Aprendizaje Automático - Modelos Predictivos

Lucía Herraiz Cano
Aprendizaje Automático
Abril 2025

Abstract

En este proyecto se desarrolla un análisis predictivo sobre el rendimiento académico de estudiantes de secundaria en dos institutos de Madrid durante el año 2005. A partir de un conjunto de datos, se construyen y comparan dos modelos para predecir la nota final del curso (T3). También se lleva a cabo un estudio de los datos y de las variables más influyentes en el desarrollo del estudiante. Este estudio busca no solo obtener predicciones precisas, sino también generar conocimiento accionable para mejorar el rendimiento estudiantil desde una perspectiva integral.

1 Exploratory data analysis (EDA)

En esta sección se analiza la estructura y distribución del conjunto de datos, identificando patrones, valores atípicos y relaciones relevantes entre variables. También se detalla el proceso de limpieza y preparación necesario para el modelado posterior.

El desarrollo detallado de estos análisis se encuentra principalmente documentado en *Exploratory_Data_Analysis*. No obstante, ciertas observaciones derivadas del análisis conjunto con modelos predictivos pueden consultarse en *Model1_Testing* y *Model2_Testing*.

1.1 Limpieza de datos

El proceso de limpieza de datos viene recogido en la función `data_cleaning_pipeline` e incluye el manejo de outliers, la imputación de valores faltantes, la corrección de errores, la codificación de las variables categóricas y la estandarización de los datos.

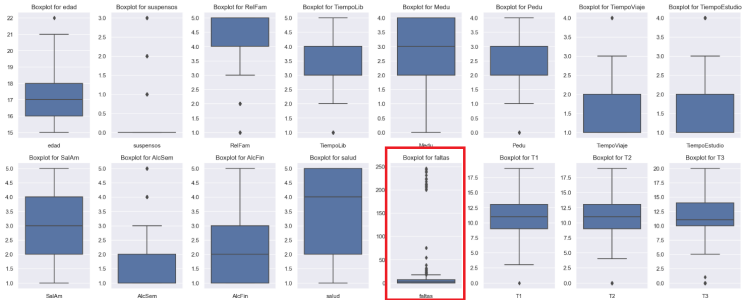

Figure 1: Boxplot de los outliers de las variables

Al estudiar los outliers, la variable más llamativa fue *faltas*. Se estableció un valor máximo de 150, correspondiente al número de días lectivos del calendario escolar estándar, considerando cualquier valor superior como erróneo. Se probaron dos estrategias adicionales, eliminar registros, e imputar los outliers por la mediana, pero ambas estrategias disminuyeron la *performance* de los modelos en

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

varios puntos. Esto muestra no sólo que *faltas* es una variable altamente relevante, sino que en este caso los outliers son muy informativos.

Sólo 5 variables tenían valores faltantes, y con el objetivo de perder la menor cantidad de datos, todos los valores se imputaron siguiendo distintas estrategias. *AlcSem, Relfam* y *TiempoEstudio* se imputaron por la moda al ser variables categóricas, y tener menos de un 3% de valores faltantes. *Medu* y *Pedu*, al tener un porcentaje más relevante de valores faltantes, tener una correlación de Pearson alta (0.653), y al ser variables con mucho peso, como se verá posteriormente, se imputaron con un regresor (IterativeImputer) que emplea el resto de datos para predecir sus valores de manera más robusta.

Se corrigieron los datos de la columna *razon* al tener claves distintas para el mismo valor ("otras", "otros").

Finalmente, tras estudiar el balance de las clases, se vió que *EstPadres, EstSup* y *apoyo* estaban claramente desbalanceadas (80%-20%). Sin embargo, dado que ninguna es una variable muy relevante, y dado que los resultados reflejan bien el balance que se suele dar en la población real, se decidió mantener los datos y tener cuidado con los modelos que le den relevancia a estas variables.

