FACULTAD DE ESTUDIOS ESTADÍSTICOS

GRADO EN ESTADISTICA APLICADA

Curso 2024/2025

Trabajo de Fin de Grado

TÍTULO: Análisis de modelos predictivos con machine learning para determinar la probabilidad de jugar a videojuegos

***Alumno*: Lucas de la Fuente Toubes**

***Tutor*: Silvia Pineda San Juan**

Junio de 2025

UNIVERSIDAD COMPLUTENSE
MADRID

Índice

Índice Ilustraciones

Resumen

El estudio del comportamiento de los jugadores de videojuegos es clave para comprender las preferencias y hábitos de consumo en esta creciente industria. Este trabajo se enfoca en analizar los perfiles de jugadores utilizando modelos estadísticos y técnicas de clasificación. El objetivo principal es identificar patrones relevantes y segmentar a los jugadores según variables sociodemográficas y de comportamiento.

Para ello, se realizó una depuración de la base de datos, seguida de un análisis exploratorio mediante frecuencias y relaciones entre variables clave. Posteriormente, se aplicaron modelos como regresión logística binaria, Naive Bayes, árboles de clasificación y Random Forest. Los resultados muestran que ciertos factores, como seguir contenido de videojuegos en redes sociales, tienen una fuerte relación con jugar o no a los videojuegos. Además, el modelo de Bosques Aleatorios (Random Forest) destacó por su precisión del 88%, superando a otros enfoques.

Estos hallazgos permiten una mejor comprensión del perfil de los jugadores y tienen aplicaciones potenciales en estrategias de marketing y diseño de videojuegos.

Aunque este análisis ofrece una visión inicial sobre los jugadores de videojuegos, el tamaño reducido de la muestra limita la robustez de los hallazgos y su generalización a una población más amplia. Futuras investigaciones con muestras más grandes podrían confirmar estas tendencias.

Abstract

The study of video game players' behavior is essential to understanding preferences and consumption habits in this growing industry. This work focuses on analyzing player profiles using statistical models and classification techniques. The main objective is to identify relevant patterns and segment players based on sociodemographic and behavioral variables.

To achieve this, the database was cleaned and subjected to an exploratory analysis through frequencies and relationships between key variables. Subsequently, models such as binary logistic regression, Naive Bayes, classification trees, and Random Forest were applied. The results show that certain factors, such as following video game content on social media, have a strong relationship with whether individuals play video games or not. Additionally, the Random Forest model stood out with an accuracy of 88%, outperforming other approaches.

These findings provide a better understanding of player profiles and offer potential applications in marketing strategies and video game design. However, while this analysis provides an initial insight into video game players, the small sample size limits the robustness of the findings and their generalization to a broader population. Future research with larger datasets could validate these trends.

Introducción

Contexto y motivación del trabajo.

En los últimos años, la industria de los videojuegos ha experimentado un crecimiento exponencial, consolidándose como una de las principales fuentes de entretenimiento a nivel global y transformándose en un fenómeno cultural y económico de gran relevancia. *"Las comunidades de internet comparten un interés o pasión común. Los intereses pueden ser tan variados como las aficiones o casuísticas de las personas"* [1]. Este auge no solo ha popularizado los videojuegos entre millones de personas, sino que también ha generado una cantidad significativa de datos relacionados con el comportamiento de los jugadores, las mecánicas de juego y las preferencias de consumo. En este contexto, el análisis de datos se ha convertido en una herramienta clave para comprender patrones, identificar tendencias y optimizar diversos aspectos de la experiencia del jugador, desde el diseño de los videojuegos hasta estrategias de mercado.

La motivación de este trabajo radica en explorar cómo los datos pueden aprovecharse para aportar valor en diferentes aspectos de esta industria, especialmente en la identificación de los factores que determinan si una persona juega o no a videojuegos. Este enfoque no solo contribuye al entendimiento del perfil de los jugadores, sino que también ofrece información útil para la industria en términos de diseño, segmentación y estrategias basadas en datos.

Pregunta de investigación.

A pesar de la disponibilidad de grandes volúmenes de datos relacionados con los videojuegos, estos no siempre se emplean de manera efectiva para comprender las características y los factores que influyen en el comportamiento de las personas respecto a jugar o no a videojuegos. Identificar estos patrones puede aportar perspectivas valiosas sobre el perfil de los jugadores, lo que a su vez beneficia a la industria en aspectos como el diseño de productos, la formulación de estrategias de marketing y el desarrollo de comunidades. En este trabajo, se busca aportar claridad sobre este tema mediante la aplicación de modelos estadísticos y de árboles de decisión, analizando de manera precisa qué variables tienen un impacto significativo en la decisión de jugar.

La pregunta de investigación que guía este trabajo es: ¿Cuáles son los factores determinantes que influyen en si una persona juega o no a videojuegos? Abordar esta cuestión no solo permite comprender en mayor profundidad el comportamiento de los usuarios, sino que también puede servir como base para el desarrollo de estrategias basadas en datos que optimicen la experiencia del jugador y fortalezcan la conexión entre los videojuegos y sus audiencias.

Objetivos generales y específicos.

El objetivo general de este trabajo es aplicar modelos de análisis de datos para identificar y analizar los factores clave que influyen en el comportamiento de juego. Para lograrlo, se establecen cuatro objetivos específicos:

1) Seleccionar y preparar un conjunto de datos adecuado que incluya características demográficas y conductuales.

2) Implementar diferentes modelos estadísticos, como la regresión logística binaria [2], métodos de selección automática de variables, Naive Bayes y técnicas basadas en árboles de decisión, incluyendo Random Forest y Bagging [3][4], para predecir la variable objetivo "jugar".
3) Evaluar el rendimiento de estos modelos utilizando métricas de calidad estándar.
4) Interpretar los resultados obtenidos para extraer conclusiones relevantes sobre el perfil de los jugadores y su comportamiento.

Estructura del documento.

El presente documento se organiza en varios capítulos. En primer lugar, se introduce el marco teórico, que abarca las definiciones de conceptos clave, los fundamentos metodológicos de los modelos utilizados y una revisión de literatura relevante. Posteriormente, en el capítulo de metodología estadística, se describe el conjunto de datos empleado, las técnicas de análisis utilizadas y el proceso de selección de variables. En el capítulo de resultados, se presentan los hallazgos obtenidos a través de gráficos y visualizaciones, acompañados de una interpretación detallada. Finalmente, el documento concluye con una discusión sobre las implicaciones prácticas de los resultados, las limitaciones del estudio y posibles líneas de investigación futura, con el objetivo de contribuir al entendimiento del comportamiento de los jugadores y al desarrollo de estrategias basadas en datos en el ámbito de los videojuegos.

Marco teórico

Conceptos clave y definiciones.

Estadístico V de Cramer

En la construcción de modelos predictivos, la selección de variables relevantes es crucial para mejorar la calidad de las predicciones. Para identificar qué variables están relacionadas con la variable objetivo, es recomendable comenzar con un análisis gráfico que permita visualizar posibles asociaciones. Sin embargo, aunque este método proporciona información valiosa, sus conclusiones pueden ser subjetivas. Por esta razón, es fundamental emplear medidas cuantitativas para evaluar la relación entre variables.

Uno de los estadísticos más utilizados en estos casos es el chi-cuadrado (X^2). No obstante, este estadístico puede tomar valores muy grandes, dificultando la interpretación de su magnitud sin realizar contrastes de hipótesis. El V de Cramer, basado en el estadístico chi-cuadrado, ofrece una solución a esta limitación, ya que su valor está acotado entre 0 y 1, lo que facilita su interpretación. Su fórmula es:

$$V = \sqrt{\frac{\mathrm{X}^2}{n \times \min(l-1, k-1)}}$$

Donde:

- X^2 es la estadística de chi-cuadrado calculada,
- n es el tamaño de la muestra,

- k y l representan el número de categorías de las variables involucradas.

El V de Cramer toma el valor 0 cuando las variables son independientes, y 1 cuando existe una dependencia total entre ellas. Se prefiere el uso del V de Cramer frente a la prueba chi-cuadrado porque permite estudiar cualquier tipo de relación, no solo lineales, y proporciona una métrica normalizada que permite la comparación entre diferentes conjuntos de datos. El análisis mediante este estadístico permite identificar aquellas variables con mayor potencial predictivo, lo que facilita la selección de las más relevantes para el modelo.

Escalamiento multidimensional (MDS) para variables

El escalamiento multidimensional (MDS) es una técnica que permite representar relaciones de proximidad entre objetos como distancias en un espacio de menor dimensión, facilitando su visualización e interpretación [5]. Aunque el escalamiento multidimensional fue inicialmente diseñado para representar observaciones, este puede ser fácilmente adaptado para la representación de variables.

Tal y como ocurre con la versión “tradicional”, el objetivo es obtener un gráfico en el que se representen las variables, de manera que aquellas que se muestren cercanas estén relacionadas y aquellas que se muestren alejadas, no. En este caso, de nuevo, es necesario aportar una matriz de disimilaridades, que aporte información sobre la medida de asociación que se quiera plasmar en el gráfico, en nuestro caso el V de Cramer.

El principal inconveniente de esta medida es que se trata de medida de similaridad, por lo que debe ser transformada previamente. La transformación más frecuente en el ámbito del MDS para variables es también:

$$\delta_{ij} = 1 - s_{ij}$$

Esta transformación es aplicable a medidas acotadas entre 0 y 1 por lo que es útil para el V de Cramer. En este caso, grandes distancias implicarán ausencia de relación mientras que observaciones cercanas estarán relacionadas aunque se desconoce el tipo exacto de relación.

Clúster de variables

El escalamiento multidimensional (MDS) suele complementarse con análisis clúster para visualizar gráficamente las agrupaciones formadas y su homogeneidad [6]. En este caso, se emplea un clúster de variables jerárquico, basado en el análisis de componentes principales para datos cualitativos (PCAMIX).

- Análisis clúster jerárquico: genera una jerarquía de agrupaciones en función de la similitud entre variables.
- PCAMIX (para datos cualitativos): es una extensión del análisis de correspondencias que transforma las variables categóricas en dummies y extrae componentes que resumen la información.

La agrupación jerárquica se basa en la fusión de grupos que minimicen la pérdida de homogeneidad, evaluada a partir de los autovalores obtenidos en PCAMIX. Aunque la representación en MDS ayuda a determinar el número óptimo de clústeres, no es completamente precisa, por lo que se complementa con el dendograma, que permite visualizar la similitud entre variables.

Para garantizar la estabilidad de los clústeres, se recomienda:

1. Muestras bootstrap: submuestras aleatorias con reemplazo para evaluar la robustez de las agrupaciones.
2. Índice de Rand: métrica entre 0 y 1 que cuantifica la similitud entre dos agrupaciones, indicando si la estructura de clústeres es consistente.

Este enfoque permite interpretar mejor la estructura de los datos cualitativos y validar la estabilidad de las agrupaciones en el análisis MDS.

Modelo de regresión logística binario

La regresión busca predecir una variable dependiente Y a partir de un conjunto de m variables independientes X_1, X_2, ..., X_m. Cuando Y es continua se utiliza un modelo lineal, pero para variables binarias (donde Y = 0 o Y = 1) se asume que Y sigue una distribución Binomial. Así, se modela la probabilidad del evento (por ejemplo, Y = 1) mediante:

$$P(Y = 1 \mid X_1, \dots, X_m) = f(X_1, \dots, X_m)$$

Dado que una función lineal podría producir valores fuera del intervalo (0,1), se utiliza la función logística como enlace:

$$P(Y = 1 \mid X_1, \dots, X_m) = \frac{e^{\beta_0 + \beta_1 X_1 + \cdots + \beta_m X_m}}{1 + e^{\beta_0 + \beta_1 X_1 + \cdots + \beta_m X_m}}$$

Interpretación de Parámetros:

- **Variable Dicotómica:**
  Al incluir variables cualitativas mediante variables dummy, el coeficiente β1 se interpreta a través del odds ratio. Es decir, $e^{\beta 1}$ indica cuánto cambian las probabilidades del evento al comparar las dos categorías (por ejemplo, X = 1 frente a X = 0).
- **Variable Categórica:**
  Se generan tantos coeficientes como categorías menos una (categoría de referencia). Cada coeficiente se interpreta de forma similar a las variables dicotómicas, comparando la categoría correspondiente con la de referencia.
- **Variable Continua:**
  Aquí, $e^{\beta 1}$ representa el cambio en el odds del evento por cada incremento unitario en la variable, manteniendo constantes las demás variables.

Estimación de Parámetros:
Los parámetros βj se estiman mediante el método de máxima verosimilitud. Dado que no existe una solución analítica directa, se emplean métodos iterativos de optimización para encontrar los valores que maximizan la función de verosimilitud basada en la distribución Binomial.

Análisis del Modelo:

- **Estadístico de Wald:** Se utiliza para contrastar la significatividad de cada parámetro, comprobando si difieren significativamente de cero.

- **Análisis de Tipo II:** Evalúa la importancia de cada variable midiendo la disminución en la verosimilitud al eliminarla, lo que permite ordenar las variables según su contribución al modelo.

Evaluación del Modelo:

- **Matriz de Confusión:** Permite comparar las clasificaciones predichas (utilizando un punto de corte, comúnmente 0.5) con las observaciones reales, de donde se derivan medidas como la tasa de acierto, sensibilidad (verdaderos positivos) y especificidad (verdaderos negativos).
- **Índice Kappa:** Ajusta la tasa de acierto para tener en cuenta los aciertos que podrían producirse por azar, ofreciendo una medida más robusta de la concordancia entre predicción y realidad.
- **Curva ROC y AUC:** La curva ROC representa la sensibilidad frente a 1-especificidad para todos los puntos de corte, permitiendo evaluar el desempeño del modelo sin fijar un umbral específico. El área bajo la curva (AUC) cuantifica esta capacidad, siendo 1 el modelo perfecto y 0.5 equivalente a una predicción aleatoria.

Métodos automáticos de selección de variables

En regresión logística binaria, la selección de variables es clave para construir modelos óptimos sin probar todas las combinaciones posibles. Para ello, se emplean métodos automáticos como:

- Backward (hacia atrás): Parte de un modelo con todas las variables y elimina progresivamente las menos influyentes hasta que la eliminación de cualquier otra empeore el modelo. No permite reincorporar variables eliminadas.
- Forward (hacia adelante): Inicia con un modelo vacío y agrega variables una a una según su contribución al modelo. Una vez añadidas, no pueden eliminarse.
- Stepwise (paso a paso): Combina ambos enfoques, permitiendo tanto la incorporación como la eliminación de variables según su impacto en cada iteración.

Para medir la mejora del modelo durante la selección, se usan criterios como el AIC (criterio de información de Akaike) y el BIC (criterio de información bayesiano), que penalizan la complejidad del modelo para evitar sobreajuste. La validación cruzada también es recomendable para evaluar el rendimiento final de los modelos generados.

Modelo de Naive Bayes

El modelo Naive Bayes es un método de clasificación supervisada basado en la teoría de la probabilidad, específicamente en el Teorema de Bayes. Su objetivo es calcular la probabilidad de que una observación pertenezca a una determinada categoría de la variable dependiente, dadas un conjunto de variables independientes.

Matemáticamente, el modelo se basa en la siguiente ecuación:

$$P(Y = j \mid X1, X2, \ldots, Xm) = \frac{P(X1, X2, \ldots, Xm \mid Y = j)P(Y = j)}{P(X1, X2, \ldots, Xm)}$$

Dado que el denominador es el mismo para todas las categorías, en la práctica se trabaja con la forma proporcional:

$$P(Y = j \mid X1, X2, \ldots, Xm) \propto P(X1, X2, \ldots, Xm \mid Y = j)P(Y = j)$$

Para simplificar los cálculos, Naive Bayes asume independencia condicional entre las variables independientes, lo que permite descomponer la probabilidad conjunta en el producto de probabilidades individuales:

$$P(X1, X2, \ldots, Xm \mid Y = j) = \prod_{i=1}^{m} P(Xi \mid Y = j)$$

Esta suposición, aunque rara vez se cumple completamente en datos reales, permite que el modelo sea computacionalmente eficiente y fácil de implementar.

Estimación de Probabilidades

Para aplicar el modelo, se deben estimar dos tipos de probabilidades:

- Probabilidad de la clase P(Y=j), que se obtiene a partir de la frecuencia de cada categoría en los datos.
- Probabilidades condicionales P(Xi|Y=j) que dependen del tipo de variable:
    - Para variables categóricas, se estiman mediante frecuencias relativas.
    - Para variables continuas, se suele asumir una distribución normal:

También pueden utilizarse métodos no paramétricos como Kernel Density Estimation (KDE) para estimar la densidad sin asumir una distribución específica.

Problema de la Frecuencia Nula y Ajuste de Laplace

Cuando una combinación de valores entre la variable dependiente y una independiente no aparece en los datos, la probabilidad condicional se estima en cero, lo que impide que el modelo clasifique correctamente ciertos casos. Para evitar esto, se aplica el ajuste de Laplace, que consiste en sumar una constante positiva λ a todas las frecuencias observadas, evitando la aparición de probabilidades nulas.

Ventajas:

- Es computacionalmente eficiente y escalable a grandes volúmenes de datos.
- Puede aplicarse tanto a variables categóricas como continuas.
- Es robusto a la presencia de variables irrelevantes.
- Tolera bien valores faltantes en las variables explicativas.

Desventajas:

- La independencia entre variables rara vez se cumple, lo que puede afectar la precisión del modelo.
- No captura interacciones entre las variables independientes.
- Puede verse afectado por el problema de la frecuencia nula si no se aplica el ajuste adecuado.

Evaluación del Modelo

El rendimiento del modelo Naive Bayes se evalúa utilizando métricas como la matriz de confusión, el índice Kappa y la curva ROC con el área bajo la curva (AUC).

También es recomendable aplicar validación cruzada para obtener una medida más fiable del desempeño del modelo.

Selección de Variables

Dado que los métodos tradicionales como AIC o BIC no son aplicables en este caso, una alternativa es utilizar el algoritmo Recursive Feature Elimination (RFE), que selecciona iterativamente las variables más relevantes en función de su capacidad predictiva. También se recomienda agrupar variables similares para evitar redundancias y mejorar la interpretación del modelo.

Modelos Lasso y Elastic Net

Los modelos Lasso y Elastic Net son técnicas de regularización utilizadas en regresión lineal para mejorar la capacidad predictiva y la interpretación del modelo. Estas técnicas permiten reducir la complejidad del modelo al penalizar los coeficientes de las variables explicativas, evitando el sobreajuste y mejorando la selección de características relevantes.

Regresión Lasso

El modelo Lasso (Least Absolute Shrinkage and Selection Operator) introduce una penalización basada en la norma L1, lo que tiene como efecto la reducción de algunos coeficientes a cero, permitiendo así la selección automática de variables. La función de costo de la regresión lineal con penalización Lasso se expresa como:

$$\min_{\beta} \sum_{i=1}^{n} (y_i - X_i\beta)^2 + \lambda \sum_{j=1}^{m} |\beta_j|$$

donde:

- y_i es la variable dependiente,
- X_i representa las variables independientes,
- β_j son los coeficientes del modelo,
- λ es el parámetro de regularización que controla la intensidad de la penalización.

Cuando λ es grande, más coeficientes son reducidos a cero, simplificando el modelo y mejorando su interpretabilidad.

Regresión Elastic Net

El modelo Elastic Net es una extensión de Lasso que combina la penalización L1 con la penalización L2 (utilizada en la regresión Ridge). Su función de costo es:

$$\min_{\beta} \sum_{i=1}^{n} (y_i - X_i\beta)^2 + \alpha \left(\lambda \sum_{\mathrm{j=1}}^{\mathrm{m}} \beta^2 + (1-\lambda) \sum_{\mathrm{j=1}}^{\mathrm{m}} \left|\beta_{\mathrm{j}}\right| \right)$$

Esta combinación permite:

- Seleccionar variables como Lasso, eliminando coeficientes irrelevantes.

- Evitar problemas de multicolinealidad, manteniendo coeficientes pequeños pero no necesariamente cero, como en Ridge.

El parámetro α regula la combinación de ambas penalizaciones:

- Si α=1, el modelo se comporta como Lasso.
- Si α=0, se comporta como Ridge.
- Valores intermedios permiten encontrar un equilibrio entre ambos enfoques.

Ventajas y Desventajas

Ventajas de Lasso:

- Realiza selección automática de variables, simplificando el modelo.
- Mejora la interpretabilidad al reducir el número de predictores.
- Es útil en problemas con muchas variables, eliminando las menos relevantes.

Desventajas de Lasso:

- Si hay alta correlación entre variables predictoras, puede eliminar algunas de ellas aunque sean relevantes.
- No permite seleccionar más variables de las que hay observaciones, lo que puede limitar su uso en ciertos contextos.

Ventajas de Elastic Net:

- Maneja mejor la multicolinealidad, ya que no fuerza la eliminación completa de variables correlacionadas.
- Combina las ventajas de Lasso y Ridge, proporcionando una mayor flexibilidad.
- Permite seleccionar más variables que Lasso, lo que puede ser beneficioso en modelos con muchas características.

Desventajas de Elastic Net:

- Requiere ajustar dos parámetros de regularización ($\lambda\ y\ \alpha$), lo que aumenta la complejidad del ajuste del modelo.
- La interpretación del modelo puede ser más compleja en comparación con Lasso o Ridge por la combinación de penalizaciones.

Aplicaciones y Evaluación del Modelo

Estos métodos se utilizan en problemas de regresión donde hay muchas variables predictoras, especialmente en situaciones con datos altamente correlacionados o donde se busca un modelo parsimonioso. La evaluación del modelo se realiza a través de métricas como:

- Error Cuadrático Medio (MSE)
- Coeficiente de determinación R^2
- Validación cruzada para seleccionar el mejor valor de λ

Ambos modelos son herramientas fundamentales en el aprendizaje automático y la estadística aplicada, ayudando a mejorar la estabilidad y capacidad predictiva en entornos con datos complejos.

Árboles de clasificación

Los árboles de clasificación son modelos predictivos basados en una estructura jerárquica que divide el conjunto de datos en diferentes grupos o "nodos" de acuerdo con las características de las variables independientes. El objetivo es clasificar los datos en categorías o clases específicas basándose en la información disponible. A diferencia de los modelos lineales, los árboles de clasificación no requieren que las relaciones entre las variables sean lineales, lo que les permite manejar relaciones no lineales y complejas de manera eficaz.

Funcionamiento

El proceso de construcción de un árbol de clasificación implica la selección de una variable para dividir el conjunto de datos en nodos más pequeños, lo que permite la clasificación de la variable dependiente. Este proceso de división continúa recursivamente, creando un árbol con ramas que representan las decisiones tomadas en función de los valores de las variables predictoras. Cada nodo del árbol representa una "pregunta" sobre una característica de los datos, y las ramas que salen de este nodo corresponden a las respuestas posibles a esa pregunta.

El criterio utilizado para seleccionar las divisiones en cada nodo es fundamental para la calidad del modelo. Los criterios más comunes son el índice de Gini y la entropía, que miden la impureza de los nodos y buscan maximizar la homogeneidad dentro de cada uno. El índice de Gini, por ejemplo, mide la probabilidad de que un elemento seleccionado al azar de un nodo sea clasificado incorrectamente, mientras que la entropía evalúa la cantidad de incertidumbre en una división.

Utilidad

Los árboles de clasificación son especialmente útiles en una amplia variedad de situaciones debido a su capacidad para modelar relaciones complejas sin necesidad de transformaciones o suposiciones de linealidad. Son intuitivos y fáciles de interpretar, lo que los hace accesibles incluso para quienes no son expertos en análisis estadístico. Además, su capacidad para gestionar tanto variables numéricas como categóricas los convierte en una herramienta versátil.

Entre sus principales ventajas, destacan:

- **Interpretabilidad**: El modelo puede ser visualizado y entendido fácilmente, lo que permite identificar cómo se toman las decisiones.
- **No linealidad**: A diferencia de modelos como la regresión, los árboles no asumen ninguna relación lineal entre las variables.
- **Manejo de interacciones**: Pueden capturar interacciones complejas entre las variables sin necesidad de especificarlas explícitamente.

Sin embargo, también tienen limitaciones, como el riesgo de sobreajuste (overfitting), especialmente cuando el árbol es muy profundo. Para mitigar este problema, se pueden aplicar técnicas como la poda (pruning), que consiste en recortar ramas del árbol que no contribuyen significativamente al poder predictivo del modelo.

Teoría Detrás de los Árboles de Clasificación

El fundamento de los árboles de clasificación se basa en la teoría de división recursiva (recursive partitioning), que consiste en dividir repetidamente el conjunto de datos en subconjuntos más pequeños con el objetivo de mejorar la pureza de las clases dentro de cada subconjunto. La división se realiza eligiendo el mejor punto de corte para cada variable en cada paso, utilizando un criterio de impureza como el índice de Gini o la entropía.

La complejidad computacional de los árboles de clasificación varía según la cantidad de datos y la profundidad del árbol. Un árbol demasiado grande puede sobreajustarse a los datos de entrenamiento, mientras que un árbol muy pequeño puede ser insuficiente para capturar la complejidad de las relaciones entre las variables.

Conclusión

En resumen, los árboles de clasificación son una herramienta robusta y flexible para realizar tareas de clasificación en situaciones donde las relaciones entre las variables no son necesariamente lineales. Su capacidad para modelar interacciones complejas de forma intuitiva los convierte en un método ampliamente utilizado en problemas de clasificación en diversas áreas, desde la estadística hasta el aprendizaje automático y la inteligencia artificial.

Modelos Bagging y Random Forest

Bagging

Bagging (Bootstrap Aggregating) es un método de aprendizaje automático que tiene como objetivo mejorar la precisión y la estabilidad de los modelos predictivos. Bagging se basa en la creación de múltiples modelos a partir de subconjuntos aleatorios del conjunto de datos original, y luego combina sus predicciones para obtener una predicción final más robusta.

La idea principal de Bagging es reducir la varianza del modelo, lo que lo hace menos susceptible al sobreajuste (overfitting) que puede ocurrir cuando un modelo es demasiado complejo o se ajusta demasiado a los datos de entrenamiento. Este proceso se realiza mediante el siguiente esquema:

1. **Submuestreo aleatorio con reemplazo**: Se generan múltiples subconjuntos del conjunto de datos original mediante el muestreo aleatorio con reemplazo (bootstrap). Cada subconjunto tiene el mismo tamaño que el conjunto original, pero algunos ejemplos del conjunto original pueden repetirse mientras que otros pueden quedar fuera.
2. **Entrenamiento de modelos independientes**: Para cada subconjunto, se entrena un modelo independiente, que puede ser cualquier tipo de modelo, aunque comúnmente se utilizan árboles de decisión debido a su naturaleza no lineal y su capacidad para manejar datos complejos.
3. **Promedio o voto mayoritario**: Una vez entrenados los modelos, se combinan sus resultados. En problemas de regresión, se calcula el promedio de las predicciones de todos los modelos; en problemas de clasificación, se utiliza el voto mayoritario, donde la clase más frecuente entre las predicciones de los modelos es seleccionada como la predicción final.

El principal beneficio de Bagging es que, al promediar o votar las predicciones de múltiples modelos, el método reduce la varianza y mejora la generalización, lo que a su vez reduce el riesgo de sobreajuste.

Random Forest

Random Forest (RF) es una extensión del método de Bagging que utiliza árboles de decisión como base, pero con un enfoque adicional para mejorar aún más el rendimiento y la diversidad de los modelos individuales en el conjunto.

En RF, el proceso de entrenamiento sigue los pasos básicos de Bagging, pero con una modificación clave: en cada división del árbol se selecciona un subconjunto aleatorio de características, lo que introduce una mayor aleatoriedad y diversidad en los árboles individuales. Este paso evita que los árboles sean demasiado similares entre sí y mejora la capacidad del modelo para generalizar.

De forma más detallada:

1. **Construcción de árboles de decisión**: Al igual que en Bagging, se generan múltiples árboles de decisión a partir de diferentes subconjuntos de datos. Sin embargo, en cada nodo de cada árbol, en lugar de considerar todas las características disponibles, solo se elige un subconjunto aleatorio de características para realizar la división. Esto garantiza que los árboles sean más diversos y no estén correlacionados entre sí.
2. **Agregación de resultados**: Al igual que en Bagging, se utilizan el promedio o el voto mayoritario para combinar las predicciones de todos los árboles en el bosque. En regresión, se promedian las predicciones, mientras que en clasificación se selecciona la clase más común entre los árboles.
3. **Reducción de correlación**: El enfoque aleatorio en la selección de características durante la construcción de los árboles reduce la correlación entre ellos, lo que mejora la precisión global del modelo y evita el sobreajuste.

Teoría Detrás de RF y Bagging

La base teórica de Bagging y RF se encuentra en el concepto de ensambles de modelos, es decir, la combinación de múltiples modelos individuales para formar un modelo conjunto que tiene un rendimiento superior al de cualquier modelo individual. Esto se fundamenta en el principio de que, aunque los modelos individuales puedan cometer errores, la combinación de varios modelos independientes tiende a mitigar esos errores y a producir resultados más precisos.

- **Reducción de la varianza**: En Bagging, la principal ventaja es la reducción de la varianza del modelo. Al entrenar múltiples modelos con subconjuntos diferentes de los datos y promediar sus resultados, se reduce el impacto de las fluctuaciones en los datos de entrenamiento y se obtiene un modelo más estable y robusto.
- **Mejora de la precisión**: En RF, la introducción de aleatoriedad en la selección de características (además del muestreo de datos) mejora la precisión al reducir la correlación entre los árboles y al forzar que los árboles aprendan de diferentes perspectivas del conjunto de datos.

