores para las características seleccionadas por los rankers y para todas las características. Precisión ± error absoluto medio.

Ranker	Clasificadores			
	J48	**SMO**	**Naïve Bayes**	**Random Forest**
Todas	**80.90±0.17**	**75.86±0.29**	**72.14±0.27**	**82.36±0.21**
Class	67.47±0.31	66.25±0.32	66.27±0.31	67.47±0.30
Corr	68.92±0.30	66.14±0.32	68.35±0.30	69.06±0.29
Gain	66.48±0.31	66.48±0.32	66.35±0.32	66.49±0.31
Info	69.46±0.29	68.36±0.31	**69.00±0.29**	69.48±0.28
OneR	66.43±0.32	66.15±0.32	66.11±0.31	66.43±0.31
RelF	**70.09±0.29**	68.63±0.31	68.93±0.30	**70.07±0.28**
Symm	69.46±0.29	68.36±0.31	69.00±0.29	69.48±0.28
Shapley	66.23±0.32	65.83±0.32	66.08±0.32	66.21±0.32
Banzhaf	69.91±0.29	**68.85±0.31**	68.70±0.30	69.96±0.27

Para el conjunto de datos Connect4 (véase la tabla 5.19) se observa que no hay una gran variabilidad en las precisiones obtenidas por los cuatro clasificadores para cada una de las selecciones dadas por los rankers, estando todas ellas entre el 65.83% y el 70.09%, observándose las mayores diferencias para el clasificador J48 con un recorrido de 3.86 puntos porcentuales y las menores para el clasificador naïve Bayes con un recorrido de 2.92 puntos. No obstante, la ordenación de los rankers de acuerdo a la precisión, varía de un clasificador a otro, salvo el peor que siempre sería *Shapley*. Sin embargo, *Banzhaf* es el mejor ranker para el clasificador SMO y el segundo mejor para los clasificadores J48 y random forest. Si se tienen en consideración las precisiones medias obtenidas por los clasificadores para cada uno de los rankers, se observan tres grupos, el grupo con mejores precisiones está formado por *Info, RelF*, *Symm* y *Banzhaf* con unas precisiones medias superiores al 69%, el segundo grupo está formado solo por el ranker *Corr* con una precisión media del 68.12%, y el tercer y último grupo por *Class*, *Gain*, *One*R y *Shapley* con unas precisiones medias superiores al 66%. Como en todos los casos anteriores, uno de los rankers basados en teoría de juegos, en concreto *Banzhaf*, figura entre aquellos que han realizado una mejor selección de características en términos de las precisiones obtenidas posteriormente para los clasificadores utilizados.

TABLA 5.20. Para el conjunto de datos **Tic-tac-toe**, desempeño de 4 clasificadores para las características seleccionadas por los rankers y para todas las características. Precisión ± error absoluto medio.

Ranker	Clasificadores			
	J48	**SMO**	**Naïve Bayes**	**Random Forest**
Todas	**85.07±0.17**	**98.33±0.02**	69.62±0.37	**96.56±0.22**
Class	72.44±0.37	69.73±0.30	71.29±0.38	70.56±0.34
Corr	**79.44±0.26**	**75.26±0.25**	**72.44±0.37**	**84.03±0.21**
Gain	**79.44±0.26**	**75.26±0.25**	**72.44±0.37**	**84.03±0.21**
Info	**79.44±0.26**	**75.26±0.25**	**72.44±0.37**	**84.03±0.21**
OneR	72.44±0.37	69.73±0.30	71.29±0.38	70.56±0.34
RelF	**79.44±0.26**	**75.26±0.25**	**72.44±0.37**	**84.03±0.21**
Symm	**79.44±0.26**	**75.26±0.25**	**72.44±0.37**	**84.03±0.21**
Shapley	**79.44±0.26**	**75.26±0.25**	**72.44±0.37**	**84.03±0.21**
Banzhaf	**79.44±0.26**	**75.26±0.25**	**72.44±0.37**	**84.03±0.21**

En el análisis del conjunto de datos Tic-tac-toe (véase la tabla 5.20) se observan claramente dos grupos de rankers, uno con peores desempeños para todos los clasificadores, formado por *Class* y *One*R, y otro grupo, con los mejores desempeños para todos los clasificadores, formado por el resto de clasificadores. Por tanto, *Shapley* y *Banzhaf* se enceuntran en el grupo con mejores desempeños en base a las precisiones obtenidas con los clasificadores.

TABLA 5.21. Para el conjunto de datos **Credit**, desempeño de 4 clasificadores para las características seleccionadas por los rankers y para todas las características. Precisión ± error absoluto medio.

Ranker	Clasificadores			
	J48	**SMO**	**Naïve Bayes**	**Random Forest**
Todas	**86.23±0.20**	84.93±0.15	85.65±0.19	82.03±0.20
Class	84.64±0.23	**85.80±0.14**	85.65±0.19	**87.97±0.19**
Corr	**86.23±0.21**	**85.80±0.14**	**86.67±0.20**	86.67±0.20
Gain	**86.23±0.22**	**85.80±0.14**	86.09±0.19	84.78±0.20
Info	85.65±0.22	**85.80±0.14**	86.23±0.19	84.64±0.20
OneR	84.64±0.23	**85.80±0.14**	85.65±0.19	87.97±0.19
RelF	85.07±0.24	85.36±0.15	85.36±0.20	85.07±0.21
Symm	85.65±0.22	**85.80±0.14**	86.23±0.19	84.64±0.20
Shapley	84.64±0.23	**85.80±0.14**	85.65±0.19	**87.97±0.19**
Banzhaf	85.65±0.21	85.22±0.15	86.09±0.20	86.96±0.20

En el conjunto de datos Credit (véase la tabla 5.21), aunque no se observa gran variabilidad en las precisiones obtenidas por los

clasificadores para las características seleccionadas por los rankers, las precisiones van desde el 84.64% hasta el 87.97%, para cada uno de los clasificadores se observan diferentes agrupaciones de los rankers. Por ello, se consideran las precisiones medias para obtener una agrupación y ordenación de los mismos. De este modo, el grupo de rankers con las mejores precisiones medias está formado por *Class, Corr, One*R y *Shapley* con valores superiores al 86%, y el grupo con peores precisiones medias está formado por *Gain, Info, RelF, Symm* y *Banzhaf* con valores superiores al 85%. Por tanto, aunque se consideran dos grupos, estos están muy próximos, por ejemplo, la precisión media de *Banzhaf* es del 85.98%. Además, *Shapley* ofrece las mejores precisiones para dos de los clasificadores, aunque el propio *Shapley* y *Banzhaf* dan las peores precisiones para uno de los calsificadores cada uno de ellos. Todo lo indicado refuerza lo dicho sobre que la agrupación y ordenación de los rankers en términos de sus precisiones varía entre clasificadores.

TABLA 5.22. Para el conjunto de datos **Mushroom**, desempeño de 4 clasificadores para las características seleccionadas por los rankers y para todas las características. Precisión ± error absoluto medio.