Tras limpiar los datos, se hizo un *dummy encoding*, en lugar de *one hot encoding*, sobre las variables categóricas, para no añadir columnas innecesarias al dataset. La *performance* de los modelos se vió muy beneficiada por la estandarización de las variables numéricas, lo que tiene sentido ya que los datos se interpretan por los modelos de manera similar, sin depender de su escala. Sin embargo, es importante añadir que para evitar el *data leakage*, la estandarización de los datos se llevó a cabo durante todo el proyecto tras separar los datos entre *Train* y *Test*, utilizando solo la información de los datos de *Train*.

1.2 Análisis no supervisado

Con el objetivo de comprender mejor la relevancia y relación de las variables, se implementaron técnicas de aprendizaje no supervisado.

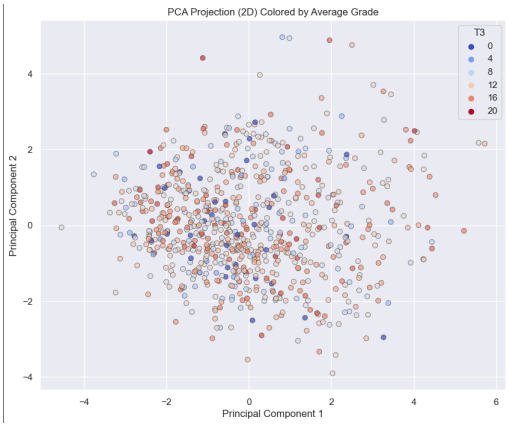

Figure 2: (a) PCA con 2 PC

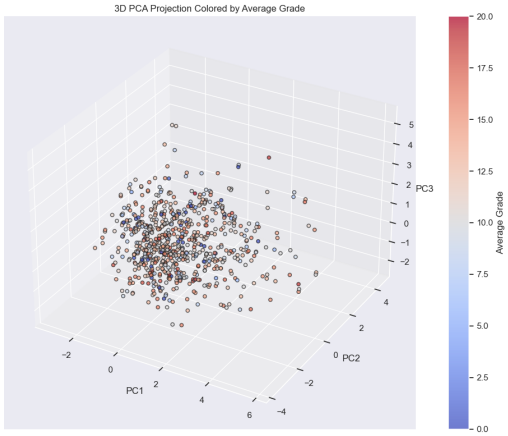

Figure 3: (b) PCA con 3 PC

Figure 4: Visualización de los datos proyectados sobre 2 y 3 componentes principales mediante PCA.

En primer lugar, se implementó **Principal Component Analysis (PCA)** con el objetivo de estudiar la reducción de dimensionalidad y analizar los componentes principales. La varianza explicada por los modelos con 2 y 3 componentes principales es inferior al 30%; no obstante, el estudio de sus loadings aporta información relevante. Para el 1º modelo, como cabía esperar, las variables más relevantes son *T1, T2* y *suspensos*, lo cual reafirma su relevancia directa en el rendimiento académico. Sin embargo, es más interesante estudiar el 2º modelo, puesto que al eliminar estas variables, las sutituyen *AlcSem, AlcFin, SalAm, TiempoLib*, entre otras, que indican que después de las **notas**, la **vida social del estudiante** es uno de los factores más relevantes para su desempeño académico. Además de estos, un tercer grupo de variables de peso que aparecen en los PC son *Medu, Pedu, Relfam* y, empleando

Recursive Feature Elimination (RFE), se obtienen también *Mtrab_docencia* y *Ptrab_docencia*, lo cual pone de manifiesto que el **entorno familiar** es otro de los factores más influyentes en las notas.

Sin embargo, PCA requiere 16 componentes para explicar el 80% de la varianza. Al combinarlo con modelos como SVR o regresión lineal, su rendimiento disminuyó, por lo que se descartó su uso más allá de la exploración inicial. Este comportamiento podría deberse a la naturaleza lineal de PCA, incapaz de capturar relaciones no lineales en los datos. Por ello, se probaron técnicas no lineales como **ISOMAP** y **Kernel-PCA**, que operan en espacios transformados mediante distancias geodésicas o kernels y que pueden capturar relaciones más complejas entre los datos. No obstante, su combinación con modelos predictivos redujo el rendimiento entre un 20-30%,[1] Podemos conluir que la reducción de dimensionalidad implica la pérdida de información relevante para la predicción. Las variables no son fácilmente separables y por ello, finalmente, se empleó el **dataset completo** para el entrenamiento de los modelos.

