Análisis de Hábitos Estudiantiles y Rendimiento Académico

Trabajo Final de Análisis de Datos

Ángel Velázquez Bolívar

29/05/2025

Índice

1 Introducción

El presente trabajo tiene como finalidad analizar el impacto que varios hábitos de vida tienen sobre el rendimiento académico de los estudiantes. Para ello, se realizará un análisis de datos combinando técnicas descriptivas, predictivas y exploratorias.

1.1 Origen de los datos

Los datos se han obtenido del dataset *Student Habits vs Academic Performance* en formato CSV, publicado en Kaggle. Este dataset contiene información simulada sobre algunos hábitos de 1000 estudiantes y sus calificaciones.

1.2 Objetivos

- Realizar un análisis descriptivo completo de las variables, con el fin de comprender su comportamiento individual y sus relaciones.
- Construir modelos de regresión para identificar los factores más relevantes en la predicción de la calificación final.
- Evaluar la posibilidad de reducir la dimensionalidad del conjunto de datos mediante técnicas como el análisis de componentes principales.
- Aplicar métodos de clasificación no supervisada para segmentar el alumnado según patrones comunes de comportamiento.
- Emplear técnicas de clasificación supervisada, como árboles de decisión, random forest y boosting, para predecir el rendimiento académico.

Este enfoque permitirá no solo construir modelos predictivos precisos, sino también generar interpretaciones útiles sobre qué variables influyen más en el rendimiento académico, y cómo se agrupan los estudiantes según sus hábitos.

2 Análisis descriptivo

2.1 Descripción de las variables

El dataset consta de 16 variables, aunque para este trabajo se han utilizado 15. A continuación se detalla cada una con su tipo, escala y relevancia para el análisis. Los nombres mostrados están modificados, ya que los originales eran en inglés:

Variable	Tipo	Descripción/Escala
Edad	int	Edad del estudiante (rango 17-24)
Sexo	Factor	Sexo del estudiante (Male/Female/Other)
Horas_estudio_diario	num	Horas de estudio diario (rango 0-8)
Horas_redes_sociales	num	Horas de uso de redes sociales diario (rango 0-7)
Horas_netflix	num	Horas de uso de Netflix diario (rango 0-5)
Trabajo	Factor	Indica si el estudiante trabaja o no (Yes/No)
Asistencia	num	Proporción de clases asistidas (0-1 tras transformación)
Horas_sueño	num	Horas diarias de sueño (rango 3-10)
Dieta	Factor	Calidad de la alimentación (Fair/Good/Poor)
Dias_ejercicio_semanal	int	Días de ejercicio a la semana (rango 0-6)
Educacion_padres	Factor	Nivel educativo de los padres (None/High School/Bachelor/Master)
Calidad_internet	Factor	Calidad del internet del estudiante (Poor/Average/Good)
Salud_mental	int	Indica la salud mental del estudiante (escala 1-10)
Actividades_extracurriculares	Factor	Realización de actividades extracurriculares (Yes/No)
Calificacion	num	Nota final promedio (escala 1-10 tras transformación)

2.2 Análisis

2.2.1. Limpieza de los datos

Como en todo proyecto de análisis de datos, el primer paso debe ser la limpieza y el estudio descriptivo de los datos. La limpieza ha consistido en comprobar que no había datos faltantes en el conjunto de datos y en la eliminación de una de las variables. La variable eliminada ha sido el identificador de cada estudiante, ya que es innecesario para nuestros análisis. Después de esto se cambió el nombre de las variables, ya que los nombres originales estaban en inglés y podían resultar algo ambiguos.

Finalmente, se transformaron las variables cualitativas en variables factor para facilitar su manejo en los análisis y se modificó el rango de valores de las variables Calificación y Asistencia. De esta forma, las calificaciones quedaron en una escala del 1 al 10, y la asistencia en un porcentaje del 1 al 100 %.

2.2.2. Resumen estadístico

El estudio descriptivo comienza por un estudio estadístico básico. Para ello se ha utilizado la función `describe` del paquete `Psych`. En la tabla siguiente se pueden observar los resultados más relevantes de dicha función:

Cuadro 1: Estadísticos descriptivos de las variables

Variable	**Media**	**Desv. típ.**	**Mín.**	**Máx.**	**Mediana**	**Asimetría**
Edad	20.50	2.31	17.00	24.00	20.00	0.01
Sexo*	1.56	0.57	1.00	3.00	2.00	0.42
Horas_estudio_diario	3.55	1.47	0.00	8.30	3.50	0.05
Horas_redes_sociales	2.51	1.17	0.00	7.20	2.50	0.12
Horas_Netflix	1.82	1.08	0.00	5.40	1.80	0.24
Trabajo*	1.22	0.41	1.00	2.00	1.00	1.39
Asistencia	0.84	0.09	0.56	1.00	0.84	-0.24
Horas_sueño	6.47	1.23	3.20	10.00	6.50	0.09
Dieta*	1.75	0.75	1.00	3.00	2.00	0.45
Días_ejercicio_semanal	3.04	2.03	0.00	6.00	3.00	-0.03
Educación_padres*	2.00	0.94	1.00	4.00	2.00	0.66
Calidad_internet*	1.77	0.71	1.00	3.00	2.00	0.36
Salud_mental	5.44	2.85	1.00	10.00	5.00	0.04
Actividades_extracurriculares*	1.32	0.47	1.00	2.00	1.00	0.78
Calificación	6.96	1.69	1.84	10.00	7.05	-0.16

Observamos que la edad media de los estudiantes es de 20.5 años y que en promedio dedican 3.55 horas diarias al estudio. La calificación media es de 6.96 puntos.

En cuanto a las variables categóricas, por su codificación como variables factor se interpreta que:

- La media de 1.56 en la variable Sexo indica un ligero predominio de mujeres en la muestra.
- La media de 1.22 en la variable Trabajo indica que la mayoría de estudiantes no trabaja.
- La media de 1.75 en la variable Dieta indica una alimentación algo desequilibrada en promedio.
- La media de 2.00 en la variable Educación_padres indica que la mayoría de los padres tienen niveles de educación intermedios.
- La media de 1.77 en la variable Calidad_internet indica que el internet de los estudiantes es razonablemente bueno.
- La media de 1.32 en la variable Actividades_extracurriculares indica que la mayoria de estudiantes no participa en estas actividades.

De todas formas, todo esto se verá más claro en la visualización de cada variable que se expone a continuación.

2.2.3. Distribución de las variables numéricas

Comenzamos representando el diagrama de caja de cada variable numérica. Los resultados pueden verse en la Figura 1. Estos gráficos nos permiten detectar valores extremos, además de visualizar la distribución de los datos.

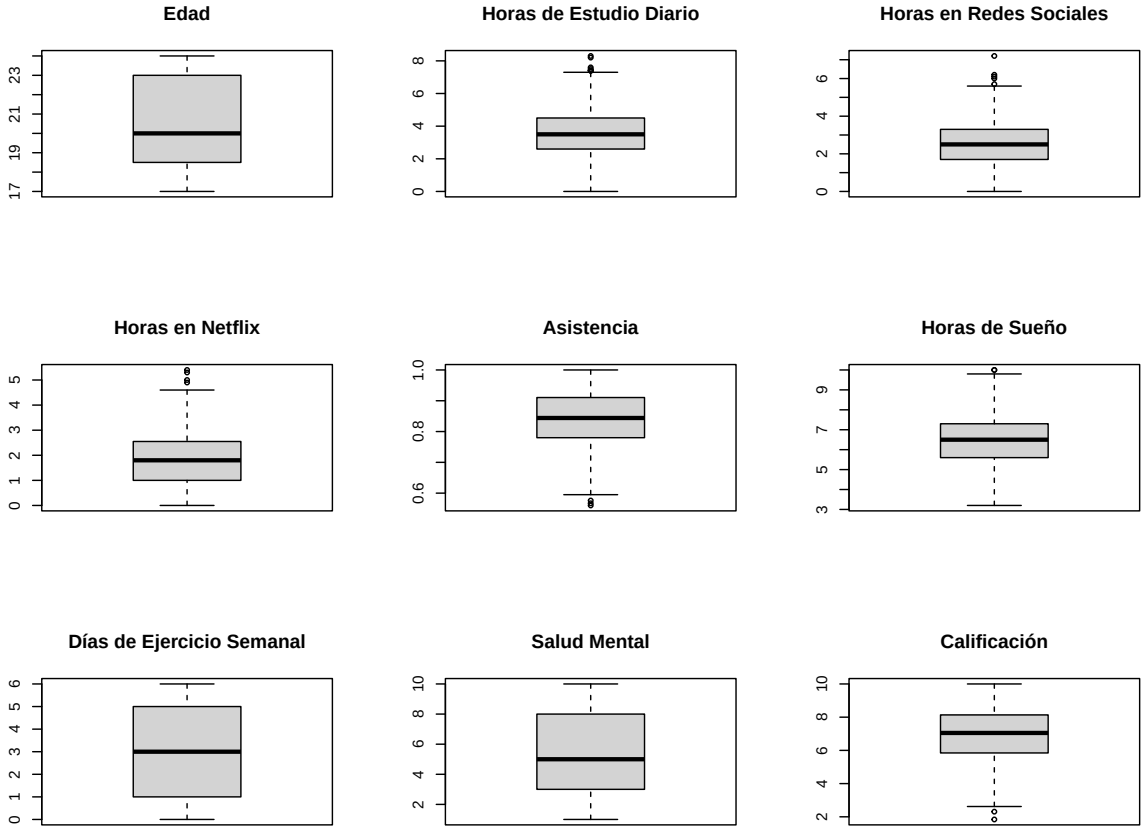

Figura 1: Diagrama de cajas de las variables numéricas

Para una mejor visualización de la distribución de los datos se representaron los histogramas de cada variable numérica. Los resultados pueden verse en la Figura 2.

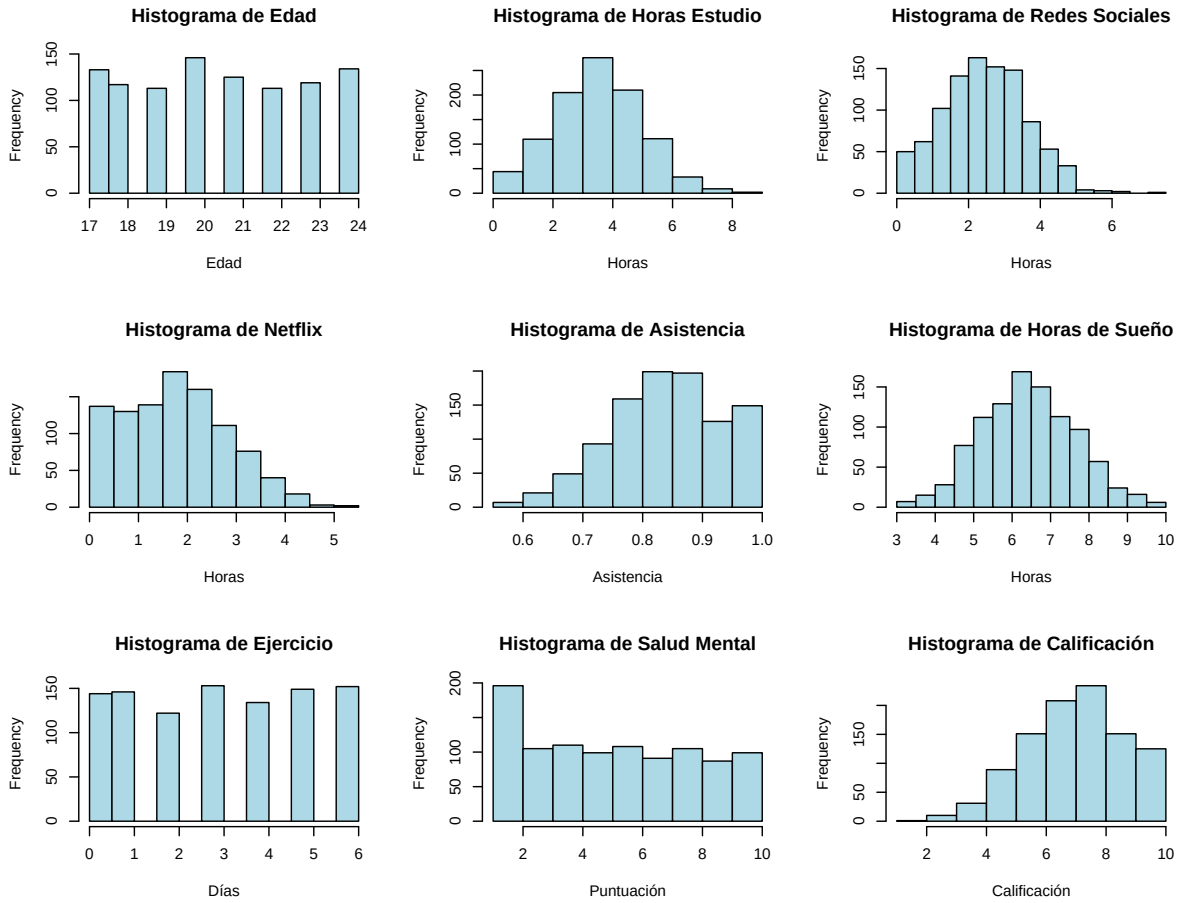

Figura 2: Histogramas de las variables numéricas

Resumimos a continuación las observaciones más relevantes obtenidas de estos gráficos:

- Edad: El histograma muestra una distribución bastante uniforme, de manera que hay aproximadamente los mismo estudiante de cada edad entre los 17 y 24 años. El boxplot no evidencia la presencia de valores atípicos.
- Horas de estudio diario: El histograma muestra una distribución relativamente simétrica y centrada en torno a 3–4 horas diarias. El boxplot revela una dispersión moderada, con algunos casos extremos de estudiantes que superan las 7 horas diarias.
- Horas en redes sociales: El histograma muestra una distribución relativamente simétrica, ligeramente desplazada hacia la izquierda y centrada en torno a 2–3 horas diarias. Aunque escasos, existen algunos valores notablemente altos (hasta más de 6 horas), lo que se refleja en el boxplot mediante la presencia de valores extremos.
- Horas viendo Netflix: La distribución es similar a la de las redes sociales: la mayoría de los estudiantes consume menos de 2 horas al día, pero hay una cola hacia valores superiores. El boxplot confirma esta tendencia con algunos valores extremos.
- Asistencia: Esta variable, que representa la proporción de asistencia a clase, presenta una distribución muy concentrada en valores altos (superiores al 80 %). Sin embargo

existen algunos valores incluso por debajo del 60 %, que aparecen reflejados en el boxplot mediante la presencia de valores extremos.

