Medidas de influencia de características en problemas de clasificación, utilizando estructuras de agrupación en teoría de juegos.

UNIVERSIDAD MIGUEL HERNÁNDEZ DE ELCHE

GRADO EN ESTADÍSTICA EMPRESARIAL

FACULTAD DE CIENCIAS SOCIALES Y JURÍDICAS DE ELCHE

TRABAJO DE FIN DE GRADO

Autor: Jordi Mas Moreno

Tutores: Juan Carlos Gonçalves Dosantos, Joaquín Sánchez Soriano

Curso académico 2024 – 2025

ÍNDICE

Resumen

Este Trabajo de Fin de Grado tiene como objetivo principal analizar en profundidad, la influencia que ejercen los distintos valores o características de las variables explicativas discretas en modelos de clasificación supervisada. Para llevar a cabo este estudio, se emplearán herramientas procedentes de la teoría de juegos cooperativos, un marco teórico que permite evaluar la contribución individual de cada jugador (cada valor posible dentro de cada variable) al resultado colectivo, es decir, al rendimiento del modelo.

En concreto, se utilizará el concepto de uniones a priori, un modelo que permite considerar agrupaciones entre los valores, donde cada agrupación está definida por una variable, para determinar de forma más precisa el impacto que tiene cada uno de ellos sobre la capacidad predictiva del modelo. Esta metodología permitirá no solo identificar qué variables son más relevantes, sino también cómo ciertas combinaciones de valores influyen en la mejora o deterioro del rendimiento del modelo de clasificación.

Abstract

The main objective of this Final Degree Project is to analyse in depth the influence exerted by the different values or characteristics of discrete explanatory variables in the context of supervised ranking models. To carry out this study, we will use tools from cooperative game theory, a theoretical framework that allows us to evaluate the individual contribution of each player, in this case, of each possible value within each variable, to the collective outcome, that is, to the performance of the model.

Specifically, we will use the concept of a priori unions, a model that allows us to consider groupings among the values, where each grouping is defined by a variable, in order to determine more precisely the impact that each of them has on the predictive capacity of the model. This methodology will allow not only to identify which variables are more relevant, but also how certain combinations of values influence the improvement or deterioration of the performance of the classification model.

1 Introducción

Vivimos en una época donde, mediante el análisis y estudio de grandes volúmenes de datos, es posible convertir toda esa información en conocimiento útil para comprender y predecir comportamientos, y así poder tomar decisiones estratégicas basadas en los resultados extraídos. Las técnicas de clasificación de datos nos ayudan a encontrar patrones que nos permitan agrupar individuos con características similares o incluso predecir el comportamiento futuro de nuevos registros. Sin embargo, al profundizar un poco más en este análisis, puede surgir la pregunta: ¿cuáles son las características que más influyen al realizar la predicción del comportamiento de un nuevo individuo? Resolver esta cuestión será de gran utilidad a la hora de construir un modelo más preciso, comprensible y eficiente.

Dependiendo de si se conocen o no las clases o grupos en los que se debe dividir el conjunto de datos, podemos encontrar dos tipos principales de clasificación: clasificación no supervisada y clasificación supervisada. Cuando las clases no se conocen de antemano, utilizaríamos la clasificación no supervisada y, en el caso contrario, optaríamos por la clasificación supervisada [1]. A lo largo de este proyecto, nos centraremos en la clasificación supervisada, donde existen varias técnicas para su aplicación, poniendo más énfasis en Support Vector Machine (SVM) y Random Forest, siendo dos de las técnicas más utilizadas debido a su efectividad y facilidad de implementación en distintas situaciones.

Para poder abordar la cuestión planteada en el primer párrafo sobre la importancia de las características, nos apoyaremos en la teoría de juegos, concretamente en los juegos cooperativos con uniones a priori. Esta teoría de juegos estudia la toma de decisiones en situaciones donde un grupo de dos o más jugadores interactúan entre sí y sus decisiones finales están marcadas por lo que otros jugadores deciden o por lo que esperan que otros jugadores hagan [2]. Su objetivo principal es encontrar patrones de comportamiento en contextos en los que los resultados dependen de las acciones de los jugadores [3]. En otras palabras, un juego es una situación en la que un grupo de jugadores debe decidir su estrategia para maximizar su beneficio, teniendo en cuenta que dependerá tanto de sus decisiones como de las de otros jugadores.

En el caso de los juegos cooperativos, también conocidos como juegos coalicionales, los sujetos pueden comunicarse y negociar entre ellos, por lo que pueden formar coaliciones para maximizar su beneficio [2]. Por el contrario, en los juegos no cooperativos los jugadores toman las decisiones de forma independiente. No obstante, conocen a los demás jugadores y las especificaciones del juego, por lo tanto, deberán tomar estas decisiones intentando predecir lo que harán los otros jugadores para así actuar en consecuencia, un ejemplo muy famoso de este tipo de juego es el dilema del prisionero.

La forma en la que vamos a utilizar estos juegos cooperativos con uniones a priori es tratando a cada característica como un jugador distinto y a cada variable como una unión de jugadores. Por ejemplo, los jugadores "soltero" y "casado" formarán parte de la unión a priori "estado civil". Estos jugadores y uniones nos ayudarán a crear conjuntos diferentes de datos, para así poder ir guardando la proporción de observaciones correctamente clasificadas y, en nuestro caso, poder aplicar después el valor de Owen y así extraer qué características influyen más a que un jugador esté bien clasificado.

2 Clasificación supervisada

Para resolver nuestro problema de clasificación, nos centraremos en la clasificación supervisada, siendo esta una de las tareas más comunes dentro de los denominados Sistemas Inteligentes [4]. Estos sistemas buscan identificar patrones en un conjunto de datos y asignar la clase correspondiente a cada elemento del conjunto. Este proceso se puede lograr aplicando una serie de algoritmos, los cuales permiten generalizar a nuevos casos teniendo en cuenta ejemplos ya clasificados. La clasificación, se puede llevar cabo aplicando tanto métodos estadísticos como técnicas de inteligencia artificial [4], cada uno con sus propias fortalezas y debilidades. Algunos ejemplos que podemos encontrar dentro de cada tipo son:

- Estadística: Regresión Logística, Análisis Discriminante, SVM, …
- Inteligencia Artificial: Redes Neuronales, Árboles de Decisión, K-Nearest Neighbors, Random Forest, …

Hemos mencionado que la clasificación supervisada es una de las tareas más utilizadas en los Sistemas Inteligentes porque esta clasificación se ha empleado en una alta variedad de casos como, por ejemplo, en el diagnóstico de enfermedades, la concesión de créditos bancarios, la clasificación de imágenes o la detección de fraudes, entre otros. Esta es una pequeña muestra de los muchos casos en los que se puede utilizar esta técnica. Siempre que encontremos datos históricos etiquetados, es muy recomendable usar este tipo de clasificación para hacer predicciones precisas sobre nuevos casos.

A continuación, vamos a introducir los términos: *conjunto de datos de entrenamiento* y *conjunto de datos test*. Como ya hemos comentado anteriormente, el aprendizaje supervisado se centra en resolver el problema de clasificación a partir de un conjunto de elementos, los cuales ya se conoce de antemano su clase. Este conjunto es el que se conoce comúnmente como *conjunto de datos de entrenamiento*. En cambio, el *conjunto de datos test* se utiliza para evaluar el rendimiento del modelo, a diferencia del de entrenamiento, que se utiliza para estimar los parámetros de este [4].

Para conocer el rendimiento del modelo nos ayudamos de la matriz de confusión, que consiste en una matriz que está formada por las columnas "Predicho" y las filas "Observado". Siguiendo con el ejemplo de la introducción, en la siguiente tabla podemos ver que las observaciones se pueden clasificar en soltero y casado, para poder hacerlo más generalizado, atribuiremos a las clases soltero y casado el valor "0" y "1" respectivamente:

	Predicho soltero Y = 0	Predicho casado Y = 1	Total Observado
Observado soltero Y = 0	$n_{0,0}$	$n_{0,1}$	n_0
Observado casado Y = 1	$n_{1,0}$	$n_{0,1}$	n_1
Total Predicho	$\hat{n}_0$	$\hat{n}_1$	n

Tabla 1: Representación de una Matriz de Confusión

Una vez tenemos creada la tabla podemos analizar varios indicadores:

- Precisión: porcentaje de clasificados correctamente.

$$\frac{n_{0,0} + n_{1,1}}{n} * 100$$

- Sensibilidad (ratio de positivos): porcentaje de casados (1) correctamente clasificados.

$$\frac{n_{1,1}}{n_1} * 100$$

- Especificidad (ratio de negativos): porcentaje de solteros (0) correctamente clasificados.

$$\frac{n_{0,0}}{n_0} * 100$$

- Tasa de falsos positivos: porcentaje de solteros clasificados como casados.

$$\frac{n_{0,1}}{n_0} * 100$$

- Tasa de falsos negativos: porcentaje de casados clasificados como solteros.

$$\frac{n_{1,0}}{n_1} * 100$$

Para garantizar que el modelo no se sobreajuste a los datos de entrenamiento y pueda generalizar correctamente a nuevas observaciones, es fundamental que cada registro de la base de datos esté presente en alguno de estos dos conjuntos. No obstante, lo ideal sería que el conjunto de test sea independiente del conjunto de entrenamiento [4]. Así nos aseguramos de que el rendimiento del modelo no esté sesgado por las observaciones que han sido utilizadas para entrenar, ya que, si utilizáramos las mismas observaciones en ambos conjuntos, el modelo podría no generalizar bien a nuevos casos.

Para obtener cada uno de los conjuntos, se pueden emplear técnicas de muestreo [4]. Por ejemplo, el muestreo aleatorio simple, donde las observaciones se eligen de forma aleatoria, o el muestreo estratificado, donde se asegura que las clases estén repartidas proporcionalmente en ambos subconjuntos. Este último es más recomendable en los casos donde hay una clase que es mucho más frecuente que las demás, ya que con este muestreo conseguimos que cada clase tenga una buena representación en los subconjuntos de entrenamiento y de test.

En resumen, la clasificación supervisada es de las mejores herramientas que podemos utilizar en el ámbito de los Sistemas Inteligentes, especialmente cuando existen grandes cantidades de datos etiquetados. Sin embargo, su rendimiento depende en gran medida de la calidad de los datos, la correcta división de los subconjuntos y la elección adecuada del algoritmo de clasificación, lo cual son aspectos que deben considerarse seriamente al desarrollar cualquier modelo predictivo.

2.1 Support Vector Machine

El primer algoritmo que vamos a tratar es *Support Vector Machine* (SVM), uno de los más reconocidos dentro del aprendizaje supervisado, ya que se aplica en una gran variedad de problemas de regresión y clasificación, tales como el procesamiento de señales médicas, el análisis del lenguaje natural y el reconocimiento tanto de imágenes como de voz, entre otros [5].

Para llevar a cabo SVM, necesitaremos un conjunto de individuos, al que denominaremos como Ω. Además, será necesario identificar la variable objetivo categórica (y), que estará compuesta por dos categorías: $\{-1, +1\}$ y el conjunto de variables explicativas $x_{i1}, x_{i2}, \dots, x_{ik}$. Por lo tanto, para cada individuo $i \in \Omega$ tendremos:

- $x_i = (x_{i1}, x_{i2}, \dots, x_{ik})$: un vector de variables explicativas.
- $y_i \in \{-1, +1\}$: las clases en las que se puede dividir el conjunto de datos.

Este algoritmo sirve para predecir la clase de nuevas observaciones futuras, siempre y cuando se conozcan las variables explicativas:

$$x_{new} \rightarrow \hat{y}_{new} \in \{-1, +1\}$$

Para lograr esta predicción, SVM tiene como objetivo separar dos clases distintas de observaciones de la mejor manera posible, identificando un hiperplano que maximice el margen entre los puntos más cercanos de cada clase a dicho hiperplano [5]. La expresión que lo define es la siguiente: [6]

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_K X_K = 0$$

Las betas $\beta_1, \ldots, \beta_K$ son los coeficientes que determinan la influencia de cada variable X_i en la separación de las clases. Y la beta β_0 determina el desplazamiento del hiperplano respecto al origen.

