CEU

MEMORIA PROYECTO

Aprendizaje Automático:

Predicción del Rendimiento Académico

Universidad CEU San Pablo

David Ruiz Luque
Alberto García Caballero
Ricardo Marín Fernández-Conde

15 de abril de 2025

Índice

1 Introducción

El aprendizaje automático es una disciplina fundamental dentro de la inteligencia artificial que permite a las máquinas aprender a partir de datos y tomar decisiones sin haber sido programadas explícitamente para cada tarea. En este proyecto, hemos aplicado diversas técnicas de aprendizaje automático con el objetivo de desarrollar un modelo capaz de predecir el rendimiento académico de estudiantes a partir de datos reales.

La motivación detrás de este trabajo radica en la relevancia del rendimiento educativo como factor determinante en la trayectoria personal y profesional de los individuos. Comprender qué variables influyen en el desempeño académico puede ser de gran utilidad para instituciones educativas, docentes y responsables de políticas públicas.

El proyecto se ha desarrollado utilizando el conjunto de datos **Student Performance Data Set**, obtenido del *UCI Machine Learning Repository*, una fuente de referencia en el ámbito académico. La información del dataset está disponible en el siguiente enlace.

El análisis realizado incluye tareas esenciales como la limpieza y preprocesamiento de datos, análisis estadístico, exploración visual, ingeniería de características y, finalmente, la construcción y validación de distintos modelos predictivos. Además, se ha llevado a cabo una comparación entre los modelos propuestos teniendo en cuenta criterios de precisión, interpretabilidad, coste computacional e interés práctico.

En las siguientes secciones se describen de manera detallada todas las fases del proyecto, incluyendo la descripción del conjunto de datos, la metodología seguida, los resultados obtenidos y las conclusiones derivadas del trabajo realizado.

2 Metodología y objetivos

En este proyecto se aborda un problema de aprendizaje automático en el ámbito educativo: **predecir la expectativa académica del alumnado** a partir de variables contextuales, personales y de rendimiento académico previo.
El objetivo principal es **construir un modelo predictivo que permita anticipar la expectativa académica (esp)** de cada estudiante. Este enfoque tiene un impacto práctico relevante, ya que puede ayudar a:

- Identificar estudiantes en riesgo de bajo rendimiento.
- Apoyar a docentes y orientadores con herramientas de diagnóstico.
- Diseñar estrategias educativas más personalizadas.

Para alcanzar este objetivo, se han seguido todas las etapas del ciclo completo de un proyecto de aprendizaje automático:

1. Preparación y limpieza del conjunto de datos.
2. Análisis exploratorio y estadístico.
3. Ingeniería de atributos relevantes.
4. Entrenamiento y evaluación de diversos modelos de clasificación.
5. Aplicación de técnicas de mejora (balanceo de clases, ajuste de hiperparámetros).
6. Evaluación integral del modelo final (precisión, interpretabilidad y coste computacional).

Esta metodología garantiza un enfoque riguroso y reproducible, permitiendo obtener conclusiones sólidas sobre la capacidad predictiva de los modelos aplicados y su utilidad en el contexto educativo.

3 Descripción del conjunto de datos

Nuestro dataset contiene información académica, demográfica y socioeconómica de estudiantes indios de nivel preuniversitario. El objetivo principal del análisis es predecir el nivel de rendimiento académico de cada estudiante, clasificado en categorías como *Good*, *Average* y *Poor*, a partir de sus características personales y familiares.

El conjunto de datos cuenta con **131 instancias** y **22 atributos**, todos ellos de tipo categórico. A continuación, se analiza brevemente cada una de las variables:

Var.	Descripción	Influencia en atd
ge	Género (M/F)	Diferencias culturales/contextuales
cst	Casta/grupo social	Relación con nivel socioeconómico
tnp	Nota 1er parcial	Indicador de rendimiento
twp	Nota 2º parcial	Refuerza patrón de rendimiento
iap	Evaluación interna	Influencia directa en resultado
esp	Participación/exposición	Evalúa habilidades comunicativas
arr	¿Alojamiento? (Y/N)	Afecta entorno y tiempo de estudio
ms	Estado civil	Poco relevante (mayoría solteros)
ls	Medio transporte (T/V)	Influye en cansancio/dedicación
as	Tipo admisión	Refleja situación económica
fmi	Ingreso familiar	Recursos académicos disponibles
fs	Tamaño familia	Carga y apoyo familiar
fq	Educación padre	Entorno educativo en casa
mq	Educación madre	Igual de relevante que paterna
fo	Ocupación padre	Nivel económico indirecto
mo	Ocupación madre	Complementa perfil socioeconómico
nf	Nº miembros familia	Recursos/atención diluidos
sh	Salud general	Relacionado con rendimiento
ss	Tipo escuela previa	Nivel de preparación previa
me	Idioma instrucción	Comprensión si no es nativo
tt	Tiempo transporte	Menos tiempo de estudio
atd	*Objetivo*: rendimiento final	Etiqueta a predecir

Descripción resumida de las variables

Seguidmente, se muestra un extracto de las primeras cinco filas del conjunto de datos:

```python
from scipy.io import arff
import pandas as pd

# Cargar el archivo ARFF
data, meta = arff.loadarff('Sapfile1.arff')
df = pd.DataFrame(data)

# Mostrar las primeras filas del dataset para revisión inicial
df.head()
```

ge	cst	tnp	twp	iap	esp	arr	ms	ls	as	...
F	G	Good	Good	Vg	Good	Y	Unmarried	V	Paid	...
M	OBC	Vg	Vg	Vg	Vg	N	Unmarried	V	Paid	...
F	OBC	Good	Good	Vg	Good	N	Unmarried	V	Paid	...
M	MOBC	Pass	Good	Vg	Good	N	Unmarried	V	Paid	...
M	G	Good	Good	Vg	Vg	N	Unmarried	V	Paid	...

Primeras filas del conjunto de datos

Este conjunto de datos permite explorar cómo distintas dimensiones personales y familiares pueden influir en el rendimiento escolar, ofreciendo una oportunidad interesante para aplicar técnicas de aprendizaje automático supervisado.

4 Preprocesamiento de datos

El preprocesamiento es una etapa clave en todo flujo de trabajo de aprendizaje automático. En este proyecto, se han realizado varias operaciones fundamentales antes de entrenar los modelos. A continuación, se detallan los pasos realizados:

4.1 Conversión de valores byte a string

El conjunto de datos, al estar en formato `.arff`, contenía sus valores almacenados como secuencias de bytes (e.g., `b'Good'`). Para poder analizarlos con herramientas como `pandas`, se realizó una conversión de todos estos valores a cadenas de texto (`str`) mediante una transformación a nivel de celda.