- Horas de sueño: La distribución es bastante simétrica y centrada en torno a 6.5 horas, con un rango que va desde poco más de 3 horas hasta 10. El boxplot muestra algunos valores altos que podrían considerarse extremos, pero no excesivamente preocupantes.
- Días de ejercicio semanal: El histograma muestra una distribución relativamente uniforme. Se detectan algunos estudiantes que no realizan ejercicio y otros que lo hacen todos los días, aunque sin extremos anómalos según el boxplot.
- Salud mental: Se distribuye de manera bastante uniforme a lo largo de la escala de 1 a 10, aunque con una clara concentración en torno a valores bajos (por debajo de 2). El boxplot no muestra presencia de valores extremos.
- Calificación: El histograma muestra una distribución relativamente simétrica, ligeramente desplazada hacia la derecha y con la mayoría de los estudiantes concentrados entre el 6 y el 8. Existen algunos valores bajos (por debajo del 4) que se identifican como posibles casos atípicos en el boxplot.

Esta visualización gráfica ha permitido identificar distribuciones aproximadamente simétricas en la mayoría de las variables. Se ha evidenciado también la presencia de algunos valores atípicos en variables como horas de estudio, redes sociales, Netflix, asistencia, horas de sueño y calificación, que podrían ser relevantes en futuros análisis más detallados.

2.2.4. Distribución de las variables categóricas

De manera similar a lo realizado sobre las variables numéricas, ahora visualizamos la distribución de los datos en las variables categóricas. Para ello, se han representado los gráficos de barras de estas 6 variables, donde se representan gráficamente las frecuencias relativas de cada valor. Los resultados pueden verse en la Figura 3.

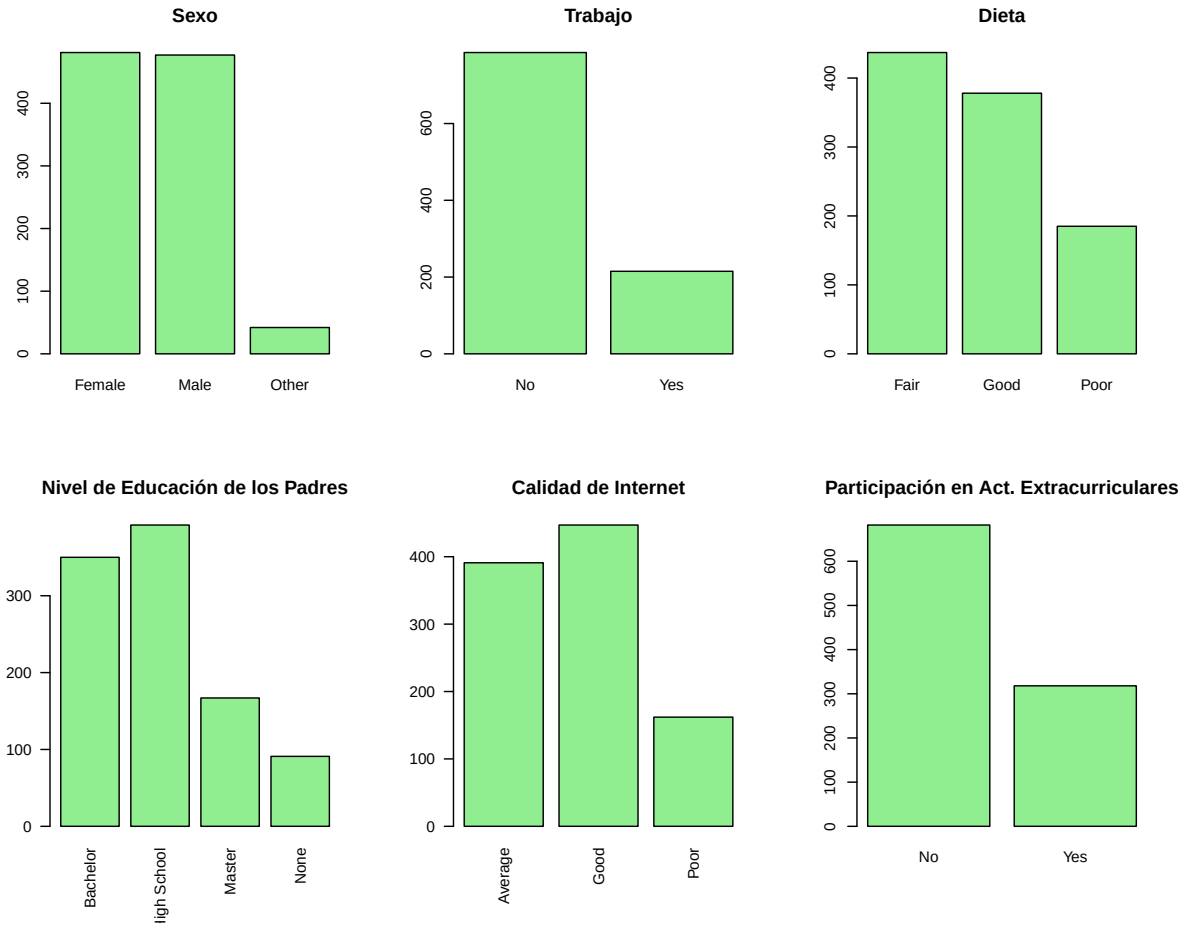

Figura 3: Gráficos de barras de las variables categóricas

Resumimos a continuación las observaciones más relevantes obtenidas de estos gráficos:

- Sexo: El gráfico de barras muestra una distribución equilibrada, aunque con una ligera mayoría de estudiantes de sexo femenino.
- Trabajo: El gráfico de barras muestra que una minoría compagina los estudios con una actividad laboral, lo cual podría influir en la disponibilidad de tiempo para estudiar.
- Dieta: El gráfico de barras muestra que la mayoría de los estudiantes lleva una dieta equilibrada, aunque un porcentaje significativo tiene hábitos alimenticios menos saludables.
- Nivel educativo de los padres: El gráfico de barras muestra una concentración moderada en niveles educativos medios y universitarios.
- Calidad de Internet: El gráfico de barras muestra una mayor frecuencia de estudiantes que disponen de una conexión de calidad media o alta. Dado el importante papel de la conectividad en la educación hoy en día, esta variable podría tener implicaciones en el rendimiento académico.
- Participación en actividades extracurriculares: El gráfico de barras muestra que la

mayoría de estudiantes no participa en actividades extracurriculares, lo cual podría estar relacionado con la carga de estudio o la disponibilidad de tiempo libre.

Este análisis proporciona un contexto esencial para entender el comportamiento de las variables categóricas y su posible relación con el rendimiento académico.

2.2.5. Matriz de correlaciones y posibles relaciones

Para finalizar esta sección de análisis descriptivo, se han tratado de identificar las relaciones lineales existentes entre las variables numéricas del conjunto de datos. Para ello, se ha calculado la matriz de correlaciones de Pearson. En la Figura 4 se muestra una representación gráfica de dicha matriz, en la que se incluyen los coeficientes de correlación para cada par de variables.

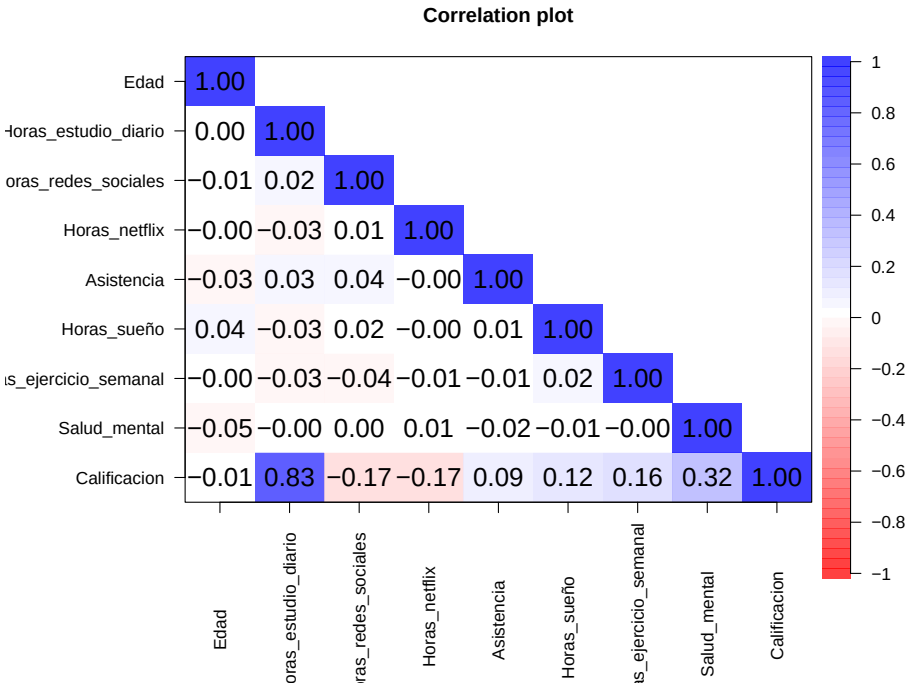

Figura 4: Matriz de correlaciones entre las variables numéricas.

Como se observa, la mayoría de las variables presentan correlaciones lineales muy débiles entre sí. Sin embargo, destacan algunas relaciones relevantes que podrían tener implicaciones sobre el rendimiento académico, medido mediante la variable Calificación:

- Horas de estudio diario: muestra una fuerte correlación positiva con la Calificación ($r = 0.83$), lo que indica que, en general, cuanto más estudia un estudiante, mejores resultados obtiene.
- Salud mental: presenta una correlación positiva moderada con la calificación ($r = 0.32$), lo que sugiere que un mejor estado psicológico puede estar asociado con un mayor rendimiento académico.
- Días de ejercicio semanal y Horas de sueño: muestran correlaciones positivas más suaves con la calificación ($r = 0.16$ y $r = 0.12$, respectivamente), lo que podría

indicar una posible relación entre el bienestar físico y el rendimiento.

- Horas de redes sociales y Horas de Netflix: muestran correlaciones negativas suaves con la calificación (r = -0.17 y r = -0.17), lo que podría indicar que el consumo elevado de ocio digital podría tener un impacto negativo en el rendimiento académico.

El resto de variables presentan correlaciones muy bajas, tanto entre sí como con la variable objetivo. Estos resultados revelan que las predicciones basadas únicamente en modelos lineales podrían no capturar por completo las posibles relaciones subyacentes, especialmente si estas son de naturaleza no lineal.

3 Regresión

Para entender mejor los factores que influyen en el rendimiento académico, vamos a construir modelos de regresión que permitan predecir la calificación final a partir de diversas variables explicativas recogidas en el estudio. Para ello, se aplicarán diferentes técnicas de regresión, comenzando por un modelo lineal múltiple básico que incluya todas las variables disponibles en el conjunto de datos, tanto numéricas como categóricas (estas últimas transformadas automáticamente mediante codificación dummy).

Este enfoque progresivo permitirá comparar modelos, interpretar sus resultados y, finalmente, obtener un modelo predictivo que sea eficaz para explicar las calificaciones finales en función de factores personales y hábitos del alumnado.

3.1 Modelo lineal múltiple inicial

Como primer paso en el análisis, se ha construido un modelo de regresión lineal múltiple utilizando todas las variables explicativas disponibles en el conjunto de datos. Para la construcción y posterior evaluación de los modelos, se han dividido los datos en dos subconjuntos. Un subconjunto de entrenamiento con el 70 % de los datos y un subconjunto de test con el 30 % de los datos, seleccionados aleatoriamente.

La variable dependiente fue la `Calificación`, mientras que las variables independientes incluyeron el resto de las variables recogidas en el estudio.

Se muestra a continuación el resumen del modelo:

```
Call:
lm(formula = Calificacion ~ ., data = datos[train, ])

Residuals:
     Min       1Q   Median       3Q      Max
-2.33604 -0.33351  0.01864  0.33691  1.60564

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)              0.5328561  0.2984536   1.785   0.0746 .
Edad                    -0.0003634  0.0087308  -0.042   0.9668
SexoMale                 0.0654341  0.0418614   1.563   0.1185
SexoOther                0.0331804  0.1027593   0.323   0.7469
Horas_estudio_diario     0.9615698  0.0137596  69.883  < 2e-16 ***
```

```
Horas_redes_sociales              -0.2576843  0.0174961  -14.728  < 2e-16 ***
Horas_netflix                     -0.2157671  0.0189294  -11.399  < 2e-16 ***
TrabajoYes                         0.0293989  0.0493144    0.596   0.5513
Asistencia                         1.5353508  0.2197240    6.988 6.68e-12 ***
Horas_sueño                        0.2118636  0.0171066   12.385  < 2e-16 ***
DietaGood                         -0.0061005  0.0452935   -0.135   0.8929
DietaPoor                         -0.0468863  0.0563883   -0.831   0.4060
Dias_ejercicio_semanal             0.1356236  0.0100623   13.478  < 2e-16 ***
Educacion_padresHigh School        0.0313490  0.0473118    0.663   0.5078
Educacion_padresMaster            -0.0134384  0.0610956   -0.220   0.8260
Educacion_padresNone              -0.0862333  0.0760308   -1.134   0.2571
Calidad_internetGood              -0.1019362  0.0447991   -2.275   0.0232 *
Calidad_internetPoor              -0.0894168  0.0599406   -1.492   0.1362
Salud_mental                       0.1921541  0.0071973   26.698  < 2e-16 ***
Actividades_extracurricularesYes  -0.0649742  0.0431282   -1.507   0.1324
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1
1

Residual standard error: 0.5344 on 680 degrees of freedom
Multiple R-squared:  0.9055,    Adjusted R-squared:  0.9028
F-statistic: 342.8 on 19 and 680 DF,  p-value: < 2.2e-16
```

La salida de la función `summary(modelo1)` ofrece una visión detallada del ajuste del modelo lineal múltiple construido con todas las variables disponibles. Las variables con un asterisco (***) en la columna $Pr(>|t|)$ son altamente significativas ($p < 0{,}001$). Entre ellas destacan: `Horas_estudio_diario`, `Horas_redes_sociales`, `Horas_netflix`, `Asistencia`, `Horas_sueño`, `Dias_ejercicio_semanal` y `Salud_mental`. Estas variables contribuyen de forma robusta al modelo. Otras variables, como el sexo, el tipo de dieta o el nivel educativo de los padres, no resultan significativas, lo que sugiere un menor impacto en la calificación final. Por este motivo, se aplicarán posteriormente técnicas de selección de variables para mejorar la eficiencia del modelo.