En el caso de que tengamos una observación $x_i = (x_{i1}, x_{i2}, \ldots, x_{ik})$ que haga que la expresión anterior sea igual a 0, significa que esa observación forma parte del hiperplano, pero si hace que sea distinta de 0, significa que la observación se encuentra en uno de los dos lados del hiperplano.

La posición del hiperplano se define mediante una serie de puntos u observaciones, conocidos como *vectores de soporte* [5] como podemos ver en la *Figura 1*.

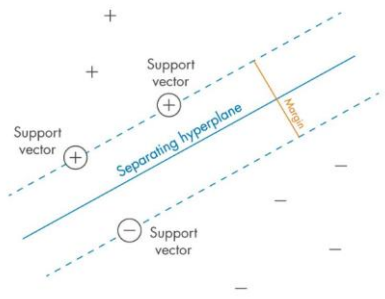

Figura 1: Elementos destacables en SVM [5]

Al hablar de margen, nos referimos a la distancia máxima entre el hiperplano y los puntos de datos más cercanos de cada clase. Por lo tanto, cuanto mayor sea el margen, mejor será la capacidad de generalización del modelo.

Inicialmente, SVM solo puede identificar este hiperplano en problemas donde las clases son linealmente separables; sin embargo, en la mayoría de los casos el algoritmo admite un número reducido de clasificaciones erróneas [5]. Lo que conocemos como margen rígido *(figura 3)* y margen blando *(figura 4)*, respectivamente.

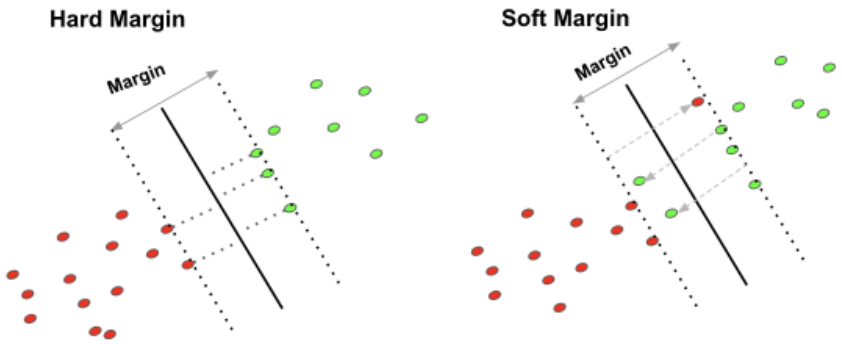

Figura 2: Margen Rígido (izquierda) y Margen Blando (derecha) [7]

Como podemos observar en las imágenes anteriores, la diferencia que encontramos en el margen rígido y blando *(figuras 2)*, es que el rígido requiere que los datos del conjunto de entrenamiento sean linealmente separables, mientras que el blando construye un hiperplano que no separa perfectamente las dos clases, lo que permite que algunas observaciones estén dentro del margen o en el lado incorrecto del hiperplano.

Para el margen rígido, consideraríamos el siguiente problema de optimización para calcular el hiperplano:

$$\min_{\hat{\beta}_0,\hat{\beta}_1,\ldots,\hat{\beta}_K} \frac{1}{2}\sum_{j=1}^{k}\hat{\beta}_j^2$$

$$s.t. \quad y_i\left(\hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \cdots + \hat{\beta}_k x_{ik}\right) \geq 1, \quad i = 1, \ldots, n$$

$$\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_k \in \mathbb{R}$$

Los coeficientes $\hat{\beta}_j$ corresponden a los pesos de cada variable explicativa, es decir, cada $\hat{\beta}_j$ determina la importancia de cada variable x_j en la clasificación. Una vez explicado esto, podemos decir que la función objetivo se centra en minimizar el tamaño de los coeficientes, lo que es equivale a decir que minimiza la suma de los cuadrados de los $\hat{\beta}_j$.

La primera restricción asegura que cada observación esté bien clasificada. Lo que significa que:

- Si $y_i = +1$ (pertenece a la clase +1), la expresión $\hat{\beta}_0 + \hat{\beta}_1 X_1 + \cdots + \hat{\beta}_K X_K$ debe ser mayor o igual a 1. Esto significa que todos los puntos que representan la clase +1 deben estar a una distancia de al menos 1 del hiperplano.

- Si $y_i = -1$ (pertenece a la clase -1), en este caso la expresión $\hat{\beta}_0 + \hat{\beta}_1 X_1 + \cdots + \hat{\beta}_K X_K$ debería ser negativa y, además, también significa que los puntos de la clase -1 deben estar a una separación de al menos 1 del hiperplano.

Por otro lado, para calcular el hiperplano en el margen blando, el modelo de optimización sería diferente:

$$\min_{\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_K} \frac{1}{2}\sum_{j=1}^{k} \hat{\beta}_j^2 - \frac{C}{n}\sum_{i=1}^{n} \xi_i$$

$$s.t. \quad y_i\left(\hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \cdots + \hat{\beta}_k x_{ik}\right) \geq 1 - \xi_i, \quad i = 1, \ldots, n$$

$$\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_k \in \mathbb{R}$$

$$\xi_i \geq 0, \quad i = 1, \ldots, n$$

En la función objetivo encontramos la misma expresión que en el modelo del margen rígido, con la diferencia de que se le añade la segunda expresión $\frac{C}{n}\sum_{i=1}^{n} \xi_i$, la cual corresponde al nivel de error permitido.

Las variables de holgura ξ_i permiten que ciertas observaciones no respeten el margen, por lo que habrá individuos que se encuentren en el interior de este o incluso llegando a estar en el lado incorrecto del hiperplano. Es decir, representan el error de clasificación. Cuando $\xi_i = 0$, significa que el individuo i está bien clasificado. Si estuviera entre 0 y 1 significaría que la observación i se encuentra en el lado correcto del hiperplano, pero dentro del margen. En cambio, si fuera > 1, el individuo i estaría mal clasificado. En resumen, cuanto mayor es el valor de ξ_i más violación del margen o de la clasificación hay, sin embargo, solo encontraríamos un error de clasificación cuando ξ_i supere 1. [6]

El valor C es una constante positiva que determina el grado de tolerancia del proceso respecto a las observaciones que se encuentran dentro del margen o mal clasificadas, por lo que podríamos decir que controla el equilibrio entre la maximización del margen y la minimización de errores. [6]

- Al disminuir C el margen aumenta, por lo tanto, encontraremos más vectores de soporte, esto hace que mejore generalización, aunque cometiendo más errores.
- Al aumentar C el margen disminuye, lo que hace que sea menos tolerante a las violaciones del margen. No obstante, un valor alto de C puede llevar al sobreajuste ya que se intenta clasificar correctamente todos los puntos.

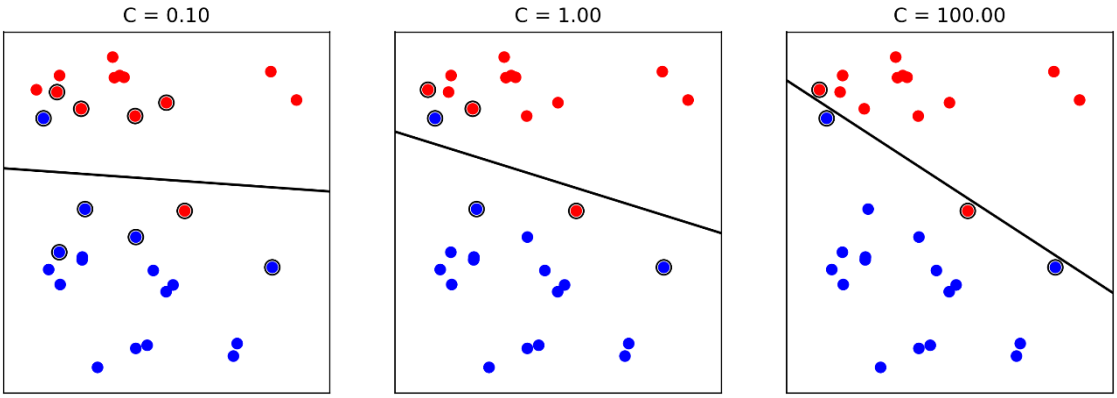

Figura 3: Diferencia al aplicar diferentes valores de C [8]

En cuanto a la primera restricción, es similar a la que encontramos en el margen rígido, con la diferencia de que, en lugar de exigir que cada punto esté en el lado correcto del margen, se permite que haya algunas observaciones que estén dentro de él o incluso en el lado incorrecto del hiperplano.

SVM pertenece a una clase de algoritmos de Machine Learning conocidos como métodos kernel. Estos métodos utilizan funciones que transportan los datos a un espacio dimensional superior, lo que permite separar las clases de una forma más simple. Al transformar los datos, los límites de decisión o hiperplanos no lineales pueden convertirse en límites lineales en este espacio ampliado [5].

Entre los kernels más utilizados destacan:

- Kernel lineal: adecuado cuando los datos ya son separables de por sí sin transformaciones adicionales. *(Figura 4)*
- Kernel polinomial: útil cuando existen relaciones más complejas entre los datos, por tanto, necesita polinomios de distintos grados para modelarlos. *(Figura 5)*
- Kernel RBF (Radial Basis Function): ideal para problemas donde los límites entre las clases no son lineales. *(Figura 6)*

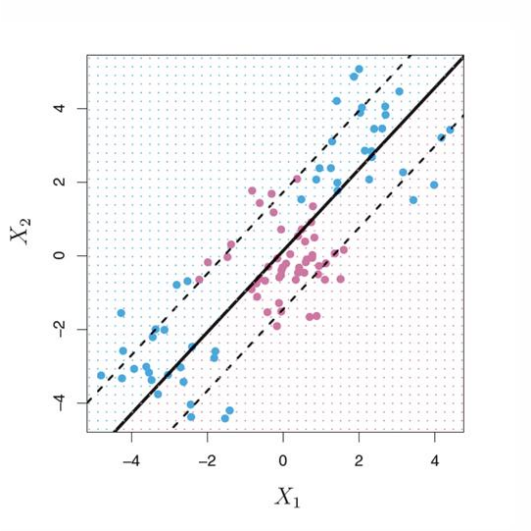

Figura 4: Aplicación Kernel Lineal [6]

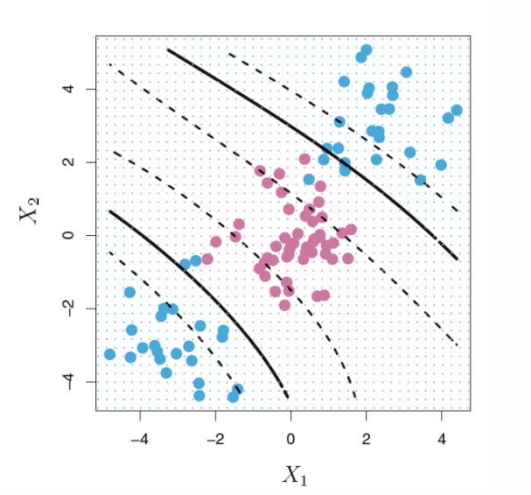

Figura 5: Aplicación Kernel Polinómica [6]

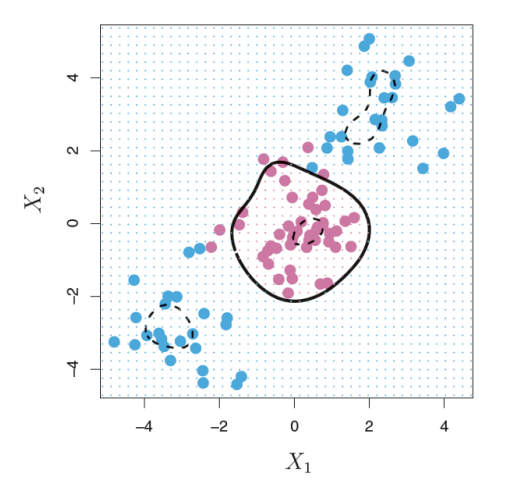

Figura 6: Aplicación Kernel Radial [6]

Algunas de las ventajas y desventajas que podríamos encontrar al utilizar este algoritmo podrían ser [9]:

- Es efectivo en espacios de alta dimensionalidad.
- Es eficiente en la gestión de memoria.
- La eficacia del modelo está relacionada con la elección del kernel.
- Es poco eficiente con grandes datasets.
- En el caso de que el número de características sea mucho mayor que el de las muestras es muy importante escoger el kernel apropiado.
- No proporciona estimaciones de probabilidad de manera directa (requiere métodos adicionales).
- El hiperplano de separación depende de las observaciones más próximas, aunque estas sean erróneas.
- Es necesario escalar los datos adecuadamente.

