Máster en sistemas inteligentes
Trabajo de Fin de Máster
junio, 2024

Algoritmos de selección de características en aprendizaje automático

Creado por:
Diego Sánchez Martín

Revisado por:
María Angélica González Arrieta
Roberto López
Universidad de Salamanca

Departamento de Informática y Automática
Universidad de Salamanca

Revisado por:

Dr. María Angélica González Arrieta
angelica@usal.es
Universidad de Salamanca
Dr. Roberto López
robertolopez@artelnics.com
Artelnics

Información del Autor:

Diego Sánchez Martín
sanchhez08@usal.es
Universidad de Salamanca

Agradezco profundamente a mis padres
por su apoyo incondicional y amor
y por ayudarme a alcanzar mis metas.

Resumen

En el contexto actual de la inteligencia artificial y el aprendizaje automático en el que nos encontramos, la seleccion de características se ha consolidado como un área crucial en el éxito de los modelos de machine learning. Ante los conjuntos de datos de alta dimensionalidad conocidos como Big Data, la enorme cantidad de características disponibles para analizar puede ser abrumadora, lo cual ha generado cuantiosos desafíos para el entrenamiento de modelos eficientes y precisos. La reducción del tamaño de los denominanos datasets de datos entra en juego como algo indispensable a la hora de plantear este tipo de problemas, eliminando características irrelevantes o redundantes que pueden afectar negativamente al flujo de trabajo en busca de la solución deseada. Enfocarse en las características más relevantes del conjunto inicial ayuda a lograr una serie de beneficios, como mejorar la precisión, reducir el tiempo de entrenamiento o facilitar la interpretación de los modelos, pues tienden a ser más robustos al sobreajuste y el entrenamiento de estos se vuelve más rápido y computacionalmente eficaz.

En este trabajo, se estudian tres enfoques de selección de características. El primero es el enfoque basado en filtros (Filter Method), en el cual se han implementado los algoritmos Relief y Correlation para evaluar la relevancia de las características y basarse en ellos para la selección de atributos. El segundo es un enfoque basado en wrappers (Wrapper Method), el cual se centra en otros dos algoritmos: Selección Secuencial Hacia Adelante (Forward Feature Selection) y el Algoritmo Genético, que evalúan iterativamente diferentes conjuntos de características y seleccionan aquellos subconjuntos que optimizan el rendimiento del un modelo. Y por último, un enfoque integrado (Embedded Method), en el que se utilizan los métodos de Bosque Aleatorio (Random Forest) y LASSO R1 para seleccionar las características más relevantes, penalizando la complejidad del modelo y favoreciendo aquellas con mayor coeficiente en cuanto a importancia.

A través de un meticuloso análisis llevado a cabo con algunos de estos algoritmos, los resultados obtenidos permiten afirmar que la elección del algoritmo de selección de variables adecuado dependerá de las necesidades específicas de cada proyecto. Si se busca una solución rápida y poco exigente en cuanto a resultados, los algoritmos del enfoque basado en filtros pueden ser una buena opción. En cambio, si el objetivo primordial es la máxima exactitud en las predicciones, incluso a costa de un elevado coste computacional y un mayor tiempo de espera, los métodos embedded y wrapper se presentan como las alternativas más adecuadas.

Cabe destacar que este estudio abre la puerta a futuras investigaciones que analicen en mayor profundidad las ventajas y desventajas de cada enfoque, así como el desarrollo de nuevas técnicas de selección de variables que combinen la eficiencia de los métodos filter con la precisión de los embedded y wrapper.

Abstract

In the current context of artificial intelligence and machine learning in which we find ourselves, feature selection has established itself as a crucial area in the success of machine learning models. In the face of high-dimensional datasets known as Big Data, the sheer number of features available for analysis can be overwhelming, which has created significant challenges for training efficient and accurate models. Reducing the size of so-called data datasets comes into play as indispensable when approaching this type of problem, eliminating irrelevant or redundant features that can negatively affect the workflow in search of the desired solution. Focusing on the most relevant features of the initial set helps to achieve a number of benefits, such as improving accuracy, reducing training time or facilitating the interpretation of the models, as they tend to be more robust to overfitting and training them becomes faster and computationally efficient.

In this paper, three feature selection approaches are analized. The first is the filter-based approach (Filter Method), in which Relief and Correlation algorithms have been implemented to evaluate the relevance of features and rely on them for attribute selection. The second is a wrapper-based approach (Wrapper Method), which focuses on two other algorithms: Forward Feature Selection and the Genetic Algorithm, which iteratively evaluate different sets of features and select those subsets that optimize the performance of a model. And finally, an integrated approach (Embedded Method), in which Random Forest and LASSO R1 methods are used to select the most relevant features, penalizing the complexity of the model and favoring those with higher coefficients in terms of importance.

Through a meticulous analysis carried out with some of these algorithms, the results obtained allow us to affirm that the choice of the appropriate variable selection algorithm will depend on the specific needs of each project. If a fast solution with low accuracy requirements is sought, the algorithms of the filter-based approach may be a good option. On the other hand, if the primary objective is maximum prediction accuracy, even at the cost of high computational cost and longer waiting time, embedded and wrapper methods are the most suitable alternatives.

It should be noted that this study opens the door to future research that will explore in greater depth the advantages and disadvantages of each approach, as well as the development of new variable selection techniques that combine the efficiency of filter methods with the accuracy of embedded and wrapper methods.

Tabla de contenido

Listado de figuras

Listado de tablas

1. Introducción

En el vasto campo del aprendizaje automático, la selección de características desempeña un papel fundamental al mejorar la eficiencia y la precisión de los modelos de machine learning. La capacidad de identificar las variables más relevantes en un conjunto de datos no solo simplifica los modelos, sino que también ayuda a evitar el sobreajuste y a mejorar la interpretabilidad. En este trabajo de investigación, se abarcan algunos algoritmos de selección de características, abordando sus fundamentos teóricos, aplicaciones prácticas y el panorama actual en la literatura. Estos algoritmos se han convertido en una herramienta esencial en la caja de herramientas de cualquier científico de datos. Funcionan mediante la evaluación constante y posterior clasificación de las variables en función de su relevancia para el problema específico que se está tratando [32].

La selección de características no es una técnica nueva, sus raíces se remontan a los primeros días del análisis de datos y estadísticas. Sin embargo, su importancia ha crecido exponencialmente con el auge del Big Data y el aprendizaje automático. En los últimos años, como se puede ver en la figura 1 la cantidad de datos generados y recopilados ha aumentado de manera exponencial. Según un informe de la IDC [60], se estima que la cantidad de datos globales se duplicará cada dos años, alcanzando los 175 zettabytes para 2025. Este crecimiento masivo ha subrayado la necesidad de técnicas eficientes de reducción de dimensionalidad para manejar, procesar y extraer información útil de grandes volúmenes de datos.

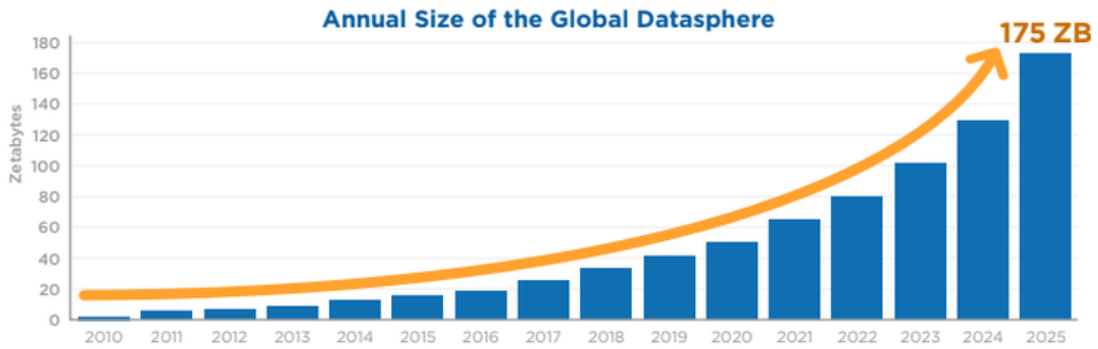


Figura 1: Crecimiento proyectado de los datos generados a nivel mundial [60].

Los usos de la reducción de atributos son muy variados y abarcan una amplia gama de campos, desde el análisis de imágenes [23], un uso muy extendido en la genética [33] o hasta la predicción del mercado financiero [10]. En todos estos campos, la problemática de la complejidad dimensional o el sobreajuste de los modelos sigue siendo un desafío actualmente, pudiendo llegar a tratar en ocasiones con miles de datos en un unico problema benchmarking, esto junto a un ajuste demasiado rígido puede suponer un problema al modelo y hacerle perder su capacidad de generalizarse a nuevos datos, haciendo más difícil y computacionalmente costosa la solución anhelada [36]. En la actualidad, la comunidad científica está inmersa en la búsqueda de métodos más sofisticados y eficientes para abordar problemas de reducción de

dimensionalidad en grandes conjuntos de datos y en entornos de alta magnitud [76]. En este estudio, se implementan y comparan tres enfoques principales de selección de características: filtro, envoltura y embebidos. Los métodos de filtro evalúan las variables independientemente de cualquier modelo de aprendizaje automático [43], los métodos de envoltura utilizan el rendimiento de un modelo específico para evaluar subconjuntos de variables, mientras que los métodos embebidos incorporan la selección de características directamente en el proceso de construcción del modelo [89].

Este problema encuentra aplicaciones tanto en problemas de clasificación como de regresión. En clasificación, donde el objetivo es asignar observaciones a categorías predefinidas, la selección de características puede mejorar significativamente la precisión y eficiencia al identificar y utilizar solo las características más relevantes y así los modelos pueden entrenarse más rápido y con menos riesgo de sobreajuste. Por otro lado, en problemas de regresión, donde se busca predecir valores continuos, ayuda a construir modelos más interpretables y precisos. Esto es especialmente crucial en campos como la economía y la biomedicina, donde es esencial entender la influencia de diferentes variables independientes sobre el resto. La combinación de técnicas de selección de características con métodos de regresión como la regresión lineal, la regresión polinómica y la regresión con regularización (LASSO, Ridge) permite manejar datos de alta dimensionalidad de manera eficiente y efectiva [16, 84, 63].

A pesar de los avances en la selección de características, persisten varios desafíos y consideraciones que deben ser abordados. Uno de los principales desafíos es la escalabilidad de los algoritmos de selección de características cuando se enfrentan a conjuntos de datos de alta dimensionalidad, donde el número de características puede ser extremadamente grande [48]. Además, la selección de características puede ser sensible al ruido y a la redundancia en los datos, lo que puede afectar negativamente el rendimiento del modelo [86]. Otro desafío importante es la necesidad de equilibrar la simplicidad del modelo con su precisión, ya que una selección excesiva de características puede llevar a un modelo simplificado que no capture completamente las complejidades de los datos, mientras que una selección insuficiente puede resultar en un modelo sobreajustado [15]. Además, la interpretación y la reproducibilidad de los resultados de la selección de características son esenciales, especialmente en campos como la bioinformática y la medicina, donde la comprensión de las relaciones entre variables es crucial para la toma de decisiones [64].

El manejo de características altamente correlacionadas representa otro reto significativo, ya que puede llevar a una redundancia en la información aportada al modelo y complicar su interpretación [13]. La integración de técnicas de machine learning con otros procesos de aprendizaje automático y la evaluación de su impacto en el ciclo completo de desarrollo del modelo son áreas que requieren una atención continua y una investigación más profunda. Esto incluye la consideración de cómo las técnicas existentes pueden afectar a la robustez y la generalización del modelo en entornos cambiantes [39], así como el desarrollo de métodos automatizados que puedan adaptarse dinámicamente a diferentes tipos de datos y necesidades de modelado [50].

Se espera que este trabajo contribuya a una mejor comprensión de los desafíos y oportunidades que presenta la selección de características en el contexto del Big Data, y que proporcione resultados útiles para la aplicación de estas técnicas en diferentes áreas del conocimiento.

1.1. Objetivos del trabajo

Los objetivos de este trabajo se centran en estudiar y comparar diferentes enfoques de algoritmos de selección de características utilizando datasets para problemas de regresión y clasificación. El análisis se divide en evaluar la eficiencia en términos de tiempo de ejecución, la precisión de los resultados, la consistencia de los algoritmos en diferentes plataformas y finalmente, identificar las combinaciones más efectivas de algoritmos y plataformas. Este enfoque proporciona una visión comprensiva sobre el rendimiento y precisión de los algoritmos de selección de características en distintas herramientas y contextos, es por ello que, durante la elaboración de este proyecto se han abarcado los siguientes puntos:

1. **Comparar la eficiencia de algoritmos de selección de características:** Evaluar y comparar los tiempos de ejecución de diversos algoritmos de selección de características en múltiples plataformas. Para ello se realizarán experimentos sistemáticos para medir el tiempo de procesamiento requerido utilizando dos datasets de regresión y dos de clasificación. Este objetivo tiene como fin proporcionar recomendaciones prácticas para investigadores y profesionales que buscan optimizar el rendimiento y la precisión en tareas de selección de características.

2. **Evaluar la precisión de algoritmos de selección de características:** Comparar la precisión obtenida así como el error subyacente de los algoritmos de selección de características en los diferentes datasets en cada una de las cinco plataformas mencionadas y buscando determinar cuáles son los algoritmos y plataformas que ofrecen los resultados más precisos para cada tipo de dataset. Este análisis permitirá determinar cuáles son los algoritmos y plataformas que ofrecen los resultados más precisos para cada tipo de dataset, proporcionando así información valiosa para la toma de decisiones en la selección de herramientas y técnicas adecuadas.

3. **Analizar la Consistencia de Resultados entre Diferentes Plataformas:** Comprobar si existe uniformidad entre los resultados obtenidos al haber aplicado los mismos algoritmos de selección de características en las cinco plataformas diferentes con el fin de identificar posibles variaciones en los resultados debido a las implementaciones específicas de cada plataforma. Esto permitirá identificar posibles discrepancias y entender cómo las diferencias en la implementación de los algoritmos pueden afectar los resultados finales.

4. **Identificar las Mejores Combinaciones de Algoritmos y Plataformas:** Se seleccionarán las combinaciones más efectivas de algoritmos de selección de

características y plataformas en función de la relación entre tiempo de ejecución y precisión centrándose en identificar las más eficientes en esos términos y facilitando cuáles son las técnicas más adecuadas para aplicaciones prácticas en el ámbito del Big Data.

2. Estado del arte

La selección de atributos es una práctica esencial en el campo de la inteligencia artificial y del aprendizaje automático, que implica la identificación de un subconjunto relevante de características o variables dentro de un conjunto de datos más amplio. Su objetivo principal es mejorar el rendimiento de los modelos predictivos, reduciendo la dimensionalidad del conjunto de datos, lo cual no solo facilita el procesamiento y análisis de la información, sino que también puede aumentar la precisión y la interpretabilidad de los modelos. Este proceso es especialmente importante en situaciones donde el conjunto de datos original contiene un gran número de atributos, muchos de los cuales pueden ser irrelevantes o redundantes, introduciendo ruido que afecta negativamente al rendimiento del modelo.

Los usos de la selección de atributos son muy variados y abarcan una amplia gama de campos. En el análisis de imágenes, por ejemplo, la selección de atributos permite identificar características visuales clave que pueden ser cruciales para tareas como el reconocimiento de patrones o la clasificación de objetos, ambas ampliamente usadas en el ámbito de la medicina [78, 44, 19]. En genética, esta técnica es fundamental para manejar la inmensa cantidad de datos generados por tecnologías como la secuenciación del genoma, ayudando a identificar los marcadores genéticos más relevantes que pueden estar asociados con enfermedades o características específicas [66, 59, 49, 70]. Asimismo, en el ámbito de la predicción del mercado financiero, la selección de atributos permite seleccionar indicadores económicos y financieros que son más relevantes para prever movimientos del mercado, optimizando así los modelos de previsión y toma de decisiones [38, 88]. Además de estos campos, la selección de atributos encuentra aplicaciones en áreas como el procesamiento del lenguaje natural [7, 73], la bioinformática [18, 80], la detección de fraudes [58, 69], y el diagnóstico médico [26, 27, 67], entre otros. En cada uno de estos contextos, la capacidad de seleccionar adecuadamente los atributos más significativos es esencial para mejorar la eficiencia y la efectividad de los análisis y modelos predictivos, subrayando la importancia transversal de esta técnica en diversas disciplinas científicas y prácticas.

2.1. Enfoques de selección de características

2.1.1. Enfoques basados en filtros

Los algoritmos de selección de características que implementan un enfoque basado en filtros (Filter Methods) son capaces de identificar las características más relevantes del conjunto de datos sin necesidad de utilizar un predictor específico. Emplean métricas de evaluación para calcular un índice de relevancia y aquellas características que obtienen valores más bajos dentro del ranking son filtradas, pues se considera que tienen menor probabilidad de ser útiles para la tarea de análisis de datos. A pesar de su eficiencia computacional, estos métodos no garantizan la obtención del conjunto óptimo de atributos relevantes, especialmente cuando las características

están altamente relacionadas [22]. Por tanto, aunque son los más rápidos y menos costosos computacionalmente en comparación con los enfoques wrapper y embedded, pueden perder información valiosa respecto a otros enfoques más sofisticados.

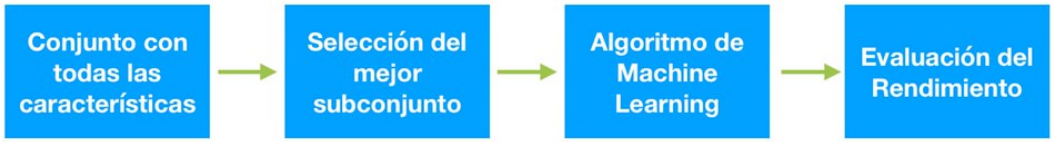


Figura 2: Flujo de los métodos basados en filtros [1].

2.1.1.1. Algoritmo Relief

El algoritmo Relief estima la calidad de los atributos en función de lo bien que sus valores se distinguen entre instancias cercanas entre si. Para ello, dada una instancia seleccionada aleatorioamente:

$$x_i = x_{1i}, x_{2i}, , x_{ni} \tag{1}$$

Relief busca sus dos vecinos más cercanos: uno de la misma clase, denominado acierto más cercano ó H, y el otro, de una clase diferente denominado error más cercano ó M. A continuación, actualiza la estimación de la calidad de todas las características, en función de los valores H y M. Este algoritmo puede trabajar con características discretas y continuas, pero se limita a problemas de dos clases (Clasificación).