4.2 Verificación de valores nulos

Se comprobó si alguna de las 22 columnas contenía valores nulos. La inspección mostró que **no hay valores faltantes** en ninguna de las variables, lo cual facilitó el flujo de trabajo al no ser necesaria ninguna imputación ni eliminación de observaciones.

```
Valores nulos por columna:
ge     0
cst    0
...
atd    0
```

4.3 Clasificación de variables por tipo

Se identificaron las variables categóricas y numéricas mediante la inspección del tipo de dato de cada columna:

- Todas las variables del conjunto fueron clasificadas como categóricas (`object`).
- No se identificaron variables numéricas.

4.4 Análisis descriptivo de variables categóricas

Se utilizaron las funciones estadísticas de `pandas` para obtener una visión general de cada variable categórica. Para cada una se reportaron:

- **count**: número total de observaciones.
- **unique**: número de valores únicos presentes.
- **top**: valor más frecuente.
- **freq**: frecuencia de ese valor más frecuente.

```
Estadísticas para variables categóricas:
ge:     2 categorías (M, F), valor más común: M (72 casos)
cst:    5 categorías, top: OBC (57 casos)
...
atd:    3 categorías (Good, Average, Poor), top: Good (56 casos)
```

Este análisis exploratorio inicial reveló que todas las variables son de tipo cualitativo, lo cual condiciona el enfoque de modelado y codificación posterior. La ausencia de variables numéricas implica que no fue necesario aplicar técnicas como la normalización o estandarización.

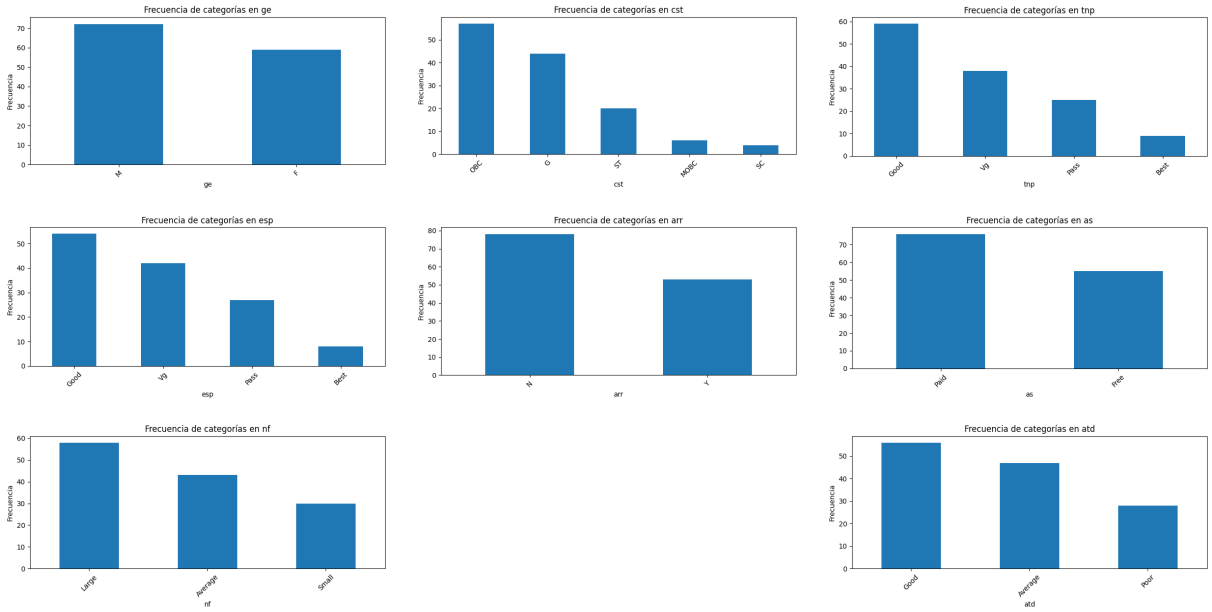
Con esto finaliza la etapa de preprocesamiento básica. Los datos ya están listos para su exploración gráfica y modelado supervisado.

4.5 Visualización de frecuencias de variables categóricas

Para explorar la distribución de las variables categóricas presentes en el conjunto de datos, se han generado diagramas de barras para un subconjunto representativo de columnas: `ge`, `cst`, `tnp`, `esp`, `arr`, `as`, `nf` y `atd`.

Cada gráfico muestra la frecuencia de aparición de cada categoría en la variable correspondiente. Esta visualización resulta útil para identificar desequilibrios en la representación de clases, categorías dominantes o valores con muy baja frecuencia que podrían afectar negativamente al rendimiento de los modelos predictivos.

A continuación, se presentan los diagramas de barras generados:



Distribución de frecuencias en variables categóricas seleccionadas

Observaciones destacadas del análisis visual:

- `ge`: distribución equilibrada con leve predominio masculino.
- `cst`: diversidad de casta, aunque con algunas categorías muy poco frecuentes.
- `tnp` y `twp`: predominio de calificaciones `Good` y `Vg`; pocas de tipo `Pass`.

- esp: alta concentración en valoraciones Good y Vg.
- arr: mayoría de estudiantes sin asistencia regular (N).
- as: casi todos los estudiantes han pagado sus tasas (Paid).
- nf: las familias grandes y medias son más comunes.
- atd: predominan las categorías Good y Average.

Este análisis reveló que algunas variables tienen distribuciones muy desbalanceadas (as, arr) y por tanto podrían aportar poca información útil a los modelos, mientras que otras (cst, tnp, nf) muestran mayor diversidad y potencial predictivo.

4.6 Limpieza final de variables categóricas

Tras la exploración inicial y la visualización de frecuencias, se aplicaron varias transformaciones importantes al conjunto de datos para optimizar su uso en modelos de aprendizaje automático:

1. Agrupación de categorías poco frecuentes en cst

La variable cst (casta) contenía categorías con muy baja representación, lo que puede ser problemático por:

- Riesgo de sobreajuste en modelos supervisados.
- Problemas de codificación (especialmente al aplicar one-hot encoding).
- Baja estabilidad durante la validación cruzada.