Ventajas y Desventajas

Ventajas:

- **Mejora de la precisión**: Ambos métodos, Bagging y RF, tienden a generar modelos con una mayor precisión que los modelos individuales debido a la agregación de múltiples predicciones.
- **Reducción del sobreajuste**: Especialmente en RF, el modelo es menos propenso al sobreajuste debido a la introducción de aleatoriedad en la selección de características.
- **Manejo de datos complejos**: Pueden manejar variables tanto numéricas como categóricas y capturan relaciones no lineales entre las variables.

Desventajas:

- **Complejidad computacional**: La necesidad de construir y almacenar múltiples modelos hace que los métodos de Bagging y RF sean más costosos en términos de tiempo de cómputo y memoria.
- **Interpretación limitada**: Aunque RF y Bagging proporcionan buenos resultados, la interpretación del modelo es más difícil que la de un solo árbol de decisión, ya que se trata de un conjunto de árboles.

Conclusión

En resumen, los modelos Bagging y RF son enfoques de ensamble que mejoran la precisión y la estabilidad de los modelos predictivos al combinar múltiples modelos entrenados de manera independiente. Mientras que Bagging utiliza un enfoque más simple, RF introduce una mayor aleatoriedad durante la construcción de los árboles, lo que mejora aún más la capacidad de generalización del modelo. Ambos métodos son ampliamente utilizados en problemas de clasificación y regresión debido a su robustez y capacidad para manejar datos complejos.

Metodología estadística

Descripción de los datos

Fuente de datos

Los datos utilizados en este estudio provienen de la investigación titulada *"Videojuegos y jóvenes: lugares, experiencias y tensiones", sacada de Gómez Miguel, A. y Calderón Gómez, D. (2023). Videojuegos y jóvenes: lugares, experiencias y tensiones. Madrid: Centro Reina Sofía de Fad Juventud. DOI: 10.5281/zenodo.7970990*. Se trata de un estudio de ámbito nacional dirigido a adolescentes y jóvenes de entre 15 y 29 años residentes en España. La información fue recogida mediante una encuesta online con cuestionario cerrado, utilizando un panel online para su distribución.

El trabajo de campo se llevó a cabo en noviembre de 2022, y los datos han sido ponderados según el nivel de estudios para garantizar su representatividad.

Variables estudiadas

En el Anexo 1 se presenta una tabla con la descripción detallada de las variables analizadas en este estudio.

Población y muestra

La población objetivo del estudio son adolescentes y jóvenes de 15 a 29 años que residen en España. Se utilizó un muestreo estratificado por afijación proporcional en función del género y la edad.

El tamaño de la muestra es de 1.513 personas, con selección aleatoria de casos en base a cuotas de género y grupos de edad. La distribución final de la muestra fue ponderada según nivel educativo para asegurar una mayor representatividad.

Bajo el supuesto de muestreo aleatorio simple y máxima heterogeneidad (p=q=0,5), el margen de error del estudio es de ±2,52%, con un nivel de confianza del 95%.

Proceso de análisis

Depuración de la base de datos

Todo el proceso de análisis se ha realizado utilizando el lenguaje de programación R [7][8]. Para las tareas de limpieza de datos y análisis exploratorio, se ha seguido el enfoque práctico propuesto por Calviño Martínez y Alonso Revenga [9], que resulta especialmente útil para aplicar técnicas de depuración y visualización en el contexto del análisis de datos con R.

En primer lugar, se han seleccionado únicamente las preguntas que han sido respondidas por la totalidad de los participantes. Esto se debe a que algunas preguntas solo se plantean a ciertos grupos, lo que genera valores en blanco que no representan datos faltantes reales, sino la ausencia de aplicación de la pregunta. Para evitar este problema y asegurar un análisis coherente, se han elegido las siguientes variables:

- Id, Jugar, Grupo de edad, Sexo, Estudios, Veo directos, Veo videos, Leo prensa, Sigo redes, Eventos presenciales, Veo eSports, Creo contenido y Rasgos principales.

La variable Rasgos principales se ha creado a partir de la variable original Rasgos, que inicialmente contenía 17 categorías distintas. Para simplificar el análisis, estas categorías han sido agrupadas en cinco grandes grupos. Posteriormente, estas nuevas categorías se han transformado en variables *dummy*, donde cada variable toma el valor 1 si el participante posee el rasgo asociado y 0 en caso contrario.

Para realizar una primera exploración de los datos y calcular los estadísticos descriptivos, todas estas variables han sido convertidas en factores en R. Cabe destacar que todas las variables seleccionadas son cualitativas: algunas son binarias (con valores 0 y 1), mientras que otras son categóricas con múltiples niveles.

Por último, se han eliminado dos variables:

1. Edad: Se trata de una variable numérica, pero ya contamos con su versión categórica (Grupo de edad), lo que facilita el análisis y la interpretación de los datos.
2. Rasgos principales: Dado que esta variable ha sido transformada en cinco variables *dummy*, su presencia en la base de datos deja de ser necesaria.

Esta depuración permite trabajar con una base de datos más estructurada y adaptada a los objetivos del estudio.

El resumen de nuestras variables es este, vemos como se distribuyen nuestras variables respecto a sus categorías, cuantas observaciones hay en cada una, la frecuencia relativa y la frecuencia acumulada:

Tabla 1 Frecuencia y Distribución de las Variables del Estudio

Variable	Categorías	Frecuencia Absoluta (n)	Frecuencia Relativa (%)	Frecuencia Acumulada (%cum)
Jugar	No	326	21.5	21.5
	Sí	1187	78.5	100.0
Grupo_edad	De 15 a 19	430	28.4	28.4
	De 20 a 24	533	35.2	63.6
	De 25 a 29	550	36.4	100.0
Sexo	Hombres	764	50.5	50.5
	Mujeres	749	49.5	100.0
Estudios	Hasta secundarios obligatorios	133	8.8	8.8
	Secundarios posobligatorios	810	53.5	62.3
	Superiores	570	37.7	100.0
Veo_directos	Nunca	382	25.2	25.2
	Con menor frecuencia	305	20.2	45.4
	Al menos 1 vez cada 15 días	244	16.1	61.5
	Al menos 1 vez por semana	337	22.3	83.8
	Todos los días	245	16.2	100.0
Veo_videos	Nunca	272	18.0	18.0
	Con menor frecuencia	330	21.8	39.8
	Al menos 1 vez cada 15 días	300	19.8	59.6
	Al menos 1 vez por semana	352	23.3	82.9
	Todos los días	259	17.1	100.0
Leo_prensa	Nunca	477	31.5	31.5

	Con menor frecuencia	357	23.6	55.1
	Al menos 1 vez cada 15 días	300	19.8	74.9
	Al menos 1 vez por semana	240	15.9	90.8
	Todos los días	139	9.2	100.0
Sigo_redes	Nunca	295	19.5	19.5
	Con menor frecuencia	317	21.0	40.5
	Al menos 1 vez cada 15 días	283	18.7	59.2
	Al menos 1 vez por semana	303	20.0	79.2
	Todos los días	315	20.8	100.0
Eventos_presenciales	Nunca	672	44.4	44.4
	Con menor frecuencia	461	30.5	74.9
	Al menos 1 vez cada 15 días	177	11.7	86.6
	Al menos 1 vez por semana	128	8.5	95.1
	Todos los días	75	5.0	100.0
Veo_eSports	Nunca	458	30.3	30.3
	Con menor frecuencia	367	24.3	54.6
	Al menos 1 vez cada 15 días	268	17.7	72.3
	Al menos 1 vez por semana	288	19.0	91.3
	Todos los días	132	8.7	100.0
Creo_contenido	Nunca	686	45.3	45.3

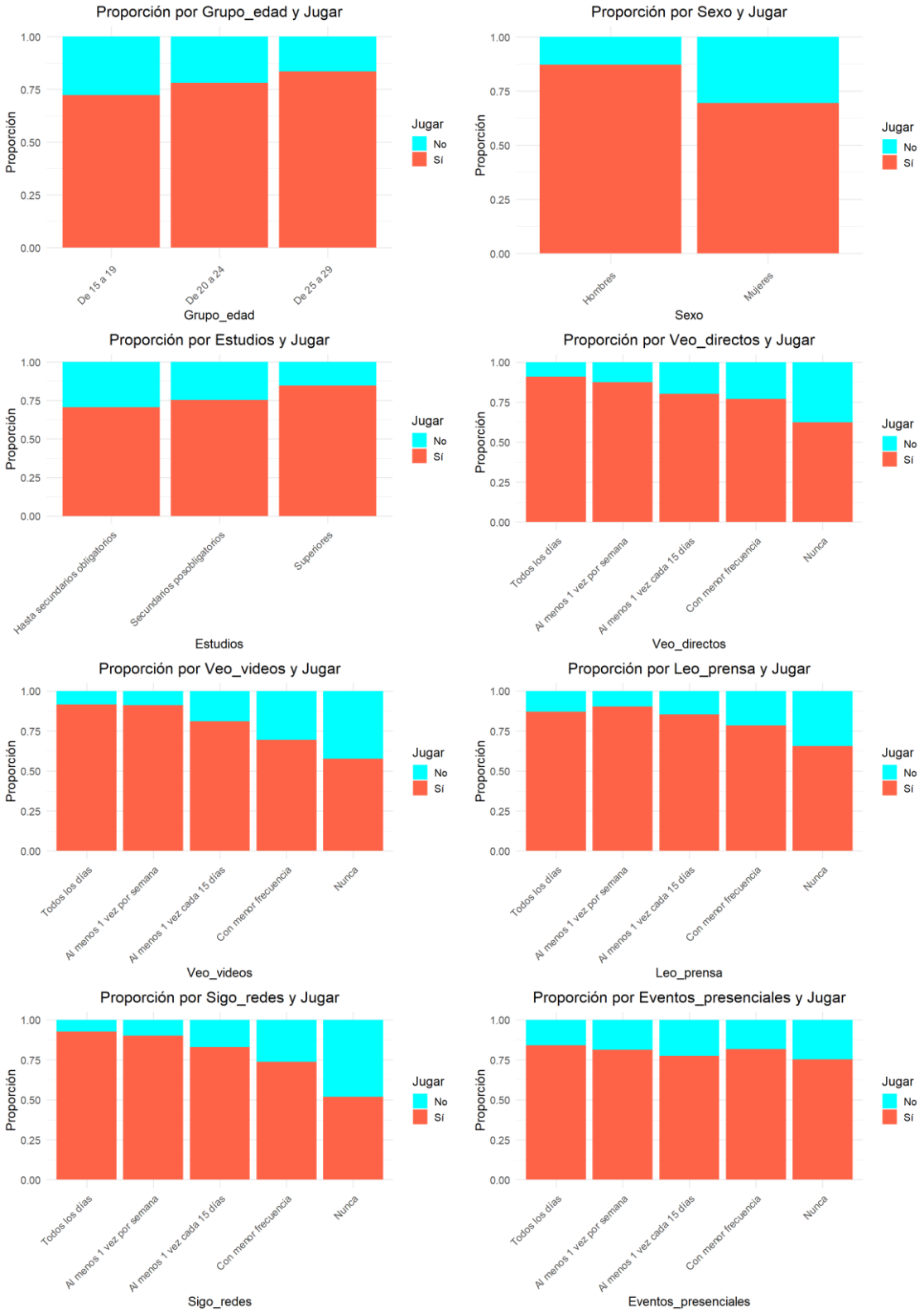
	Con menor frecuencia	251	16.6	61.9
	Al menos 1 vez cada 15 días	197	13.0	74.9
	Al menos 1 vez por semana	164	10.8	85.7
	Todos los días	215	14.2	100.0
Pasion_videojuegos	No	1065	70.4	70.4
	Sí	448	29.6	100.0
Equipamiento_habilidad	No	961	63.5	63.5
	Sí	552	36.5	100.0
Socializacion_comunidad	No	1245	82.3	82.3
	Sí	268	17.7	100.0
Impacto_personal_salud	No	1386	91.6	91.6
	Sí	127	8.4	100.0
Identidad_cultura_gamer	No	1395	92.2	92.2
	Sí	118	7.8	100.0

Estas son las frecuencias de las categorías de nuestras variables categóricas, como sabemos necesitamos que todas las categorías estén bien representadas, es decir, que tengan un porcentaje superior al 2-5%, como vemos en nuestros datos todas están bien representadas menos las categorías no sabe no contesta (Ns/Nc) en la que no hay ninguna respuesta, por tanto vamos a eliminar estas categorías de nuestras variables.

Hacemos un análisis de las relaciones entre variables, vamos a ver las relaciones bidimensionales entre Jugar que es nuestra variable objetivo y las demás variables, vamos a hacer un gráfico de proporciones para que se encuentren a la misma altura todas las categorías y poder ver bien como están representadas.

Como se puede ver las variables que aportan menos información son: eventos_presenciales, creo_contenido, pasión_videojuegos, equipamiento_habilidad e identidad_cultura_gamer, ya que en todas sus categorías existe el mismo reparto de frecuencias.

Mientras que las variables que sí que aportan más información útil son: grupo_edad, sexo, estudios, veo_directos, veo_videos, leo_prensa, sigo_redes, veo_eSports, socialización_comunidad e impacto_personal_salud, ya que sus categorías tienen distintos repartos de frecuencias para nuestra variable objetivo.


Proporción por Grupo_edad y Jugar
Proporción
1.00
0.75
0.50
0.25
0.00
De 15 a 19
De 20 a 24
De 25 a 29
Grupo_edad
Jugar
No
Sí
Proporción por Sexo y Jugar
Hombres
Mujeres
Sexo
Proporción por Estudios y Jugar
Hasta secundarios obligatorios
Secundarios posobligatorios
Superiores
Estudios
Proporción por Veo_directos y Jugar
Todos los días
Al menos 1 vez por semana
Al menos 1 vez cada 15 días
Con menor frecuencia
Nunca
Veo_directos
Proporción por Veo_videos y Jugar
Veo_videos
Proporción por Leo_prensa y Jugar
Leo_prensa
Proporción por Sigo_redes y Jugar
Sigo_redes
Proporción por Eventos_presenciales y Jugar
Eventos_presenciales

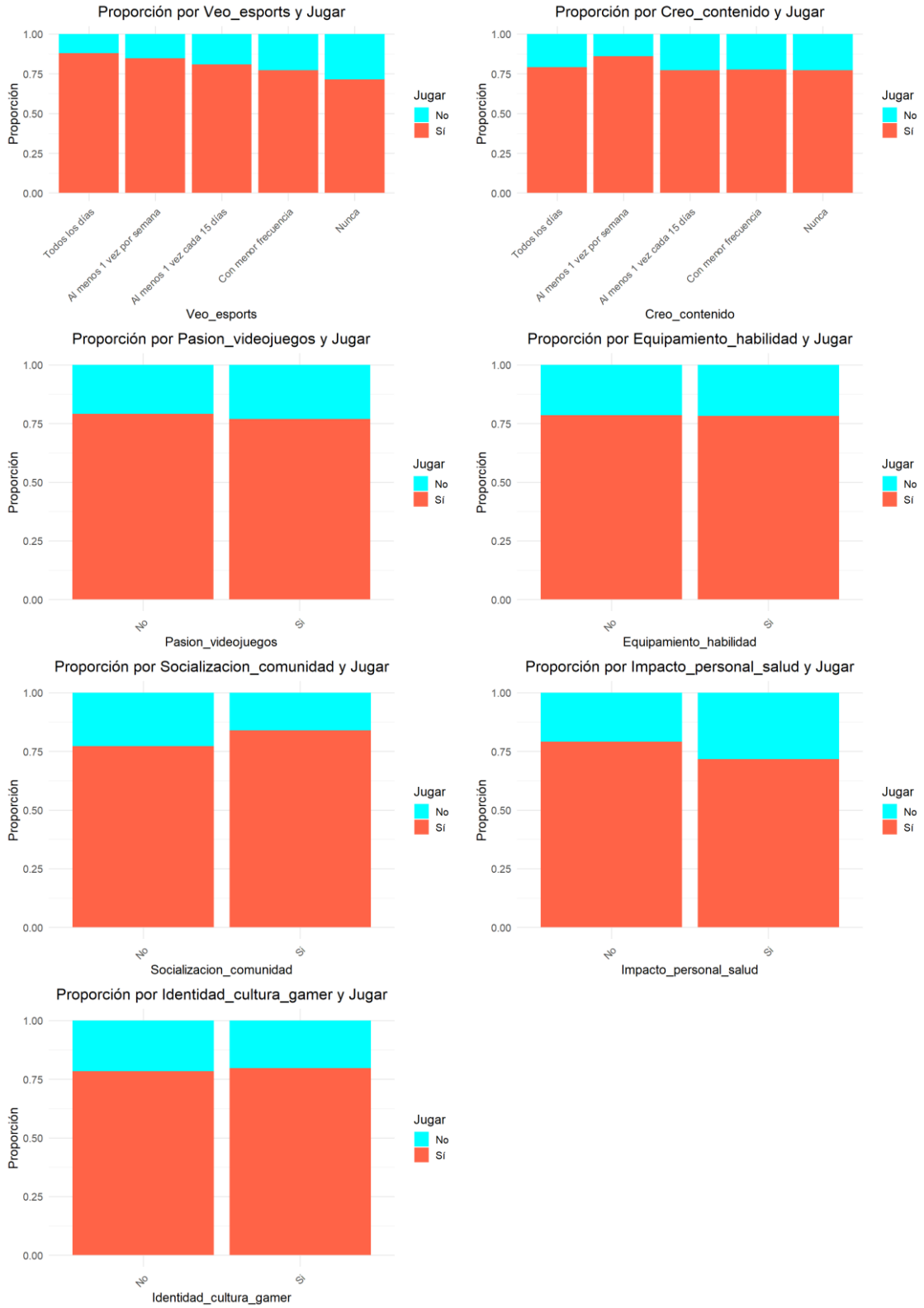


Ilustración 1 Relación entre la Variable Objetivo y las Variables del Estudio

A pesar de que la información gráfica aporta mucha información, esta puede ser subjetiva, por eso vamos a calcular alguna medida como el V de Cramer para poder cuantificar estas relaciones. Este estadístico está relacionado con el chi-cuadrado, pero tiene la ventaja de que está acotado entre 0 y 1, tomando el valor 0 cuando las variables son independientes y el valor 1 cuando son totalmente dependientes.

Para el siguiente gráfico, he creado además dos variables aleatorias que también analizaremos. Estas variables se generan con una distribución uniforme en el intervalo [0,1] y no contienen ninguna relación real con la variable objetivo. Lo hacemos para evaluar si hay alguna variable en el conjunto de datos que tenga menos poder predictivo que estas, lo que indicaría que es poco útil para el análisis. Si encontramos variables con una relación más débil que estas aleatorias, deberíamos considerar eliminarlas.

Gráfico V de Cramer:

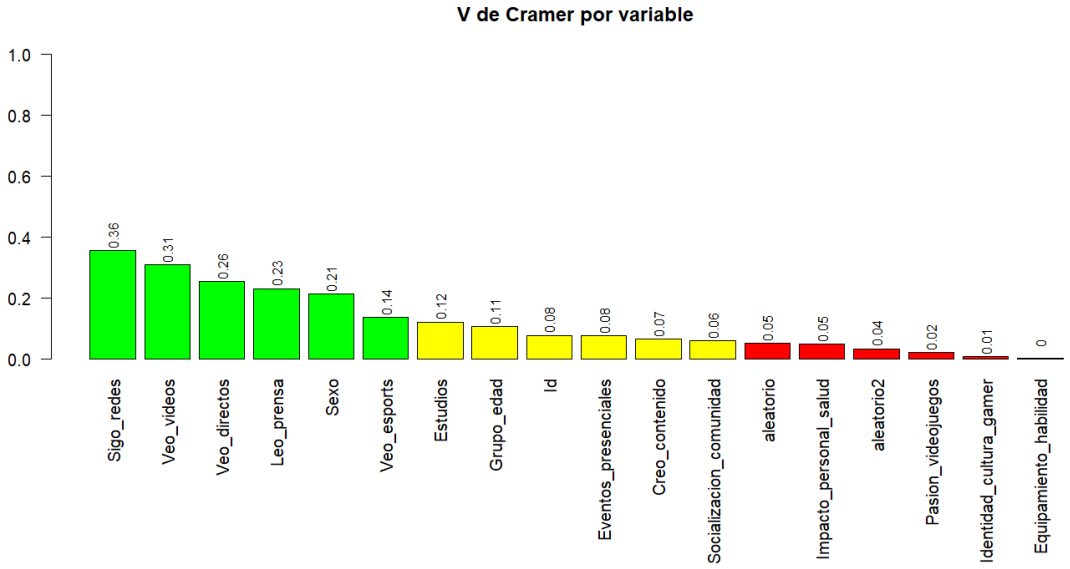


Ilustración 2 Gráfico V de Cramer

En verde vemos las variables más relacionadas con la variable Jugar y teniendo en cuenta estos resultados las variables menos relacionadas con Jugar son Impacto_personal_salud, Pasión_videojuegos, Identidad_cultura_gamer y Equipamiento_habilidad, pues son las que menor importancia tienen, representadas en rojo, teniendo un valor similar o inferior al de las aleatorias. Este gráfico lo utilizamos de forma descriptiva para analizar la importancia de las variables en un contexto univariante. Sin embargo, cabe destacar que el hecho de que una variable no sea relevante en un análisis univariante no implica necesariamente que carezca de importancia en un análisis multivariante, ya que podría aportar información relevante al interactuar con otras variables, contribuir a la agrupación de categorías, o capturar aspectos que no se observan en un análisis individual. Además, en este caso, dado que no trabajamos con un conjunto de datos de alta dimensionalidad, no se ha aplicado un filtrado estricto basado en estos resultados.

Aplicación del MDS para variables:

En este caso obtenemos el MDS utilizando como medida de disimilaridad el estadístico V de Cramer, toma un valor del stress que puede considerarse suficiente con una calidad baja según el *gold standard*, vemos que el proceso ha convergido en 15 iteraciones. El resultado del MDS depende del valor inicial aplicado en el proceso de optimización y, aunque la función mds() tiene implementado un método por el cuál la elección de los valores iniciales del proceso de optimización suele dar buenos resultados, esto no es siempre así. Por ello, resulta de utilidad llevar a cabo dicho

proceso de optimización varias veces con inicios aleatorios y verificar si se ha obtenido el mejor valor posible.

A la vista del resultado, aunque se puede observar una ligera disminución en el stress (pasa de 0.168 a 0.166), podemos concluir que la solución inicial es suficientemente buena por lo que la tomamos como solución “óptima”. Esto es así debido a que, como ya se ha explicado, la función está diseñada para utilizar valores iniciales razonables y resulta aconsejable utilizar la solución que proporcionan si ésta no es significativamente peor.

Antes de proceder con la visualización de las variables en el plano MDS, procedemos a analizar la aportación de las variables al stress para determinar si existe alguna que esté peor representada que el resto, para lo que recurrimos al gráfico denominado stressplot. Nótese que el stress se obtiene a partir del sumatorio de todos los pares de observaciones por lo que es posible determinar cuál ha sido la aportación al mismo de cada variable. El gráfico siguiente muestra estas aportaciones en porcentajes, por lo que los porcentajes altos indicarán que dichas observaciones están peor representadas:

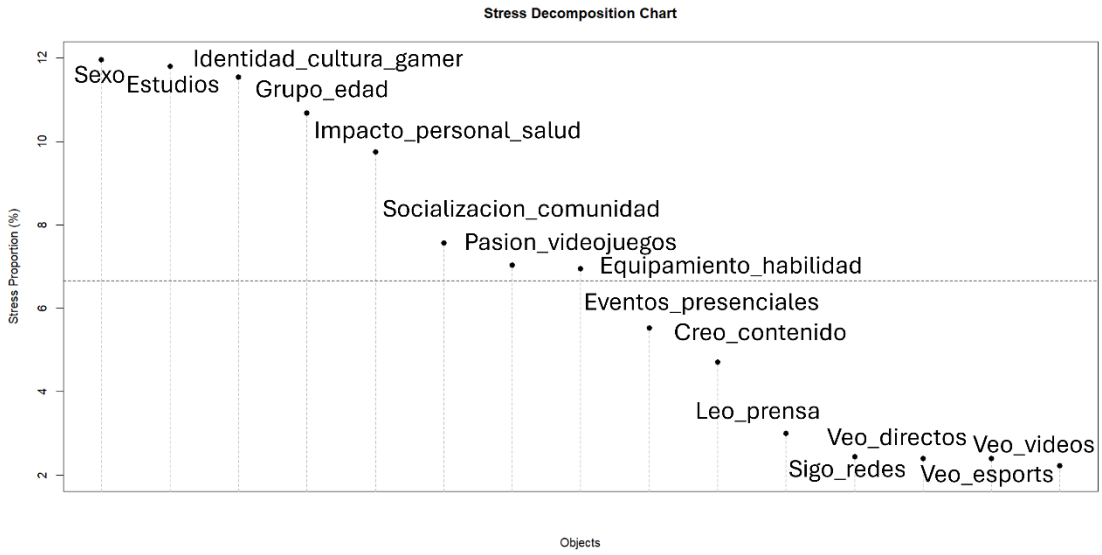


Ilustración 3 Aportación de las variables al Stress

Dado que contamos con 15 variables, si todas estuvieran igual de bien/mal representadas, sus aportaciones deberían rondar el 6% (es lo que representa la línea horizontal). No obstante, podemos ver que no es así en este caso y que variables como, Sexo, Estudios, Identidad_cultura_gamer, Grupo_edad e Impacto_personal_salud son las que peor representadas están, por lo que sus distancias pueden no representar de una manera tan fidedigna las disimilaridades originales. En el otro extremo, observamos Veo_videos, Veo_esports, Veo_directos, Sigo_redes, Leo_prensa y por lo que las distancias observadas en el gráfico partiendo de estas variables representarán muy bien las disimilaridades.

Para finalizar, vamos a obtener la representación gráfica y a extraer conclusiones sobre el parecido/diferencia de las variables entre sí:

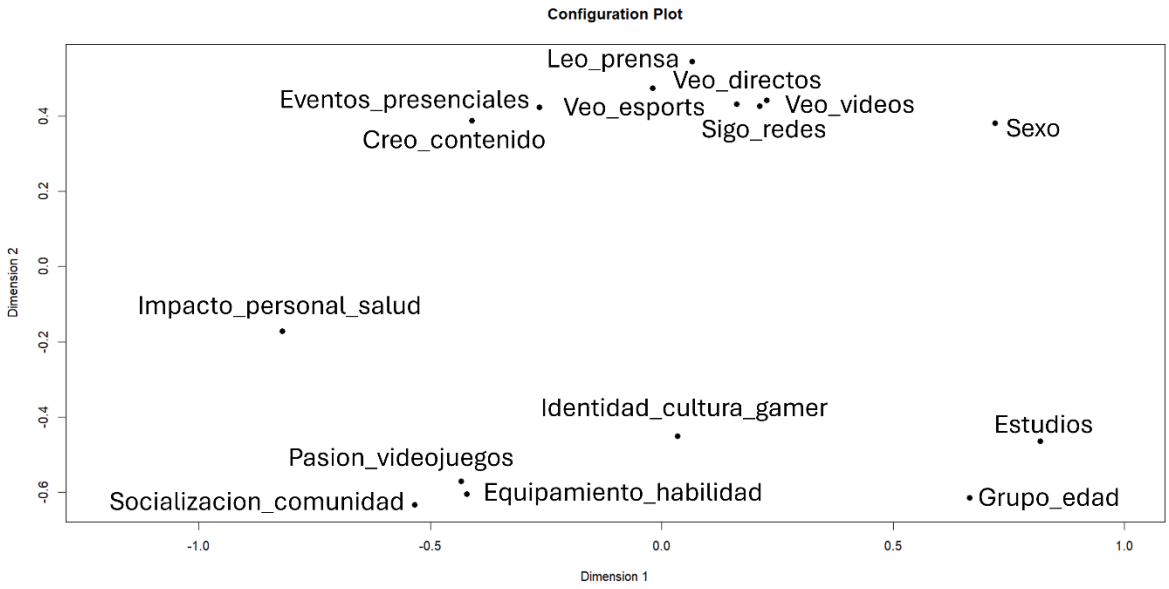


Ilustración 4 Similitudes entre variables

Como se puede comprobar con el grafico algunas variables aparecen próximas como Veo_videos, Veo_esports, Veo_directos, Sigo_redes y Leo_prensa, otro grupo es Pasión_videojuegos, Equipamiento_habilidad y Socialización_comunidad lo que indica una relación lineal directa entre estas variables o patrones similares, en el caso opuesto observamos variables muy alejadas entre si como Socialización_comunidad y Leo_prensa o Equipamiento_habilidad y Veo_esports, normalmente esto representa una relación lineal inversa o patrones diferentes.