Los Algoritmos Basados en Relief (RBA) conservan las ventajas generalizadas de los algoritmos de filtrado, es decir, son relativamente rápidos y las características seleccionadas no dependen del algoritmo de aprendizaje automático específico que se vaya a utlizar posteriormente [79]. Esto significa que las características seleccionadas pueden ser utilizadas en diferentes modelos sin necesidad de recalcularlas, ahorrando un posterior esfuerzo computacional. Como dato, pueden aplicarse no sólo para seleccionar las características 'top' del conjunto inicial, sino que también pueden aplicarse como conocimiento para guiar algoritmos estocásticos de aprendizaje automático como los algoritmos evolutivos.

Existe una extensión, ReliefF, que no sólo trata problemas multiclase, sino que también es más robusto y capaz de tratar con datos incompletos y ruidosos. ReliefF se adaptó posteriormente a problemas de regresión continua, dando lugar al algoritmo RReliefF. La familia de métodos Relief es especialmente atractiva porque puede aplicarse en todas las situaciones, tiene un sesgo bajo, incluye la interacción entre características y puede capturar dependencias locales que otros métodos pasan por alto [74].

2.1.1.2. Algoritmo Correlación

La correlación es una medida conocida de similitud entre dos variables aletorias. Si dos variables aleatorias son linealmente dependientes, entonces su coeficiente de correlacion es ±1. Si las variaciones no están correlacionadas, el coeficiente de correlación es 0. El coeficiente de correlación es invariante a la escala y a la traslación, por tanto, dos características con varianzas diferentes pueden tener el mismo valor de esta medida. Si tenemos n vectores de características d-dimensionales:

$$x_i = [{}^i x_1, {}^i x_d] \qquad i = 1, ..., n \tag{2}$$

de K clases posibles. La correlación mutua de un par de características x_i y x_j se define como:

$$r_{x_i,x_j} = \frac{\sum_k {}^k x_i^k x_j - n\overline{x}_i\overline{x}_j}{\sqrt{(\sum_k {}^k x_i^2 - n\overline{x}_i^2)(\sum_k {}^k x_j^2 - n\overline{x}_j^2)}} \tag{3}$$

Si dos atributos x_i y x_j son independientes, entonces no están correlacionados [34].

Diversos estudios han investigado y aplicado la correlación en distintos contextos. Por ejemplo, en el trabajo [71] se analiza la utilización de la información mutua para detectar y evaluar dependencias entre variables, ofreciendo una alternativa robusta al coeficiente de correlación de Pearson para relaciones no lineales. Asimismo, en el trabajo de Hardin et al. [35] se desarrollan métodos robustos para medir la correlación entre genes en microarrays, destacando la importancia de métodos alternativos en el análisis de datos genómicos. Por otro lado, en estudios como el de Gao et al. [29] se examinan la correlación entre variables en datos longitudinales mediante modelos mixtos lineales, permitiendo una comprensión más detallada de la evolución conjunta de variables clínicas. Este enfoque es crucial para aplicaciones en medicina y otras ciencias donde la relación entre variables puede cambiar a lo largo del tiempo.

2.1.1.3. Otros enfoques basados en Filtros

A grandes rasgos, podemos clasificar las medidas desarrolladas para filtrado de características en: información, distancia, coherencia, similitud y medidas estadísticas. Hay muchos métodos de selección de variables en el enfoque basado en filtros descritos en la literatura. En la tabla 1, junto con las correspondientes referencias se proporcionan detalles de varios de ellos. No todas las características de filtrado pueden utilizarse para todas las clases de tareas en minería de datos. Por ello, los filtros también se clasifican en función de la tarea: clasificación, regresión o clustering.

Tabla 1: Algoritmos más comunes del enfoque basado en filtros.

Nombre	Clase filter	Aplicable a tarea
Information Gain	Univariante	Clasificación
Gain Ratio	Univariante	Clasificación
Correlation	Univariante	Clasificación, Regresión
Chi-square	Univariante	Clasificación
Fisher Score	Univariante	Clasificación
Relief and ReliefF	Univariante	Clasificación, Regresión
Feature Weighting K-means	Multivariante	Clustering

Entre estos algoritmos, **Information Gain** mide la relevancia de una característica en función de la reducción de la entropía que proporciona, siendo ampliamente utilizado en clasificación. Un estudio reciente emplea Information Gain para mejorar la correlación estructural en alertas de seguridad, mostrando así su eficacia en la clasificación de estas alertas [6].

Gain Ratio, una variante del Information Gain, ajusta la evaluación de rasgos corrigiendo el sesgo hacia características con muchos valores distintos. Este ajuste se logra dividiendo el Information Gain por la entropía intrínseca de la característica, proporcionando una evaluación más estable.

Chi-square evalúa la independencia de una característica respecto a la clase objetivo, siendo útil en tareas de clasificación. Este método es especialmente relevante para datos categóricos, donde permite identificar características que tienen una relación significativa con la clase objetivo [51].

El **Fisher Score** mide la capacidad de una característica para discriminar entre diferentes clases, calculando la relación entre la variabilidad entre clases y la variabilidad dentro de las clases. El procedimiento es eficaz en la selección de características que maximicen la separación entre clases, mejorando así la precisión de los modelos de clasificación [31].

2.1.2. Enfoques basados en Wrappers

También denominados métodos de bucle cerrado o métodos de envoltura (Wrapper Methods), envuelven la selección de características alrededor del algoritmo de aprendizaje y utilizan la precisión del rendimiento o la tasa de error del proceso de clasificación como criterio de evaluación. Al disminuir el error de estimacioń de un clasificador específico, elige el subconjunto más discriminativo. Los métodos de envoltura realizan la selección de atributos basándose en el rendimiento del algoritmo de aprendizaje, seleccionando el rasgo más óptimo para cada paso en la predicción,

de ahí que consigan una mejor precisión y un mayor rendimiento en comparación con los métodos basados en filtros [47, 72, 75].

Los filtros univariantes se usan para evaluar y usualmente clasificar una única característica, mientras que los filtros multivariantes evalúan un subconjunto completo de estas. La generación de dichos subconjuntos depende de la estrategia de búsqueda. Aunque existen muchas, las cuatro más habituales como punto de inicio para la generación de subsets son:

1. Búsqueda hacia delante.
2. Búsqueda hacia atrás.
3. Búsqueda bidireccional.
4. Búsqueda basada en una heurística.

La primera suele comenzar con un conjunto vacío y considera la posibilidad de añadir una o más características al conjunto iterativamente mientras lo va rellenando. En cambio, la busqueda hacia atrás considera la eliminación de una o más características del set de datos. En cuanto a la exploración bidireccional, nos encontramos conque parte de ambos lados desde un conjunto vacío y un conjunto completo, considerando simultáneamente conjuntos de atributos más grandes y más pequeños a la vez. La selección heurística genera un subjunto inicial basado en una heurística (como un algoritmo genético) y, a continuación, lo analiza más a fondo.

Figura 3: Flujo de los métodos de envoltura [1].

Las estrategias de búsqueda más habituales que pueden utilizarse con filtros multivariantes pueden clasificarse en: algoritmos exponenciales, algoritmos secuenciales y algoritmos aleatorios. Los algorimos exponenciales evalúan un número de subconjuntos que crece exponencialmente con el tamaño del espacio de atributos mientras que los algorimos secuenciales los añaden o eliminan de forma secuencial. Los algoritmos aleatorios incorporan la aleatoriedad en su procedimiento de búsqueda [39].

2.1.2.1. Algoritmo genético

Los algoritmos genéticos o Genetic Algorithms (GA) son una técnica de optimización inspirada en los procesos evolutivos de la naturaleza. En los algorimos

genéticos, una población de individuos, cada uno representado por un conjunto de genes (cromosomas), evoluciona a lo largo de generaciones. Los individuos más aptos para sobrevivir y reproducirse son los que tienen mayor probabilidad de transmitir sus genes a las siguientes generaciones. De esta manera, la población se va adaptando gradualmente al entorno, mejorando su rendimiento en la tarea que se esté optimizando [53].

Las cinco operaciones más importantes en los algoritmos genéticos son la codificación de cromosomas, la evaluación de la aptitud, los mecanismos de selección utilizados, los operadores genéticos y los criterios para detener la ejecución del algoritmo cuando se considere oportuno. El Algoritmo genético opera en un espacio de búsqueda binario ya que los cromosomas son codificados como cadenas de bits [11]. Para empezar se crea una población inicial (aleatoria) y se va evaluando iterativamente mediante una función de aptitud. Para el cromosoma binario que está siendo examinado se asigna un valor dependiendo de el resultado de la función. El valor '1' de un gen indica que se ha seleccionado esa característica concreta indexada por la posición actual mientras que en el caso contrario, es decir, el valor '0', corresponde a una característica que no se selecciona para la evaluación cromosómica y por tanto sería descartada.

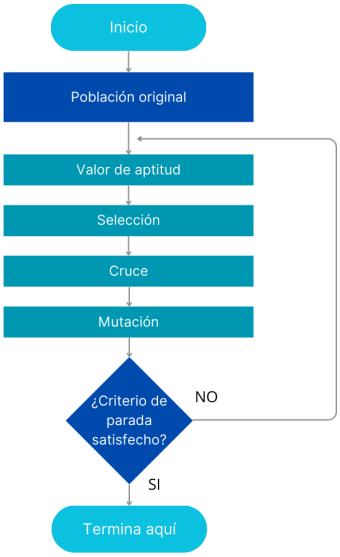


Figura 4: Esquema del algoritmo genético. (Fuente propia)

Los GAs son una herramienta poderosa para resolver problemas en diversos campos, especialmente en aquellos donde la búsqueda exhaustiva es impracticable. Su capacidad para adaptarse y aprender los hace ideales para problemas dinámicos y cambiantes. Además, su paralelismo inherente permite una rápida exploración del espacio de búsqueda, lo que los convierte en una opción atractiva para problemas de gran complejidad.

2.1.2.2. Selección secuencial hacia adelante

Los algoritmos de selección de características hacia delante (Sequential Forward Selection) se basan en un enfoque iterativo para construir un subconjunto óptimo de atributos a partir de un conjunto inicial vacío de candidatos. En cada iteración, el algoritmo evalúa la contribución de cada característica restante no seleccionada ya con anterioridad y elige la que mejora el rendimiento del modelo predictivo. Este proceso se repite hasta que se alcanza un número predeterminado de características o se cumple un criterio de parada específico. Entre las diversas clases de métodos existentes, la selección hacia delante basada en información mutua se ha hecho muy popular y se utiliza ampliamente en la práctica [28]. Sin embargo, las evaluaciones comparativas de estos métodos se han visto limitadas por estar basadas en conjuntos de datos y clasificadores específicos.

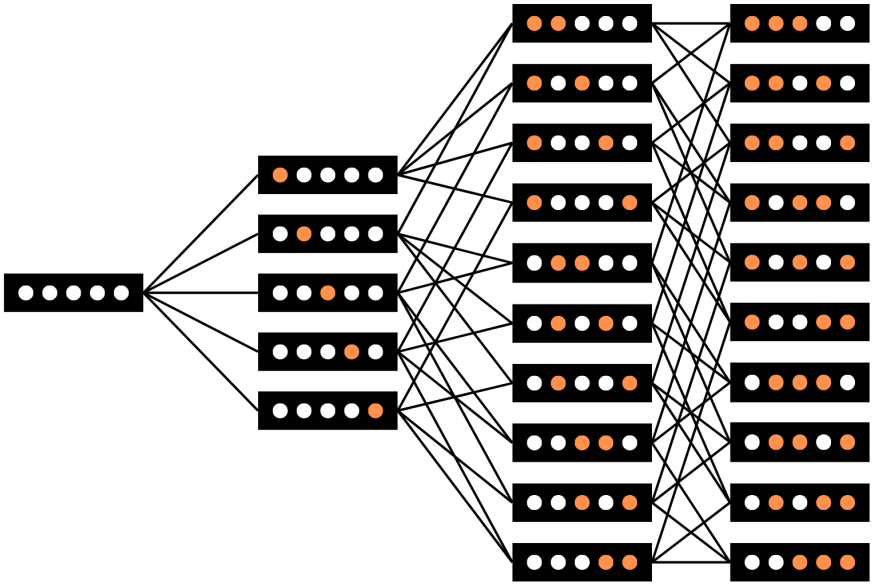


Figura 5: Búsqueda secuencial hacia delante. (Fuente propia)

La principal ventaja de los algoritmos de selección hacia delante (SFS), radica en su simplicidad conceptual y facilidad de implementación. A diferecia de otras técnicas más complejas, estos métodos no requieren cálculos intensivos ni modelos estadísticos sofisticados. Esto los convierte en una de las opciones más atractivas para problemas de gran tamaño aunque en ocasiones requieran de grandes recursos computacionales. A pesar de ello, no se ha dejado de investigar y se han propuesto alternativas para disminuir progresivamente el problema del coste computacional [82].

2.1.2.3. Otros enfoques basados en Wrappers

Además del Algoritmo de Selección Secuencial hacia Delante y del Algoritmo Genético, existen otros métodos basados en wrappers que son ampliamente utilizados

en la selección de características.

El **Algoritmo de Selección Secuencial hacia Atrás** (Sequential Backward Selection o SBS) funciona de forma similar al SFS, salvo por la condición de que la búsqueda se realiza en sentido inverso. De hecho, el algoritmo comienza con un conjunto que contiene todas las características disponibles y, a continuación, va eliminando aquellas cuya eliminación aumenta más la precisión de la clasificación en ese punto. Esta tarea se repite de nuevo hasta que no es posible mejorar el rendimiento del modelo y por tanto, no puede suprimirse ninguna de las características restantes. Al igual que la busqeuda hacia delante, este método no garantiza encontrar el subconjunto de características óptimo, pero si una rápida convergencia hacia una solución [4].

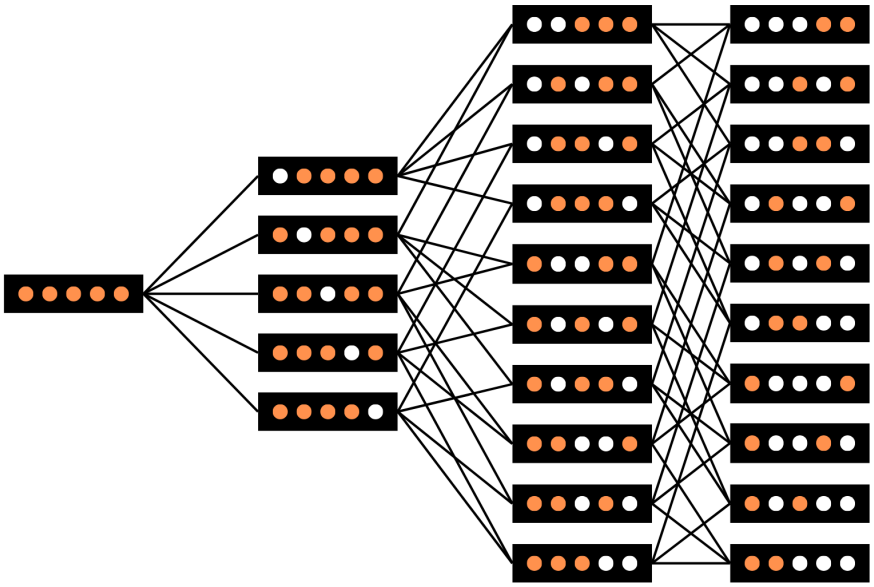


Figura 6: Búsqueda secuencial hacia atrás. (Fuente propia)

Por otro lado, como se aprecia en la figura 7, la **Selección Exhaustiva de Características** (Exhaustive Feature Selection o EFS) es un enfoque más amplio, ya que explora todas las posibles combinaciones de características para determinar el subconjunto óptimo. Aunque este método es computacionalmente costoso, garantiza la selección del mejor subconjunto en términos de rendimiento del modelo. Sin embargo, su aplicabilidad práctica puede ser limitada en conjuntos de datos con un gran número de atributos debido a su complejidad computacional.

Otro enfoque relevante es el **Algoritmo de Eliminación Recursiva de características** (Recursive Feature Elimination o RFE) que es un método iterativo que evalúa y elimina sistemáticamente los atributos menos importantes para mejorar el rendimiento del modelo. En cada iteración, el modelo se entrena con el conjunto actual y se elimina aquella con menor relevancia, repitiendo este proceso hasta alcanzar un número óptimo de datos. Aunque es efectivo para identificar características relevantes, RFE puede ser computacionalmente intensivo, especialmente con conjuntos de datos de alta dimensionalidad.

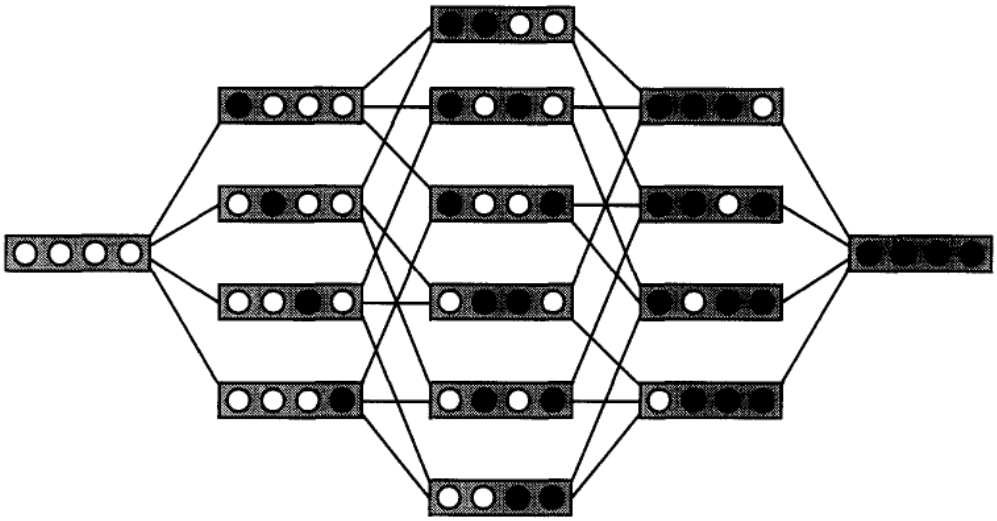


Figura 7: Búsqueda exhaustiva de características [2].

2.1.3. Enfoques integrados

Los métodos integrados o embebidos (Embedded Methods) son mecanismos de selección integrados en el algoritmo de aprendizaje que utiliza sus propiedades para guiar la evaluación de las características. Para ello, durante el proceso de entrenamiento realizan la selección como parte de la ejecución del algoritmo de modelado. Estos métodos tienden a ser más eficientes computacionalmente que los métodos wrappers porque integran simultáneamente el modelado con la selección en paralelo y, a diferencia de los métodos basados en filtros o los métodos basados en wrappers, los algoritmos de elección de atributos del enfoque embedded, la parte de aprendizaje y la parte de selección no pueden separarse, la estructura de la clase de funciones consideradas desempeña un papel crucial [45].