El modelo presenta un coeficiente de determinación $R^2 = 0.9055$ y un R^2 ajustado $= 0.9028$, lo cual indica que más del 90 % de la variabilidad en la calificación está explicada por las variables independientes del modelo. Esto indica un muy buen ajuste para un modelo de regresión.

Tras la construcción del modelo se ha procedido a su diagnóstico para validar su aplicabilidad. En la Figura 5 se muestran los gráficos de diagnóstico generados con la función `autoplot(modelo1, which = 1:4)`.

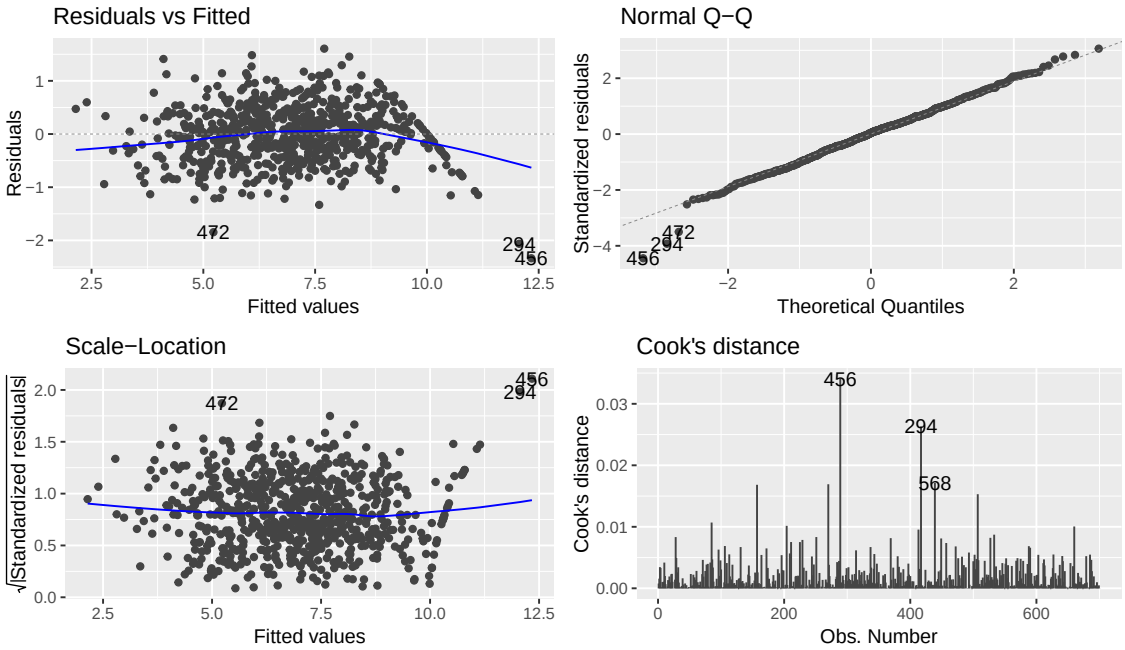

Figura 5: Gráficos de diagnóstico para el modelo inicial.

Lo que más nos interesa de la Figura 5 son los 3 primeros gráficos. Estos gráficos nos permiten explorar visualmente la validez del modelo según el comportamiento de los residuos. La interpretación de estos gráficos es la siguiente:

- **Residuals vs Fitted**: Este gráfico evalúa la relación entre los residuos y los valores ajustados por el modelo. Idealmente, los residuos deben distribuirse aleatoriamente alrededor de la línea horizontal. En este caso, la nube de puntos es bastante homogénea, aunque se aprecia una ligera curvatura. Aun así, no se observan patrones graves que comprometan la validez del modelo. Podemos considerar que se cumple el supuesto de linealidad.
- **(Normal Q-Q)**: Este gráfico permite comprobar si los residuos siguen una distribución normal. En un modelo bien ajustado, los puntos deberían alinearse aproximadamente sobre la línea diagonal. En este caso, la mayoría de los puntos siguen la recta, a pesar de algunas desviaciones leves en los extremos, que podría deberse a la presencia de outliers. Sin embargo, el ajuste global puede considerarse bueno.
- **Scale-Location**: Este gráfico evalúa si la varianza de los residuos es constante (homocedasticidad). La línea azul muestra una tendencia casi plana, y los residuos estandarizados se distribuyen de manera relativamente uniforme a lo largo del rango de valores ajustados. No se identifican patrones en abanico, por lo que podemos considerar el supuesto de homocedasticidad razonablemente cumplido.

En conjunto, los gráficos de diagnóstico apoyan la validez del modelo. Aunque se observan algunas desviaciones menores en la normalidad, esto no compromete gravemente la calidad del modelo.

Para una evaluación aún más exhaustiva del modelo, se procedió a la visualización de los gráficos que se muestran en la Figura 6.

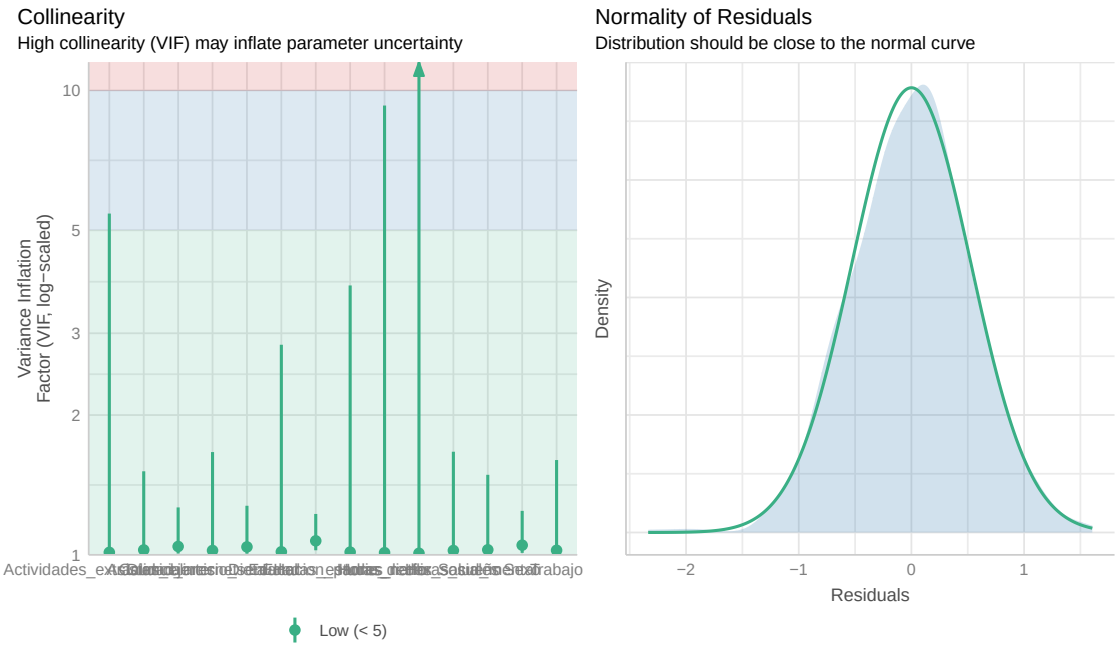

Figura 6: Gráficos de colinealidad y normalidad de los residuos.

La interpretación de estos gráficos es la siguiente:

- **Colinealidad:** El gráfico de los factores de inflación de la varianza (VIF), mostrado en escala logarítmica, evalúa la colinealidad entre las variables predictoras. La presencia de colinealidad entre variables explicativas puede afectar la precisión de las estimaciones del modelo. En este caso, aunque algunas barras se extienden por encima de 5 o 10, los valores VIF reales están todos por debajo del umbral de preocupación (< 5), por lo que no hay evidencia de colinealidad. Debido a la posible confusión que pueden causar las barras de este gráfico, se ha consultado también la salida numérica de la función`vif()`, que nos da el VIF. Todos los valores son cercanos a 1, como se muestra a continuación:

```
> vif(modelo1)
                                GVIF Df GVIF^(1/(2*Df))
Edad                        1.015483  1        1.007712
Sexo                        1.049226  2        1.012086
Horas_estudio_diario        1.013885  1        1.006919
Horas_redes_sociales        1.007950  1        1.003967
Horas_netflix               1.011077  1        1.005523
Trabajo                     1.022991  1        1.011430
Asistencia                  1.024858  1        1.012353
Horas_sueño                 1.021904  1        1.010893
Dieta                       1.040284  2        1.009922
Dias_ejercicio_semanal      1.021945  1        1.010913
Educacion_padres            1.072093  3        1.011670
Calidad_internet            1.042472  2        1.010453
Salud_mental                1.025572  1        1.012705
Actividades_extracurriculares 1.012524  1        1.006242
```

Estos valores indican que no existe colinealidad problemática entre los predictores del modelo.

- **Normalidad de los residuos:** Este gráfico compara la distribución empírica de los residuos con una curva normal teórica. En este caso, la distribución observada es simétrica y se ajusta bastante bien a la curva normal, aunque con ligeras desviaciones en los extremos, lo cual es coherente con los resultados obtenidos en el QQ-plot.

En resumen, el modelo inicial proporciona una base sólida para comprender los factores que influyen en el rendimiento académico y podemos considerarlo adecuado como punto de partida para el análisis. Sin embargo, los indicadores de colinealidad y la posible redundancia de variables motivan la necesidad de simplificar y refinar el modelo, lo que abordaremos en los siguientes apartados.

3.2 Evaluación predictiva

Una vez construido y diagnosticado el modelo de regresión lineal múltiple, se procedió a evaluar su capacidad predictiva sobre el subconjunto de prueba con los datos no utilizados en el ajuste. Se generaron las predicciones mediante la función `predict()` y se compararon con las calificaciones reales. La Figura 7 muestra el gráfico de dispersión entre las predicciones y los valores reales.

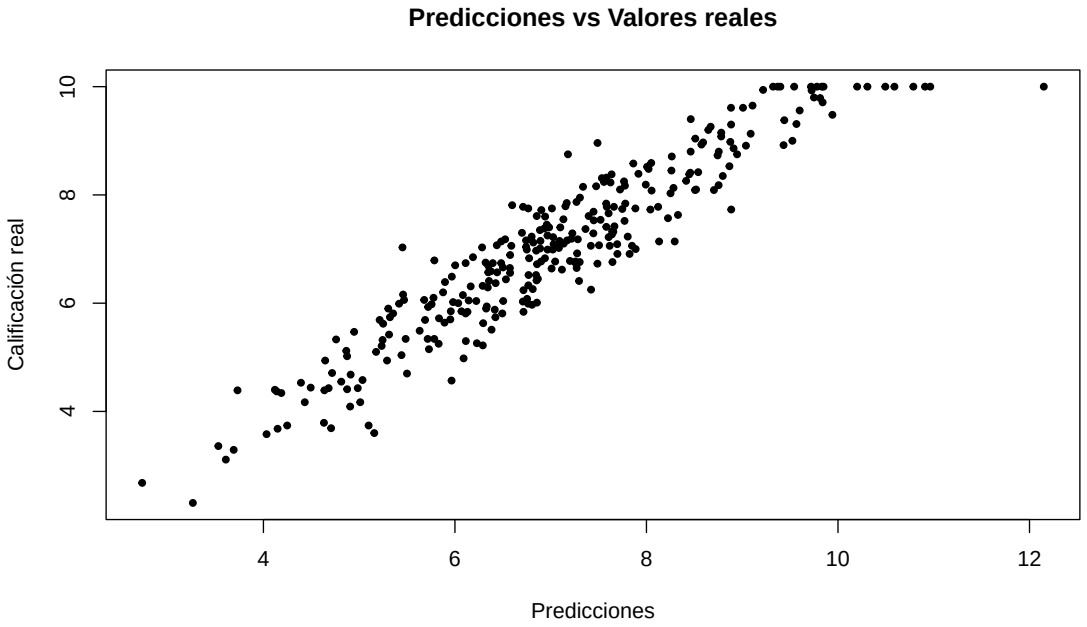

Figura 7: Gráfico de dispersión para el modelo inicial.

En general, se observa una fuerte alineación de los puntos en torno a la diagonal, lo cual indica una alta capacidad predictiva del modelo. No se detectan patrones de error sistemático. No obstante, se identifican unos pocos casos en los que las predicciones superan el valor de 10, alcanzando incluso valores entre 11 y 12. Dado que la variable `Calificación` fue reescalada para estar en el intervalo [0, 10], estas predicciones resultan teóricamente imposibles.

Este comportamiento se explica por el hecho de que un modelo de regresión lineal múltiple

no impone restricciones sobre el rango de salida. Las predicciones pueden, por tanto, exceder el intervalo lógico de la variable dependiente. En este caso, se trata de una pequeña cantidad de observaciones puntuales y no afecta sustancialmente al rendimiento general del modelo.

Para evaluar el ajuste del modelo, se calcularon las siguientes métricas:

- Correlación entre predicciones y valores reales: Se obtuvo una correlación de 0.941, lo cual indica una asociación muy fuerte entre los valores estimados por el modelo y las calificaciones reales observadas. Este resultado refuerza la fiabilidad del modelo en contextos predictivos.
- Resumen de rendimiento: Se utilizó la función `model_performance()`, que nos da varias métricas de interés sobre el modelo:
  - Coeficiente de determinación: Se obtuvo un R^2 de 0.905 y un R^2 ajustado de 0.903. Estos valores ya se comentaron en la sección 3.1.
  - Error cuadrático medio (RMSE): El valor obtenido fue de 0.527, lo cual indica que, en promedio, la desviación entre las calificaciones predichas y las reales es de medio punto. Es un error pequeño, que refuerza la precisión del modelo.
  - Criterios de información (AIC, AICc, BIC): Se obtuvieron los siguientes valores: AIC = 1130.90, AICc = 1132.27, y BIC = 1226.47. Estas métricas penalizan la complejidad del modelo y serán especialmente útiles en las siguientes secciones al comparar versiones simplificadas. Un menor valor indica mejor equilibrio entre ajuste y complejidad.

En conjunto, estas métricas muestran que el modelo se ajusta adecuadamente a los datos, y que además ofrece un buen rendimiento predictivo en datos nuevos. A pesar de algunas predicciones fuera del rango lógico, el modelo demuestra ser eficaz.

3.3 Mejora del modelo: selección de variables

Con el objetivo de simplificar el modelo inicial y mejorar su interpretabilidad sin sacrificar precisión, se aplicó un proceso de selección automática de variables mediante el algoritmo stepwise, utilizando el criterio AIC como función objetivo. Este procedimiento fue implementado con la función `stepAIC()` del paquete `MASS`.

El modelo resultante contó con 10 variables predictivas, eliminando aquellas que no aportaban información significativa al ajuste del modelo. Así, el modelo quedó con las variables: Horas_estudio_diario, Horas_redes_sociales, Horas_netflix, Asistencia, Horas_sueño, Dias_ejercicio_semanal, Calidad_internetGood, Calidad_internetPoor, Salud_mental y Actividades_extracurricularesYes.

Como vemos, se han incluido algunas variables categóricas transformadas en variables dummies automáticamente durante el ajuste.

Sobre este nuevo modelo se realizó el mismo proceso de evaluación que se aplicó al modelo inicial. Las principales métricas obtenidas fueron:

- R^2: 0.905

- RMSE: 0.529
- Correlación en el conjunto de prueba: 0.943
- AIC: 1119.74
- BIC: 1174.36

Podemos comparar estos valores con los del modelo inicial utilizando la función `compare_performance()`:

```
> compare_performance(modelo1, modelo.red, verbose = FALSE)
# Comparison of Model Performance Indices

Name        | Model |  AIC (weights) | AICc (weights) |  BIC (weights)
-------------------------------------------------------------------------
modelo1     |    lm | 1130.9 (0.004) | 1132.3 (0.002) | 1226.5 (<.001)
modelo.red  |    lm | 1119.7 (0.996) | 1120.2 (0.998) | 1174.4 (>.999)

Name        |    R2 | R2 (adj.) |  RMSE | Sigma
-----------------------------------------------------
modelo1     | 0.905 |     0.903 | 0.527 | 0.534
modelo.red  | 0.905 |     0.903 | 0.529 | 0.533
```

Además, se volvió a evaluar el cumplimiento de los supuestos de colinealidad y normalidad de los residuos, obteniendo valores muy similares al modelo inicial.

Como vemos, el modelo reducido mantiene el mismo nivel de precisión que el modelo completo ($R^2 = 0{,}905$), mientras que el AIC y el BIC se han reducido ligeramente. Podría parecer que el nuevo modelo obtenido no supone apenas mejora respecto al inicial. Sin embargo, se ha conseguido obtener la misma precisión con una complejidad menor, lo que favorece la interpretabilidad del modelo, disminuye el riesgo de sobreajuste y mejora la eficiencia computacional. Podemos considerar así que este modelo supone una mejora respecto al modelo original.

3.4 Selección de variables por subconjuntos

Como método alternativo a la selección automática tipo stepwise, se aplicó un enfoque de selección por subconjuntos mediante la función `regsubsets()` del paquete `leaps`. Este método explora todas las combinaciones posibles de predictores para encontrar el subconjunto de variables que optimiza criterios de calidad como el R^2 ajustado o el BIC.

Se ajustaó un modelo con hasta 8 predictores posibles, y se calculó el R^2 ajustado y el BIC (Bayesian Information Criterion) para cada tamaño de subconjunto. Podemos ver las gráficas de la evolución de estas métricas en la Figura 8.

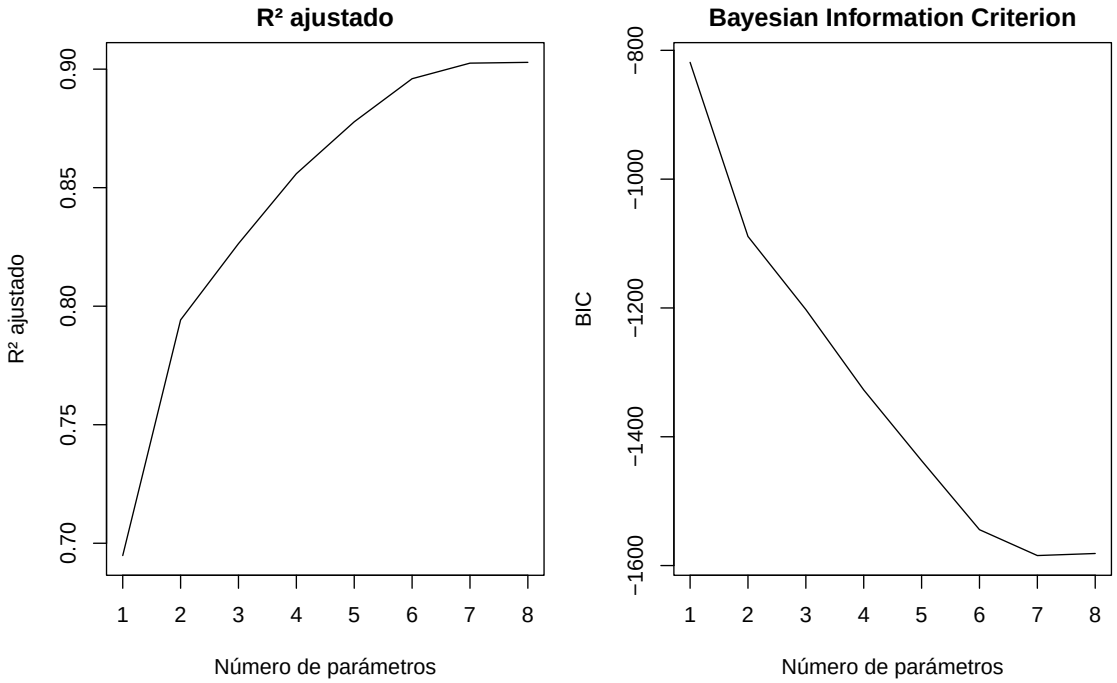

Figura 8: Gráficos de diagnóstico para el modelo obtenido mediante selección por subconjuntos.

Si utilizamos el R^2 como criterio, el mejor modelo se obtiene cuando tenemos 8 variables predictoras, donde:

```
> # Mejor modelo según R  ajustado
> mejor.r <- which.max(res$adjr2)
> res$adjr2[mejor.r]
[1] 0.9028633
```

Es aquí donde se logra el mejor equilibrio entre ajuste y complejidad. El modelo en cuestión utiliza las siguientes variables predictoras:

```
> coef(regfit, mejor.r)
                (Intercept)    Horas_estudio_diario    Horas_redes_sociales
                 0.53089684              0.96272875             -0.25563772
             Horas_netflix               Asistencia             Horas_sueño
                -0.21416057              1.49512397              0.20947719
Dias_ejercicio_semanal    Calidad_internetGood             Salud_mental
                 0.13743372             -0.07442582              0.19346014
```

A continuación, se construyó este modelo para proceder a compararlo con los modelos que habíamos construido.

De nuevo, se realizó sobre el modelo el proceso de evaluación completo que se aplicó a los otros modelos. Se volvió a evaluar el cumplimiento de los supuestos de colinealidad y normalidad de los residuos, obteniendo valores muy similares a los modelos anteriores.

A continuación, se muestran las métricas obtenidas con la función `model_performance()` para este modelo:

```
> model_performance(modelo.sub)
# Indices of model performance
```

AIC	AICc	BIC	R2	R2 (adj.)	RMSE	Sigma
1120.059	1120.443	1170.121	0.904	0.903	0.530	0.534

De nuevo, se puede observar que las métricas son prácticamente idénticas a los modelos anteriores. Finalmente, se utilizó la función `compare_performance()` con el objetivo de comparar los 3 modelos y seleccionar el definitivo:

```
> compare_performance(modelo1, modelo.red, modelo.sub, rank = TRUE)
# Comparison of Model Performance Indices
```

Name	Model	R2	R2 (adj.)	RMSE	Sigma	AIC weights
modelo.red	lm	0.905	0.903	0.529	0.533	0.538
modelo.sub	lm	0.904	0.903	0.530	0.534	0.460
modelo1	lm	0.905	0.903	0.527	0.534	0.002

Name	AICc weights	BIC weights	Performance-Score
modelo.red	0.530	0.107	66.11 %
modelo.sub	0.469	0.893	51.98 %
modelo1	0.001	5.17e-13	28.57 %

Los resultados de la función `compare_performance()` confirman que tanto el modelo reducido por selección automática (`modelo.red`) como el modelo por subconjuntos (`modelo.sub`) ofrecen un rendimiento prácticamente idéntico al modelo completo (`modelo1`), con un R^2 ajustado de 0.903 y valores de RMSE y sigma muy similares.

Sin embargo, el modelo completo es penalizado en los criterios de información (AIC, AICc, BIC) debido a su alta complejidad (19 predictores), lo cual se refleja en su bajo *performance-score* (28.57 %). El modelo reducido y el modelo por subconjuntos mejoran significativamente este valor, al presentar un desempeño similar con menor complejidad. Finalmente, se selecciona como la opción más equilibrada el modelo reducido, con un *performance-score* del 66.11 %. Este modelo con 10 predictores destaca como la opción más equilibrada.

El modelo por subconjuntos, a pesar de utilizar dos predictores menos, no resulta ser el mejor. Esta ligera ventaja en la complejidad no compensa la pérdida de información (aunque pequeña) que refleja el AIC.

3.5 Conclusión del análisis de regresión

En este capítulo se ha desarrollado un análisis de regresión lineal múltiple para predecir la calificación de los estudiantes en función de diferentes variables personales, académicas y de estilo de vida. Se construyó un modelo completo con todas las variables, que mostró un excelente ajuste (R^2 ajustado 0.903), seguido de dos modelos alternativos más simples obtenidos mediante técnicas de selección automática y por subconjuntos.

El diagnóstico del modelo completo confirmó el cumplimiento razonable de los supuestos clásicos de la regresión (linealidad, homocedasticidad, normalidad de residuos y baja colinealidad). Posteriormente, los modelos simplificados mantuvieron prácticamente el mismo

nivel de rendimiento, demostrando que muchas de las variables originales no aportaban valor predictivo relevante.

Tras comparar los modelos mediante múltiples métricas de rendimiento y penalización por complejidad, se concluye que el modelo reducido obtenido por `stepAIC()` es el más adecuado, al lograr una excelente capacidad predictiva con un conjunto limitado de variables significativas. Este modelo permite una interpretación más clara y una mayor eficiencia sin pérdida de calidad en las predicciones.

4 Reducción de dimensionalidad

Antes de abordar los métodos de clasificación, se evaluó la posibilidad de aplicar técnicas de reducción de dimensionalidad, concretamente el análisis de componentes principales (ACP) y el análisis factorial (AF), con el fin de simplificar el conjunto de variables explicativas y facilitar la interpretación de los patrones subyacentes.

Estas técnicas permiten identificar estructuras latentes, condensar información redundante y reducir la complejidad del modelo. No obstante, para que su aplicación sea efectiva, deben cumplirse ciertos requisitos en el conjunto de datos.

Se comenzó analizando la matriz de correlación entre las variables numéricas del conjunto de datos. Tal como se mostró en la Figura 4, la mayoría de las correlaciones son muy bajas, en muchos casos inferiores a 0.05, lo que indica una escasa relación lineal entre las variables. Esta falta de correlación generalizada dificulta que las componentes principales puedan condensar la información de forma eficaz.

En el caso del ACP, el gráfico de autovalores de las componentes obtenido confirmó esta situación. Como se muestra en la Figura 9, la varianza explicada por cada componente fue prácticamente homogénea, sin que ninguna componente principal concentrase una proporción significativa de la varianza total. Esto sugiere que las variables contienen información muy independiente entre sí.

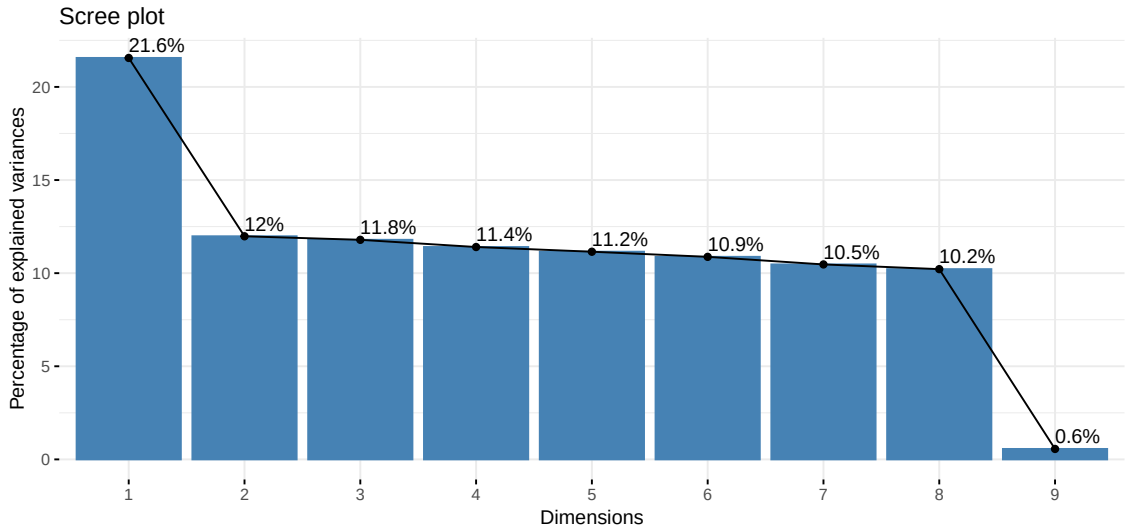

Figura 9: Porcentaje de varianza explicada por componente.

En cuanto al análisis factorial, se evaluaron dos condiciones esenciales para su aplicabilidad, como vemos en el siguiente código:

```
> #Determinante de la matriz de correlación
> det(cor_matrix)
[1] 0.0976899
>
> #Medida de kaiser-M-O
> print(KMO(cor_matrix),digits=4)
Kaiser-Meyer-Olkin factor adequacy
Call: KMO(r = cor_matrix)
Overall MSA =  0.1729
```

- El determinante de la matriz de correlación fue de 0.0976899, lo que indica que las variables están poco relacionadas entre sí (normalmente se considera que las variables están lo suficiente relacionadas para aplicar análisis factorial si este valor está por debajo de 0.00001).
- La medida de Kaiser-Meyer-Olkin arrojó un valor de 0.1729, lo que se interpreta como "muy bajo"según los estándares habituales (valor mínimo recomendable: 0.5).

Ambos indicadores sugieren que el análisis factorial no es apropiado para este conjunto de datos, ya que no se identifican factores latentes comunes que justifiquen su uso.

Tras esta evaluación, se decidió no aplicar ni ACP ni análisis factorial, dado que no existe estructura de correlación suficiente que permita reducir dimensionalidad de forma significativa y no se cumplen las condiciones estadísticas básicas para aplicar análisis factorial.

5 Clasificación

5.1 Clasificación no supervisada

El objetivo de este apartado es analizar si, en ausencia de clases predeterminadas (dadas por la calificación), es posible detectar perfiles de estudiantes con características similares mediante técnicas de clasificación no supervisada. Con dicho fin, se utilizaron métodos jerárquicos y no jerárquicos, partiendo de un conjunto de variables numéricas relacionadas con los hábitos y salud mental de los alumnos.

Inicialmente se seleccionaron las siguientes variables: `Horas_estudio_diario`, `Horas_redes_sociales`, `Horas_netflix`, `Horas_sueño`, `Dias_ejercicio_semanal`, `Asistencia` y `Salud_mental`. Todas ellas tipificadas mediante la función `scale()` para que tuvieran media 0 y desviación típica 1, eliminando así la influencia de diferentes escalas.

A continuación, se construyó una matriz de distancias euclídeas y se aplicó el primer método: el algoritmo de clasificación jerárquica con el método de Ward. En la Figura 10 se observa el dendrograma generado, cuya estructura sugiere la existencia de 4 grupos diferenciados, que se representan en distintos colores.

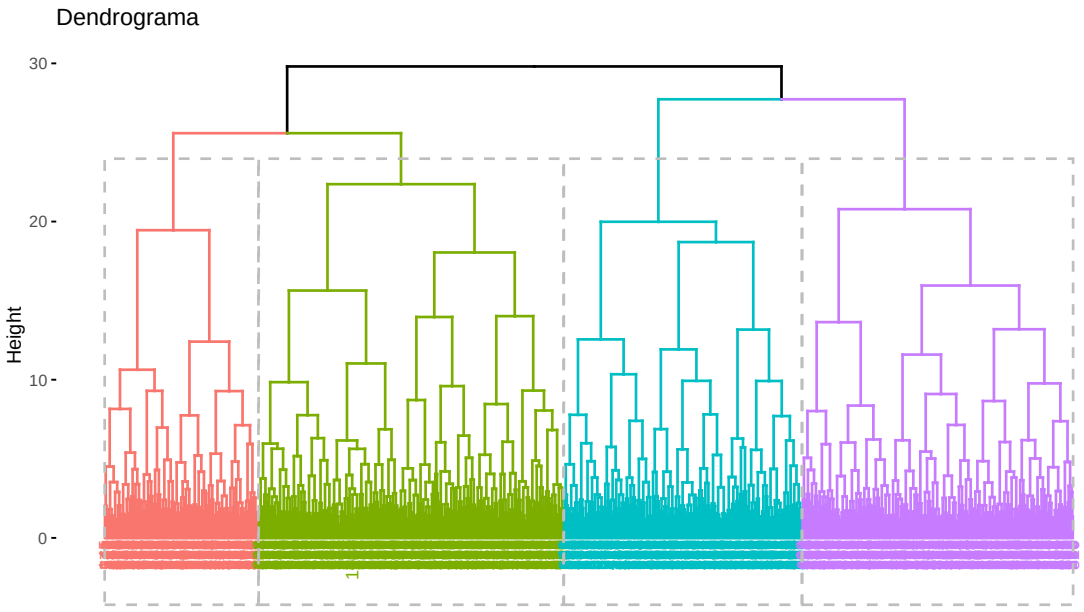

Figura 10: Dendograma de la clasificación con el método de Ward.

A pesar de estas divisiones claras, al proyectar los grupos en el plano de las dos primeras componentes principales, como se observa en la Figura 11, hay un solapamiento considerable entre las observaciones, lo que anticipaba una baja estructura de agrupamiento en los datos.

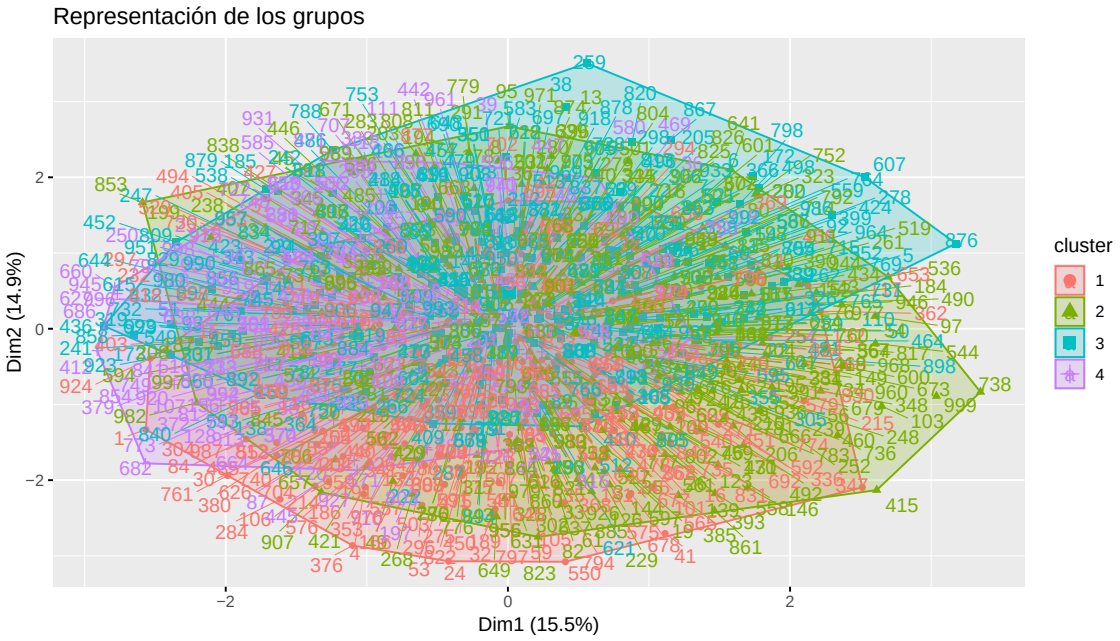

Figura 11: Visualización de los grupos generados con el método de Ward.

Sin embargo, no se puede considerar este gráfico como un criterio fiable de la calidad del agrupamiento, ya que este se basa en el análisis de componentes principales, técnica que, como se expuso en la sección anterior, no es adecuada para este conjunto de datos, dado que las variables no presentan una estructura latente clara.

Por ello, se aplicó el coeficiente de silueta promedio, una métrica utilizada para evaluar la calidad del agrupamiento que mide la similitud de un objeto con los demás objetos de su grupo en comparación con otros grupos. Este coeficiente toma valores en $[-1, 1]$, donde los valores negativos indican un mal agrupamiento, valores cercanos a 0 indican una estructura muy débil, y valores cercanos a 1 indica un agrupamiento bien definido.

En este caso, se obtuvo un coeficiente de silueta promedio de 0.05574751, por lo que, como sospechábamos, el agrupamiento es muy débil.

Con el objetivo de mejorar el agrupamiento, se repitió el proceso utilizando el algoritmo k-medias. Para determinar el número óptimo de grupos, se utilizó la función `fviz_nbclust()`, que como se muestra en la Figura 12, sugirió valores entre 3 y 5. Se seleccionó $k = 4$ para facilitar la comparación con el método anterior.

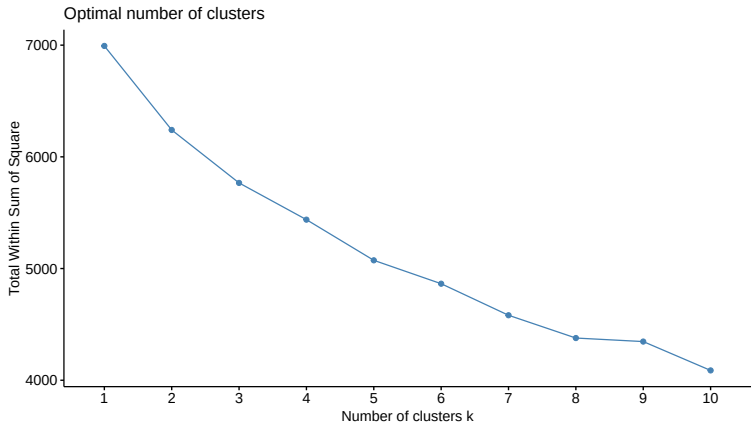

Figura 12: Número de grupos óptimo.

Tras realizar el agrupamiento con la función `hclust()`, se proyectaron los grupos de la misma manera que con el método de Ward, obteniendo resultados muy similares. Sin embargo, el coeficiente de silueta promedio mejoró ligeramente, con un valor de 0.1054213.

Optimización del subconjunto de variables

Dado el escaso rendimiento obtenido con las siete variables originales, se probó a realizar el análisis únicamente con las cuatro variables más directamente asociadas a las rutinas diarias de los estudiantes: `Horas_estudio_diario`, `Horas_redes_sociales`, `Horas_netflix` y `Horas_sueño`.

Esta reducción del espacio de características permitió obtener cierta mejora en la calidad del agrupamiento, con un coeficiente de silueta promedio para el método de Ward de 0.1232509. Sin embargo, los grupos seguían presentando mucho solapamiento.

Finalmente, los mejores resultados se obtuvieron para este subconjunto con el algoritmo k-medias. En este caso se obtuvo un coeficiente de silueta promedio de 0.1232509. En la Figura 13, se puede ver la proyección de los grupos generados en el plano de las dos primeras componentes principales. Se aprecian dos grupos claramente diferenciados, mientras que los otros dos presentan un alto solapamiento, dificultando su separación visual. Este

resultado es coherente con lo observado previamente en el análisis de componentes principales, donde ya se puso de manifiesto la ausencia de una estructura latente clara entre las variables.

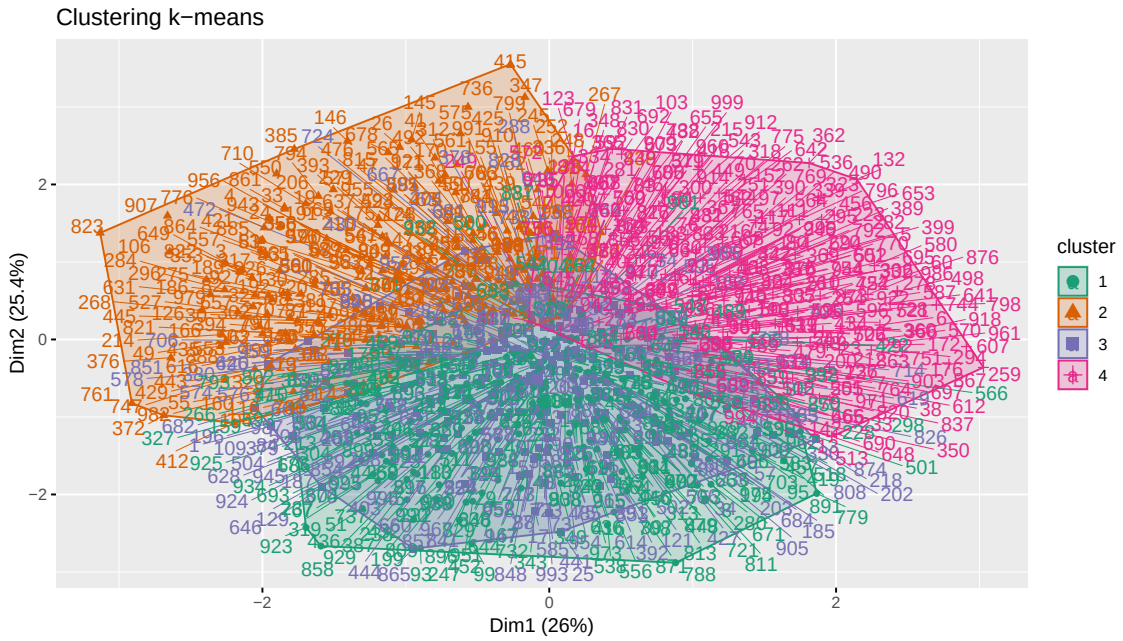

Figura 13: Visualización de los grupos generados con el algoritmo k-medias para el subconjunto reducido.

Para entender el perfil medio de cada grupo generado con el algoritmo k-medias, se calcularon las medias tipificadas de cada variable por grupo, como se muestra en el siguiente código:

```
> datos_clasif_df <- as.data.frame(datos_clasif)
> datos_clasif_df$cluster <- as.factor(kmedias$cluster)
> aggregate(. ~ cluster, data = datos_clasif_df, FUN = mean)
  cluster Horas_estudio_diario Horas_redes_sociales Horas_netflix
1       1           -0.1288032           -0.4968923     0.6422092
2       2           -0.3505250            0.6053427     0.7553026
3       3           -0.4524660           -0.6693433    -0.7867429
4       4            0.8514324            0.5865257    -0.5891339
  Horas_sueño
1  -0.8284278
2   0.8667348
3   0.4023115
4  -0.2887615
```

Como los datos han sido estandarizados, las medias están en unidades de desviación típica. Por lo tanto, un valor positivo implica que ese grupo tiene, en promedio, valores mayores que la media global de esa variable; y un valor negativo implica que ese grupo tiene, en promedio, valores menores que la media global. De esta forma, se interpreta que:

- Grupo 1: Presenta valores bajos en horas de sueño y en uso de redes sociales, pero un valor elevado en horas de Netflix. Su dedicación al estudio es ligeramente inferior a la media. Este grupo podría representar a estudiantes que ven series hasta altas horas de la noche.

- Grupo 2: Se caracteriza por tener valores muy altos en horas de redes sociales, Netflix y sueño, mientras que el tiempo de estudio es claramente inferior a la media. Este grupo podría representar a estudiantes con un estilo de vida poco enfocado en el estudio y que dedican mucho tiempo a las pantallas.
- Grupo 3: Muestra los valores más bajos en casi todas las variables, especialmente en uso de redes sociales y Netflix, así como también por debajo de la media en estudio y sueño. Podría tratarse de un grupo con baja actividad general.
- Grupo 4: Se diferencia por dedicar mucho más tiempo al estudio que el resto, con valores elevados también en uso de redes sociales, pero con menos horas de sueño y bajo consumo de Netflix. Este grupo parece representar a estudiantes más enfocados académicamente, aunque con menor descanso.

Sin embargo, en general, las diferencias entre grupos no eran suficientemente marcadas, y los perfiles identificados no permitieron una interpretación clara y consistente, en línea con los bajos valores del coeficiente de silueta obtenidos.

Finalmente se calculó el coeficiente de silueta promedio para este último agrupamiento, obteniéndose un valor de 0.1710149, el más alto de todos los intentos realizados. Aunque sigue siendo un valor bajo (lo que indica una estructura de agrupamiento débil), sí representa una mejora respecto a los modelos anteriores.

Conclusión

La clasificación no supervisada no permitió detectar grupos bien separados de estudiantes según sus hábitos. A pesar de varios intentos y de la optimización del conjunto de variables utilizadas, los resultados obtenidos muestran que los datos no presentan una estructura de agrupamiento fuerte. Así lo confirma el bajo valor del coeficiente de silueta promedio en todos los modelos (siempre inferior a 0.18), y el solapamiento visual entre grupos observado en las representaciones gráficas.

En consecuencia, se concluye que no existen perfiles claramente diferenciados de estudiantes en función de las variables analizadas. Las técnicas de clasificación no supervisada no logran identificar agrupaciones naturales robustas en este conjunto de datos, posiblemente debido a una fuerte variabilidad individual y a la ausencia de correlaciones estructurales claras entre los hábitos considerados.

A pesar de ello, se ha considerado pertinente incluir este apartado para documentar los intentos de segmentación y valorar de forma crítica su aplicabilidad al conjunto de datos utilizado.

5.2 Clasificación supervisada

Pasamos ahora a analizar la capacidad predictiva de algunos algoritmos de clasificación supervisada con el fin de predecir el rendimiento académico de los estudiantes, en base a un conjunto de variables explicativas.

Para ello, comenzamos transformando la variable continua `Calificacion` en una variable categórica `Rendimiento`, con tres niveles: *Bajo* (0–5), *Medio* (5–8) y *Alto* (8–10), lo que

permite reformular el problema como una tarea de clasificación. Aplicaremos árboles de decisión para después profundizar en técnicas como random forest y boosting.

Cada técnica se evaluará mediante la precisión obtenida sobre un conjunto de prueba y a través de la matriz de confusión.

5.2.1. Árboles de decisión

Comenzamos contruyendo un árbol de decisión, que permite predecir la clase de un individuo a partir de decisiones secuenciales sobre sus características.

Para ello, se entrenó un modelo con la función `rpart()`, utilizando como variable objetivo `Rendimiento` y como predictores el resto de las variables disponibles en el conjunto de entrenamiento (70 % de los datos). Inicialmente se ajustó un árbol sin restricciones, `cp = 0`, para posteriormente analizar su comportamiento y optimizar este parámetro de complejidad. Con este fin, se utilizó el gráfico que se muestra en la Figura 14, que muestra la evolución del error del modelo en función de dicho parámetro.

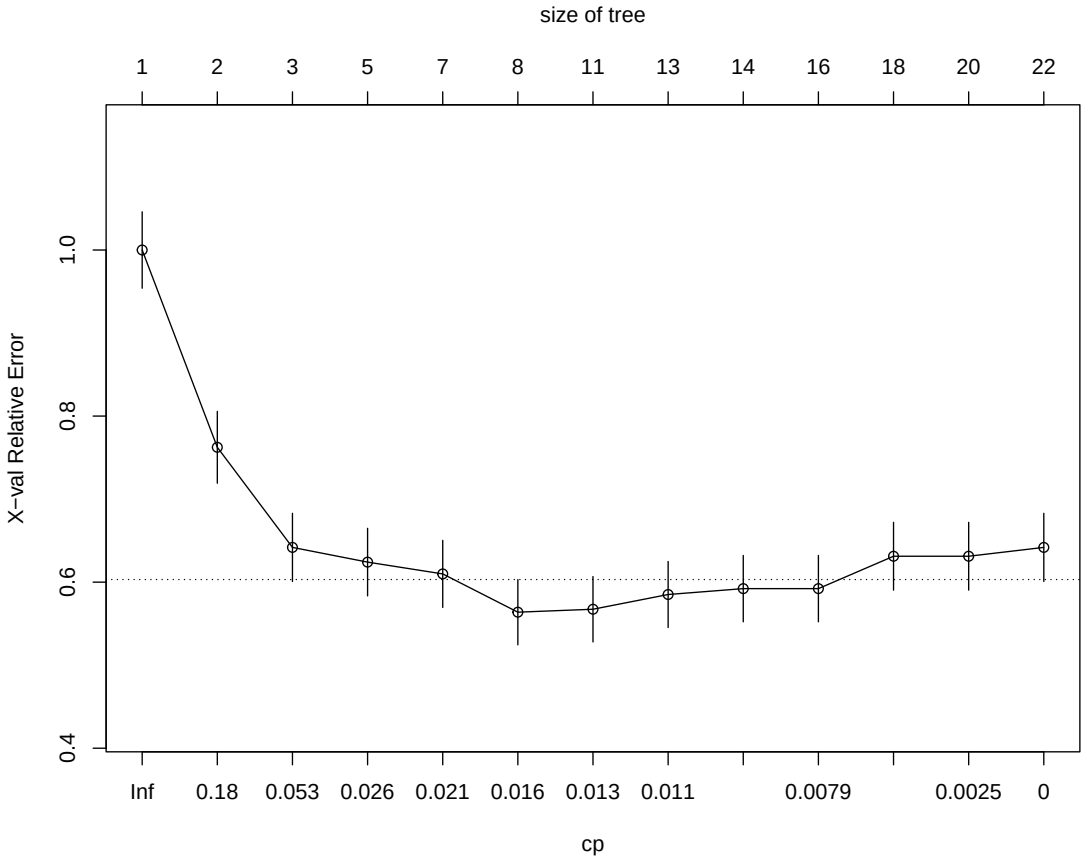

Figura 14: Evolución del error en función del parámetro de complejidad.

A partir de este gráfico se seleccionó un valor óptimo de `cp = 0.016`, para el cual el error de validación cruzada es mínimo. Aplicando este parámetro, se podó el árbol inicial, evitando el sobreajuste y obteniendo como resultado el árbol que se muestra en la Figura 15.

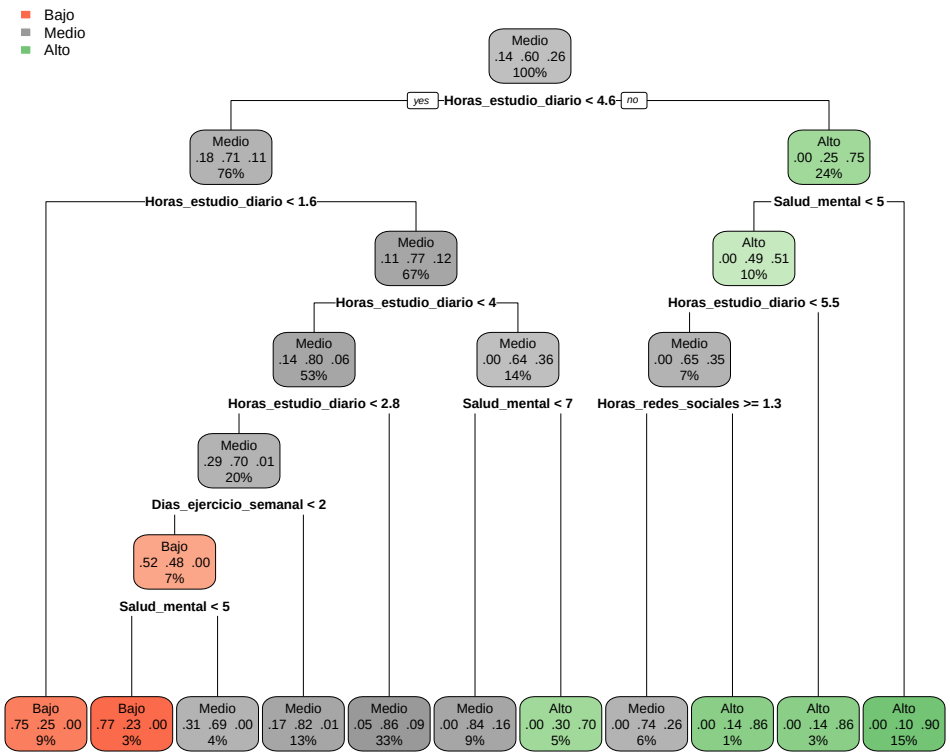

Figura 15: Árbol de decisión generado para predecir el rendimiento académico.

Este árbol muestra una estructura clara, donde la variable `Horas_estudio_diario` aparece como principal factor discriminante. Además de esta, el árbol utiliza las variables textttHoras_redes_sociales, `Dias_ejercicio_semanal` y `Salud_mental`. Cada nodo hoja muestra el porcentaje de observaciones que contiene y las proporciones de cada clase (*Bajo*, *Medio*, *Alto*). Algunas conclusiones que se pueden sacar de esta representación son:

- Los estudiantes con pocas horas de estudio (menos de 1.6 horas diarias) y baja salud mental tienden a clasificarse como de *bajo rendimiento*.
- En la mayoría de nodos intermedios predomina la clase *Medio*, lo que indica que es el perfil más común.
- Los estudiantes con un número elevado de horas de estudio (más de 5.5 horas diarias), combinadas con bajo uso de redes sociales o altos niveles de salud mental, se asocian a una mayor probabilidad de pertenecer a la clase *Alto*.

A continuación, se evaluó el rendimiento del modelo sobre el conjunto de prueba. Tras predecir la clase de cada observación con el árbol generado, se obtuvo la siguiente matriz de confusión:

```
> confusionMatrix(pred_arbol, test$Rendimiento)
Confusion Matrix and Statistics

          Reference
Prediction Bajo Medio Alto
     Bajo    24    10    0
```

```
      Medio    10    154    30
      Alto      0     11    61

Overall Statistics

                Accuracy : 0.7967
```

Como muestra el parámetro *Accuracy*, el modelo logró clasificar correctamente el 79.67 % de las observaciones del conjunto de prueba, una precisión notable considerando la simplicidad del árbol. De los 34 estudiantes de bajo rendimiento, el modelo clasificó correctamente a 24 de ellos, y se equivocó en 10, asignándolos erróneamente a la clase *medio*. Entre los 194 estudiantes con rendimiento medio, el modelo acertó en 154 casos. Sin embargo, 10 fueron clasificados como *bajo* y 30 como *alt*, lo cual muestra cierta dificultad en distinguir claramente entre clases adyacentes. Finalmente, para la clase *alto* el modelo acertó en 61 de los 72 casos, y cometió 11 errores, asignándolos como de la clase *medio*.

5.2.2. Random Forest

Para tratar de mejorar la precisión de la clasificación, se aplicó la técnica *Random Forest*. Este método construye múltiples árboles de decisión sobre distintas particiones aleatorias del conjunto de entrenamiento y promedia sus predicciones, lo cual suele reducir la varianza y mejorar el rendimiento del modelo.

Se comenzó utilizando la función `randomForest()` para ajustar el modelo sobre el conjunto de entrenamiento. Se evaluó el comportamiento del error OOB a medida que aumentaba el número de árboles mediante el gráfico mostrado en la Figura 16. Este error estima el comportamiento que tendrá el modelo con nuevos datos. En base a la evolución del error, se seleccionaron 140 árboles, donde el error se estabiliza. Para llegar a esta cifra, se utilizó el código siguiente, ya que resulta complicado seleccionar el número de árboles adecuado a partir del gráfico:

```
> modelo_rf <- randomForest(Rendimiento ~ ., data = datos_clasif_sup[train,
    ],ntree = 500, do.trace = 20)
ntree      OOB      1      2      3
   20:  24.43% 58.76% 11.96% 34.59%
   40:  19.86% 53.61%  6.70% 31.89%
   60:  19.14% 52.58%  6.22% 30.81%
   80:  19.86% 58.76%  5.26% 32.43%
  100:  19.29% 60.82%  5.50% 28.65%
  120:  19.86% 63.92%  5.50% 29.19%
  140:  19.00% 61.86%  4.78% 28.65%
  160:  18.57% 61.86%  4.78% 27.03%
  180:  19.00% 61.86%  5.02% 28.11%
  200:  19.14% 61.86%  5.02% 28.65%
  220:  18.43% 61.86%  5.02% 25.95%
  240:  18.00% 61.86%  4.31% 25.95%
  260:  17.57% 60.82%  4.55% 24.32%
  280:  17.71% 60.82%  4.55% 24.86%
  300:  17.43% 60.82%  4.31% 24.32%
  320:  17.57% 60.82%  4.31% 24.86%
  340:  18.00% 62.89%  4.31% 25.41%
  360:  17.57% 61.86%  4.31% 24.32%
  380:  17.86% 62.89%  4.07% 25.41%
```

```
400:   17.71 %  62.89 %   4.07 %  24.86 %
420:   17.71 %  62.89 %   4.07 %  24.86 %
440:   18.00 %  62.89 %   4.31 %  25.41 %
460:   17.86 %  63.92 %   4.07 %  24.86 %
480:   17.71 %  63.92 %   3.83 %  24.86 %
500:   18.00 %  63.92 %   4.07 %  25.41 %
```

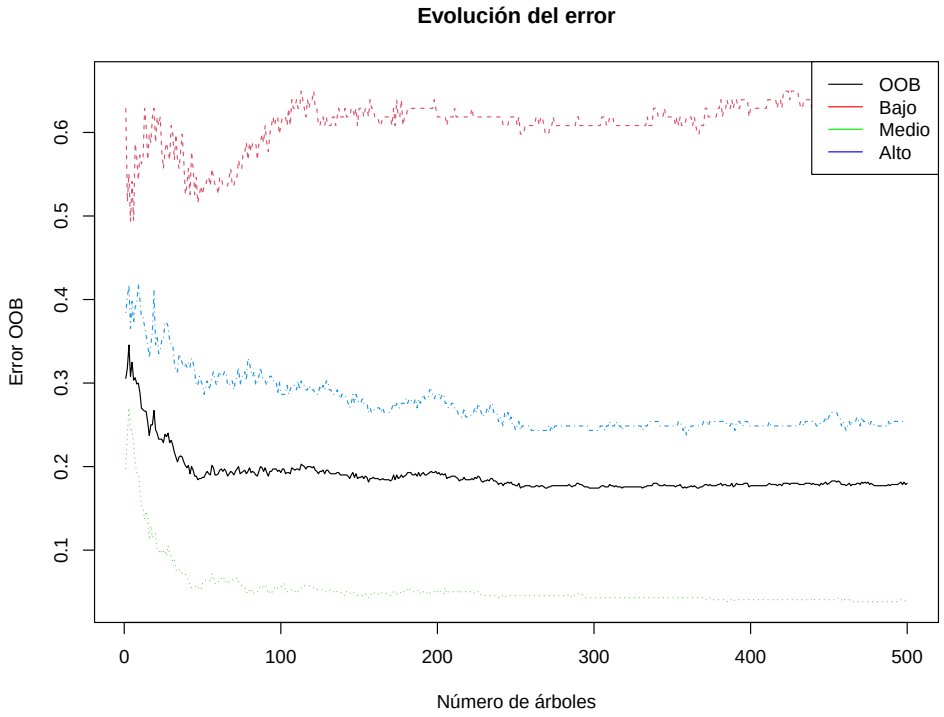

Figura 16: Evolución del error OOB en función del número de árboles.