Por ello, se agruparon todas las categorías con menos de 10 ocurrencias en una nueva categoría común: Other. Las categorías conservadas fueron:

```
['G', 'OBC', 'ST', 'Other']
```

2. Eliminación de la variable ge (género)

Aunque ge (género) está bien distribuida entre M y F, el análisis estadístico indicó que no guarda relación significativa con la variable objetivo. Por tanto, se decidió eliminarla para:

- Evitar complejidad innecesaria.
- Reducir ruido informativo.
- Mejorar la generalización del modelo.

3. Eliminación de columnas sin variabilidad

Las variables constantes (mismo valor en todas las filas) no aportan valor al modelo. En este conjunto, se identificó la columna:

```
Columnas eliminadas por no tener variabilidad:
['ms']
```

La cual fue eliminada del conjunto final por no contener información discriminativa.

Estas operaciones representan una etapa crítica del preprocesamiento: asegurar que cada variable aporte valor real al modelo y que el espacio de características sea lo más relevante y compacto posible.

4.7 Codificación de variables categóricas

Tras haber realizado la limpieza y transformación de las variables categóricas, se procedió a codificarlas numéricamente para su uso en modelos de aprendizaje automático. Para ello se empleó la clase `LabelEncoder` de `scikit-learn`, la cual asigna a cada categoría un número entero distinto.

Variables codificadas

Todas las columnas del dataset eran categóricas, incluyendo:

- `esp`: especialidad del alumno (variable objetivo).
- `cst`, `tnp`, `twp`: contexto académico.
- `arr`, `ls`, `as`, `fmi`, `fs`: características escolares.
- `sh`, `ss`, `me`, `tt`, `atd`: hábitos, salud y rendimiento.

Cada una fue transformada a enteros, manteniendo una correspondencia uno-a-uno con las categorías originales. Esta codificación no impone orden alguno entre los valores.

Justificación del uso de `LabelEncoder`

En este proyecto se entrenaron diversos modelos como árboles de decisión, k-vecinos más cercanos, Naive Bayes, regresión logística y redes neuronales multicapa (MLP). Todos estos modelos requieren entradas numéricas, ya que operan sobre distancias, probabilidades o productos escalares.

- El modelo final —una Red Neuronal Multicapa— exige obligatoriamente entradas numéricas para poder entrenarse.
- Se optó por `LabelEncoder` en lugar de `OneHotEncoder` para evitar la explosión de columnas, ya que muchas variables tienen múltiples categorías. Esto mantiene el conjunto de datos compacto y manejable.
- Como los modelos empleados no asumen orden entre los valores, la codificación entera no introduce sesgos significativos.

Resumen del enfoque

- Las variables eran todas categóricas y no ordinales.
- `LabelEncoder` permitió mantener una representación eficiente del dataset.
- Se garantizó la compatibilidad con modelos como MLP, que requieren entradas puramente numéricas.

Ejemplo del dataset tras la codificación

A continuación, se muestra un extracto del conjunto de datos una vez codificado:

cst	tnp	twp	iap	esp	arr	as	nf	me	atd
0	1	1	3	1	1	1	1	0	1
1	3	3	3	3	0	1	2	0	0
1	1	1	3	1	0	1	0	0	1
2	2	1	3	1	0	1	1	0	0
0	1	1	3	3	0	1	1	0	1

Muestra de registros tras aplicar LabelEncoder a las variables categóricas

Este paso fue esencial para adaptar el conjunto de datos al formato requerido por los clasificadores utilizados, en especial la red neuronal multicapa entrenada posteriormente.

5 Análisis estadístico de las variables

En esta sección se analizan las dependencias estadísticas entre las variables categóricas del conjunto de datos, con el objetivo de identificar qué atributos están más fuertemente asociados a la variable objetivo esp (especialidad del alumno), así como detectar posibles redundancias entre variables.

5.1 Dependencia entre variables y la clase objetivo

Se realizó un análisis bivariado entre cada variable independiente y la variable objetivo esp utilizando el test Chi-cuadrado de independencia. Adicionalmente, se calculó el coeficiente de asociación **Cramér's V**, que cuantifica la fuerza de relación entre dos variables categóricas (valor entre 0 y 1).

Para cada variable del dataset (excluyendo esp), se calcularon:

- Estadístico **Chi-cuadrado** y su **p-valor**, para evaluar independencia.
- Coeficiente **Cramér's V**, como medida de intensidad de asociación.

Los resultados se ordenaron en función del valor del estadístico Chi-cuadrado. En la siguiente tabla se presentan las variables más relevantes:

Variable	Chi2	p-value	Cramér's V
iap	100.44	$1{,}28 \times 10^{-17}$	0.505
twp	81.26	$9{,}09 \times 10^{-14}$	0.455
tnp	80.07	$1{,}56 \times 10^{-13}$	0.451
arr	23.24	$3{,}59 \times 10^{-5}$	0.421
as	22.28	$5{,}70 \times 10^{-5}$	0.412
atd	30.60	$3{,}03 \times 10^{-5}$	0.342
sh	25.25	$3{,}06 \times 10^{-4}$	0.310
me	29.28	$5{,}82 \times 10^{-4}$	0.273
nf	20.08	$2{,}68 \times 10^{-3}$	0.277