Figura 8: Flujo de los métodos integrados. (Fuente propia)

Esto se debe a que el método integrado evita la ejecución repetitiva del clasificador y el examen de cada subconjunto de características analizado. Además, estos algoritmos tienen menor riesgo de sobreajuste que los métodos mencionados y, al igual que ellos, este enfoque tiene en cuenta las dependencias entre atributos, pero sólo es específico para un algoritmo de aprendizaje determinado. Sin embargo, la

complejidad computacional es un problema importante, especialemente en datos de alta dimensión [8].

2.1.3.1. Bosque aleatorio (Random Forest)

Los algoritmos basados en bosque aleatorio son un método de aprendizaje muy popular y altamente preciso para tareas de clasificación y regresión de alta dimensión fundamentados por la idea de agregación de modelos.

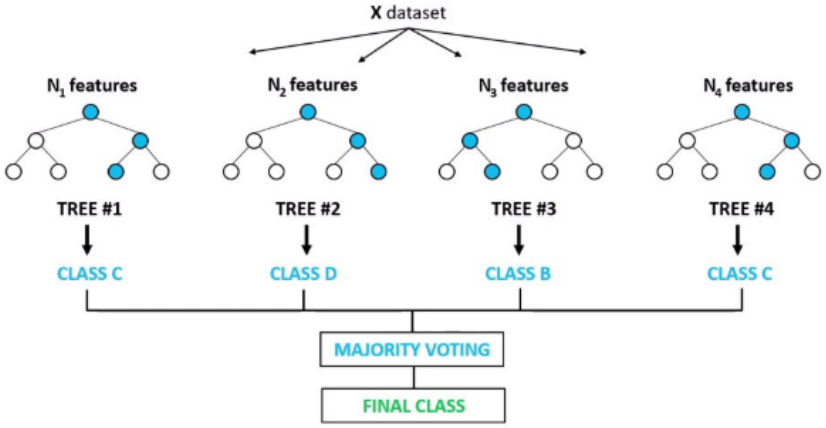


Figura 9: Esquema del algoritmo random forest [3].

La idea clave que subyace en el marco de los bosques aleatorios es hacer crecer un gran número de árboles de decisión insesgados a partir de muestras aleatorias de los datos de entrenamiento con reemplazo, donde cada árbol vota por una clase y el bosque elige la clasificación que tiene más votos entre todos los árboles del bosque. Una de las principales ventajas de los bosques aleatorios es que pueden medir la puntuación de importancia de cada característica para conocer su impacto en la predicción de las clases. Sin embargo, en problemas de gran dimensión, el número de características puede ser enorme, lo que dificulta la investigación manual de las puntuaciones de importancia de estas y la selección de las más relevantes para la clasificación. En este sentido, el procedimiento de selección automática de atributos basado en puntuaciones de importancia puede conducir a seleccionar los más relevantes, compactos y discriminativos [77].

Los algoritmos de Random Forest han demostrado ser ampliamente utilizados en diversas investigaciones contemporáneas debido a su alta precisión y su capacidad de manejar grandes volúmenes de datos. Por ejemplo, en el estudio [54], se emplea este método para predecir enfermedades cardiovasculares con una precisión notable, destacando su eficacia en aplicaciones médicas. Además, en el trabajo [17], se utiliza este enfoque para la clasificación de imágenes satelitales, mejorando significativamente la precisión de los modelos tradicionales. Otro estudio de referencia es [85], donde se aplican bosques aleatorios para la detección de fraudes en transacciones financieras, demostrando su robustez y eficiencia en el procesamiento de datos de alta

dimensión. Estas investigaciones subrayan su relevancia y aplicabilidad en distintos campos, consolidando su posición como una herramienta esencial en el análisis de datos modernos.

Estos algoritmos se han establecido como una técnica de aprendizaje supervisado extremadamente potente y versátil. Su capacidad para manejar datos complejos, combinada con la evaluación automática de la importancia de las características, los convierte en una opción preferida para muchos investigadores y profesionales. A pesar de los desafíos asociados con la gran cantidad de datos, los métodos automatizados de selección de atributos basados en puntuaciones de importancia permiten optimizar los modelos y mejorar su interpretabilidad. Así, los Random Forests continúan siendo una herramienta valiosa y ampliamente adoptada en el análisis de datos, ofreciendo soluciones precisas y eficientes para una gran gama de aplicaciones.

2.1.3.2. LASSO R1

El método LASSO (Least Absolute Shrinkage and Selection Operator) es una técnica de regresión muy utilizada en el campo del aprendizaje automático y la estadística tanto para la selección de variables, como para la regularización, con el objetivo de optimizar la certeza predictiva y la interpretabilidad del modelo. Introducido por Tibshirani en 1996, el método agrega una penalización basada en la suma de los valores absolutos de los coeficientes a la función de costo de la regresión lineal. Este tipo de regularización puede forzar algunos coeficientes a ser exactamente cero, facilitando así la selección de un subconjunto reducido de atributos relevantes. Un ejemplo notable del uso de LASSO en la investigación es el trabajo de Hastie et al.[37], donde se explora su aplicabilidad en modelos de datos de alta dimensión.

Ha sido utilizado en una variedad de campos para resolver problemas donde el conjunto de datos es bastante grande y se pretende perfeccionar la exactitud de los modelos predictivos. En el estudio de Wang et al. [83], LASSO se empleó para seleccionar genes relevantes en el análisis de expresión génica, demostrando su eficacia en la reducción del número de variables sin perder capacidad pronosticadora. Otro ejemplo significativo es el trabajo de Bunea et al. [14], donde se utilizó para la selección de características en modelos de redes neuronales, logrando mejorar la interpretabilidad y reducir la complejidad del modelo. Además, en el ámbito económico, se aplicó LASSO en el estudio de Mullainathan y Spiess [57] para predecir variables macroeconómicas, encontrando que este método puede manejar de manera eficiente grandes conjuntos de datos y proporcionar predicciones robustas.

LASSO se ha consolidado como una herramienta esencial en la modelización de datos debido a su capacidad para realizar selección de características y regularización simultáneamente. Su uso en la investigación ha demostrado ser valioso en una variedad de aplicaciones, desde la biología computacional hasta la economía, proporcionando modelos más simples y fáciles de interpretar sin sacrificar precisión. A pesar de las limitaciones, como la posible eliminación de variables con efectos pequeños pero significativos, los beneficios de LASSO en términos de reducción de la complejidad del modelo y mejora de la interpretabilidad lo hacen una opción

preferida para muchos analistas de datos. Como lo resalta el estudio de Hastie et al. citado anteriormente [37], las innovaciones y adaptaciones continuas del método LASSO siguen expandiendo sus aplicaciones y mejorando su eficacia en el análisis de datos complejos.

2.1.3.3. Otros enfoques basados en el enfoque embebido

Además de Random Forest y LASSO, existen otros enfoques embebidos que han demostrado ser eficaces en la reducción de la dimensionalidad y el modelado simultáneo. Uno de estos enfoques es el **Método de Regularización de Red Elástica** (Elastic Net), que combina las penalizaciones de LASSO y Ridge Regression. Fue introducido en el artículo de Zou y Hastie [91] como una solución para problemas de colinealidad en los datos, permitiendo la reducción del conjunto de datos en escenarios donde LASSO podría fallar. Este método ha sido particularmente útil en áreas como la bioinformática, donde las variables pueden estar altamente correlacionadas y la selección precisa de características es crucial para la interpretación de los datos.

Otro enfoque embebido destacado es el método de **LARS** (Least Angle Regression), desarrollado por Efron et al. [24]. LARS es una técnica eficiente para el filtrado de características que puede ser vista como una versión más rápida de LASSO. Este método es especialmente útil para modelos lineales de alta dimensión, proporcionando un camino continuo desde la regresión nula hasta la regresión completa a medida que los coeficientes se ajustan. En estudios de predicción de riesgos crediticios, como el de Zhu y Hastie [90], LARS ha demostrado ser capaz de manejar grandes volúmenes de datos y mejorar la precisión de las predicciones al seleccionar características relevantes de manera eficiente.

Por último, el método de **Regresión Logística Regularizada** ha sido ampliamente adoptado en aplicaciones donde la interpretación del modelo es esencial. Este enfoque combina la regresión logística con penalizaciones L1 (LASSO) o L2 (Ridge), lo que permite la selección de características relevantes mientras se mantiene la interpretabilidad del modelo. En el estudio de Simon et al. [68], se demostró que la regresión logística regularizada es eficaz para la predicción de enfermedades utilizando datos de alta dimensión, proporcionando modelos que no solo son precisos sino también fácilmente interpretables por los profesionales de la salud.

2.2. Aplicaciones de los algoritmos de selección de características

2.2.1. Clasificación

La clasificación es el problema que consiste en identificar a cuál de un conjunto de categorías (subpoblaciones) pertenece una nueva observación, a partir de un conjunto de datos de entrenamiento que contiene observaciones (o instancias) cuya pertenencia a una categoría es conocida. Muchos problemas del mundo real pue-

den modelarse como problemas de clasificación, como la asignación de un correo electrónico dado a las clases 'spam' o 'no spam', la asignación automática de las categorías (por ejemplo, 'Deportes' y 'Entretenimiento') de las noticias que llegan, y la asignación de un diagnóstico a un paciente dado según lo descrito por las características observadas del paciente (sexo, presión arterial, presencia o ausencia de ciertos síntomas, etc...).

En la fase de entrenamiento, los datos del conjunto original se analizan basándose en los modelos de generación de características, como el modelo de espacio vectorial para datos de texto. Estos atributos pueden ser categóricos (por ejemplo, 'A', 'B', 'AB' u 'O', para el grupo sanguíneo), ordinales (por ejemplo, 'grande', 'mediano' o 'pequeño'), de valor entero (por ejemplo, el número de apariciones de una palabra en un correo electrónico) o de valor real (por ejemplo, una medida de la presión arterial). Algunos algoritmos sólo trabajan con variables discretas y requieren que las de valor real o entero se dividan en grupos (por ejemplo, menos de 5, entre 5 y 10, o más de 10). Después de representar los datos a través de estas características extraídas, el algoritmo de aprendizaje utilizará la información de la etiqueta, así como los propios datos para aprender una función de mapa f (o un clasificador) de las características a las etiquetas.

Durante la fase de predicción, cuando se presenta una nueva observación sin etiqueta, la función de mapeo aprendida es aplicada para así poder predecir la categoría de la observación. Los algoritmos de clasificación pueden variar desde los modelos lineales simples, como la regresión logística y los árboles de decisión, hasta métodos más complejos como las máquinas de vectores de soporte (SVM), las redes neuronales y los bosques aleatorios. Cada uno de estos tiene sus propias ventajas y limitaciones en términos de precisión, interpretabilidad y requisitos computacionales. La precisión del clasificador se evalúa generalmente utilizando métricas como la exactitud, la precisión, el recall (sensibilidad) y el F1-score, las cuales permiten cuantificar la calidad de las predicciones realizadas. Además, técnicas como la validación cruzada y la división de los datos en conjuntos de entrenamiento y prueba son recomendadas para garantizar que el modelo generalice bien a datos no vistos, evitando problemas como el sobreajuste.

2.2.2. Regresión

La regresión es una técnica utilizada para modelar y analizar relaciones entre variables con el objetivo de predecir valores continuos. A diferencia de la clasificación, donde las observaciones pueden asignarse a valores con categorías discretas, en la regresión se busca encontrar una función que relacione una variable dependiente con una o más variables independientes, permitiendo realizar pronósticos cuantitativos. Los problemas de regresión son comunes en una variedad de campos, incluyendo la economía, la biología, la ingeniería y las ciencias sociales. Un ejemplo típico es la predicción del precio de una vivienda basado en características como su tamaño, ubicación y número de habitaciones, o la estimación de la progresión de una enfermedad en función de factores como la edad del paciente y los niveles de ciertos biomarcadores.

El algoritmo de regresión aprende la relación entre las variables independientes (también conocidas como predictoras) y la variable dependiente (también llamada variable de respuesta) en la fase de entrenamiento a partir de un conjunto de datos de ensayo. Este proceso implica ajustar los parámetros para minimizar la diferencia entre las predicciones y los valores observados en los datos de entrenamiento. Los tipos de características pueden ser similares a los de los problemas de clasificación: categóricas, ordinales, de valor entero o de valor real. Por ejemplo, en la regresión lineal, el modelo busca una línea recta que mejor se ajuste a las varialbles, mientras que en la regresión polinómica, puede ajustar una curva más compleja. Otros tipos de regresión, como la regresión logística, la regresión de soporte vectorial y la regresión con regularización (por ejemplo, Lasso y Ridge), también se utilizan dependiendo de la naturaleza del dataset y la relación esperada entre las características.

Una vez que el modelo de regresión ha sido entrenado, puede ser utilizado para predecir valores continuos de la variable dependiente en las llamada fase de predicción, basándose en nuevas observaciones de las variables independientes. La precisión se evalúa utilizando métricas como el error cuadrático medio (MSE), el error absoluto medio (MAE) y el coeficiente de determinación (R^2), que miden la calidad del ajuste del modelo al conjunto de datos.

2.2.3. Otras aplicaciones

La selección de características no sólo puede aplicarse en problemas de clasificación y regresión, también desempeña un papel significativo en otras áreas como el **clustering**. En este contexto, ayuda a identificar y utilizar únicamente las variables más relevantes que influencian la formación de grupos, mejorando así la calidad y la interpretabilidad de los clusters obtenidos. Este enfoque es esencial porque la inclusión de atributos irrelevantes o redundantes puede degradar el rendimiento del algoritmo de clustering, haciendo que los grupos resultantes no sean representativos de las verdaderas relaciones en el conjunto de datos.

El primer algoritmo conocido para la agrupación es K-means. Este algoritmo se utiliza ampliamente en la partición de datos en grupos de la misma característica, donde el número de grupos (clusters) está dado de antemano [9]. K-means es eficiente y fácil de implementar, lo que lo hace popular en diversas aplicaciones, desde el análisis de mercado hasta el reconocimiento de patrones. Además, su capacidad para manejar grandes conjuntos de datos lo convierte en una herramienta esencial en el campo del aprendizaje automático no supervisado. Por otro lado, métodos más avanzados como la selección supervisada para clustering consideran información adicional disponible (por ejemplo, etiquetas parciales o información de supervisión débil) para mejorar la precisión del agrupamiento y la relevancia de las variables seleccionadas. Estas técnicas han demostrado ser efectivas en aplicaciones prácticas como la segmentación de clientes, donde es crucial identificar los atributos que diferencian a distintos segmentos de mercado [40].

2.3. Desafíos y consideraciones en la selección de características

2.3.1. Validación cruzada

En aprendizaje automático, se construyen modelos predictivos a partir de un conjunto de datos llamado çonjunto de aprendizaje". Para evaluar estos conjuntos, se dividen los datos en conjuntos de entrenamiento y de prueba. El modelo se entrena con el conjunto de entrenamiento y posteriormente se evalúa con el conjunto de prueba. Este proceso se repite varias veces usando diferentes subconjuntos para obtener una estimación más precisa del rendimiento [12].

El objetivo es crear un modelo que generalice bien a nuevos datos, no solo a los datos utilizados para el proceso de entrenamiento. Se utilizan diferentes funciones de pérdida y métricas de error para evaluar el rendimiento del modelo.

Algunos de los tipos de validación cruzada más empleados son los siguientes:

1. Submuestreo único aleatorio.
2. Submuestreo aleatorio k-fold.
3. Validación cruzada k-fold.

2.3.1.1. Submuestreo único aleatorio

Entre las diversas estrategias de remuestreo, una de las más sencillas es el método de muestreo único, el cual elige aleatoriamente algunos casos del set de datos original para el conjunto de prueba, mientras que los casos restantes constituyen el conjunto de entrenamiento. A menudo, el conjunto de prueba contiene entre el 10 % y el 30 % de los casos disponibles, y el conjunto de entrenamiento entre el 90 % y el 70 % de los casos respectivamente.

Si el set de datos original es suficientemente grande y, en consecuencia, tanto el conjunto de entrenamiento como el de prueba son grandes, el error de prueba observado puede ser una estimación fiable del verdadero error del modelo para casos nuevos no vistos. Sin embargo, hay varios desafíos y consideraciones que deben tenerse en cuenta:

1. **Varianza del estimador:** Dado que este método implica particionar en una sola división los datos, el estimador del error puede tener alta varianza. Esto significa que los resultados pueden variar significativamente dependiendo de qué subconjunto de datos se elija para el entrenamiento y la prueba. Para mitigar este efecto, es común repetir el proceso varias veces con diferentes divisiones aleatorias y promediar los resultados obtenidos.
2. **Representatividad del conjunto de prueba:** Es crucial asegurarse de que el conjunto de prueba sea representativo de la distribución general de los datos.

Si el subconjunto seleccionado para el conjunto de prueba no captura adecuadamente la variabilidad del conjunto completo de datos, la estimación del error del modelo podría ser sesgada.

3. **Tamaño del conjunto de datos:** En sets de datos pequeños, la división en conjuntos de entrenamiento y prueba puede resultar en que el modelo no tenga suficiente información para aprender adecuadamente durante la etapa de entrenamiento. Esto puede llevar a una estimación poco precisa en nuevos datos. En tales casos, técnicas como la validación cruzada k-fold pueden ser más apropiadas.

4. **Balance de clases:** En problemas de clasificación, es importante verificar que el submuestreo aleatorio no genere divisiones desbalanceadas en cuanto a la representación de las distintas clases. La inestabilidad podría afectar negativamente la capacidad para generalizar, especialmente en clases minoritarias.

El submuestreo único aleatorio es una técnica fundamental, la cual proporciona una manera simple y rápida de evaluar el rendimiento de un modelo. No obstante, es básico considerar los desafíos mencionados y complementar este enfoque con otras técnicas de validación cuando sea necesario para así obtener una evaluación más robusta y fiable del modelo.

2.3.1.2. Submuestreo aleatorio k-fold

En el submuestreo aleatorio k-fold, el método de división única se repite k veces, de forma que se generan k pares de particiones de entrenamiento y de prueba:

$$D_{entrenamiento,i}, \quad D_{prueba,i} \qquad i = 1, ..., n \tag{4}$$

El rendimiento se calcula como la media de los 'n' conjuntos de pruebas. Cualquier par de conjuntos de entrenamiento y de prueba es disjunto, es decir, que estos no tienen ningún caso en común:

$$D_{entrenamiento,i} \cap D_{prueba,i} = \emptyset \tag{5}$$

Sin embargo, dos particiones de entrenamiento o dos de prueba pueden solaparse.