Clúster de variables:

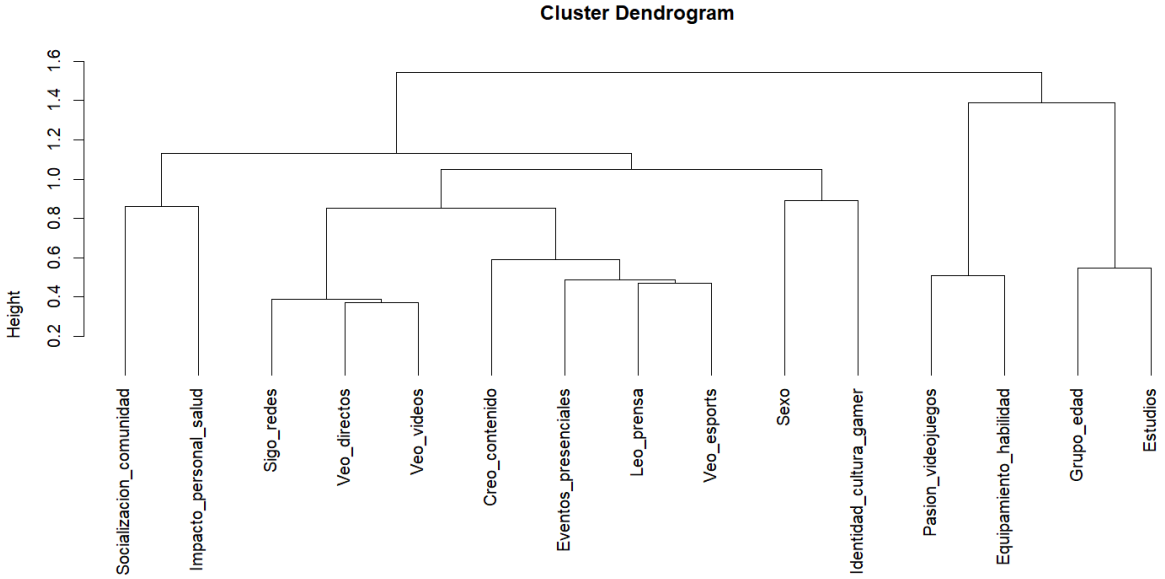


Ilustración 5 Clúster de variables

Como se puede observar, en el proceso de agrupación los primeros pares de variables son Veo_directos y Veo_videos (los cuales se muestran próximos en el MDS), Leo_prensa y Veo_esports y pasión_videojuegos y Equipamiento_habilidad. Se observan tres grandes agrupaciones en el dendrograma aunque puede que sean más, que pasamos a representar gráficamente mediante colores:

Ilustración 6 Similitudes de variables por clúster inicial

Como se puede observar, el clúster “verde” está formado por observaciones próximas y lejanas entre sí por lo que no parece una agrupación muy lógica. Esto se debe a que el número de clústeres seleccionado es menor de lo que debería. Para determinar el número óptimo de grupos se puede recurrir al estudio de la estabilidad, que se realiza obteniendo varias muestras bootstrap del conjunto de datos, aplicando el proceso de clúster jerárquico y calculando el índice de Rand ajustado para cada una de las agrupaciones obtenidas. Nótese que no se aplica el índice de Rand estudiado en la teoría sino una modificación de este. No obstante, esencialmente miden lo mismo por lo que también ofrece valores próximos a 1 cuando las agrupaciones coinciden y valores próximos a 0 (incluso negativos), en caso contrario. La lógica detrás de este estudio es que la agrupación más natural de los datos será aquella más estable (a la que menos le afecten las perturbaciones de los datos).

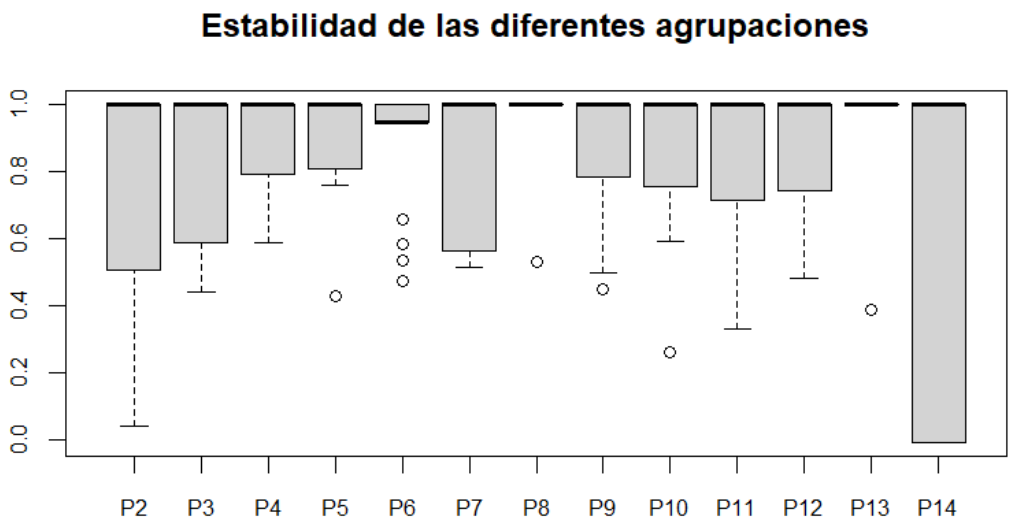


Ilustración 7 Estabilidad de las diferentes agrupaciones

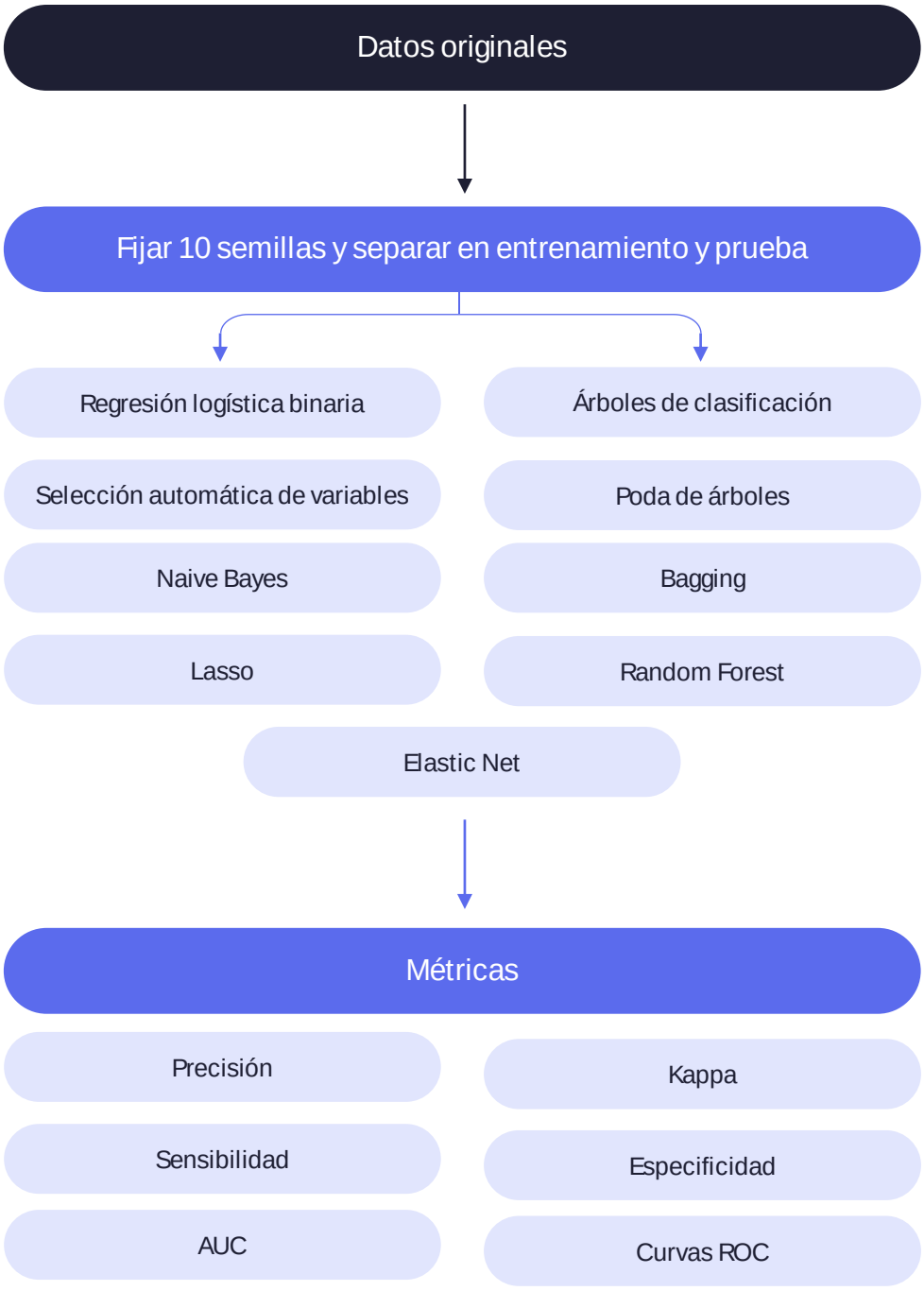
El gráfico anterior nos muestra que ninguna de las agrupaciones es evidente pues el índice de Rand toma valores muy variables. Nos da agrupaciones desde dos hasta 14. Si analizamos el resto de los valores, parece que 4 grupos sería buena opción pues muestra valores altos de Q2 y Q3, por lo que se considerará como el número óptimo (cabe destacar que 5 grupos también sería una opción pero, como no ofrece una gran mejora, se opta por la agrupación más sencilla).

Ilustración 8 Similitudes de variables por clúster definitivo

Tal y como se puede observar, las agrupaciones obtenidas concuerdan bastante con los resultados gráficos del MDS. Contamos con un grupo muy grande con Veo_videos, Veo_esports, Veo_directos, Sigo_redes, Leo_prensa, Eventos_presenciales, Creo_contenido, Identidad_cultural_gamer y Sexo, otro formado por Impacto_personal_salud y Socialización_comunidad, otro formado por Pasión_videojuegos y Equipamiento_habilidad y el ultimo formado por Grupo_edad y

Estudios, por lo que el conjunto de variables de cada agrupación estará relacionado (pero recordemos que el tipo de relación no tiene por qué ser lineal ni directo). Puede sorprender la agrupación “verde” pues es la más grande y contiene “Identidad_cultural_gamer” que se muestra alejada en el plano MDS. En este caso es importante destacar que esta variable es de las peores representadas en el MDS, por lo que su ubicación en el plano no recoge bien su relación con el resto de las variables. Si hacemos un grupo más se separarían del grupo más grande Sexo e Identidad_cultural_gamer que formarían el quinto grupo.

Modelos



Modelo de regresión logística binario

Lo primero que haremos será reorganizar el orden de las categorías en aquellas variables que contienen cinco posibles respuestas, que van desde "todos los días" hasta "nunca". Invertiremos este orden para facilitar la interpretación de los odds-ratio al calcularlos posteriormente.

Observamos que el 78% de los encuestados juegan. Si aplicáramos un modelo básico en el que asumiéramos que todas las observaciones corresponden a personas que juegan, obtendríamos una tasa de acierto del 78%. Sin embargo, este modelo presentaría una especificidad del 0%, ya que no sería capaz de identificar correctamente los "no eventos".

Antes de construir los modelos de regresión logística, realizaremos una partición de los datos en conjuntos de entrenamiento y prueba, repitiendo este proceso 10 veces. Esto nos permitirá evaluar el modelo en distintos subconjuntos de datos. Para ello, utilizamos la función createDataPartition, asignando el 80% de los datos al conjunto de entrenamiento y el 20% restante al de prueba. Dado que se trata de un proceso aleatorio, es recomendable fijar una semilla para garantizar la reproducibilidad de los resultados; en este caso, establecemos 10 semillas diferentes.

A continuación, construimos un primer modelo de regresión logística binaria utilizando todas las variables disponibles. Este modelo cuenta con 38 parámetros, de los cuales 14 resultan significativos al nivel del 5%. Sin embargo, al analizar la tabla ANOVA, observamos que no todas las variables son relevantes y que, además, las variables binarias no aportan suficiente información debido a su escasa variabilidad.

Para mejorar el modelo, seleccionamos únicamente las seis variables que resultan significativas al 5%: Grupo_edad, Sexo, Estudios, Veo_videos, Sigo_redes y Eventos_presenciales (esta última se encuentra en el límite de significatividad, pero la incluimos para su evaluación).

Este segundo modelo tiene 18 parámetros, de los cuales 12 son significativos al nivel del 5%, y ahora todas las variables incluidas en el modelo también lo son. Finalmente, calculamos los odds-ratio de los efectos del modelo, lo que nos permitirá interpretar sus parámetros y, en otras palabras, cuantificar el impacto de las variables independientes sobre la variable dependiente.

Tabla 2 Modelo regresión logística binario odds ratio

Variable	**Odds_Ratio**	**P_Valor**	**Significación**
(Intercept)	0,976	0,836	
Grupo_edadDe 20 a 24	1,304	0,221	
Grupo_edadDe 25 a 29	1,683	0,027	*
SexoMujeres	0,474	0,000	***
EstudiosSecundarios posobligatorios	1,146	0,650	
EstudiosSuperiores	2,009	0,023	*
Veo_videosCon menor frecuencia	1,066	0,684	
Veo_videosAl menos 1 vez cada 15 días	1,865	0,033	*
Veo_videosAl menos 1 vez por semana	3,160	0,000	***
Veo_videosTodos los días	3,002	0,003	**

Sigo_redesCon menor frecuencia	2,488	0,000	***
Sigo_redesAl menos 1 vez cada 15 días	3,599	0,000	***
Sigo_redesAl menos 1 vez por semana	5,312	0,000	***
Sigo_redesTodos los días	6,867	0,000	***
Eventos_presencialesCon menor frecuencia	0,683	0,070	.
Eventos_presencialesAl menos 1 vez cada 15 días	0,474	0,009	**
Eventos_presencialesAl menos 1 vez por semana	0,516	0,043	*
Eventos_presencialesTodos los días	0,448	0,096	.

 # Grupo_edadDe 25 a 29: aumenta un 68% la posibilidad de jugar al pasar de grupo de edad de 15 a 19 a 25 a 29

 # SexoMujeres: disminuye un 53% la posibilidad de jugar al pasar de hombres a mujeres

 # EstudiosSuperiores: aumenta un 100% la posibilidad de jugar al pasar de hasta secundarios obligatorios a superiores

 # Veo_videosAl menos 1 vez cada 15 días: aumenta un 87% la posibilidad de jugar al pasar de nunca a al menos 1 vez cada 15 días

 # Veo_videosAl menos 1 vez por semana: aumenta un 216% la posibilidad de jugar al pasar de nunca a al menos 1 vez por semana

 # Veo_videosTodos los días: aumenta un 200% la posibilidad de jugar al pasar de nunca a todos los días

 # Sigo_redesCon menor frecuencia: aumenta un 149% la posibilidad de jugar al pasar de nunca a con menor frecuencia

 # Sigo_redesAl menos 1 vez cada 15 días: aumenta un 260% la posibilidad de jugar al pasar de nunca a al menos 1 vez cada 15 días

 # Sigo_redesAl menos 1 vez por semana: aumenta un 431% la posibilidad de jugar al pasar de nunca a al menos 1 vez por semana

 # Sigo_redesTodos los días: aumenta un 587% la posibilidad de jugar al pasar de nunca a todos los días

 # Eventos_presencialesAl menos 1 vez cada 15 días: disminuye un 53% la posibilidad de jugar al pasar de nunca a al menos 1 vez cada 15 días

 # Eventos_presencialesAl menos 1 vez por semana: disminuye un 48% la posibilidad de jugar al pasar de nunca a al menos 1 vez por semana

Evaluación del modelo

Una vez entendido el modelo y visto que es significativo, resta hacer una evaluación de este a partir de las distintas medidas estudiadas. Para ello recurrimos a la función confusionMatrix de la librería caret, la cual requiere que le facilitemos las predicciones dadas por el modelo (para lo cual obtenemos las probabilidades predichas y, a continuación, consideramos eventos aquellas observaciones con probabilidades por encima de 0.5), así como los valores reales.

Tabla 3 Modelo regresión logística binario medidas entrenamiento punto de corte 0,5

Métrica	**Media**
Accuracy	0,807
Kappa	0,292
Sensitivity	0,950
Specificity	0,285

El modelo anterior consigue clasificar correctamente casi el 81% de los datos y da lugar a un índice de Kappa de 0.29 lo que implica una calidad pobre. Adicionalmente, podemos ver como el modelo consigue detectar mejor los “eventos” que los “no eventos” pues sensibilidad y especificidad toman valores dispares. Cuando la variable dependiente está desbalanceada (reparto de categorías muy diferente de 50-50) es habitual que ocurra lo anterior: tasa de acierto alta, kappa medio y sensibilidad y especificidad dispares. Dado que en esos casos hay una categoría que es poco frecuente, puede resultar excesivo exigirle al modelo que las probabilidades predichas superen el punto de corte de 0,5. En esos casos, usar como punto de corte la proporción de eventos en el conjunto de datos suele dar buenos resultados:

Tabla 4 Modelo regresión logística binario medidas entrenamiento punto de corte 0,78

Métrica	**Media**
Accuracy	0,715
Kappa	0,336
Sensitivity	0,717
Specificity	0,709
AUC	0,788

Como era de esperar, algunas de las observaciones han pasado de ser predichas como 1 a ser predichas como 0, lo que ha implicado un aumento significativo de la especificidad del 29% al 71%, sin que el resto de los indicadores se haya modificado significativamente, la precisión ha bajado un 9%, el kappa ha mejorado un 5% y lo que más se ha visto afectado es la sensibilidad pasando de un 95% a un 72%.

Por último, una vez comprobada empíricamente la gran influencia que tiene el punto de corte en algunas medidas de evaluación (nótese que el índice Kappa no es tan sensible), pasamos a obtener la curva ROC (y su área) para contar con otra herramienta de evaluación de modelos.

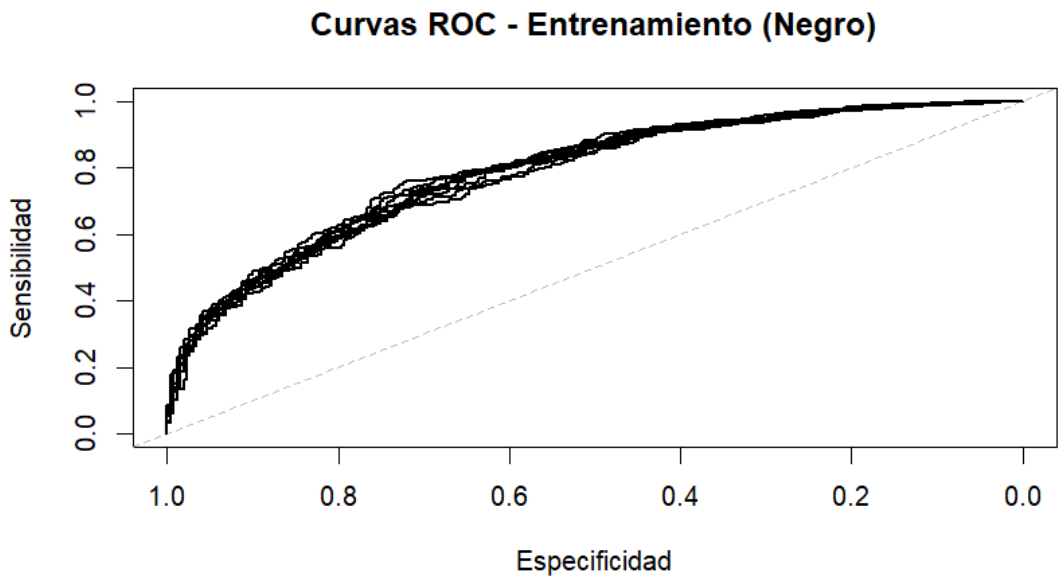


Ilustración 9 Modelo regresión logística binario curva ROC entrenamiento

Observamos, de nuevo, que el modelo construido es de buena calidad, pues el valor de su AUC es de 0,79. Dado que evaluar los modelos con la información de los datos de entrenamiento puede dar lugar a conclusiones excesivamente optimistas, es recomendable recurrir a la partición de prueba para obtener estimaciones más realistas de las medidas de evaluación.

Tabla 5 Modelo regresión logística binario medidas prueba punto de corte 0,78

Métrica	**Media**
Accuracy	0,713
Kappa	0,325
Sensitivity	0,719
Specificity	0,689
AUC	0,772

Como se puede observar, todos los indicadores de calidad han disminuido ligeramente, lo que es esperable. No obstante, el descenso no es excesivo, lo que es indicativo de que el modelo es relativamente estable. Obtenemos a continuación la curva ROC y el área (AUC). Se ha optado por representar conjuntamente las curvas en entrenamiento (negro) y en prueba (rojo) para facilitar la comparación.

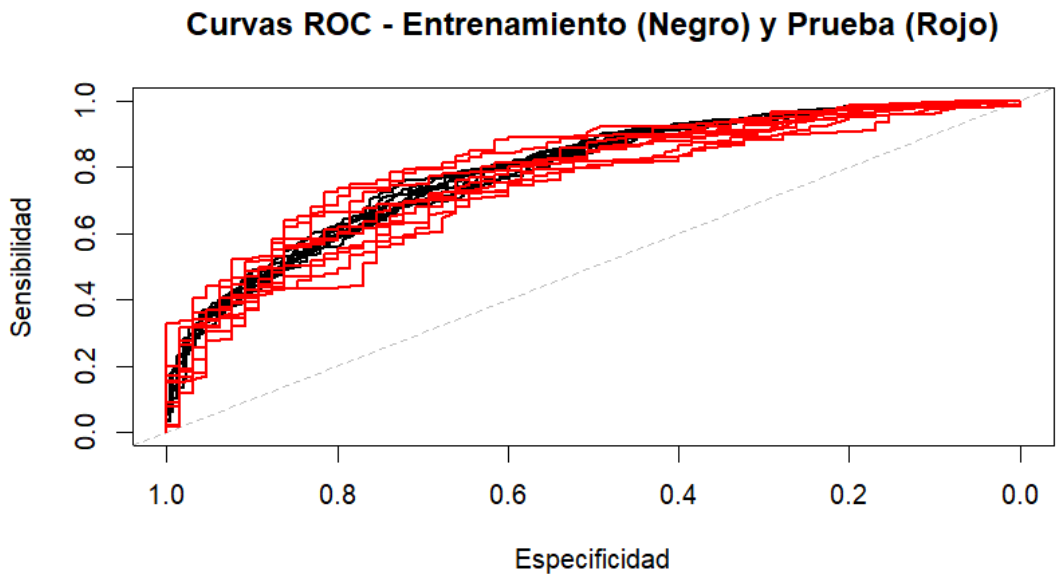


Ilustración 10 Modelo regresión logística binario curva ROC entrenamiento y prueba

Tal y como ha ocurrido con los anteriores indicadores, el AUC ha disminuido ligeramente al obtenerlo sobre los datos de prueba, ahora toma un valor de 0,77, pues la curva correspondiente ha resultado ser "menos cóncava" en algunos casos ya que en otros ha superado a la de entrenamiento pero en media son muy parecidas. Para finalizar este apartado de evaluación, vamos a ver cómo llevar a cabo validación cruzada (en este caso repetida). Recordemos que la validación cruzada es especialmente útil a la hora de comparar modelos.

Para que esta función trabaje correctamente, la variable dependiente debe estar formateada de otra manera, poniendo primero la respuesta "Si" y luego el "No".

Dado que se trata de un proceso pseudo aleatorio, fijando la semilla podremos replicar los resultados en otro momento. Indicamos que queremos aplicar validación cruzada repetida en 5 partes y repetido 20 veces. Así mismo, le indicamos que queremos que nos proporcione el resumen de los modelos "multiclase" y que debe guardar las probabilidades para poder utilizarlas posteriormente.

Mostramos la media y los intervalos de confianza (obtenidas gracias a las repeticiones llevadas a cabo) de una multitud de métricas.

Tabla 6 Modelo regresión logística binario resumen medidas

Accuracy	Kappa	Sensitivity	Specificity	AUC
0,799	0,267	0,943	0,274	0,765
AccuracyIC	KappaIC	SensitivityIC	SpecificityIC	AUCIC
(0,7655 - 0,8325)	(0,1415 - 0,3921)	(0,9061 - 0,9804)	(0,1633 - 0,3846)	(0,7001 - 0,8289)

Como podemos ver, las medidas anteriores nos permiten determinar la calidad del modelo tanto en términos de bondad media como de su estabilidad. Por ejemplo, podemos ver que la tasa de acierto siempre es elevada, con un valor medio del 80%, y que el índice kappa muestre un ajuste entre pobre y justo. Nótese que algunas de las

medidas anteriores (las referidas a la matriz de confusión) se refieren al punto de corte "clásico" de 0,5. No obstante, como ya se ha comentado previamente, puede ser recomendable modificar dicho punto de corte en aquellos casos en los que la variable dependiente esté desbalanceada. La función thresholder nos permite obtener varias métricas para los puntos de corte deseados, tal y como vemos a continuación, utilizando la información de validación cruzada:

Tabla 7 Modelo regresión logística binario diferencia en puntos de corte

parameter	prob_threshold	Accuracy	Kappa	Sensitivity	Specificity
none	0,5	0,799	0,267	0,943	0,274
none	0,78	0,705	0,310	0,712	0,679

Como se puede observar, el punto de corte de 0,78 (lo que se corresponde con el porcentaje de eventos del conjunto de datos) da lugar a resultados más equilibrados (sensibilidad y especificidad más similares) pero a cambio de perder ligeramente en la tasa de acierto.

Métodos automáticos de selección de variables

A continuación, vamos a estudiar cómo se pueden seleccionar las variables que van a formar parte del modelo de manera automática. Previamente nos hemos limitado a quedarnos únicamente con las variables que eran significativas pero esta puede ser una estrategia demasiado restrictiva pues es posible que otras variables también sean útiles pero que su información esté duplicada por la presencia de una tercera variable (o bien demasiado laxa y que haya variables que se puedan eliminar). Como ya se ha explicado, vamos a recurrir a los métodos backward, forward y stepwise utilizando como criterio el AIC y el BIC. Para ello, haremos uso de la función step, la cual precisa conocer los modelos máximo y mínimo entre los que debe llevarse a cabo el proceso, así como la dirección de la búsqueda.

Una vez obtenidos los 6 posibles modelos, los incluimos en una lista y obtenemos su fórmula y el número de parámetros que los contienen (esto último puede resultar de utilidad para detectar si los modelos se han repetido).

Modelo	**Fórmula**
1 Stepwise BIC	Jugar ~ Sigo_redes + Sexo + Grupo_edad
2 Stepwise AIC	Jugar ~ Sigo_redes + Grupo_edad + Sexo + Veo_videos + Eventos_presenciales + Estudios + Socializacion_comunidad
3 Forward BIC	Jugar ~ Sigo_redes + Sexo + Grupo_edad
4 Forward AIC	Jugar ~ Sigo_redes + Grupo_edad + Sexo + Veo_videos + Eventos_presenciales + Estudios + Socializacion_comunidad
5 Backward BIC	Jugar ~ Sexo + Estudios + Sigo_redes
6 Backward AIC	Jugar ~ Grupo_edad + Sexo + Estudios + Veo_videos + Sigo_redes + Eventos_presenciales +

	Pasion_videojuegos + Equipamiento_habilidad + Impacto_personal_salud

El número de parámetros según cada modelo es este (8 19 8 19 8 21), por lo que vemos 4 modelos distintos, el primero y el tercero son iguales, y el segundo y el cuarto también, el 5 y el 6 son distintos a los demás.

En primer lugar, creamos una lista que contenga todos los modelos que se deseen comparar (es decir, modeloInicial, modelo2, modeloStepBIC, modeloStepAIC, modeloBackBIC, modeloBackAIC) y seguidamente creamos un bucle para recorrer dicha lista y llevar a cabo la validación cruzada repetida.

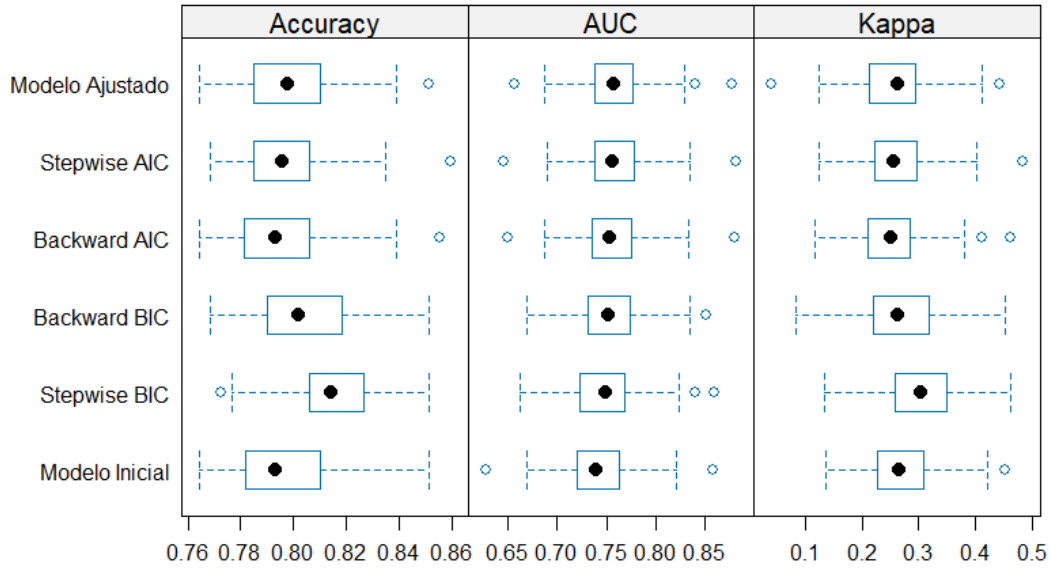


Ilustración 11 Método automático de selección de variables comparación de métricas

Tabla 8 Método automático de selección de variables comparación de métricas

Modelos	AUC	IC AUC	Kappa	IC Kappa	Accuracy	IC Accuracy
Modelo Inicial	0,750	(0,676 - 0,824)	0,275	(0,134 - 0,416)	0,797	(0,761 - 0,833)
Modelo Ajustado	0,762	(0,687 - 0,837)	0,264	(0,112 - 0,416)	0,797	(0,761 - 0,833)
Stepwise BIC	0,748	(0,662 - 0,834)	0,308	(0,144 - 0,472)	0,814	(0,773 - 0,855)
Stepwise AIC	0,762	(0,688 - 0,836)	0,273	(0,154 - 0,392)	0,798	(0,760 - 0,836)
Backward BIC	0,750	(0,679 - 0,821)	0,269	(0,125 - 0,413)	0,808	(0,779 - 0,837)
Backward AIC	0,761	(0,691 - 0,831)	0,273	(0,151 - 0,395)	0,799	(0,757 - 0,841)

Todos los modelos son muy parecidos pero por el principio de parsimonia (que indica elegir el modelo más sencillo cuando tiene una capacidad predictiva similar) nos

vamos a quedar con el tercer modelo que es el modelo StepBIC que tiene mayor precisión, mayor kappa y un AUC similar al resto de modelos y lo importante es que este modelo tiene 8 parámetros por lo que facilita su interpretación. Con los datos de entrenamiento obtenemos estos resultados:

Tabla 9 Método automático de selección de variables medidas entrenamiento

Métrica	Media
Accuracy	0,708
Kappa	0,312
Sensitivity	0,717
Specificity	0,675
AUC	0,767

En resumen, se ha construido un modelo con un AUC de 0.77, una tasa de acierto del 71% y un índice Kappa de 0.31, relativamente estable con una sensibilidad y especificidad parecida. Para finalizar, recordamos la importancia de llevar a cabo una evaluación final realista del modelo StepBIC utilizando los datos de prueba con el punto de corte de 0,78.