En conclusión, SVM es un modelo bastante fiable dentro del aprendizaje supervisado. Como en la mayoría de los algoritmos de clasificación supervisada su rendimiento depende de la correcta aplicación del algoritmo, lo que en este caso conlleva una correcta elección del kernel y la preparación adecuada de los datos.

2.2 Árboles de Clasificación y Random Forest

Árboles de clasificación

Los árboles de clasificación son un algoritmo de aprendizaje automático que organiza los datos de una forma similar a un árbol genealógico, es decir, en una estructura jerárquica compuesta por nodos y ramas. Su objetivo principal es dividir el conjunto de datos en diferentes clases de la manera más homogénea posible.

Este algoritmo facilita la predicción de la clase o grupo al que pertenece un nuevo elemento, basándose en un conjunto de variables predictoras [10]. Cuando se tiene una variable objetivo categórica, los árboles de clasificación generan una serie de reglas de división, extraídas de las variables explicativas, que permiten ir separando el conjunto de datos original hasta llegar a tener pequeños subconjuntos clasificados según la variable categórica. Sin profundizar demasiado, estas reglas de división se basan en una cota o una clase, dependiendo de la naturaleza de la variable explicativa. Más adelante veremos esta idea con más detalle.

Al igual que en SVM, necesitamos un conjunto de individuos Ω y una variable categórica (y) con categorías: $\{1,2,\dots,l\}$. Una vez tenemos estos elementos, el conjunto de posibles valores para las variables explicativas $X_1, X_2, \dots, X_K$, se divide en J regiones $R_1, R_2, \dots, R_J$

llamadas nodos hoja, los cuales explicaremos a continuación. A cada una de estas regiones $j \in \{1,\dots,J\}$ se le asigna una clase $\ell \in \{1,2,\dots,l\}$. Es decir, a R_j se le asignara una clase ℓ. Por lo tanto, una nueva observación x_{new} será clasificada del siguiente modo:

$$si\, x_{new} \in R_j, entonces\, \hat{y}_{new} = \ell$$

Los árboles de clasificación están compuestos por:

- Los nodos, que pueden ser de tres tipos:
  - Nodo raíz: representa todo el conjunto de datos, es decir, el conjunto de datos inicial.
  - Nodos hoja: representan las regiones finales del árbol, en las que se asigna una clase a las observaciones.
  - Nodos intermedios: son las regiones previas a los nodos hoja que se van formando.
- Las ramas, que conectan los nodos, es donde implementamos las reglas de división, que como ya hemos comentado, dependen de la naturaleza de las variables:
  - Para **variables cuantitativas**, la separación suele realizarse con condiciones de la forma:

    $$X \leq cota \;\&\; X > cota$$

    donde $cota$ representa un umbral específico.

  - Para **variables categóricas**, se utilizan condiciones como:

    $$X = l \;\&\; X \neq l$$

    donde l es una de las categorías o clases.

Este algoritmo busca reducir constantemente la impureza de cada nodo, con el objetivo final de conseguir que todas las regiones generadas sean lo más homogéneas posible. Consideramos un nodo completamente puro si todas sus observaciones pertenecen a la misma clase. Para cuantificar la pureza de un nodo, se utilizan métricas como [6]:

- Índice de Gini:

$$G = \sum_{l=1}^{L} p_l(1 - p_l)$$

- Entropía cruzada:

$$Entriopía = -\sum_{l=1}^{L} p_l \log(p_l)$$

En las dos expresiones, un valor bajo indicaría que un nodo es más homogéneo. Siendo p_l la proporción de observaciones que pertenecen a la clase l en las dos fórmulas. [6]

Si se generara un árbol demasiado profundo, podríamos tener nodos hoja con un número muy reducido de observaciones, por lo que correríamos el riesgo de sobreajustar la muestra de entrenamiento, lo que haría que se perdiera capacidad de generalización. Una solución a este problema sería establecer ciertos criterios de parada. Uno de los más comunes es establecer un parámetro que ayude a tener un número mínimo de individuos por nodo [11].

En el momento que tenemos formado el árbol de decisión, podemos comenzar a predecir la clase de nuevas observaciones. Como hemos indicado anteriormente, si $x_{new} \in R_j$, $entonces\ \hat{y}_{new} = l$. Cada una de estas regiones R_j contiene un pequeño subconjunto del conjunto de entrenamiento original (x_i, y_i), $i = 1, \dots, n$, por ejemplo, n_j puntos, los cuales serán clasificados según el tipo que sea más común entre todos estos individuos.

Entonces, para cada clase $m \in \{1, \dots, l\}$, podríamos estimar la probabilidad de que se dé la clase m dado que el vector está en la región R_j, esto es, $P(Y) = (m|X \in R_j)$, como:

$$\hat{p}_m(R_j) = \frac{1}{n_j} \sum_{x_i \in R_j} l(y_i = m),$$

siendo $\hat{p}_m$ la proporción de puntos en la región R_j que pertenecen a la clase m.

La clase asignada se puede expresar como: $\ell = arg \max_{m=1,\dots,l} \hat{p}_m(R_j)$

En la siguiente figura podremos observar un ejemplo de árbol de decisión:

Cliente	Edad	Género	Ingresos	¿Compra?
1	25	Hombre	Bajo	No
2	30	Mujer	Medio	Sí
3	40	Hombre	Bajo	No
4	50	Mujer	Alto	Sí
5	22	Mujer	Bajo	No
6	35	Hombre	Alto	No
7	28	Mujer	Medio	Sí
8	60	Hombre	Bajo	No
9	45	Mujer	Medio	Sí
10	19	Hombre	Bajo	No
11	32	Mujer	Alto	Sí
12	47	Hombre	Medio	Sí
13	59	Hombre	Alto	No
14	43	Mujer	Bajo	No
15	20	Hombre	Medio	Sí

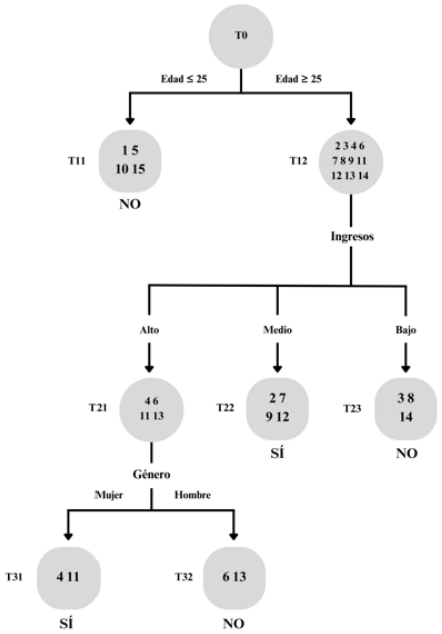

Figura 7: Ejemplo de Árbol de decisión

En este árbol podemos observar que en la cima se encuentra el nodo raíz. A medida que se aplican condiciones, las observaciones se van distribuyendo en los nodos intermedios hasta llegar a los nodos hoja. En el caso de las variables cuantitativas (numéricas) y cualitativas (categóricas), la partición se realizará de acuerdo lo indicado anteriormente.

Para las variables numéricas, las divisiones se realizan utilizando cotas; en este caso, el valor 25. Mientras que, para las variables categóricas, los datos se separan según si pertenecen o no a una categoría específica, en este caso “alto”, “medio” o “bajo”, o dependiendo del género, sean “mujer” o “hombre”.

Una vez construido el árbol final a partir del conjunto de entrenamiento, a cada nodo hoja se le asignará la clase que predomina entre las observaciones que componen ese nodo, como sucede en el ejemplo de la *figura 7* en el nodo *T11*.

Algunas de las ventajas y desventajas que encontramos con los árboles de clasificación son: [6]

- Son fáciles de interpretar, incluso por personas sin conocimientos técnicos.
- Puede manejar tanto datos categóricos como numéricos.
- Permite abordar problemas donde existen más de dos clases en el conjunto de datos
- Permiten trabajar con datos desequilibrados, es decir, encontramos una clase mucho más frecuente que otras.

- También se pude llegar a utilizar para problemas de regresión y predecir valores numéricos en vez de categóricos.
- No es necesarios que los datos sigan una distribución especifica.
- Los árboles de decisión no son tan precisos en comparación a otros modelos.
- La estructura del árbol final puede cambiar si ocurren pequeñas variaciones en los datos.
- Son propensos al sobreajuste si no se establecen criterios adecuados.

Random Forest

Debido a las desventajas que hemos comentado, se desarrolló el algoritmo Random Forest (Bosques Aleatorios), el cual consiste en combinar múltiples árboles de decisión para así, obtener un modelo que pueda solucionar la mayoría de los problemas que acarreaba utilizar un solo árbol. Este modelo pertenece a la familia de algoritmos (*ensemble learning*), los cuales se basan en la aplicación de varios métodos de predicción individuales para obtener una mayor precisión en el proceso [12]. Por esta razón Random Forest es de los métodos más utilizados en big data, ya que es uno de los algoritmos más eficientes dentro de la clasificación supervisada.

Random Forest se encarga de generar múltiples árboles de decisión a partir de diferentes subconjuntos extraídos del conjunto de entrenamiento original, los cuales se obtienen mediante una técnica llamada bootstrap (muestreo con reemplazo). Este mecanismo hace que aumente la diversidad a la vez que se reduce la varianza del modelo.

Cuando construimos los árboles, lo común es que cada uno solo utilice un conjunto de variables, que normalmente es la raíz cuadrada del conjunto total de variables ($\sqrt{k}$) [6]. En el caso de que se llegaran a utilizar todas las variables disponibles, el Bosque Aleatorio se conocería como Bagging (Bootstrap Aggregating).

Una vez obtenida la colección de árboles, la predicción final para un nuevo individuo se podrá obtener mediante:

- Clasificación: asignándose la clase más frecuente entre las predicciones de los árboles individuales, lo que se conoce como votación por mayoría.
- Regresión: calculando el promedio de las predicciones de los árboles finales.

Al utilizar este algoritmo podemos encontrar estas ventajas y desventajas: [12] [13]

- Al combinar árboles de clasificación, obtenemos una mayor precisión, reducimos el riesgo de sobreajuste y además mejoramos la capacidad predictiva.
- A diferencia de los árboles individuales es menos sensible a pequeñas variaciones en los datos.
- Puede manejar grandes volúmenes de datos de manera eficiente.

- No existen problemas con los valores ausentes, ya que se pueden estimar a partir de la media del conjunto de árboles.
- Nos proporciona información sobre la importancia de cada variable a la hora de predecir.
- Debido a la combinación de múltiples árboles, es menos interpretable que un solo árbol de decisión.
- Tiene mayor coste computacional que un árbol individual porque entrenar y almacenar todos los árboles requiere muchos más recursos.
- Al generar múltiples árboles hace que la predicción sea más lenta en comparación a otros modelos.

En resumen, Random Forest representa una mejora importante respecto a los árboles de decisión únicos, ya que reduce el sobreajuste y aumenta la precisión. Además, es un modelo más robusto y confiable, con mucha más capacidad de generalización.

3 Juegos Cooperativos

Los juegos cooperativos consisten en un conjunto de jugadores que pueden comunicarse entre ellos, negociar y llegar a acuerdos. Dentro de este tipo de juegos podemos encontrar dos tipos diferentes, los TU y los NTU. En los TU, que son en los que nos vamos a centrar, la utilidad se puede repartir de cualquier forma entre los jugadores, en cambio, si tuviéramos restricciones que hicieran que no se pudiera repartir, en cualquier caso, se conocerían como NTU. Cuando hablamos de utilidad, nos referimos al valor que un jugador obtiene o pierde tras la negociación dentro del juego.