2.3.1.3. Validación cruzada k-fold

La validación cruzada es una técnica estadística habitual para valorar cómo se generalizarán los resultados de un análisis estadístico en la agrupación de datos independientes [42]. Es similar al método de submuestreo aleatorio, pero aquí el estudio se realiza de forma que no se solapen dos distribuciones de prueba. En la validación cruzada de k pliegues, el dataset disponible se divide en k subjuntos disjuntos de aproximadamente el mismo tamaño. Esta partición se realiza mediante muestreo

aleatorio de los datos de aprendizaje sin reemplazo. El modelo se entrena utilizando k - 1 particiones que, juntas, representan el nuevo conjunto de entrenamiento. A continuación, el modelo creado se aplica al resto de datos faltantes, que se denominan conjunto de validación y con esto se mide el rendimiento. Este proceso se repite hasta que cada uno de los k pliegues haya servido como conjunto de validación. La media de las k mediciones de rendimiento en los conjuntos de validación es el rendimiento de la validación cruzada. Esta técnica es ampliamente utilizada a día de hoy en problemas de selección de características [65, 5].

2.3.2. Sobreajuste

El sobreajuste dentro del aprendizaje automático supervisado supone un problema vital que impide generalizar correctamente los modelos para que se ajusten bien los datos de entrenamiento a los datos observados hasta ese punto, así como a los datos no visualizados aún en el conjunto de pruebas. Por otra parte, los modelos sobreajustados tienden a memorizar todos los datos, incluido el ruido inevitable del conjunto de entrenamiento, en lugar de aprender la disciplina oculta tras los datos. Las causas de este fenómeno pueden ser complicadas. Por lo general, podemos clasificarlas en 3 tipos:

1. Aprendizaje de ruido en el conjunto de entranemiento: Cuando el set de entrenamiento es demasiado pequeño en tamaño, tiene mucho ruido o tiene datos menos representativos.

2. Complejidad de la hipótesis: el compromiso de complejidad, concepto clave, es un compromiso entre la varianza y el sesgo.

3. Procedimientos de comparaciones múltiples que son omnipresentes en los algoritmos de inducción: Durante estos procesos, siempre se comparan múltiples elementos basándose en las puntuaciones de una función de evaluación y seleccionando el componente con la máxima puntuación generada. Sin embargo, este proceso probablemente elegirá algunos aspectos que no mejorarán e, incluso, reducirán la precisión de la clasificación [21].

Si se utilizan estimaciones del rendimiento de los distintas subpoblaciones de variables con validación cruzada y algoritmos de búsqueda de alta carga computacional, estas estimaciones pueden dar lugar a predicciones sesgadas. Por lo tanto, pueden llegar a no ser del todo fiables a la hora de comparar métodos de selección de atributos [61].

2.3.3. Problemática de la dimensionalidad

Las herramientas modernas de análisis de datos tienen que trabajar con documentos de alta dimensión, cuyos componentes no se distribuyen de forma independiente. Estos espacios muestran propiedades geométricas sorprendentes y contraintuitivas que tienen una gran influencia en las prestaciones de las plataformas de

análisis. Los modelos construidos mediante el aprendizaje solo son válidos en el rango o volumen del espacio en el que se dispone de información de aprendizaje. Sea cual sea el modelo o la clase de modelos, la generalización de datos muy diferentes de todos los puntos de aprendizaje es imposible [81]. En otras palabras, la generalización pertinente es posible a partir de la interpolación, pero no de la extrapolación. Por lo tanto, uno de los ingredientes clave para el éxito en el desarrollo de algoritmos de aprendizaje es disponer de suficiente conocimiento para el aprendizaje de modo que esto llene el espacio o parte del espacio en el que el modelo debe ser válido. Es fácil ver que, manteniéndose inalterada a cualquier otra restricción, la información de aprendizaje debería crecer exponencialmente con la dimensión, si 10 datos parecen razonables para aprender un modelo de 1 dimensión, 100 son necesarios para aprender uno de 2 dimensiones, 1000 para uno de 3, etc. Este aumento exponencial es la primera consecuencia de lo que se denomina la maldición de la dimensionalidad, que es la expresión de todos los fenómenos que relacionados con esta problemática y que suelen tener consecuencias desafortunadas en el comportamiento y el rendimiento de los algoritmos de aprendizaje.

Para reducir este problema, algunas técnicas y algoritmos reducen el espacio, eliminando los atributos que no aportan información importante y extrayendo relaciones entre las características disponibles para así reducir la masa del set original [55, 87]. Esto rebaja la dificultad, aumenta la comprensión, facilita el análisis, mejora la visualización y reduce los costes de procesamiento y los requisitos de espacio de almacenamiento.

2.3.4. Interpretabilidad del modelo

Hay muchas razones por las que seguir las predicciones de un modelo de aprendizaje automático puede ser ventajoso, siendo la principal de ellas la mejora de la precisión. Muchos estudios demuestran que estos son más precisos que las personas en una variedad de dominios. Desde hace varias décadas hasta la actualidad esta diferencia se ha ampliado más y más a medida que los modelos se han vuelto más precisos [41]. Aunque seguir las estimaciones de un modelo debería permitir a las personas tomar decisiones mucho más rápidas y coherentes, también hay situaciones en las que fiarse complementamente de sus indicaciones puede ser desventajoso, sobre todo los pronósticos son incorrectos o no del todo exactos.

La interpretabilidad del modelo desempeña un papel fundamental al desarrollar y adoptar sistemas de aprendizaje automático, pues proporciona transparencia y comprensión sobre cómo se toman las decisiones. Ribeiro, Singh y Guestrin [62] argumentan que la capacidad de explicar las predicciones de cualquier clasificador es esencial para establecer la confianza en su funcionamiento. Este enfoque en la explicabilidad es crucial para garantizar su aceptación y aplicación en diversas áreas. En consonancia con esta idea, Lundberg y Lee [52] proponen un enfoque unificado para interpretar pronosticos, lo que permite una comprensión más profunda de cómo las características individuales influyen en las decisiones del modelo en su conjunto. Esta comprensión detallada es especialmente importante en escenarios donde se requiere explicar el razonamiento detrás de las decisiones tomadas por el modelo,

como en aplicaciones médicas y legales. Además, los modelos interpretables son más deseables debido a su capacidad para proporcionar explicaciones, lo que aumenta la confianza y la comprensión de estos por parte de los usuarios finales [30].

3. Herramientas utilizadas e implementaciones realizadas

En esta sección, se describen las diferentes herramientas y plataformas utilizadas para implementar y comparar los algoritmos de selección de características. Se proporciona una visión general de cada herramienta, destacando sus características principales y la forma en que se utilizaron los algoritmos en cada una de ellas. La elección de estas herramientas se basa en su popularidad y capacidad para manejar tareas complejas de análisis de datos y aprendizaje automático. Aquí se presentarán los esquemas generados y por ende, utilizados para llevar a cabo el trabajo, incluyendo los seis algoritmos de selección de características: Relief, Correlation, Genetic Algorithm, Forward Feature Selection, Random Forest y LASSO. Cada enfoque estará acompañado por una descripción de la estructura representada y su aplicación en el contexto del estudio.

3.1. KNIME

KNIME (Konstanz Information Miner) es una plataforma de análisis de datos que facilita la creación de flujos de trabajo visuales y la integración de diferentes componentes de análisis y minería de datos. En este estudio, se utilizó KNIME para implementar varios algoritmos de selección de características, aprovechando su interfaz gráfica intuitiva y su amplia biblioteca de nodos que soportan diversos métodos estadísticos y de aprendizaje automático. A continuación, se describen los flujos de trabajo utilizados para cada enfoque y se presentan las figuras correspondientes que ilustran estos flujos en KNIME.

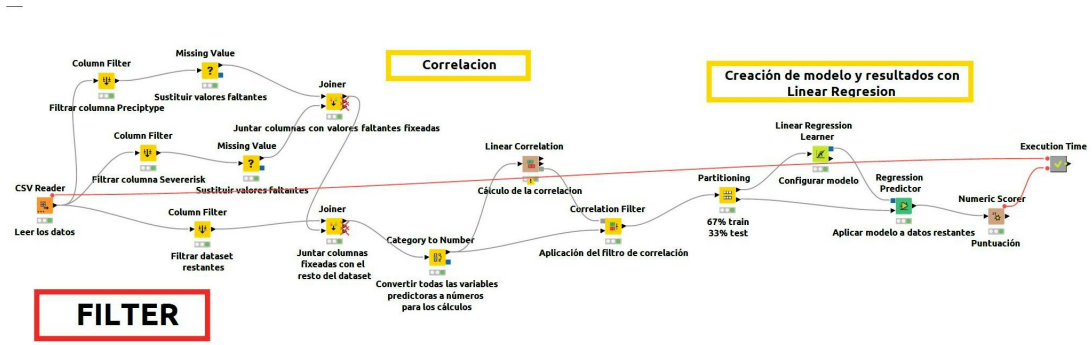


Figura 10: Flujo de datos en el enfoque filter en KNIME. (Fuente propia)

Para el enfoque Filter, se utilizaron algoritmos como Relief, que evalúan la relevancia de cada característica individualmente. La Figura 10 muestra el flujo de trabajo que representa este enfoque en KNIME, donde se seleccionan las características basadas en criterios estadísticos.

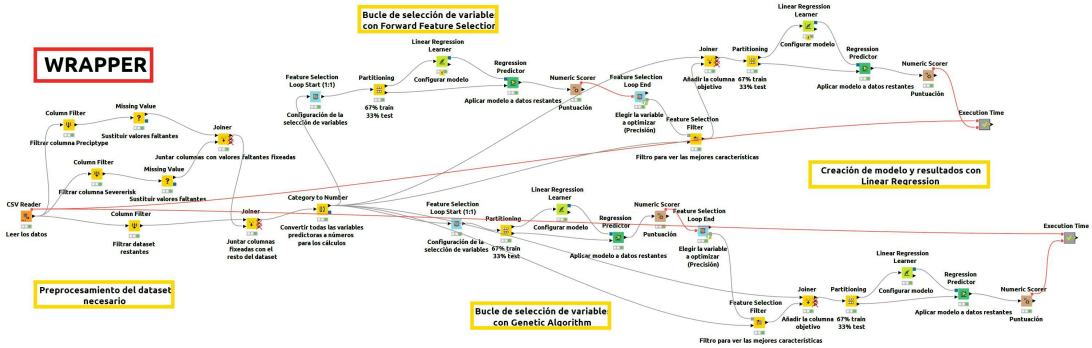


Figura 11: Flujo de datos en el enfoque wrapper en KNIME. (Fuente propia)

El enfoque Wrapper implica la utilización de algoritmos como el Algoritmo Genético y la Selección Sequencial hacia delante, que evalúan conjuntos de características mediante la optimización de un modelo predictivo. La Figura 11 ilustra este flujo de trabajo en KNIME, donde los algoritmos buscan iterativamente las mejores combinaciones de atributos.

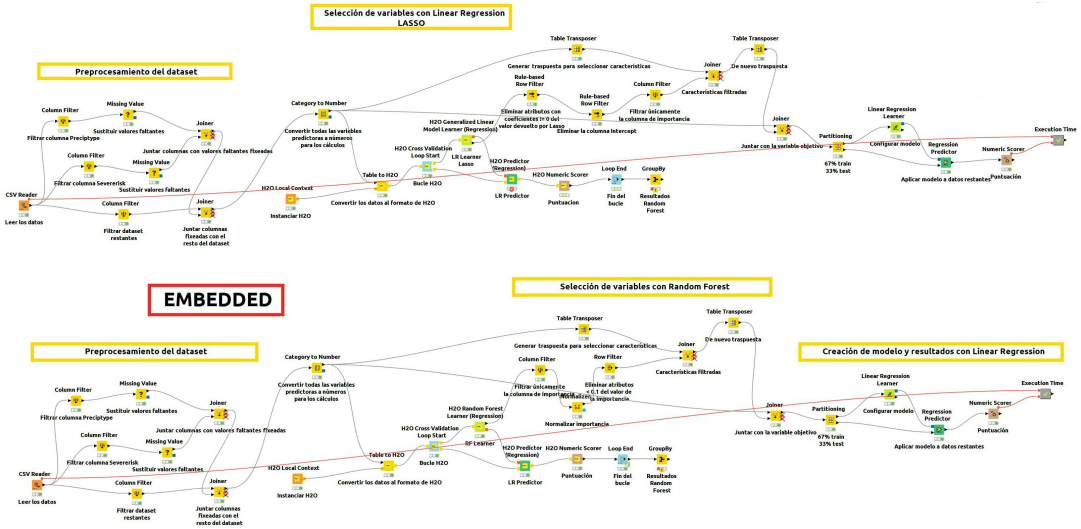


Figura 12: Flujo de datos en el enfoque embedded en KNIME. (Fuente propia)

El enfoque Embedded incorpora la selección de características dentro del propio proceso de entrenamiento del modelo, utilizando algoritmos como Random Forest y LASSO. La Figura 12 muestra cómo se implementa este enfoque en KNIME, donde esta selección se integra directamente en el algoritmo de aprendizaje.

Para evaluar la precisión de los modelos en problemas de regresión y clasificación, se utilizaron dos nodos Scorer diferentes en KNIME debido a las diferencias en las métricas de evaluación entre ambos tipos de problemas. En los problemas de regresión, se utilizó el nodo Numeric Scorer, como se muestra en la Figura 13, que permite calcular métricas como el error absoluto medio (MAE) y la raíz del error cuadrático medio (RMSE). Para los problemas de clasificación, se empleó el nodo Scorer, ilustrado en la Figura 14, que calcula métricas como la precisión o el error subyacente, esenciales para evaluar el rendimiento en tareas de este tipo.

Figura 13: Nodo Numeric Scorer en problemas de regresión. (Fuente propia)

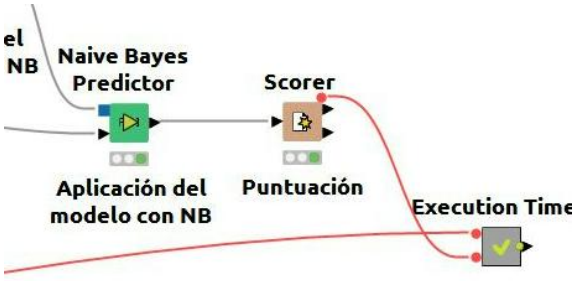


Figura 14: Nodo Scorer en problemas de clasificación. (Fuente propia)

3.2. RAPIDMINER

RAPIDMINER es otra plataforma potente de análisis de datos que proporciona una interfaz visual para el diseño de procesos de minería de datos. Similar a KNIME, permite a los usuarios construir flujos de trabajo complejos mediante la combinación de múltiples operadores. En este trabajo, se utilizó RapidMiner para implementar y comparar diferentes algoritmos de machine learning, beneficiándose de su flexibilidad y extensas capacidades analíticas.

La figura 15 ilustra la configuración general utilizada en RapidMiner para la selección de características. Este esquema abarca todos los enfoques considerados en este trabajo: filter, wrapper y embedded, aplicados a cada dataset. Para cada uno de estos enfoques, se han implementado dos métodos distintos. En el enfoque filter, se emplearon los métodos Relief y Correlación; en el enfoque wrapper, se utilizaron el Algoritmo Genético y la Búsqueda Sequencial hacia Delante y en el enfoque embedded, se aplicaron Random Forest y Lasso. Cada uno de estos métodos fue configurado y ajustado para identificar las variables más relevantes del dataset, optimizando así el rendimiento de los modelos predictivos subsecuentes.

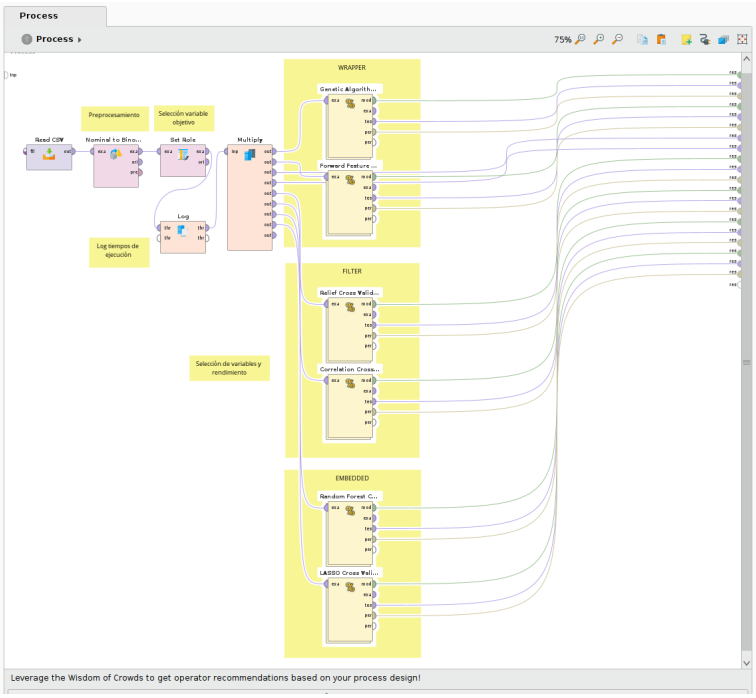


Figura 15: Flujo de datos en diferentes enfoques. (Fuente propia)

En la figura 16 podemos observar el flujo de trabajo interno específico para los problemas de clasificación. En estos casos, se empleó el nodo Naive Bayes para la construcción del modelo de clasificación. La elección de atributos se realiza previamente utilizando los métodos mencionados (Relief, Correlation, Algoritmo Genético, Forward Feature Selection, Random Forest y Lasso), dependiendo del enfoque aplicado. Esta configuración permite evaluar la eficacia de diferentes técnicas en la mejora del rendimiento del clasificador Naive Bayes.

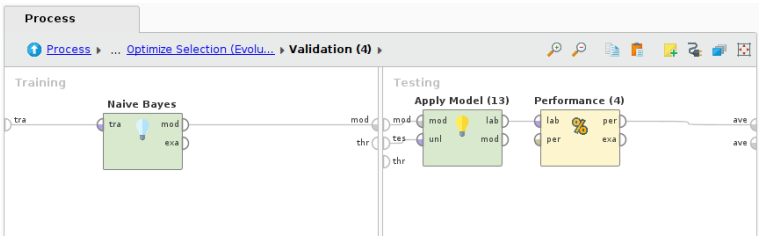


Figura 16: Flujo interno en los problemas de clasificación. (Fuente propia)

De nuevo, en la imagen 17 se detalla el flujo de datos interno en los problemas de regresión con RAPIDMINER. Para construir el modelo de regresión se sustituye el nodo Naive Bayes por el nodo Linear Regression. Al igual que en los problemas de clasificación, la selección de características se lleva a cabo antes de la modelización, aplicando los métodos correspondientes según el enfoque. Esta configuración asegura que el modelo de regresión lineal trabaje con un conjunto de datos optimizados, mejorando así su capacidad predictiva.