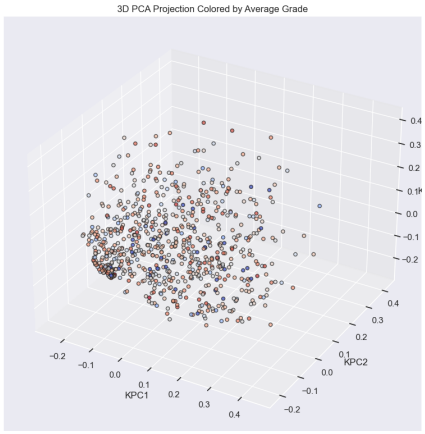

Figure 5: *
(a) Kernel-PCA con 3 PC

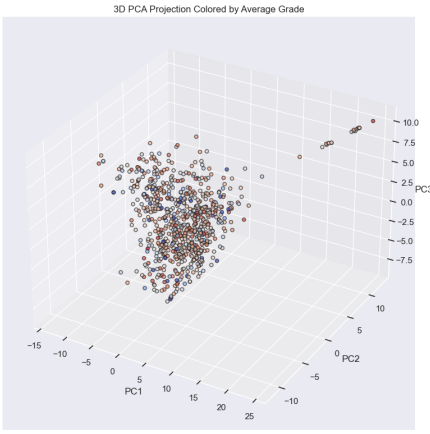

Figure 6: *
(b) ISOMAP con 3 PC

Figure 7: Aplicación de Kernel-PCA e ISOMAP con 3 PC

Asimismo, se aplicaron técnicas de **Clustering** a los datos (combinadas con PCA). Utilizando la métrica del *Elbow Method* y de la *Silueta*, se obtuvo el número óptimo de clusters (2-3), que coincide con los grupos de variables relevantes identificados anteriormente. No obstante, los valores obtenidos tras aplicar K-Means (Silueta: 0.233; Índice de Davies-Bouldin: 1.341) indican una calidad media en la segmentación, reafirmando la dificultad al separar los datos. Experimentalmente, se probó a añadir una nueva variable a los datos indicando la pertenecia a los clusters para entrenar a un regresor lineal, pero esto no modificó su *performance*, por lo que finalmente, usando la filosofía de la *Navaja de Occam*, se descartó el uso activo del clustering en los modelos.

Aunque las técnicas de aprendizaje no supervisado no fueron incorporadas directamente en los modelos finales, su aplicación contribuyó significativamente a una comprensión más profunda de la estructura y relaciones entre las variables del conjunto de datos, lo cual resultó de gran valor para el desarrollo y justificación del enfoque predictivo adoptado. La conclusión más relevante, utilizar el dataset entero sin reducir la dimensionalidad, llevó al descarte de modelos como KNNeighbours, que podían verse afectados por el *curse of dimensionality*.

1.3 Análisis adicional

En paralelo al análisis anterior, y con el objetivo de ampliar el conocimiento sobre las variables, se utilizaron las capacidades explicativas de distintos modelos para evaluar la importancia relativa de cada característica.

Se estudió la representación conjunta de cada par de variables, empleando como código de color las distintas clases (0-20) para identificar posibles relaciones entre los datos[2], pero sólo se encontró

[1] Exceptuando Logistic Regression, dónde su combinación con Kernel-PCA lo aumentó.
[2] Ver la matriz completa en el fichero *Exploratory_Data_Analysis*