Por último, una vez comprobada empíricamente la gran influencia que tiene el punto de corte en algunas medidas de evaluación (nótese que el índice Kappa no es tan sensible), pasamos a obtener la curva ROC (y su área) para contar con otra herramienta de evaluación de modelos.

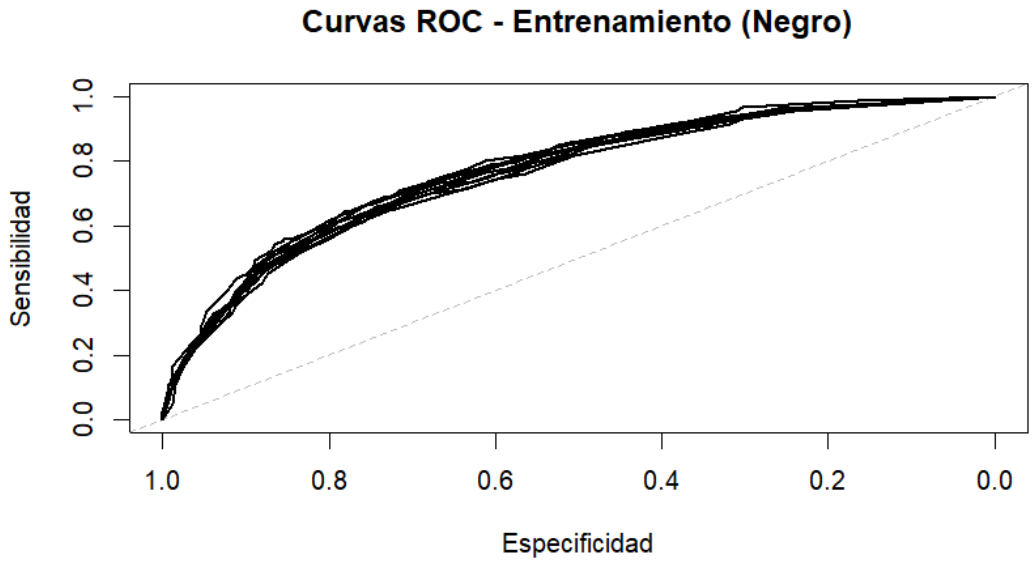


Ilustración 12 Método automático de selección de variables curva ROC entrenamiento

Observamos, de nuevo, que el modelo construido es de buena calidad, pues el valor de su AUC es de 0,77. Dado que evaluar los modelos con la información de los datos de entrenamiento puede dar lugar a conclusiones excesivamente optimistas, es recomendable recurrir a la partición de prueba para obtener estimaciones más realistas de las medidas de evaluación.

Tabla 10 Método automático de selección de variables medidas prueba

Métrica	Media
Accuracy	0,714
Kappa	0,312
Sensitivity	0,731
Specificity	0,652
AUC	0,756

En resumen, se ha construido un modelo con 8 parámetros con un AUC de 0.76, una tasa de acierto del 71% y un índice Kappa de 0.31, relativamente estable con una sensibilidad y especificidad parecida. Vemos que en este caso no varían casi los valores entre entrenamiento y prueba.

Obtenemos a continuación la curva ROC y el área (AUC). Se ha optado por representar conjuntamente las curvas en entrenamiento (negro) y en prueba (rojo) para facilitar la comparación.

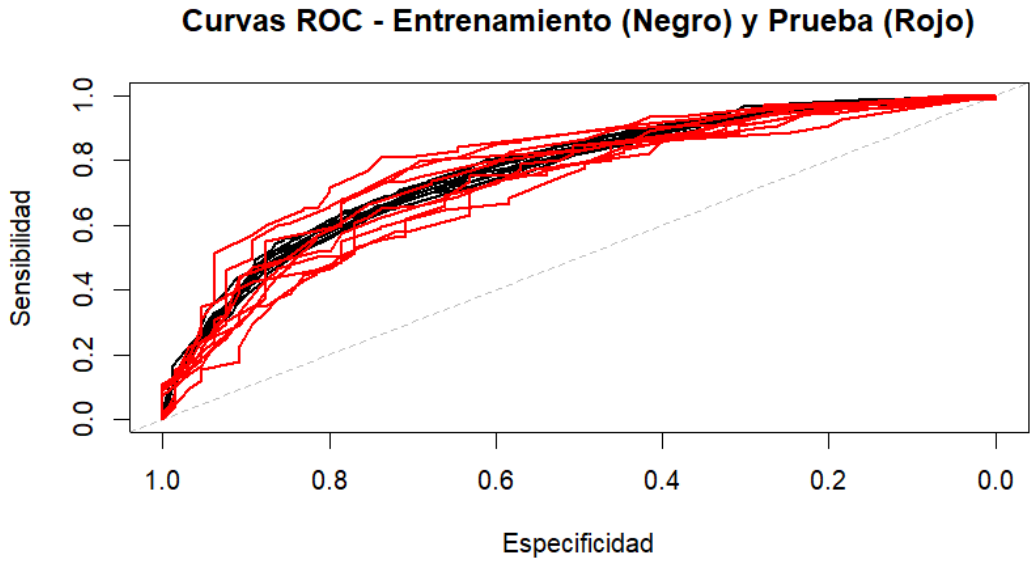


Ilustración 13 Método automático de selección de variables curva ROC entrenamiento y prueba

Tal y como ha ocurrido con los anteriores indicadores, el AUC ha disminuido ligeramente al obtenerlo sobre los datos de prueba, ahora toma un valor de 0,76, pues la curva correspondiente ha resultado ser “menos cóncava” en algunos casos ya que en otros ha superado a la de entrenamiento pero en media son muy parecidas. Finalmente, calculamos los odds-ratio de los efectos del modelo, lo que nos permitirá interpretar sus parámetros y, en otras palabras, cuantificar el impacto de las variables independientes sobre la variable dependiente.

Tabla 11 Método automático de selección de variables odds ratio

Variable	**Odds_Ratio**	**P_Valor**	**Significación**
(Intercept)	1,038	0,862	

Sigo_redesCon menor frecuencia	2,756	0,000	***
Sigo_redesAl menos 1 vez cada 15 días	4,278	0,000	***
Sigo_redesAl menos 1 vez por semana	8,314	0,000	***
Sigo_redesTodos los días	14,401	0,000	***
SexoMujeres	0,488	0,000	***
Grupo_edadDe 20 a 24	1,618	0,011	*
Grupo_edadDe 25 a 29	2,233	0,000	***

Observamos que es un modelo con 8 parámetros donde 7 son significativos a un nivel de significación del 5%, es decir, todos menos la constante, viendo la tabla ANOVA podemos decir que las tres variables que tenemos son significativas, por tanto por último nos queda interpretar los Odds-ratio.

 # Sigo_redesCon menor frecuencia: aumenta un 176% la posibilidad de jugar al pasar de nunca a con menor frecuencia

 # Sigo_redesAl menos 1 vez cada 15 días: aumenta un 328% la posibilidad de jugar al pasar de nunca a al menos 1 vez cada 15 días

 # Sigo_redesAl menos 1 vez por semana: aumenta un 731% la posibilidad de jugar al pasar de nunca a al menos 1 vez por semana

 # Sigo_redesTodos los días: aumenta un 1340% la posibilidad de jugar al pasar de nunca a todos los días

 # SexoMujeres: disminuye un 47% la posibilidad de jugar al pasar de hombres a mujeres

 # Grupo_edadDe 20 a 24: aumenta un 62% la posibilidad de jugar al pasar de grupo de edad de 15 a 19 a 20 a 24

 # Grupo_edadDe 25 a 29: aumenta un 123% la posibilidad de jugar al pasar de grupo de edad de 15 a 19 a 25 a 29

En este estudio, se realizó un modelo de regresión logística binaria seleccionando manualmente las variables con el objetivo de compararlo con los modelos generados mediante métodos automáticos de selección de variables. Los resultados obtenidos indican que el modelo construido manualmente presenta un AUC menor en comparación con los modelos generados automáticamente, lo cual es consistente con la expectativa de que los métodos automáticos suelen optimizar la selección de variables para maximizar la capacidad predictiva.

Es importante mencionar que, aunque el modelo manual no representa la forma más adecuada de selección de variables, se incluyó en el análisis como referencia comparativa. Este enfoque permite observar cómo las decisiones subjetivas pueden influir en los resultados y refuerza la importancia de emplear técnicas automáticas basadas en criterios objetivos, especialmente cuando el objetivo principal es obtener un modelo con mejor capacidad predictiva. Este resultado también subraya que la

selección automática de variables no solo es más eficiente, sino que proporciona modelos con un rendimiento superior, como se evidencia en este caso.

Modelo de Naive Bayes

Pese a que los modelos NB funcionan relativamente bien aunque se incluyan variables explicativas poco relacionadas con la dependiente, si se cuenta con muchas variables, el efecto conjunto de las mismas puede dar lugar a sobreajuste.

Para evitarlo, se deben incluir en el modelo únicamente aquellas que estén relacionadas con la variable dependiente. A diferencia de lo que ocurre con otros modelos, NB no cuenta con un método propio de selección de variables por lo que debemos recurrir a métodos generales.

En este caso, nos centramos en el método recursivo de eliminación de variables (más conocido como RFE por sus siglas en inglés recursive feature elimination). Este método se asemeja al método de selección de variables “hacia atrás” de los modelos de regresión puesto que consiste en eliminar iterativamente las variables independientes menos importantes del modelo.

De cara a medir qué variables son más/menos importantes, teniendo en cuenta las propiedades de independencia del modelo NB, lo que se hace es estudiar, a través del área bajo la curva ROC, el potencial predictivo univariante de cada una de ellas, lo que permite, a su vez, ordenarlas.

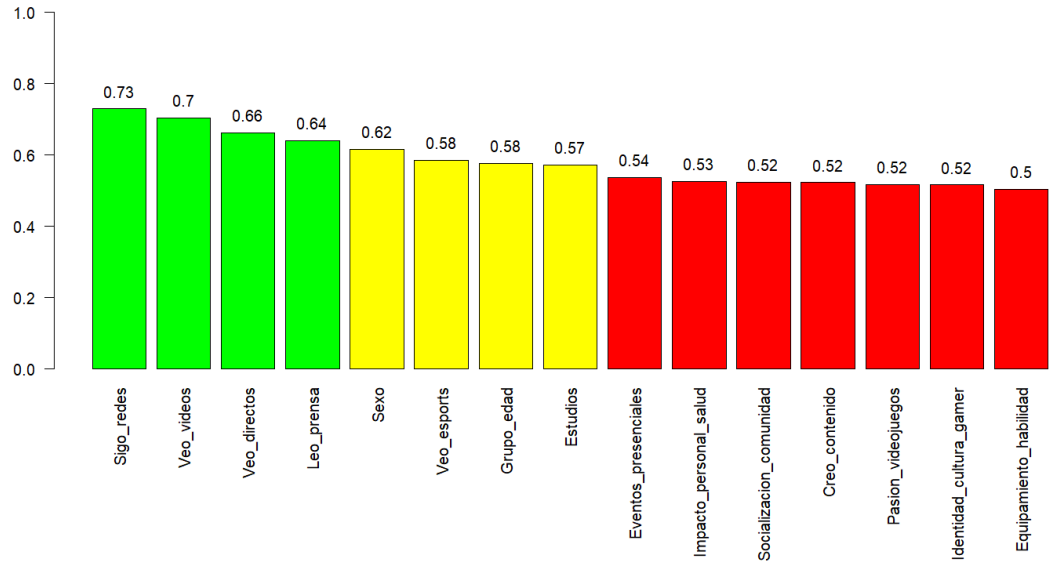


Ilustración 14 Modelo Naive Bayes potencial predictivo de las variables

Como se puede observar, la variable con mayor potencial predictivo es Sigo_redes, seguido de Veo_videos, Veo_directos y Leo_prensa, representadas en verde. En el otro extremo, podemos observar que las variables Equipamiento_habilidad, Identidad_cultura_gamer, Pasión_videojuegos, Creo_contenido y Socialización_comunidad resultarán de poca utilidad dado que su AUC es muy próximo a 0.5, estas variables están representadas en rojo.

Una vez obtenido el ranking, se debe determinar cuántas de las variables más importantes se deben incluir en el modelo. Para ello, se recurre al método RFE

explicado previamente, lo que nos permitirá obtener las “mejores” combinaciones de variables y evaluarlas mediante validación cruzada repetida (VCR).

En este caso se va a utilizar el AUC como medida para determinar la combinación óptima de variables:

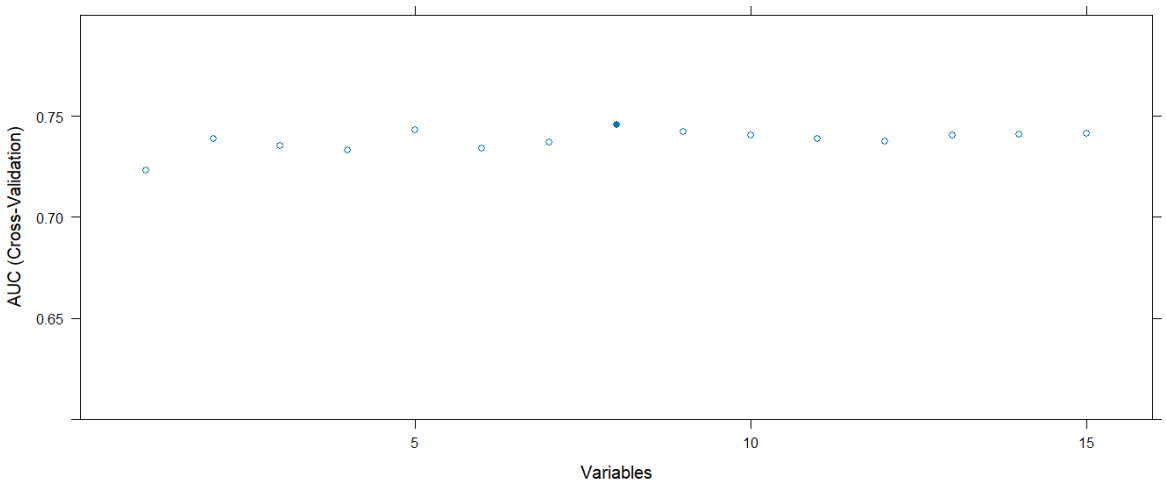


Ilustración 15 Modelo Naive Bayes comparación AUC

Tal y como se puede observar, el RFE nos indica seleccionar 8 variables para construir los modelos NB. No obstante, observamos que no existen grandes diferencias (en cuanto al AUC) con aquellos modelos que constan de 3-15. Según el principio de parsimonia, esto nos indicaría que lo recomendable es trabajar únicamente con las 3 primeras variables.

Vamos a ver las medidas del Kappa y la Precisión para estos datos también:

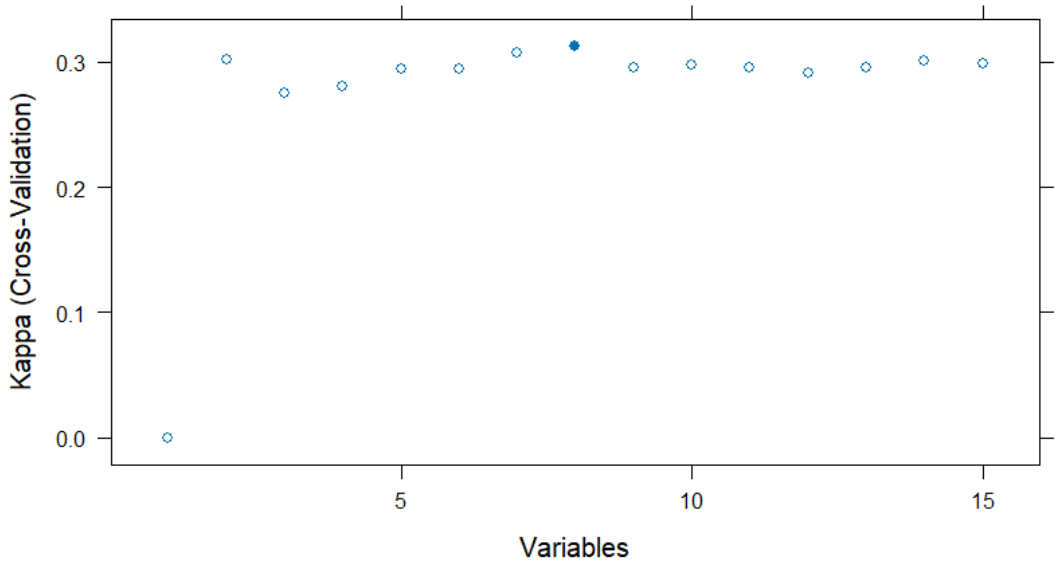


Ilustración 16 Modelo Naive Bayes comparación Kappa

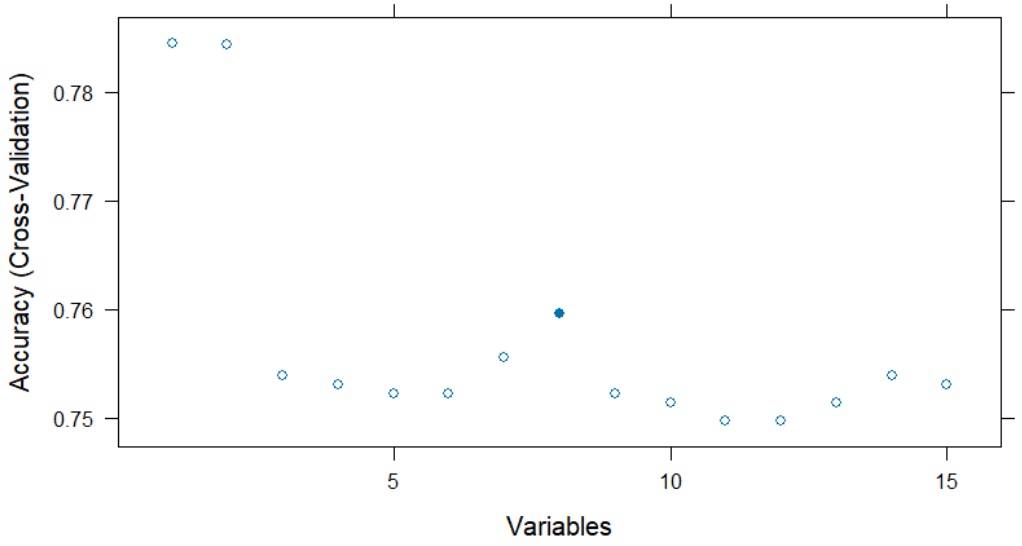


Ilustración 17 Modelo Naive Bayes comparación Precisión

Podemos observar un patrón muy similar al del AUC: 8 variables dan lugar al resultado óptimo, pero 3 variables ofrecen resultados muy similares sin una gran pérdida de calidad y con una gran reducción en la complejidad del modelo

Una vez obtenido el conjunto óptimo de variables, debemos determinar la parametrización óptima del modelo. Para ello, recurrimos a la validación cruzada repetida. En este caso, debemos indicar que el método es naive_bayes, qué variables queremos utilizar como independientes y los parámetros correspondientes para la estimación de las probabilidades condicionadas

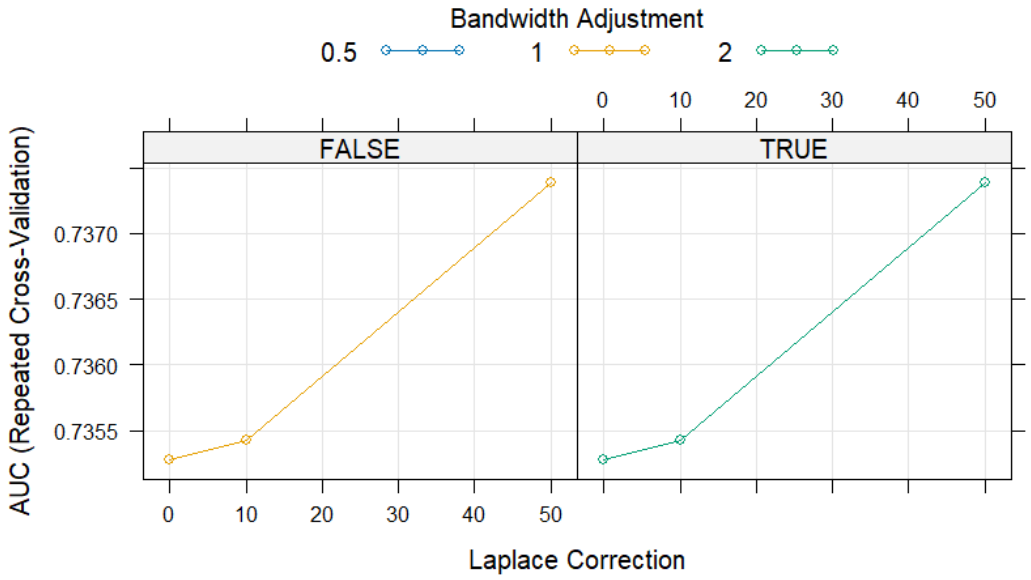


Ilustración 18 Modelo Naive Bayes corrección de Laplace con 3 variables

Al representar los resultados, se crea un gráfico en el que se muestra el AUC medio de las distintas repeticiones llevadas a cabo mediante VCR para cada combinación de parámetros. En este caso, podemos ver que el mejor resultado se obtiene cuando se

aplica KDE sobre las variables cualitativas con un ajuste de ventana de 0,5 aunque no se vea ya que valen exactamente lo mismo que con 1 y 2 y un ajuste de Laplace de 50.

Aunque ya se ha comentado que, por el principio de parsimonia, es preferible contar con un modelo con “pocas” variables, a modo ilustrativo vamos a parametrizar también la combinación de 8 variables. Este proceso puede resultar de utilidad cuando la mejor combinación de variables independientes no es evidente puesto que, al parametrizar, podemos obtener versiones mejoradas de los modelos que contienen las variables seleccionadas.

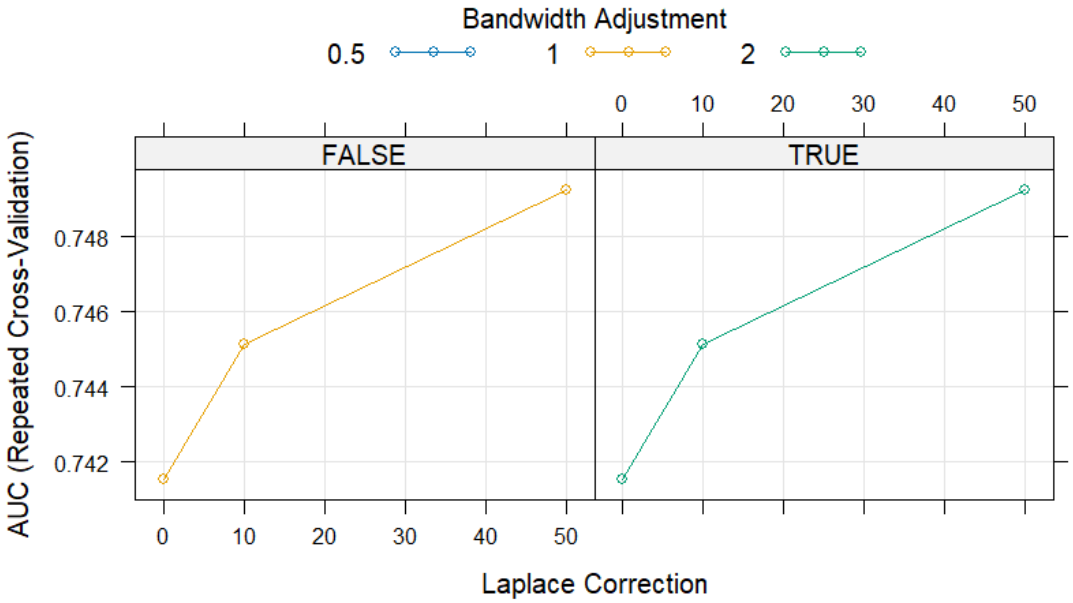


Ilustración 19 Modelo Naive Bayes corrección de Laplace con 8 variables

Como se puede observar, según se modifican las variables explicativas disponibles, los resultados obtenidos varían. No obstante, en este caso también resulta preferible aplicar el ajuste de Laplace de 50, utilizar estimación no paramétrica y que el ajuste de la ventana sea 0,5. Adicionalmente, podemos ver que el AUC medio del mejor modelo es ligeramente superior utilizando 8 variables, que utilizando sólo 3. No obstante, para poder comparar el mejor modelo de los dos casos podemos recurrir a la función resamples, la cual extrae los resultados de las repeticiones para todas las medidas disponibles para una lista de modelos que le facilitemos. En este caso hemos optado por analizar únicamente el AUC y el índice Kappa pero se podrían incluir otras medidas sin más que añadirlas a la lista.

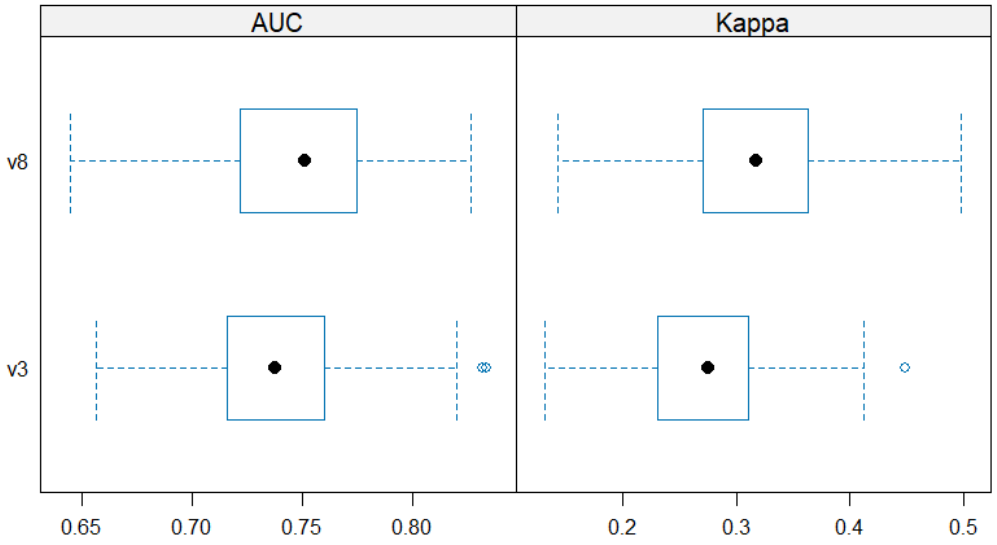
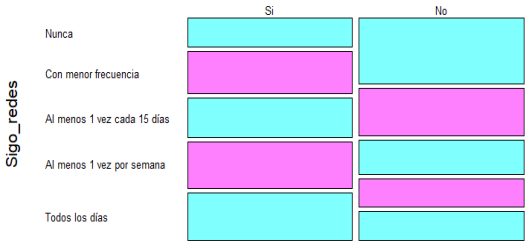
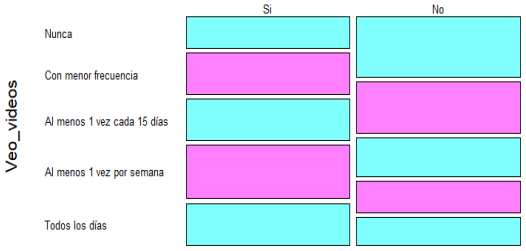


Ilustración 20 Modelo Naive Bayes comparación métricas

Como podemos ver, en cuanto a las dos métricas, los mejores resultados se obtienen para el modelo v8 en todos los estadísticos analizados y en el gráfico. En cuanto a la variabilidad, podemos ver que no existen grandes diferencias entre los modelos, pero dado que la diferencia entre el modelo v3 y v8 en términos de calidad es muy baja y que el modelo v8 incorpora más variables y muestra una ligera mejora, optamos por este último, ya que proporciona un análisis más completo.