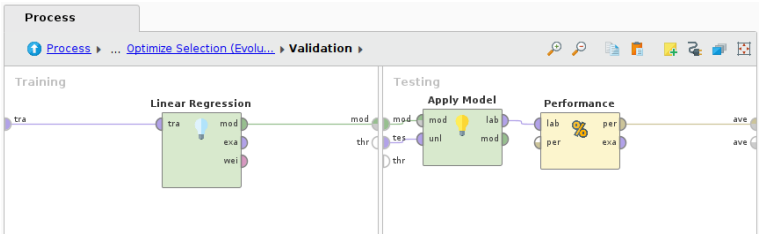


Figura 17: Flujo interno en los problemas de regresión. (Fuente propia)

Los resultados obtenidos en ambos problemas se pueden ver en las figuras 18 y 19. Para cada tipo de problema (clasificación o regresión) se presentan diversas métricas de error. En el caso de la clasificación, la matriz de confusión ofrece una visión detallada del rendimiento del modelo, mostrando el número de instancias correctamente e incorrectamente clasificadas en cada clase y, a partir de la precisión, se puede calcular el error correspondiente, proporcionando una medida clara de la eficacia del clasificador. Para el problema de regresión se incluyen RMSE (Root Mean Squared Error), Absolute Error, Relative Error y Root Relative Squared Error, proporcionando una evaluación cuantitativa del rendimiento del modelo de regresión, permitiendo comparar la precisión de las predicciones con los valores reales y cuantificar la magnitud de los errores cometidos por el modelo.

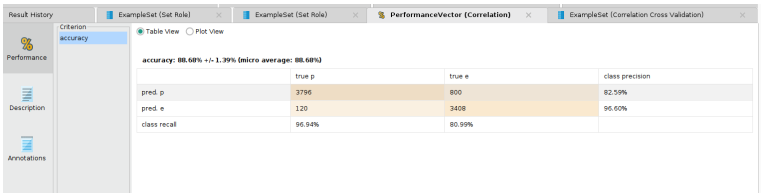


Figura 18: Resultados en los problemas de clasificación. (Fuente propia)

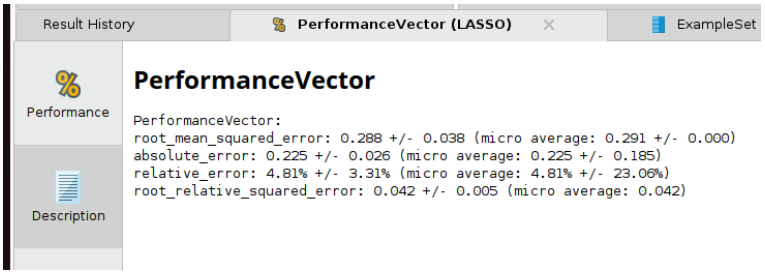


Figura 19: Resultados en los problemas de regresión. (Fuente propia)

3.3. WEKA

WEKA (Waikato Environment for Knowledge Analysis) es una colección de algoritmos de aprendizaje automático para tareas de minería de datos. La plataforma es de código abierto y proporciona una interfaz gráfica fácil de usar, así como acceso a bibliotecas para desarrolladores que deseen integrar capacidades de análisis de datos en sus propias aplicaciones.

Esta subsección, se centrará en el método AttributeSelectedClassifier, que es una técnica que combina selección de atributos con la clasificación posterior. Esto se realiza aplicando un filtro para seleccionar las características más relevantes antes de proceder con la categorización.

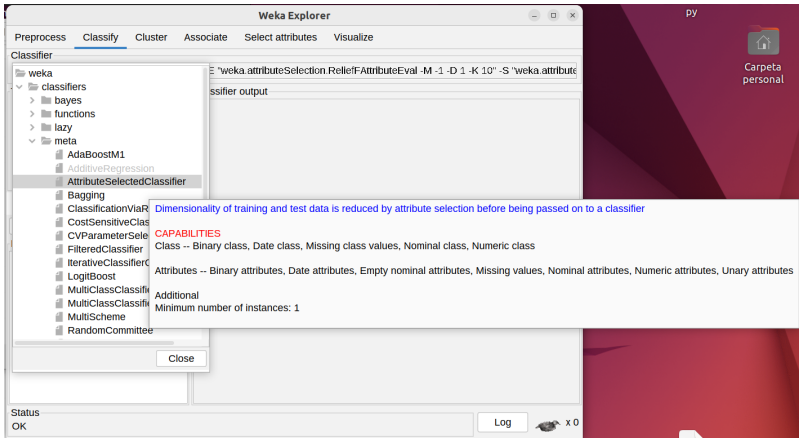


Figura 20: Descripción método AttributeSelectedClassifier. (Fuente propia)

El AttributeSelectedClassifier es un meta-clasificador que se utiliza para mejorar el rendimiento. Primero, selecciona los atributos más importantes del conjunto de datos utilizando un algoritmo de selección de variables (por ejemplo, un filtro o un método de búsqueda). Luego, aplica un clasificador al conjunto de datos reducido, mejorando así la precisión y eficiencia del modelo final.

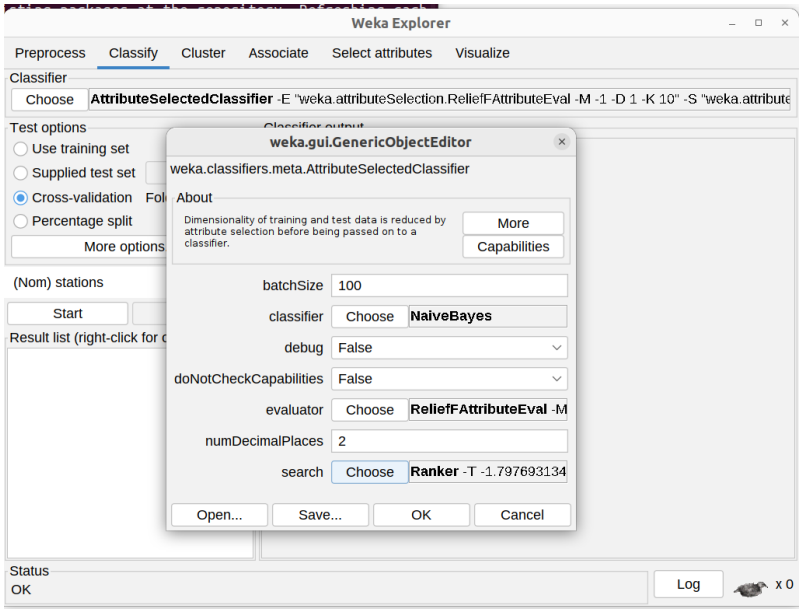
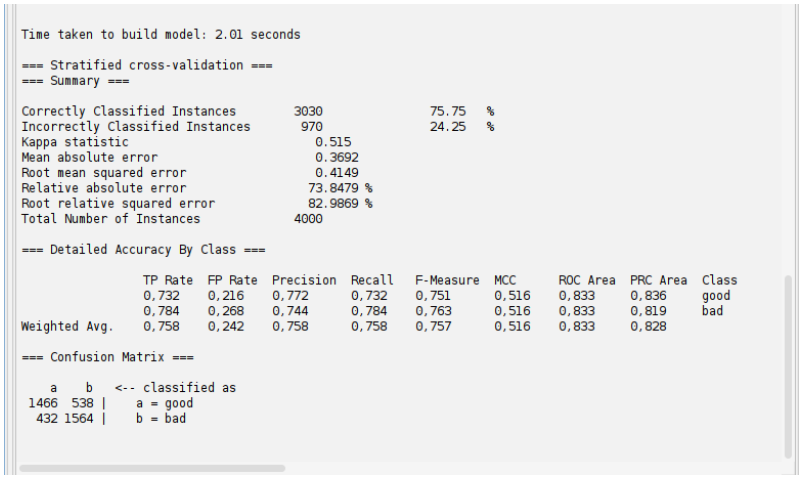


Figura 21: Ejemplo de AttributeSelectedClassifier y enfoque filter. (Fuente propia)

En la figura 21, se muestra la configuración que se ha utilizado a la hora de implementar el método AttributeSelectedClassifier con el enfoque de filtro. En esta configuración, se han cambiado los valores de çlassifier", .ºvaluatorz "search"dependiendo

del tipo de problema a estudiar, ya sea por el tipo de enfoque (wrapper, filter o embedded) o por el tipo de problema (clasificación o regresión). Este proceso de ajuste de configuración permite adaptar el método a las necesidades específicas del análisis, optimizando su rendimiento.



```
Time taken to build model: 2.01 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        3030               75.75   %
Incorrectly Classified Instances       970               24.25   %
Kappa statistic                          0.515
Mean absolute error                      0.3692
Root mean squared error                  0.4149
Relative absolute error                 73.8479 %
Root relative squared error             82.9869 %
Total Number of Instances             4000

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,732    0,216    0,772      0,732    0,751      0,516    0,833     0,836     good
                 0,784    0,268    0,744      0,784    0,763      0,516    0,833     0,819     bad
Weighted Avg.    0,758    0,242    0,758      0,758    0,757      0,516    0,833     0,828

=== Confusion Matrix ===

    a    b   <-- classified as
 1466  538 |    a = good
  432 1564 |    b = bad
```

Figura 22: Ejecución algoritmo genético con datos de las manzanas. (Fuente propia)

En la figura 22, se ilustra la ejecución de un Algoritmo Genético utilizando el dataset de las Manzanas. Se presentan las estadísticas calculadas por el método AttributeSelectedClassifier, incluyendo las instancias correctamente clasificadas y las incorrectamente clasificadas, así como una matriz de confusión para visualizar los resultados.

3.4. R

En esta sección se presenta la aplicación de diversos métodos de selección de características y modelado de datos utilizando el lenguaje de programación R. Se han empleado varios paquetes especializados para llevar a cabo las tareas de análisis, asegurando un enfoque robusto y flexible para resolver problemas tanto de clasificación como de regresión.

Para ambos problemas, se utilizaron los paquetes **mlr3pipelines**, **mlr**, **mlr3**, **mlr3filters**, **mlr3fselect**, y **mlr3learners**, los cuales proporcionan los bloques esenciales para la creación y gestión de todo el flujo de datos de machine learning implementado. Estos paquetes permiten una integración y manipulación eficientes de diferentes algoritmos de selección de características y clasificadores. Para los problemas de clasificación en particular, el paquete **e1071** fue usado para implementar Naive Bayes, el cual se empleó en todos los procesos de modelado de clasificación dentro de cada enfoque.

Dentro de los métodos embebidos, se emplearon el Random Forest y Lasso, donde la selección de características está integrada en el proceso de modelado. El paquete **glmnet** se empleó específicamente para implementar el método Lasso, mientras

que el paquete **randomForest** se empleó para la esquematización del algoritmo de Bosque Aleatorio.

En la figura 23 se presentan las estadísticas del problema de clasificación. Los resultados incluyen: el error del problema, la precisión del algoritmo y las variables seleccionadas. Estas métricas proporcionan una visión clara del rendimiento de Naive Bayes, permitiendo evaluar su efectividad en función de las características seleccionadas por cada método.

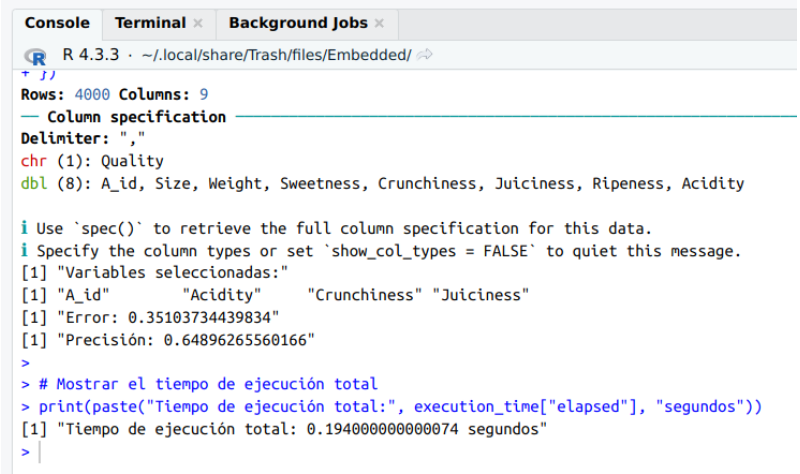


Figura 23: Estadísticas problema de clasificación en R. (Fuente propia)

La figura 24 detalla las estadísticas calculadas en el problema de regresión, incluyendo las métricas MAE (Mean Absolute Error), RMSE (Root Mean Squared Error), MSE (Mean Squared Error) y R^2 (coeficiente de determinación). Todas estas medidas ofrecen una evaluación minuciosa del rendimiento del modelo de regresión, proporcionando detalles sobre la exactitud y el error de las predicciones.

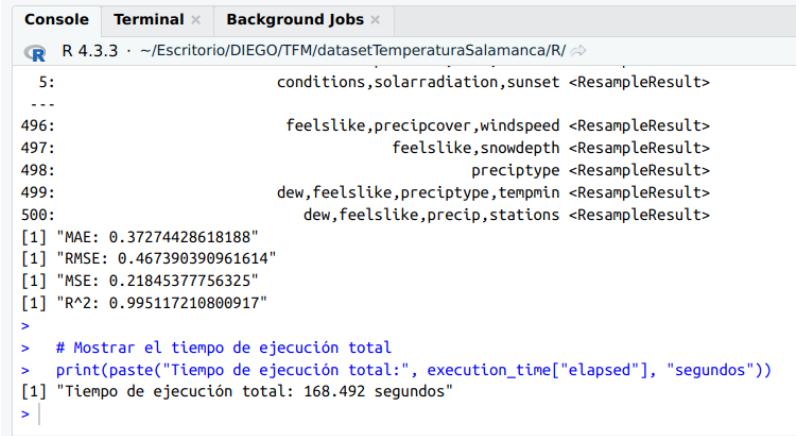


Figura 24: Estadísticas problema de regresión en R. (Fuente propia)

3.5. SciKitLearn

Esta subsección presenta la aplicación de la biblioteca SciKit-Learn, es una librería de Python ampliamente utilizada para aprendizaje automático, que proporciona herramientas simples y eficientes para el análisis predictivo de datos. En este estudio, se ha empleado para abordar todos los problemas planteados.

Para los problemas de clasificación, se ha utilizado Naive Bayes con la ayuda de la clase **GaussianNB** de SciKit-Learn. Este clasificador es adecuado para conjuntos de datos donde las características siguen una distribución normal, y es particularmente eficaz para problemas de clasificación con múltiples características independientes. En cuanto a los problemas de regresión, se ha utilizado el modelo de Regresión Lineal, implementado a través de la clase **LinearRegression** de SciKit-Learn. Este modelo es fundamental en análisis predictivo y es eficaz para describir la relación entre una variable dependiente continua y una o más variables independientes.

Los resultados de la ejecución de este código se muestran en 25. Las métricas de evaluación incluyen el error del algoritmo analizado y la precisión del modelo como se ha visto en ejemplos anteriores. Este error indica la proporción de instancias incorrectamente clasificadas, mientras que la precisión mide la proporción de instancias correctamente clasificadas.

Figura 25: Resultados en problema de clasificación con SciKit Learn. (Fuente propia)

Las métricas de evaluación generadas para los problemas de regresión son el MSE (Mean Squared Error), RMSE (Root Mean Squared Error), R^2 (coeficiente de determinación) y MAE (Mean Absolute Error), se pueden observar en la figura 26. Estas métricas proporcionan una evaluación detallada del rendimiento del modelo de regresión. El MSE mide el promedio de los errores al cuadrado, el RMSE proporciona una medida de la magnitud del error, el R^2 indica la proporción de la varianza en la variable dependiente que es explicada por las variables independientes, y el MAE mide el promedio de los errores absolutos entre las predicciones y los valores reales.

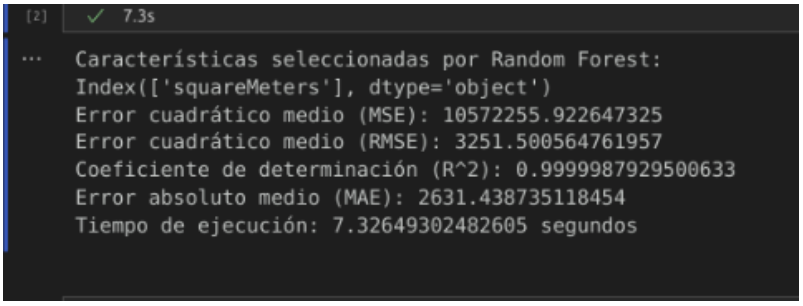


Figura 26: Resultados en problema de regresión con SciKit Learn. (Fuente propia)

4. Caso de estudio

En este caso de estudio se han seleccionado cuatro conjuntos de datos representativos, abarcando tanto problemas de clasificación como de regresión lineal como ya se ha indicado con anterioridad. Los datasets de clasificación incluyen un conjunto de datos para la categorización de setas venenosas [46] en función de si contienen veneno o no, y un conjunto de datos sobre la calidad de las manzanas [25], dividas en buenas o malas según una serie de factores. Para los problemas de regresión lineal, se ha optado por un dataset de predicción de la temperatura en Salamanca [20], y otro para la predicción del precio de la vivienda en París [56]. A continuación, se detalla el proceso de recopilación y preprocesamiento de estos datos, proporcionando una base sólida para los experimentos y análisis posteriores.

4.1. Recopilación de datos

La recopilación de datos es un paso crucial en cualquier estudio de investigación, ya que la calidad y la relevancia de estos impactan directamente en la validez y aplicabilidad de los resultados obtenidos. Para el caso de estudio, se han utilizado diversas fuentes, principalmente la plataforma Kaggle, conocida por su extensa colección de conjuntos de características para análisis y competencia en ciencia de datos, y también, la web de Visual Crossing, una plataforma especializada en análisis y visualización de datos meteorológicos.