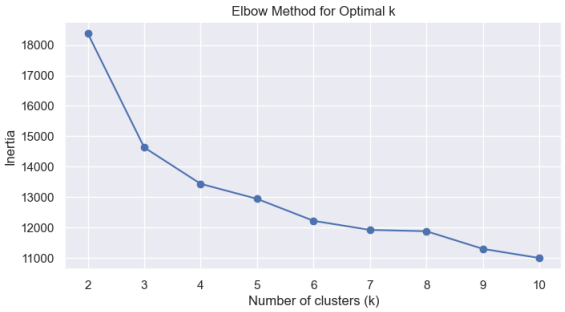

Figure 8: (a) Elbow Method

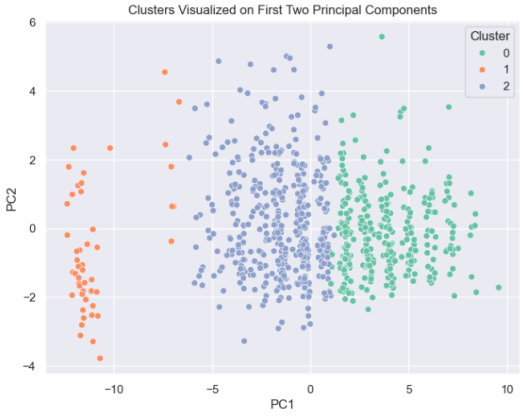

Figure 9: (b) Separación en clusters

un patrón claro con *T1* y *T2*, demostrando de nuevo que son variables muy significativas para la predicción.

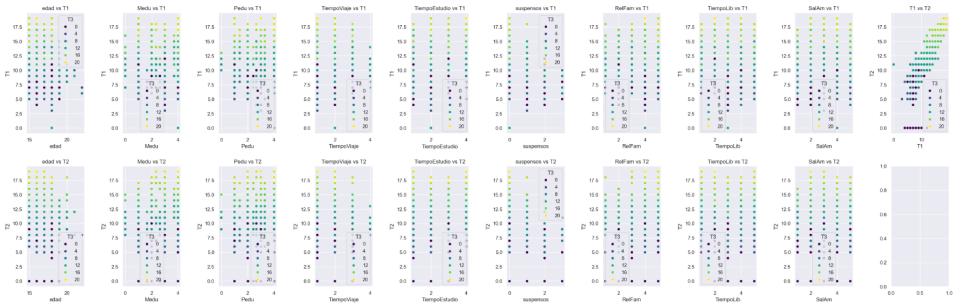

Figure 10: Scatter Plot con T1 y T2

Uno de los algoritmos con mejor desempeño en ambos modelos, como se verá más adelante, fue **Random Forest**. Vemos que las variables a las que asigna un mayor peso coinciden en su mayoría con las obtenidas con PCA y con otros métodos. Asimismo, vemos la asignación de mayor peso a algunas variables nuevas (*razon_otras, apoyo_si*, etc.), pero esto puede depender del corte de los datos, ya que los árboles suelen tener una varianza alta.
Por último, podemos ver la relevancia de T1 y T2 reflejada en que T2 tiene asignada casi la totalidad de la *Feature importance*.
Cabe recalcar que T1 y T2 están muy correlacionadas (Pearson: 0.863), por lo que una asignación de pesos desequilibrada, como en este modelo, no significa que una de ellas sea significativamente más importante que la otra.

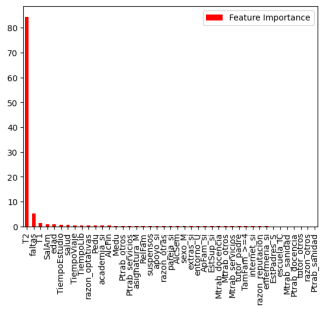

Figure 11: Feature importance en Random Forest

A pesar de que el modelo de **regresión** logística no tiene el mejor rendimiento, al ser un clasificador, es interesante emplearlo, junto con regularización Lasso, para ver la evolución de los pesos de las variables para cada una de las clases, lo que aporta una mayor profundidad en el análisis. El estudio se ha realizado sin tener en cuenta *T1* y *T2* y podemos ver que para las notas más bajas las variables relacionadas con la vida social y la familia son las más importantes, mientras que para las notas más altas, influyen variables más variadas. Es interesante ver que para las notas más altas (Clase 20), muchas de las variables relacionadas con la vida social (*AlcSem, AlcFin* y *faltas*) se comportan de manera casi idéntica, lo que refleja un grado de relación grande, que ya hemos visto en el análisis anterior.