Para finalizar, vamos a analizar el modelo ganador. Comenzamos observando las distribuciones condicionadas del modelo, para conocer la dirección de la influencia de las variables explicativas sobre la dependiente:





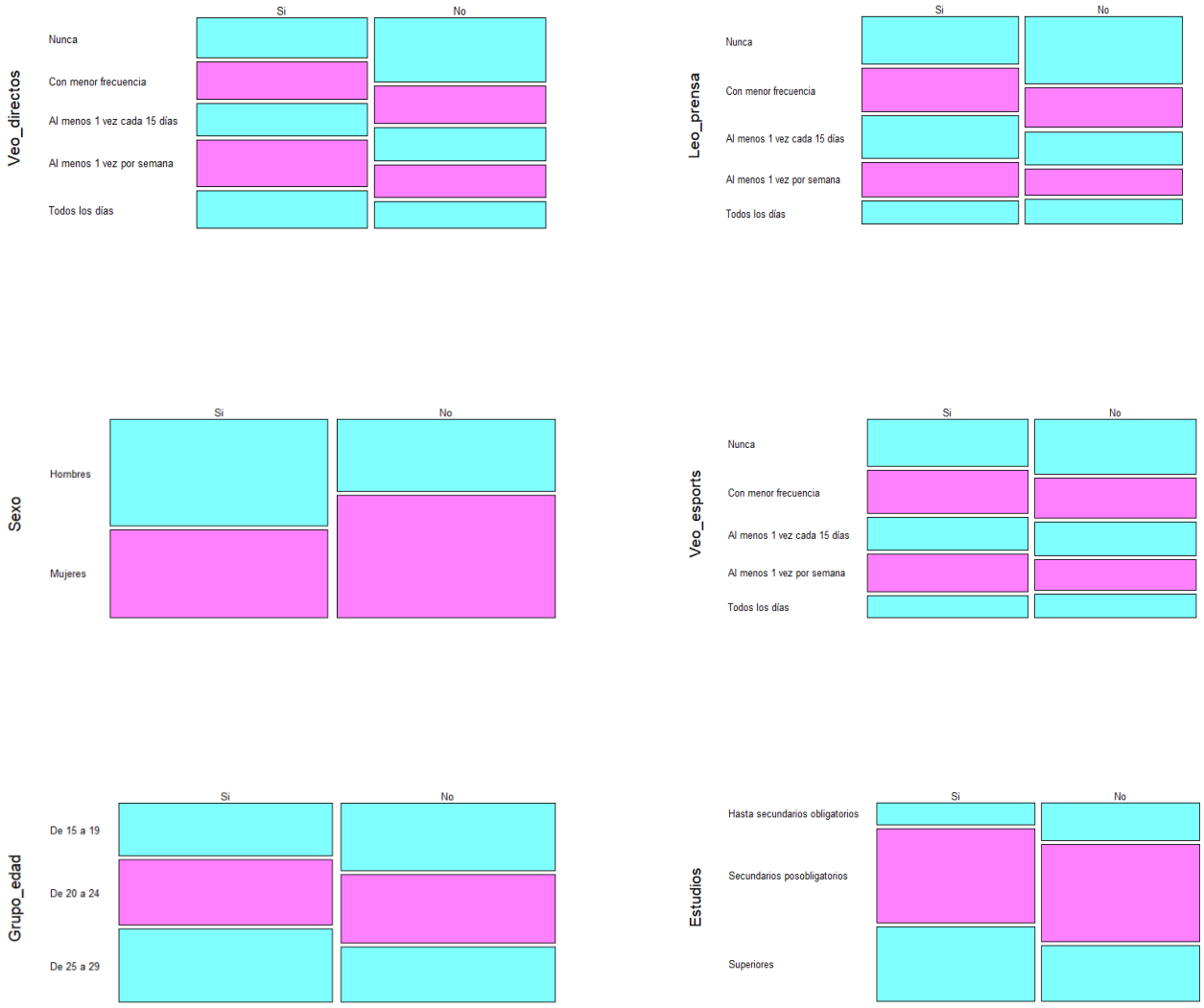


Ilustración 21Distribución de las variables explicativas sobre la dependiente

Como se puede observar, los eventos (jugar, denotado como “Si”), están caracterizados por valores más grandes en las tres primeras variables en categorías como “Todos los días” y “Al menos 1 vez por semana”, mientras que los valores más grandes para los no eventos están caracterizados por valores más grandes en las categorías como “Nunca” o “Con menor frecuencia”. Para las variables Leo_prensa y Veo_esports tanto los eventos como los no eventos están caracterizados por valores grandes en las categorías de “Nunca” o “Con menor frecuencia”, con la variable Sexo podemos ver que los hombres juegan más que las mujeres, por Edad se ve que juegan las personas adultas más que las jóvenes y respecto a los Estudios los que juegan tienen estudios secundarios posobligatorios y superiores mientras que para los que no juegan la mayoría tienen estudios secundarios posobligatorios. Vemos ahora las medidas de evaluación en entrenamiento:

Tabla 12 Modelo Naive Bayes medidas entrenamiento

Métrica	Media

Accuracy	0,693
Kappa	0,310
Sensitivity	0,684
Specificity	0,725
AUC	0,771

Observamos que contamos con un modelo de calidad media, con una tasa de acierto del 69%, un índice Kappa de 0,31, una especificidad un poco mayor a la sensibilidad y un área bajo la curva ROC de 0.77. Por último, una vez comprobada empíricamente la gran influencia que tiene el punto de corte en algunas medidas de evaluación (nótese que el índice Kappa no es tan sensible), pasamos a obtener la curva ROC (y su área) para contar con otra herramienta de evaluación de modelos.

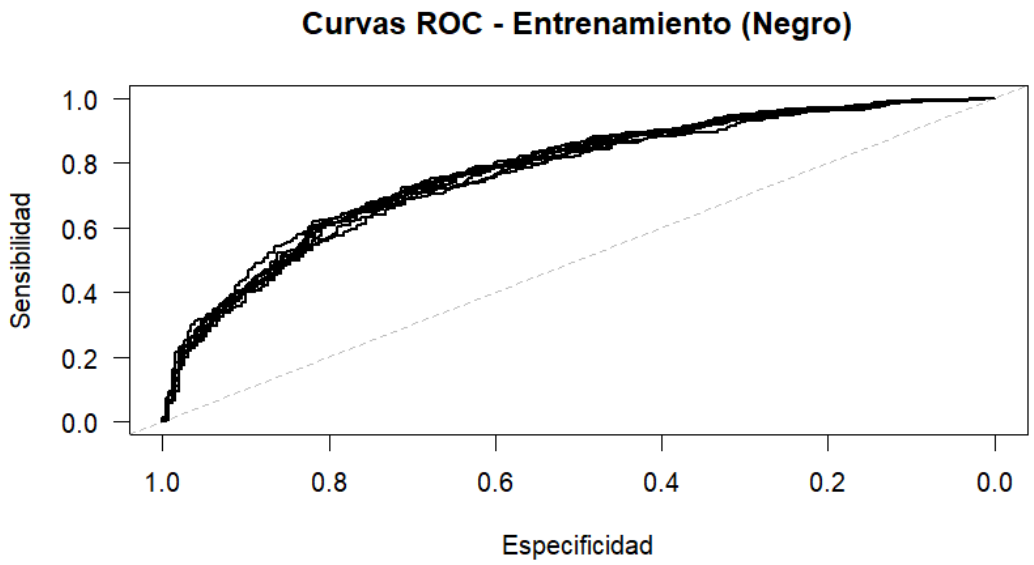


Ilustración 22 Modelo Naive Bayes curva ROC entrenamiento

Observamos, de nuevo, que el modelo construido es de buena calidad, pues el valor de su AUC es de 0,77. Dado que evaluar los modelos con la información de los datos de entrenamiento puede dar lugar a conclusiones excesivamente optimistas, es recomendable recurrir a la partición de prueba para obtener estimaciones más realistas de las medidas de evaluación. Finalizamos este documento mediante las medidas de evaluación del modelo en la partición de prueba con el punto de corte en 0,78:

Tabla 13 Modelo Naive Bayes medidas prueba

Métrica	Media
Accuracy	0,684
Kappa	0,295
Sensitivity	0,677
Specificity	0,711
AUC	0,759

Observamos que contamos con un modelo de calidad media, con una tasa de acierto del 68%, un índice Kappa de 0,30, una especificidad un poco mayor a la sensibilidad y un área bajo la curva ROC de 0.76.

Obtenemos a continuación la curva ROC y el área (AUC). Se ha optado por representar conjuntamente las curvas en entrenamiento (negro) y en prueba (rojo) para facilitar la comparación.

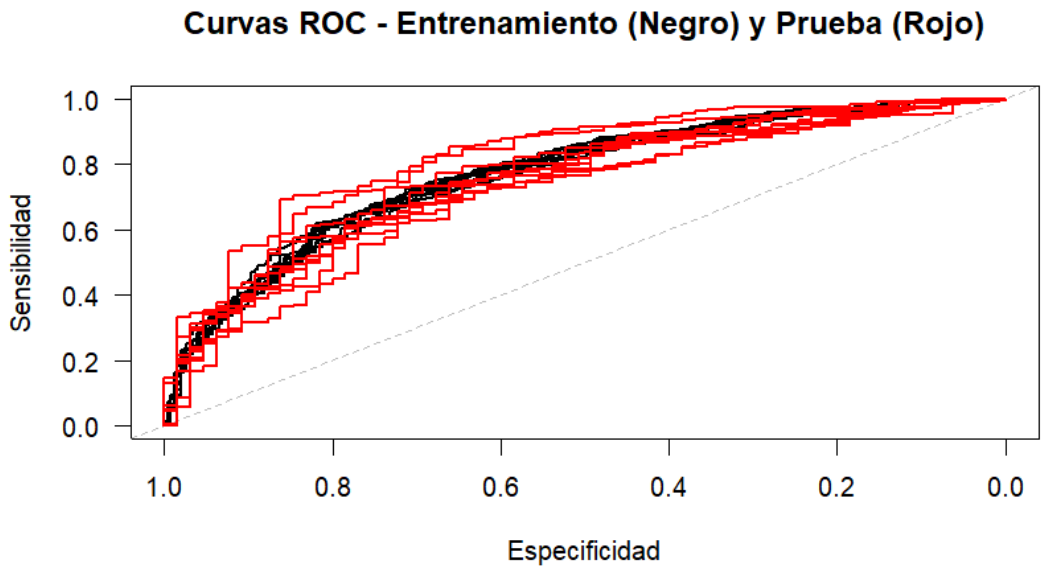


Ilustración 23 Modelo Naive Bayes curva ROC entrenamiento y prueba

Tal y como ha ocurrido con los anteriores indicadores, el AUC ha disminuido ligeramente al obtenerlo sobre los datos de prueba, ahora toma un valor de 0,76, pues la curva correspondiente ha resultado ser “menos cóncava” en algunos casos ya que en otros ha superado a la de entrenamiento pero en media son muy parecidas.

Modelos Lasso y Elastic Net

En este apartado, exploraremos dos métodos de regularización utilizados en modelos de regresión: Lasso y Elastic Net. Ambos modelos buscan mejorar la capacidad predictiva y reducir la complejidad al penalizar los coeficientes de las variables, favoreciendo así la selección automática de variables relevantes y evitando el sobreajuste [10].

Lasso emplea una penalización basada en la norma L1, lo que permite que algunos coeficientes se reduzcan exactamente a cero, facilitando la interpretación del modelo. Por otro lado, Elastic Net combina la penalización L1 de Lasso con la penalización L2 de Ridge, permitiendo mayor flexibilidad en la selección de variables cuando existe colinealidad entre ellas.

Para evaluar el rendimiento de estos modelos, dividiremos los datos en conjuntos de entrenamiento y prueba, asegurando la reproducibilidad mediante la fijación de semillas. Compararemos los resultados con las métricas que estábamos trabajando antes. Además, analizaremos las curvas ROC para visualizar la capacidad discriminativa de cada modelo.

Modelo Lasso

El gráfico de barras muestra la frecuencia con la que cada variable fue seleccionada en el proceso de Lasso. Las barras están coloreadas de verde para las variables seleccionadas en todas las repeticiones, lo que indica que son las más relevantes y consistentes en el modelo. Las variables con menor frecuencia de selección se presentan en otros colores, lo que permite identificar cuáles tienen una relevancia más variable o menor consistencia a lo largo de las repeticiones.

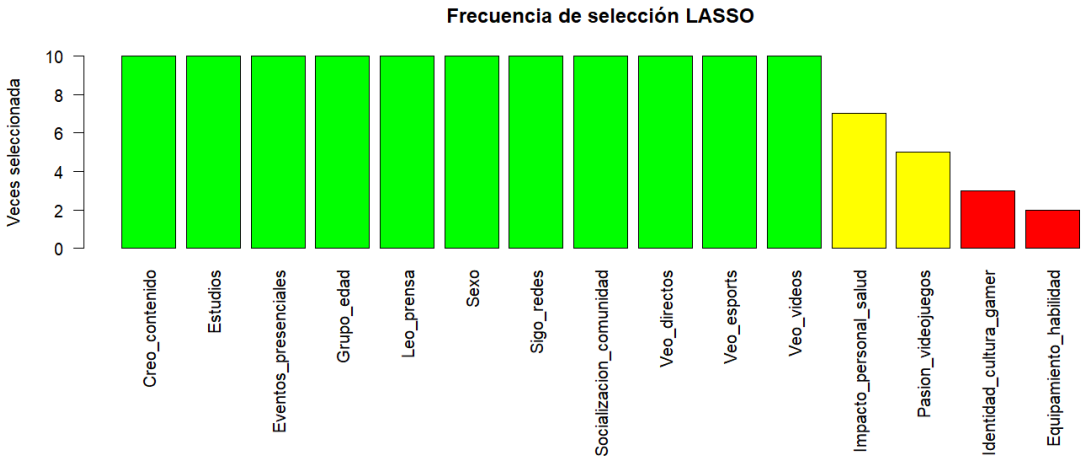


Ilustración 24 Modelo Lasso selección de variables

Tomamos el punto de corte en 0,78 y para los datos de entrenamiento obtenemos estos resultados:

Tabla 14 Modelo Lasso medidas entrenamiento

Métrica	Media
Accuracy	0,642
Kappa	0,273
Sensitivity	0,599
Specificity	0,797
AUC	0,787

En resumen, se ha construido un modelo con un AUC de 0,79, una tasa de acierto del 64% y un índice Kappa de 0,27, relativamente estable con una sensibilidad de 0,6 y especificidad de 0,8.

A continuación, pasamos a obtener la curva ROC (y su área) para contar con otra herramienta de evaluación de modelos.

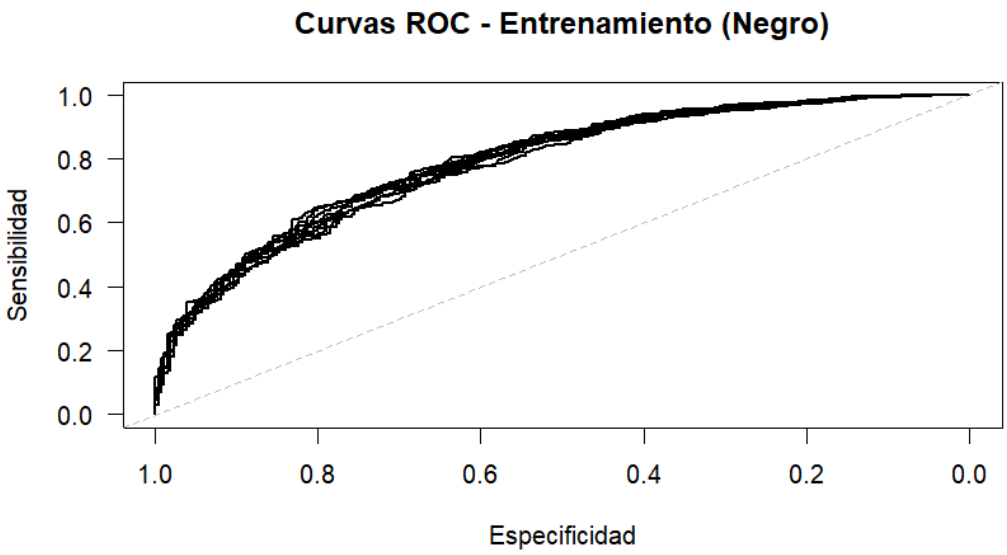


Ilustración 25 Modelo Lasso curva ROC entrenamiento

Observamos, de nuevo, que el modelo construido es de buena calidad, pues el valor de su AUC es de 0,79. Dado que evaluar los modelos con la información de los datos de entrenamiento puede dar lugar a conclusiones excesivamente optimistas, es recomendable recurrir a la partición de prueba para obtener estimaciones más realistas de las medidas de evaluación.

Tabla 15 Modelo Lasso medidas prueba

Métrica	Media
Accuracy	0,648
Kappa	0,281
Sensitivity	0,606
Specificity	0,800
AUC	0,778

En resumen, se ha construido un modelo con un AUC de 0,78, una tasa de acierto del 65% y un índice Kappa de 0,28, relativamente estable con una sensibilidad de 0,6 y especificidad de 0,8. Vemos que en este caso no varían casi los valores entre entrenamiento y prueba incluso mejoran un poco en prueba.

Obtenemos a continuación la curva ROC y el área (AUC). Se ha optado por representar conjuntamente las curvas en entrenamiento (negro) y en prueba (rojo) para facilitar la comparación.

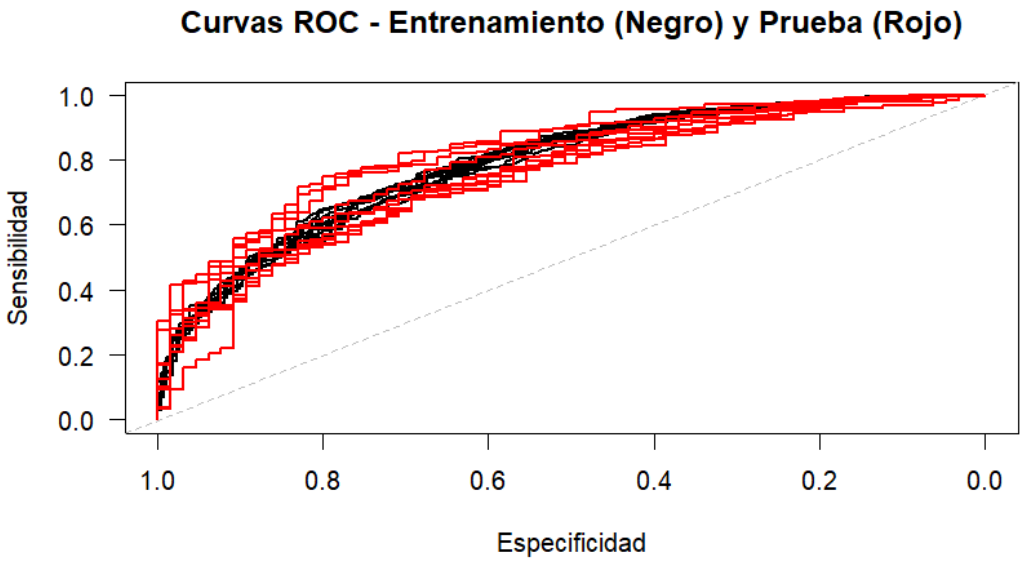


Ilustración 26 Modelo Lasso curva ROC entrenamiento y prueba

Tal y como ha ocurrido con los anteriores indicadores, el AUC ha disminuido ligeramente al obtenerlo sobre los datos de prueba, ahora toma un valor de 0,78, pues la curva correspondiente ha resultado ser “menos cóncava” en algunos casos ya que en otros ha superado a la de entrenamiento pero en media son muy parecidas.

Modelo Elastic Net

El gráfico de barras refleja la frecuencia con la que cada variable fue seleccionada durante el proceso de Elastic Net. Las barras verdes representan aquellas variables que fueron seleccionadas en todas las repeticiones, señalando que son las más relevantes y estables en el modelo. Las variables con una menor frecuencia de selección están coloreadas de forma diferente, permitiendo observar cuáles tienen una relevancia más variable o menos consistente entre las diferentes ejecuciones. Cabe destacar que las variables mostradas son las mismas que en el modelo Lasso, y los colores asignados coinciden para ambos modelos, facilitando una comparación visual directa.

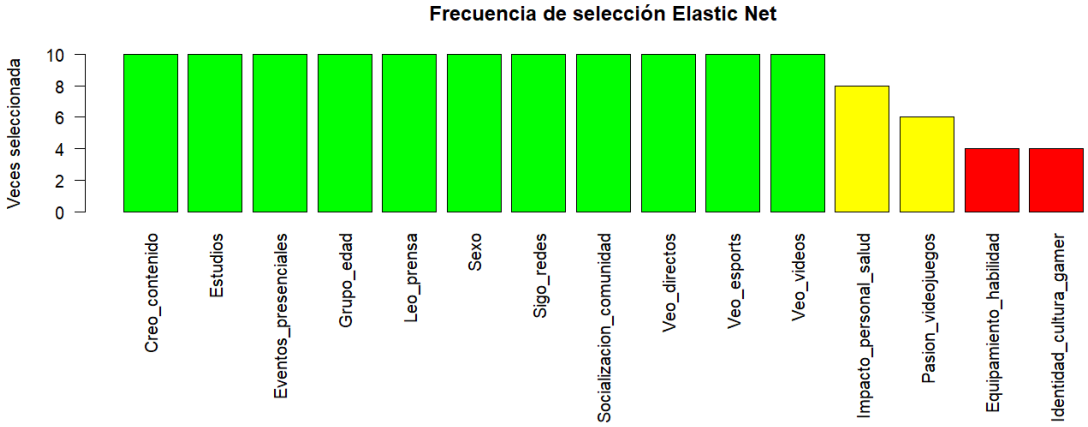


Ilustración 27 Modelo Elastic Net selección de variables

Tomamos el punto de corte en 0,78 y para los datos de entrenamiento obtenemos estos resultados:

Tabla 16 Modelo Elastic Net medidas entrenamiento

métrica	Media
Accuracy	0,641
Kappa	0,272
Sensitivity	0,598
Specificity	0,797
AUC	0,787

En resumen, se ha construido un modelo con un AUC de 0,79, una tasa de acierto del 64% y un índice Kappa de 0,27, relativamente estable con una sensibilidad de 0,6 y especificidad de 0,8.

A continuación, pasamos a obtener la curva ROC (y su área) para contar con otra herramienta de evaluación de modelos.

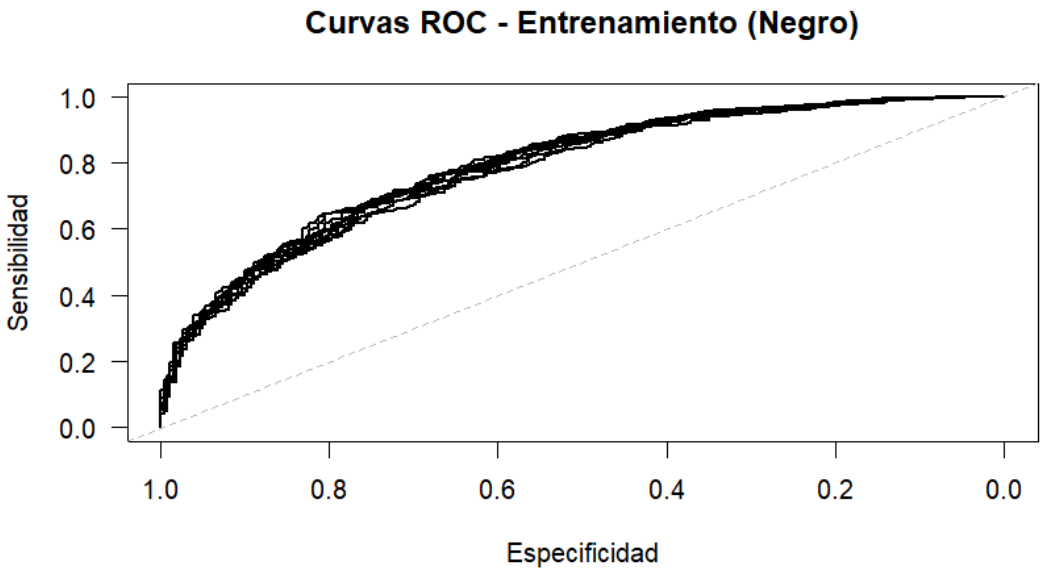


Ilustración 28 Modelo Elastic Net curva ROC entrenamiento

Observamos, de nuevo, que el modelo construido es de buena calidad, pues el valor de su AUC es de 0,79. Dado que evaluar los modelos con la información de los datos de entrenamiento puede dar lugar a conclusiones excesivamente optimistas, es recomendable recurrir a la partición de prueba para obtener estimaciones más realistas de las medidas de evaluación.

Tabla 17 Modelo Elastic Net medidas prueba

Métrica	Media
Accuracy	0,650
Kappa	0,283
Sensitivity	0,608
Specificity	0,800

AUC	0,780

En resumen, se ha construido un modelo con un AUC de 0,78, una tasa de acierto del 65% y un índice Kappa de 0,28, relativamente estable con una sensibilidad de 0,6 y especificidad de 0,8. Vemos que en este caso no varían casi los valores entre entrenamiento y prueba incluso mejoran un poco en prueba.

Obtenemos a continuación la curva ROC y el área (AUC). Se ha optado por representar conjuntamente las curvas en entrenamiento (negro) y en prueba (rojo) para facilitar la comparación.

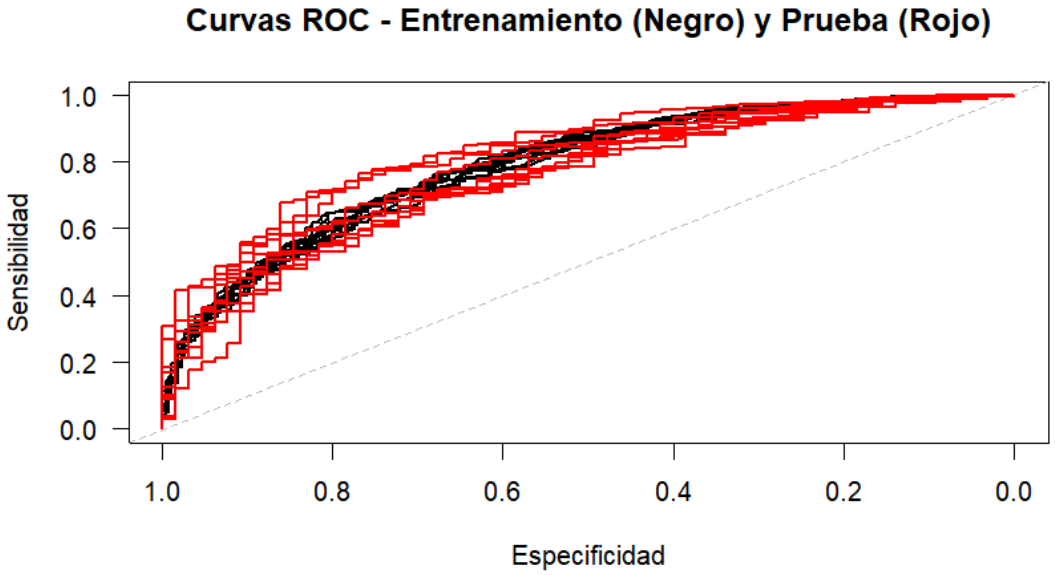


Ilustración 29 Modelo Elastic Net curva ROC entrenamiento y prueba

Tal y como ha ocurrido con los anteriores indicadores, el AUC ha disminuido ligeramente al obtenerlo sobre los datos de prueba, ahora toma un valor de 0,78, pues la curva correspondiente ha resultado ser “menos cóncava” en algunos casos ya que en otros ha superado a la de entrenamiento pero en media son muy parecidas.

Podemos observar que los modelos Lasso y Elastic Net dan resultados muy parecidos con muy pocas diferencias por lo que se podría usar cualquiera de los dos.

Árboles de clasificación

Los árboles de clasificación son una herramienta poderosa en el análisis de datos debido a su capacidad para modelar relaciones no lineales entre las variables independientes y la variable dependiente. Se estructuran como un árbol invertido, donde cada nodo interno representa una decisión lógica que divide los datos, y el proceso continúa desde la raíz hasta llegar a los nodos terminales, que definen la clase final [11]. A diferencia de los modelos lineales, que asumen que las relaciones entre las variables son lineales y siguen un patrón específico (como ocurre en la regresión lineal o logística), los árboles de clasificación no requieren esta suposición. Esto los hace particularmente útiles cuando no podemos asumir que los datos siguen una relación lineal.

Este enfoque es especialmente valioso al trabajar con datos complejos o no estructurados, ya que los árboles de clasificación son capaces de manejar interacciones entre variables de manera natural, sin necesidad de transformaciones complicadas o preprocesamiento de los datos. Además, su capacidad para generar representaciones visuales claras facilita la interpretación del modelo, permitiendo entender cómo se toman las decisiones de clasificación en función de las características de los datos.

En este trabajo, utilizamos árboles de clasificación para predecir la clase de la variable dependiente, evitando la necesidad de suponer que las relaciones entre las variables sean lineales, como ocurre en otros modelos estadísticos. Este enfoque permite una mayor flexibilidad y aplicabilidad a una amplia gama de problemas prácticos.

Para construir los árboles de clasificación, utilizamos la función rpart() del paquete rpart, que es ampliamente utilizada en R para este tipo de modelos. En este caso, es importante especificar que el objetivo es realizar una clasificación, por lo que se debe establecer el argumento method = "class". Esto indica que estamos tratando con un problema de clasificación y no de regresión.

Además, se debe seleccionar el criterio para la división de los nodos del árbol. En este caso, se ha optado por el índice de Gini, que es un criterio comúnmente utilizado en árboles de clasificación, ya que mide la pureza de los nodos y busca dividir los datos de manera que los nodos resultantes sean lo más homogéneos posible en cuanto a las clases de la variable objetivo.

Una vez que el árbol de clasificación se ha entrenado, se visualiza utilizando la función rpart.plot(). En este paso, se ha incluido el argumento extra = 105, que permite mostrar más detalles en los nodos del árbol, proporcionando información adicional como la distribución de las clases en cada nodo y las probabilidades asociadas a cada clase. Es importante mencionar que este argumento no es aplicable a árboles de regresión, ya que en estos árboles la información mostrada es diferente.