Ranker	Clasificadores			
	J48	**SMO**	**Naïve Bayes**	**Random Forest**
Todas	**100.0±0.00**	**100.0±0.00**	95.83±0.04	**100.0±0.00**
Class	94.85±0.06	92.82±0.07	86.34±0.16	94.99±0.06
Corr	99.11±0.02	99.11±0.01	98.03±0.03	99.11±0.02
Gain	**100.0±0.00**	**99.90±0.00**	**98.94±0.02**	**100.0±0.00**
Info	99.90±0.00	**99.90±0.00**	95.64±0.04	99.90±0.00
OneR	99.90±0.00	**99.90±0.00**	95.64±0.04	99.90±0.00
RelF	99.90±0.00	99.80±0.00	98.03±0.03	99.90±0.00
Symm	**100.0±0.00**	**99.90±0.00**	**98.94±0.02**	**100.0±0.00**
Shapley	**100.0±0.00**	**99.90±0.00**	98.33±0.03	**100.0±0.00**
Banzhaf	99.66±0.01	99.51±0.00	98.52±0.02	99.54±0.001

Para el conjunto de datos Mushroom (véase la tabla 5.22) se observa que, salvo para *Class*, los resultados de las precisiones para todos los rankers están muy próximas, sobre todo para J48, SMO y random forest. De este modo, el clasificador que marca diferencias para hacer las agrupaciones de rankers es naïve Bayes. En base a las precisiones promedio, se observan tres grupos. El grupo con mejores precisiones medias lo forman los rankers *Gain, RelF, Symm, Shapley* y *Banzhaf* con valores superiores al 99%, el segundo grupo lo forman *Corr, Info* y *One*R

con precisiones medias del 98.84%, y el último grupo formado por *Class* con una precisión media del 92.25%. Una vez más, *Shapley* y *Banzhaf* están en el grupo con mejores precisiones, de hecho, *Shapley* es el ranker con mejor precisión en tres de los cuatro clasificadores.

Para finalizar con el análisis y discusión de los resultados, se presenta una tabla en la que se muestra cuántas veces ha sido cada uno de los rankers el que ofrece mejores precisiones con los clasificadores que se han utilizado como evaluación de la bondad de la selección de características obtenidas por los rankers. Nótese que al haber empates es posible que la suma sea superior a 44, tanto en los mejores resultados de precisión como en los peores.

TABLA 5.23. Número de veces que se ha obtenido la precisión más alta y más baja para cada uno de los clasificadores en los 11 conjuntos de datos utilizados.

Ranker	**Clasificadores**							
	J48		**SMO**		**Naïve Bayes**		**Random Forest**	
	Mejor	**Peor**	**Mejor**	**Peor**	**Mejor**	**Peor**	**Mejor**	**Peor**
Class	1	8	3	7	1	5	2	7
Corr	3	2	5	1	3	2	5	1
Gain	8	1	8	1	6	2	7	1
Info	4	1	7	1	6	1	5	2
OneR	1	6	5	5	2	3	1	5
RelF	6	3	6	3	3	5	5	3
Symm	5	1	7	1	6	1	6	2
Shapley	7	4	8	3	6	3	8	2
Banzhaf	7	1	7	2	6	1	7	1

En la siguiente tabla se presentan los porcentajes en los que cada ranker ha dado las mejores y peores precisiones en la aplicación de los clasificadores en los 11 conjuntos de datos utilizados como testbed.

TABLA 5.24. Porcentaje de veces que cada uno de los rankers ha dado mejor y peor precisión para los cuatro clasificadores.

Ranker	Class	Corr	Gain	Info	OneR	RelF	Symm	Shapley	Banzh
Mejor	15.9%	36.4%	65.9%	50.0%	20.5%	45.5%	54.5%	65.9%	61.4%
Peor	61.4%	13.6%	11.4%	11.4%	43.2%	31.8%	11.4%	27.3%	11.4%

En la tabla 5.24 se observa que *Shapley* y *Banzhaf* están entre los tres rankers que mejores resultados han ofrecido en precisión en la aplicación de los clasificadores y *Banzhaf* es, junto con otros tres rankers, el que menos veces ha sido el ranker que ha dado peores precisiones. Por tanto, los rankers basados en teoría de juegos ofrecen buenas prestaciones en la selección y ordenación de características.

5.6. Conclusiones del análisis

En este capítulo se ha realizado un experimento computacional para analizar si el diseño de rankers basados en teoría de juegos es adecuado para problemas de clasificación. Para ello se han elegido 11 conjuntos de datos con diferentes características, siete rankers y cuatro clasificadores de tipologías distintas buscando la mayor amplitud en el experimento computacional llevado a cabo. Algunas conclusiones que se pueden extraer del experimento son las siguientes.

Como se ha visto en la sección anterior, los rankers han dado lugar a selecciones diferentes de características, en general, aunque en algunos casos ha habido una gran coincidencia en la ordenación de las características y con ello en la selección de las cinco mejores. En este punto cabe destacar que cada ranker tiene sus propias características y están basados en distintos principios de búsqueda de la relevancia de las características. La principal conclusión que se puede extraer de este primer análisis es que tanto *Shapley* como *Banzhaf* han seleccionado las características claramente más relevantes cuando estas existían, tal y como han hecho la mayoría de los otros siete rankers. Por tanto, se puede concluir que ambos tienen un buen desempeño para ser tenidos en cuenta como rankers en la selección de características en los problemas de clasificación.

Para validar los resultados obtenidos en la selección de características, los resultados se han evaluado aplicando cuatro clasificadores distintos, J48, SMO, naïve Bayes y random forest. El análisis de las precisiones obtenidas muestra que tanto *Shapley* como *Banzhaf* dan los mejores resultados en la mayoría de los conjuntos de datos utilizados como testbed. Este hecho junto con que las selecciones de características no han sido completamente similares en todos los conjuntos de datos, refuerza la idea

de que la selección de características utilizando rankers basados en la teoría de juegos es una opción razonable y que merece la pena seguir profundizando en su análisis en futuras investigaciones.

A lo comentado en el párrafo anterior, cabe añadir que los rankers basados en teoría de juegos funcionan bien, incluso cuando hay muchas características y el muestreo es pequeño. En este punto es evidente que hay que tener en cuenta que el esfuerzo computacional es el mayor handicap para estos rankers basados en teoría de juegos y que un futuro desarrollo que reduzca este esfuerzo podría ser importante para que ganaran peso y relevancia en los problemas de clasificación, en particular, y en los problemas de aprendizaje automático en general.

También se puede concluir que *Shapley* y *Banzhaf* están muy igualados, y por ello, habría que analizar qué sucede si se utilizaran valores probabilísticos dando un mayor peso a las coaliciones pequeñas, es decir, a aquellos conjuntos con cinco o menos características. Además, esto permitiría reducir el esfuerzo computacional y, quizás, mejoraría la ordenación de las características. Esto, por ejemplo, es lo que se hizo en Cohen et al. (2005, 2007) donde se fijó el tamaño de las coaliciones que se iban a muestrear para calcular el valor de Shapley. Aquí se apunta a algo similar, pero utilizando valores probabilísticos.