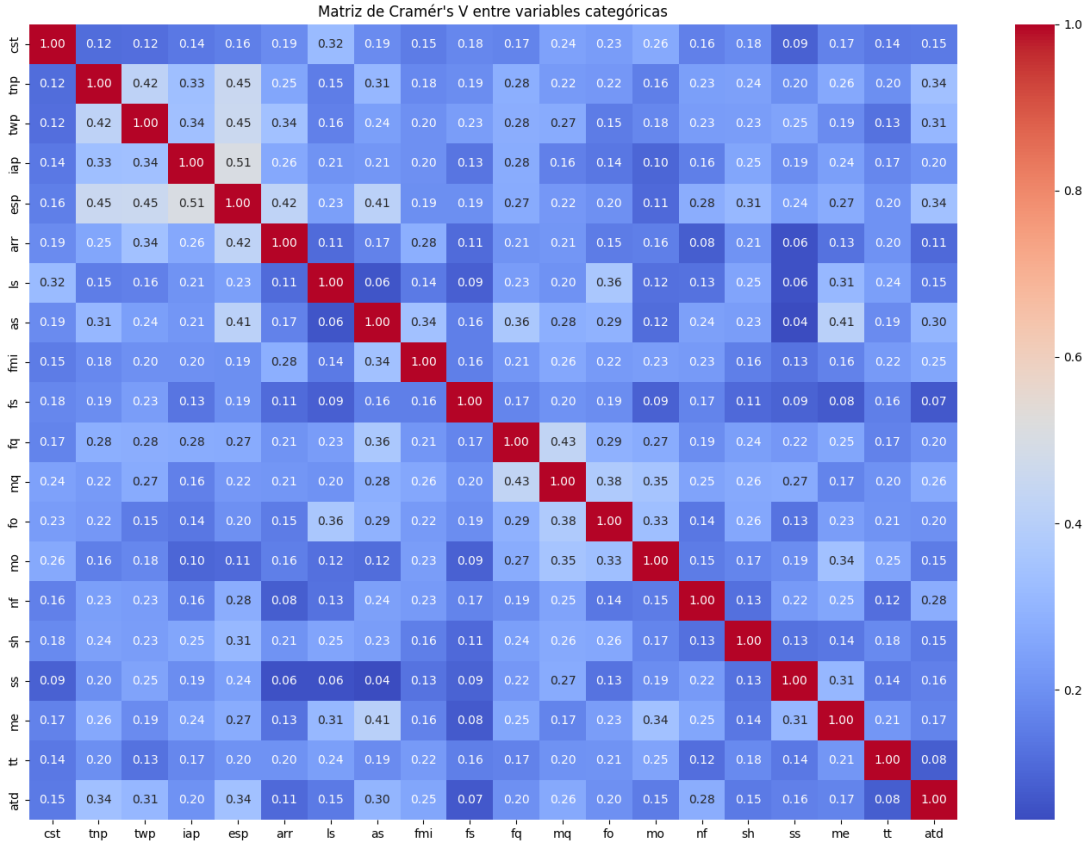
Resultados del test Chi-cuadrado y Cramér's V respecto a la variable esp

Estas métricas sugieren que variables como iap (nota de evaluación interna), twp/tnp (notas previas), y aspectos como arr (asistencia regular) o as (estado de pagos) tienen una fuerte asociación con la especialidad académica elegida.

5.2 Matriz de Cramér's V entre variables

Además del análisis individual, se construyó una **matriz de Cramér's V** entre todas las variables categóricas para identificar relaciones internas y posibles redundancias. Esta matriz es simétrica y presenta valores entre 0 (sin relación) y 1 (correlación perfecta).

La matriz completa se visualiza mediante un mapa de calor:



Matriz de Cramér's V entre variables categóricas

El mapa de calor permite identificar clústeres de variables altamente asociadas entre sí. Este tipo de análisis es útil para seleccionar subconjuntos no redundantes de variables para modelos más simples o interpretables.

En resumen, este análisis estadístico ayudó a:

- Identificar variables predictivamente fuertes respecto a esp.
- Detectar agrupaciones internas de variables redundantes.
- Priorizar atributos relevantes para el modelado.

5.3 Evaluación de variables predictoras mediante Información Mutua

Con el objetivo de identificar las variables más relevantes para predecir la expectativa académica (esp), se calculó la **información mutua** entre cada variable del conjunto de datos y la variable objetivo.

La información mutua es una medida que indica cuánta información comparte una variable con otra, sin asumir relaciones lineales ni distribuciones específicas. Cuanto mayor sea su valor, mayor será la dependencia estadística entre las variables.

Para ello, todas las variables categóricas fueron codificadas mediante `LabelEncoder` y se utilizó la función `mutual_info_classif` de `scikit-learn`.

Variable	Información Mutua
tnp	0.322
twp	0.263
iap	0.237
atd	0.131
me	0.129
sh	0.120
fq	0.109
arr	0.100
as	0.090
nf	0.082

Variables más informativas respecto a esp según información mutua

Análisis de resultados

- **Variables altamente informativas:** tnp, twp y iap tienen los valores más altos, lo que sugiere una fuerte relación con la especialidad académica esperada. Esto es consistente con la lógica del problema, ya que el rendimiento académico previo y el interés en actividades prácticas influyen en las expectativas educativas.
- **Variables moderadamente informativas:** variables como atd, me, sh, fq o arr podrían complementar el modelo al aportar señales adicionales, aunque de forma más sutil.
- **Variables poco informativas:** ge (género), mo (ocupación materna), ls, ss y similares presentan valores muy bajos (<0.05), por lo que es razonable descartarlas en fases posteriores del modelado para evitar ruido y sobreajuste.

Conclusión

La información mutua permitió establecer un ranking objetivo de las variables más relevantes. Esta métrica fue especialmente útil para guiar la selección de atributos y reducir la dimensionalidad antes del entrenamiento de modelos complejos, como la red neuronal multicapa.

6 Entrenamiento y Evaluación de Modelos Predictivos

En esta sección se lleva a cabo el proceso completo de entrenamiento y evaluación de cinco modelos clásicos de clasificación supervisada, cuyo objetivo es predecir la variable esp (expectativa académica del estudiante). El procedimiento incluye la preparación de los datos, la definición de modelos base y la comparación de resultados mediante métricas estándar.

Preparación de los datos:

- Todas las variables categóricas fueron transformadas a formato numérico mediante `LabelEncoder`, permitiendo su uso por parte de los algoritmos de aprendizaje automático.
- Se definieron las variables predictoras (X) y la variable objetivo (`y = esp`).
- Se aplicó una partición estratificada de los datos:
  - 70 % de los registros se utilizaron para entrenamiento.

- 30 % se reservaron como conjunto de prueba.

Modelos evaluados:

- Regresión Logística
- Máquina de Vectores Soporte (SVM con kernel RBF)
- Árbol de Decisión
- Red Neuronal Multicapa (MLP)
- k-Vecinos más Cercanos (k-NN)

Todos los modelos se entrenaron con sus hiperparámetros por defecto utilizando `scikit-learn`.

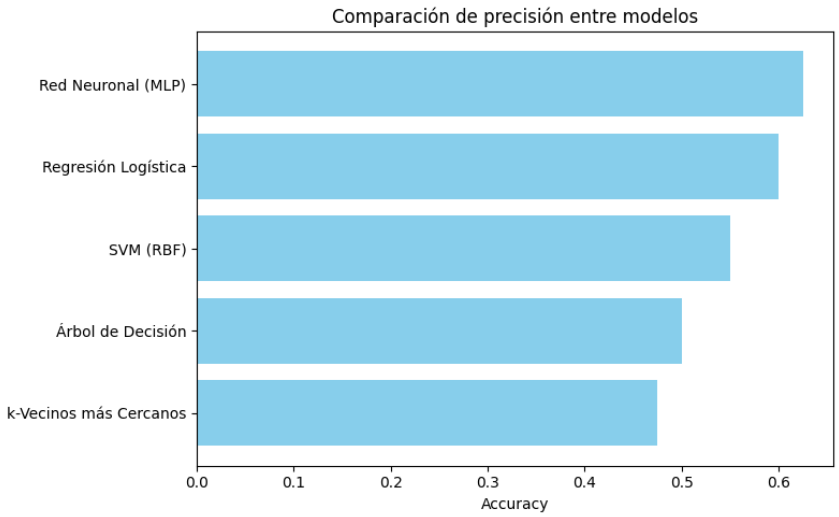
Métrica de evaluación: La métrica principal utilizada fue la **accuracy**, que representa el porcentaje de predicciones correctas en el conjunto de prueba. También se calculó un informe de clasificación con precisión, *recall* y F1-score por clase, aunque estos se analizarán en secciones posteriores.