A través de este enfoque, podemos evaluar cómo el árbol divide el espacio de características para predecir la clase de la variable dependiente, permitiéndonos interpretar de manera visual y estructurada el modelo de clasificación.

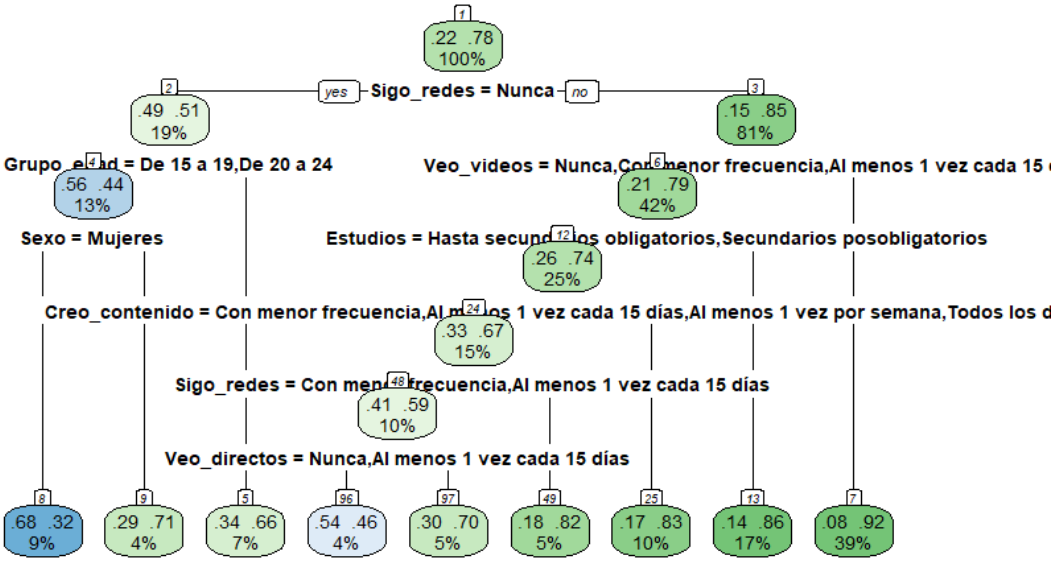


Ilustración 30 Árbol de clasificación con criterio de Gini

La función utilizada ha generado un árbol de clasificación con una profundidad de 6 niveles y un total de 9 hojas terminales. Cada nodo del árbol contiene información esencial que permite interpretar cómo se distribuyen las observaciones a medida que se avanza en las decisiones de clasificación. Concretamente, en cada nodo se muestran tres elementos principales: en la primera fila aparecen las proporciones relativas de las dos categorías de la variable objetivo (en este caso, "no juega" y "juega", en ese orden); en la segunda fila, se incluye el porcentaje de observaciones del total de la muestra que han llegado hasta ese nodo; y, finalmente, el color del nodo refleja visualmente la proporción de eventos: los nodos verdes indican que más del 50% de los casos pertenecen a la categoría "juega", mientras que los azules señalan lo contrario.

El árbol comienza en el nodo raíz, que agrupa al 100% de los datos, y se divide según la variable más informativa: si la persona sigue o no redes sociales. A partir de ahí, cada división sigue un criterio que busca maximizar la pureza de los nodos resultantes, lo que se traduce en divisiones según otras variables como el grupo de edad, el sexo, los hábitos de consumo de vídeos o directos, la educación o la frecuencia con la que se crea contenido.

De forma más específica, si observamos la hoja número 8 (rama izquierda), encontramos un grupo caracterizado por nunca seguir redes sociales, tener entre 15 y 24 años, ser mujeres. En este nodo, solo el 32% de los individuos pertenecen a la categoría "juega", lo que indica una baja probabilidad de juego en este perfil. Por el contrario, la hoja número 7 (rama derecha) agrupa a personas que sí siguen redes, ven vídeos al menos una vez por semana o a diario, y en ese grupo, el 92% de los casos corresponden a personas que juegan, siendo este el nodo con mayor proporción de jugadores en todo el árbol.

Este tipo de visualización permite entender con claridad cómo distintos factores interactúan para explicar la variable objetivo, y cómo el modelo va afinando progresivamente sus decisiones para identificar los perfiles más o menos propensos a jugar a videojuegos.

Construimos a continuación un árbol con las mismas propiedades que el anterior, pero con el criterio de entropía:

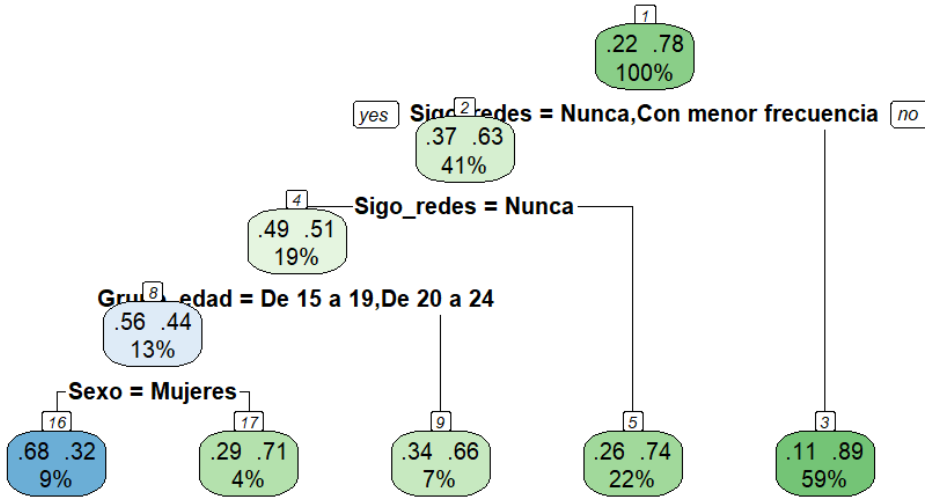


Ilustración 31 Árbol de clasificación con criterio de Entropía

La función ha generado un árbol con una profundidad de 4 y con 5 hojas, cada uno de los cuales muestra proporciones diferentes de evento. En este caso, si nos fijamos en las hojas más “extremas”, la primera es igual que en el caso anterior, podemos concluir que las personas que nunca siguen redes, en un grupo de edad de 15 a 24 y que son mujeres son las menos propensas a jugar, pues sólo el 32% de las mismas lo hace (hoja 16). Por el contrario, las personas que sí que siguen redes Al menos 1 vez cada 15 días, Al menos 1 vez por semana o Todos los días son las más probables a jugar, pues el 89% de ellas lo hace (hoja 3).

Por último, vamos a construir 4 árboles más con otros valores de minbucket y vamos a comparar los 6 modelos obtenidos a partir del área bajo la curva ROC, tanto en entrenamiento como en prueba y ver el tamaño de cada uno:

Tabla 18 Árbol de clasificación comparación de modelos

Modelos	modeloGini1	modeloGini3	modeloGini5	modeloEnt1	modeloEnt3	modeloEnt5
AUC entrenamiento	0.7694071	0.7634523	0.6623251	0.8076528	0.7228978	0.7182960
AUC prueba	0.7403116	0.7501136	0.6422265	0.7364492	0.6980850	0.6951639
Núm. hojas	15	9	3	22	5	4

Nos fijamos en que establecer un minbucket pequeño implica árboles más grandes pero, en este caso, vemos que no resultan especialmente mejores que los pequeños. De nuevo aplicando el principio de parsimonia, podemos concluir que el mejor árbol es el que fija el minbucket en el 3% con el criterio de Gini, pues ofrece un AUC en

entrenamiento prácticamente igual y en prueba mejor que el mayor árbol, pero con menos hojas.

Para finalizar este apartado, vamos a obtener otros estadísticos del árbol ganador a partir de la matriz de confusión. Como ya hicimos en regresión logística binaria, vamos a probar a modificar el punto de corte de 0.5 a 0.78 (la proporción de eventos) para estudiar qué estrategia resulta preferible, en este caso sigue siendo preferible el punto de corte en 0.78. Mostramos los resultados para el conjunto de entrenamiento y el de prueba:

Tabla 19 Árbol de clasificación medidas entrenamiento y prueba

Entrenamiento

Métrica	Media
Accuracy	0,752
Kappa	0,333
Sensitivity	0,804
Specificity	0,563
AUC	0,710

Prueba

Métrica	Media
Accuracy	0,740
Kappa	0,295
Sensitivity	0,800
Specificity	0,523
AUC	0,687

Como podemos ver es un modelo estable ya que no varían mucho los resultados entre entrenamiento y prueba, la precisión disminuye en un 1% y el índice de kappa en un 3% pero sigue siendo de calidad justa, la sensibilidad y especificidad también bajan para el conjunto de datos de prueba.

Obtenemos a continuación la curva ROC y el área (AUC). Se ha optado por representar conjuntamente las curvas en entrenamiento (negro) y en prueba (rojo) para facilitar la comparación.

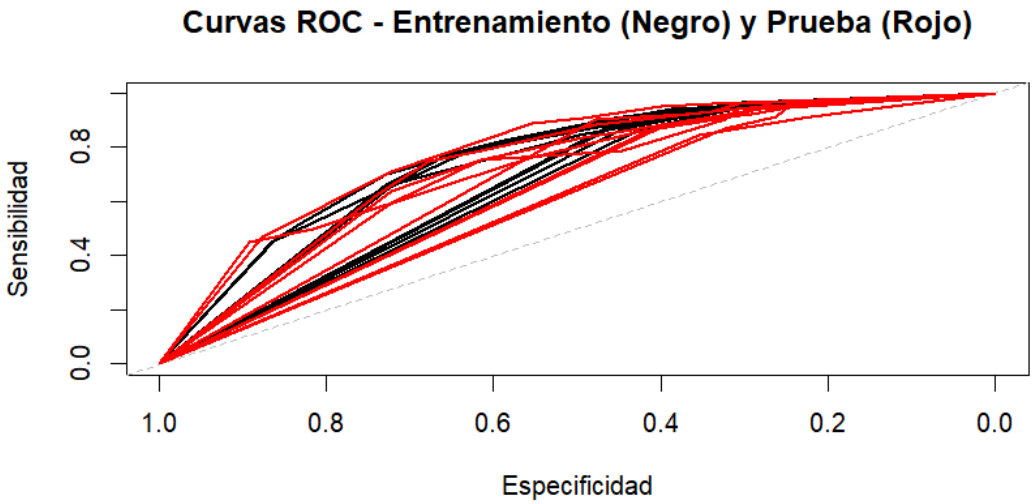


Ilustración 32 Árbol de clasificación curva ROC entrenamiento y prueba

Ambas curvas se sitúan por encima de la diagonal aleatoria, lo que indica una capacidad predictiva aceptable del modelo. Aunque las curvas del conjunto de prueba

tienden a estar ligeramente más cerca de la diagonal, reflejando un rendimiento algo inferior al del entrenamiento, el comportamiento es coherente y no indica un sobreajuste evidente. En conjunto, el modelo logra un equilibrio razonable entre sensibilidad y especificidad en ambos conjuntos.

Importancia de variables

Aunque los árboles de clasificación y regresión se pueden interpretar con gran facilidad, resulta interesante acompañar su representación gráfica (o sus reglas) con un ranking con la importancia de las variables, es decir, indicar qué variables aportan más información a la hora de predecir la variable dependiente.

Como ya se ha visto en la teoría, la importancia de las variables viene dada por la mejora producida en el criterio de división al realizar la partición de los nodos. Por ese motivo, la magnitud de la importancia no es interpretable directamente, pero el ranking que se produce y la distancia entre las variables, sí.

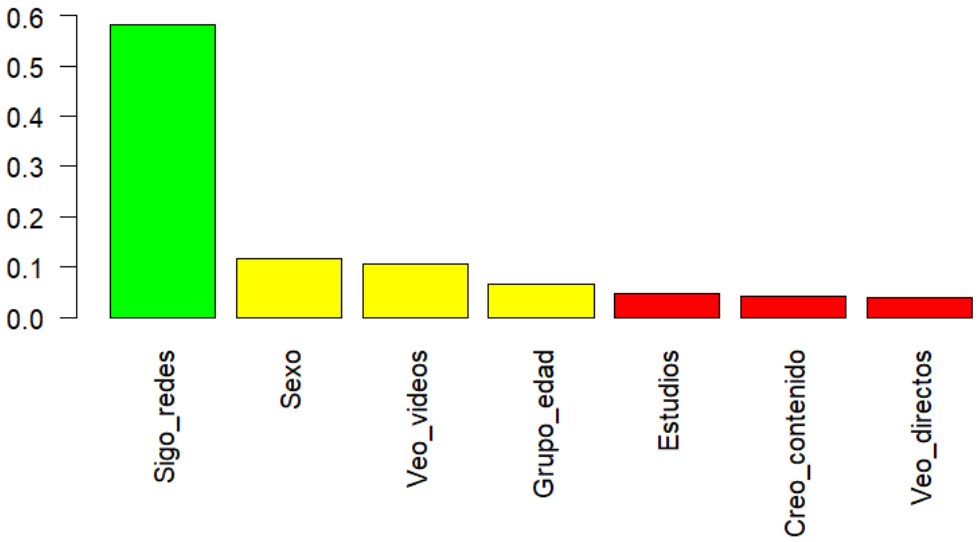


Ilustración 33 Árbol de clasificación e Importancia de variables

Dado que en la construcción de los árboles de regresión se aplica el MSE (que se mide en las unidades de la variable dependiente al cuadrado), la importancia de las variables de este tipo de árboles suele ser muy grande por lo que es recomendable relativizarla para poder interpretar de una manera más sencilla (es lo que se ha hecho en el gráfico de barras). Como se puede observar, la variable más importante es Sigo_redes, que aporta más del 60% de la información del modelo, seguido de Sexo y Veo_videos con algo más del 10%. Observamos que, de las 15 variables independientes del conjunto de datos “solo” aparecen 7 de ellas en el gráfico anterior, lo que implica que las otras 8 no se han utilizado en el árbol.

Poda de árboles

Otro aspecto importante de los árboles de clasificación es su poda. Se entiende por poda el proceso mediante el cual se estudia la posibilidad de eliminar alguna(s) hoja(s) del árbol construido con el objetivo de reducir el sobreajuste.

Cuando se construyen árboles es recomendable establecer un valor de minbucket razonablemente pequeño para que éste tenga la posibilidad de crecer y explotar toda su capacidad predictiva (si directamente se construye un árbol pequeño es posible que no se haya utilizado alguna variable que pudiera mejorarlo) pero, al hacer esto, aumenta la posibilidad de que el árbol detecte patrones en los datos que sean específicos del conjunto de datos de entrenamiento y, por ende, aparezca el sobreajuste.

Como ya se vio en la teoría, para llevar a cabo la poda se debe generar la secuencia de valores críticos de α que dan lugar a un crecimiento/decrecimiento del árbol. Así mismo, para cada uno de los subárboles generados (definidos por el valor de α) se lleva a cabo un proceso de evaluación tanto en los datos originales como con validación cruzada.

Vamos a estudiar la poda de un árbol de clasificación, hay que tener en cuenta que el error representado será el cociente entre la tasa de clasificación errónea del árbol analizado y dicha tasa evaluada en la raíz.

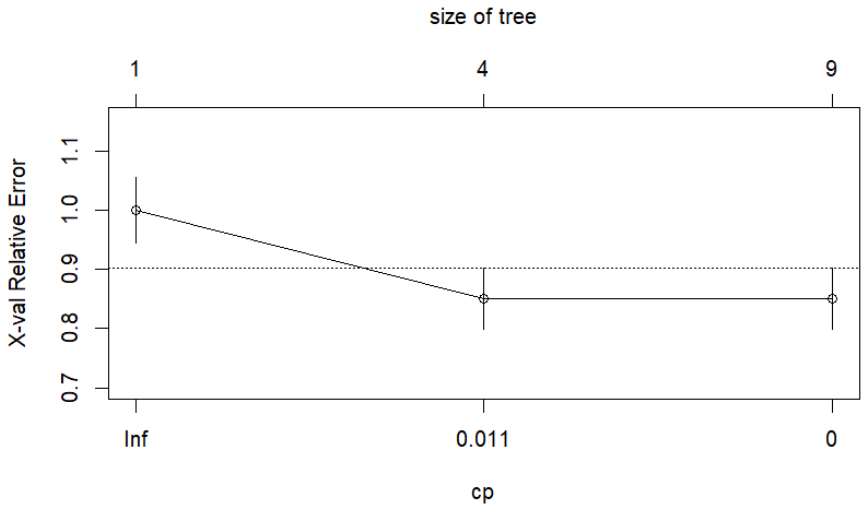


Ilustración 34 Poda de árboles tamaño del árbol

Cuando se trabaja con árboles de clasificación es habitual que no se observe la secuencia completa de subárboles (por ejemplo, en este caso se pasa de un subárbol con una hoja a otro con cuatro hojas a otro de nueve hojas). Esto se debe principalmente al hecho de que los hijos tengan la misma categoría mayoritaria que el padre, lo que se traduce en que el número de observaciones clasificadas de manera incorrecta se mantiene constante. Así mismo, para no alargar el proceso de la poda, la función rpart establece un umbral de *α* de manera que sólo se consideren aquellos valores de cp que sean suficientemente diferentes (su diferencia supere el umbral).

Como se puede observar en la salida anterior, en este caso el tamaño óptimo de la poda no resulta tan evidente. De hecho, utilizando la información que proporciona la línea horizontal de la representación gráfica, podríamos concluir que el árbol con cuatro hojas sería el óptimo pues es el primer punto que se sitúa debajo de la misma aunque tiene el mismo valor que el árbol maximal, pero utilizaremos el de menor tamaño ya que no hay una gran mejora.

Es importante destacar que la poda de los árboles de clasificación llevada a cabo de este modo (basada en la tasa de clasificación errónea) puede no ser la mejor opción,

sobre todo si se está trabajando con una variable dependiente desbalanceada debido a que esta medida es un poco “tosca”. Para trabajar con el AUC o el índice Kappa se debe recurrir a la función xpred.rpart, que nos permite acceder a los resultados internos de la validación cruzada, para así poder calcular la medida deseada. Por cómo está diseñada esta función, es necesario volver a construir el árbol justo antes de utilizarla para evitar problemas.

Con estos resultados obtenemos información sobre todas las observaciones en el conjunto de datos de entrenamiento (primera dimensión) y los posibles valores del parámetro α (segunda dimensión). En cuanto a la tercera dimensión, esta contiene información de distinto tipo: Para el valor 1, se muestra la categoría predicha según su máxima probabilidad, que han sido numeradas de 1 en adelante según el orden alfabético. Gracias a esta información, se puede reconstruir la matriz de confusión para cada valor de *α* con la información de la validación cruzada. Para los valores 2 y 3, el número de observaciones de las correspondientes categorías que hay en la hoja de predicción (de nuevo, las categorías son ordenadas alfabéticamente). Para el valor 4, el número total de observaciones ubicadas en la hoja de predicción.

Gracias a los últimos apartados, se pueden calcular las probabilidades a posteriori de las distintas categorías de la variable dependiente y, a partir de ellas, el área bajo la curva ROC. En el código siguiente, se va a recurrir a un bucle que recorra los valores críticos de *α* y permita calcular el AUC y el índice Kappa, para poder así podar correctamente el árbol.

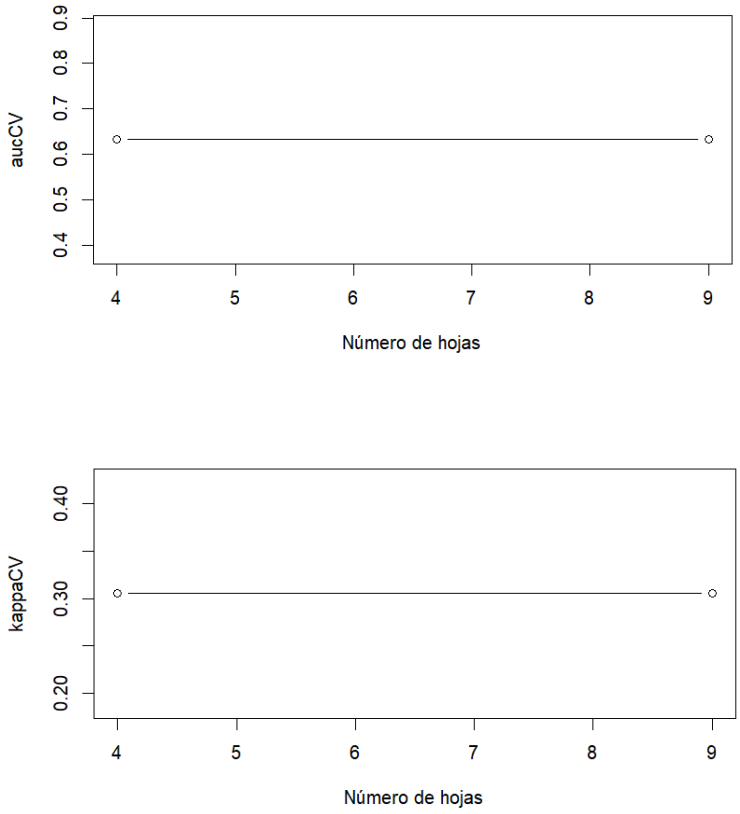


Ilustración 35 Poda de árboles AUC y Kappa para distintos árboles

Como Podemos ver en estos gráficos toman los mismos valores de AUC y Kappa en ambos casos con 4 hojas y con 9, por lo que usamos el menor número de hojas posibles

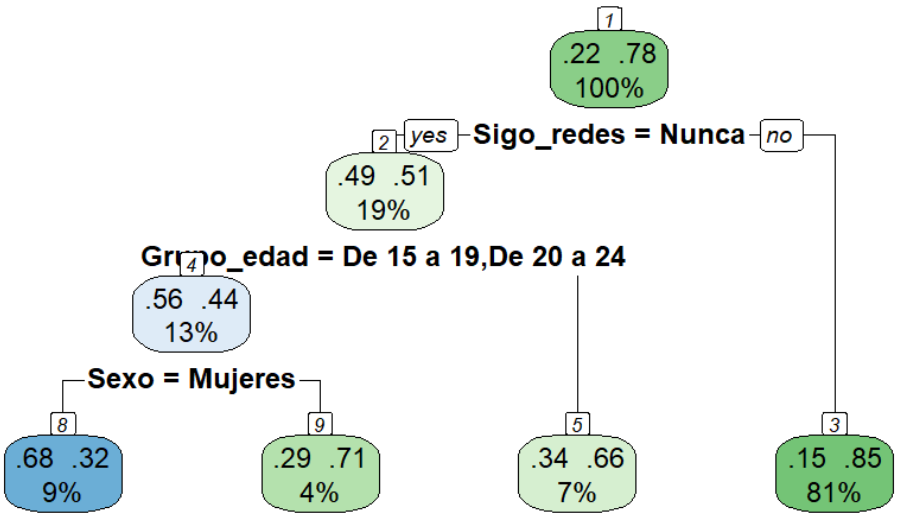


Ilustración 36 Árbol podado

Como podemos observar, este árbol es más pequeño que el maximal y tiene 5 hojas menos que el que habíamos considerado mejor previamente, tiene un parecido razonable con el árbol creado con el criterio de entropía con un 3% de minbucket, ya que tiene una hoja menos que ese y utiliza las mismas variables para crear el árbol.

Para finalizar, vamos a obtener otros estadísticos del árbol ganador a partir de la matriz de confusión. Usaremos el punto de corte de 0,78 y mostramos los resultados para el conjunto de entrenamiento y el de prueba:

Tabla 20 Poda de árboles medidas entrenamiento y prueba

Entrenamiento

Métrica	Media
Accuracy	0,778
Kappa	0,319
Sensitivity	0,871
Specificity	0,437
AUC	0,664

Prueba

Métrica	Media
Accuracy	0,776
Kappa	0,312
Sensitivity	0,871
Specificity	0,429
AUC	0,656

Como podemos ver es un modelo estable ya que no varían mucho los resultados entre entrenamiento y prueba, la precisión se mantiene y el índice de kappa también pero sigue siendo de calidad justa, la especificidad baja para el conjunto de datos de prueba, esta vez hemos tenido una mejor sensibilidad y una peor especificidad, esto indica que el modelo es más efectivo en identificar los casos positivos (clase "Sí"), pero lo hace a expensas de cometer más errores al clasificar los casos negativos (clase "No").

Obtenemos a continuación la curva ROC y el área (AUC). Se ha optado por representar conjuntamente las curvas en entrenamiento (negro) y en prueba (rojo) para facilitar la comparación.

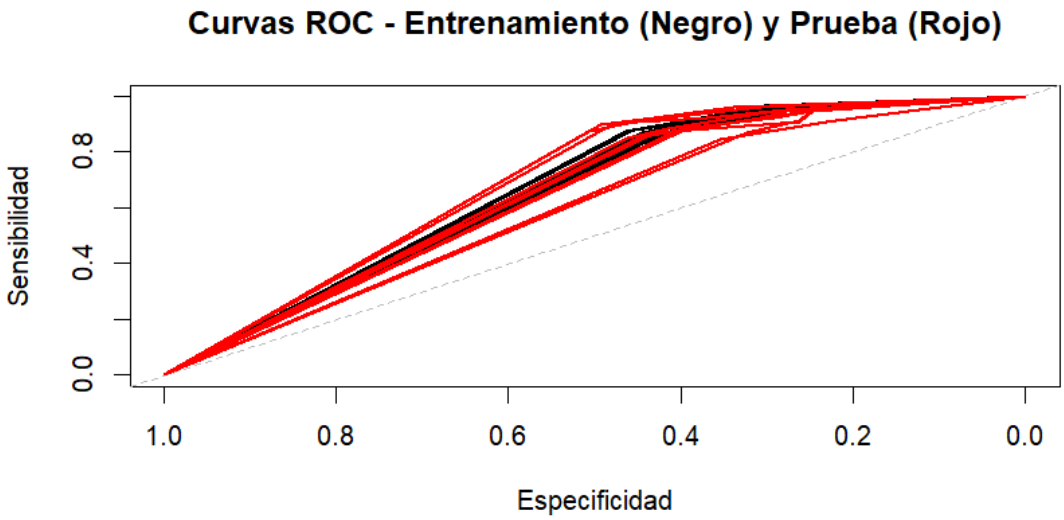


Ilustración 37 Poda de árboles curva ROC entrenamiento y prueba

En ambos casos, las curvas se sitúan por encima de la diagonal aleatoria, lo cual indica que el modelo tiene cierta capacidad predictiva. Sin embargo, las curvas no se acercan completamente al vértice superior izquierdo, lo que sugiere que la discriminación entre clases no es perfecta. La similitud entre las curvas de entrenamiento y prueba sugiere que el modelo no está sobreajustado, aunque su rendimiento general parece algo limitado, como indica también el área bajo la curva (AUC).

En resumen hemos usado la poda de árboles para simplificar el modelo original, reduciendo su complejidad y el riesgo de sobreajuste, lo que ha resultado en un modelo más generalizable con una sensibilidad mejorada, aunque a costa de una menor especificidad. Al analizar los resultados, observamos que en el modelo podado la precisión es ligeramente superior tanto en el conjunto de entrenamiento como en el de prueba, mientras que para el modelo sin podar el índice Kappa muestra una mejora moderada en entrenamiento pero no tan significativa en prueba. Esto sugiere que el modelo podado logra un mejor balance entre las clases en el conjunto de entrenamiento, aunque su capacidad para generalizar sigue siendo comparable a la del modelo sin podar.

Modelos Bagging y Random Forest

Bagging

Esta técnica se basa en generar múltiples muestras Bootstrap (es decir, submuestras con remuestreo de los datos originales) para así poder obtener árboles con distintas formas que logren capturar todo el potencial predictivo de las variables explicativas para, a continuación, agregar la información de estos para obtener un único valor de predicción. De hecho, el nombre de Bagging procede de la unión de Bootstrap y Aggregation. Los datos OOB en Bagging son las observaciones que no se utilizan para

entrenar un modelo específico y que sirven para validar ese modelo, permitiendo evaluar su desempeño de manera eficiente durante el proceso de entrenamiento sin necesidad de un conjunto de prueba adicional.

Una vez generado el modelo, podemos “imprimirlo” para hacernos una primera idea de los resultados. En este caso, se ha generado un bagging para clasificación con 500 árboles con una tasa de error (basada en los datos OOB) del 18.41%. Así mismo, observamos que nos ofrece el error obtenido en cada uno de los niveles (serían los equivalente a 1 − sensibilidad y 1 − especificidad), así como la matriz de confusión (también usando los datos OOB). Como era de esperar, como los eventos son más numerosos que los no eventos, el error que se comete para estos últimos es mayor.

Lo primero que se debe determinar es el número óptimo de árboles (para lo cual recurrimos a la información sobre las observaciones OOB). En el caso de variables dependientes cualitativas, la función randomForest almacena información sobre la tasa de fallo para cada posible número de árboles utilizando para ello las observaciones OOB.

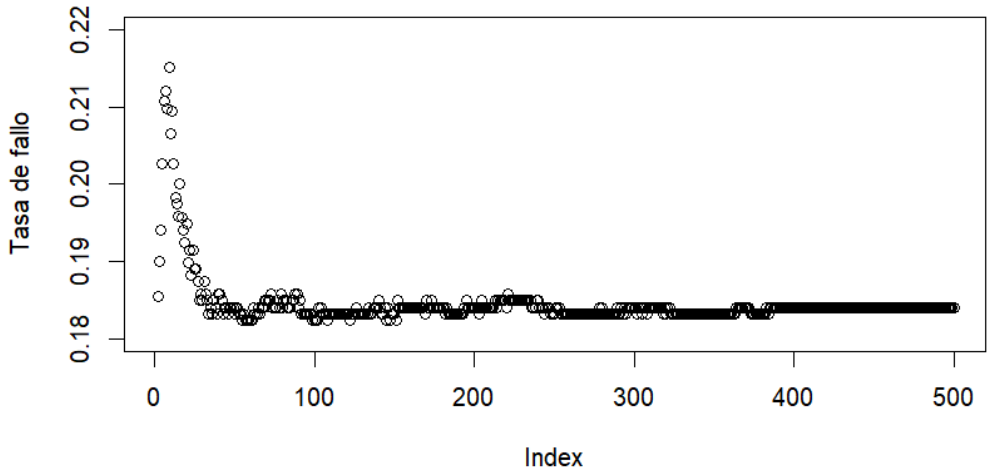


Ilustración 38 Bagging Tasa de fallo por número de árbol

Podemos ver cómo mejora la calidad de las predicciones en cuanto se pasa de un árbol a una cantidad mayor de los mismos (aunque existan ciertas subidas y bajadas, se observa una tendencia decreciente, aunque el menor punto es el primero). Cuando se evalúan los modelos con la tasa de clasificación errónea, debido a que esta medida es más tosca, este comportamiento “menos suave” es bastante frecuente. Teniendo en cuenta la información anterior, podríamos establecer que el número óptimo de árboles es 1 pues en ese momento se consigue el valor mínimo, si no a partir de 100 ese valor se estabiliza.

No obstante, como ya se ha mencionado en múltiples ocasiones, cuando se trabaja con variables desbalanceadas (y no solo en ese caso), es recomendable evaluar los modelos (y usar dicha información para determinar la parametrización óptima) con otras métricas, como el AUC o el índice Kappa.

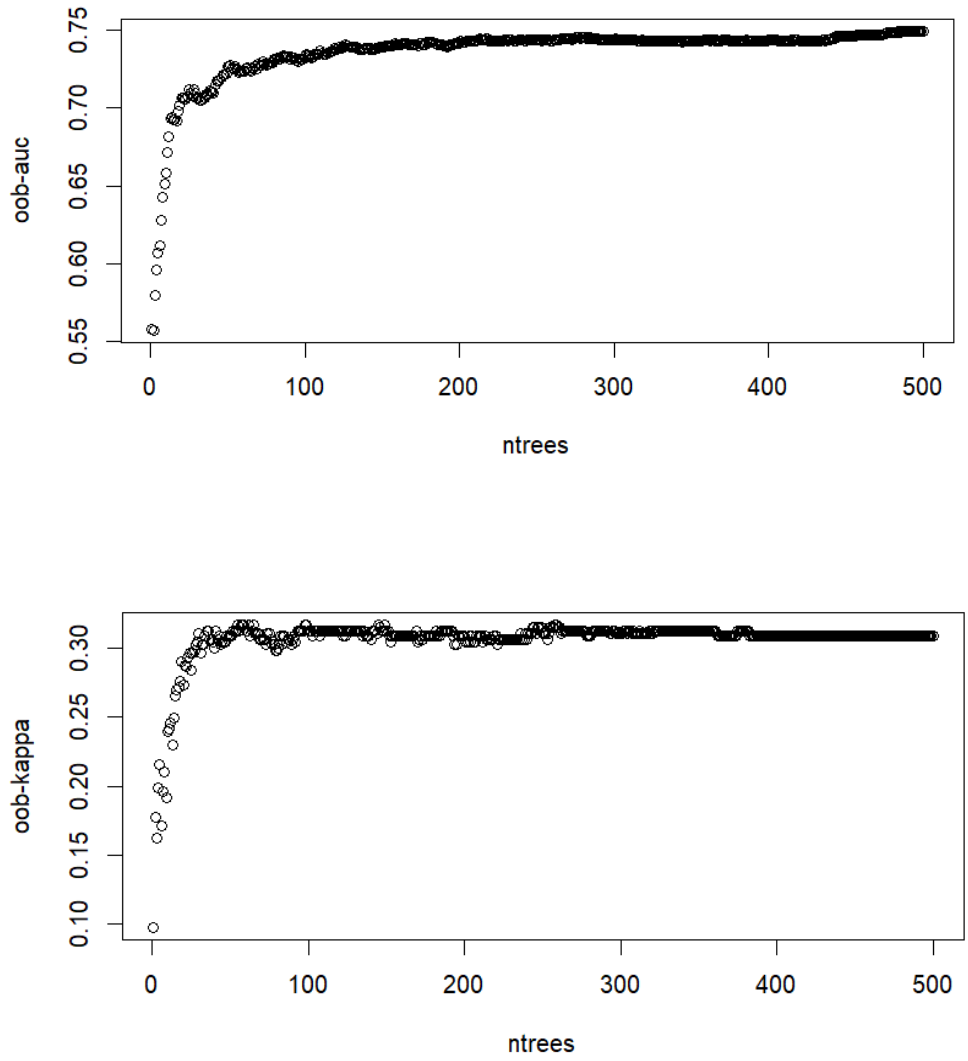
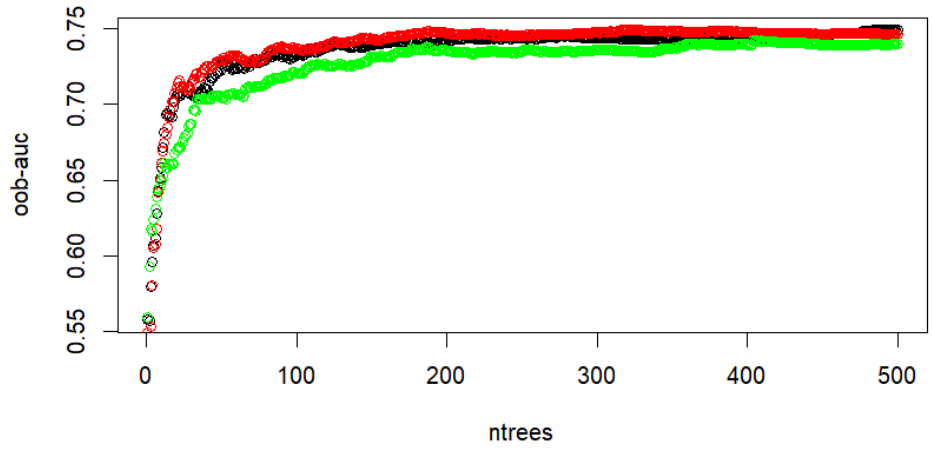


Ilustración 39 Bagging AUC y Kappa por número de árbol

A la vista de los resultados, podemos concluir que un valor de ntree de 200 da lugar a buenos resultados, en el sentido de que no se observa una gran mejora si se aumenta. Podemos observar también que el modelo resultante tendrá un AUC de aproximadamente 0.75 y un índice Kappa alrededor de 0.3, lo que implica que la calidad del modelo es moderada.

Veamos, por último, si se puede mejorar este modelo disminuyendo o aumentando el tamaño de las hojas:



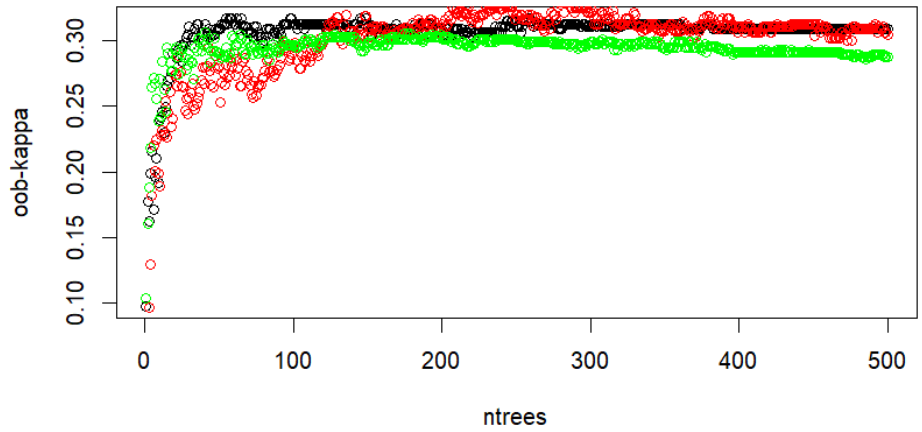


Ilustración 40 Bagging AUC y Kappa por número de árbol con diferentes tamaños de hojas

En estos gráficos vemos el AUC y el Kappa para la serie original, de tamaño reducido y de tamaño ampliado (negra, roja y verde), observamos que son muy similares pero para el AUC es mejor la serie roja desde el principio y para el kappa es mejor la roja a partir de 150, así que nos vamos a quedar con esta.

Como podemos observar, se han mejorado todos los indicadores de calidad con respecto al modelo bagging que se construyó en primer lugar, obteniendo una tasa de error del 19.82% y una sensibilidad y especificidad de 0.94 y 0.30, por lo que la especificidad no es buena, respectivamente (en las salidas aparecen los opuestos a estas cantidades) basándose en los datos OOB. Obtenemos a continuación una estimación de la calidad en términos de otros indicadores en la partición de entrenamiento y prueba:

Tabla 21 Bagging medidas entrenamiento y prueba

Entrenamiento

Métrica	Media
Accuracy	0,875

Prueba

Métrica	Media
Accuracy	0,705

Kappa	0,679
Sensitivity	0,863
Specificity	0,919
AUC	0,960

Kappa	0,300
Sensitivity	0,718
Specificity	0,657
AUC	0,749

Con el punto de corte en 0.78 vemos como los estadísticos mejoran drásticamente, en el conjunto de datos de entrenamiento tenemos una precisión del 88% y un Kappa de 0.68, con una sensibilidad y especificidad de 0.86 y 0.92 respectivamente, con un área bajo la curva de 0.96. Para el conjunto de datos de entrenamiento todos los estadísticos se reducen, siendo la precisión del 71% y un Kappa de 0.30 y la sensibilidad y especificidad de 0.72 y 0.66 respectivamente, con un área bajo la curva de 0.75, lo cual indica que el modelo ya no distingue tan claramente entre clases como en entrenamiento. Esta diferencia entre ambos conjuntos puede ser un indicio de sobreajuste, es decir, que el modelo ha aprendido demasiado bien los patrones del entrenamiento, perdiendo capacidad de generalización.

Obtenemos a continuación la curva ROC y el área (AUC). Se ha optado por representar conjuntamente las curvas en entrenamiento (negro) y en prueba (rojo) para facilitar la comparación.

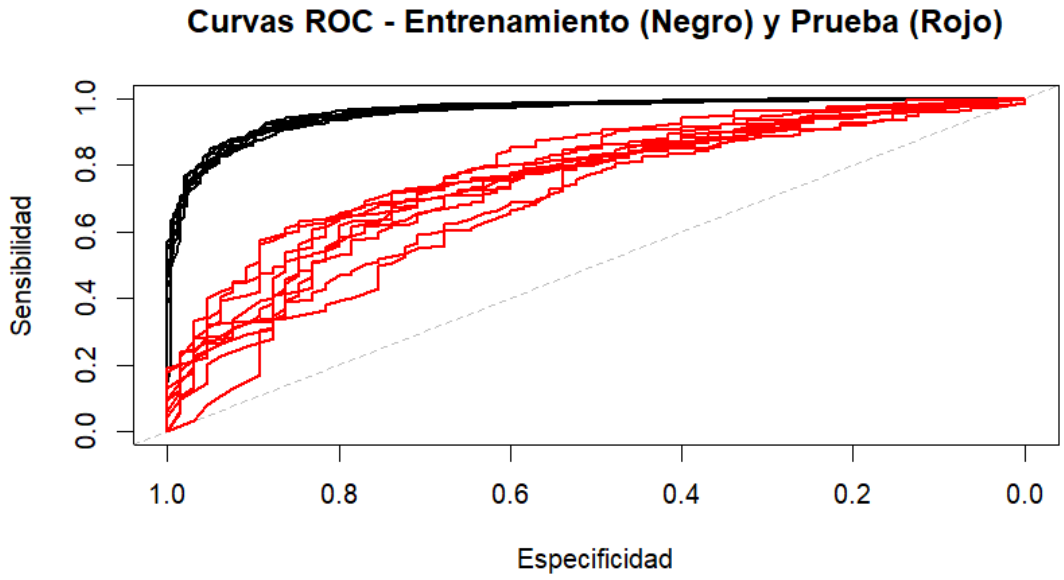


Ilustración 41 Bagging curva ROC entrenamiento y prueba

En el conjunto de entrenamiento, los valores de AUC son consistentemente muy altos. Esto indica que el modelo tiene una excelente capacidad de discriminación en esos datos: es decir, clasifica de manera muy precisa entre los casos positivos y negativos. Las curvas ROC correspondientes a este conjunto están muy próximas al vértice superior izquierdo del gráfico, lo cual es característico de modelos con un rendimiento casi perfecto. Por el contrario, en el conjunto de prueba, estos valores son menores siendo las curvas ROC más dispersas y cercanas a la diagonal. Aunque estos valores siguen siendo aceptables, muestran una disminución significativa respecto al entrenamiento. Esto sugiere que el modelo pierde capacidad de generalización cuando se enfrenta a nuevos datos, lo que refuerza la hipótesis de cierto sobreajuste.

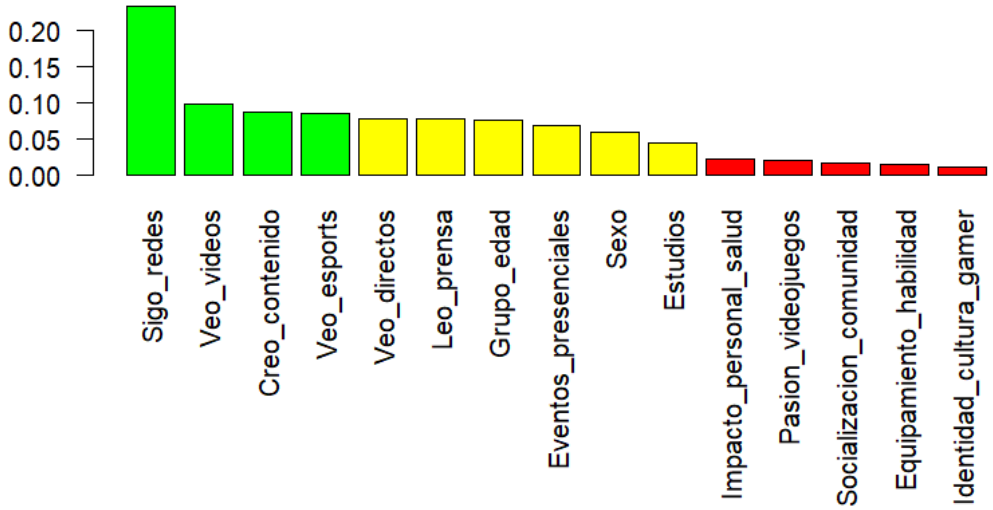


Ilustración 42 Bagging e Importancia de variables

El gráfico de importancia de variables muestra de forma visual qué características tienen mayor peso en el modelo de clasificación. En este caso, “Sigo_redes” destaca como la variable más influyente, con mucha diferencia respecto al resto, lo que indica que seguir redes sociales está fuertemente relacionado con la probabilidad de que una persona juegue. Le siguen en importancia variables como “Veo_videos”, “Creo_contenido” y “Veo_esports”, lo que sugiere que el consumo y producción de contenido digital relacionado con videojuegos también tiene un papel relevante en la predicción. Estas cuatro variables han sido representadas en color verde para destacar su peso destacado en el modelo.

Las variables intermedias, como “Veo_directos”, “Leo_prensa”, “Grupo_edad”, “Eventos_presenciales”, “Sexo” y “Estudios”, aparecen en amarillo, indicando que también tienen cierta influencia pero no tan marcada. Por último, las variables con menor contribución al modelo, como “Impacto_personal_salud”, “Pasion_videojuegos”, “Socializacion_comunidad”, “Equipamiento_habilidad” e “Identidad_cultura_gamer”, están coloreadas en rojo, ya que su capacidad para ayudar al modelo a tomar decisiones es bastante limitada. Esto puede deberse a que su información es más redundante o menos diferenciadora respecto a la variable objetivo.

Random Forest

Esta técnica es una generalización del modelo bagging en el cual se introduce una nueva fuente de aleatoriedad que permite obtener árboles menos similares, lo que se traduce en una reducción de la correlación entre los mismos y, a su vez, en una mejora de la capacidad predictiva.

Comenzamos creando un RF de 500 árboles con un tamaño de hoja equivalente al 5% de los datos y ntree y mtry por defecto (en este caso, por tratarse de un RF para clasificación, este último será la raíz cuadrada del número de variables explicativas disponibles).

Se ha creado un modelo RF para clasificación, con 500 árboles y seleccionando únicamente 3 variables en cada división de nodos, lo que da lugar a un modelo con una tasa de error de 19.49%. Con un AUC y un Kappa así:

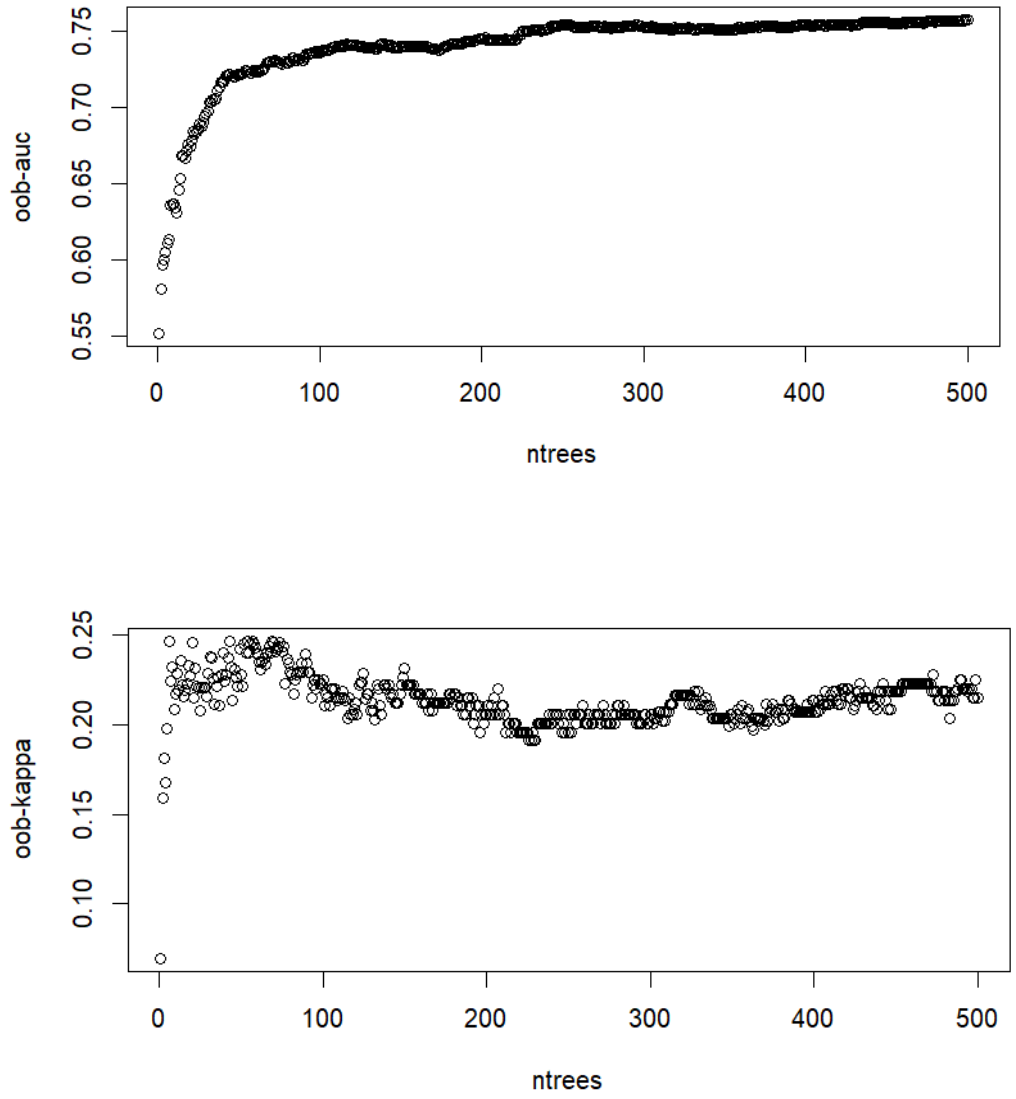


Ilustración 43 Random Forest AUC y Kappa

Vemos como el AUC mejora con la cantidad de árboles, teniendo un valor de 0.75 a partir de los 200 árboles, pero el Kappa no ya que su mejor valor está en torno a los 70 con un valor de 0.25. Vamos a probar a continuación a reducir el tamaño de hoja para determinar si puede dar lugar a mejores resultados.

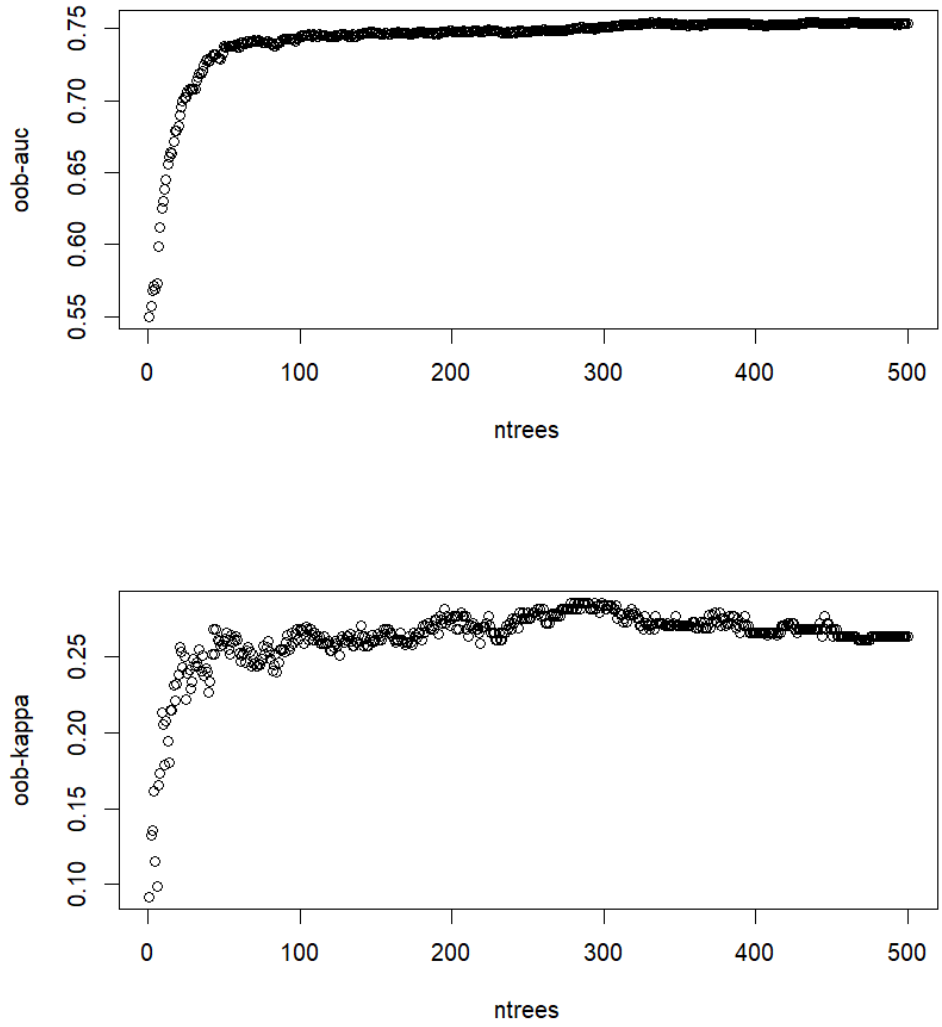


Ilustración 44 Random Forest AUC y Kappa con menor tamaño de hoja

Vemos cómo disminuir el tamaño de hoja ha supuesto cierta mejora en el AUC pero sobre todo en el índice Kappa. El mejor valor se alcanza cuando el número de árboles ronda los 250.

Gracias a los dos modelos creados previamente, hemos podido concluir que el número apropiado de árboles se puede fijar en 250 y que el tamaño de hoja equivalente al 1% de los datos funciona de manera adecuada. Este último parámetro se podría fijar en un valor inferior pero eso implicaría un mayor tiempo de cálculo sin que implique necesariamente una mejora en la capacidad predictiva. El último parámetro que nos faltaría por cambiar es mtry. Algunos autores sugieren probar 4-5 valores diferentes entre 2 y el número total de variables disponibles. Para ello, vamos a crear un bucle que recorra los distintos posibles valores y guarde los resultados obtenidos. A continuación, construimos un gráfico en el que representamos el AUC y Kappa obtenido frente al valor de mtry:

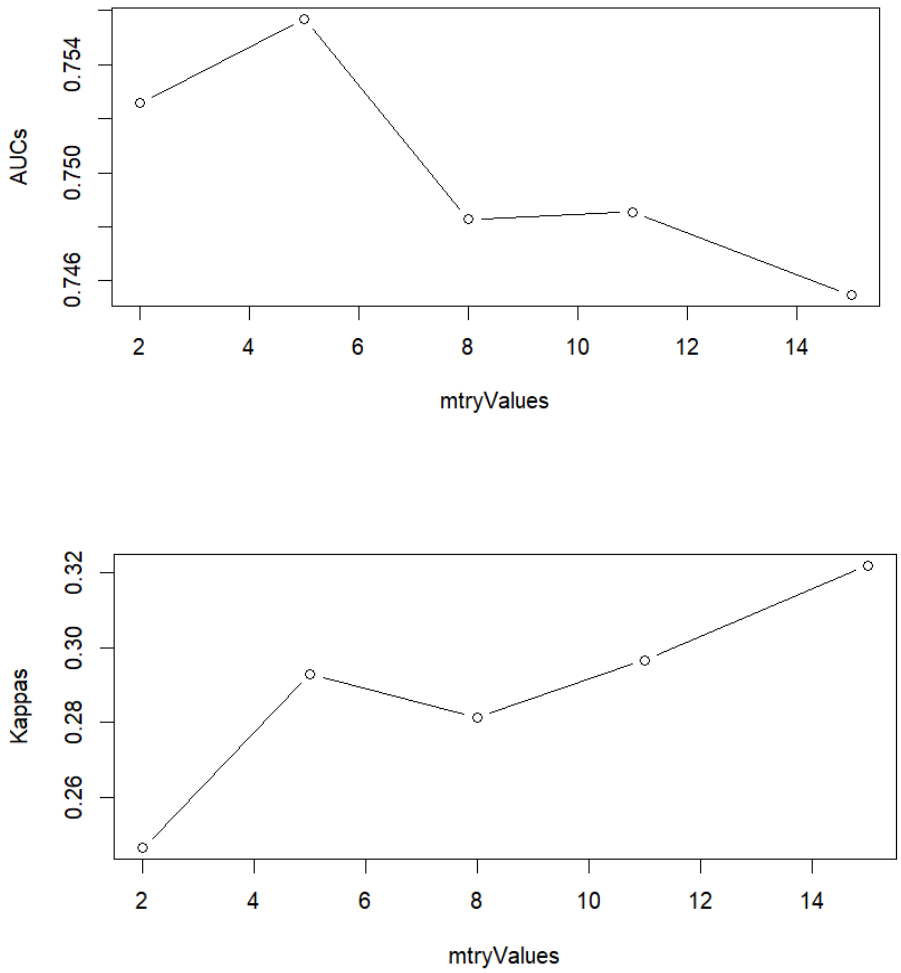


Ilustración 45 Random Forest AUC y Kappa respecto a mtry

Tal y como podemos observar, los dos indicadores no coinciden en el valor óptimo de mtry. No obstante, teniendo en cuenta los dos indicadores (pero dándole más peso al AUC pues es una medida más sensible), podemos concluir que 5 es un buen valor de compromiso. Observamos, además, que incluir esta nueva fuente de aleatoriedad permite mejorar los resultados con respecto al bagging (cuando mtry es igual a 15) tanto en AUC como en índice Kappa.

Como podemos observar, se han mejorado todos los indicadores de calidad con respecto al modelo random forest que se construyó en primer lugar, obteniendo una tasa de error del 18.99% y una sensibilidad y especificidad de 0.96 y 0.28, por lo que la especificidad no es buena, respectivamente (en las salidas aparecen los opuestos a estas cantidades) basándose en los datos OOB, hay que tener en cuenta que estos datos son con un punto de corte de 0.5 por lo que ahora daremos los resultados con nuestro punto de corte en 0.78. Obtenemos a continuación una estimación de la calidad en términos de otros indicadores en la partición de entrenamiento y prueba:

Tabla 22 Random Forest medidas entrenamiento y prueba

Entrenamiento	Prueba

Métrica	Media
Accuracy	0,879
Kappa	0,679
Sensitivity	0,879
Specificity	0,879
AUC	0,950

Métrica	Media
Accuracy	0,726
Kappa	0,314
Sensitivity	0,757
Specificity	0,614
AUC	0,760

Con el punto de corte en 0.78 vemos como los estadísticos mejoran drásticamente, en el conjunto de datos de entrenamiento tenemos una precisión del 88% y un Kappa de 0.68, que se considera bueno, con una sensibilidad y especificidad de 0.88, con un área bajo la curva de 0.95. Para el conjunto de datos de entrenamiento todos los estadísticos se reducen, siendo la precisión del 73% y un Kappa de 0.31, que se considera justo y la sensibilidad y especificidad de 0.76 y 0.61 respectivamente, con un área bajo la curva de 0.76.

Obtenemos a continuación la curva ROC y el área (AUC). Se ha optado por representar conjuntamente las curvas en entrenamiento (negro) y en prueba (rojo) para facilitar la comparación.

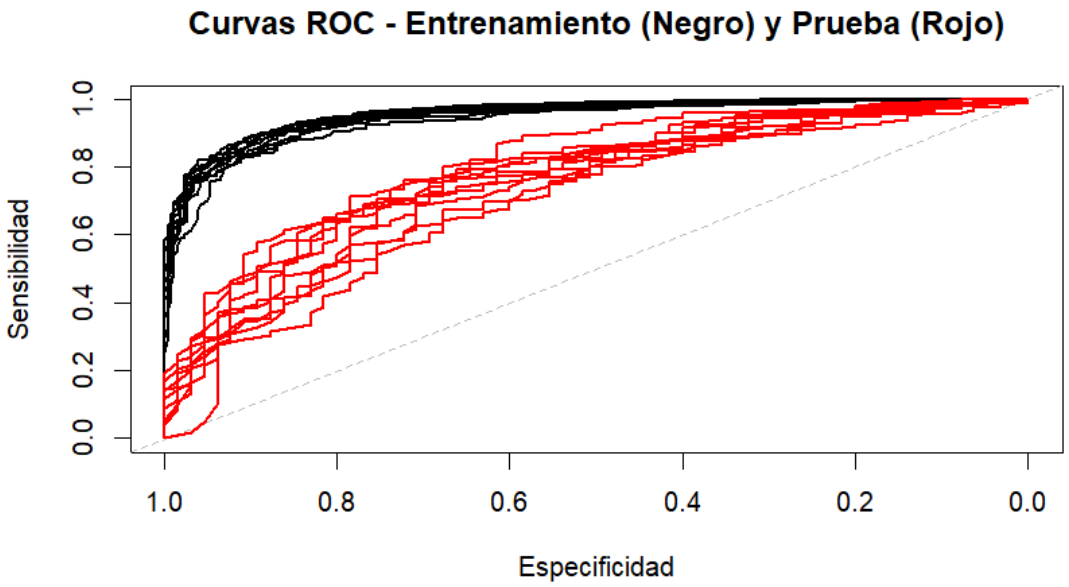


Ilustración 46 Random Forest curva ROC entrenamiento y prueba

En el conjunto de entrenamiento, las AUC se mantienen consistentemente elevadas, lo que indica una excelente capacidad de discriminación entre clases. El modelo es capaz de clasificar con gran precisión los datos sobre los que ha sido entrenado, por lo que sus curvas ROC son muy próximas al vértice superior izquierdo, característica de modelos con un rendimiento casi perfecto.

Sin embargo, en el conjunto de prueba, estos valores son menores siendo las curvas ROC más dispersas y cercanas a la diagonal. Aunque estos resultados siguen siendo aceptables, suponen una caída considerable respecto al rendimiento en entrenamiento, lo que sugiere que el modelo presenta sobreajuste. Es decir, ha aprendido demasiado bien los patrones del conjunto de entrenamiento, pero su capacidad de generalización a nuevos datos se ve comprometida. En consecuencia,

las curvas ROC en test están más próximas a la diagonal, reflejando una discriminación más limitada.

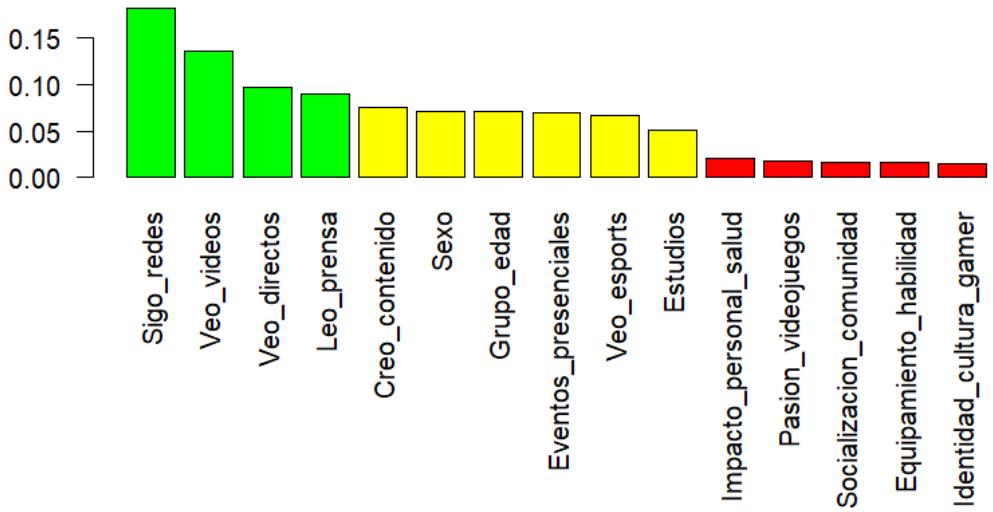


Ilustración 47 Random Forest e Importancia de variables

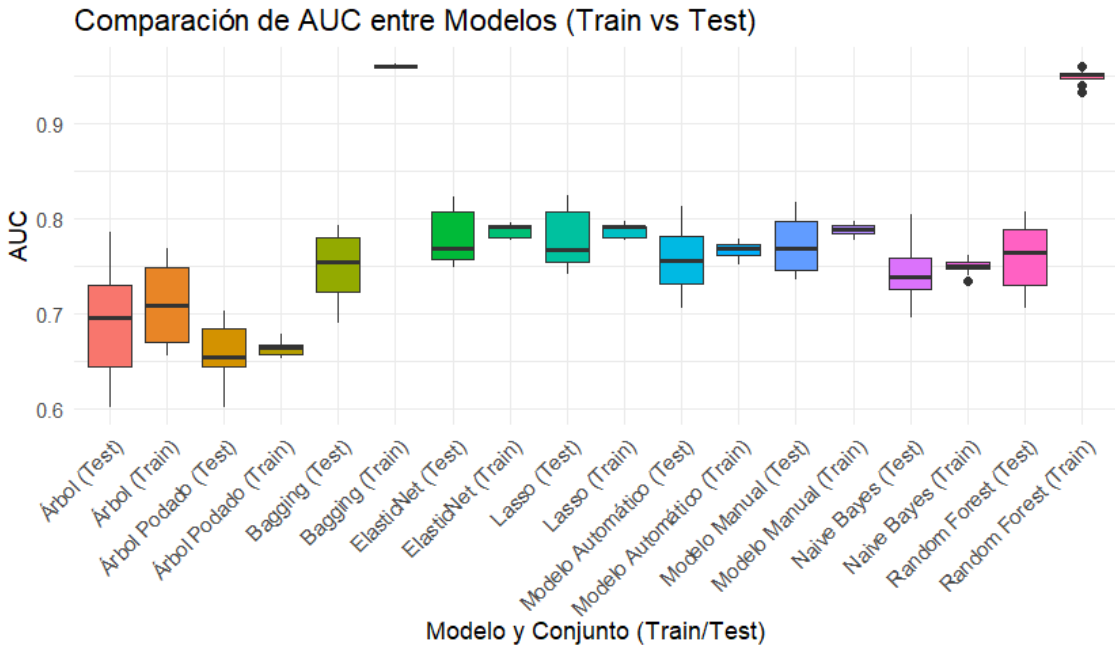
El gráfico de importancia de variables muestra de forma visual qué características tienen mayor peso en el modelo de clasificación. Las cuatro variables más influyentes, destacadas en verde, son "Sigo_redes", "Veo_videos", "Veo_directos" y "Leo_prensa", que en conjunto representan los principales canales de exposición y consumo de contenido relacionado con videojuegos. En especial, "Sigo_redes" sobresale como la más determinante, lo que sugiere un fuerte vínculo entre la interacción en redes sociales y el interés en jugar.

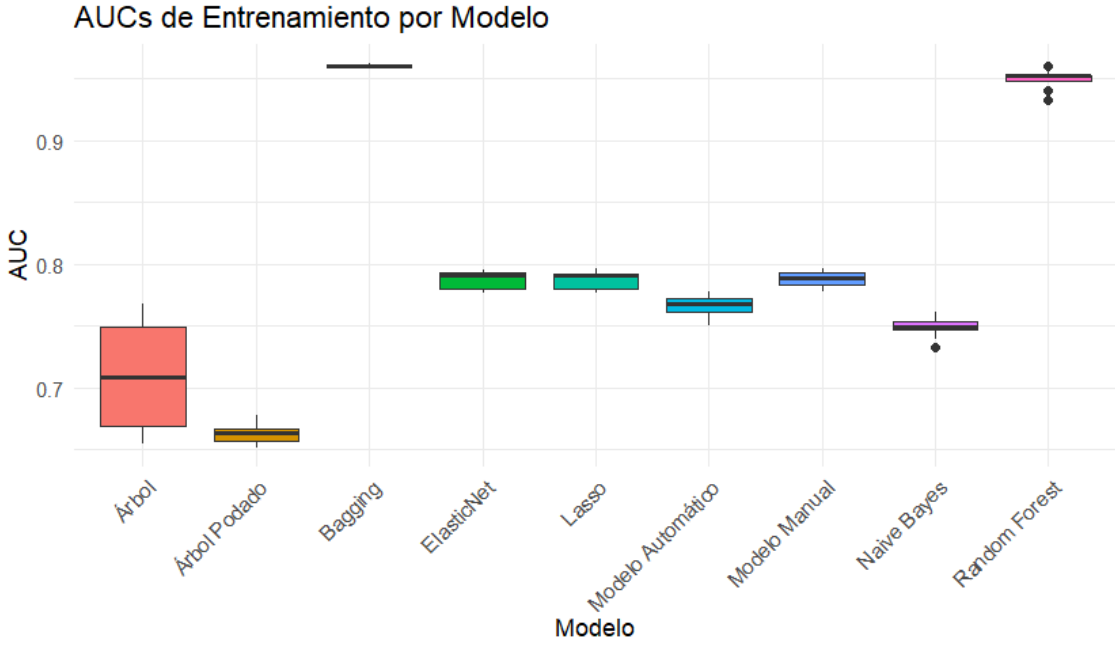
Las variables de importancia intermedia, marcadas en amarillo, abarcan aspectos sociodemográficos como el sexo, la edad, y el nivel de estudios, junto con otras conductas de consumo y participación como "Eventos_presenciales", "Veo_esports", y "Creo_contenido". Estas aportan valor al modelo, aunque en menor medida, y reflejan cómo el contexto personal y los hábitos relacionados con el ecosistema gamer influyen en la decisión de jugar.

Por último, las variables con menor peso predictivo (en rojo) incluyen factores personales como la pasión por los videojuegos, el impacto percibido en la salud, y aspectos más subjetivos como la identidad gamer o la socialización en comunidad. Su baja importancia sugiere que, aunque pueden tener relación con la experiencia del jugador, no son determinantes para predecir el comportamiento de juego dentro de este modelo.

Resultados

Comparación entre los modelos


Comparación de AUC entre Modelos (Train vs Test)
AUC
0.9
0.8
0.7
0.6
Árbol (Test)
Árbol (Train)
Árbol Podado (Test)
Árbol Podado (Train)
Bagging (Test)
Bagging (Train)
ElasticNet (Test)
ElasticNet (Train)
Lasso (Test)
Lasso (Train)
Modelo Automático (Test)
Modelo Automático (Train)
Modelo Manual (Test)
Modelo Manual (Train)
Naive Bayes (Test)
Naive Bayes (Train)
Random Forest (Test)
Random Forest (Train)
Modelo y Conjunto (Train/Test)


AUCs de Entrenamiento por Modelo
AUC
0.9
0.8
0.7
Árbol
Árbol Podado
Bagging
ElasticNet
Lasso
Modelo Automático
Modelo Manual
Naive Bayes
Random Forest
Modelo

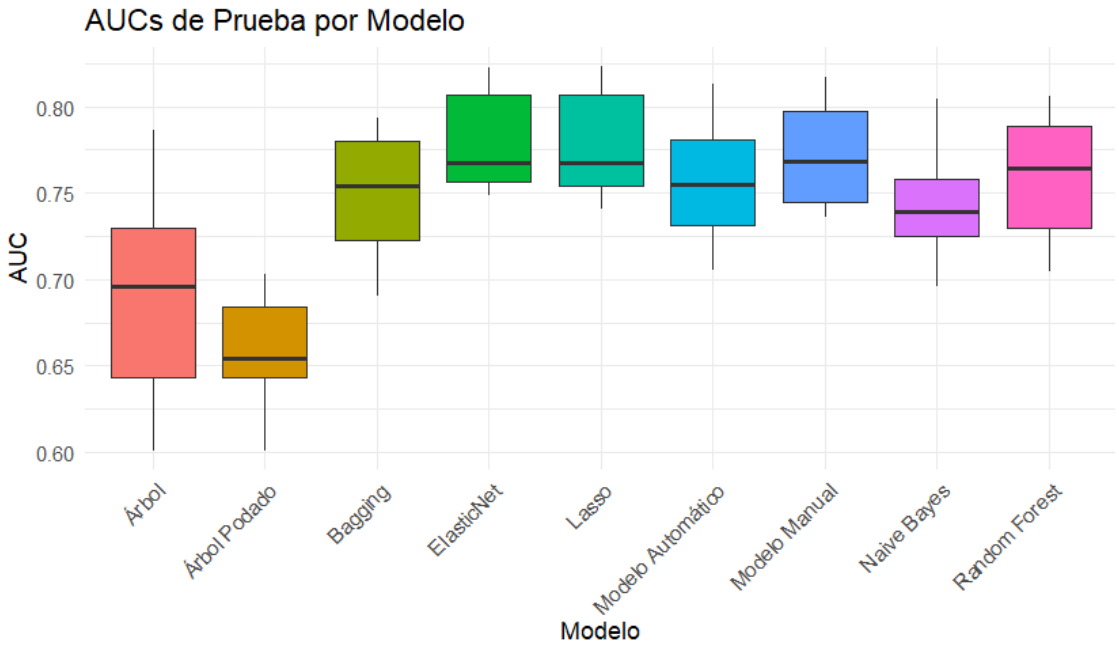


Ilustración 48 Comparación entre los modelos según su AUC en entrenamiento y prueba

El gráfico de caja (boxplot) compara el rendimiento de nueve modelos de clasificación en términos de su AUC, evaluados tanto en los conjuntos de entrenamiento como de prueba. Los modelos considerados son: Modelo Manual, Modelo Automático, Naive Bayes, Lasso, Elastic Net, Árbol, Árbol Podado, Bagging y Random Forest.

Los modelos Elastic Net y Lasso muestran un desempeño sólido y consistente, con valores de AUC elevados tanto en entrenamiento como en prueba, y una baja dispersión. Esto sugiere que estos modelos generalizan bien y tienen un buen equilibrio entre sesgo y varianza.

El Modelo Manual presenta un rendimiento competitivo en ambos conjuntos, aunque con algo más de variabilidad y una leve caída del AUC en prueba, lo que podría indicar cierta tendencia al sobreajuste.

El Modelo Automático, por su parte, tiene un rendimiento algo inferior, pero más estable entre entrenamiento y prueba. Esto podría deberse a una menor complejidad del modelo o a un enfoque más conservador en su construcción.

Naive Bayes obtiene resultados modestos, con AUC más bajos y mayor dispersión, especialmente en el conjunto de prueba. Esto sugiere que el modelo tiene dificultades para capturar relaciones complejas en los datos.

En cuanto a los modelos basados en árboles, los modelos Árbol y Árbol Podado presentan los AUC más bajos y una gran variabilidad, lo que indica un desempeño pobre y posiblemente una alta sensibilidad a los datos de entrenamiento.

En contraste, los modelos Bagging y Random Forest alcanzan valores de AUC muy altos en el conjunto de entrenamiento, lo que sugiere una gran capacidad para ajustarse a los datos. Sin embargo, en el conjunto de prueba se observa una caída considerable en su rendimiento, lo que indica una posible tendencia al sobreajuste. Esta diferencia tan marcada entre ambos conjuntos sugiere que, aunque estos modelos son muy potentes, pueden estar capturando ruido en lugar de patrones generales, lo que afecta negativamente su capacidad de generalización.

En conjunto, los resultados muestran que Elastic Net, Lasso, Bagging y Random Forest son los modelos con mejor comportamiento global, mientras que Árboles simples, Árboles Podados y Naive Bayes muestran limitaciones importantes en su capacidad predictiva.

Variable más importante

Según las ilustraciones 1, 2, 14, 24, 27, 33, 42 y 47, observamos que la variable más importante en la clasificación es Sigo_redes, la cual hace referencia a la frecuencia con la que los participantes siguen contenido relacionado con videojuegos en redes sociales. Las opciones de respuesta abarcan un rango desde *todos los días*, *al menos una vez por semana*, *al menos una vez cada quince días*, *con menor frecuencia* o *nunca*.

En los gráficos se aprecia claramente cómo esta variable influye en la predicción, mostrando diferencias significativas entre los distintos niveles de frecuencia. Además, su peso dentro del modelo sugiere que el consumo de contenido en redes sociales está estrechamente relacionado con la variable objetivo. Cabe destacar que las categorías de esta variable están bien representadas, con un porcentaje de alrededor del 20% en cada una, lo que garantiza una distribución equilibrada y evita posibles sesgos en la clasificación.

Asimismo, al analizar su importancia en los diferentes modelos utilizados, se observa que Sigo_redes mantiene una posición destacada en la selección de variables, lo que indica su relevancia en la clasificación. Esta observación se mantiene consistente a lo largo de los distintos enfoques aplicados, reforzando la idea de que esta variable desempeña un papel clave en la predicción.

Discusión

Comparación de los modelos y coherencia con la teoría

Los resultados obtenidos muestran que los modelos Elastic Net y Lasso fueron los más efectivos en términos de AUC, destacándose por su estabilidad y alto rendimiento tanto en entrenamiento como en prueba. Este resultado es consistente con lo esperado teóricamente, ya que ambos modelos utilizan regularización para prevenir el sobreajuste y mejorar la generalización. En particular, Elastic Net, al combinar las penalizaciones L1 (Lasso) y L2 (Ridge), es capaz de manejar mejor las relaciones entre variables altamente correlacionadas, lo que lo convierte en una opción robusta para este tipo de datos.

En contraste, el Modelo Manual mostró un rendimiento competitivo en entrenamiento, pero sufrió una notable caída en prueba. Esto sugiere que el modelo pudo haber sobreajustado los datos de entrenamiento al seleccionar características que no generalizan bien a nuevos datos. Este comportamiento es característico de modelos en los que la selección de variables no está regulada mediante técnicas de validación cruzada o criterios de penalización.

Por otro lado, el Modelo Automático, aunque mostró menor variabilidad entre entrenamiento y prueba, tuvo un desempeño inferior en comparación con Elastic Net y Lasso. Este resultado indica que el modelo logró una estabilidad razonable, pero a costa de una menor capacidad predictiva. Es posible que su estrategia de selección de

características haya sido más conservadora, lo que redujo el riesgo de sobreajuste pero también limitó su capacidad de capturar patrones complejos en los datos.

El modelo Naive Bayes presentó de los peores rendimientos entre todos los modelos evaluados, con un AUC bajo y una alta dispersión en las métricas de prueba. Este resultado es coherente con la teoría, dado que Naive Bayes asume independencia entre las variables predictoras, lo cual raramente se cumple en conjuntos de datos reales. Si las variables están correlacionadas, como es el caso en este estudio, el modelo tiende a cometer errores en la clasificación. Además, al ser un modelo probabilístico basado en distribuciones, su desempeño puede verse afectado por valores atípicos o una representación desigual de los datos.

En cuanto a los modelos basados en árboles, como Árbol de decisión, Árbol Podado, Bagging y Random Forest, se observa un comportamiento dispar. Los modelos Árbol y Árbol Podado mostraron un rendimiento bajo y una alta variabilidad, lo cual es consistente con su naturaleza altamente dependiente de los datos de entrenamiento y su tendencia a sobreajuste cuando no se regulan adecuadamente. En cambio, los modelos de árboles como Bagging y Random Forest ofrecieron un rendimiento excelente en el conjunto de entrenamiento, pero con una caída considerable en el conjunto de prueba. Este patrón es indicativo de sobreajuste, ya que estos modelos, aunque robustos, tienden a capturar incluso el ruido presente en los datos de entrenamiento si no se controla su complejidad (por ejemplo, mediante el número de árboles o la profundidad máxima. Aunque estos modelos tienen un gran potencial, los resultados indican que su aplicación requiere precaución, ya que pueden ofrecer resultados muy optimistas en entrenamiento pero no siempre se traducen en un buen desempeño sobre nuevos datos.

En general, los resultados confirman que los modelos regularizados, como Elastic Net y Lasso, ofrecen un mejor equilibrio entre precisión y generalización, mientras que modelos más simples como Naive Bayes pueden verse limitados por sus supuestos teóricos.

Importancia de la variable Sigo_redes

La importancia de esta variable puede explicarse por el papel central que desempeñan las redes sociales en la interacción y difusión de contenido en la comunidad de jugadores. Es posible que el seguimiento de contenido relacionado con videojuegos no solo refleje interés en el tema, sino que también sirva como un indicador indirecto de otros comportamientos relevantes, como la frecuencia de juego, la relación con la industria o la participación en comunidades en línea.

Además, esta variable mantuvo una alta importancia en todos los modelos evaluados, lo que sugiere que su relevancia es consistente independientemente del enfoque utilizado. Su impacto en la predicción confirma su utilidad como un fuerte predictor en la clasificación y destaca su relación con las características analizadas en este estudio.

Limitaciones del estudio

A pesar de los buenos resultados obtenidos con algunos modelos, existen ciertas limitaciones que deben considerarse. En primer lugar, la distribución de los datos y la posible presencia de sesgos puede haber influido en el rendimiento de los modelos. Aunque Sigo_redes presenta un equilibrio en sus categorías, otras variables podrían

no estar igualmente representadas, lo que podría haber afectado la capacidad de generalización de los modelos.

El tamaño del conjunto de datos es otro factor que considerar. Si bien los modelos de aprendizaje automático pueden adaptarse a diferentes volúmenes de datos, un conjunto demasiado pequeño puede dificultar el aprendizaje de patrones complejos y hacer que los modelos sean más sensibles a pequeñas variaciones. Esto podría explicar la variabilidad observada en algunos modelos, especialmente en aquellos con menos regularización.

Por otro lado, una limitación importante en este estudio es la capacidad computacional disponible. Algunos modelos, en especial los basados en técnicas de ensamble como Random Forest o Elastic Net, requieren un gran número de repeticiones y validaciones para obtener resultados óptimos. Debido a restricciones en el hardware y software utilizado, fue necesario limitar el número de iteraciones y ajustar la complejidad de los modelos. Con un mayor poder de cómputo, habría sido posible explorar una gama más amplia de combinaciones de parámetros y realizar un ajuste más preciso, lo que potencialmente podría haber mejorado el rendimiento de ciertos modelos.

Comparación con estudios previos

Los resultados obtenidos son coherentes con lo que se esperaría en la literatura sobre modelos de clasificación. Se ha demostrado en diversos estudios que Elastic Net y Lasso son especialmente útiles en contextos donde existe una alta correlación entre variables, ya que permiten seleccionar un subconjunto óptimo de predictores sin comprometer la capacidad de generalización del modelo.

Por otro lado, el bajo rendimiento de Naive Bayes también es consistente con la teoría, dado que su supuesto de independencia entre variables es poco realista en la mayoría de los problemas de clasificación. Modelos similares han mostrado limitaciones cuando los datos presentan estructuras complejas o correlaciones significativas entre predictores.

Finalmente, la importancia de variables relacionadas con el consumo de contenido en redes sociales también ha sido resaltada en estudios previos sobre hábitos digitales, lo que refuerza la validez de los resultados obtenidos en este trabajo.

Conclusión

Este estudio demuestra que los modelos regularizados, como Elastic Net y Lasso, ofrecen un rendimiento superior en términos de capacidad predictiva y estabilidad. Ambos modelos no solo lograron altos valores de AUC, sino que también mostraron una excelente capacidad para generalizar, lo que sugiere que son adecuados para la clasificación en este tipo de problemas.

En cuanto a las variables utilizadas en los modelos, Sigo_redes ha sido la más influyente en la predicción. Este hallazgo destaca una clara relación entre el consumo de contenido relacionado con videojuegos en redes sociales y la variable objetivo. Estos resultados sugieren que el comportamiento en redes sociales de los jugadores puede ser un indicador clave para la clasificación, lo que abre nuevas oportunidades para estudios posteriores sobre cómo la actividad en plataformas sociales impacta en el comportamiento de los jugadores.

Sin embargo, hay varias limitaciones que deben ser consideradas. El tamaño de la muestra y la capacidad computacional disponible fueron factores que condicionaron el alcance de este análisis. La falta de recursos más avanzados impidió ejecutar modelos más complejos y realizar un número mayor de iteraciones para un ajuste más preciso. Estas limitaciones podrían haber influido en el rendimiento final de algunos modelos y, por lo tanto, deben ser tomadas en cuenta en futuras investigaciones.

En resumen, los resultados obtenidos sirven como base para futuras investigaciones en este campo, particularmente en el estudio de la relación entre las interacciones en redes sociales y el comportamiento de los jugadores. Además, el análisis realizado proporciona una sólida plataforma para el desarrollo de modelos predictivos más avanzados, con el potencial de mejorar la precisión de las predicciones a medida que aumenten los recursos computacionales y el tamaño de los datos disponibles.

Bibliografía

1. Martos, S. S., Navarro, V., & Planells, A. J. (2018). Características de las comunidades de jugadores de videojuegos. Game & Play: La cultura del juego digital, 37.

2. Pérez, A., Kizys, R., & Manzanedo, L. (2015). Regresión logística binaria. Universitat Oberta de Catalunya, 1-17.

3. Parmar, A., Katariya, R., & Patel, V. (2019). A review on random forest: An ensemble classifier. In International conference on intelligent data communication technologies and internet of things (ICICI) 2018 (pp. 758-763). Springer International Publishing.

4. Sutton, C. D. (2005). Classification and regression trees, bagging, and boosting. Handbook of statistics, 24, 303-329.

5. Mair, P., Groenen, P. J., & de Leeuw, J. (2022). More on multidimensional scaling and unfolding in R: smacof version 2. Journal of Statistical Software, 102, 1-47.

6. Kassambara, A. (2017). Practical guide to cluster analysis in R: Unsupervised machine learning (Vol. 1). Sthda.

7. Crawley, M. J. (2012). The R book. John Wiley & Sons.

8. Gardener, M. (2012). Beginning R: The statistical programming language. John Wiley & Sons.

9. Calviño Martínez, A., & Alonso Revenga, J. M. (2022). Introducción a la ciencia de datos con R : Preparación de los datos y análisis no supervisado. García Maroto Editores. https://www-ingebook-com.bucm.idm.oclc.org/ib/NPcd/IB_BooksVis?cod_primaria=1000187&codigo_libro=12580

10. Waldmann, P., Mészáros, G., Gredler, B., Fuerst, C., & Sölkner, J. (2013). Evaluation of the lasso and the elastic net in genome-wide association studies. Frontiers in genetics, 4, 270.

11. Dinov, I. D. (2018). Data science and predictive analytics. Cham, Switzerland.

Anexos

Tabla variables estudiadas

Variable	Descripción	Categorías de respuesta
Jugar	Indica si la persona juega o no a videojuegos.	"Sí", "No"
Grupo_edad	Grupo de edad de los participantes.	"15-19 años", "20-24 años", "25-29 años"
Sexo	Género binario.	"Hombres", "Mujeres"
Estudios	Nivel de estudios alcanzado.	"Hasta secundarios obligatorios", "Secundarios posobligatorios", "Universitarios"
Veo_directos	Frecuencia con la que la persona ve retransmisiones en directo sobre videojuegos.	"Nunca", "Con menor frecuencia", "Al menos 1 vez cada 15 días", "Al menos 1 vez por semana", "Todos los días"
Veo_videos	Frecuencia con la que la persona consume videos sobre videojuegos (no en directo).	"Nunca", "Con menor frecuencia", "Al menos 1 vez cada 15 días", "Al menos 1 vez por semana", "Todos los días"
Leo_prensa	Frecuencia con la que la persona lee prensa especializada en videojuegos.	"Nunca", "Con menor frecuencia", "Al menos 1 vez cada 15 días", "Al menos 1 vez por semana", "Todos los días"
Sigo_redes	Frecuencia con la que la persona sigue contenido sobre videojuegos en redes sociales.	"Nunca", "Con menor frecuencia", "Al menos 1 vez cada 15 días", "Al menos 1 vez por semana", "Todos los días"
Eventos_presenciales	Frecuencia con la que la persona asiste a eventos presenciales relacionados con videojuegos.	"Nunca", "Con menor frecuencia", "Al menos 1 vez cada 15 días", "Al menos 1 vez por semana", "Todos los días"

Veo_esports	Frecuencia con la que la persona consume contenido sobre deportes electrónicos (eSports).	"Nunca", "Con menor frecuencia", "Al menos 1 vez cada 15 días", "Al menos 1 vez por semana", "Todos los días"
Creo_contenido	Frecuencia con la que la persona crea contenido relacionado con videojuegos (videos, streamings, blogs, etc.).	"Nunca", "Con menor frecuencia", "Al menos 1 vez cada 15 días", "Al menos 1 vez por semana", "Todos los días"
Pasion_videojuegos	Indica si la persona se considera apasionada de los videojuegos.	"Sí", "No"
Equipamiento_habilidad	Indica si la persona considera que posee equipamiento y habilidades avanzadas en videojuegos.	"Sí", "No"
Socializacion_comunidad	Indica si la persona socializa y forma parte de comunidades relacionadas con los videojuegos.	"Sí", "No"
Impacto_personal_salud	Indica si la persona percibe un impacto de los videojuegos en su vida personal y su salud.	"Sí", "No"
Identidad_cultura_gamer	Indica si la persona se identifica con la cultura gamer.	"Sí", "No"