Finalmente, los resultados obtenidos para *Shapley* y *Banzhaf* son especialmente relevantes porque no están diseñados para que los clasificadores utilizados en este capítulo se desempeñen bien en la predicción, por tanto, la selección de características que dan es, si cabe, más destacable todavía.

Como conclusión de todo el experimento comptuacional llevado a cabo, merece la pena seguir investigando sobre el uso de la teoría de juegos en problemas de clasificación porque puede dar resultados de interés para la literatura del aprendizaje automático.

Capítulo 6. Conclusiones y futuras líneas de investigación

Las conclusiones generales que se muestran a continuación, no pretenden ser una agrupación de aquellas que se han presentado de forma más minuciosa de todo lo realizado al final de todos los capítulos, sino que resultan de la consecución de objetivos descritos en el capítulo 1 de esta tesis, así como de las aportaciones realizadas a la literatura fruto de esta investigación.

6.1. Conclusiones

6.1.1. Consecución de objetivos

En concreto los objetivos marcados en esta memoria han sido encuadrados en la sección 1.1 y todos ellos han sido alcanzados. A continuación, se indica la contextualización y detalle de estos cinco objetivos fundamentales:

- Se ha estudiado de forma detallada, atendiendo al primer objetivo, el método de selección de características. Para ello se ha realizado una revisión bastante exhaustiva de la literatura existente que se refleja a lo largo de toda la memoria.
- Se ha elaborado, siguiendo las directrices del segundo objetivo, una clasificación con las diferentes aplicaciones de uso del método de selección de características más significativas. Cuyo tema ha sido abordado en el campo del aprendizaje automático en muchos estudios, algunos de ellos se han mostrado a lo largo de esta tesis que ha centrado la atención en la selección de características. De manera que ya en el capítulo 1, se han expuesto diversidad de

aplicaciones. En el capítulo 2, en concreto, en el apartado 2.5.2, se habla de los principales métodos de selección de características. En el capítulo 3, en particular en la sección 3.4, se comparan el análisis discriminante con el método de selección de características que ofrece la métrica Waci de RBS. Por supuesto, en el capítulo 4 de esta tesis, también se refleja la importancia de la selección de características, ya que antes de llevar a cabo el experimento de la sección 4.3.4 se seleccionan los 5 atributos más relevantes mediante el método de selección de características descrito en la sección 4.2.1 y los resultados se muestran en la tabla 4.8 de esa sección 4.3.4, que refleja los cinco atributos más relevantes de cada conjunto de datos, según la evaluación de atributos de la relación de ganancia. Incluso, en el capítulo 5, en la sección 5.4, se comparan métodos de selección de características implementados en Weka con los algoritmos de selección de características basado en los valores de Shapley y Banzhaf. Cuyos resultados se pueden observar en las tablas 5.1 a la 5.11 de la sección 5.5.

- Se ha estudiado la selección de características, atendiendo al tercer objetivo, mediante un modelo matemático basado en la teoría de juegos. Queda contextualizado en la sección 5.3 y 5.4. En la 5.3, se define un juego cooperativo asociado a un problema de clasificación y se calculan los valores de Shapley y Banzhaf asociados a ese juego, seleccionando aquellas características con un mayor valor por considerarse que tienen una mayor influencia en la precisión de la predicción. En la sección 5.4 se compara la selección de características mediante los valores de Shapley y Banzhaf con varios de los algoritmos de selección de características implementados en Weka.
- Se ha comparado, según el cuarto objetivo, en distintas situaciones de carga el nuevo método RBS con técnicas estadísticas clásicas como el análisis discriminante. En la sección 3.3 se realiza una explicación teórica del experimento y en la sección 3.4 se lleva a cabo la comparación de forma empírica tanto cuantitativa como cualitativa.
- Se han introducido medidas del desempeño de los clasificadores, según marcaba el quinto y último objetivo, basadas en benchmarking. En la sección 4.2 se mostraba la definición y el

análisis teórico de las medidas de rendimiento introducidas en este trabajo. Mientras que en la sección 4.3, se llevaba a cabo el experimento para explicar cómo funcionan las medidas de rendimiento y cómo pueden utilizarse para analizar el rendimiento de los clasificadores en términos de entropía.

6.1.2. Aportaciones a la literatura de aprendizaje automático

En cuanto a las aportaciones de esta investigación a la literatura, a modo de resumen, se pueden destacar las siguientes:

- Se ha probado, en el capítulo 3, que el nuevo método RBS permite alcanzar mayor precisión en la clasificación que la obtenida por el análisis discriminante. Además, computacionalmente RBS es muy interesante para la selección automática de características.
- Se han propuesto, en el capítulo 4, un clasificador aleatorio y dos intuitivos como clasificadores de referencia, los cuales captan la entropía del conjunto de datos. También se definen tres nuevas medidas de rendimiento de los clasificadores basadas en la π de Scott, la precisión de los clasificadores y los clasificadores de referencia propuestos. Dichas métricas miden en cuánto mejora la clasificación un determinado clasificador con respecto a los de referencia, siendo posible compararlos.
- Se ha demostrado, en el capítulo 5, que los rankers basados en teoría de juegos son adecuados para problemas de clasificación, incluso cuando hay muchas características. En la literatura hay variedad de trabajos sobre el valor de Shapley que muestran el interés sobre el análisis de las características en los modelos de clasificación; pero no hay tantas investigaciones sobre el valor de Banzhaf. Este trabajo aúna un estudio realizado sobre el valor de Shapley y el valor de Banzhaf comparativamente.

6.2. Futuras líneas de investigación

La enumeración de las futuras líneas de investigación que en esta sección se recopilan a modo de resumen de las recogidas en secciones anteriores, muestran aquellos aspectos que, aunque a priori, no se tenía

previsto contemplar, resultan ser muy interesantes de desarrollar en trabajos posteriores.

- Es necesario incorporar dentro del propio algoritmo RBS un método de discretización de variables numéricas, en lugar de ser un paso en la etapa de preprocesamiento. También se tiene presente mejorar el tiempo computacional cuando se aplique RBS en conjuntos de datos excesivamente grandes.
- Se deberá seguir investigando en la selección de características utilizando rankers basados en la teoría de juegos.
- Se ve conveniente reducir el esfuerzo computacional para los rankers basados en teoría de juegos, ya que podría ser un condicionante para que ganaran protagonismo en los problemas de clasificación.
- Se deberá investigar en próximos trabajos qué sucede con los valores de Shapley y Banzhaf si se utilizaran valores probabilísticos dando un mayor peso a las coaliciones pequeñas.

Bibliografía y referencias

[1] Aas, K.; Jullum, M.; Løland, A. Explaining Individual Predictions when Features are Dependent: more Accurate Approximations to Shapley Values. arXiv:1903.10464v3 [stat.ML]. 2020.

[2] Afghah, F.; Razi, A.; Soroushmehr, R.; Ghanbari, H.; Najarian, K. Game Theoretic Approach for Systematic Feature Selection; Application in False Alarm Detection in Intensive Care Units. Entropy. 2018, 20, 190.

[3] Aggarwal, C.C. Data Mining: The Textbook. Springer. 2015.