A continuación, se evaluó el modelo sobre el conjunto de prueba. Se predijeron las clases de los individuos y se compararon con las verdaderas etiquetas mediante la siguiente matriz de confusión:

```
> confusionMatrix(pred_rf, test$Rendimiento)
Confusion Matrix and Statistics

          Reference
Prediction Bajo Medio Alto
     Bajo    18     2    0
     Medio   16   171   37
     Alto     0     2   54

Overall Statistics

               Accuracy : 0.81
```

Como puede observarse, el modelo mejoró ligeramente el rendimiento respecto al árbol de decisión, alcanzando una precisión del **81 %**. La mejora se refleja especialmente en la clase *medio*, con 17 aciertos más que en el modelo anterior.

Por último, se generó la gráfica de importancia de variables, que se muestra en la Figura 17.

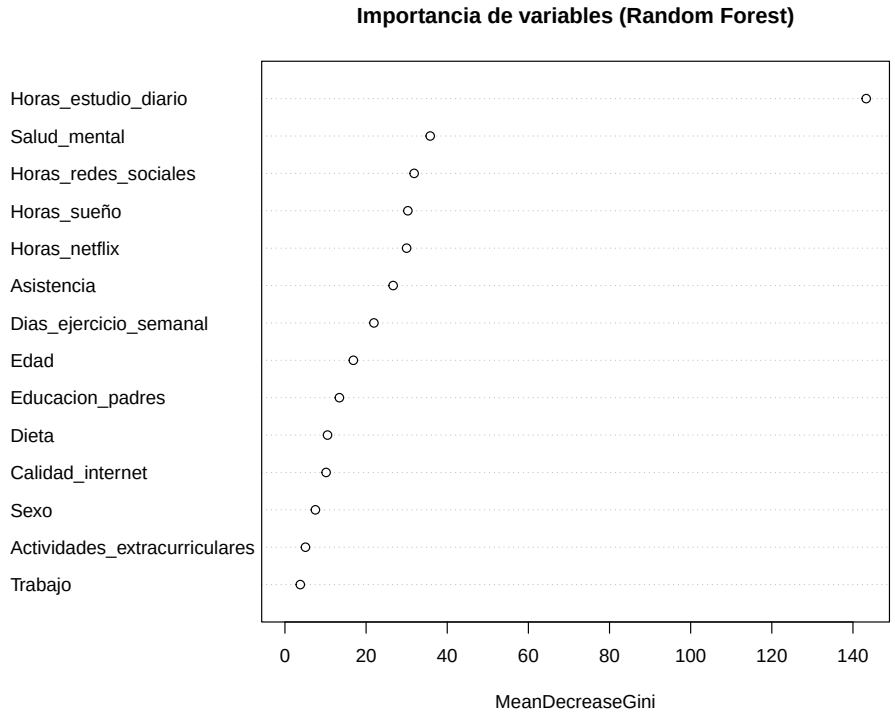

Figura 17: Importancia de las variables según el modelo Random Forest.

La variable más importante fue `Horas_estudio_diario`, con una notoria diferencia respecto al resto; seguida de `Salud_mental` y `Horas_redes_sociales`, lo que concuerda con los resultados previos del árbol de decisión. Estas variables parecen ser las más relevantes para predecir el rendimiento académico en este conjunto de datos.

5.2.3. Boosting

Finalmente, se aplicó la técnica de *boosting*, un método de ensamble basado en la combinación de múltiples árboles de decisión entrenados secuencialmente, de forma que cada modelo corrige los errores del anterior.

Para ajustar el modelo, se empleó el algoritmo `gbm` (Gradient Boosting Machine) mediante la función `train()` del paquete `caret`. En primer lugar, se realizó una búsqueda de los mejores parámetros, de esta forma se pudieron optimizar los siguientes valores: profundidades de los árboles (`interaction.depth`), tasas de aprendizaje (`shrinkage`), número de árboles (`n.trees`) y mínimo número de observaciones en nodo terminal (`n.minobsinnode`).

El mejor modelo, fue el que utilizó 500 árboles de profundidad 3, una tasa de aprendizaje (`shrinkage`) de 0.1 y al menos 20 observaciones por nodo, como muestra este código:

```
> modelo_boost$bestTune
   n.trees interaction.depth shrinkage n.minobsinnode
     500                 1       0.1             20
```

Una vez entrenado el modelo con estos parámetros, se realizaron las predicciones sobre el conjunto de prueba. La matriz de confusión resultante se muestra a continuación:

```
> confusionMatrix(pred_boost, test$Rendimiento)
Confusion Matrix and Statistics

          Reference
Prediction Bajo Medio Alto
     Bajo    26     8    0
     Medio    8   155   28
     Alto     0    12   63

Overall Statistics

               Accuracy : 0.8133
```

El modelo alcanzó una precisión del 81.33 %, mejorando ligeramente al modelo de *random forest* (81 %).

En cuanto a la importancia de las variables, la Figura 18 muestra la relevancia de cada variable para el modelo según la reducción del error cuadrático medio.

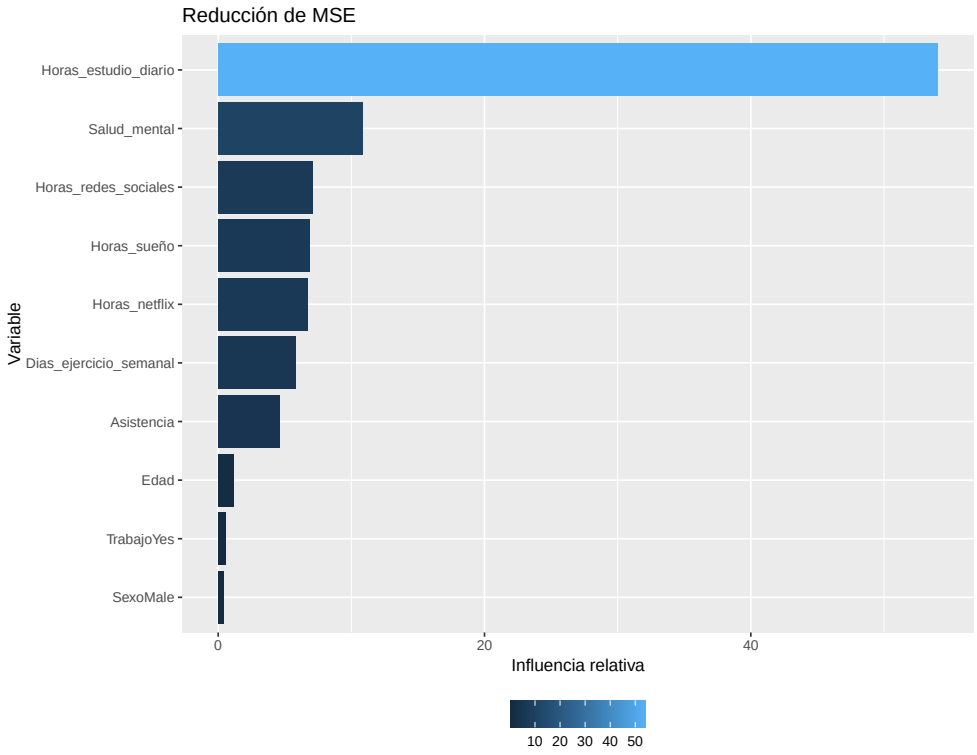

Figura 18: Importancia de las variables en el modelo de boosting.

Como puede observarse, el modelo de boosting también destaca `Horas_estudio_diario` como la variable más influyente, seguida por `Salud_mental` y `Horas_redes_sociales`. Esto coincide con los patrones detectados en los modelos anteriores, reforzando la idea de que estas variables son determinantes en el rendimiento académico.

Finalmente, seleccionamos este último modelo como el definitivo entre los modelos de clasificación supervisada que hemos construido, ya que es el que mayor precisión nos

reporta.

Conclusión

Los modelos de clasificación supervisada aplicados han mostrado una buena capacidad para predecir el rendimiento académico de los estudiantes a partir de sus hábitos y estado de salud mental. En todos los casos, se obtuvieron precisiones superiores al 79 %, siendo los métodos de*Random Forest* y *Boosting* los que ofrecieron mejores resultados.

El modelo de *boosting*, en particular, alcanzó la mayor precisión (81.33 %), superando ligeramente a los otros. Además, todos los modelos coincidieron en seleccionar las variables `Horas_estudio_diario`, `Salud_mental` y `Horas_redes_sociales` como las más relevantes para la predicción del rendimiento, lo que refuerza la coherencia del análisis.

Estos resultados muestran que es posible identificar patrones predictivos relevantes mediante técnicas supervisadas, siempre que se disponga de datos suficientes y adecuadamente tratados.

6 Conclusión

A lo largo de este trabajo se ha realizado un análisis detallado del rendimiento académico en función de los hábitos de los alumnos, utilizando un conjunto de datos simulado de estudiantes. A través de diferentes técnicas estadísticas, se ha buscado no solo modelizar la calificación final, sino también entender los factores que más influyen en su variación.

En primer lugar, el análisis descriptivo permitió obtener una visión clara del comportamiento de las variables, tanto numéricas como categóricas. Se identificaron relaciones interesantes como la fuerte correlación entre horas de estudio y calificación, o la influencia negativa del uso excesivo de redes sociales.

En el análisis de regresión se construyó un modelo lineal múltiple que explicó más del 90 % de la variabilidad en las calificaciones. Posteriormente, se aplicaron técnicas de selección de variables, logrando modelos más simples con un rendimiento similar, lo que mejora la interpretabilidad sin pérdida de precisión.

En cuanto a la reducción de dimensionalidad, los indicadores estadísticos mostraron que ni el análisis de componentes principales ni el análisis factorial eran adecuados, al no existir suficiente estructura de correlación entre las variables.

En la parte de clasificación no supervisada no se lograron detectar perfiles bien diferenciados de estudiantes en función de sus hábitos, aunque con el uso de las técnicas correctas se consiguió segmentar a los estudiantes en cuatro grupos, a pesar de presentar estos diferencias poco marcadas. Finalmente, en la clasificación supervisada, se logró predecir el rendimiento académico categorizado (*bajo, medio, alto*) con modelos que alcanzaron precisiones cercanas al 81 %, destacando el modelo de *boosting* como el más efectivo.

En conjunto, los resultados obtenidos permiten concluir que, aunque no existen agrupamientos naturales evidentes entre los estudiantes, sí es posible predecir su rendimiento

académico con un grado razonable de precisión a partir de un conjunto reducido de variables, siendo las horas de estudio, la salud mental y el uso de redes sociales los principales determinantes.