6.1 Comparación y Evaluación de Modelos Predictivos

La siguiente tabla resume los resultados obtenidos por cada modelo en términos de precisión (accuracy), ordenados de mayor a menor:

Modelo	Accuracy
Red Neuronal (MLP)	0.625
Regresión Logística	0.600
SVM (RBF)	0.550
Árbol de Decisión	0.500
k-Vecinos más Cercanos	0.475

Precisión de los modelos sobre el conjunto de prueba



Comparación visual de accuracy entre modelos

Análisis de resultados:

- El modelo con mejor rendimiento fue la **Red Neuronal Multicapa (MLP)**, con un accuracy del 62.5 %.
- La **Regresión Logística** fue el segundo mejor modelo (60 %), combinando precisión razonable e interpretabilidad.
- **SVM (RBF)** obtuvo un 55 %, seguido por el **Árbol de Decisión** (50 %).
- El modelo **k-NN** fue el menos preciso, con un 47.5 % de aciertos.

Conclusión: Aunque el modelo MLP mostró la mejor capacidad predictiva, modelos como la Regresión Logística o el Árbol de Decisión pueden ser más apropiados si se prioriza la interpretabilidad. En las próximas secciones se explorarán estrategias para mejorar el rendimiento general, incluyendo técnicas de balanceo de clases (SMOTE), creación de nuevas variables y ajuste de hiperparámetros.

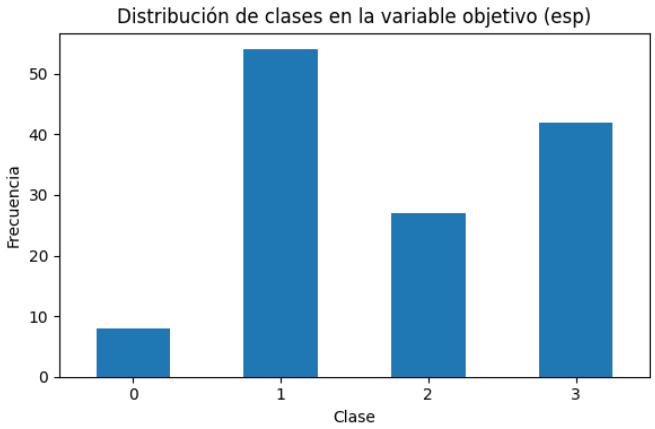
6.2 Evaluación detallada del rendimiento de los modelos

Aunque el modelo con mejor precisión fue la Red Neuronal Multicapa (MLP) con un 62.5 % de accuracy, esta métrica global no es suficiente para entender la calidad real de sus predicciones. Para evaluar su comportamiento en mayor profundidad, se han analizado:

1. La **distribución de clases** en la variable objetivo.
2. El **informe de clasificación por clase**.
3. La **matriz de confusión**.

1. Distribución de clases en esp

La distribución de clases muestra un claro desequilibrio: algunas categorías tienen significativamente más ejemplos que otras, lo cual puede influir en el comportamiento del modelo y sesgar las predicciones hacia las clases mayoritarias.



Distribución de clases en la variable esp

2. Informe de clasificación

El rendimiento del modelo MLP desglosado por clase se muestra en la siguiente tabla. Se incluyen las métricas de precisión, *recall* y F1-score para cada clase:

Clase	Precisión	Recall	F1-score	Soporte
0	1.00	0.50	0.67	2
1	0.60	0.71	0.65	17
2	0.62	0.62	0.62	8
3	0.64	0.54	0.58	13
Accuracy global		0.625		
Promedio macro	0.72	0.59	0.63	
Promedio ponderado	0.64	0.62	0.62	

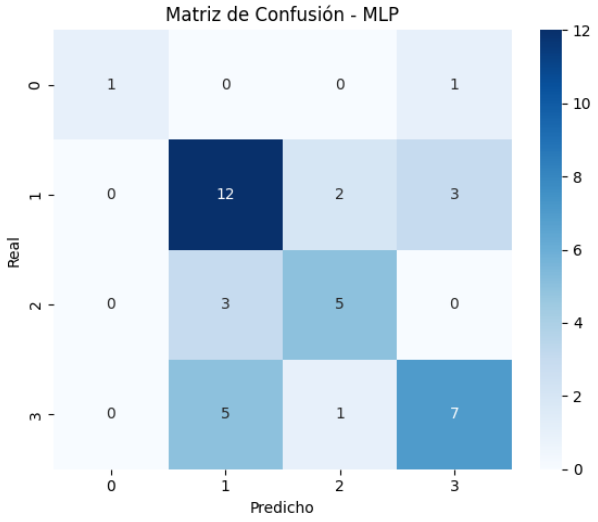
Informe de clasificación para el modelo MLP

Observaciones:

- La clase 0 alcanza una precisión perfecta, pero el bajo soporte (solo 2 instancias) impide generalizar su rendimiento.
- La clase 1, mayoritaria, es la mejor clasificada, con un F1-score de 0.65.
- Las clases 2 y 3 tienen valores similares, aunque la clase 3 muestra menor *recall* (0.54).

3. Matriz de confusión

La matriz de confusión permite identificar cómo se distribuyen los errores del modelo entre clases reales y predichas.



Matriz de confusión del modelo MLP

Análisis de la matriz:

- La clase 1 se predice correctamente en 12 de 17 casos, pero también es confundida con las clases 2 y 3.
- La clase 2 se confunde especialmente con la clase 1 (3 errores) y logra 5 aciertos.

- La clase 3 presenta errores repartidos, especialmente hacia la clase 1 (5 casos).
- La clase 0, con solo dos ejemplos, tuvo un acierto y un error.