[4] Aha, D.W.; Kibler, D.; Albert, M.K. Instance-Based Learning Algorithms. Machine Learning. 1991, 6, 37–66.

[5] Aharonov, R.; Segev, L.; Meilijson, I.; Ruppin, E. Localization of Function via Lesion Analysis. Neural Computation. 2003, 15, 885-913.

[6] Algaba, E.; Fragnelli, V.; Sanchez-Soriano, J. (eds) Handbook of the Shapley Value. Series in Operations Research. CRC Press. 2019a.

[7] Algaba, E.; Fragnelli, V.; Sanchez-Soriano, J. The Shapley Value, a Paradigm of Fairness. In Handbook of the Shapley Value. Algaba, E., Fragnelli, V., Sanchez-Soriano, J. (eds). Series in Operations Research. CRC Press, Chapter 2. 2019b, 17-29.

[8] Algaba, E.; Béal, S.; Fragnelli, V.; Lloca, N.; Sanchez-Soriano, J. Relationship between Labeled Network Games and other Cooperative Games Arising from Attributes Situations. Economics Letters. 2019c, 185, 108708.

[9] Alloghani, M.; Aljaaf, A.; Hussain, A.; Baker, T.; Mustafina, J.; Al-Jumeily, D.; Khalaf, M. Implementation of Machine Learning Algorithms to Create Diabetic Patient re-Admission profiles. BMC Medical Informatics and Decision Making. 2019, 19, 253.

[10] Almiñana, M.; Escudero, L.F.; Pérez-Martín, A.; Rabasa, A.; Santamaría, L. A Classification Rule Reduction Algorithm Based on Significance Domains. TOP. 2012, 22, 367-416.

[11] Amigó, J.M.; Balogh, S.G.; Hernández, S. A. Brief Review of Generalized Entropies. Entropy. 2018, 20, 813.

[12] Ao, YL; Li, HQ; Zhu, L.; Ali, S.; Yang, ZG. Identifying Channel Sand-Body from Multiple Seismic Attributes with an Improved Random Forest Algorithm. Journal of Petroleum Science and Engineering. 2019, 173, 781-792.

[13] Aremu, O.O.; Cody, R.A.; Hyland-Wood, D.; McAree, P.R. A Relative Entropy Based Feature Selection Framework for Asset data in Predictive Maintenance. Computers & Industrial Engineering. 2020, 145, 106536.

[14] Arista-Jalife, A.; Calderón-Auza, G.; Fierro-Radilla, A.; Nakano, M. Clasificación de Imágenes Urbanas Aéreas: Comparación entre Descriptores de Bajo Nivel y Aprendizaje Profundo. Información Tecnológica. 2017, 28 (3), 209-224.

[15] Bai, L.; Han, Z.; Ren, J.; Qin, X. Research on Feature Selection for Rotating Machinery Based on Supervision Kernel Entropy Component Analysis with Whale Optimization Algorithm. Applied Soft Computing. 2020, 92, 106245.

[16] Banzhaf, J.F. Weighted Voting Doesn't Work. Rutgers Law Review. 1965, 19, 317-343.

[17] Bax, J.J.; Van Der Bijl, P.; Delgado, V. Machine Learning for Electrocardiographic Diagnosis of Left Ventricular Early Diastolic Dysfunction, Journal of the American College of Cardiology. 2018, 71 (15), 1661-1662.

[18] Bayes, T. An Essay Towards Solving a Problema in the Doctrine of Chances. Philosophical Transactions. 1763, 53, 370-418.

[19] Ben-Hur, A.; Horn, D.; Siegelmann, H.; Vapnik, V.N. Support Vector Clustering. Journal of Machine Learning Research. 2001, 2, 125-137.

[20] Berezinski, P.; Jasiul, B.; Szpyrka, M. An Entropy-Based Network Anomaly Detection Method. Entropy. 2015, 17, 2367-2408.
[21] Breiman, L. Random Forests. Machine Learning. 2001, 45, 5–32.
[22] Breiman, L.; Friedman, J.H.; Olshen, R.A.; Stone, C.J. Classification and Regression Trees. Wadsworth International Group. 1984.
[23] Brodley C. E.; Utgoff P. E. Multivariate versus Univariate Decision Trees. Technical Report UM-CS. 1992, 92, 8.
[24] Brown, G.; Pocock, A.; Zhao, M.J.; Luján, M. Conditional Likelihood Maximisation: A Unifying Framework for Information Theoretic Feature Selection. Journal of Machine Learning Research. 2012, 13, 27-66.
[25] Cambronero, C. G.; Moreno, I. G. Algoritmos de Aprendizaje: Knn & Kmeans. Inteligencia en Redes de Comunicación, Universidad Carlos III de Madrid. 2006, 23.
[26] Cardillo, G.P.; Fu, K.S. Divergence Furthermore, Linear Classifiers for Feature Selection. IEEE Transactions Automatic Control. 1967, 780.
[27] Castro, J.; Gomez, D.; Tejada, J. Polynomial Calculation of the Shapley Value Based on Sampling. Computers & Operations Research. 2009, 36, 1726-1730.
[28] Castro, J.; Gomez, D.; Molina, E.; Tejada, J. Improving Polynomial Estimation of the Shapley Value by Stratified Random Sampling with Optimum Allocation. Computers & Operations Research. 2017, 82, 180-188.
[29] Cerda y Cifuentes. Uso de Curvas ROC en Investigación Clínica. Aspectos Teórico-Prácticos Revista Chilena de Infectología. 2012, 29 (2), 138-141.
[30] Chandrashekar, G.; Sahin, F. A Survey on Feature Selection Methods. Computers & Electrical Engineering. 2014, 40, 16–28.
[31] Chen, C.; Xu, HG. Color Recognition Based on C4.5 Decision Tree Algorithm. International Conference on Computer Engineering and Information System. 2016, 52 , 158-163.
[32] Cheng, R.J.; Yu, W.; Song, Y.; Chen, D.; Ma, X.; Cheng, Y. Intelligent Safe Driving Methods Based on Hybrid Automata and

Ensemble CART Algorithms for Multihigh-Speed Trains. IEEE Transactions on Cybernetics. 2019, 49(10), 3816-3826.

[33] Cheung, C.F.; Li, F.L. A Quantitative Correlation Coefficient Mining Method for Business Intelligence in Small and Medium Enterprises of Trading Business. Expert Systems with Applications. 2012, 39, 6279-6291.

[34] Chien, Y.T. Adaptive Strategies of Selecting Feature Subsets in Pattern Recognition. In Proceedings of the IEEE Symposium on Adaptive Processes (8th) Decision and Control. 1969, 8, 36.

[35] Chu, C.C.F.; Chan, D.P.K. Feature Selection Using Approximated High-Order Interaction Components of the Shapley Value for Boosted Tree Classifier. IEEE ACCESS. 2020, 8, 112742-112750.

[36] Cleary, J.G.; Trigg, L.E. K*: An Instance-Based Learner Using an Entropic Distance Measure. Machine Learning Proceedings. 1995, 108-114.