Kaggle es una plataforma en línea que proporciona recursos y herramientas para científicos de datos y analistas. Fundada en 2010, Kaggle ofrece una amplia variedad de datasets, competiciones, tutoriales, y una comunidad activa de usuarios que colaboran y compiten en la resolución de problemas de análisis de datos. La plataforma es especialmente útil para obtener agrupaciones de atributos etiquetados y estructurados, que son esenciales para el desarrollo de modelos de machine learning. Además, Kaggle permite a los usuarios publicar sus propios conjuntos de datos y compartir sus trabajos, facilitando el intercambio de conocimientos y técnicas.

Visual Crossing ofrece acceso a una amplia gama de información meteorológica precisa y detallada, así como herramientas de análisis y visualización para comprender y utilizarla de manera efectiva. Para este estudio, Visual Crossing se

utilizó específicamente para recopilar los condicones meteorológicas relacionadas con la temperatura en Salamanca, proporcionando así información clave para el análisis y la investigación realizada.

A continuación, se detallan todas las características de los datasets analizados:

1. **Dataset de Clasificación de Setas Venenosas:** El dataset de clasificación de setas contiene información sobre 23 características de diferentes especies de setas, que ayudan a determinar si una seta es venenosa o no. Algunas de las características incluyen:
   - **Class :** si es comestible o venenosa.
   - **Cap Shape:** forma de la tapa.
   - **Cap Surface:** superficie de la tapa.
   - **Cap Color:** color de la tapa.
   - **Bruises:** presencia de moretones.
   - **Odor:** olor de la seta.
   - **Gill Attachment:** unión de las branquias.
   - **Gill Spacing:** espacio entre las branquias.
   - **Gill Size:** tamaño de las branquias.
   - **Gill Color:** color de las branquias.
   - **Stalk Shape:** forma del tallo.
   - **Stalk Root:** raíz del tallo.
   - **Stalk Surface Above Ring:** superficie del tallo por encima del anillo.
   - **Stalk Surface Below Ring:** superficie del tallo por debajo del anillo.
   - **Stalk Color Above Ring:** color del tallo por encima del anillo.
   - **Stalk Color Below Ring:** color del tallo por debajo del anillo.
   - **Veil Type:** tipo de velo.
   - **Veil Color:** color del velo.
   - **Ring Number:** número de anillos.
   - **Ring Type:** tipo de anillo.
   - **Spore Print Color:** color de la esporada.
   - **Population:** población.
   - **Habitat:** hábitat.
2. **Dataset de Calidad de las Manzanas:** Este dataset contiene 9 variables y está diseñado para evaluar la calidad de las manzanas en función de las siguientes características:
   - **A_id:** Identificador único para cada fruta.

- **Size:** Tamaño de la fruta.
- **Weight:** Peso de la fruta.
- **Sweetness:** Grado de dulzura de la fruta.
- **Crunchiness:** Textura que indica la crocancia de la fruta.
- **Juiciness:** Nivel de jugosidad de la fruta.
- **Ripeness:** Etapa de madurez de la fruta.
- **Acidity:** Nivel de acidez de las fruta.
- **Quality:** Calidad general de la fruta.

3. **Dataset de Predicción de la Temperatura en Salamanca:** En este conjunto nos encontramos con 33 atributos que proporcionan una visión detalle del clima en Salamanca. Su finalidad es analizar y compreder las condiciones meteorológicas de la ciudad así como planificar el modelado para estimar futuras predicciones. Sus propiedades son:

   - **name:** Nombre de la ubicación.
   - **datetime:** Fecha y hora de la observación.
   - **tempmax:** Temperatura máxima del día.
   - **tempmin:** Temperatura mínima del día.
   - **temp:** Temperatura media.
   - **feelslikemax:** Sensación térmica máxima.
   - **feelslikemin:** Sensación térmica mínima.
   - **feelslike:** Sensación térmica media.
   - **dew:** Punto de rocío.
   - **humidity:** Humedad relativa.
   - **precip:** Precipitación acumulada.
   - **precipprob:** Probabilidad de precipitación.
   - **precipcover:** Cobertura de precipitación.
   - **preciptype:** Tipo de precipitación (lluvia, nieve, etc.).
   - **snow:** Nieve acumulada.
   - **snowdepth:** Profundidad de la nieve.
   - **windgust:** Ráfaga de viento máxima.
   - **windspeed:** Velocidad del viento.
   - **winddir:** Dirección del viento.
   - **cloudcover:** Cobertura de nubes.
   - **visibility:** Visibilidad.
   - **solarradiation:** Radiación solar.
   - **solarenergy:** Energía solar acumulada.

- **uvindex:** Índice UV.
- **sunrise:** Hora del amanecer.
- **sunset:** Hora del atardecer.
- **moonphase:** Fase lunar.
- **conditions:** Condiciones meteorológicas generales.
- **description:** Descripción del clima.
- **icon:** Ícono representativo del clima.
- **stations:** Estaciones meteorológicas.
- **sealevelpressure:** Presión atmosférica al nivel del mar.
- **severerisk:** Riesgo de clima severo.

4. **Dataset de Predicción del Precio de la Vivienda en París:** Este dataset incluye 17 variables relacionadas con las características de los apartamentos en París y su valor predicho. Las variables son:

   - **squareMeters:** Metros cuadrados del apartamento.
   - **numberOfRooms:** Número de habitaciones.
   - **hasYard:** Indica si tiene patio.
   - **hasPool:** Indica si tiene piscina.
   - **floors:** Número de pisos.
   - **cityCode:** Código postal.
   - **cityPartRange:** Indica lo caro que es el vecindario (a mayor valor, más caro).
   - **numPrevOwners:** Número de propietarios anteriores.
   - **made:** Año de construcción.
   - **isNewBuilt:** Indica si es una construcción nueva.
   - **hasStormProtector:** Indica si tiene protector contra tormentas.
   - **basement:** Metros cuadrados del sótano.
   - **attic:** Metros cuadrados del ático.
   - **garage:** Tamaño del garaje.
   - **hasStorageRoom:** Indica si tiene cuarto de almacenamiento.
   - **hasGuestRoom:** Indica si tiene cuarto de invitados.
   - **price:** Valor del precio del apartamento.

4.2. Preprocesamiento de los datos

Este paso ha supuesto un proceso indispensable en el desarrollo de todo el trabajo realizado, garantizando la calidad y la precisión de los cálculos. El preprocesamiento incluye una serie de técnicas y transformaciones necesarias para preparar los datos en bruto antes de su análisis. Dependiendo del conjunto utilizado y de los requerimientos específicos de cada análisis, las técnicas de preprocesamiento pueden variar considerablemente.

Algunas agrupaciones, como la de las manzanas y la del precio de las viviendas en París, no requirieron ningún tipo de preprocesamiento. Estos datasets ya estaban en un formato adecuado para el análisis, con datos limpios y completos, listos para ser utilizados directamente. Los otros dos sí requirieron de una tranformación específica para poder usarlos. En el caso de las setas venenosas se han transformado sus variables predictoras de categóricas a numéricas para que los algoritmos pudieran interpretar y procesar los datos correctamente. Con respecto al dataset de la temperatura en Salamanca, se enfrentaron desafíos adicionales debido a la presencia de valores faltantes en dos columnas, lo que llevo a tener que aplicar multitud de pasos adicionales para este problema en concreto como los nodos **Missing Value** en KNIME y **Replace Missing Values** de RAPIDMINER o el método **ReplaceMissingValues** en WEKA. Para poder realizar todo el flujo de se utilizaron técnicas de imputación. Se pueden apreciar algunas de ellas en las figuras 27, 28 y 29.

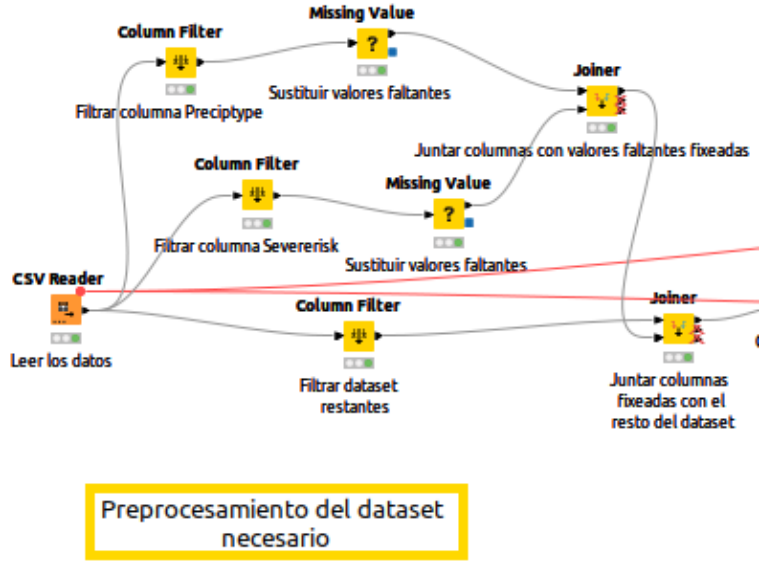


Figura 27: Preprocesamiento de los datos en KNIME. (Fuente propia)

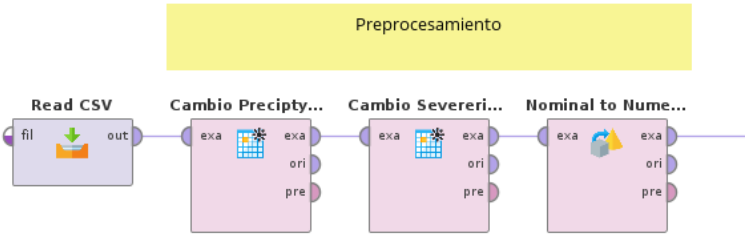


Figura 28: Preprocesamiento de los datos en RAPIDMINER. (Fuente propia)

```
# Cargar el tiempo de inicio
start_time = time.time()

# Cargar el conjunto de datos y hacer preprocesamiento
datos = pd.read_csv("/home/diego/Escritorio/DIEGO/TFM/datasetTemperaturaSalamanca/Salamanca_Spain_2021-12-01_to_2023-12-01.csv")

# Preprocesamiento: Reemplazar valores faltantes en las columnas preciptype y severerisk
datos['preciptype'] = datos['preciptype'].replace('?', 'none')
most_common_severerisk = datos['severerisk'].mode()[0]
datos['severerisk'] = datos['severerisk'].replace('?', most_common_severerisk)
```

Figura 29: Preprocesamiento de los datos en SCIKIT. (Fuente propia)

Dado que uno de los objetivos del trabajo implicaba el uso de la plataforma Weka, fue necesario desarrollar un programa en Python para convertir todos los archivos .csv a .arff, ya que, Weka únicamente trabaja con este tipo de archivos.

5. Resultado de los experimentos

Este apartado tiene como objetivo presentar los resultados obtenidos de los experimentos realizados utilizando diferentes algoritmos de selección de características y modelado predictivo. Los experimentos se llevaron a cabo en diversos datasets, evaluando tanto el tiempo de ejecución como la precisión de los algoritmos implementados. A continuación se detallan los resultados obtenidos para cada una de estas métricas.

5.1. Tiempo de ejecución de los algoritmos implementados

En el contexto del machine learning, el tiempo de ejecución se refiere al tiempo que tarda un algoritmo en seleccionar un subset de características relevantes a partir de un conjunto de datos original. Este factor es crucial para la evaluación del rendimiento de diferentes algoritmos y en este estudio nos ayuda a comprender mejor las diferencias entre los distintos enfoques, ya que puede influir significativamente en la viabilidad y aplicabilidad de un método en escenarios prácticos.

Con el objetivo de realizar una comparativa exhaustiva del rendimiento de los algoritmos de selección de características implementados, se ha optado por la elaboración de gráficos de barras individuales para cada algoritmo y conjunto de datos analizado. Esta representación gráfica, con una escala que abarca desde 0 hasta 500 segundos, permite una visualización clara y precisa del tiempo de ejecución de cada algoritmo en cada escenario experimental. De este modo, se facilita la identificación de los algoritmos que presentan un mayor tiempo de ejecución en comparación con los demás.

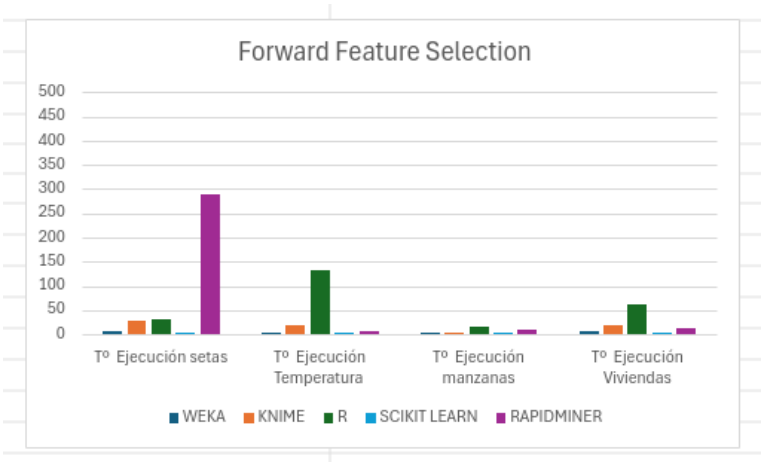


Figura 30: Tiempos de ejecución con Forward Feature Selection. (Fuente propia)

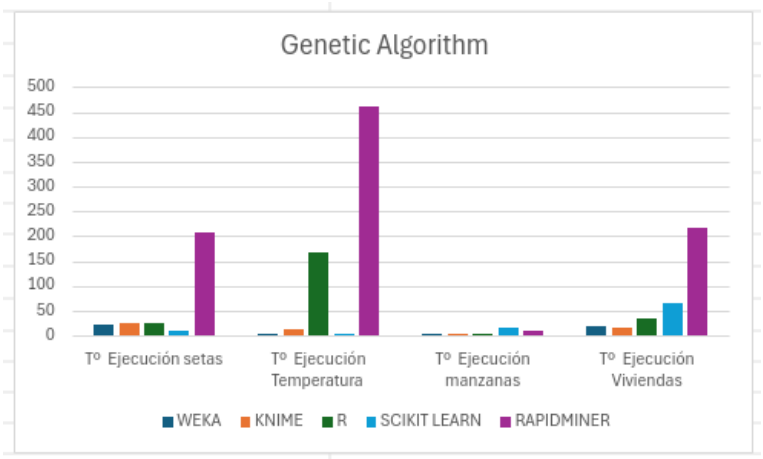


Figura 31: Tiempos de ejecución con Genetic Algorithm. (Fuente propia)

Como se puede observar en las figuras 31 y 30, los algoritmos del enfoque wrapper son los que presentan un mayor consumo de recursos computacionales. Entre ellos, el Algoritmo Genético destaca como el más costoso (exceptuando el valor atípico encontrado en Relief en la figura 35), con un tiempo máximo de ejecución de 462 segundos en el conjunto de datos "Temperatura"(RapidMiner) y una media de 65 segundos por ejecución. La Selección Secuencial Hacia Delante se posiciona como el segundo algoritmo con mayor tiempo de ejecución, alcanzando un máximo de 290 segundos en el conjunto de datos "Setasz una media de 32 segundos por ejecución.

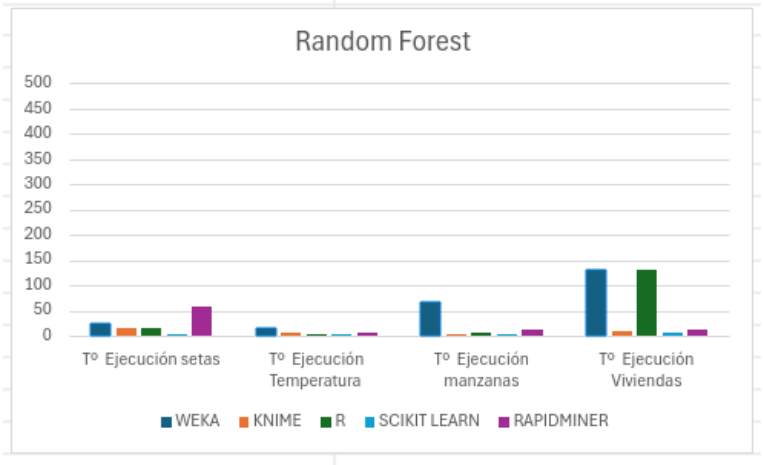


Figura 32: Tiempos de ejecución con el algoritmo Random Forest. (Fuente propia)

Los algoritmos del enfoque embebido, si bien no son tan costosos computacionalmente como los del enfoque wrapper, presentan un consumo de recursos notable. El Random Forest 32 tiene una media de 26 segundos por ejecución, mientras que el algoritmo LASSO 33. presenta una media de 14 segundos por ejecución. En cuanto al enfoque filter, el algoritmo de Correlación, mostrado en la figura 34 destaca por tener el tiempo de ejecución más bajo, con una media de 2 segundos por ejecución, mientras que el algoritmo Relief, el cual podemos ver la figura 35 destaca por un

valor atípico encontrado en el dataset de las Setas Venenosas cuando se ha ejecutado con RAPIDMINER (2605 segundos).

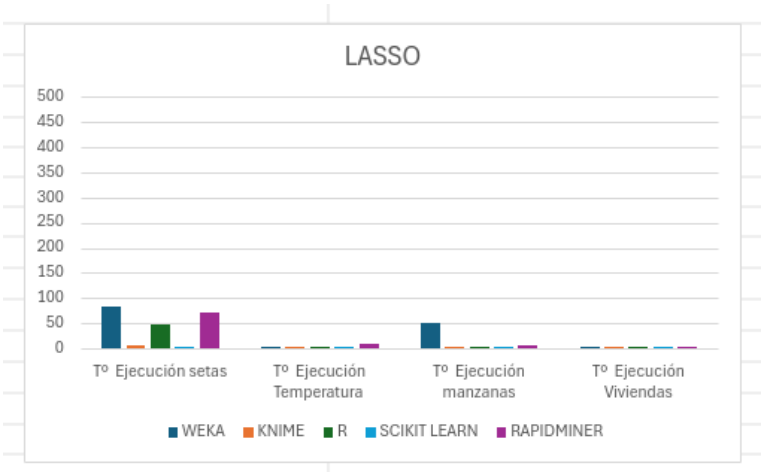


Figura 33: Tiempos de ejecución con LASSO. (Fuente propia)

Un dato que afecta negativamente a este enfoque y que pone en duda la fiabilidad de ejecuciones en esta herramienta. En la figura 36 se puede ver la gráfica con el tiempo de ejecución en la escala principal y aún así los resultados en comparativa son mucho más elevados que los de su homólogo para ser un enfoque basado en filtros. La eficiencia de estos algoritmos será discutida en la siguiente sección, donde se analizarán las comparativas del error entre enfoques y la relación entre el tiempo de ejecución y el error promedio.