A Código R Completo

```r
#=================================================
# CARGA DE PAQUETES
#=================================================
library(psych)
library(psychTools)
library(ggfortify)
library(ggplot2)
library(car)
library(MASS)
library(leaps)
library(factoextra)
library(cluster)
library(clusterSim)
library(dendextend)
library(rpart)
library(rpart.plot)
library(caret)
library(randomForest)
library(gbm)

#=================================================
# CARGA Y LIMPIEZA DE DATOS
#=================================================

# Cargar datos
datos <- read.table("student_habits_performance.csv", sep = ",",
   dec = ".", header = TRUE)

# Vista preliminar
head(datos)

# Comprobación de valores faltantes
sum(is.na(datos))  # No hay valores faltantes

# Eliminar columna de ID (irrelevante para el análisis)
datos <- datos[, -1]

# Renombrar columnas a español
colnames(datos) <- c("Edad", "Sexo", "Horas_estudio_diario", "
   Horas_redes_sociales", "Horas_netflix",
                     "Trabajo", "Asistencia", "Horas_sueño", "
                        Dieta", "Dias_ejercicio_semanal",
                     "Educacion_padres", "Calidad_internet", "
                        Salud_mental",
                     "Actividades_extracurriculares", "
                        Calificacion")

# Conversión de variables character a factor
str(datos)
```

```r
datos$Sexo <- as.factor(datos$Sexo)
datos$Trabajo <- as.factor(datos$Trabajo)
datos$Dieta <- as.factor(datos$Dieta)
datos$Educacion_padres <- as.factor(datos$Educacion_padres)
datos$Calidad_internet <- as.factor(datos$Calidad_internet)
datos$Actividades_extracurriculares <- as.factor(datos$
   Actividades_extracurriculares)

# Escalar variables a formatos adecuados
datos$Calificacion <- datos$Calificacion / 10  # De 0-100 a
   escala 0-10
datos$Asistencia   <- datos$Asistencia / 100   # De porcentaje a
   proporción (0-1)

#=================================================
# ANÁLISIS DESCRIPTIVO
#=================================================

# Estadísticos descriptivos básicos
describe(datos)

# Boxplots para detección visual de outliers
par(mfrow = c(3, 3))
boxplot(datos$Edad, main = "Edad")
boxplot(datos$Horas_estudio_diario, main = "Horas de Estudio
   Diario")
boxplot(datos$Horas_redes_sociales, main = "Horas en Redes
   Sociales")
boxplot(datos$Horas_netflix, main = "Horas en Netflix")
boxplot(datos$Asistencia, main = "Asistencia")
boxplot(datos$Horas_sueño, main = "Horas de Sueño")
boxplot(datos$Dias_ejercicio_semanal, main = "Días de Ejercicio
   Semanal")
boxplot(datos$Salud_mental, main = "Salud Mental")
boxplot(datos$Calificacion, main = "Calificación")

# Histogramas para distribución de variables numéricas
par(mfrow = c(3, 3))
hist(datos$Edad, main = "Histograma de Edad", xlab = "Edad", col
   = "lightblue", border = "black")
hist(datos$Horas_estudio_diario, main = "Histograma de Horas
   Estudio", xlab = "Horas", col = "lightblue", border = "black")
hist(datos$Horas_redes_sociales, main = "Histograma de Redes
   Sociales", xlab = "Horas", col = "lightblue", border = "black")
hist(datos$Horas_netflix, main = "Histograma de Netflix", xlab =
   "Horas", col = "lightblue", border = "black")
hist(datos$Asistencia, main = "Histograma de Asistencia", xlab =
   "Asistencia", col = "lightblue", border = "black")
hist(datos$Horas_sueño, main = "Histograma de Horas de Sueño",
   xlab = "Horas", col = "lightblue", border = "black")
```

```r
hist(datos$Dias_ejercicio_semanal, main = "Histograma de
   Ejercicio", xlab = "Días", col = "lightblue", border = "black")
hist(datos$Salud_mental, main = "Histograma de Salud Mental",
   xlab = "Puntuación", col = "lightblue", border = "black")
hist(datos$Calificacion, main = "Histograma de Calificación",
   xlab = "Calificación", col = "lightblue", border = "black")

# Gráficos de barras para variables categóricas
par(mfrow = c(2, 3))
barplot(table(datos$Sexo), main = "Sexo", col = "lightgreen",
   border = "black")
barplot(table(datos$Trabajo), main = "Trabajo", col = "lightgreen
   ", border = "black")
barplot(table(datos$Dieta), main = "Dieta", col = "lightgreen",
   border = "black")
barplot(table(datos$Educacion_padres), main = "Nivel de Educación
    de los Padres",
        col = "lightgreen", border = "black", las = 2)
barplot(table(datos$Calidad_internet), main = "Calidad de
   Internet",
        col = "lightgreen", border = "black", las = 2)
barplot(table(datos$Actividades_extracurriculares), main = "
   Participación en Act. Extracurriculares",
        col = "lightgreen", border = "black")

#=====================================================
# MATRIZ DE CORRELACIONES
#=====================================================

# Matriz de correlaciones entre variables numéricas
cor_matrix <- cor(datos[, sapply(datos, is.numeric)])

# Visualización de correlaciones
corPlot(cor_matrix, numbers = TRUE, upper = FALSE, xlas = 2)



#=====================================================
# REGRESIÓN LINEAL MÚLTIPLE
#=====================================================

# División de los datos en entrenamiento y prueba
set.seed(1009)
train <- sample(nrow(datos), 0.7 * nrow(datos))
test  <- datos[-train, ]

# Modelo inicial con todas las variables
modelo1 <- lm(Calificacion ~ ., data = datos[train, ])
summary(modelo1)
```

```r
### DIAGNÓSTICO DEL MODELO INICIAL

# Evaluación global del modelo (normalidad de los residuos y
   colinealidad)
check_model(modelo1, check = c("vif", "normality"))
vif(modelo1)

# Gráficos de diagnóstico
autoplot(modelo1, which = 1:4)


### EVALUACIÓN PREDICTIVA DEL MODELO INICIAL

# Predicciones sobre el conjunto de prueba
predicciones <- predict(modelo1, test)

# Gráfico de predicciones vs valores reales
par(mfrow=c(1,1))
plot(predicciones, test$Calificacion, pch = 20,
     xlab = "Predicciones", ylab = "Calificación real",
     main = "Predicciones vs Valores reales")

# Métricas de rendimiento
cor(predicciones, test$Calificacion)
model_performance(modelo1)


### MODELO REDUCIDO: SELECCIÓN AUTOMÁTICA

# Modelo reducido mediante stepwise (stepAIC)
modelo.red <- stepAIC(modelo1, direction = "both", trace = TRUE)
summary(modelo.red)

# Evaluación global del modelo (normalidad de los residuos y
   colinealidad)
check_model(modelo.red, check = c("vif", "normality"))
vif(modelo.red)

# Gráficos de diagnóstico
autoplot(modelo.red, which = 1:4)

# Evaluación del modelo reducido
predicciones.red <- predict(modelo.red, test)

# Gráfico de predicciones vs valores reales (modelo reducido)
plot(predicciones.red, test$Calificacion, pch = 20,
     xlab = "Predicciones", ylab = "Calificación real",
     main = "Predicciones vs Valores reales (modelo reducido)")

# Métricas de rendimiento
cor(predicciones.red, test$Calificacion)
```

```r
model_performance(modelo.red)

#Comparación de los modelos
compare_performance(modelo1, modelo.red, verbose = FALSE)



### MODELO REDUCIDO (SELECCIÓN POR SUBCONJUNTOS)

# Aplicación del algoritmo
regfit <- regsubsets(Calificacion ~ ., data = datos[train, ])
res <- summary(regfit)
res

# Gráficos: evolución del R  ajustado y BIC
par(mfrow = c(1, 2), mar = c(4, 4, 2, 1))
plot(1:8, res$adjr2, type = "l", xlab = "Número de parámetros",
   ylab = "R  ajustado", main = "R  ajustado")
plot(1:8, res$bic, type = "l", xlab = "Número de parámetros",
   ylab = "BIC", main = "Bayesian Information Criterion")

# Variables seleccionadas en el mejor modelo (según R  ajustado)
mejor.r <- which.max(res$adjr2)
res$adjr2[mejor.r]
coef(regfit, mejor.r)

# Construcción del modelo con las variables seleccionadas
modelo.sub <- lm(Calificacion ~ Horas_estudio_diario +
                    Horas_redes_sociales +
                    Horas_netflix +
                    Asistencia +
                    Horas_sueño +
                    Dias_ejercicio_semanal +
                    Calidad_internet +
                    Salud_mental,
                  data = datos[train, ])
summary(modelo.sub)

# Diagnóstico del modelo por subconjuntos
check_model(modelo.sub, check = c("vif", "normality"))
vif(modelo.sub)
autoplot(modelo.sub, which = 1:4)

# Evaluación predictiva
predicciones.sub <- predict(modelo.sub, test)

# Gráfico de dispersión: predicciones vs reales
par(mfrow=c(1,1))
plot(predicciones.sub, test$Calificacion, pch = 20,
     xlab = "Predicciones", ylab = "Calificación real",
     main = "Predicciones vs Valores reales")
```

```r
# Métricas de rendimiento
cor(predicciones.sub, test$Calificacion)
model_performance(modelo.sub)


### COMPARACIÓN DE LOS 3 MODELOS

compare_performance(modelo1, modelo.red, modelo.sub, rank = TRUE)



#=====================================================
# REDUCCIÓN DE LA DIMENSIONALIDAD
#=====================================================

#=====================================================
# ANÁLISIS DE COMPONENTES PRINCIPALES (ACP)
#=====================================================

# Seleccionar solo las variables numéricas
vars_numericas <- datos[, c("Edad", "Horas_estudio_diario", "
   Horas_redes_sociales",
                            "Horas_netflix", "Asistencia", "Horas
                               _sueño",
                            "Dias_ejercicio_semanal", "Salud_
                               mental", "Calificacion")]

# Aplicar ACP usando la matriz de correlaciones (equivale a
   estandarizar)
CP <- princomp(vars_numericas, cor = TRUE)

# Resumen de componentes y varianza explicada
summary(CP)

# Scree plot: autovalores de los componentes
fviz_eig(CP, addlabels = TRUE)

#No es aplicable el ACP

#=====================================================
# ANÁLISIS FACTORIAL
#=====================================================

#Determinante de la matriz de correlación
det(cor_matrix)

#Medida de kaiser-M-O
print(KMO(cor_matrix),digits=4)

#No es aplicable el análisis factorial
```

```r
#=================================================
# CLASIFICACIÓN
#=================================================

#=================================================
# CLASIFICACIÓN NO SUPERVISADA
#=================================================

### PREPARACIÓN DE LOS DATOS

# Selección de variables numéricas relevantes (comentar una de
   las dos)
#  datos_clasif <- datos[, c("Horas_estudio_diario",
                            # "Horas_redes_sociales",
                            # "Horas_netflix",
                            # "Horas_sueño",
                            # "Dias_ejercicio_semanal",
                            # "Asistencia",
                            # "Salud_mental")]

datos_clasif <- datos[, c("Horas_estudio_diario",
                          "Horas_redes_sociales",
                          "Horas_netflix",
                          "Horas_sueño")]

# Estandarización de las variables
datos_clasif <- scale(datos_clasif)


# Matriz de distancias

prox <- dist(datos_clasif, method = "euclidean")


### CLUSTERING JERÁRQUICO (WARD)

# Aplicación del método de agrupación de Ward
agrupacion <- hclust(prox, method = "ward.D2")

# Representación del dendrograma con colores por grupo
fviz_dend(agrupacion, main = "Dendrograma", k = 4,
          cex = 0.7,
          color_labels_by_k = T,
          rect = T)


# GRUPOS
# Corte del dendrograma en 4 grupos
```

```r
grupos <- cutree(agrupacion, k = 4)

# Tamaño de cada grupo
table(grupos)

#Representación de los grupos en las dos primeras componentes
   principales
fviz_cluster(list(data = datos_clasif, cluster = grupos),
             main = "Representación de los grupos",
             ellipse.type = "convex",
             repel = T,
             show.clust.cent = T)


### EVALUACIÓN DE LA AGRUPACIÓN

# Coeficiente de silueta promedio
sil <- silhouette(grupos, dist(datos_clasif))
mean(sil[, 3])

#=====================================================
# k-MEDIAS
#=====================================================

# Elección del número óptimo de grupos
fviz_nbclust(datos_clasif, kmeans, method = "wss")

# Tomamos k = 4
kmedias <- kmeans(datos_clasif, centers = 4, nstart = 25)

# Estructura de los grupos
table(kmedias$cluster)

# Visualización en espacio reducido (ACP)
fviz_cluster(kmedias, data = datos_clasif,
             main = "Clustering k-means",
             ellipse.type = "convex",
             palette = "Dark2",
             repel = T,
             show.clust.cent = T)

#Evaluación de la agrupación: Coeficiente de silueta promedio
sil <- silhouette(kmedias$cluster, dist(datos_clasif))
mean(sil[, 3])


# Interpretación de cada grupo

datos_clasif_df <- as.data.frame(datos_clasif)
datos_clasif_df$cluster <- as.factor(kmedias$cluster)
aggregate(. ~ cluster, data = datos_clasif_df, FUN = mean)
```

```r
#=================================================
# CLASIFICACIÓN SUPERVISADA
#=================================================


### PREPARACIÓN DE LOS DATOS

# Transformamos la variable Calificación a factor con 3 niveles
datos$Rendimiento <- cut(datos$Calificacion,
                         breaks = c(0, 5, 8, 10),
                         labels = c("Bajo", "Medio", "Alto"),
                         right = T)

# Comprobamos la distribución de clases
table(datos$Rendimiento)

# Eliminamos la variable original Calificación
datos_clasif_sup <- datos [,-15]


# DIVISIÓN ENTRENAMIENTO/PRUEBA

train <- sample(nrow(datos_clasif_sup), 0.7 * nrow(datos_clasif_
   sup))
test  <- datos_clasif_sup[-train, ]

#=================================================
# ÁRBOL DE DECISIÓN
#=================================================
# Árbol sin restricción (cp = 0) para visualizar el error y
   seleccionar el cp óptimo
arbol_completo <- rpart(Rendimiento ~ ., data = datos_clasif_sup[
   train, ],
                        method = "class", cp = 0)

# Gráfico de complejidad para encontrar el cp óptimo
plotcp(arbol_completo)

# Tras observar el gráfico, elegimos un cp que minimice el error
   de validación cruzada
arbol_podado <- rpart(Rendimiento ~ ., data = datos_clasif_sup[
   train, ],
                      method = "class", cp = 0.013)

#Visualización del árbol
rpart.plot(arbol_podado,
           extra = 104)
```

```r
### PREDICCIÓN Y EVALUACIÓN

# Predicción sobre el conjunto de prueba
pred_arbol <- predict(arbol_podado, newdata = test, type = "class
    ")

# Matriz de confusión
confusionMatrix(pred_arbol, test$Rendimiento)


#=====================================================
# RANDOM FOREST
#=====================================================

# Entrenamiento del modelo
set.seed(1009)
modelo_rf <- randomForest(Rendimiento ~ ., data = datos_clasif_
    sup[train, ],ntree = 500, do.trace = 20)

colores <- c("black", "red", "green", "blue")
matplot(modelo_rf$err.rate,
        type="l",
        xlab="Número de árboles",
        ylab="Error OOB",
        main="Evolución del error")
legend("topright",
       colnames(modelo_rf$err.rate),
       col = colores,
       lty = 1)

#Tomo 140 árboles
set.seed(1009)
modelo_rf <- randomForest(Rendimiento ~ ., data = datos_clasif_
    sup[train, ],ntree = 140)

### PREDICCIÓN Y EVALUACIÓN

# Predicción sobre el conjunto de prueba
pred_rf <- predict(modelo_rf, newdata = test)

# Matriz de confusión
confusionMatrix(pred_rf, test$Rendimiento)

#Importancia de las variables
varImpPlot(modelo_rf, main = "Importancia de variables (Random
    Forest)")


#=====================================================
# BOOSTING
```

```r
#=================================================

#Búsqueda de los mejores hiperparámetros
validacion <- trainControl(method = "cv", number = 10)

hiperparametros <- expand.grid(interaction.depth = c(1, 3),
                               n.trees = c(100, 500),
                               shrinkage = c(0.1, 0.01, 0.001),
                               n.minobsinnode = c(1, 10, 20))

set.seed(1009)
modelo_boost <- train(
  Rendimiento ~ .,
  data = datos_clasif_sup[train, ],
  method = "gbm",
  trControl = validacion,
  tuneGrid = hiperparametros,
  verbose = FALSE
)

modelo_boost$bestTune

# Predicciones
pred_boost <- predict(modelo_boost, newdata = test)

# Matriz de confusión
confusionMatrix(pred_boost, test$Rendimiento)

#Importancia de las variables

imp <- summary(modelo_boost, plotit = F)
ggplot(imp[1:10,],
       aes(x = reorder(var, rel.inf),
           y = rel.inf,
           fill = rel.inf)) +
  geom_col() +
  coord_flip() +
  labs(x = "Variable",
       y = "Influencia relativa",
       title = "Reducción de MSE",
       fill = NULL) +
  theme(legend.position = "bottom")
```