[37] Cohen, J. A Coefficient of Agreement for Nominal Scales. Educational and Psychological Measurement. 1960, 20, 37-46.

[38] Cohen, M.J.; Lane, Ch.; Reiss, K.C.; Surdiz, J.A.; Bardi, E.; Brown, M.T. Vegetation Based Classification Trees for Rapid Assesment of Isolated Wetland Condition. Elsevier. Ecological Indicators. 2005, 5, 189-206.

[39] Cohen, S.; Dror, G.; Ruppin, G. Feature Selection Based on the Shapley Value. In Proceedings of the Nineteenth (19th) International Joint Conference on Artificial Intelligence. 2005, 665-670.

[40] Cohen, S.; Dror, G.; Ruppin, G. Feature Selection via Coalitional Game Theory. Neural Computation. 2007, 19, 1939-1961.

[41] Cortes, C.; Vapnik, V. Support Vector Networks. Machine Learning. 1995, 20, 273-297.

[42] Costa, E.P.; Lorena, A.C.; Carvalho, A.C.; Freitas, A.A. A Review of Performance Evaluation Measures for Hierarchical Classifiers. In Proceedings of the AAAI-07 Workshop Evaluation Methods for Machine Learning II. 2007, 1-6.

[43] Cover, T.M.; Hart, P.E. Nearest Neighbor Pattern Classification. IEEE Transaction on Information Theory. 1967, 13, 21-27.

[44] Dasarathy, B.V. Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques. IEEE Computer Society Press. 1991.
[45] Davila-Pena, L.; García-Jurado, I.; Casas-Méndez, B. Assessment of the Influence of Features on a Classification Problem: An Application to COVID-19 Patients. European Journal of Operational Research. 2022, 299, 631-641.
[46] Deng, HX; Diao, YF; Wu, W.; Zhang, J.; Ma, M.; Zhong, X. A High-Speed D-CART Online Fault Diagnosis Algorithm for Rotor Systems. Applied Intelligence. 2020, 50(1), 29-41.
[47] Deng, X.; Papadimitriou, C.H. On the Complexity of Cooperative Solution Concepts. Mathematics of Operations Research. 1994, 19(2), 257-266.
[48] de Stefano, C.; Maniaci, M.; Fontanella, F.; di Freca, A.S. Reliable Writer Identification in Medieval Manuscripts through Page Layout Features: The "Avila" Bible Case. Engineering Applications of Artificial Intelligence. 2018, 72, 99–110.
[49] Dua, D.; Graff, C. UCI Machine Learning Repository. University of California, Irvine, School of Information and Computer Sciences. 2019. Disponible en línea: http://archive.ics.uci.edu/ml (consultado el 23 de abril de 2021).
[50] Duch, W.; Wieczorek, T.; Biesiada, J.; Blachnik, M. Comparison of Feature Ranking Methods Based on Information Entropy. IEEE International Joint Conference on Neural Networks, (IEEE Cat. No.04CH37541). 2004, 2, 1415-1419.
[51] Eesa, AS; Orman, Z; Brifcani, AMA. A New Feature Selection Model Based on ID3 and Bees Algorithm for Intrusion Detection System. Turkish Journal of Electrical Engineering and Computer Sciences. 2015, 23 (2), 615-622.
[52] Fan, Q.; Wang, Z.; Li, D.; Gao, D.; Zha, H. Entropy-Based Fuzzy Support Vector Machine for Imbalanced Datasets. Knowledge-Based Systems. Elsevier. 2017, 115, 87–99.
[53] Ferrer-Troyano, F.J.; Aguilar-Ruiz, J.S.; Riquelme, J.C. Non-Parametric Nearest Neighbor with Local Adaptation. In 10th Portuguese Conference on Artificial Intelligence. EPIA 2001. Springer-Verlag. 2001, 2258, 22-29.

[54] Ferri, C.; Hernández-Orallo, J.; Modroiu, R. An Experimental Comparison of Performance Measures for Classification. Pattern Recognition Letters. 2009, 30, 27-38.
[55] Fisher, R.A. The Use of Multiple Measurements in Taxonomic Problems. Annals of Eugenics. 1936, 7(2), 179-184.
[56] Frank, E.; Hall, M.A.; Witten, I.H. The WEKA Workbench, Apéndice en línea para Data Mining: Practical Machine Learning Tools and Techniques. Fourth Edition. Morgan Kaufmann Publishers. 2016.
[57] Fryer, D.; Strümke, I.; Nguyen, H. Shapley Values for Feature Selection: The Good, the Bad, and the Axioms. arXiv:2102.10936[cs.LG]. 2021.
[58] Fu, K.S.; Cardillo, G.P. An Optimum Finite Sequential Procedure For Feature Selection Furthermore, Pattern Classification. IEEE Transactions Automatic Control. 1967, 12, 588.
[59] Fujino, A.; Ueda, N.; Saito, K. Semisupervised Learning for a Hybrid Generative/Discriminative Classifier Based on the Maximum Entropy Principle. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2008, 30, 424-437.
[60] Fukunaga, K. Introduction to Statistic al Pattern Recognition. Second Edition. Academic Press. 1990.
[61] Giráldez, R. Mejoras en Eficiencia y Eficacia de Algoritmos Evolutivos para Aprendizaje Supervisado. Tesis Doctoral, Universidad de Sevilla. 2003.
[62] Gong, W.; Wang. WS. Application Research of Support Vector Machine in E-Learning for Personality. IEEE International Conference on Cloud Computing and Intelligence Systems. 2011, 638-642.
[63] Goodman, L.A.; Kruskal, W.H. Measures of Association for Cross Classifications. Journal of the American Statistical Association 1954, 49, 732–764.
[64] Guha, R.; Khan, H.A.; Singh, P.K.; Sarkar, R.; Bhattacharjee, D. CGA: a New Feature Selection Model for Visual Human Action Recognition. Neural Computing and Applications. 2021, 33, 5267-5286.

[65] Hastie, T.; Tibshirani, R. Classification by Pairwise Coupling. In Advances in Neural Information Processing Systems. 1998, 507-513.
[66] Ho, T.K. Random Decision Forests. In Proceedings of the 3rd International Conference on Document Analysis and Recognition. 1995, 1, 278–282.
[67] Ho, T.K. The Random Subspace Method for Constructing Decision Forests'. IEEE Transactions Pattern Analysis Machine Intelligence. 1998, 20, 832–844.
[68] Holub, A.; Perona, P.; Burl, M.C. Entropy-Based Active Learning for Object Recognition. In Proceedings of the 2008 IEEE Computer Society Entropy-Based Active Learning for Object Recognition Conference on Computer Vision and Pattern Recognition Workshops. 2008, 1-8.
[69] Hultström K. Image Based Wheel Detection Using Random Forest Classification. Tesis doctoral, Universidad de Lund. 2013.
[70] Jiao, Y.; Du, P. Performance Measures in Evaluating Machine Learning Based Bioinformatics Predictors for Classifications. Quantitative Biology. 2016, 4, 320-330.
[71] John, G.; Kohavi, R.; Pfleger, K. Irrelevant Features and the Subset Selection Problem. In Proceedings of the Fifth (5th) International Conference on Machine Learning. 1994, 121-129.
[72] John, G.H.; Langley, P. Estimating Continuous Distributions in Bayesian Classifiers. In Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence. 1995, 338-345.
[73] Jothi, N.; Husain, W.; Rashid, N.A.A. Predicting Generalized Anxiety Disorder Among Women Using Shapley Value. Journal of Infection and Public Health. 2021, 14, 103-108.
[74] Jurs, P.C. Mass Spectral Feature Selection and Structural Correlations Using Computerized Learning Machines. Analytical Chemistry. 1970, 42, 1633–1638.
[75] Jurs, P.C.; Kowalski, B.R.; Isenhour, T.L.; Reilley, C.N. Computerized Learning Machines Applied to Chemical Problems. Convergence Rate and Predictive Ability of Adaptive Binary Pattern classifiers. Analytical Chemistry. 1969, 41, 690–695.