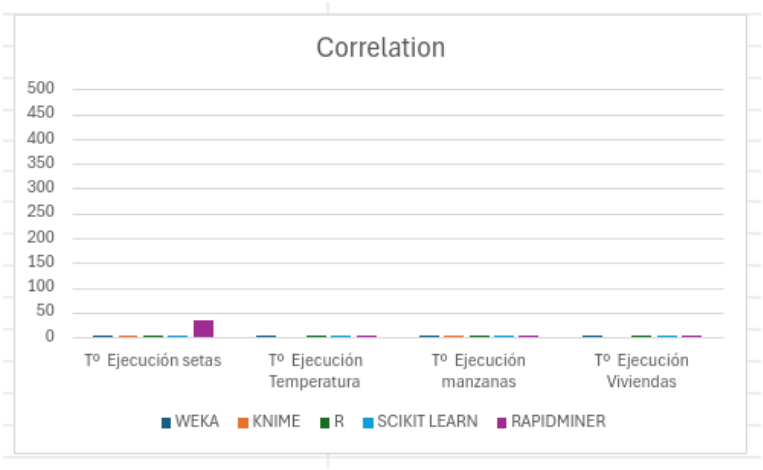


Figura 34: Tiempos de ejecución con Correlation. (Fuente propia)

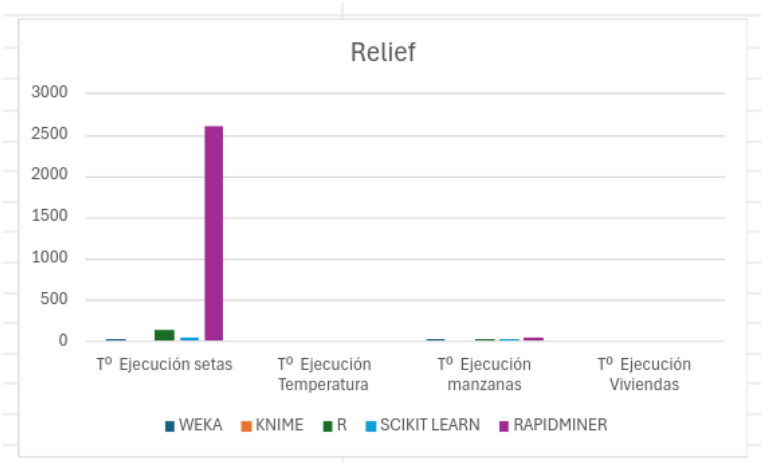


Figura 35: Tiempos de ejecución con Relief. (Fuente propia)

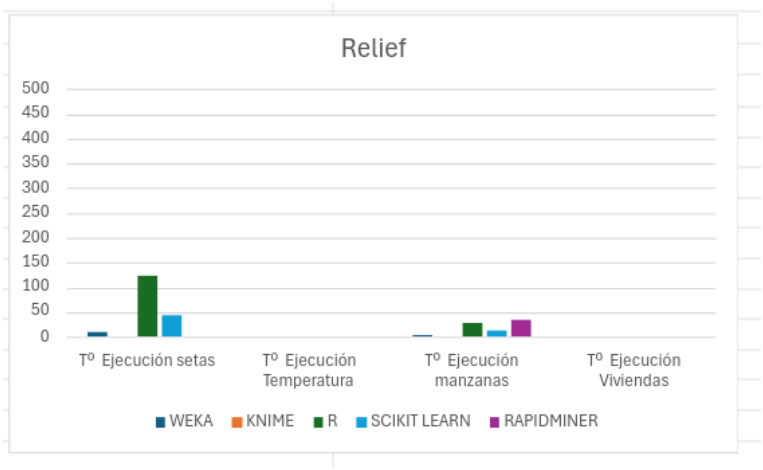


Figura 36: Tiempos de ejecución con Relief sin el valor atípico. (Fuente propia)

5.2. Precisión de los algoritmos implementados

La precisión de los algoritmos implementados se evalúa mediante una comparativa del error obtenido, utilizando cada uno de los datasets. A continuación, se presentan los resultados generados en forma de gráficos de barras agrupados por los métodos empleados, mostrando el error en cada una de las herramientas. Las tablas 2 y 3, se presenta el error de clasificación para los datasets de setas y manzanas, respectivamente. Es importante destacar que la gráfica muestra el error en una escala de 0 a 1 por ser estos los datasets relativos a los problemas de clasificación en cuyo caso el error varía dentro de esos rangos, lo cual es crucial para interpretar los resultados. Para el dataset de las setas, los algoritmos del enfoque wrapper muestran la mayor precisión, con errores mínimos.

Tabla 2: Comparativa error en diferentes plataformas con dataset de las setas.

Herramienta	**Random Forest**	**LASSO R1**	**Forward Feature Selection**	**Genetic Algorithm**	**Correlation**	**Relief**
WEKA	0.9857	0.9788	0.977	0.9835	0.8621	0.8982
KNIME	0.92	0.893	0.973	0.965	0.628	X
R	0.921	0.895	0.893	0.89	0.9	0.886
SCIKIT LEARN	0.8861	0.8787	0.795	0.9415	0.637	0.8289
RAPIDMINER	0.97	1	0.979	0.9964	0.886	0.993

Sin embargo, los algoritmos de enfoque embedded prácticamente igualan esta precisión, demostrando un rendimiento muy competitivo entre ambos, lo que no esclarece a estos efectos si es mejor usar unos u otros en términos de precisión y dejando el punto de inflexión en la relación de estas magnitudes con las soluciones obtenidas en los tiempos de ejecución. En contraste, el enfoque filter se muestra claramente inferior en ambos planteamientos, con los errores más elevados en todas las plataformas. En concreto, el algoritmo de Correlación destaca por sus resultados en KNIME y SCIKIT al ser estos dos los peores de toda la tanda de ejecuciones. En el dataset de las manzanas, hay un claro empate entre los enfoques wrapper y embedded, con diferencias prácticamente insignificantes en sus errores de clasificación. Al igual que con el dataset anterior, los algoritmos del enfoque filter tienen un rendimiento notablemente menor, pero no tan marcado como lo hacían antes. Lo que sí podemos afirmar de esta comparativa, es que a mayor número de variables (el dataset de las setas tiene 23 frente a las 9 del conjunto de las manzanas) los resultados son mucho mejores, rozando prácticamente la perfección y los modelos tienen más capacidad de decisión al contar con un abanico de características más grande de posibles soluciones.

Tabla 3: Comparativa error en diferentes plataformas con dataset de las manzanas.

Herramienta	Random Forest	LASSO R1	Forward Feature Selection	Genetic Algorithm	Correlation	Relief
WEKA	0.72425	0.74725	0.74825	0.7575	0.678	0.64275
KNIME	0.7447	0.76212	0.74545	0.7303	0.66667	X
R	0.7435	0.7195	0.73029	0.6489	0.6489	0.648962
SCIKIT LEARN	0.7625	0.745	0.72625	0.7525	0.73625	0.76125
RAPIDMINER	0.86	0.865	0.87	0.827	0.79	0.795

Con respecto a los problemas de regresión, se examinan los datasets de predicción de temperaturas y de estimación de precios de viviendas. En el dataset de las temperaturas (Tabla 4), el valor de Mean Absolute Error (MAE) es utilizado para evaluar la precisión. Las medias de los valores obtenidos son las siguientes: Random Forest (0.25582), LASSO (0.25462), Forward Feature Selection (0.27774), Genetic Algorithm (0.27694) y Correlation (0.50522). Para este Dataset en concreto, LASSO presenta el menor MAE, seguido muy de cerca por Random Forest, lo que indica una alta precisión por parte del enfoque Embedded. Los algoritmos Forward Feature Selection y Genetic Algorithm también muestran una precisión similar, aunque ligeramente inferior a los dos primeros. El enfoque basado en Correlation, con un MAE significativamente mayor, resulta ser el menos preciso para este problema.

En referencia al dataset de los precios de las viviendas (Tabla 5), las medias de los valores de MAE son las siguientes:

- Random Forest 9811.7618.
- LASSO 9270.682.
- Selección secuencial hacia delante 9064.5752.
- Genetic Algorithm 9475.3324.
- Correlation 12482.0935.

Aquí, los datos se ven afectados por esos valores atípicos en las predicciones efectuadas con RAPIDMINER, que elevan de forma considerable el error medio dejando una sensación confusa en la medición. Por este motivo se ha llevado a cabo una segunda interpretación de los resultados sin tener en cuenta esta herramienta, lo que ha ajustado significativamente los resultados y ha permitido medir con exactitud el resto de computos llevados a cabo.

Tabla 4: Comparativa error en diferentes plataformas con dataset de la temperatura.

Herramienta	**Random Forest**	**LASSO R1**	**Forward Feature Selection**	**Genetic Algorithm**	**Correlation**
WEKA	0.2081	0.2081	0.2103	0.2117	0.2081
KNIME	0.252	0.46	0.216	0.218	1.447
R	0.234	0.38	0.2277	0.372	0.464
SCIKIT LEARN	0.4	4.72E-28	0.3447	0.345	9.76E-27
RAPIDMINER	0.185	0.225	0.39	0.238	0.407

Tabla 5: Comparativa error en diferentes plataformas con dataset de las viviendas.

Herramienta	**Random Forest**	**LASSO R1**	**Forward Feature Selection**	**Genetic Algorithm**	**Correlation**
WEKA	2319.63	1479.49	1478.73	1478.85	2644.38
KNIME	2644.409	2688.48	1489.266	1461.968	1493,43
R	2686.28	2686.28	1490.64	2345.45	2686.286
SCIKIT LEARN	2631.43	1507.67	1515.88	1484.64	2631.43
RAPIDMINER	38777.06	37991.49	39348.36	40605.754	41966.278

Este ajuste revela una perspectiva diferente, ya que al no contar con la última fila de valores, el enfoque wrapper se vuelve mucho más eficiente en comparación con los otros. Forward Feature Selection con una media de error absoluto de 1493.63 y Genetic Algorithm con 1692.727 se ha convierten en los dos mejores opciones para este conjunto siendo como los más precisos superando sin competencia a LASSO y Random Forest con medias de 2570.43 y 2090.48 respectivamente en el enfoque embebido y a la Correlacion con 2363.8815 en el enfoque filter. Todo esto puede derivarse de la tabla 5. De nuevo y al igual que en los datasets de clasificación, esta comparativa deja la sensación de que, a más variables haya en el dataset, más precisas serán las métricas al poder elegir entre una gama más amplia de datos.

6. Conclusiones y líneas de trabajo futuras

6.1. Conclusiones

En este trabajo se han comparado tres enfoques principales de selección de características: filtro, envoltura y embebidos, implementados en cinco plataformas diferentes (KNIME, R, Scikit Learn, RapidMiner y Weka), utilizando dos datasets de clasificación y dos de regresión. Los objetivos eran evaluar la eficiencia en términos de tiempo de ejecución, la precisión de los algoritmos, la consistencia de los resultados entre diferentes plataformas y, finalmente, identificar las combinaciones más efectivas de algoritmos y plataformas.

Los resultados obtenidos permiten extraer las siguientes conclusiones en función de los objetivos buscados:

- **Comparar la Eficiencia de Algoritmos de Selección de Características:**
  - Los métodos de selección de características embebidos demostraron ser los más eficientes en la relación tiempo de ejecución/precisión. Al integrarse directamente en el proceso de construcción del modelo, estos algoritmos requieren menos tiempo de procesamiento en comparación con los métodos de envoltura que logran mejores resultados que estos y los métodos basados en filtros.
  - Los métodos de envoltura, aunque ofrecen una alta precisión, resultaron ser los más costosos computacionalmente debido a la necesidad de evaluar múltiples subconjuntos de características utilizando un modelo predictivo.
  - El enfoque basado en filtros, aunque ha demostrado ser el más rápido en tiempo de ejecución, no ha conseguido la misma precisión que los métodos de envoltura y los embebidos, pasando a un segundo plano si lo que se prioriza es la eficiencia.
- **Evaluar la Precisión de Algoritmos de Selección de Características:**
  - Los algoritmos de envoltura proporcionaron la mayor precisión general. Dependiendo de la herramienta utilizada, en algunos casos los métodos de envoltura superaron a los embebidos en precisión y en otros fue al contrario. En promedio, los métodos de envoltura demostraron ser ligeramente mejores en términos de precisión. Aun así hay que destacar al enfoque embebido, ya que ofrece resultados comparables con los conseguidos en los wrappers.
  - El enfoque filter ha logrado proporcionar una tasa de error considerablemente baja y se ha acercado a los resultados de los otros dos, dejando entre ver que puede ser una alternativa si no se dispone de recursos computacionales altos.

- **Analizar la Consistencia de Resultados entre Diferentes Plataformas:**
  - Se ha observado uniformidad en casi todos los resultados obtenidos al aplicar los mismos algoritmos de selección de características en diferentes plataformas para cada algoritmo analizado, lo que lleva a convicción en cuanto a fiabilidad de las métricas obtenidas y las posibles interpretaciones.
  - Hay pequeñas variaciones debido a las implementaciones específicas de cada plataforma, como es el caso de RAPIDMINER para los resultados obtenidos en cuanto al error en dataset de los precios de las viviendas o el tiempo de ejecución del algoritmo Relief con el dataset de las setas venenosas. Esto lleva a plantear dudas en cuanto a la fiabilidad de futuros estudios relacionados con la aplicación. Habría que seguir realizando experimentos con más datos para verificar la discrepancia.
- **Identificar las Mejores Combinaciones de Algoritmos y Plataformas:**
  - Las combinaciones más efectivas en términos de balance entre tiempo de ejecución y precisión de este estudio han sido los métodos embebidos, en plataformas como Scikit Learn y KNIME las cuales proporcionaron una buena integración y optimización.
  - Los métodos de envoltura aunque comparables, han sido más adecuados en las plataformas Scikit Learn y WEKA.
  - Para aplicaciones que requieren una rápida preselección de características, los métodos de filtro fueron recomendados en todas las plataformas excepto RapidMiner, debido a su velocidad y simplicidad.
  - Con todos los resultados, Scikit Learn corona como la mejor herramienta por su versatilidad, fiabilidad y robustez en las implementaciones llevadas a cabo.

6.2. Líneas de investigación futuras

Las conclusiones de este trabajo dejan las puertas abiertas para futuras investigaciones. Una línea de investigación prometedora es la exploración de algoritmos híbridos, que combinan las fortalezas de diferentes enfoques como filtro, envoltura y embebidos para mejorar la precisión y la eficiencia en tiempo de ejecución. Dado que cada enfoque tiene sus propias ventajas y desventajas, una combinación inteligente puede potencialmente superar las limitaciones individuales. Desarrollar y evaluar algoritmos híbridos que integren métodos de filtro como una etapa preliminar para reducir el espacio de características seguido por métodos de envoltura o embebidos para refinar la selección podría ser particularmente útil en datasets de alta dimensionalidad, donde una reducción inicial de todos los atributos existentes puede hacer más manejables los métodos más intensivos en computación.

Otra dirección importante es la evaluación en diversos conjuntos de datos. La eficacia de los algoritmos de selección de características puede variar significativamente

dependiendo del tipo de datos y del dominio de aplicación. Ampliar el estudio a una mayor variedad de datasets permitirá una mejor comprensión de la capacidad de generalización de los resultados. Continuar probando los algoritmos estudios u otros nuevos en otros conjuntos de datos que abarquen diferentes tipos de problemas como clasificación, regresión, series temporales, imágenes y texto es esencial. Esto incluye evaluar el rendimiento en datasets con distintas características como tamaños variados, niveles de ruido y distribuciones de los datos. Dentro de esas avaluaciones, incorporar concretamente al clustering añadiría mucho valor a la investigación, permitiendo comparar resultados anteriores con otras perspectivas nuevas y diferentes.

Puede llegar a ser muy valioso considerar el impacto de la calidad de los datos como, por ejemplo, presencia de ruido o de datos faltantes dentro del dataset. Estas condiciones pueden afectar significativamente al rendimiento del modelo. Es crucial entender cómo estos factores influyen en los algoritmos y desarrollar técnicas para mitigar sus efectos. Analizar el impacto de diferentes niveles de ruido y patrones de datos inexistentes e investigar métodos robustos que puedan manejar eficazmente datos imperfectos son áreas de gran interés. Además, la integración de técnicas de limpieza y preprocesamiento de datos como etapas preliminares puede ser beneficiosa.

Referencias

[1] Metodos de seleccion de caracteristicas 2 | ligdimar gonzalez | flickr. [*Citado en págs. v, v, 6 y 9.*]

[2] Processo de busca no espaço de estados de subconjuntos de atributos [11]. | download scientific diagram. [*Citado en págs. v y 13.*]

[3] Random forest classifier and its hyperparameters | by ankit chauhan | analytics vidhya | medium. [*Citado en págs. v y 14.*]

[4] Naoual El Aboudi and Laila Benhlima. Review on wrapper feature selection approaches. *Proceedings - 2016 International Conference on Engineering and MIS, ICEMIS 2016*, 11 2016. [*Citado en pág. 12.*]

[5] Afek Ilay Adler and Amichai Painsky. Feature importance in gradient boosting trees with cross-validation feature selection. *Entropy 2022, Vol. 24, Page 687*, 24:687, 5 2022. [*Citado en pág. 21.*]

[6] Taqwa Ahmed Alhaj, Maheyzah Md Siraj, Anazida Zainal, Huwaida Tagelsir Elshoush, and Fatin Elhaj. Feature selection using information gain for improved structural-based alert correlation. *PLOS ONE*, 11:e0166017, 11 2016. [*Citado en pág. 8.*]

[7] M. Anand, Kishan Bhushan Sahay, Mohammed Altaf Ahmed, Daniyar Sultan, Radha Raman Chandan, and Bharat Singh. Deep learning and natural language processing in computation for offensive language detection in online social networks by feature selection and ensemble classification techniques. *Theoretical Computer Science*, 943:203–218, 1 2023. [*Citado en pág. 5.*]

[8] Jun Chin Ang, Andri Mirzal, Habibollah Haron, and Haza Nuzly Abdull Hamed. Supervised, unsupervised, and semi-supervised feature selection: A review on gene selection. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 13:971–989, 9 2016. [*Citado en pág. 14.*]

[9] Youssef Asnaoui, Yassine Akhiat, and Ahmed Zinedine. Feature selection based on attributes clustering. *5th International Conference on Intelligent Computing in Data Sciences, ICDS 2021*, 2021. [*Citado en pág. 18.*]

[10] George S. Atsalakis and Kimon P. Valavanis. Surveying stock market forecasting techniques – part ii: Soft computing methods. *Expert Systems with Applications*, 36:5932–5941, 4 2009. [*Citado en pág. 1.*]

[11] Oluleye Babatunde, Leisa Armstrong, Jinsong Leng, and Dean Diepeveen. A genetic algorithm-based feature selection. *Research outputs 2014 to 2021*, 1 2014. [*Citado en pág. 10.*]

[12] Daniel Berrar. Cross-validation. *Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics*, 1-3:542–545, 1 2018. [*Citado en pág. 19.*]

[13] Verónica Bolón-Canedo, Noelia Sánchez-Maroño, and Amparo Alonso-Betanzos. A review of feature selection methods on synthetic data. *Knowledge and Information Systems*, 34:483–519, 3 2013. [*Citado en pág. 2.*]

[14] Florentina Bunea, Yiyuan She, and Marten H. Wegkamp. Optimal selection of reduced rank estimators of high-dimensional matrices. *The Annals of Statistics*, 39, 4 2010. [*Citado en pág. 15.*]

[15] Girish Chandrashekar and Ferat Sahin. A survey on feature selection methods. *Computers and Electrical Engineering*, 40:16–28, 1 2014. [*Citado en pág. 2.*]

[16] Chao Chen and Andy Liaw. Using random forest to learn imbalanced data. *2004*, 2004. [*Citado en pág. 2.*]

[17] Shihan Chen, Yuanjian Yang, Fei Deng, Yanhao Zhang, Duanyang Liu, Chao Liu, and Zhiqiu Gao. A high-resolution monitoring approach of canopy urban heat island using a random forest model and multi-platform observations. *Atmospheric Measurement Techniques*, 15:735–756, 2 2022. [*Citado en pág. 14.*]

[18] Zheng Chen, Meng Pang, Zixin Zhao, Shuainan Li, Rui Miao, Yifan Zhang, Xiaoyue Feng, Xin Feng, Yexian Zhang, Meiyu Duan, Lan Huang, and Fengfeng Zhou. Feature selection may improve deep neural networks for the bioinformatics problems. *Bioinformatics*, 36:1542–1552, 3 2020. [*Citado en pág. 5.*]

[19] Chiranji Lal Chowdhary and D. P. Acharjya. Segmentation and feature extraction in medical imaging: A systematic review. *Procedia Computer Science*, 167:26–36, 1 2020. [*Citado en pág. 5.*]

[20] Visual Crossing Corporation. Salamanca temperature prediction dataset. *https://www.visualcrossing.com/*, 2024. [*Citado en pág. 33.*]

[21] Milan Decuyper, Mariele Stockhoff, Stefaan Vandenberghe, al, and Xue Ying. An overview of overfitting and its solutions. *Journal of Physics: Conference Series*, 1168:022022, 2 2019. [*Citado en pág. 21.*]

[22] Włodzisław Duch. Filter methods. *Studies in Fuzziness and Soft Computing*, 207:89–117, 2006. [*Citado en pág. 6.*]

[23] Richard O Duda, Peter E Hart, and David G.Stork. (pdf) pattern classification. 2001. [*Citado en pág. 1.*]

[24] Bradley Efron, Trevor Hastie, Iain Johnstone, Robert Tibshirani, Hemant Ishwaran, Keith Knight, Jean Michel Loubes, Pascal Massart, David Madigan, Greg Ridgeway, Saharon Rosset, J. I. Zhu, Robert A. Stine, Berwin A. Turlach, Sanford Weisberg, and Iain Johnstone. Least angle regression. *https://doi.org/10.1214/009053604000000067*, 32:407–499, 4 2004. [*Citado en pág. 16.*]

[25] Nidula Elgiriyewithana. Apple quality dataset. *https://www.kaggle.com/datasets/nelgiriyewithana/apple-quality*, 2024. [*Citado en pág. 33.*]

[26] Mohammad Mahdi Ershadi and Abbas Seifi. Applications of dynamic feature selection and clustering methods to medical diagnosis. *Applied Soft Computing*, 126:109293, 9 2022. [*Citado en pág. 5.*]

[27] Ya Ju Fan and Wanpracha Art Chaovalitwongse. Optimizing feature selection to improve medical diagnosis. *Annals of Operations Research*, 174:169–183, 2 2010. [*Citado en pág. 5.*]

[28] Macedo Francisco, Oliveira M. Rosário, Pacheco António, and Valadas Rui. Theoretical foundations of forward feature selection methods based on mutual information - sciencedirect, 2019. [*Citado en pág. 11.*]

[29] Feng Gao, J. Philip Miller, Chengjie Xiong, Jingqin Luo, Julia A. Beiser, Ling Chen, and Mae O. Gordon. Estimating correlation between multivariate longitudinal data in the presence of heterogeneity. *BMC Medical Research Methodology*, 17:1–11, 8 2017. [*Citado en pág. 7.*]

[30] Daniel G Goldstein, Jake M Hofman, Microsoft Research JENNIFER WORTMAN VAUGHAN, Forough Poursabzi-Sangdeh, Jennifer Wortman Vaughan, and Hanna Wallach. Manipulating and measuring model interpretability. *Conference on Human Factors in Computing Systems - Proceedings*, 67, 2 2018. [*Citado en pág. 23.*]

[31] Quanquan Gu, Zhenhui Li, and Jiawei Han. Generalized fisher score for feature selection, 2011. [*Citado en pág. 8.*]

[32] Isabelle Guyon and Andre@tuebingen Mpg De. An introduction to variable and feature selection andré elisseeff. *Journal of Machine Learning Research*, 3:1157–1182, 2003. [*Citado en pág. 1.*]

[33] Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46:389–422, 2002. [*Citado en pág. 1.*]

[34] Michal Haindl, Petr Somol, Dimitrios Ververidis, and Constantine Kotropoulos. Feature selection based on mutual correlation. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 4225 LNCS:569–577, 2006. [*Citado en pág. 7.*]

[35] Johanna Hardin, Aya Mitani, Leanne Hicks, and Brian VanKoten. A robust measure of correlation between two genes on a microarray. *BMC Bioinformatics*, 8:1–13, 6 2007. [*Citado en pág. 7.*]

[36] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. The elements of statistical learning. 2009. [*Citado en pág. 1.*]

[37] Trevor Hastie, Robert Tibshirani, and Martin Wainwright. Statistical learning with sparsity: The lasso and generalizations. *Statistical Learning with Sparsity: The Lasso and Generalizations*, pages 1–337, 1 2015. [*Citado en págs. 15 y 16.*]

[38] Suriyan Jomthanachai, Wai Peng Wong, and Khai Wah Khaw. An application of machine learning regression to feature selection: a study of logistics performance and economic attribute. *Neural Computing and Applications*, 34:15781–15805, 4 2022. [*Citado en pág. 5.*]

[39] A. Jović, K. Brkić, and N. Bogunović. A review of feature selection methods with applications. *2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2015 - Proceedings*, pages 1200–1205, 7 2015. [*Citado en págs. 2 y 9.*]

[40] Tushar Kansal, Suraj Bahuguna, Vishal Singh, and Tanupriya Choudhury. Customer segmentation using k-means clustering. *Proceedings of the International Conference on Computational Techniques, Electronics and Mechanical Systems, CTEMS 2018*, pages 135–139, 12 2018. [*Citado en pág. 18.*]

[41] Jon Kleinberg, Himabindu Lakkaraju, Jure Leskovec, Jens Ludwig, and Sendhil Mullainathan. Human decisions and machine predictions. *The Quarterly Journal of Economics*, 133:237–293, 2 2018. [*Citado en pág. 22.*]

[42] Ron Kohavi. (pdf) a study of cross-validation and bootstrap for accuracy estimation and model selection, 2001. [*Citado en pág. 20.*]

[43] Ron Kohavi and George H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97:273–324, 12 1997. [*Citado en pág. 2.*]

[44] K.Kranthi Kumar, Kavitha Chaduvula, and Babu RAO Markapudi. (pdf) a detailed survey on feature extraction techniques in image processing for medical image analysis, 2020. [*Citado en pág. 5.*]

[45] Thomas Navin Lal, Olivier Chapelle, Jason Western, and André Elisseeff. Embedded methods. *Studies in Fuzziness and Soft Computing*, 207:137–165, 2006. [*Citado en pág. 13.*]

[46] UCI Machine Learning. Mushroom classification dataset. *https://www.kaggle.com/datasets/uciml/mushroom-classification*, 2017. [*Citado en pág. 33.*]

[47] Huei Diana Lee, Maria-Carolina Monard, and José Augusto Baranauskas. (pdf) empirical comparison of wrapper and filter approaches for feature subset selection, 1999. [*Citado en pág. 9.*]

[48] Jundong Li, Kewei Cheng, Suhang Wang, Fred Morstatter, Robert P. Trevino, Jiliang Tang, and Huan Liu. Feature selection. *ACM Computing Surveys (CSUR)*, 50, 12 2017. [*Citado en pág. 2.*]

[49] Shanshan Li, Jian Yu, Huimin Kang, and Jianfeng Liu. Genomic selection in chinese holsteins using regularized regression models for feature selection of whole genome sequencing data. *Animals 2022, Vol. 12, Page 2419*, 12:2419, 9 2022. [*Citado en pág. 5.*]

[50] Huan Liu and Hiroshi Motoda. Feature selection for knowledge discovery and data mining. *The Springer International Series in Engineering and Computer Science*, 1998. [*Citado en pág. 2.*]

[51] Huan Liu and Hiroshi Motoda. Computational methods of feature selection. *Computational Methods of Feature Selection*, pages 1–419, 10 2007. [*Citado en pág. 8.*]

[52] Scott M. Lundberg and Su In Lee. A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, 2017-December:4766–4775, 5 2017. [*Citado en pág. 22.*]

[53] Mitchell Melanie. An introduction to genetic algorithms. 1. genetics-computer simulation.2. genetics-mathematical models. 1999. [*Citado en pág. 10.*]

[54] Praveen Kumar Misra, Narendra Kumar, Anuradha Misra, and Alex Khang. Heart disease prediction using logistic regression and random forest classifier. *Data-Centric AI Solutions and Emerging Technologies in the Healthcare Ecosystem*, pages 83–112, 1 2023. [*Citado en pág. 14.*]

[55] Dunja Mladenić. Feature selection for dimensionality reduction. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 3940 LNCS:84–102, 2006. [*Citado en pág. 22.*]

[56] MsSmartyPants. Paris housing price prediction dataset. *https://www.kaggle.com/datasets/mssmartypants/paris-housing-price-prediction*, 2021. [*Citado en pág. 33.*]

[57] Sendhil Mullainathan and Jann Spiess. Machine learning: An applied econometric approach. *Journal of Economic Perspectives*, 31:87–106, 3 2017. [*Citado en pág. 15.*]

[58] F Fadaei Noghani and M.-H Moattar. Ensemble classification and extended feature selection for credit card fraud detection. *Journal of AI and Data Mining*, 5:235–243, 7 2017. [*Citado en pág. 5.*]

[59] Marlena Osipowicz, Bartek Wilczynski, and Magdalena A. MacHnicka. Careful feature selection is key in classification of alzheimer's disease patients based on whole-genome sequencing data. *NAR Genomics and Bioinformatics*, 3, 6 2021. [*Citado en pág. 5.*]

[60] David Reinsel, John Gantz, and John Rydning. The digitization of the world from edge to core. 2018. [*Citado en págs. v y 1.*]

[61] Juha Reunanen and Isabelle Guyon. (pdf) overfitting in making comparisons between variable selection methods, 2003. [*Citado en pág. 21.*]

[62] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?": Explaining the predictions of any classifier. *NAACL-HLT 2016 - 2016*

Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Demonstrations Session, pages 97–101, 2 2016. [*Citado en pág. 22.*]

[63] Tibshirani Robert. Regression shrinkage and selection via the lasso on jstor, 1996. [*Citado en pág. 2.*]

[64] Yvan Saeys, Iñaki Inza, and Pedro Larrañaga. A review of feature selection techniques in bioinformatics. *Bioinformatics (Oxford, England)*, 23:2507–2517, 10 2007. [*Citado en pág. 2.*]

[65] Chenhui Shao, Kamran Paynabar, Tae Hyung Kim, Jionghua Jin, S. Jack Hu, J. Patrick Spicer, Hui Wang, and Jeffrey A. Abell. Feature selection for manufacturing process monitoring using cross-validation. *Journal of Manufacturing Systems*, 32:550–555, 10 2013. [*Citado en pág. 21.*]

[66] Jinhong Shi, Yan Yan, Matthew G. Links, Longhai Li, Jo Anne R. Dillon, Michael Horsch, and Anthony Kusalik. Antimicrobial resistance genetic factor identification from whole-genome sequence data using deep feature selection. *BMC Bioinformatics*, 20:1–14, 12 2019. [*Citado en pág. 5.*]

[67] Swati Shilaskar and Ashok Ghatol. Feature selection for medical diagnosis : Evaluation for cardiovascular diseases. *Expert Systems with Applications*, 40:4146–4153, 8 2013. [*Citado en pág. 5.*]

[68] Noah Simon, Jerome Friedman, Trevor Hastie, and Rob Tibshirani. Regularization paths for cox's proportional hazards model via coordinate descent. *Journal of statistical software*, 39:1–13, 3 2011. [*Citado en pág. 16.*]

[69] Ajeet Singh and Anurag Jain. Adaptive credit card fraud detection techniques based on feature selection method. *Advances in Intelligent Systems and Computing*, 924:167–178, 2019. [*Citado en pág. 5.*]

[70] Rabindra Kumar Singh and M. Sivabalakrishnan. Feature selection of gene expression data for cancer classification: A review. *Procedia Computer Science*, 50:52–57, 1 2015. [*Citado en pág. 5.*]

[71] R. Steuer, J. Kurths, C. O. Daub, J. Weise, and J. Selbig. The mutual information: detecting and evaluating dependencies between variables. *Bioinformatics (Oxford, England)*, 18 Suppl 2, 2002. [*Citado en pág. 7.*]

[72] Jozsef Suto, Stefan Oniga, and Petrica Pop Sitar. Comparison of wrapper and filter feature selection algorithms on human activity recognition. *2016 6th International Conference on Computers Communications and Control, ICCCC 2016*, pages 124–129, 6 2016. [*Citado en pág. 9.*]

[73] Jun Suzuki, Hideki Isozaki, and Eisaku Maeda. Convolution kernels with feature selection for natural language processing tasks. *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 119–126, 2004. [*Citado en pág. 5.*]

[74] Noelia Sánchez-Maroño, Amparo Alonso-Betanzos, and María Tombilla-Sanromán. Filter methods for feature selection - a comparative study. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 4881 LNCS:178–187, 2007. [*Citado en pág. 6.*]

[75] Luis Talavera. An evaluation of filter and wrapper methods for feature selection in categorical clustering. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 3646 LNCS:440–451, 2005. [*Citado en pág. 9.*]

[76] Jiliang Tang, Salem Alelyani, and Huan Liu. Feature selection for classification: A review. *Data Classification: Algorithms and Applications*, pages 37–64, 1 2014. [*Citado en pág. 2.*]

[77] Md Taufeeq Uddin and Md Azher Uddiny. A guided random forest based feature selection approach for activity recognition. *2nd International Conference on Electrical Engineering and Information and Communication Technology, iCEEiCT 2015*, 10 2015. [*Citado en pág. 14.*]

[78] Scott E. Umbaugh, Yansheng Wei, and Mark Zuke. Feature extraction in image analysis: A program for facilitating data reduction in medical image classification. *IEEE Engineering in Medicine and Biology Magazine*, 16:62–73, 7 1997. [*Citado en pág. 5.*]

[79] Ryan J. Urbanowicz, Melissa Meeker, William La Cava, Randal S. Olson, and Jason H. Moore. Relief-based feature selection: Introduction and review. *Journal of biomedical informatics*, 85:189–203, 9 2018. [*Citado en pág. 6.*]

[80] Ryan J. Urbanowicz, Randal S. Olson, Peter Schmitt, Melissa Meeker, and Jason H. Moore. Benchmarking relief-based feature selection methods for bioinformatics data mining. *Journal of Biomedical Informatics*, 85:168–188, 9 2018. [*Citado en pág. 5.*]

[81] Michel Verleysen and Damien François. The curse of dimensionality in data mining and time series prediction. *Lecture Notes in Computer Science*, 3512:758–770, 2005. [*Citado en pág. 22.*]

[82] Dimitrios Ververidis and Constantine Kotropoulos. Sequential forward feature selection with low computational cost | ieee conference publication | ieee xplore. *2005 13th European Signal Processing Conference*, 2005. [*Citado en pág. 11.*]

[83] Sijian Wang, Bin Nan, Saharon Rosset, and Ji Zhu. Random lasso. *The annals of applied statistics*, 5:468–485, 3 2011. [*Citado en pág. 15.*]

[84] Svante Wold, Michael Sjöström, and Lennart Eriksson. Pls-regression: a basic tool of chemometrics. *Chemometrics and Intelligent Laboratory Systems*, 58:109–130, 10 2001. [*Citado en pág. 2.*]

[85] Shiyang Xuan, Guanjun Liu, Zhenchuan Li, Lutao Zheng, Shuo Wang, and Changjun Jiang. Random forest for credit card fraud detection. *ICNSC 2018 - 15th IEEE International Conference on Networking, Sensing and Control*, pages 1–6, 5 2018. [*Citado en pág. 14.*]

[86] Lei Yu and Huan Liu. Efficient feature selection via analysis of relevance and redundancy. *Journal of Machine Learning Research*, 5:1205–1224, 2004. [*Citado en pág. 2.*]

[87] Rizgar Zebari, Adnan Abdulazeez, Diyar Zeebaree, Dilovan Zebari, and Jwan Saeed. A comprehensive review of dimensionality reduction techniques for feature selection and feature extraction. *Journal of Applied Science and Technology Trends*, 1:56–70, 5 2020. [*Citado en pág. 22.*]

[88] Liang Zhao, Zhikui Chen, Yueming Hu, Geyong Min, and Zhaohua Jiang. Distributed feature selection for efficient economic big data analysis. *IEEE Transactions on Big Data*, 4:164–176, 8 2016. [*Citado en pág. 5.*]

[89] Zhi Hua Zhou. Ensemble methods: Foundations and algorithms. *Ensemble Methods: Foundations and Algorithms*, pages 1–218, 1 2012. [*Citado en pág. 2.*]

[90] Mu Zhu and Trevor J. Hastie. Feature extraction for nonparametric discriminant analysis. *Journal of Computational and Graphical Statistics*, 12:101–120, 3 2003. [*Citado en pág. 16.*]

[91] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 67:301–320, 4 2005. [*Citado en pág. 16.*]