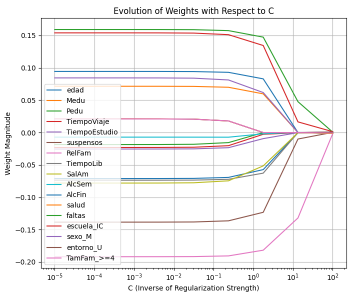

Figure 12: (a) Clase 1

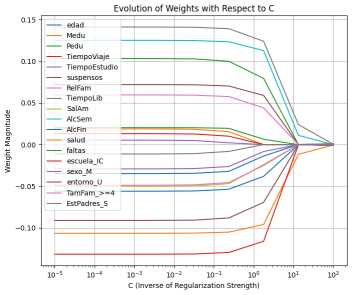

Figure 13: (b) Clase 10

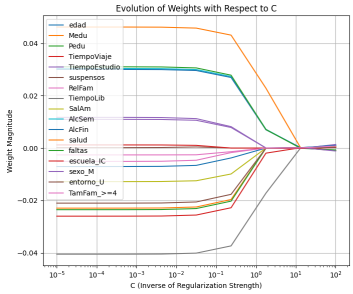

Figure 14: (c) Clase 20

Figure 15: Feature Importance para tres clases

2 Implementación y comparación de los dos modelos predictivos

En esta sección se presentan los algoritmos con mayor rendimiento para ambos enfoques, junto con los criterios que justifican la elección de los modelos finales.

El desarrollo completo del análisis se encuentra en *notebooks*. Las pruebas realizadas sobre los datos con los modelos iniciales se encuentran en *Model1_Testing* y *Model2_Testing*. La selección de los parámetros de los modelos finales por validación cruzada se puede consultar en *Final_Model1_Tuning* y *Final_Model2_Tuning*. Por último, las métricas mostradas en el informe se obtuvieron en *Metrics_Evaluator*.

2.1 Análisis Técnico de los Modelos Finales

Los modelos finales fueron todos programados manualmente.[3] **Linear Regression**, **Logistic Multiclass Regressor**, **Support Vector Regressor (SVR)**, **Bagging Tree Ensemble**, **Random Forest Tree Ensemble** y **Boosting Tree Ensemble**.

El propósito de este informe no es profundizar en la explicación de los modelos, por lo que me limitaré a presentar brevemente la idea principal de aquellos que no han sido abordados en clase.

Variantes de Linear Regression: Inicialmente, el modelo de Regresión Lineal presentó el mejor rendimiento entre las alternativas evaluadas. Con el objetivo de mejorar su desempeño, diseñé distintas variantes. Los dos modelos que lograron incrementar la *performance* fueron:

- **LR_Ensemble**: Un Ensemble de modelos de regresión lineal utilizando Random Forest como método de agregación.
- **LR_Relations**: Una versión modificada que incorpora relaciones específicas identificadas en los datos (EDA). Esta variante incrementa de manera consistente los *scores* de LR en varios puntos.

Stacking Regressor: Un Stacking Regressor, definido brevemente, combina múltiples modelos de regresión (*estimators*) y utiliza sus predicciones como entradas para un meta modelo. Esto permite al meta modelo aprender cómo combinar las predicciones base para minimizar el error global. El código diseñado crea un Stacking Regressor de dos capas, pero se pueden emplear más para realizar otras funciones, como seleccionar varibles, eliminar ruido, etc.

Los modelos que he introducido como *estimators* son aquellos que han mostrado el mejor desempeño sobre el conjunto de datos (*Gradient Boosting, Random Forest* y *Bagging*), con los hiperparámetros optimizados por validación cruzada.

[3] Los códigos de los modelos pueden ser consultados en el módulo *Models*.

Figure 16: Funcionamiento del Stacking Regressor

2.2 Modelo 1: Enfoque Predictivo con la Información Completa del Dataset

Tras analizar múltiples modelos, se concluyó que los regresores superan consistentemente a los clasificadores, con una mejora media del 20% en las métricas. Por ello, se descartó en una fase inicial continuar con clasificadores y se priorizó profundizar en modelos de regresión.

Los modelos de clasificación probados junto con su *Accuracy* media fueron **Logistic Regression** (0.28), **Logistic Regression con Kernel-PCA** (0.57), **Support Vector Classifier (SVC)** (0.31), **Random Forest Classifier** (0.46) y **Bagging Classifier** (0.45). Se programaron manualmente Logistic Regression, Random Forest Classifier y Bagging Classifier.

La Tabla 1 muestra los mejores modelos de regresión. Para garantizar la consistencia de las métricas, y evitar depender de puntuaciones obtenidas en particiones específicas del conjunto de datos, los valores presentados se corresponden con los promedios obtenidos tras entrenar más de 1200 modelos por cada técnica, utilizando datos reordenados aleatoriamente en cada iteración.

Otros modelos de regresión entrenados, junto con sus valores R^2 medios fueron **Support Vector Regressor (SVR) con Kernel-PCA** (0.80), **Linear Regression con ISOMAP** (0.61) y **Linear Regression con Kernel-PCA** (0.80). Todos ellos fueron descartados para el proceso de validación cruzada ya que tienen una *performance* inferior. Los modelos de árboles individuales fueron descartados a favor de los *Ensembles* (TE).

Table 1: Modelos de regresión (Modelo 1)

Comparativa entre modelos					
Nombre	R² Score Test	MAE	MSE	R² Score Train	ΔScore
Linear Regression (LR)	0.8301 ± 0.0335	0.9968	2.6484	0.853	0.0229
LR con relaciones	0.8338 ± 0.0327	1.0010	2.6682	0.854	0.0207
Ensemble de LR	0.8320 ± 0.0334	1.0011	2.6847	0.852	0.0200
Stacking Regressor	0.8664 ± 0.0308	**0.8890**	**2.1194**	0.936	0.0696
Bagging TE.	0.8531 ± 0.0366	0.9271	2.3367	0.979	0.1259
Random Forest TE.	0.8552 ± 0.0343	0.9119	2.2354	0.980	0.1248
Gradient Boosting TE.	**0.8682 ± 0.0304**	0.8945	2.1335	0.926	0.0578
SVR	0.8331 ± 0.0371	0.9265	2.7643	0.838	**0.0049**

Podemos observar que **Gradient Boosting Regressor** presenta el mejor rendimiento general, mientras que **Stacking Regressor** tiene los errores más bajos. No obstante, la diferencia entre ambos modelos es inferior al 1%, lo que indica un desempeño comparable. Además, al contrario que Bagging y Random Forest, tienen una ΔScore[4] muy baja, lo que indica poco *overfitting*. En base a los resultados obtenidos, se seleccionó el **Stacking Regressor** como modelo final, dado su foco hacia la minimización del error de predicción. Además, al integrar Gradient Boosting junto con otros de los modelos con mejor rendimiento individual, se logra mantener un R² score comparable al de estos modelos, beneficiándose al mismo tiempo de la robustez y capacidad de generalización que ofrece el enfoque de stacking.

Tras realizar un último análisis de la importancia de las variables en el modelo final usando *SHAP*, podemos ver que todas las observaciones previas al entrenamiento de los modelos se han cumplido. La mayor parte de las variables de peso están relacionadas con las **notas**, y podemos ver el efecto positivo (*RelFam, salud*) o negativo (*asignatura_M, faltas*) que tienen en el target.

[4] ΔScore = R² Score Train - R² Score Test

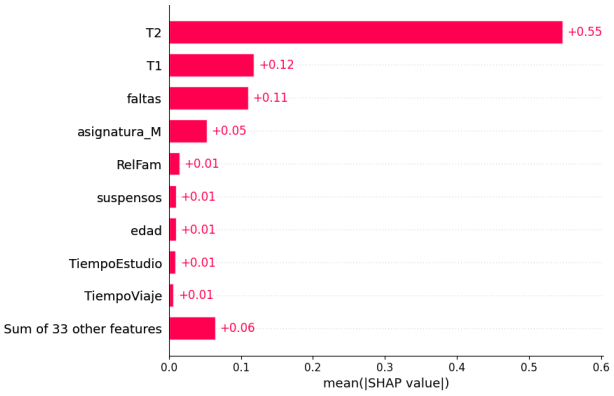

Figure 17: (a) Feature importance

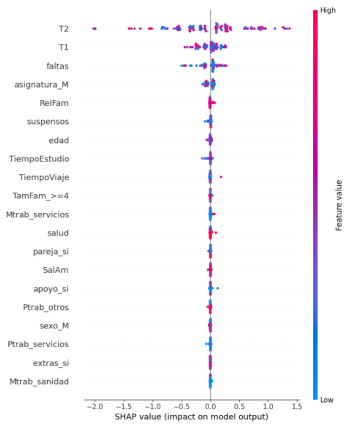

Figure 18: (b) Gráfico Shap

2.3 Modelo 2: Enfoque Predictivo sin las variables T1 y T2

Al eliminar las variables de más peso, los modelos reducen a la mitad su poder predictivo, pero se mantiene la superioridad de los regresores frente a los clasificadores.

En esta ocasión, dados los resultados anteriores, los únicos clasificadores probados, junto con su *Accuracy* media fueron **Logistic Regression** (0.237) y **Classification Stack**[5] (0.137). En el Modelo 1, la *performance* del regresor logístico mejoró al combinarlo con **Kernel-PCA**, pero con el 2º Modelo, se ha disminuido su *Accuracy* a 0.0933. Esto se debe probablemente a que los primeros componentes principales capturaban en gran medida la varianza explicada por las variables T1 y T2. Al eliminarlas, la estructura de varianza se ve alterada significativamente, lo que reduce la capacidad del modelo.

Adicionalmente a los regresores mostrados en la Tabla 2, se probaron a su vez **Linear Regression** (0.21), **Linear Regression con Clustering** (0.18), **Linear Regression con Kernel-PCA** (0.11) y **LR_Relations** (0.23). Dado que en el modelo anterior Gradient Boosting tuvo la mejor *performance*, se investigaron métodos optimizados como **CatBoosting** (0.26), **XGBoost** (0.30) y **Light GBM** (0.23). Todos estos métodos de Boosting están optimizados para tratar con características como datasets grandes, muchas variables categóricas o gran dimensionalidad, características que coinciden con nuestro dataset, pero tras probarlos, fueron descartados al tener una *performance* inferior a Gradient Boosting.

Los resultados mostrados, al igual que con el Modelo 1, son el promedio de los resultados tras entrenar más de 900 modelos. Podemos observar que en este caso, el **Stacking Regressor** es también el modelo con mejores resultados.

Table 2: Modelos de regresión (Modelo 2)

Comparativa entre modelos					
Nombre	R^2 Score Test	MAE	MSE	R^2 Score Train	ΔScore
Linear Regression	0.1999 ± 0.0625	2.5932	12.763	0.302	**0.1021**
Stacking Regressor	$\mathbf{0.3153 \pm 0.0699}$	2.4093	**10.717**	0.783	0.4677
Bagging TE.	0.2998 ± 0.0806	2.3954	10.971	0.904	0.6042
Random Forest TE.	0.3056 ± 0.0768	**2.3881**	10.916	0.905	0.5994
Gradient Boosting TE.	0.3005 ± 0.0810	2.4709	11.025	0.681	0.3805
SVR	0.2856 ± 0.0657	2.4193	11.423	0.730	0.4444

Los resultados obtenidos son muy similares a los anteriores en cuanto a la comparación de modelos. En este caso se seleccionó el **Stacking Regressor** utilizando los mismos criterios que anteriormente, robustez y *performance*. Sin embargo, este modelo no debería ser usado para realizar predicciones,

[5]Usando Logistic Regression, SVC y Random Forest Classification

puesto que no es capaz de generalizar bien. El estudio de este modelo se debería limitar al análisis de las variables importantes en ausencia de notas previas.

Por concluir el análisis, se han usado dos gráficas *SHAP* para estudiar el efecto de las variables en la predicción final. La mayoría de las variables más relevantes coinciden con las obtenidas en el Modelo 1. Podemos estudiar el efecto positivo (*EstSup_si*) o negativo (*apoyo_si, suspensos, faltas*) que coincide con lo estudiado, sin embargo, este análisis es menos preciso que en el Modelo 1 puesto que el modelo tiene más varianza, y los resultados de los gráficos dependen mucho del corte de los datos.

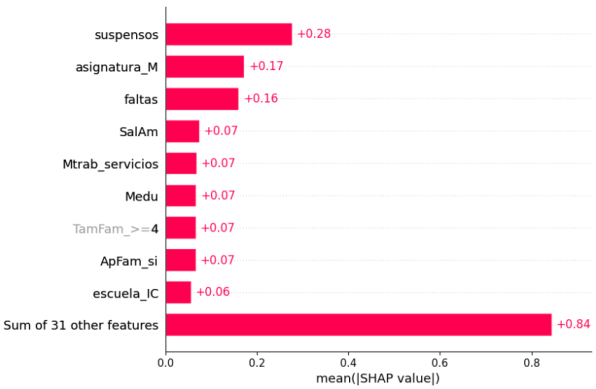

Figure 19: (a) Feature importance

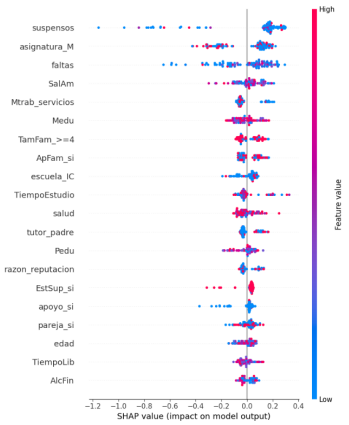

Figure 20: (b) Gráfico Shap

3 Conclusiones accionables

Para concluir, se expondrán algunas de las acciones que se pueden llevar a cabo para mejorar el rendimiento de los estudiantes en base al análisis completo del proyecto.

El análisis de los datos hecho en tres partes (EDA y conclusiones de los modelos 1 y 2) coincide en sus conlusiones consistentemente. Influyen tres grupos principales de variables a la hora de predecir el desempeño de los estudientes, los **factores académicos**, la **vida social** del estudiante, y su **entorno familiar**. Con esta información, los centros educativos pueden enfocar sus medidas en estas áreas.

En primer lugar, dado que los factores académicos son los más influyentes, los institutos deberían controlar una serie de métricas sobre sus estudiantes (notas, faltas, suspensos) e identificar rápidamente cualquier comportamiento que pueda empeorarlas. En caso de detectarlo, tratar de corregirlos lo antes posible.

El segundo factor más influyente es la vida social, sensibilizar al alumnado sobre la importancia de equilibrar vida social y estudio puede favorecer su rendimiento. Este tipo de medidas deberían aplicarse en cursos más bajos para que tengan efecto en el desarrollo de hábitos del estudiante.

Por último, el último grupo de variables importantes identificadas es entorno familiar, promover políticas de conciliación familiar podría tener un impacto positivo en sus resultados académicos. Concienciar a los padres sobre su efecto en la educación de sus hijos es clave.

References

[1] Javier Béjar. *Strategies and Algorithms for Clustering Large Datasets: A Review.* Universidad Politécnica de Cataluña. `https://upcommons.upc.edu/bitstream/handle/2117/23415/R13-11.pdf`

[2] Alfonso Cervantes Barragan. (2024). *Interpreting and Validating Clustering Results with K-Means*. Medium. `https://medium.com/@a.cervantes2012/interpreting-and-validating-clustering-results-with-k-means-e98227183a4d`

[3] Connie Zhou. (2023). *Unraveling Data Patterns with Isomap: A Guide to Dimensionality Reduction — Part 4.* `https://medium.com/@conniezhou678/unraveling-data-patterns-with-isomap-a-guide-to-dimensionality-reduction-part-4-1d774eee69a5`

[4] (2025). *Gradient Boosting in ML*. Geeks for Geeks. https://www.geeksforgeeks.org/ml-gradient-boosting/

[5] Casper Hansen. (2020). *Stack machine learning models: Get better results*. IBM Developers. https://developer.ibm.com/articles/stack-machine-learning-models-get-better-results/