[76] Keerthi, S.S.; Shevade, S.K.; Bhattacharyya, C.; Murthy, K.R.K. Improvements to Platt's SMO Algorithm for SVM Classifier Design, Neural Computation. 2001, 13, 637–649.
[77] Keinan, A.; Sandbank, B.; Hilgetag, C.C.; Meilijson, I.; Ruppin, E. Fair Attribution of Functional Contribution in Artificial and Biological Networks. Neural Computation. 2004, 16, 1887-1915.
[78] Keinan, A.; Sandbank, B.; Hilgetag, C.C.; Meilijson, I.; Ruppin, E. Axiomatic Scalable Neurocontroller Analysis via the Shapley Value. Artificial Life. 2006, 12(3), 333–352.
[79] Kelleher, J.D.; Namee, B.M.; D'Arcy, A. Fundamentals of Machine Learning for Predictive Data Analytics: Algorithms, Worked Examples, and Case Studies. The MIT Press. 2015.
[80] Kohavi, R.; John, G.H. Wrappers for Feature Subset Selection. Artificial Intelligence. 1997, 97(1-2), 273–324.
[81] Kohavi, R.; Provost, F. Glossary of Terms. Machine Learning Special Issue on Applications of Machine Learning and the Knowledge Discovery Process. Machine Learning. 1998, 30, 271-274.
[82] Kubat, M. An Introduction to Machine Learning. Second Edition. Springer. 2017.
[83] Kumar, I.E.; Venkatasubramnian, S.; Scheidegger, C.; Friedler, S.A. Problems with Shapley-Value-Based Explanations as Feature Importance Measures. arXiv:2002.11097v2 [cs.AI]. 2020.
[84] Labatut, V.; Cherifi, H. Evaluation of Performance Measures for Classifiers Comparison, arXiv preprint arXiv:1112.4133. 2011.
[85] Langley, W.I.; Thompson, K. An analysis of Bayesian Classifiers. In Proceedings of the AAAI-94, 1994. MIT Press. 1994, 223–228.
[86] Leardi, R.; Boggia, R.; Terrile, M. Genetic Algorithms as a Strategy for Feature Selection. Journal of Chemometric. 1992, 6, 267–281.
[87] Lê Cao, K.A.; Boitard, S.; Besse, P. Sparse PLS Discriminant Analysis: Biologically Relevant Feature Selection and Graphical Displays for Multiclass Problems. BMC Bioinformatics. 2011, 12(253), 1-16.
[88] Lee, H.M.; Chen, C.M.; Chen, J.M.; Jou, Y.L. An Efficient Fuzzy Classifier with Feature Selection Based on Fuzzy Entropy. IEEE

Transitions System Man Cybern. Part B Cybern. 2001, 31, 426-432.
[89] Lin, F.; Liang, D.; Chen, E. Financial Ratio Selection for Business Crisis Prediction. Expert Systems with Applications. 2011, 38, 15094-15102.
[90] Liu, B. Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data. Second Edition. Springer. 2011, 622.
[91] Liu, H.; Yu, L. Toward Integrating Feature Selection Algorithms for Classification and Clustering. IEEE Transactions on Knowlegde and Data Engineering. 2005, 17, 491–502.
[92] Liu H.; Setiono R. Feature Selection and Classification: a Probabilistic Wrapper Approach. In Proceedings of the IEA-AIE. 1996.
[93] Liu, XH.; Wang, EX.; Zheng, YQ. Random Forest Algorithm Optimization of Enterprise Financial Information Management System. Latin American Applied Research. 2018, 48 (4), 255-260.
[94] Lu, K.; Chen, MR. Research on Application of C4.5 Algorithm in Performance Analysis. 6th International Conference on Electronics, Mechanics, Culture and Medicine (EMCM). 2016, 45, 290-294.
[95] Lundberg, S.M.; Lee, S.I. A Unified Approach to Interpreting Model Predictions. 31st Conference on Neural Information Processing Systems. 2017, 30.
[96] Lyon, J.; Stappers, B.W.; Cooper, S.; Brooke, J.M.; Knowles, J.D. Fifty Years of Pulsar Candidate Selection: From Simple Filters to a New Principled Real-Time Classification Approach. Monthly Notices of the Royal Astronomical Society. 2016, 459, 1104–1123.
[97] Lyon, R.J. HTRU2. Disponible en línea: https://doi.org/10.6084/m9.figshare.3080389.v1 (consultado el 23 de abril de 2021).
[98] Ma, S.; Tourani, R. Predictive and Causal Implications of using Shapley Value for Model Interpretation, arXiv:2008.05052v1 [cs.LG]. 2020.
[99] Maleki, S.; Tran-Thanh L.; Hines, G.; Rahwan, T.; Rogers, A. Bounding the Estimation Error of Sampling-Based Shapley Value

Approximation. Computer Science and Game Theory, arXiv:1306.4265v2[cs.GT]. 2014.

[100] Mann, I.; Shapley, L.S. Values of Large Games, IV: Evaluating the Electoral College by Montecarlo Techniques. Rand Corporation. 1960.

[101] Mannor, S.; Peleg, D.; Rubinstein, R. The Cross Entropy Method for Classification. In Proceedings of the 22nd International Conference on Machine Learning. Association for Computing Machinery. 2005, 561-568.

[102] McLachlan, G.J. Discriminant Analysis and Statistical Pattern Recognition. Wiley Interscience. 2004.

[103] Manterola, C. Cómo Interpretar un Artículo sobre Pruebas Diagnósticas. Revista Médica Clínica las Condes. 2009, 20(5) 708-717.

[104] Meddouri, N.; Khoufi, H.; Maddouri, M. Parallel Learning and Classification for Rules Based on Formal Concepts. Procedia Computer Science. 2014, 35, 358-367.

[105] Miao, J.; Niu, L. A Survey on Feature Selection. Procedia Computer Science. 2016, 91, 919- 926.

[106] Michalski, R.S. Discovering Classification Rules Using Variable-Valued Logic System VL1. In Proceedings of the Third (3rd) International Joint Conference on Artificial Intelligence. 1973, 162-172.

[107] Mitra, P.; Murthy, C.A.; Pal, S.K. Unsupervised Feature Selection Using Feature Similarity. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2002, 24(3), 301-312.