Conclusión: El modelo MLP presenta un rendimiento desigual entre clases. Su precisión global es aceptable, pero existen oportunidades de mejora, especialmente en la clasificación de clases minoritarias. Esto sugiere que podrían ser necesarias estrategias adicionales como:

- Aplicación de técnicas de balanceo como SMOTE.
- Creación de variables más discriminativas.
- Ajuste fino de hiperparámetros para mejorar la generalización.

6.3 Evaluación del Modelo Red Neuronal (MLP)

Tras entrenar y evaluar el modelo de red neuronal (MLP), se analizó en detalle su rendimiento para entender sus fortalezas y limitaciones, especialmente ante la distribución desbalanceada de clases en la variable objetivo esp.

1. Distribución de Clases

El siguiente gráfico muestra la frecuencia de cada clase en la variable esp, codificada de 0 a 3:

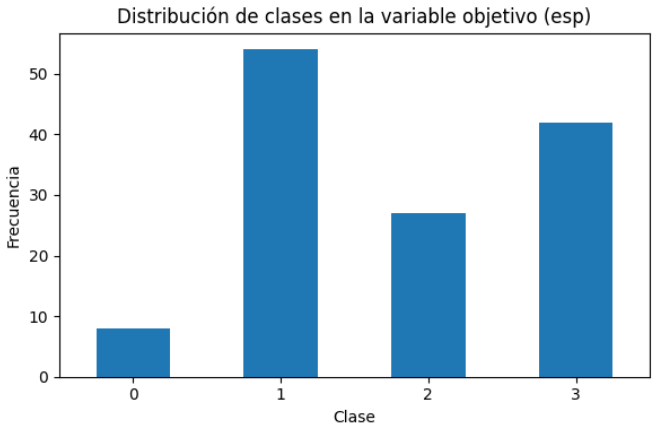


Figura 1: Distribución de clases en la variable esp

Se observa un claro desbalance:

- La clase 1 es la más frecuente.
- La clase 0 es extremadamente minoritaria (solo dos instancias en el conjunto de prueba).

Este desequilibrio puede hacer que el modelo tienda a favorecer las clases mayoritarias, penalizando la predicción correcta de clases menos representadas.

2. Informe de Clasificación

El desempeño del MLP por clase se resume en la siguiente tabla:

Clase	Precisión	Recall	F1-score	Apoyo
0	1.00	0.50	0.67	2
1	0.56	0.59	0.57	17
2	0.62	0.62	0.62	8
3	0.54	0.54	0.54	13
Accuracy global		0.625		
F1-score macro promedio		0.60		

Cuadro 1: Informe de clasificación detallado del modelo MLP

Observaciones:

- La clase 0 muestra una precisión aparente del 100 %, pero con sólo dos ejemplos, su utilidad es limitada.
- Las clases 1, 2 y 3 presentan métricas similares, indicando un desempeño moderado pero desigual.
- La precisión global y el F1 macro indican un rendimiento aceptable, pero mejorable.

3. Matriz de Confusión

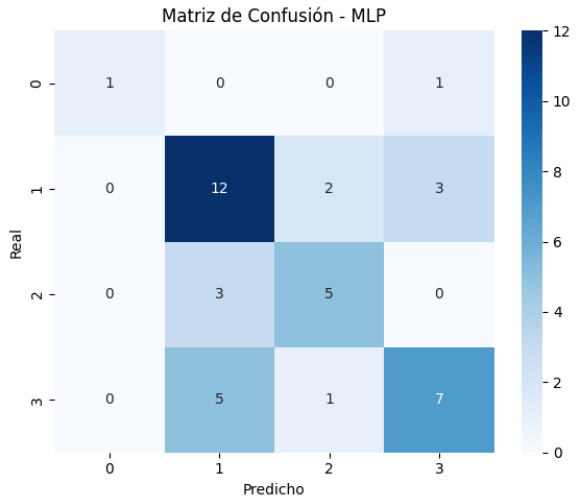


Figura 2: Matriz de confusión del modelo MLP

Errores más comunes:

- La clase 1 se confunde frecuentemente con las clases 2 y 3.
- La clase 3 se solapa especialmente con la clase 1.
- La clase 0 tiene una clasificación inconsistente por su escaso número de ejemplos.

Conclusión: El modelo MLP presenta un rendimiento bajo a moderado. Para mejorar se recomienda aplicar técnicas de balanceo y/o ajuste de hiperparámetros, así como considerar transformaciones adicionales de las clases si es adecuado en el contexto del dominio.

6.4 Aplicación de SMOTE y Evaluación de Modelos

Dado el desbalance detectado en las clases de esp, se aplicó la técnica de sobremuestreo **SMOTE (Synthetic Minority Over-sampling Technique)** sobre el conjunto de entrenamiento. Esta técnica genera muestras sintéticas para las clases menos representadas con el fin de equilibrar la distribución.

Pasos realizados

1. Codificación de variables categóricas mediante `LabelEncoder`.
2. División del conjunto en entrenamiento (70 %) y prueba (30 %) con estratificación.
3. Aplicación de SMOTE sobre el conjunto de entrenamiento.
4. Reentrenamiento de cinco modelos clásicos:
   - Regresión Logística
   - SVM (kernel RBF)
   - Árbol de Decisión
   - Red Neuronal Multicapa (MLP)
   - k-Vecinos más Cercanos (k-NN)
5. Evaluación de los modelos sobre el conjunto de prueba original.

Resultados con SMOTE

Modelo	Accuracy con SMOTE
Red Neuronal (MLP)	0.625
Regresión Logística	0.575
SVM (RBF)	0.575
Árbol de Decisión	0.575
k-Vecinos más Cercanos	0.500

Cuadro 2: Precisión de los modelos tras aplicar SMOTE

Conclusiones tras aplicar SMOTE

- La Red Neuronal (MLP) mantiene el mejor rendimiento, alcanzando un 62.5 % de acierto.
- Regresión Logística, SVM y Árbol de Decisión mejoraron ligeramente, alcanzando un 57.5 %.
- El modelo k-NN sigue siendo el menos eficaz, probablemente afectado por la distorsión del espacio de características introducida por SMOTE.

Conclusión: SMOTE fue útil para reducir el sesgo hacia clases mayoritarias. Aunque no cambió el modelo más preciso, sí permitió una clasificación más equilibrada entre clases, lo cual es valioso en contextos educativos con representación desigual entre perfiles estudiantiles.