Por lo tanto, tenemos que un juego TU es un $par\ (N, v)$ siendo $N = \{1, \dots, n\}$ el conjunto finito de jugadores y v una función característica tal que

$$v = 2^N \longrightarrow \mathbb{R}$$

Siendo $v(\emptyset) = 0$ [14]. Nótese que v corresponde al pago o beneficio que consigue cada coalición de jugadores.

Todos los juegos con utilidad transferible se denotan como G, por lo que llamaremos G^n a la clase de todos los juegos con n jugadores y conoceremos como S a la coalición en sí, siendo $S \subset N$ y el tamaño de coaliciones S se define como $|S|$. [14]

Por ejemplo, sea (N, v) donde N = {1,2,3}, entonces v se puede expresar como un vector fila: [15]

$$v = [v(1), v(2), v(3), v(12), v(13), v(23), v(123)]$$

Cuando un juego $(N,v) \in G^n$, (N,v) puede ser:

- Superaditivo [16]: la utilidad de la unión de dos conjuntos es mayor o igual que la suma de las utilidades individuales de cada conjunto.

$$v(S \cup T) \geq v(S) + v(T) \quad \forall\, S,T \in 2^N, S \cap T \neq \emptyset$$

- Subaditivo [17]: la utilidad de la unión de dos conjuntos es menor o igual que la suma de las utilidades individuales de cada conjunto.

$$v(S \cup T) \leq v(S) + v(T) \quad \forall\, S,T \in 2^N, S \cap T \neq \emptyset$$

- Aditivo [15]: la utilidad de la unión de dos conjuntos es exactamente igual que la suma de las utilidades individuales de cada conjunto.

$$v(S \cup T) = v(S) + v(T) \quad \forall\, S,T \in 2^N, S \cap T \neq \emptyset$$

Dependiendo del valor de las utilidades un juego también podrá ser 0 – normalizado, si $v(i) = 0\ \forall\, i \in N$, es decir, que todas las utilidades individuales tengan valor 0. Pero, en el caso de que todas las utilidades individuales sean 0 y la utilidad total sea 1, es decir, que la coalición de todos los jugadores sea igual a 1 $(v(N) = 1)$, lo llamaremos 0 – 1 normalizado. [18]

Por otro lado, un juego podrá ser considerado monótono cuando al añadir un jugador cualquiera a la coalición S, su valor siempre aumenta, es decir $v(S) \leq v(T)$ cuando $S \subset T$ [16]. Y 0 – monótono cuando $v(AB) + v(C) \leq v(ABC)$, es decir, que la coalición sea mayor que la suma de las coaliciones individuales [17].

Por último, un juego puede ser:

- Convexo [16]: si la suma de la utilidad de dos coaliciones no supera la suma de su unión y su intersección.

$$v(S) + v(T) \leq v(S \cup T) + v(S \cap T) \quad \forall\, S,T \in 2^N$$

- Estrictamente Convexo: si la suma de la utilidad de dos coaliciones es menor o igual a la suma de su unión y su intersección.

$$v(S) + v(T) < v(S \cup T) + v(S \cap T) \quad \forall\, S,T \in 2^N$$

- Cóncavo [17]: si la suma de la utilidad de dos coaliciones supera o iguala la suma de su unión y su intersección.

$$v(S) + v(T) \geq v(S \cup T) + v(S \cap T) \quad \forall S,T \in 2^N$$

- Estrictamente Cóncavo: si la suma de la utilidad de dos coaliciones supera la suma de su unión y su intersección.

$$v(S) + v(T) > v(S \cup T) + v(S \cap T) \quad \forall S,T \in 2^N$$

A continuación, vamos a introducir un ejemplo que iremos utilizando a medida que desarrollemos los siguientes apartados:

Ejemplo 1: *Sea un juego* (N, v) *con N = {A, B, C} y función característica*:

$$v(A) = v(B) = v(C) = 0$$

$$v(AB) = 4 \quad v(AC) = 3 \quad v(BC) = 6$$

$$v(ABC) = 8$$

Reparto y valor de Shapley

El objetivo de los juegos TU consiste en que se forme la coalición entre todos los jugadores del conjunto N, para así poder repartir la ganancia entre todos ellos. Este reparto no es más que un vector $x = (x_1, \dots, x_n) \in \mathbb{R}^n$, donde cada x_i representa la cantidad asignada a cada jugador i.

En esta sección nos vamos a centrar en el Valor de Shapley el cual, se calcula promediando los vectores de contribuciones marginales asociados a todos los posibles ordenes de los jugadores [19].

Este valor extrae la solución del reparto imponiendo ciertas condiciones: [2]

- Principio de eficiencia: este principio consiste en que toda la ganancia debe ser repartida entre los jugadores de la coalición.

$$\sum_{i \in N} \varphi_i(N, v) = v(N) \quad \forall (N, v) \in G^n$$

- Principio de jugador nulo: indica que si un jugador no aporta nada a ninguna coalición no debe recibir ninguna ganancia.

$$\varphi_i(N,v) = 0 \qquad \forall\ (N,v) \in G^n$$

- Principio de simetría: significa que, si un jugador aporta lo mismo que otro a toda coalición de jugadores, es decir, son intercambiables, deben recibir el mismo valor.

$$\varphi_i(N,v) = \varphi_j(N,v) \qquad \forall\ (N,v) \in G^n$$

- Principio de aditividad: este principio hace que, si se dividiera el juego original en dos juegos más pequeños, la suma de la ganancia repartida debería ser la misma que en el juego original.

$$\varphi(N,v+w) = \varphi(N,v) + \varphi(N,w) \qquad \forall\ (N,v),(N,w) \in G^n$$

Teóricamente el único valor que existe en G^n que satisfaga los anteriores principios es el valor de Shapley [15] el cual, se puede obtener con la siguiente formula:

$$Sh_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!\,(n-|S|-1)!}{n!}\,(v(S \cup \{i\}) - v(S)) \qquad [20]$$

La expresión se puede interpretar como el valor esperado del jugador i en el juego (N,v). En la formula, $|S|$ es el tamaño de la coalición S y n es el tamaño del conjunto total de jugadores N. Esta expresión calcula el promedio de todas las contribuciones marginales del jugador i a cualquier coalición de S jugadores. Estas contribuciones son el cambio de valor al incluir al jugador i en las diferentes coaliciones.

Volviendo a nuestro ejemplo de nuevo, tenemos que el valor de Shapley es:

$$Sh_A = \frac{1!\,(3-1-1)!}{3!}(4-0) + \frac{1!\,(3-1-1)!}{3!}(3-0) + \frac{2!\,(3-2-1)!}{3!}(8-6) = \frac{11}{6} = 1.83$$

$$Sh_B = \frac{1!\,(3-1-1)!}{3!}(4-0) + \frac{1!\,(3-1-1)!}{3!}(6-0) + \frac{2!\,(3-2-1)!}{3!}(8-3) = \frac{20}{6} = 3.33$$

$$Sh_C = \frac{1!\,(3-1-1)!}{3!}(3-0) + \frac{1!\,(3-1-1)!}{3!}(6-0) + \frac{2!\,(3-2-1)!}{3!}(8-4) = \frac{17}{6} = 2.83$$

Uniones a priori: el valor coalicional

Hay ocasiones donde podemos encontrar que ciertos jugadores tienen afinidades lo que hace que haya más probabilidad de que ocurran coaliciones entre ellos. Esto hace que varie el juego y que desde el principio se tenga en cuenta que existen jugadores que tienen ciertas relaciones, por lo tanto, estos jugadores aparecerán agrupados en un sistema de uniones "a priori".

Debido a esta circunstancia, Owen introdujo en 1977 el conocido *valor coalicional.* Este valor proporciona una forma justa de repartir el valor entre los jugadores, considerando que pueden existir acuerdos "a priori" que pueden influir en la negociación y en el reparto del valor. [21]

Para calcular este valor vamos a considerar el juego cociente, el juego donde las uniones son los jugadores. En esta situación, utilizaremos el valor de Shapley para decidir la cantidad que recibe cada unión. Ahora repartiremos utilizando de nuevo Shapley dentro de cada unión para conocer la cantidad asignada a cada jugador.

En resumen, Owen crea un método que modifica el valor de Shapley para que se adapte a juegos donde hay uniones ya predefinidas.

Si tenemos un juego $(N, v) \in G^n$, con particiones de N a las que llamaremos $P = \{P_1, \dots, P_m\}$, pasa a denotarse como (N, v, P) que se refiere a un juego con uniones "a priori" [21]. Este conjunto de juegos con sistema de coaliciones "a priori" de jugadores n los conoceremos como G_P^n. [15]

El valor coalicional o valor de Owen en un juego $(N, v, P) \in G_P^n$ se define como

$$\psi_i(N, v, P) = \sum_{S \subset M : j \notin S} \sum_{K \subset P_j : i \notin K} \frac{k!\,(p_j - k - 1)!\,s!\,(m - s - 1)!}{p_j!\,m!} (v(Q \cup K \cup \{i\}) - v(Q \cup K))$$

siendo $i \in N$, j el unico índice para el que $i \in P_j$ y $Q = \bigcup_{K \in S} P_k$. [15]

Al igual que para juegos TU, podemos definir las correspondientes propiedades de eficiencia, simetría, jugador nulo y aditividad de la siguiente forma [21]:

- Eficiencia:

$$\sum_{i \in N} \varphi_i(N, v, P) = v(N)$$

- Simetría en cada unión: esta propiedad consiste en que si dos jugadores son intercambiables dentro de una unión entonces:

$$\varphi_i(N, v, P) = \varphi_j(N, v, P)$$

- Simetría en el cociente: en este caso, esta propiedad consiste en que si dos uniones son intercambiables dentro de una coalición entonces:

$$\sum_{i \in P_k} \varphi_i(N, v, P) = \sum_{j \in P_l} \varphi_j(N, v, P)$$

- Jugador nulo: Si un jugador no aporta ningún valor al juego entonces:

$$\varphi_i(N, v, P) = 0$$

- Aditividad: $\forall\ (N, v, P), (N, w, P) \in G_p^n$

$$\varphi(N, v + w, P) = \varphi(N, v, P) + \varphi(N, w, P)$$

También en estos casos, el único valor φ en G_P^n que satisfaga estas condiciones es el valor de Owen. [15]

A continuación, vamos a adaptar el *ejemplo 1* desarrollado en apartados anteriores para calcular el valor de Owen:

En este caso tenemos el juego (N, v, P) *siendo* $P = \{\{AB\}, \{C\}\}$.

$$\psi_A = \frac{1!\,0!\,0!\,1!}{2!\,2!}(4-0) + \frac{0!\,1!\,1!\,0!}{2!\,2!}(3-0) + \frac{1!\,0!\,1!\,0!}{2!\,2!}(8-6) = \frac{9}{4} = 2.25$$

$$\psi_B = \frac{1!\,0!\,0!\,1!}{2!\,2!}(4-0) + \frac{0!\,1!\,1!\,0!}{2!\,2!}(6-0) + \frac{1!\,0!\,1!\,0!}{2!\,2!}(8-3) = \frac{15}{4} = 3.75$$

$$\psi_c = \frac{0!\,0!\,1!\,1!}{1!\,2!}(8-4) = \frac{4}{2} = 2$$

4 Medida de influencia de características

A continuación, vamos a explicar cómo mediante la aplicación de la teoría de juegos, podemos extraer un ranking de la influencia de cada característica en un problema de clasificación.

Para empezar, supongamos que tenemos un problema de clasificación supervisada con una variable objetivo Y y con variables $X_i = (X_1, X_2, \dots, X_M)$ de las cuales, nos vamos a centrar en las variables discretas. Dentro de cada variable extraeremos cada categoría o característica, a la cuál denotaremos como Z_{ij} donde $j = \{1,2 \dots, N_i\}$, siendo N_i el número de características de la variable X_i. En este caso, Z_{ij} representa la $j - ésima$ categoría de la variable X_i.