[108] Monzo, C.; Alías, F.; Morán J.A.; Gonzalvo, X. Phonetic Transcription of Acronyms in Spanish Using the C4.5 Algorithm. Procesamiento del Lenguaje Natural. 2006, 37, 275-282.

[109] Moro, S.; Cortez, P.; Rita, P. A Data-Driven Approach to Predict the Success of Bank Telemarketing. Decision Support Systems. 2014, 62, 22-31.

[110] Narendra, P.M.; Fukunaga, K. A Branch and Bound Algorithm for Feature Subset Selection. IEEE Transactions on Computers. 1977, 26(9), 917–922.

[111] Orenes, Y.; Rabasa, A.; Rodríguez-Sala, J.J.; Sánchez-Soriano, J. Benchmarking Analysis of the Accuracy of Classification Methods Related to Entropy. Entropy. 2021, 23(7), 850.

[112] Orenes, Y.; Rabasa, A.; Pérez-Martín, A.; Rodríguez-Sala, J.J.; Sánchez-Soriano, J. A Computational Experience For Automatic Feature Selection on Big Data Frameworks. International Journal of Design Nature and Ecodynamics. 2016, 11, 168–177.

[113] Parker, C. An Analysis of Performance Measures for Binary Classifiers. In Proceedings of the 2011 IEEE 11th International Conference on Data Mining. 2011, 517-526.

[114] Peng, H.C.; Long, F.H.; Ding, C. Feature Selection Based on Mutual Information: Criteria of Max-Dependency, Max-Relevance, and Minredundancy. IEEE Transaction on Pattern Analysis and Machine Intelligence. 2005, 27, 1226–1238.

[115] Platt, J. Fast Training of Support Vector Machines Using Sequential Minimal Optimization. In Advances in Kernel Methods-Support Vector Learning. MIT Press. 1998.

[116] Pradana, AC; Adiwijaya, Aditsania, A. Implementing Binary Particle Swarm Optimization and C4.5 Decision Tree for Cancer Detection Based on Microarray Data Classification. 2nd International Conference on Data and Information Science. 2019.

[117] Pudil, P.; Novovicova, J.; Kittler, J. Floating Search Methods in Feature-Selection. Pattern Recognition Letters. 1994, 15(11), 1119-1125.

[118] Qu, Y.; Li, R.; Deng, A.; Shang, C.; Shen, Q. Non-Unique Decision Differential Entropy-Based Feature Selection. Neurocomputing. 2020, 393, 187-193.

[119] Quinlan, J.R. Discovering Rules by Induction from Large Collections of Examples. In: Michie, D. (ed.), Expert Systems in the Micro-Electronic Age. Edinburgh University Press. 1979, 168-201.

[120] Quinlan, J.R. Induction of Decision Tree. Machine Learning. 1986, 1, 81-106.

[121] Quinlan, J.R. C4.5: Programs for Machine Learning. First Edition. Morgan Kaufmann Publishers. 1992.

[122] Rabasa A. Método para la Reducción de Sistemas de Reglas de Clasificación por Dominios de Significancia. Tesis Doctoral, Universidad Miguel Hernández de Elche. 2009.

[123] Rabasa, A.; Compañ, A.F.; Agulló, A.J.; Rodríguez-Sala, J.J.; Santamaría, L.; Noguera, L. Data management for an Anaesthesiology Department Optimization. WIT Transactions on Information and Communication Technologies. WIT Press. 2013, 45, 175-183.

[124] Rahman, M.A.; Khanam, F.; Ahmad, M. Multiclass EEG signal Classification Utilizing Rényi min-Entropy-Based Feature Selection from Wavelet Packet Transformation. Brain Informatics. 2020, 7, 1-11.

[125] Ramírez-Gallego, S.; García, S.; Herrera, F. Online Entropy-Based Discretization for Data Streaming Classification. Future Generation Computer System. 2018, 86, 59-70.

[126] Ramos, D.; Franco-Pedroso, J.; Lozano-Diez, A.; Gonzalez-Rodriguez, J. Deconstructing Cross-Entropy for Probabilistic Binary Classifiers. Entropy. 2018, 20, 208.

[127] Rényi, A. On Measures of Entropy and Information. In Proceedings of the 4th Berkeley Symposium on Mathematics, Statistics and Probability. 1961, 4, 547-561.

[128] Revanasiddappa, M.B.; Harish, B.S. A New Feature Selection Method Based on Intuitionistic Fuzzy Entropy to Categorize Text Documents. International Journal of Interactive Multimedia and Artificial Intelligence. 2018, 5, 106–117.

[129] Rodríguez-Sala J.J. Método para Generación y Ordenación de Reglas de Clasificación. Diseño y Estudio Computacional. Aplicación a la Inteligencia de Negocio. Tesis Doctoral, Universidad Miguel Hernández de Elche. 2014.

[130] Romeo, V.; Cuocolo, R.; Ricciardi, C.; Ugga, L.; Cocozza, S.; Verde, F.; Stanzione, A.; Napolitano, V.; Russo, D.; Improta, G.; Elefante, A.; Staibano S., Brunetti A. Prediction of Tumor Grade and Nodal Status in Oropharyngeal and Oral Cavity Squamous-cell Carcinoma Using a Radiomic Approach. Anticancer Res. 2020, 40(1), 271-280.

[131] Roth, A.E. (ed.) The Shapley Value, Essays in Honor of Lloyd S. Shapley. Cambridge University Press. 1988.
[132] Scott, W.A. Reliability of Content Analysis: The Case of Nominal Scale Coding. Public Opinion Quarterly. 1955, 19, 321-325.
[133] Shannon, C.E. A Mathematical Theory of Communication. Bell System Technical Journal. 1948, 27, 379-423.
[134] Shapley, L.S. A Value for n-Person Games. In Contributions to the Theory of Games II. Kuhn H.W., Tucker A.W. (eds). Annals of Mathematics Studies. Princeton University Press. 1953, 307-317.
[135] Shinmoto Torres, R.L.; Ranasinghe, D.C.; Shi, Q.; Sample, A.P. Sensor Enabled Wearable RFID Technology for Mitigating the Risk of Falls Near Beds. In Proceedings of the 2013 IEEE International Conference on RFID. 2013.
[136] Siedlecki, W.; Sklansky, J. A Note on Genetic Algorithms for Large-Scale Feature-Selection. Pattern Recognition Letters. 1989, 10, 335–347.
[137] Skansi, S. Introduction to Deep Learning. From Logical Calculus to Artificial Intelligence. Undergraduate Topics in Computer Science. Springer. 2018.
[138] Skiena, S.S. The Data Science Design Manual. Springer. 2017.
[139] Sokolova, M.; Lapalme, G. A Systematic Analysis of Performance Measures for Classification Tasks, Information Processing & Management. 2009, 45, 427-437.
[140] Stone, M. Cros-Validatory Choice and Assessment of Statistical Predictions. Journal of the Royal Statistical Society B. 1974, 36, 111-147.
[141] Strumbelj, E.; Kononenko, I. An Efficient Explanation of Individual Classifications using Game Theory. Journal of Machine Learning Research. 2010, 11, 1-18.
[142] Strumbelj, E.; Kononenko, I. Explaining prediction models and individual predictions with feature Contributions. Knowledge and Information Systems. 2014, 41, 647-665.
[143] Sun, JY; Zhong, GQ; Dong, J.; Saeeda, H.; Zhang, Q. Cooperative Profit Random Forests with Application in Ocean front Recognition. IEEE ACCESS. 2017, 5, 398-1408.

[144] Team, R.C. R: A language and environment for statistical computing. R Foundation for Statistical Computing. 2013. Disponible en http://www.r-project.org/

[145] Trabelsia, M.; Meddouria, N.; Maddourib, M. A New Feature Selection Method for Nominal Classifier based on Formal Concept Analysis. Procedia Computer Science. 2017, 112, 186-194.

[146] Tripathi, S.; Hemachandra, N.; Trivedi, P. On Feature Interactions Identified by Shapley Values of Binary Classification Games. arXiv:2001.03956v1 [stat.ML]. 2020.

[147] Tsallis, C. Possible Generalization of Boltzmann-Gibbs statistics. Journal of Statistial Physics. 1988, 52, 479-487.

[148] Tumer, K.; Ghosh, J. Estimating the Bayes Error Rate through Classifier Combining. In Proceedings of the 13th International Conference on Pattern Recognition. 1996, 2, 695-699.

[149] Valverde-Albacete, F.J.; Peláez-Moreno, C. Two Information-Theoretic Tools to Assess the Performance of Multi-class classifiers. Pattern Recognition Letters. 2010, 31, 1665-1671.

[150] Valverde-Albacete, F.J.; Peláez-Moreno, C. 100% Classification Accuracy Considered Harmful: The Normalized Information Transfer Factor Explains the Accuracy Paradox. PLoS ONE. 2014, 9, 10.

[151] Valverde-Albacete, F.J.; Peláez-Moreno, C. The Evaluation of Data Sources Using Multivariate Entropy Tools. Expert Systems with Applications. 2017, 78, 145-157.

[152] Valverde-Albacete, F.J.; Peláez-Moreno, C. A Framework for Supervised Classification Performance Analysis with Information-Theoretic Methods. IEEE Transactions on Knowledge. Data Engineering. 2020, 32, 2075-2087.

[153] Vapnik, V.; Golowich, S.; Smola, A. Support Vector Method for Function Approximation, Regression Estimation and Signal Processing. Advances in Neural Information Processing Systems. 1996, 9.

[154] Venables, W.N.; Ripley, B.D. Modern Applied Statistics with S. Fourth Edition. Springer. 2002.

[155] Wang, J.; Xu, S.; Duan, B.; Liu, C.; Liang, J. An Ensemble Classification Algorithm Based on Information Entropy for Data Streams. Neural Processing Letters. 2019, 50, 2101-2117.

[156] Wang, L. College English Network Independent Learning Based on ID3 Algorithm. Agro Food Industry Hi-Tech. 2017, 28 (1), 3342-3345.

[157] Wang, YB. Prediction of Rockburst Risk in Coal Mines Based on a Locally Weighted C4.5 Algorithm. IEEE ACCESS. 2021, 9, 15149-15155.

[158] Weber, R. J. Probabilistic Values for Games. In Roth A. E., ed., The Shapley Value. Essays in Honor of L. S. Shapley. Cambridge University Press. 1988, 101-119.

[159] Weka. Disponible en línea: http://ocw.uc3m.es/ingenieria-Iinformatica/Herramientas-de-la-Inteligencia-Artificial/Contenidos/Transparencias/TutorialWeka.pdf (último acceso el 9 de marzo de 2020).

[160] Weka, Waikato Environment for Knowledge Analysis. Machine Learning Group at the University of Waikato: New Zealand. Disponible en http:/Q/www.cs.waikato.ac.nz/ml/weka/. (último acceso, 15 de junio de 2021)

[161] Williams, P.L.; Beer, R.D. Nonnegative Decomposition of Multivariate Information, arXiv:1004.2515v1 [cs.IT]. 2010.

[162] Wittekind, C.; Tischoff, I. Tumor Classifications. Pathologe. 2004, 25 (6) , 481-490.

[163] Witten, I.H.; Frank, E. Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. Elsevier. 2005.

[164] Yadav, A.K.; Chandel, S.S. Solar Energy Potential Assessment of Western Himalayan Indian State of Himachal Pradesh Using J48 Algorithm of WEKA in ANN Based Prediction Model. Renewable Energy. 2015, 75, 675–693.

[165] Yang, J.H.; Honavar, V. Feature Subset Selection Using a Genetic Algorithm. IEEE Intelligent Systems and their Applications. 1998, 13 (2), 44-49.

[166] Yang, S; Guo, JZ; Jin, JW. An Improved ID3 Algorithm for Medical Data Classification, Computers & Electrical Engineering. 2018, 65, 474-487.

[167] Yu, L.; Liu, H. Efficient Feature Selection via Analysis of Relevance and Redundancy. The Journal of Machine Learning Research. 2004, 5, 1205-1224.

[168] Zaeri-Amirani, M.; Afghah, F.; Mousavi, S. A Feature Selection Method Based on Shapley Value to False Alarm Reduction in ICUs, a Genetic-Algorithm Approach, arXiv:1804.11196v1 [eess.SP]. 2018.

[169] Zhao, J.; Liang, J.; Dong, Z.; Tang, D.; Liu, Z. Accelerating Information Entropy-Based Feature Selection Using Rough Set Theory with Classified Nested Equivalence Classes. Pattern Recognition. 2020, 107, 107517.

Disponibilidad online de los conjuntos de datos

[1] http://archive.ics.uci.edu/ml/datasets/Thyroid+disease (consultado el 23 de abril de 2021).

[2] https://archive.ics.uci.edu/ml/datasets/Activity+recognition+with+healthy+older+people+using+a+batteryless+wearable+sensor (consultado el 23 de abril 2021).

[3] https://archive.ics.uci.edu/ml/datasets/Avila (consultado el 23 de abril de 2021).

[4] https://archive.ics.uci.edu/ml/datasets/adult (consultado el 23 de abril de 2021).

[5] https://archive.ics.uci.edu/ml/datasets/nursery (consultado el 23 de abril de 2021).

[6] https://archive.ics.uci.edu/ml/datasets/Bank+Marketing (consultado el 23 de abril de 2021).

[7] https://archive.ics.uci.edu/ml/datasets/HTRU2 (consultado el 23 de abril de 2021).

[8] https://archive.ics.uci.edu/ml/datasets/Connect-4 (consultado el 23 de abril de 2021).

[9] https://archive.ics.uci.edu/ml/datasets/Tic-Tac-Toe+Endgame (consultado el 23 de abril de 2021).

[10] https://archive.ics.uci.edu/ml/datasets/Credit+Approval (consultado el 23 de abril de 2021).

[11] https://archive.ics.uci.edu/ml/datasets/mushroom (consultado el 23 de abril de 2021).