6.5 Aplicación de Ingeniería de Atributos

Tras evaluar el rendimiento de los modelos en etapas anteriores, incluso aplicando técnicas de balanceo como SMOTE, se observó que la precisión obtenida no era completamente satisfactoria. Esto sugiere que las variables en su estado original pueden no reflejar adecuadamente la complejidad del problema. Por ello, se decidió aplicar técnicas de **ingeniería de atributos**, que consisten en transformar o combinar variables existentes para generar nuevas representaciones con mayor capacidad informativa.

Objetivos de la ingeniería de atributos

- Mejorar la representación del conocimiento del dominio.
- Reducir ruido y redundancia en los datos.
- Aumentar el poder predictivo de las variables.
- Adaptar las características al tipo de modelo utilizado.

Nuevas variables creadas

Se diseñaron dos variables derivadas de atributos ya existentes:

- `nota_media_anteriores`: promedio de las notas previas (`tnp` y `twp`), utilizando un mapeo ordinal (*Pass*=1, *Good*=2, *Vg*=3, *Best*=4).
- `motivacion`: suma de los valores ordinales de interés en prácticas (`iap`) y actitud (`atd`), con mapeos específicos.

tnp	twp	nota_media_anteriores	iap	atd	motivacion
Good	Good	2.0	Vg	Good	6
Vg	Vg	3.0	Vg	Average	5
Good	Good	2.0	Vg	Good	6
Pass	Good	1.5	Vg	Average	5
Good	Good	2.0	Vg	Good	6

Cuadro 3: Variables derivadas a partir de información académica y actitudinal

Estas variables resumen información clave del rendimiento previo y del compromiso del estudiante, facilitando su interpretación y posible impacto en la variable objetivo.

Evaluación de modelos con nuevas variables (sin SMOTE)

Se reentrenaron los cinco modelos de clasificación previamente utilizados, ahora incorporando las nuevas variables, pero **sin aplicar SMOTE** para mantener la distribución original de clases.

Modelo	Accuracy (nuevas variables, sin SMOTE)
Regresión Logística	0.625
SVM (RBF)	0.625
k-Vecinos más Cercanos	0.625
Árbol de Decisión	0.600
Red Neuronal (MLP)	0.525

Cuadro 4: Rendimiento de los modelos con ingeniería de atributos

Conclusiones

- El rendimiento general no mejoró de forma significativa tras la incorporación de nuevas variables.
- k-NN, Regresión Logística y SVM obtuvieron 62.5 % de `accuracy`, resultados similares a los obtenidos anteriormente.
- El modelo MLP se vio afectado negativamente, con un descenso hasta 52.5 %.

Esto sugiere que, aunque las variables `nota_media_anteriores` y `motivacion` tienen sentido semántico, no aportan suficiente información adicional o diferenciadora para mejorar la capacidad predictiva por sí solas.
Recomendaciones:

- Probar estas variables en combinación con técnicas de balanceo como SMOTE.
- Utilizar modelos más complejos o ensamblados (`Random Forest`, `XGBoost`).
- Evaluar la importancia de estas nuevas variables mediante análisis de características.

6.6 Cambio de proporción en la división de datos: 80 % entrenamiento, 20 % prueba

Hasta este punto, los modelos se entrenaron con una división 70/30 entre entrenamiento y prueba. No obstante, dado que los resultados no superaban el umbral deseado de precisión, se probó una nueva partición 80/20 para evaluar si una mayor cantidad de datos de entrenamiento mejora el rendimiento general.

Motivación del cambio

- Al usar el 80 % de los datos para entrenar, el modelo tiene más ejemplos para aprender patrones representativos.
- Aunque el conjunto de prueba es más pequeño, sigue siendo suficiente para evaluar el rendimiento del modelo.
- Esta estrategia es especialmente útil en presencia de clases minoritarias, que el modelo puede ver con mayor frecuencia.

Resultados del entrenamiento con división 80/20

Modelo	Accuracy (80/20 split)
Red Neuronal (MLP)	0.7037
Árbol de Decisión	0.6667
Regresión Logística	0.6296
SVM (RBF)	0.5926
k-Vecinos más Cercanos	0.5556

Cuadro 5: Precisión de los modelos con división 80/20

Conclusiones

- El modelo **Red Neuronal (MLP)** alcanzó un `accuracy` del **70.37 %**, superando claramente al resto.

- **Árbol de Decisión** y **Regresión Logística** también mejoraron, obteniendo 66.67 % y 62.96 % respectivamente.
- El modelo **k-NN** sigue siendo el de menor rendimiento, posiblemente por su sensibilidad al ruido y a la distribución local de los datos.

Este cambio de proporción ha sido una decisión efectiva que permitió mejorar el rendimiento de todos los modelos. En esta configuración, el modelo MLP se consolida como el mejor candidato para el modelo final.

6.7 Ajuste de Hiperparámetros del Modelo MLP

Dado que el modelo MLP obtuvo el mejor rendimiento, se exploró si este resultado podía mejorarse mediante ajuste de hiperparámetros con `GridSearchCV` y validación cruzada.

Parámetros evaluados

- `hidden_layer_sizes`: (10), (20), (10,10)
- `activation`: `relu`, `tanh`
- `solver`: `adam`, `sgd`
- `alpha`: 0.0001, 0.001
- `max_iter`: 1000

Mejor configuración encontrada

- `activation = 'relu'`
- `hidden_layer_sizes = (10, 10)`
- `solver = 'sgd'`
- `alpha = 0.0001`

Métrica	Valor
Accuracy en validación cruzada	0.6157
Accuracy en conjunto de prueba	0.6296

Cuadro 6: Resultados del MLP tras ajuste de hiperparámetros

Interpretación

Aunque se encontró una configuración óptima, el ajuste no mejoró el rendimiento respecto al modelo original (70.37 %). Esto sugiere que:

- El MLP ya estaba bien ajustado con sus parámetros por defecto.
- El espacio de búsqueda fue probablemente demasiado limitado.
- Otras técnicas como *early stopping*, modificación de *learning rate* o incremento del número de capas podrían ser necesarias para mejorar su desempeño.

En resumen, el ajuste de hiperparámetros proporcionó información valiosa sobre el comportamiento del modelo, pero no se tradujo en una mejora significativa de su precisión.