Una vez tenemos localizadas cada Z_{ij}, cada una de estas corresponderán a un jugador diferente. Por lo tanto, definimos el conjunto de jugadores como:

$$N = \{Z_{ij} \mid i = 1, \dots, M; j = 1, \dots, N_i\}$$

A partir de aquí, definimos nuestro juego con uniones a priori (N, v, P) donde:

- N es el conjunto de jugadores indicado en el párrafo anterior.
- v es la función característica (definida más adelante).
- $P = \{P_1, P_2, \dots, P_M\}$, es la distribución de los jugadores N en uniones a priori, donde cada subconjunto P_i agrupa todas las Z_{ij} pertenecientes a una misma variable X_i. Es decir:

$$P_i = \{Z_{ij} \mid j = 1, \dots, N_i\}, \quad \forall\, i = 1, \dots, M$$

Para poder resolver este estudio mediante Rstudio, tendremos que modificar el conjunto de datos original adaptándolo a cada coalición posible, es decir, crear un problema de clasificación diferente para cada coalición. Cada conjunto se obtendrá aplicando un "filtro", así se seleccionarán todos los individuos que cumplan con el jugador individual o con la coalición $S \subseteq N$.

Una vez tenemos el conjunto modificado, el siguiente paso será realizar SVM o Random Forest, para obtener así la matriz de confusión. Esta matriz nos ayudará a obtener el valor $v(S)$ para cada coalición S, como la suma de los elementos de la diagonal de dicha matriz, que corresponden a los sujetos bien clasificados:

$$v(S) = \sum_{k=1}^{C} CM_{kk}$$

donde C es el número de clases y CM es la matriz de confusión para la coalición S.

Estos valores $v(S)$ son los que utilizaremos para calcular el valor de Owen ψ_i de cada jugador Z_{ij} que finalmente nos permitirá obtener el ranking de influencias.

5 Ejemplos con datasets

Antes de mostrar los ejemplos vamos a destacar el ranking de influencia será obtenido utilizando dos versiones diferentes de código en Rstudio. En estas dos versiones, tendremos que indicar las variables, los jugadores, las uniones entre ellos y, por último, un parámetro T que indique el número de veces que se va a simular el proceso. La diferencia entre estas dos versiones radica en la parte del proceso en la que se encuentra el bucle de las simulaciones.

Para poder entender cómo funcionan los dos códigos con facilidad, debemos tener en la cabeza el proceso que se lleva a cabo para obtener el valor de Owen, explicado anteriormente. Una vez tenemos claro cómo se extrae el valor en un proceso en el que no tenemos en cuenta las simulaciones podemos empezar a diferenciar estas dos versiones.

En la versión 1, el bucle de las simulaciones se llevará a cabo dentro de cada coalición, entonces tendremos T veces cada $v(S)$, los cuales estarán creados con conjuntos de entrenamiento distintos. Para poder ejecutar el valor de Owen, previamente se calculará mediante la media de todos los $v(S)$, el $v(S)$ final para cada coalición.

Por el contrario, en la segunda versión, el bucle de las simulaciones se tomará como el bucle general, donde dentro de cada simulación se llevará a cabo el proceso para la extracción del valor de Owen sin realizar simulaciones entremedias. Por lo tanto, tendremos el valor de Owen de cada jugador repetido T veces, e igual que antes, realizaremos una media para extraer el valor final.

La mayoría de los datasets se podrán encontrar en paquetes disponibles de Rstudio, en caso contrario, se encontrará indicado en la bibliografía.

Una vez resueltas las dos versiones se realizará una pequeña conclusión para ver si existe diferencia en el ranking de influencia entre ellas.

Dataset 1: HousePrices

Disponible en el paquete: AER

Número total de observaciones: 546

Variable objetivo: *"prefer"*→ indica si la casa se encuentra en el barrio preferido de la ciudad, siendo posible "yes" (128) y "no" (418).

Variables explicativas:

- *"driveway"*: indica si la casa tiene entrada para vehículos, siendo posible "yes" (469) y "no" (77).
- *"recreation"*: indica si la casa tiene sala de recreativa, siendo posible "yes" (97) y "no" (449).
- *"fullbase"*: indica si la casa tiene un sótano acabado, siendo posible "yes" (191) y "no" (355).
- *"gasheat"*: indica si la casa utiliza gas para calentar el agua, siendo posible "yes" (25) y "no" (521).
- *"aircon"*: indica si la casa dispone de aire acondicionado, siendo posible "yes" (173) y "no" (373).

Versión 1:

		Random Forest		SVM	
Variable	**Características**	**Valor de Owen**	**Total**	**Valor de Owen**	**Total**
driveway	yes	-0,0482	0,1517	-0,0451	0,1525
	no	0,1999		0,1975	
recreation	yes	-0,0220	0,1551	-0,0213	0,1533
	no	0,1771		0,1747	
fullbase	yes	-0,0348	0,1532	-0,0355	0,1525
	no	0,1880		0,1881	
gasheat	yes	0,1425	0,1543	0,1384	0,1536
	no	0,0118		0,0152	
aircon	yes	0,0211	0,1553	0,0203	0,1539
	no	0,1342		0,1337	

Tabla 2: Resultados de la versión 1 en HousePrices

Versión 2:

		Random Forest		SVM	
Variable	**Características**	**Valor de Owen**	**Total**	**Valor de Owen**	**Total**
driveway	yes	-0,0419	0,1552	-0,0496	0,1543
	no	0,1971		0,2038	
recreation	yes	-0,0267	0,1576	-0,0321	0,1543
	no	0,1843		0,1864	
fullbase	yes	-0,0327	0,1532	-0,0460	0,1543
	no	0,1859		0,2003	
gasheat	yes	0,1338	0,1559	0,1352	0,1543
	no	0,0222		0,0190	
aircon	yes	0,0347	0,1549	0,0234	0,1543
	no	0,1202		0,1309	

Tabla 3: Resultados de la versión 2 en HousePrices

Conclusión: No influye ni el método ni la versión de código en este dataset ya que el ranking de influencia de los jugadores es el mismo, indicando que la característica con menos influencia es "*yes driveway*" y la que más, "*no driveway*". En cambio, la variable de esa característica (driveway) no es la variable que más influye ya que es aircon y recreation en la versión 1 y 2 respectivamente.

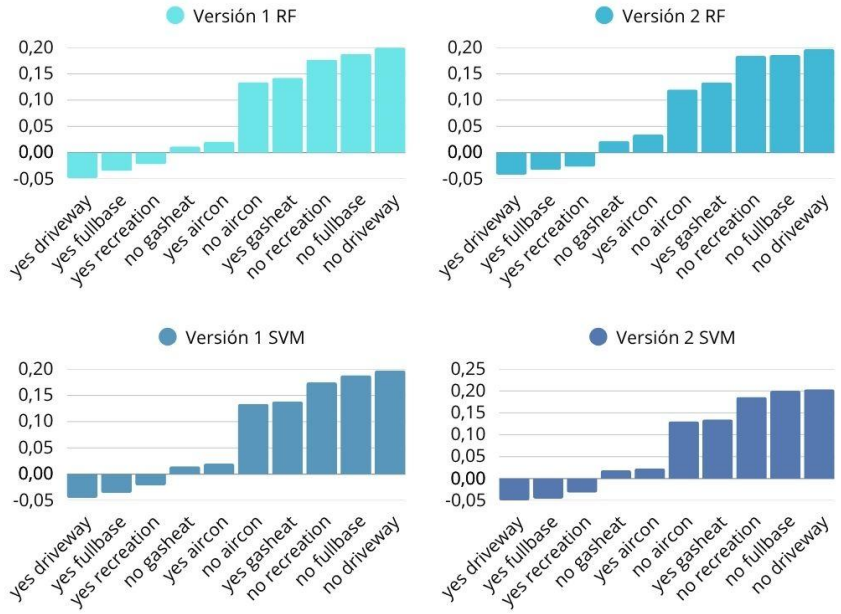

Gráfico 1: Comparación de los resultados en HousePrices

Dataset 2: Arrests

Disponible en el paquete: carData

Número total de observaciones: 5226

Variable objetivo: *"released"* → indica si el detenido fue puesto en libertad con una orden de comparecencia, siendo posible "No" (892) y "Yes" (4334).

Variables explicativas:

- *"colour"*: indica la raza del detenido, siendo posible "Black" (1288) y "White" (3938).
- *"sex"*: indica el género del detenido, siendo posible "Female" (443) y "Male" (4783).
- *"employed"*: indica si el detenido está empleado o no, siendo posible "No" (1115) y "Yes" (4111).
- *"citizen"*: indica si el detenido es ciudadano o no, siendo posible "No" (771) y "Yes" (4455).

Versión 1:

		Random Forest		SVM	
Variable	**Características**	**Valor de Owen**	**Total**	**Valor de Owen**	**Total**
colour	Black	0,0454	0,2074	0,0455	0,2076
	White	0,1620		0,1621	
sex	Female	0,1195	0,2070	0,1195	0,2074
	Male	0,0875		0,0879	
employed	Yes	0,1951	0,2083	0,1938	0,2074
	No	0,0132		0,0136	
citizen	Yes	0,1649	0,2075	0,1645	0,2073
	No	0,0426		0,0428	

Tabla 4: Resultados de la versión 1 en Arrests

Versión 2:

		Random Forest		SVM	
Variable	**Características**	**Valor de Owen**	**Total**	**Valor de Owen**	**Total**
colour	Black	0,0450	0,2087	0,0507	0,2068
	White	0,1637		0,1561	
sex	Female	0,1239	0,2086	0,1183	0,2068
	Male	0,0847		0,0885	
employed	Yes	0,1939	0,2087	0,1874	0,2068
	No	0,0148		0,0194	
citizen	Yes	0,1674	0,2087	0,1621	0,2068
	No	0,0413		0,0447	

Tabla 5: Resultados de la versión 2 en Arrests

Conclusión: El ranking de influencia de los jugadores es el mismo tanto en las dos versiones como en los dos métodos. Por lo tanto, la característica con más influencia es "*yes employed*" y la que menos, "*no employed*". En el caso de su variable (employed) solo es la más influyente en Random Forest tanto en la versión 1 como la 2. Por otro lado, en SVM la variable más influyente sería colour.

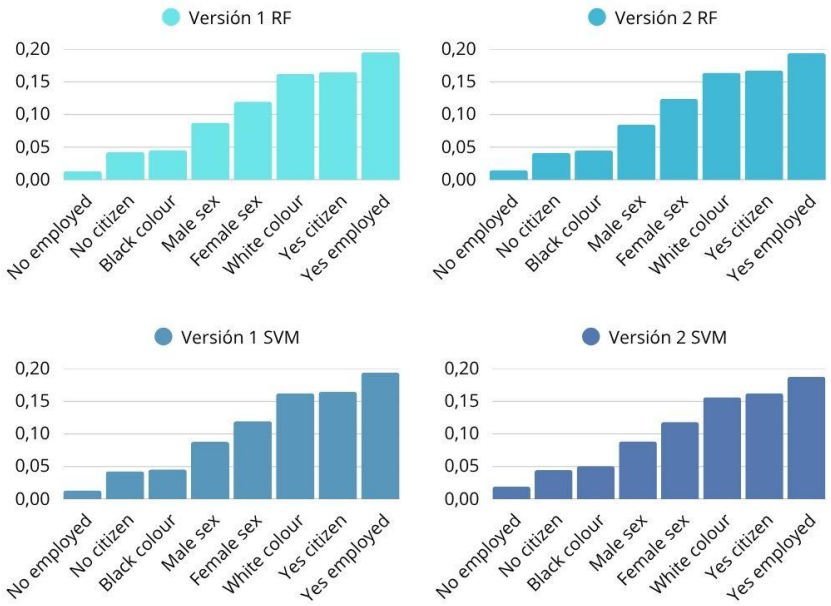

Gráfico 2: Comparación de los resultados en Arrests

Dataset 3: WVS

Disponible en el paquete: carData

Número total de observaciones: 5381

Variable objetivo: *"poverty"*→ indica si el individuo cree que lo que el gobierno está haciendo por las personas en situación de pobreza en el país es más o menos lo correcto "About Right" (1862), demasiado "Too Much" (811) o demasiado poco "Too Little" (2708).

Variables explicativas:

- *"degree"*: indica si el individuo tiene algún título universitario, siendo posible "yes" (1143) y "no" (4238).
- *"religion"*: indica si el individuo es miembro de alguna religión, siendo posible "yes" (4595) y "no" (786).
- *"country"*: indica país del individuo, siendo posible "Autralia" (1874), "Norway" (1127), "Sweden" (1003) y "USA" (1377).
- *"gender"*: indica el género del individuo, siendo posible "female" (2725) y "male" (2656).

Versión 1:

		Random Forest		SVM	
Variable	**Características**	**Valor de Owen**	**Total**	**Valor de Owen**	**Total**
degree	no	0,0776	0,1263	0,0759	0,1264
	yes	0,0488		0,0505	
religion	no	0,0604	0,1269	0,0627	0,1259
	yes	0,0665		0,0632	
country	Australia	0,0326	0,1248	0,0331	0,1269
	Norway	0,0452		0,0443	
	Sweden	0,0810		0,0820	
	USA	-0,0339		-0,0325	
gender	male	0,0394	0,1265	0,0376	0,1254
	female	0,0871		0,0878	

Tabla 6: Resultados de la versión 1 en WVS

Versión 2:

		Random Forest		SVM	
Variable	**Características**	**Valor de Owen**	**Total**	**Valor de Owen**	**Total**
degree	no	0,0712	0,1268	0,0721	0,1265
	yes	0,0556		0,0544	
religion	no	0,0632	0,1264	0,0633	0,1265
	yes	0,0632		0,0633	
country	Australia	0,0346	0,1268	0,0317	0,1265
	Norway	0,0386		0,0412	
	Sweden	0,0855		0,0800	
	USA	-0,0318		-0,0264	
gender	male	0,0366	0,1268	0,0404	0,1265
	female	0,0903		0,0861	

Tabla 7: Resultados de la versión 2 en WVS

Conclusión: En esta ocasión sí que encontramos una pequeña variación en la versión 2 de Random Forest, donde intercambia el "*no religion*" por el "*yes religion*". En este dataset vemos como la característica más influyente a "female gender" y como la que menos a "USA country". Su variable también coincide como la más influyente en todos los casos excepto en Random Forest de la versión 1 que es religion.

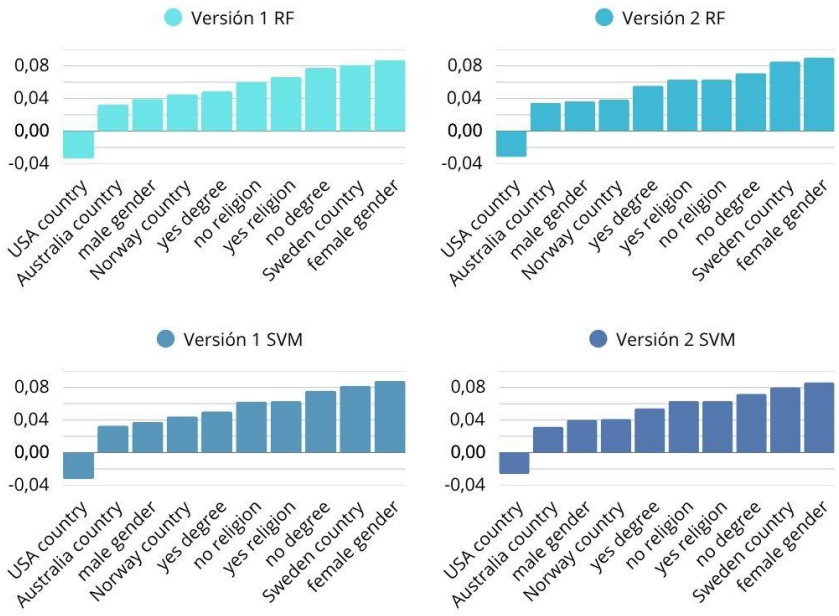

Gráfico 3: Comparación de los resultados en WVS

Dataset 4: titanic

Disponible en el paquete: COUNT

Número total de observaciones: 1316

Variable objetivo: *"survived"*→ indica si el individuo sobrevivió tras el hundimiento, siendo posible "yes" (499) y "no" (817).

Variables explicativas:

- *"class"*: indica la clase del individuo, siendo posible "1st class" (325), "2nd class" (285), "3rd class" (706) y "crew" (0).
- *"age"*: indica el grupo de edad del individuo, siendo posible "child" (109) y "adult" (1207).
- *"sex"*: indica el género del individuo, siendo posible "women" (447) y "man" (869).

Versión 1:

		Random Forest		SVM	
Variable	**Características**	**Valor de Owen**	**Total**	**Valor de Owen**	**Total**
class	1st class	0,0487	0,2464	0,0470	0,2352
	2nd class	0,0742		0,0728	
	3rd class	0,0716		0,0644	
	crew	0,0519		0,0510	
age	child	0,0477	0,2133	0,0466	0,2056
	adults	0,1656		0,1590	
gender	women	0,1327	0,3279	0,1349	0,3319
	man	0,1952		0,1970	

Tabla 8: Resultados de la versión 1 en titanic

Versión 2:

		Random Forest		SVM	
Variable	**Características**	**Valor de Owen**	**Total**	**Valor de Owen**	**Total**
class	1st class	0,0404	0,2476	0,0492	0,2344
	2nd class	0,0750		0,0742	
	3rd class	0,0793		0,0599	
	crew	0,0529		0,0511	
age	child	0,0483	0,2182	0,0615	0,2032
	adults	0,1699		0,1418	
gender	women	0,1215	0,3236	0,1423	0,3325
	man	0,2021		0,1902	

Tabla 9: Resultados de la versión 2 en titanic

Conclusión: En este caso no encontramos discrepancias en la versión 1 de los dos métodos, pero en la versión dos sí que hay bastantes diferencias. Según los resultados, las cuatro opciones coinciden en que la característica más influyente es "*man gender*" pero en el caso de la que menos influye, según la versión 1 es "*child age*" y según la versión 2, "*1st class*". En cuanto a la variable, sí que coincide la característica más influyente con la variable más influyente (gender).

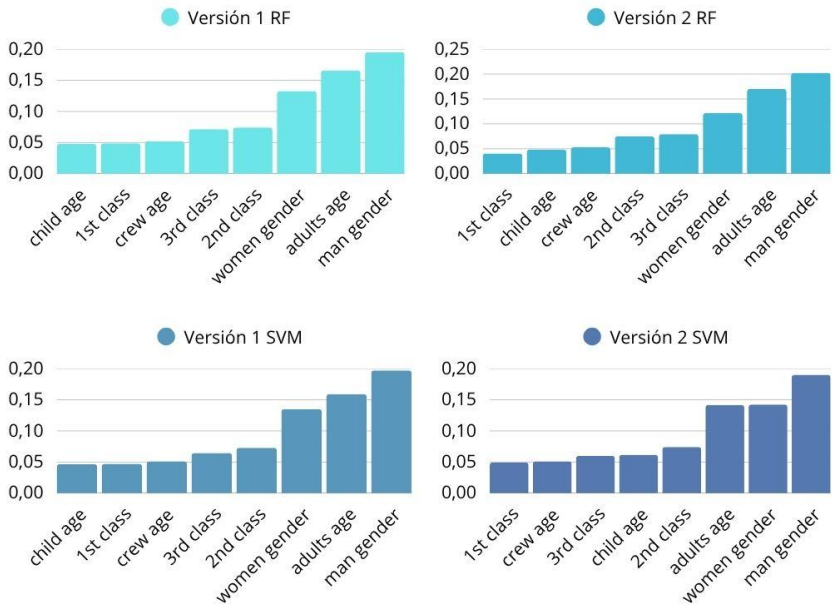

Gráfico 4: Comparación de los resultados en titanic

Dataset 5: autism

Disponible en el paquete: HLMdiag

Número total de observaciones: 604

Variable objetivo: *"bestest2"*→ indica el diagnóstico del niño a los dos años, siendo posible "autism" (389) y "pdd" (215).

Variables explicativas:

- *"sicdegp"*: indica la evaluación del desarrollo del lenguaje expresivo, siendo posible "low" (188), "med" (251) y "high" (165).
- *"gender"*: indica el género del individuo, siendo posible "male" (526) y "female" (78).
- *"race"*: indica la raza del individuo, siendo posible "white" (400) y "nonwhite" (204).

Versión 1:

		Random Forest		SVM	
Variable	**Características**	**Valor de Owen**	**Total**	**Valor de Owen**	**Total**
sicdegp	low	0,1717	0,2135	0,1813	0,2116
	med	0,0512		0,0430	
	high	-0,0094		-0,0127	
gender	male	0,0974	0,2235	0,0976	0,2147
	female	0,1261		0,1171	
race	white	0,1295	0,2068	0,1254	0,2136
	nonwhite	0,0773		0,0882	

Tabla 10: Resultados de la versión 1 en autism

Versión 2:

		Random Forest		SVM	
Variable	**Características**	**Valor de Owen**	**Total**	**Valor de Owen**	**Total**
sicdegp	low	0,1865	0,2260	0,1845	0,2185
	med	0,0390		0,0507	
	high	0,0005		-0,0167	
gender	male	0,1040	0,2285	0,0953	0,2185
	female	0,1245		0,1232	
race	white	0,1268	0,2109	0,1178	0,2185
	nonwhite	0,0841		0,1007	

Tabla 11: Resultados de la versión 2 en autism

Conclusión: Encontramos el mismo ranking menos en la versión dos de SVM. Pero si que coinciden en la característica más influyente y en la que menos que son "*low sicdegp*" y "*high sicdegp*" respectivamente. En cambio, no coincide la variable con mas influencia con la característica mas influyente ya que es gender.

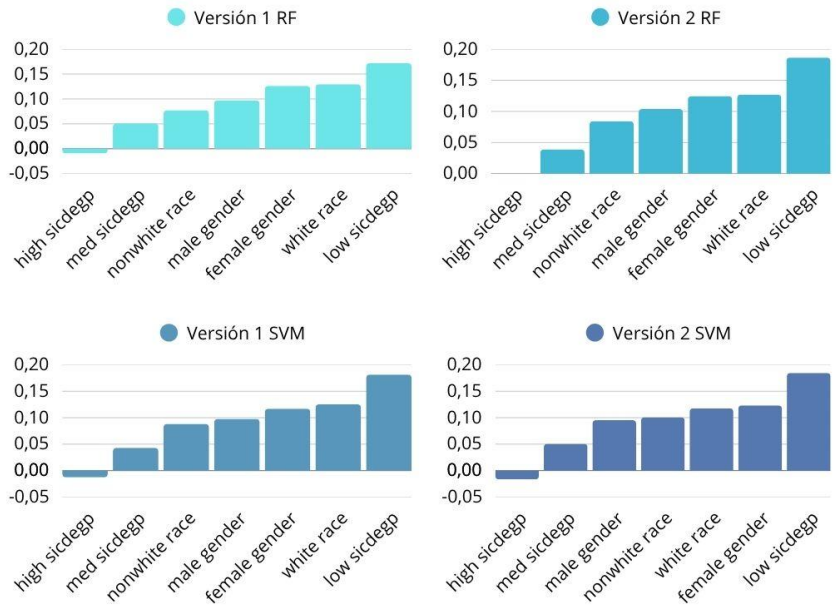

Gráfico 5: Comparación de los resultados en autism

Dataset 6: SmokeBan

Disponible en el paquete: AER

Número total de observaciones: 10000

Variable objetivo: *"smoker"*→ indica si el individuo es fumador, siendo posible "yes" (2423) y "no" (7577).

Variables explicativas:

- *"ban"*: indica si está prohibido fumar en el trabajo del individuo, siendo posible "yes" (6098) y "no" (3902).
- *"afam"*: indica si el individuo es afroamericano, siendo posible "yes" (769) y "no" (9231).
- *"hispanic"*: indica si el individuo es hispano, siendo posible "yes" (1134) y "no" (8866).
- *"gender"*: indica el género del individuo, siendo posible "male" (4363) y "female" (5637).

Versión 1:

		Random Forest		SVM	
Variable	**Características**	**Valor de Owen**	**Total**	**Valor de Owen**	**Total**
ban	no	0,0565	0,1905	0,0561	0,1894
	yes	0,1339		0,1333	
afam	no	0,0957	0,1891	0,0957	0,1886
	yes	0,0934		0,0929	
hispanic	no	0,0805	0,1900	0,0819	0,1899
	yes	0,1095		0,1081	
gender	male	0,0810	0,1898	0,0814	0,1890
	female	0,1088		0,1076	

Tabla 12: Resultados de la versión 1 en SmokeBan

Versión 2:

		Random Forest		SVM	
Variable	**Características**	**Valor de Owen**	**Total**	**Valor de Owen**	**Total**
ban	no	0,0575	0,1886	0,0562	0,1903
	yes	0,1311		0,1341	
afam	no	0,0999	0,1886	0,0925	0,1903
	yes	0,0887		0,0978	
hispanic	no	0,0826	0,1886	0,0794	0,1903
	yes	0,1060		0,1109	
gender	male	0,0829	0,1886	0,0799	0,1903
	female	0,1057		0,1104	

Tabla 13: Resultados de la versión 2 en SmokeBan

Conclusión: En este dataset podemos ver el mismo ranking tanto en la versión 1 como la 2 en Random Forest. En cambio, en SVM sí que existen diferencias entre las dos versiones. La característica más influyente sería "*yes ban*" y la que menos "*no ban*". Por otro lado, en la versión 2 todas las características son igual de influyentes tanto en Random Forest y SVM, pero en la versión 1 si que coindice la característica más influyente con la variable que es ban, pero en SVM sería hispanic.

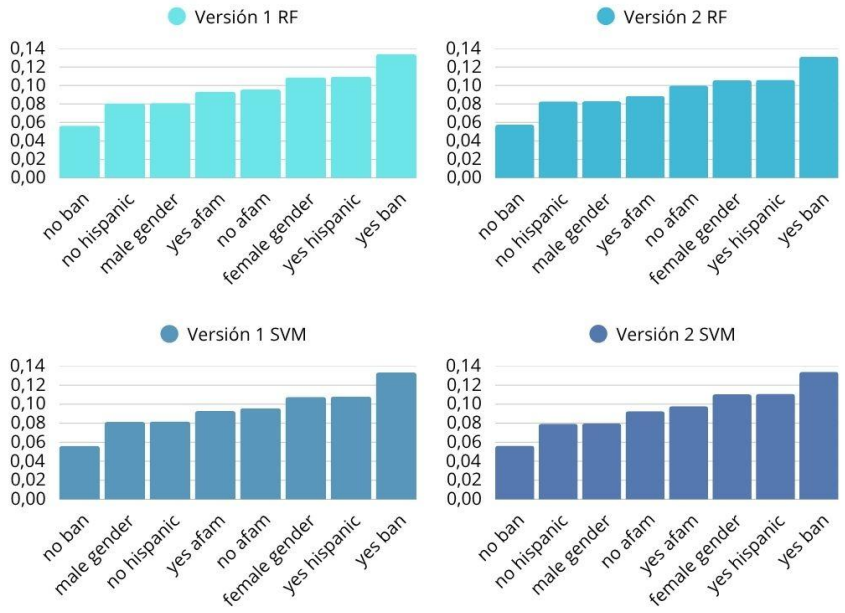

Gráfico 6: Comparación de los resultados en SmokeBan

Dataset 7: Quinidine

Disponible en el paquete: nlme

Número total de observaciones: 1471

Variable objetivo: *"Heart"*→ indica la insuficiencia cardíaca congestiva del individuo, siendo posible "No/Mild" (598), "Moderate" (375) y "Severe" (498).

Variables explicativas:

- *"Race"*: indica la raza del individuo, siendo posible "Caucasian" (968), "Latin" (384) y "Black" (119).
- *"Smoke"*: indica el hábito de fumar en el momento de la medición del individuo, siendo posible "no" (1024) y "yes" (447).
- *"Ethanol"*: indica el estado de abuso de etanol (alcohol) en el momento de la medición del individuo, siendo posible "none" (991), "current" (191) y "former" (289).
- *"Creatinine"*: indica el aclaramiento de creatinina (mg/min) del individuo, siendo posible "< 50" (418) y ">= 50" (1053).

Versión 1:

		Random Forest		SVM	
Variable	**Características**	**Valor de Owen**	**Total**	**Valor de Owen**	**Total**
Race	Caucasian	0,0315	0,1629	0,0279	0,1452
	Latin	0,0199		0,0100	
	Black	0,1115		0,1073	
Smoke	no	0,0790	0,1173	0,0691	0,1022
	yes	0,0382		0,0331	
Ethanol	none	0,0448	0,1416	0,0510	0,1149
	current	0,0716		0,0539	
	former	0,0252		0,0100	
Creatinine	< 50	0,0585	0,1284	0,0483	0,1075
	>= 50	0,0699		0,0592	

Tabla 14: Resultados de la versión 1 en Quinidine

Versión 2:

		Random Forest		SVM	
Variable	**Características**	**Valor de Owen**	**Total**	**Valor de Owen**	**Total**
Race	Caucasian	0,0295	0,1553	0,0292	0,1518
	Latin	0,0254		0,0158	
	Black	0,1004		0,1068	
Smoke	no	0,0779	0,1194	0,0699	0,1036
	yes	0,0415		0,0337	
Ethanol	none	0,0428	0,1407	0,0447	0,1076
	current	0,0670		0,0539	
	former	0,0310		0,0091	
Creatinine	< 50	0,0569	0,1260	0,0490	0,1144
	>= 50	0,0691		0,0653	

Tabla 15: Resultados de la versión 2 en Quinidine

Conclusión: Podemos encontrar diferencias tanto en las dos versiones como en los dos métodos. Sí que las cuatro opciones coinciden en la característica más influyente "Black race", pero Random Forest y SVM discrepan en la menos influyente ya que el primero indica que es "Latin Race" y el segundo, "former ethanol". En este caso, la característica más influyente coincide con la variable que es race.

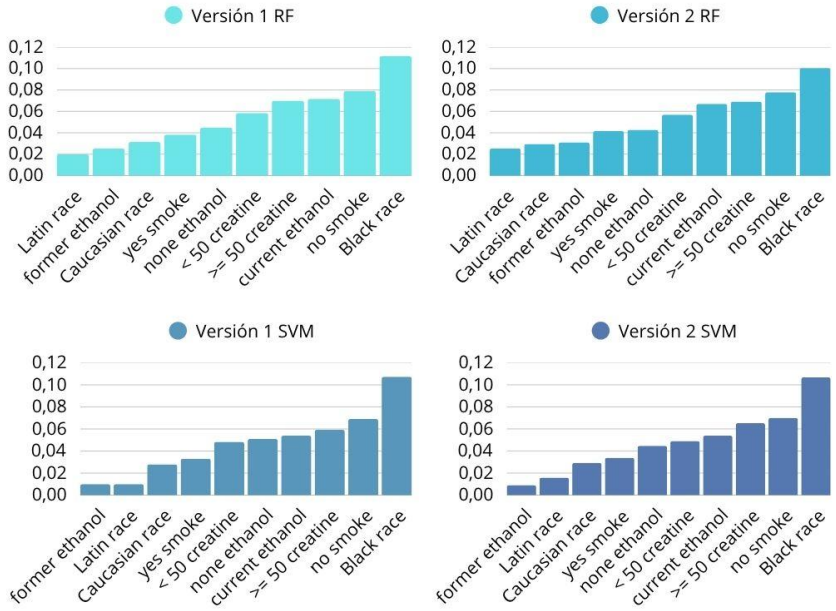

Gráfico 7: Comparación de los resultados en Quinidine

Dataset 8: Car Evaluation

Disponible en: [22]

Número total de observaciones: 1728

Variable objetivo: *"class"*→ indica la valoración del cliente al vehículo, siendo posible "unacc" (1210), "acc" (384), "good" (69) y "vgood" (65).

Variables explicativas:

- *"lug_boot"*: indica el nivel de capacidad del maletero del vehiculo, siendo posible "big" (576) y "med" (576) y "small" (576).
- *"safety"*: indica el nivel de seguridad del vehiculo, siendo posible "high" (576) y "med" (576) y "low" (576).

Versión 1:

		Random Forest		SVM	
Variable	**Características**	**Valor de Owen**	**Total**	**Valor de Owen**	**Total**
lug_boot	big	0,0672	0,3496	0,0673	0,3506
	med	0,1021		0,1060	
	small	0,1803		0,1773	
safety	high	-0,0482	0,3484	-0,0480	0,3479
	low	0,3417		0,3412	
	med	0,0549		0,0546	

Tabla 16: Resultados de la versión 1 en Car Evaluation

Versión 2:

		Random Forest		SVM	
Variable	**Características**	**Valor de Owen**	**Total**	**Valor de Owen**	**Total**
lug_boot	big	0,0680	0,3494	0,0673	0,3500
	med	0,1017		0,1034	
	small	0,1796		0,1792	
safety	high	-0,0489	0,3494	-0,0463	0,3500
	low	0,3411		0,3441	
	med	0,0572		0,0522	

Tabla 17: Resultados de la versión 2 en Car Evaluation

Conclusión: En este caso todos los rankings son iguales. Indicando como característica más influyente a “low safety” y la que menos, “high safety” pero no coincide la variable con la característica más influyente. Ahora bien, en la versión 2 las dos variables tienen la misma influencia.

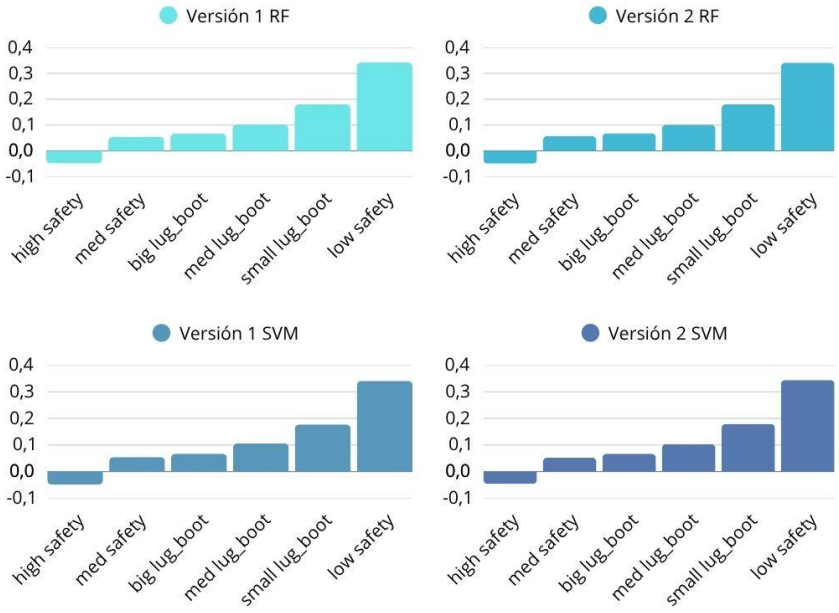

Gráfico 8: Comparación de los resultados en Car Evaluation

Dataset 9: Adult

Disponible en: [23]

Número total de observaciones: 32561

Variable objetivo: *"Income"*→ indica el ingreso anual del individuo, siendo posible ">50K" (7841) y "<=50K" (24720).

Variables explicativas:

- *"Education"*: indica el nivel de estudio del individuo, siendo posible "Advanced" (989), "Bachelors" (5355), "Dropout" (4253), "HighSchool" (10501), "Masters" (1723) y "SomeCollege" (9740).
- *"Sex"*: indica el sexo del individuo, siendo posible "Female" (10771) y "Male" (21790).

Cabe destacar que la variable "*education*" ha sido modificada del dataset original para agilizar el proceso y además mejorar la comprensión de la variable. Por lo tanto, se han agrupado las siguientes características:

- Dropout → "Preschool", "1st-4th", "5th-6th", "7th-8th", "9th", "10th", "11th", "12th"
- HighSchool → "HS-grad"
- SomeCollege → "Some-college", "Assoc-acdm", "Assoc-voc"
- Bachelors → "Bachelors"
- Masters → "Masters"
- Advanced → "Doctorate", "Prof-school"

Versión 1:

		Random Forest		SVM	
Variable	**Características**	**Valor de Owen**	**Total**	**Valor de Owen**	**Total**
Education	Advanced	0,0588	0,4008	0,0592	0,4013
	Bachelors	-0,0213		-0,0201	
	Dropout	0,1420		0,1440	
	HighSchool	0,1171		0,1172	
	Masters	0,0213		0,0175	
	SomeCollege	0,0829		0,0835	
Sex	Female	0,2790	0,3828	0,2785	0,3801
	Male	0,1039		0,1016	

Tabla 18: Resultados de la versión 1 en Adult

Versión 2:

		Random Forest		SVM	
Variable	**Características**	**Valor de Owen**	**Total**	**Valor de Owen**	**Total**
Education	Advanced	0,0591	0,4027	0,0600	0,4017
	Bachelors	-0,0201		-0,0201	
	Dropout	0,1429		0,1426	
	HighSchool	0,1167		0,1167	
	Masters	0,0203		0,0196	
	SomeCollege	0,0837		0,0829	
Sex	Female	0,2791	0,3817	0,2786	0,3795
	Male	0,1026		0,1010	

Tabla 19: Resultados de la versión 1 en Adult

Conclusión: En Random Forest podemos ver el mismo ranking. Por el otro lado, en SVM encontramos bastantes diferencias. Podemos ver que la característica "*Female*" es la más influyente en las dos versiones de Random Forest pero en SVM varía dependiendo de la versión. La menos influyente en Random Forest es "*Bachelors*" pero en SVM vuelve a variar. Aunque la característica más influyente varía dependiendo del método y versión, la variable education sigue coincidiendo con la característica más influyente.

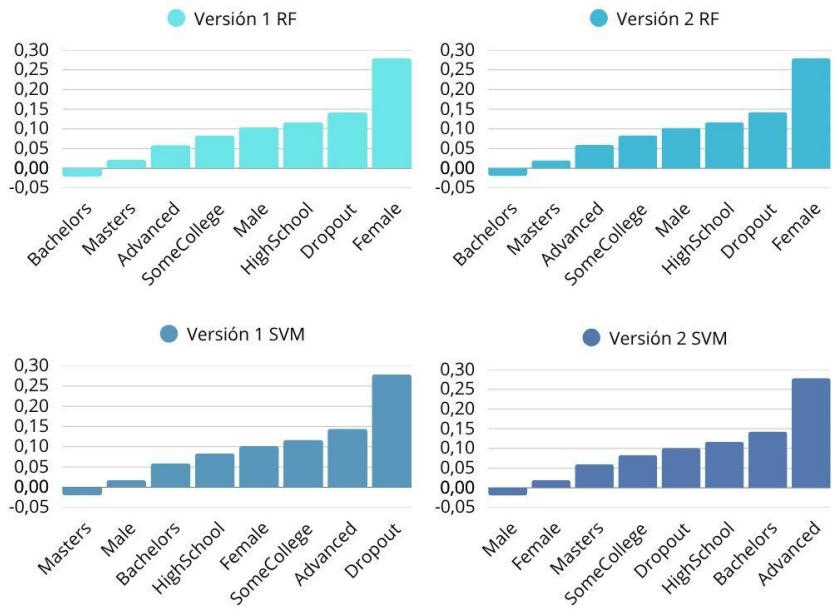

Gráfico 9: Comparación de los resultados en Adult

Dataset 10: Mushroom

Disponible en: [24]

Número total de observaciones: 8124

Variable objetivo: *"poisonous"*→ indica el hongo el comestible o venenoso, siendo posible "edible=e" (4208) y "poisonous=p" (3916).

Variables explicativas:

- *"odor"*: indica el olor de los hongos, siendo posible "almond=a" (400), "anise=l" (400), "creosote=c" (192), "fishy=y" (576), "foul=f" (2160), "musty=m" (36), "none=n" (3528), "pungent=p" (256) y "spicy=s" (576).
- *"population"*: indica el nivel de abundancia de los hongos, siendo posible "abundant=a" (384), "clustered=c" (340), "numerous=n" (400), "scattered=s" (1248), "several=v" (4040) y "solitary=y" (1712).

Debido al alto costo computacional, este dataset solo se ha resuelto con la versión 1.

Versión 1:

		Random Forest		SVM	
Variable	**Características**	**Valor de Owen**	**Total**	**Valor de Owen**	**Total**
odor	almond (a)	0,0722	0,6276	0,0731	0,6268
	anise (l)	0,0728		0,0713	
	creosote (c)	0,07153		0,0715	
	fishy (y)	0,0716		0,0723	
	foul (f)	0,0754		0,0758	
	musty (m)	0,0705		0,0716	
	none (n)	0,0495		0,0469	
	pungent (p)	0,0713		0,0721	
	spicy (s)	0,0726		0,0722	
population	abundant (a)	0,0970	0,3552	0,0945	0,3560
	clustered (c)	0,0635		0,0659	
	numerous (n)	0,0978		0,0963	
	scattered (s)	0,0479		0,0477	
	several (v)	0,0301		0,0318	
	solitary (y)	0,0189		0,0198	

Tabla 20: Resultados de la versión 1 en Mushroom

Conclusión: Podemos ver algunas diferencias dependiendo del método que seleccionemos. La característica más influyente es "*numerous*" y la que menos, "*solitary*". Por lo tanto, no coincide la variable más influyente con la característica más influyente ya que es odor.

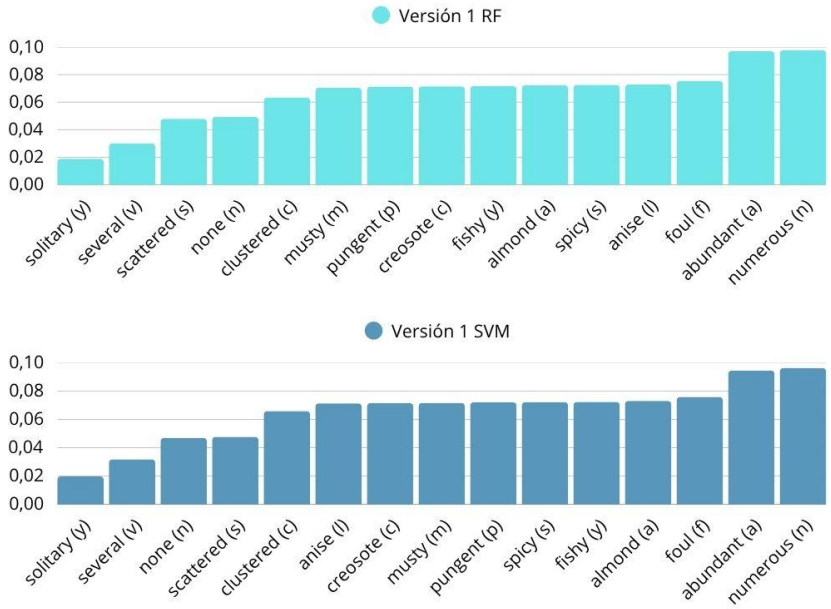

Gráfico 10: Comparación de los resultados en Mushroom

6 Conclusión

Este estudio, además de identificar las variables más influyentes, permite analizar cómo ciertas combinaciones de valores pueden afectar al rendimiento del modelo tanto positiva como negativamente. Por lo que, gracias a este proyecto, podemos concluir que la medida de influencia de Owen es una herramienta útil para predecir qué valores, dentro de las variables explicativas discretas, son más significativos para la optimalidad del modelo de clasificación.

Ahora bien, debemos destacar que nos hemos encontrado con algunas dificultades en el camino. Por ejemplo, el alto coste computacional hace que, en modelos donde tenemos un número elevado de variables discretas o de valores posibles dentro de las variables, se requiera bastante tiempo para poder ejecutar el código. Por otro lado, especialmente con Random Forest, al intentar predecir en ciertas coaliciones de valores, nos podemos encontrar con que solo existe una clase de la variable objetivo. Por ello, en algunos datasets concretos, hemos tenido que realizar alguna pequeña modificación en el código como, por ejemplo, añadir la función *droplevels* en el conjunto de entrenamiento y en el de test. Así podremos eliminar las clases no utilizadas y quedarnos solo con la única clase que existe.

Al extraer esta información, se puede mejorar la selección de variables y la reducción de dimensiones. Esto nos abre la puerta a que investigaciones futuras desarrollen infinidad de proyectos que se basen en criterios de influencia.

7 Bibliografía

[1] A. García, G. L. Martínez, G. Núñez y A. Guzmán, «CLASIFICACIÓN SUPERVISADA INDUCCIÓN DE ARBOLES DE DECISIÓN, ALGORITMO k-d,» 1998.

[2] S. Monsalve, John Nash y la teoría de juegos, 2003.

[3] C. A. RESTREPO CARVAJAL, «APROXIMACIÓN A LA TEORÍA DE JUEGOS,» *Revista Ciencias Estratégicas,* 2009.

[4] F. Parra, Estadística y Machine Learning con R, 2019.

[5] T. l. d. reservados, «Introducción a Support Vector Machine (SVM),» MathWorks, [En línea]. Available: https://es.mathworks.com/discovery/support-vector-machine.html. [Último acceso: 1 Marzo 2025].

[6] Gareth James, Trevor Hastie, Daniela Witten y Robert Tibshirani, An Introduction to Statistical Learning with Applications in R, 2023.

[7] J. R. Gonzalez, Aprendizaje Automático 1, Barcelona: Universidad Autónoma de Barcelona (UAB), 2023.

[8] A. C. Müller, Linear Models for Classification, 2018.

[9] T. l. d. reservados, «Ventajas y desventajas de SVM,» InteractiveChaos, [En línea]. Available: https://interactivechaos.com/es/manual/tutorial-de-machine-learning/ventajas-y-desventajas-de-svm. [Último acceso: 3 Marzo 2025].

[10] C. Beltrán y I. Barbona , «Una revisión de las técnicas de clasificación supervisada en la clasificación automática de textos».

[11] A. Navarro Sellés , ÁRBOLES DE CLASIFICACIÓN: DE LO CLÁSICO A LO ÓPTIMO. DESARROLLO E IMPLEMENTACIÓN DE UNA FORMULACIÓN DE ÁRBOL ÓPTIMO., Elche: Universidad Miguel Hernandez , 2022.

[12] T. l. d. reservados, «Random Forest,» InteractiveChaos, [En línea]. Available: https://interactivechaos.com/es/wiki/random-forest. [Último acceso: 15 Marzo 2025].

[13] T. l. d. reservados, «Random forest, la gran técnica de Machine Learning,» INESDI, 27 Enero 2023. [En línea]. Available: https://www.inesdi.com/blog/random-forest-que-es/. [Último acceso: 20 Marzo 2025].

[14] A. Estévez-Fernández, M. G. Fiestras-Janeiro, M. A. Mosquera y E. Sánchez- Rodríguez, A Bankruptcy Approach to the Core, 2012.

[15] M. Schuster Puga, Estructuras Jerárquicas y Juegos Cooperativos con Utilidad, Santiago de Compostela: Universidad Santiago de Compostela, 2013.

[16] R. Gilles, G. Owen y J. van den Brink, Games with permission structures: The conjunctive, 1991.

[17] C. Albert García, «Teoría de Juegos Cooperativos,» de *MÉTODOS DE ASIGNACIÓN DE LOS COSTES DE LIMPIEZA DE UN RÍO Y DEL REPARTO DEL AGUA*, Sevilla, Universidad de Sevilla, 2012.

[18] E. CERDÁ, J. PÉREZ y J. L. JIMENO, TEORÍA DE JUEGOS, Madrid: PEARSON EDUCACIÓN, S.A, 2004.

[19] T. l. d. reservados, Juegos con Utilidad Transferible.

[20] T. l. d. reservados, «Valor de Shapley,» Wikipedia, [En línea]. Available: https://es.wikipedia.org/wiki/Valor_de_Shapley. [Último acceso: 6 Abril 2025].

[21] J. M. Alonso Meijide, Contribuciones a la teoría del valor en juegos cooperativos con condicionamientos exógenos, 2002.

[22] M. Bohanec, «Car Evaluation,» UCI Machine Learning Repository, 1988. [En línea]. Available: https://archive.ics.uci.edu/dataset/19/car+evaluation. [Último acceso: 27 Mayo 2025].

[23] B. B. a. R. Kohavi., «Adult,» UCI Machine Learning Repository, 1996. [En línea]. Available: https://archive.ics.uci.edu/dataset/2/adult. [Último acceso: 27 Mayo 2025].

[24] «Mushroom,» UCI Machine Learning Repository, 1981. [En línea]. Available: https://archive.ics.uci.edu/dataset/73/mushroom. [Último acceso: 03 Junio 2025].