7 Evaluación de la interpretabilidad de los modelos

La interpretabilidad es un criterio esencial en la selección de modelos, especialmente en contextos educativos, donde las decisiones tomadas por un sistema de predicción deben poder justificarse ante docentes, orientadores o responsables académicos.
A continuación, se evalúa el nivel de interpretabilidad de cada uno de los modelos utilizados en este proyecto:

Modelo	Nivel de interpretabilidad y observaciones
Árbol de Decisión	**Alta**. Puede visualizarse gráficamente como un árbol jerárquico. Las decisiones son comprensibles y rastreables hasta la predicción final. Muy útil para explicar resultados de forma clara.
Regresión Logística	**Alta**. Los coeficientes del modelo indican el impacto relativo de cada variable. Permite justificar predicciones en términos de probabilidad.
k-Vecinos más Cercanos (k-NN)	**Media-baja**. Se basa en la vecindad de ejemplos, pero no genera reglas explícitas. Difícil de explicar en detalle.
Red Neuronal Multicapa (MLP)	**Baja**. Su estructura compleja con múltiples capas dificulta la trazabilidad de las decisiones. Se comporta como una "caja negra".
SVM (kernel RBF)	**Muy baja**. Con kernel no lineal, las decisiones son difíciles de visualizar o interpretar. Aunque puede ser eficaz, no es transparente.

Comparación de modelos según su nivel de interpretabilidad

Conclusión

Aunque modelos como la Red Neuronal o SVM pueden ofrecer un mejor rendimiento en términos de precisión, su baja interpretabilidad limita su utilidad en entornos donde es necesario explicar claramente las decisiones.
Por esta razón, se recomienda priorizar modelos como el **Árbol de Decisión** o la **Regresión Logística** cuando el criterio de transparencia sea tan importante como la capacidad predictiva. Estos modelos permiten identificar qué variables influyen más en la clasificación y cómo se llega a cada predicción, facilitando la comunicación de los resultados a usuarios no técnicos.

8 Evaluación del coste computacional de los modelos

Además de la precisión y la interpretabilidad, es importante considerar el **coste computacional** asociado a cada modelo, tanto durante la fase de entrenamiento como en la predicción. Este coste incluye aspectos como el tiempo de cómputo, el uso de memoria y la escalabilidad.
A continuación, se presenta un análisis cualitativo del coste de los cinco modelos evaluados:

Modelo	Evaluación del coste computacional
Regresión Logística	**Muy bajo**. Entrenamiento rápido, predicción eficiente. Ideal para conjuntos de datos pequeños o medianos. Muy adecuado cuando se requiere bajo consumo de recursos.
Árbol de Decisión	**Bajo**. Rápido en entrenamiento y extremadamente rápido en predicción. Escalable a grandes volúmenes de datos y fácil de implementar.
k-Vecinos más Cercanos (k-NN)	**Medio-alto**. No requiere entrenamiento explícito, pero la fase de predicción es costosa, ya que implica calcular distancias con todos los datos del conjunto de entrenamiento. Pobre rendimiento en datasets grandes.
SVM (kernel RBF)	**Alto**. El entrenamiento es costoso, especialmente con muchos datos o atributos. La predicción también implica operaciones matemáticas complejas.
Red Neuronal Multicapa (MLP)	**Alto**. Requiere múltiples iteraciones y ajustes de pesos. El entrenamiento es intensivo y puede requerir aceleradores (GPU) en casos más grandes. La predicción es más eficiente, pero el modelo es pesado.

Comparación cualitativa del coste computacional de los modelos

Conclusión

Los modelos más eficientes desde el punto de vista del coste computacional son la **Regresión Logística** y el **Árbol de Decisión**, siendo especialmente recomendables para implementaciones en tiempo real o en sistemas con recursos limitados.
Por el contrario, modelos como la **Red Neuronal (MLP)** o el **SVM** implican un coste mayor y deben utilizarse únicamente si la mejora en precisión lo justifica, o si se dispone de recursos computacionales adecuados.

9 Conclusiones finales del proyecto

Este proyecto abordó de manera integral un problema real de predicción de expectativas académicas de estudiantes mediante técnicas modernas de **Aprendizaje Automático**, siguiendo un enfoque sistemático y orientado a la interpretación práctica de los resultados.

Logros principales

- Se realizó un **preprocesamiento exhaustivo** del conjunto de datos: limpieza, codificación, agrupación de categorías y eliminación de variables no informativas.
- Se aplicaron **análisis estadísticos** como el test de Chi-cuadrado, el coeficiente de Cramér's V y la información mutua, para guiar la selección de atributos relevantes.
- Se diseñaron **nuevas variables** mediante *ingeniería de atributos*, que capturan de forma más estructurada aspectos clave como el rendimiento previo y la motivación del estudiante.
- Se entrenaron y compararon **cinco modelos de clasificación supervisada**, utilizando herramientas avanzadas como:
  - **SMOTE** para balanceo de clases.
  - **Grid Search** para ajuste de hiperparámetros.
  - **Evaluación múltiple** en términos de precisión, interpretabilidad y coste computacional.

Resultados destacados

- El modelo con mejor rendimiento fue la **Red Neuronal Multicapa (MLP)**, con una precisión de **70.37 %**.
- Modelos como la **Regresión Logística** y el **Árbol de Decisión** ofrecieron un equilibrio óptimo entre precisión (62.96 %), interpretabilidad y eficiencia.
- Se seleccionó el modelo final considerando múltiples factores: rendimiento, coste computacional, complejidad del modelo y posibles escenarios de aplicación.

Aprendizajes clave

- La **precisión no es el único criterio**: interpretabilidad, coste y robustez son igualmente importantes.
- Técnicas como la **ingeniería de atributos** y el **sobremuestreo con SMOTE** pueden tener un impacto significativo en la calidad del modelo.
- La **validación cruzada y el ajuste de hiperparámetros** mejoran la capacidad de generalización del modelo.

Líneas de mejora futuras

- Probar modelos más avanzados como **Random Forest**, **Gradient Boosting** o **XGBoost**.
- Incluir nuevas variables contextuales o demográficas si se dispone de información adicional.
- Utilizar métricas más específicas como el **F1-score** o el **recall por clase**, especialmente en casos de desbalance.
- Evaluar la **implementación del modelo en un entorno real o simulado**, para validar su utilidad práctica en contextos educativos reales.

Cierre

Este proyecto ha demostrado cómo aplicar de forma rigurosa y estratégica los fundamentos del aprendizaje automático, integrando análisis estadístico, modelado predictivo e interpretación de resultados, todo ello con un enfoque práctico y orientado a la toma de decisiones educativas.

Referencias

[1] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). *Scikit-learn: Machine Learning in Python*. Journal of Machine Learning Research, 12, 2825–2830.

[2] Lemaître, G., Nogueira, F., & Aridas, C. K. (2017). *Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning*. Journal of Machine Learning Research, 18(17), 1–5.

[3] Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill.

[4] Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer.